

**DEVELOPMENT OF A COMPUTER PROGRAM FOR THE  
VERIFICATION AND VALIDATION OF NUMERICAL SIMULATIONS IN  
ROADSIDE SAFETY**

by

Mario Mongiardini

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

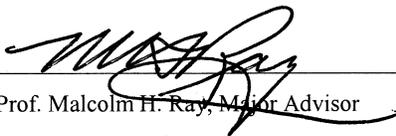
Degree of Doctor of Philosophy

in

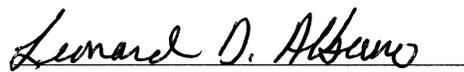
Civil Engineering

April 29, 2010

APPROVED:



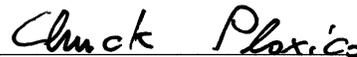
Prof. Malcolm H. Ray, Major Advisor  
(CEE Department)



Prof. Leonard D. Albano, Committee Member  
(CEE Department)



Prof. Roberto Pietroforte, Committee Member  
(CEE Department)



Dr. Chuck A. Plaxico, Committee Member  
(Battelle Memorial Institute)



Prof. Tahar El-Korchi, Department Head  
(CEE Department)

**© Copyright by Mario Mongiardini, 2010**

**All Rights reserved**

*To my parents, my uncle,  
my grandfather and my girlfriend Chiara*

*" Theory is when you know all and nothing works.  
Practice is when all works and nobody knows why.  
In this case we have put together theory and practice:  
nothing works.....and nobody knows why! "*

**(Albert Einstein)**

# Abstract

Roadside safety hardware has traditionally been approved on the basis of full-scale crash tests. In recent years, nonlinear dynamic Finite Element (FE) programs like LS-DYNA, PAM-Crash or ABAQUS Explicit have been widely used in evaluating new or improved design of roadside hardware. Although a powerful tool, numerical models must be properly verified and validated in order to provide reliable results. Typically, the verification and validation (V&V) process involves a visual comparison of two curves and is based on a purely subjective judgment. This research investigated the use of comparison metrics, which are mathematical measures that quantify the level of agreement between two curves, for comparing simulation and experimental outcomes in an objective manner.

A computer program was developed in Matlab<sup>®</sup> to automatically evaluate most of the comparison metrics available in literature. The software can be used to preprocess and compare either single or multiple channels, guiding the user through friendly graphical interfaces. Acceptance criteria suitable to represent the typical scatter of experimental tests in roadside safety were determined by comparing ten essentially identical full-scale vehicle crash tests.

The robustness and reliability of the implemented method were tested by comparing the qualitative score of the computed metrics for a set of velocity waveforms with the corresponding subjective judgment of experts. Moreover, the implemented method was applied to two real validation cases involving a numerical model in roadside safety and a model in biomechanics respectively. Eventually, the program showed to be an effective tool to be used for assessing the similarities and differences between two curves and, hence, for assisting engineers and analysts in performing verification and validation activities objectively.

Keywords: Verification, Validation, Finite Element, LS-DYNA, Full-Scale Crash Tests.

## Acknowledgments

*I wish to express my sincere gratitude to my supervisor, Prof. Malcolm H. Ray, for his advice, encouragement and patience in guiding me thorough the research. Also, I would like to thank him for introducing me to the roadside safety community and giving the opportunity to attend conferences.*

*I am grateful to my committee members, Prof. Roberto Pietroforte, Prof. Leonard D. Albano and Dr. Chuck A. Plaxico of Battelle Memorial Institute for their helpful suggestions all the way along my research. A special thank goes to Dr. Plaxico for his insightful comments related to the testing of the computer program developed in this research.*

*I thank Dr. Schwer and Mr. David Moorcroft for providing useful information about comparison metrics and data I used during my research.*

*Thanks are due to Prof. Marco Anghileri of Polytechnic of Milan, Italy, who mentored me during my Master thesis in Italy and made this long journey possible.*

*Also, I would like to thank the department chair, Prof. Tahar El-Korchi, for the financial support provided in the last five months of the dissertation and Mr. Don Pellegrino for his help in the laboratory activities. Special thanks are due to Agata Lajoie for taking care of many administrative details.*

*I am thankful to my office colleagues, Christine E. Carrigan, Douglas Heath, Sebastian Mendes and Tejas Ruparel, and to all my friends, Andrew, Hala, Leila, Jose, Karen, Louis and Rodrigo, who made my stay at WPI a memorable experience.*

*A special thank goes to Sia Najafi and Adriana Hera for the support with computer problems and the help provided with the software training sessions.*

*Least, but certainly not the last, I want to express the most sincere gratitude to my girlfriend Chiara, who has shared with me all the good and bad moments of these years, providing an unconditional support.*

*This dissertation is dedicated to my girlfriend Chiara, my parents, my brothers, my uncle and grandfather for their endless love, understanding and support.*

*Mario Mongiardini*

# Table of Contents

<b>ABSTRACT .....</b>	<b>I</b>
<b>ACKNOWLEDGMENTS .....</b>	<b>II</b>
<b>TABLE OF CONTENTS .....</b>	<b>III</b>
<b>LIST OF FIGURES .....</b>	<b>VIII</b>
<b>LIST OF TABLES .....</b>	<b>XII</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>XIV</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>- 1 -</b>
1.1    ROLE OF NUMERICAL SIMULATIONS IN ROADSIDE SAFETY .....	- 1 -
1.2    VERIFICATION AND VALIDATION .....	- 1 -
1.3    OBJECTIVE OF THE RESEARCH .....	- 4 -
<b>CHAPTER 2 LITERATURE REVIEW .....</b>	<b>- 5 -</b>
2.1    USE OF NUMERICAL ANALYSIS IN ROADSIDE SAFETY .....	- 5 -
2.1.1 <i>The issue of verification and validation</i> .....	- 11 -
2.2    COMPARISON METRICS.....	- 12 -
2.2.1 <i>Deterministic metrics</i> .....	- 13 -
2.2.1.1    Domain-specific metrics .....	- 13 -
2.2.1.2    Shape-comparison metrics.....	- 16 -
2.2.1.2.1    Single-value metrics .....	- 16 -
2.2.1.2.1.1    NARD metrics .....	- 16 -
2.2.1.2.1.2    Analysis-of-Variance (ANOVA) metrics .....	- 22 -
2.2.1.2.1.3    Other single-value metrics .....	- 27 -
2.2.1.2.2    Multiple-value metrics .....	- 28 -
2.2.1.2.2.1    MPC metrics.....	- 28 -
2.2.1.2.2.2    Global Evaluation Method (GEM) .....	- 38 -
2.2.2 <i>Stochastic metrics</i> .....	- 39 -
2.3    COMPUTER PROGRAMS FOR THE EVALUATION OF COMPARISON METRICS .....	- 41 -
2.3.1 <i>Test Report Assessment Program (TRAP)</i> .....	- 42 -
2.3.2 <i>Curve Analyzer</i> .....	- 42 -
2.3.3 <i>TNO software</i> .....	- 43 -
2.3.4 <i>ModEval</i> .....	- 44 -

2.4	SYNTHESIS OF THE LITERATURE REVIEW .....	- 45 -
2.4.1	<i>Classification of comparison metrics</i> .....	- 45 -
2.4.2	<i>Computer codes</i> .....	- 52 -
<b>CHAPTER 3 METHODOLOGY .....</b>		<b>- 54 -</b>
3.1	INITIAL SCREENING OF COMPARISON METRICS SUITABLE FOR V&V OF NUMERICAL MODELS .....	- 54 -
3.2	IMPLEMENTATION OF A SOFTWARE FOR CALCULATING COMPARISON METRICS .....	- 55 -
3.3	CODE VERIFICATION .....	- 55 -
3.4	ACCEPTANCE CRITERIA AND SUGGESTED METRICS FOR V&V IN ROADSIDE SAFETY .....	- 56 -
3.5	APPLICATION OF THE IMPLEMENTED METHOD TO THREE CASES .....	- 56 -
3.6	CONCLUSIONS .....	- 56 -
<b>CHAPTER 4 COMPUTER PROGRAM FOR COMPARING CURVES .....</b>		<b>- 57 -</b>
4.1	PROGRAM OVERVIEW .....	- 57 -
4.1.1	<i>Organization of the chapter</i> .....	- 59 -
4.2	DESCRIPTION OF THE TASKS .....	- 59 -
4.2.1	<i>Input and preprocessing</i> .....	- 59 -
4.2.1.1	Input of data .....	- 59 -
4.2.1.2	Preprocessing .....	- 60 -
4.2.1.2.1	Curve scaling .....	- 60 -
4.2.1.2.2	Manual trimming and curve shift .....	- 61 -
4.2.1.2.3	Filtering .....	- 61 -
4.2.1.2.4	Shift and/or drift correction .....	- 63 -
4.2.1.2.5	Resampling and automatic trimming .....	- 65 -
4.2.1.2.6	Curve synchronization .....	- 66 -
4.2.2	<i>Metrics calculation</i> .....	- 67 -
4.2.2.1	Metrics history and comparison on user-defined sub-intervals .....	- 69 -
4.2.2.2	Multiple channels .....	- 70 -
4.2.2.2.1	Peak values of the curves .....	- 71 -
4.2.2.2.2	Area of the curves .....	- 71 -
4.2.2.2.3	Momentum of the vehicle .....	- 71 -
4.2.2.2.4	Advantages and disadvantages of each weighting method .....	- 72 -
4.2.3	<i>Results output</i> .....	- 73 -
4.3	PROGRAM IMPLEMENTATION .....	- 74 -
4.3.1	<i>Programming structures used in the code</i> .....	- 75 -
4.3.1.1	GUI's and data structures .....	- 75 -
4.3.1.2	Data structures used in the code .....	- 75 -
4.3.1.3	Loops implemented in the code .....	- 78 -

4.3.1.4	Sign convention for flow charts.....	- 79 -
4.3.2	<i>Input and preprocessing operations</i> .....	- 79 -
4.3.2.1	Filtering .....	- 83 -
4.3.2.2	Shift/drift.....	- 85 -
4.3.2.3	Resampling and trimming .....	- 87 -
4.3.2.4	Synchronization .....	- 89 -
4.3.3	<i>Single and multiple channel computation (comparison)</i> .....	- 93 -
4.3.3.1	Peak values.....	- 94 -
4.3.3.2	Area of curves .....	- 94 -
4.3.3.3	Inertial properties .....	- 94 -
4.3.4	<i>Output of results</i> .....	- 95 -
4.3.4.1	Metrics time histories .....	- 96 -
4.3.4.1.1	Data structure of the variable containing the results .....	- 96 -
4.3.4.1.2	Creation of the output file .....	- 97 -
4.3.4.2	Metrics final values .....	- 99 -
4.3.4.2.1	Data structure of the variable containing the results .....	- 99 -
4.3.4.2.2	Creation of the output file .....	- 102 -
4.3.5	<i>Configuration file</i> .....	- 105 -
4.3.6	<i>Logical Organization of the code</i> .....	- 107 -

**CHAPTER 5 CODE VERIFICATION.....- 110 -**

5.1	VERIFICATION OF THE METRICS IMPLEMENTATION.....	- 110 -
5.2	VERIFICATION OF THE CODE MAIN FRAME.....	- 112 -
5.2.1	<i>Input and storage of the curves</i> .....	- 112 -
5.2.2	<i>Preprocessing</i> .....	- 112 -
5.2.2.1	Filtering .....	- 113 -
5.2.2.2	Resampling/Trimming .....	- 114 -
5.2.2.3	Synchronization .....	- 115 -
5.2.2.4	Shift/Drift correction .....	- 118 -
5.2.3	<i>Computation of metrics using multiple channels</i> .....	- 120 -
5.2.3.1	Weighting factors .....	- 120 -
5.2.3.2	Weighted average .....	- 122 -
5.2.4	<i>Output of results</i> .....	- 124 -

**CHAPTER 6 ACCEPTANCE CRITERIA FOR ROADSIDE SAFETY.....- 126 -**

6.1	DESCRIPTION OF FULL-SCALE CRASH TESTS.....	- 127 -
6.2	ANALYSIS OF THE REPEATED TESTS .....	- 129 -

6.2.1	<i>Preprocessing of curves</i> .....	- 130 -
6.2.1.1	Filtering .....	- 131 -
6.2.1.2	Shift correction.....	- 131 -
6.2.1.3	Resampling.....	- 131 -
6.2.1.4	Synchronization.....	- 131 -
6.2.1.5	Trimming.....	- 135 -
6.2.2	<i>Statistical analysis of the residuals</i> .....	- 135 -
6.2.2.1	90th percentile confidence envelope.....	- 135 -
6.2.2.1.1	Statistical analysis of values at each sampling point.....	- 136 -
6.2.2.1.2	Time history of the residuals.....	- 137 -
6.2.2.2	Residual distribution .....	- 144 -
6.2.3	<i>Comparison of curves</i> .....	- 153 -
6.2.3.1	Case I: Comparison versus the average curve .....	- 153 -
6.2.3.2	Comparison versus specific curves of the set.....	- 158 -
6.2.3.2.1	Case II: First curve of each sets and combination of sets.....	- 159 -
6.2.3.2.2	Case III: Last curve of each sets and the combination of sets .....	- 163 -
6.2.4	<i>Analysis of the velocity time histories</i> .....	- 169 -
6.2.4.1	90th percentile confidence envelope.....	- 169 -
6.2.4.2	Comparison of curves.....	- 171 -
6.2.4.3	Residual distribution .....	- 174 -
6.2.5	<i>Discussion of the results</i> .....	- 177 -
6.3	ACCEPTANCE CRITERIA AND PROPOSED METRICS FOR ROADSIDE SAFETY.....	- 180 -

**CHAPTER 7 APPLICATION CASES..... - 184 -**

7.1	COMPARISON OF VELOCITY WAVEFORMS .....	- 184 -
7.1.1	<i>Subject matter expert opinions</i> .....	- 185 -
7.1.2	<i>Metrics values</i> .....	- 186 -
7.1.3	<i>Comparison of metrics values and experts opinions</i> .....	- 187 -
7.1.3.1	Error metrics .....	- 190 -
7.1.3.2	Agreement metrics.....	- 191 -
7.1.3.3	ANOVA metrics.....	- 192 -
7.2	VALIDATION OF A FE MODEL FOR FULL-SCALE CRASH TEST .....	- 193 -
7.2.1	<i>Numerical model</i> .....	- 194 -
7.2.1.1	Vehicle.....	- 194 -
7.2.1.2	Barrier .....	- 196 -
7.2.2	<i>Simulation results</i> .....	- 196 -
7.2.2.1	Energy and mass conservation.....	- 196 -
7.2.2.2	Qualitative validation .....	- 198 -

7.2.2.3	Quantitative validation.....	- 200 -
7.2.2.3.1	Metrics computed for each single channel .....	- 204 -
7.2.2.3.2	Metrics computed on multiple channels.....	- 207 -
7.2.2.3.2.1	Peak values .....	- 209 -
7.2.2.3.2.2	Area I and Area II .....	- 210 -
7.2.2.3.2.3	Inertial properties .....	- 212 -
7.2.2.3.2.4	Resultant time histories .....	- 213 -
7.2.3	<i>Analysis of results</i> .....	- 214 -
7.3	VALIDATION OF A KNEE-THIGH-HIP FE MODEL .....	- 217 -
7.3.1	<i>KTH FE model</i> .....	- 217 -
7.3.2	<i>Quantitative validation</i> .....	- 218 -
7.3.3	<i>Analysis of results</i> .....	- 219 -
<b>CHAPTER 8 CONCLUSIONS AND FURTHER RESEARCH .....</b>		<b>- 220 -</b>
8.1	SUMMARY OF THE RESEARCH.....	- 220 -
8.2	COMPARISON METRICS AND ACCEPTANCE CRITERIA .....	- 221 -
8.3	DISCUSSION OF RESULTS .....	- 223 -
8.3.1	<i>Effectiveness of comparison metrics</i> .....	- 223 -
8.3.2	<i>Validation of numerical models</i> .....	- 224 -
8.4	FURTHER RESEARCH.....	- 225 -
<b>REFERENCES .....</b>		<b>- 227 -</b>
<b>APPENDIX A .....</b>		<b>- 234 -</b>
<b>APPENDIX B.....</b>		<b>- 306 -</b>
<b>APPENDIX C.....</b>		<b>- 396 -</b>

# List of Figures

FIGURE 1.1: REPRESENTATION OF THE ACKERMAN CONCEPT [3].	- 2 -
FIGURE 2.1: PHOTOGRAPH (LEFT) AND CORRESPONDING FE MODEL (RIGHT) OF THE PRE-COLLISION WITH A SIGN POST [11].	- 6 -
FIGURE 2.2: QUALITATIVE COMPARISON OF PICKUP TRUCK IMPACT WITH AN F-SHAPE PORTABLE CONCRETE BARRIER [21].	- 9 -
FIGURE 2.3: MAIN CLASSIFICATION OF COMPARISON METRICS.	- 13 -
FIGURE 2.4: COMPARISON OF MEASURED VELOCITY WAVE FORM WITH THREE SIMULATION RESULTS [43].	- 35 -
FIGURE 2.5: RUSSELL METRICS FOR 30 ACCELEROMETERS IN A SHIP BLAST MODEL VALIDATION [53].	- 36 -
FIGURE 2.6: RUSSELL COMBINED METRIC PLOTTED VERSUS THE LONGITUDINAL POSITION OF THE SENSOR IN A SHIP BLAST VALIDATION ACTIVITY [53].	- 37 -
FIGURE 2.7: EFFECT OF SOIL AND WOOD MATERIAL PARAMETER VARIATION ON THE PERFORMANCE OF MELT GUARDRAIL TERMINAL [19].	- 41 -
FIGURE 2.8: LOGICAL CLASSIFICATION OF COMPARISON METRICS.	- 49 -
FIGURE 4.1: REPRESENTATION OF (A) SHIFT AND (B) DRIFT EFFECTS.	- 64 -
FIGURE 4.2: CURVES WITH DIFFERENT SAMPLING RATES BEFORE (LEFT) AND AFTER (RIGHT) THE RESAMPLING.	- 66 -
FIGURE 4.3: EXAMPLE OF A MATLAB STRUCTURE VARIABLE [63].	- 77 -
FIGURE 4.4: EXAMPLE OF A MATLAB STRUCTURE VARIABLE [63].	- 78 -
FIGURE 4.5: SYMBOLS USED FOR THE FLOWCHARTS IN THIS MANUAL.	- 79 -
FIGURE 4.6: SCHEMATIC REPRESENTATION OF THE LOOP ENCLOSING THE GUI'S FOR THE INPUT, PREPROCESSING AND SELECTION OF METRICS.	- 81 -
FIGURE 4.7: FLOW CHART OF THE MAIN ALGORITHM FOR THE PREPROCESSING BLOCK OF THE CODE.	- 82 -
FIGURE 4.8: ALGORITHM IMPLEMENTED FOR THE SAE FILTERING.	- 84 -
FIGURE 4.9: MAIN ALGORITHM OF THE SCRIPT <i>SHIFT_DRIFT</i> .	- 85 -
FIGURE 4.10: ALGORITHM OF THE USER-DEFINED FUNCTIONS <i>SHIFT_VALUE</i> (LEFT) AND <i>DRIFT_VALUE</i> (RIGHT).	- 86 -
FIGURE 4.11: SKETCH OF THE SUB-VECTORS DERIVED FROM THE ORIGINAL DATA VECTOR.	- 86 -
FIGURE 4.12: ALGORITHM OF THE SCRIPT <i>RESAMPLING_TRIMMING</i> .	- 88 -
FIGURE 4.13: ALGORITHM OF THE SCRIPT <i>CURVE_SYNCHRONIZING</i> .	- 90 -
FIGURE 4.14: ALGORITHM OF THE USER-DEFINED FUNCTIONS <i>AREA_RES</i> (LEFT) AND <i>SRE</i> (RIGHT).	- 91 -
FIGURE 4.15: ALGORITHM OF THE USER-DEFINED FUNCTION <i>SHIFT</i> .	- 92 -
FIGURE 4.16: SKETCH OF THE BEHAVIOR OF THE SHIFT SUBROUTINE FOR A (A) POSITIVE OR (B) NEGATIVE INPUT.	- 93 -
FIGURE 4.17: FLOWCHART OF THE ALGORITHM FOR CREATION OF THE RESULTS FOLDER.	- 96 -
FIGURE 4.18: SKETCH OF THE DATA STRUCTURE USED TO STORE THE METRICS TIME HISTORIES FOR A SINGLE INPUT CHANNEL (LEFT) AND MULTIPLE CHANNELS (RIGHT).	- 97 -
FIGURE 4.19: FLOWCHART OF THE ALGORITHM FOR THE OUTPUT OF THE HISTORIES OF THE METRICS IN EXCEL FORMAT.	- 98 -
FIGURE 4.20: SKETCH OF THE DATA STRUCTURE USED TO STORE THE METRICS FINAL VALUES FOR A SINGLE INPUT CHANNEL (LEFT) AND MULTIPLE CHANNELS (RIGHT).	- 99 -

FIGURE 4.21: SKETCH OF THE DATA STRUCTURE USED TO STORE THE METRICS FINAL VALUES COMPUTED ON DIFFERENT SUB-INTERVALS FOR A SINGLE INPUT CHANNEL (LEFT) AND MULTIPLE CHANNELS (RIGHT).....	- 100 -
FIGURE 4.22: FLOW CHART OF THE ALGORITHM FOR THE COMPUTATION OF THE METRICS OVER USER-DEFINED INTERVALS. ....	- 101 -
FIGURE 4.23: FLOWCHART OF THE ALGORITHM FOR THE OUTPUT OF THE FINAL VALUES OF THE METRICS IN EXCEL FORMAT. ....	- 103 -
FIGURE 4.24: PROCESS FOR THE EXTRACTION OF THE DATA FROM THE VARIABLE CONTAINING THE METRICS VALUES FOR THE VARIOUS INTERVALS CONSIDERED. ....	- 105 -
FIGURE 4.25: FLOWCHART OF THE ALGORITHM FOR THE INITIALIZATION OF THE VARIABLES AT THE START OF THE PROGRAM. ....	- 106 -
FIGURE 4.26: FLOWCHART OF THE ALGORITHM FOR THE UPDATE FOR THE CONFIGURATION FILE AT THE END OF THE PROGRAM. ....	- 106 -
FIGURE 4.27: DIAGRAM OF THE FIVE BLOCKS WHICH COMPOSE THE COMPUTER PROGRAM.....	- 107 -
FIGURE 5.1: TIME HISTORIES USED FOR THE VERIFICATION OF THE METRICS IMPLEMENTATION. ....	- 111 -
FIGURE 5.2: ORIGINAL CURVE (TOP) AND CURVES FILTERED USING BOTH THE IMPLEMENTED ALGORITHM AND THE COMMERCIAL PROGRAM LS/PREPOST (BOTTOM). ....	- 114 -
FIGURE 5.3. COMPARISON OF AN ACCELERATION CURVE AND THE EQUIVALENT RESAMPLED CURVE USING EITHER A LINEAR (LEFT) OR A CUBIC (RIGHT) INTERPOLATION. ....	- 115 -
FIGURE 5.4: ORIGINAL BIASED WAVES (TOP) AND SYNCHRONIZED WAVES (BOTTOM).....	- 117 -
FIGURE 5.5: EXPERIMENTAL CURVES BEFORE (TOP) AND AFTER (BOTTOM) BEING SYNCHRONIZED. ....	- 118 -
FIGURE 5.6: BASELINE AND SHIFTED/DRIFTED CURVES BEFORE (TOP) AND AFTER (BOTTOM) THE PREPROCESSING.....	- 119 -
FIGURE 5.7: PLOTS OF THE CURVES USED FOR THE VERIFICATION OF THE WEIGHTING FACTOR IMPLEMENTATION. ....	- 121 -
FIGURE 5.8: PAIRS OF CURVES USED IN THE CODE VERIFICATION OF THE IMPLEMENTED ALGORITHM FOR THE WEIGHTED AVERAGE. ....	- 123 -
FIGURE 6.1: VEHICLE MODEL USED IN THE FIRST SET OF CRASH TESTS (LEFT) AND A SIMILAR VEHICLE USED IN ONE OF THE TESTS OF THE SECOND SET (RIGHT) [66].....	- 128 -
FIGURE 6.2: TEMPORARY VERTICAL CONCRETE BARRIER USED IN THE REPEATED FULL-SCALE CRASH TESTS. ....	- 129 -
FIGURE 6.3: FLOW CHART OF THE PROCEDURE FOR THE ASSESSMENT OF THE FINAL SHIFT VALUES $s(i)$ FOR EACH TEST. ....	- 133 -
FIGURE 6.4: GRAPHICAL REPRESENTATION OF THE SHIFT ROUTINE. ....	- 134 -
FIGURE 6.5: GRAPHICAL REPRESENTATION OF THE SHIFT ROUTINE FOR THE CASE OF NEGATIVE SHIFT VALUES $s(i)$ OF ALL THE TESTS.....	- 134 -
FIGURE 6.6: FLOW CHART OF THE PROCEDURE FOR THE TRANSLATION OF THE CURVES OF THE VARIOUS TESTS. ....	- 135 -
FIGURE 6.7: ACCELERATION TIME HISTORIES AND 90 <sup>TH</sup> PERCENTILE ENVELOPE BASED ON THE AVERAGE CURVE FOR SET 1 (LEFT) AND SET 2 (RIGHT). ....	- 136 -
FIGURE 6.8: ACCELERATION TIME HISTORIES AND 90 <sup>TH</sup> PERCENTILE ENVELOPE BASED ON THE AVERAGE CURVE CONSIDERING ALL TESTS (COMBINATION OF SET 1 AND SET 2).....	- 137 -
FIGURE 6.9: ACCELERATION TIME HISTORIES AND 90 <sup>TH</sup> PERCENTILE ENVELOPE BASED ON THE TIME HISTORY OF THE RESIDUALS RESPECT TO THE AVERAGE CURVE FOR SET 1 (LEFT) AND SET 2 (RIGHT). ....	- 139 -
FIGURE 6.10: ACCELERATION TIME HISTORIES AND 90 <sup>TH</sup> PERCENTILE ENVELOPE BASED ON THE TIME HISTORY OF THE RESIDUALS RESPECT TO THE AVERAGE CURVE FOR THE COMBINATION OF ALL TESTS (SET 1 AND SET 2). ....	- 139 -

FIGURE 6.11: ACCELERATION TIME HISTORIES AND 90<sup>TH</sup> PERCENTILE ENVELOPE BASED ON THE TIME HISTORY OF THE RESIDUALS RESPECT TO THE INDICATED BASELINE CURVE FOR SET 1. .... - 141 -

FIGURE 6.12: ACCELERATION TIME HISTORIES AND 90<sup>TH</sup> PERCENTILE ENVELOPE BASED ON THE TIME HISTORY OF THE RESIDUALS RESPECT TO THE INDICATED BASELINE CURVE FOR SET 2. .... - 142 -

FIGURE 6.13: ACCELERATION TIME HISTORIES AND 90<sup>TH</sup> PERCENTILE ENVELOPE BASED ON THE TIME HISTORY OF THE RESIDUALS RESPECT TO THE INDICATED BASELINE CURVE FOR THE COMBINATION OF ALL TESTS (SET 1 AND SET 2) ..... - 143 -

FIGURE 6.14: PROBABILITY DENSITY FUNCTIONS OF THE RESIDUALS CONSIDERING THE INDICATED BASELINE CURVES FOR SET 1..... - 147 -

FIGURE 6.15: PROBABILITY DENSITY FUNCTIONS OF THE RESIDUALS CONSIDERING THE INDICATED BASELINE CURVES FOR SET 2..... - 148 -

FIGURE 6.16: PROBABILITY DENSITY FUNCTIONS OF THE RESIDUALS CONSIDERING THE INDICATED BASELINE CURVES FOR COMBINATION OF SET 1 AND SET 2. .... - 149 -

FIGURE 6.17: CUMULATIVE DISTRIBUTION FUNCTIONS OF THE RESIDUALS CONSIDERING THE INDICATED BASELINE CURVES FOR SET 1. .... - 150 -

FIGURE 6.18: CUMULATIVE DISTRIBUTION FUNCTIONS OF THE RESIDUALS CONSIDERING THE INDICATED BASELINE CURVES FOR SET 2. .... - 151 -

FIGURE 6.19: CUMULATIVE DISTRIBUTION FUNCTIONS OF THE RESIDUALS CONSIDERING THE INDICATED BASELINE CURVES FOR COMBINATION OF SET 1 AND SET 2. .... - 152 -

FIGURE 6.20: VELOCITY TIME HISTORIES AND 90<sup>TH</sup> PERCENTILE ENVELOPE FOR SET 1 BASED ON THE CORRESPONDING AVERAGE CURVE (LEFT) AND THE CURVE FROM TEST LAB#1 (RIGHT). .... - 170 -

FIGURE 6.21: VELOCITY TIME HISTORIES AND 90<sup>TH</sup> PERCENTILE ENVELOPE FOR SET 2 BASED ON THE CORRESPONDING AVERAGE CURVE (LEFT) AND THE CURVE FROM TEST LAB#1 (RIGHT). .... - 170 -

FIGURE 6.22: VELOCITY TIME HISTORIES AND 90<sup>TH</sup> PERCENTILE ENVELOPE FOR THE COMBINATION OF SET 1 AND SET 2 BASED ON THE CORRESPONDING AVERAGE CURVE (LEFT) AND THE CURVE FROM TEST LAB#1 OF SET 1(RIGHT). .... - 171 -

FIGURE 6.23: PROBABILITY DENSITY FUNCTIONS CONSIDERING AS BASELINE CURVE THE AVERAGE RESPONSE (TOP) AND THE FIRST TEST (BOTTOM) OF EACH SET/COMBINATION..... - 175 -

FIGURE 6.24: CUMULATIVE DISTRIBUTION FUNCTIONS CONSIDERING AS BASELINE CURVE THE AVERAGE RESPONSE (TOP) AND THE FIRST TEST (BOTTOM) OF EACH SET/COMBINATION..... - 176 -

FIGURE 7.1: PLOT OF THE VELOCITY WAVEFORMS FOR THE BEST (LEFT) AND THE WORST (RIGHT) PAIR. .... - 185 -

FIGURE 7.2: EXPERT EVALUATION FOR THE BEST AND WORST PAIRS OF VELOCITY WAVEFORMS PROPOSED BY SCHWER [43]..... - 186 -

FIGURE 7.3: COLUMN PLOTS OF THE METRICS VALUES FOR THE AGREEMENT (TOP) AND ERROR (BOTTOM) SCORES IN THE CASE OF THE WORST PAIR OF CURVES..... - 189 -

FIGURE 7.4: COLUMN PLOTS OF THE METRICS VALUES FOR THE AGREEMENT (TOP) AND ERROR (BOTTOM) SCORES IN THE CASE OF THE BEST PAIR OF CURVES. .... - 190 -

FIGURE 7.5: HISTOGRAM OF THE RESIDUALS FOR RESPECTIVELY THE BEST (TOP) AND THE WORST (BOTTOM) PAIRS OF CURVES CONSIDERING THE FIRST CURVE OF THE PAIR AS THE BASELINE. .... - 193 -

FIGURE 7.6: FE MODEL OF THE GEO-METRO VEHICLE USED TO SIMULATE THE FULL-SCALE CRASH TEST. .... - 194 -

FIGURE 7.7: BARRIER MODEL WITH A SUMMARY OF THE MODEL AND STATISTICS AND MATERIAL PROPERTIES. .... - 196 -

FIGURE 7.8: PLOT OF THE GLOBAL ENERGY TIME HISTORIES (NORMALIZED RESPECT TO THE INITIAL ENERGY)..... - 197 -

FIGURE 7.9: VEHICLE KINEMATICS FOR THE EXPERIMENTAL TESTS (LEFT AND RIGHT) AND THE FE SIMULATION (CENTER)..... - 199 -

FIGURE 7.10: ORIGINAL AND PREPROCESSED ACCELERATION CURVES CONSIDERING (CASE WITH TEST 1). ..... - 201 -

FIGURE 7.11: ORIGINAL AND PREPROCESSED YAW RATE CURVE CONSIDERING (CASE WITH TEST 1). ..... - 202 -

FIGURE 7.12: ORIGINAL AND PREPROCESSED ACCELERATION CURVES CONSIDERING (CASE WITH TEST 2). ..... - 202 -

FIGURE 7.13: ORIGINAL AND PREPROCESSED ROTATIONAL RATE CURVES CONSIDERING (CASE WITH TEST 2). ..... - 203 -

FIGURE 7.14: COMPARISON OF THE INTEGRAL FUNCTIONS FOR EACH OF THE ACCELERATION CHANNELS INVOLVING RESPECTIVELY TEST 1 (LEFT) AND TEST 2 (RIGHT)..... - 206 -

FIGURE 7.15: COMPARISON OF THE INTEGRAL FUNCTIONS FOR EACH OF THE ROTATIONAL RATE CHANNELS INVOLVING RESPECTIVELY TEST 1 (LEFT) AND TEST 2 (RIGHT). ..... - 207 -

FIGURE 7.16: WEIGHTING FACTORS COMPUTED CONSIDERING EACH OF THE FOUR IMPLEMENTED METHODS FOR THE COMPARISON INVOLVING RESPECTIVELY TEST 1 (TOP) AND TEST 2 (BOTTOM). ..... - 208 -

FIGURE 7.17: COMPARISON OF THE WEIGHTED VALUES USING EACH OF THE FIVE PROPOSED METHODS FOR RESPECTIVELY THE SPRAGUE&GEERS (TOP), AVERAGE (MIDDLE) AND STD (BOTTOM) METRICS (CASE WITH TEST 1). ..... - 215 -

FIGURE 7.18: COMPARISON OF THE WEIGHTED VALUES USING EACH OF THE FIVE PROPOSED METHODS FOR RESPECTIVELY THE SPRAGUE&GEERS (TOP), AVERAGE (MIDDLE) AND STD (BOTTOM) METRICS (CASE WITH TEST 2). ..... - 216 -

FIGURE 7.19: FINITE ELEMENT MODEL OF THE KTH COMPLEX [69]. ..... - 217 -

FIGURE 7.20: PLOTS OF THE TEST AND NUMERICAL CURVES AND THE FE MODEL SET-UP FOR THE KNEE CONDYLE (TOP), FEMUR (MIDDLE) AND PELVIS (BOTTOM) VALIDATION CASES. .... - 218 -

# List of Tables

TABLE 2.1. EXAMPLES OF DOMAIN-SPECIFIC METRICS IN ROADSIDE SAFETY .....	- 11 -
TABLE 2.2. REPORT 350 CRITERIA FOR TEST 4-10 ON THE MINNESOTA TYPE THREE COMBINATION BRIDGE RAILING [36] .....	- 15 -
TABLE 2.3. DEFINITION OF VARIOUS SINGLE-VALUE METRICS.....	- 28 -
TABLE 2.4. METRIC COMPONENTS FOR THE THREE SIMULATION CURVE OF FIGURE 2.4.....	- 36 -
TABLE 2.5. DEFINITION OF SINGLE-VALUE METRICS .....	- 50 -
TABLE 2.6. FORMULATION OF MULTIPLE-VALUE METRICS .....	- 52 -
TABLE 4.1. SPECIFICATIONS FOR THE MOST USUAL CFC VALUES OF SAE J211 COMPLIANT FILTERS [64] .....	- 63 -
TABLE 4.2. METRICS IMPLEMENTED IN THE COMPUTER PROGRAM .....	- 68 -
TABLE 4.3. MATHEMATICAL FORMULATION OF THE SINGLE-VALUE METRICS IMPLEMENTED IN THE COMPUTER PROGRAM <sup>§</sup> .....	- 68 -
TABLE 4.4. MATHEMATICAL FORMULATION OF THE MULTIPLE-VALUE METRICS IMPLEMENTED IN THE COMPUTER PROGRAM <sup>§</sup> .....	- 69 -
TABLE 5.1. METRIC VALUES COMPUTED FOR THE CURVES USED TO VERIFY THE METRIC IMPLEMENTATION .....	- 111 -
TABLE 5.2. CHARACTERISTICS OF THE CURVES USED FOR THE CODE VERIFICATION OF THE WEIGHTING FACTORS .....	- 120 -
TABLE 5.3. VALUES OF THE WEIGHTING FACTORS COMPUTED FOR EACH OF THE THREE PROPOSED METHODS.....	- 122 -
TABLE 5.4. CHANNELS AND WEIGHTED METRIC VALUES OBTAINED FROM THE VERIFICATION OF THE MULTICHANNEL CALCULATIONS.....	- 124 -
TABLE 6.1. SUMMARY OF THE BASELINE CURVES CONSIDERED FOR THE CREATION OF THE 90 <sup>TH</sup> PERCENTILE ENVELOPES .....	- 140 -
TABLE 6.2. SUMMARY OF THE AVERAGE AND STANDARD DEVIATION OF THE RESIDUAL ERRORS FOR TESTS OF SET 1 AND SET 2 .....	- 145 -
TABLE 6.3. SUMMARY OF THE AVERAGE AND STANDARD DEVIATION OF THE RESIDUAL ERRORS FOR TESTS OF THE COMBINATION OF SET 1 AND SET 2 .....	- 145 -
TABLE 6.4. VALUES OF THE COMPARISON METRICS FOR SET#1 (BASELINE CURVE: AVERAGE RESPONSE) .....	- 154 -
TABLE 6.5. VALUES OF THE COMPARISON METRICS FOR SET#2 (BASELINE CURVE: AVERAGE RESPONSE) .....	- 155 -
TABLE 6.6. VALUES OF THE COMPARISON METRICS CONSIDERING ALL TESTS (BASELINE CURVE: AVERAGE RESPONSE).....	- 156 -
TABLE 6.7. AVERAGE VALUES OF THE MEANS AND STANDARD DEVIATIONS, AND THE CORRESPONDING ACCEPTANCE CRITERIA.....	- 157 -
TABLE 6.8. SUMMARY OF THE BASELINE CURVES CONSIDERED FOR THE COMPARISON OF EACH SET OF TESTS .....	- 158 -
TABLE 6.9. VALUES OF THE COMPARISON METRICS FOR SET#1 (BASELINE CURVE: LAB 1).....	- 160 -
TABLE 6.10. VALUES OF THE COMPARISON METRICS FOR SET#2 (BASELINE CURVE: LAB 1).....	- 161 -
TABLE 6.11. VALUES OF THE COMPARISON METRICS CONSIDERING ALL TESTS (BASELINE CURVE: LAB 1 SET#1) .....	- 162 -
TABLE 6.12. AVERAGE VALUES OF THE MEANS AND STANDARD DEVIATIONS, AND THE CORRESPONDING ACCEPTANCE CRITERIA (BASELINE CURVE: LAB 1 OF EACH SET AND LAB 1 SET#1 FOR THE COMBINATION OF THE TWO SETS).....	- 163 -
TABLE 6.13. VALUES OF THE COMPARISON METRICS FOR SET#1 (BASELINE CURVE: LAB 5).....	- 165 -
TABLE 6.14. VALUES OF THE COMPARISON METRICS FOR SET#2 (BASELINE CURVE: LAB 5).....	- 166 -
TABLE 6.15. VALUES OF THE COMPARISON METRICS CONSIDERING ALL TESTS (BASELINE CURVE: LAB 5 SET#1).....	- 167 -
TABLE 6.16. AVERAGE VALUES OF THE MEANS AND STANDARD DEVIATIONS, AND THE CORRESPONDING ACCEPTANCE CRITERIA (BASELINE CURVE: LAB 5 OF EACH SET AND LAB 5 SET#1 FOR THE COMBINATION OF THE TWO SETS).....	- 168 -

TABLE 6.17. METRICS MEAN VALUES FOR EACH SET/COMBINATION COMPUTED CONSIDERING ACCELERATION AND VELOCITY TIME HISTORIES (BASELINE: AVERAGE RESPONSE) .....	- 172 -
TABLE 6.18. METRICS MEAN VALUES FOR EACH SET/COMBINATION COMPUTED CONSIDERING ACCELERATION AND VELOCITY TIME HISTORIES (BASELINE: LAB#1) .....	- 173 -
TABLE 6.19. MEAN VALUES OF THE METRICS OF EACH SET/COMBINATION COMPUTED FOR ANY OF THE THREE DIFFERENT CONSIDERED CASES (CASE I: AVERAGE RESPONSE; CASE II: LAB#1; CASE III: LAB#5).....	- 179 -
TABLE 6.20. BOUNDARY OF THE CONFIDENCE INTERVALS COMPUTED FOR EACH OF THE THREE CONSIDERED COMPARISON CASES .....	- 181 -
TABLE 7.1. COMPARISON METRICS AND CORRESPONDING AVERAGE VALUES FOR THE BEST AND WORST PAIRS .....	- 187 -
TABLE 7.2. AVERAGE AGREEMENT AND ERROR SCORES FOR RESPECTIVELY THE BEST AND WORST PAIRS OF CURVES .....	- 188 -
TABLE 7.3. VEHICLE TYPE AND IMPACT CONDITIONS FOR EACH OF THE TWO TESTS AND THE SIMULATION .....	- 194 -
TABLE 7.4. LIST OF PHENOMENA INVESTIGATED FOR THE VERIFICATION OF IMPROVEMENTS TO THE GEO-METRO MODEL .....	- 195 -
TABLE 7.5. INERTIAL PROPERTIES OF THE GEO-METRO FE MODEL USED IN THE SIMULATIONS .....	- 195 -
TABLE 7.6. ANALYSIS SOLUTION VERIFICATION TABLE FOR THE NUMERICAL MODEL.....	- 198 -
TABLE 7.7. ROADSIDE SAFETY VALIDATION METRICS RATING TABLE FOR EACH OF THE SINGLE CHANNELS .....	- 205 -
TABLE 7.8. METRICS RATING TABLE FOR THE GEO-METRO MODEL USING THE MULTI-CHANNEL OPTION (PEAK METHOD) .....	- 210 -
TABLE 7.9. METRICS RATING TABLE FOR THE GEO-METRO MODEL USING THE MULTI-CHANNEL OPTION (AREA I METHOD) .....	- 211 -
TABLE 7.10. METRICS RATING TABLE FOR THE GEO-METRO MODEL USING THE MULTI-CHANNEL OPTION (AREA II METHOD) .....	- 212 -
TABLE 7.11. METRICS RATING TABLE FOR THE GEO-METRO MODEL USING THE MULTI-CHANNEL OPTION (INERTIAL PROP. METHOD).....	- 213 -
TABLE 7.12. METRICS RATING TABLE FOR THE GEO-METRO MODEL USING THE MULTI-CHANNEL OPTION (RESULTANT METHOD).....	- 213 -
TABLE 7.13 S&G, MPC AND ANOVA METRIC VALUES.....	- 219 -

## List of Abbreviations

<b>ANOVA</b>	<i>Analysis of Variance</i>
<b>ASI</b>	<i>Acceleration severity Index</i>
<b>ASME</b>	<i>American Society of Mechanical Engineers</i>
<b>BCT</b>	<i>Breakaway Cable Terminal</i>
<b>CEM</b>	<i>Criterion Evaluation Method</i>
<b>CFC</b>	<i>Channel Frequency Class</i>
<b>CSA</b>	<i>Comparative Shock Analysis (Program)</i>
<b>FAA</b>	<i>Federal Aviation Administration</i>
<b>FE</b>	<i>Finite Element</i>
<b>FHWA</b>	<i>Federal Highway Administration</i>
<b>GEM</b>	<i>Global Evaluation Method</i>
<b>GUI</b>	<i>Graphical User Interface</i>
<b>HIC</b>	<i>Head Injury Criteria</i>
<b>LEM</b>	<i>Limit Evaluation Method</i>
<b>MCR</b>	<i>Matlab® Compiler Runtime</i>
<b>MELT</b>	<i>Modified Eccentric Loader Terminal</i>
<b>NARD</b>	<i>Numerical Analysis of Roadside Design</i>
<b>NCAC</b>	<i>National Crash Analysis Center</i>
<b>NCCF</b>	<i>Normalized Cross-Correlation Function</i>
<b>NCHRP</b>	<i>National Cooperative Highway research Project</i>
<b>OIV</b>	<i>Occupant Impact Velocity</i>
<b>ORA</b>	<i>Occupant Ridedown Acceleration</i>
<b>PHD</b>	<i>Post Impact Head Deceleration</i>
<b>RMS</b>	<i>Root Mean Squared (error)</i>
<b>ROBUST</b>	<i>Road Upgrade of Standard</i>
<b>SAE</b>	<i>Society of Automotive Engineers</i>
<b>TEM</b>	<i>Threshold Evaluation Method</i>
<b>THIV</b>	<i>Theoretical Head Impact Velocity</i>
<b>TOA</b>	<i>Time of Arrival</i>

**TRAP**

*Test Report Analysis Program*

**V&V**

*Verification and Validation*

# Chapter 1

## Introduction

### 1.1 Role of numerical simulations in roadside safety

In the past two decades both the exponential increase of available computational power in modern computers at an affordable cost and the continuous progress and improvement of numerical codes have promoted the use of numerical simulations for designing and assessing the performance of roadside safety hardware. The use of numerical simulations presents many advantages that are usually difficult, if not impossible, to achieve using the traditional practice of performing experimental tests. In fact, the designer is able not only to analyze the performance of a specific design but also to carry out a parametric study by varying one or more critical dimensions of the roadside hardware or changing any of the initial impact conditions. Another important benefit of numerical simulations in roadside safety is the possibility to assess how changes in the design of an existing hardware may affect the safety performance. The two most common numerical methods used in engineering are the Finite Element (FE) method and the multi-body approach. The former is the preferred approach in Roadside Safety as it allows reproducing in detail the deformation and potential failure of both the vehicle and the safety hardware components, thus allowing to study in a realistic manner the accelerations to the passengers while the structures are absorbing the energy due to the impact.

### 1.2 Verification and Validation

While very powerful design tools, numerical simulations can also lead to misleading or wrong results if the model which is used to simulate has not been correctly developed. In fact, in order to be reliable, numerical models must be **verified** and **validated** before they are used in any design or decisional process. Although similar, verification and validation are two different concepts which are often confused to each other.

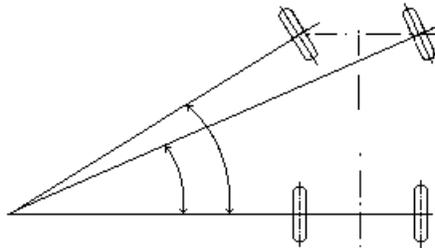
The verification of a numerical model is the process of comparing the solution of the numerical model with a known analytical solution, in case it exists. For example, considering the

propagation of a shock wave through solids, the verification of the numerical model should be the comparison of the solution obtained from the simulation and the solution obtained from the following partial differential equation [1,2]:

$$\frac{\partial^2 u}{\partial t^2} = \sqrt{\frac{E}{\rho}} \frac{\partial^2 u}{\partial x^2} \quad (1.1)$$

Note that in verification the issue is not comparing to the results of a real physical experiment, but comparing to the underlying mathematical expression of the problem. Hence, as far as the numerical model correctly approximates the solution of equation (1.1), the model is considered verified no matter if the equation may or may not be correct to its ability to predict the shock waves. In other words, verification is the assessment that a numerical model has been properly implemented in order to correctly reproduce or approximate the analytical formulation which describe the specific problem under investigation.

In roadside safety, an example of verification could be the correct definition of the geometry of the steering and suspension systems of the vehicle model. For example, the steering system in the vehicle model should comply with the well known Ackermann principle, which states that all the four wheels must roll around a common point during a turn [3]. If the geometry of the steering and suspension systems of the numerical model does not allow the inside tire to turn more than the outside tire during a turn, the model cannot be considered verified (Figure 1.1).



**Figure 1.1: Representation of the Ackerman concept [3].**

Differently from verification, the validation of a numerical model involves the comparison between the numerical simulation and a physical experiment. The American Association of Mechanical Engineers (ASME) has provided a synthetic and effective definition of the validation process: “Validation can be defined as the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the

model” [4]. This definition clearly states that in the validation process the assessment of the accuracy between the simulation and the real world has to be made relatively to the ‘*intended use*’ of that particular model. For example, a finite element model of a vehicle whose intended use is to assess and/or improve the crashworthiness of a vehicle itself may differ from a model intended to examine the interaction between the vehicle and a roadside safety system. In the former case, the deformations of parts of the vehicle would be highly relevant even if they do not affect the kinematics of the vehicle, whereas in the latter, the proper behavior of the suspension and steering system would probably play a greater role than such deformations on the evaluation of severity indexes like occupant risk or ride-down accelerations.

Clearly, the verification and validation process can be applied either at the whole model level or at any sub-component level; in this latter case the numerical model would be considered verified/validated regarding to the specific level taken into consideration.

Traditionally the Verification and Validation (V&V) process of numerical simulations involving a dynamic problem like a crash test has been accomplished by simply visually comparing two curves obtained respectively from the simulation and the corresponding experimental test. In particular, this visual assessment of how well two curves match to each other is mainly focused on finding common peaks, oscillations, shapes, etc. Although this kind of comparison gives an initial impression of how similar two curves are, it is based on a purely *subjective* judgment which could vary from one analyst to another. In contrast with the subjectivity of this comparison procedure, approval decisions need to be based as much as possible on quantitative and *objective* criteria that are unambiguous and mathematically precise. Hence, it is necessary to define objective comparison criteria based on computable measures. Comparison metrics, which are mathematical measures that quantify the level of agreement between simulation outcomes and experimental outcomes, can accomplish this goal.

In the literature it is possible to find various comparison metrics which have been created to the scope of comparing two similar curves. Usually, each of these metrics has been applied in a specific and limited engineering field and is known by a limited number of practitioners in that particular field. Even though a practitioner is aware of a particular comparison metric, often, the tediousness and the implementation problems related to the evaluation of the metric itself can discourage its use.

### 1.3 Objective of the research

The objective of this research is to implement a method to quantitatively assess the level of verification and/or validation of a numerical model. In particular, the main part of this research is focused on the development of a computer program that can be used to assess the degree of match between a single or multiple pairs of curves; the software has to be able to automatically assess how well two curves compare to each other by giving an objective score. Practically, the comparison metrics calculated by this software can be used to validate computer simulation models using data obtained from experimental tests, verify a simulation with another simulation, assess the repeatability of an experimental test or, generally speaking, perform a comparison of virtually any pair of curves. In this perspective, the program developed in this study may be a useful tool for researchers and policy makers.

Although the implemented software can be practically used to compare curves from virtually any field of study, a particular attention is put in its application for roadside safety. Moreover, acceptance criteria for the quantitative metrics computed by the developed software are proposed based on the typical scatter of experimental results in roadside safety.

Eventually, the robustness and reliability of the results obtained using the implemented software are tested by comparing the qualitative score of the computed metrics for a set of velocity waveforms with the corresponding subjective judgment of experts. Also, the implemented method is applied to two real validation cases involving respectively a numerical model in roadside safety and a second model in biomechanics. In the former case study, the acceptance criteria are used to compare experimental and numerical results in a typical roadside safety scenario.

# **Chapter 2**

## **Literature review**

In order to acquire an overall view of what has been accomplished in the validation and verification of numerical models, a comprehensive literature review is presented in this chapter. The literature review herein presented can be easily divided into three main sections. The first part covers a review of the use of numerical simulations in roadside safety and focuses on the most common practices used to verify and/or validate numerical models. As the use of explicit non-linear Finite Element (FE) solvers has practically become a standard in the simulation of crash events, the attention is mainly focused on numerical models based on this specific modeling technique. After this initial summary on the use of simulations in roadside safety, a description of the comparison metrics which have been developed in both engineering and non-engineering fields is presented with the main purpose of reviewing the principal characteristics of each metric. Also, a review of the computer codes found in the literature which can be used to automatically evaluate comparison metrics is described. Concluding, a synthesis of the literature review is presented.

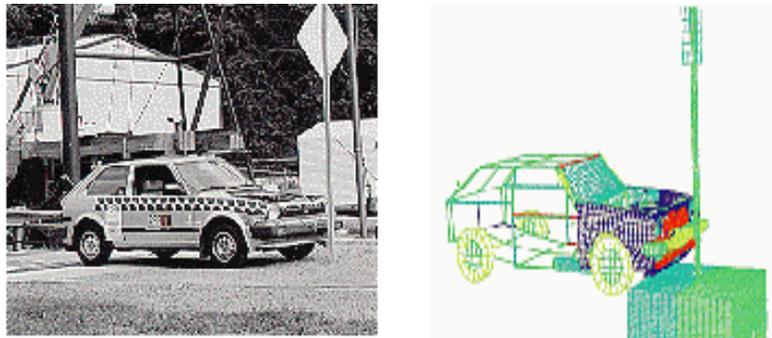
### **2.1 Use of numerical analysis in roadside safety**

The use of numerical simulations in roadside safety started in an early date with the implementation of the first simulation codes like Barrier VII and Numerical Analysis of Roadside Design (NARD) [6,7]. These early programs were based on the finite element method and presented significant limitations which made them suitable only to perform some simple parametric analyses that would be later more extensively explored using full-scale crash tests. During the 80's DYNA3D [8], a finite element solver especially tailored to simulate crash events, was developed at Livermore National Laboratories and was soon after followed by its commercial counterpart, LS-DYNA [9]. Following in the early 90's LS-DYNA started to be applied in the aerospace and automotive industry for the evaluation of crash scenarios. Another field in which numerical simulations were thoroughly applied to the design of new hardware was

roadside safety. In the next part of this section a summary of the principal works performed using numerical simulations in roadside safety is given.

One of the first works in roadside safety performed using finite element simulations was made in 1993 by Wekezer [10]. Wekezer simulated the impact of a compact car against a light pole. In that case the author proved how, even with a relatively limited number of degrees of freedom, it was possible to predict the kinematics of vehicles in impacts.

In 1994, Ray examined the impact of a 820-kg small car striking a flange-channel sign post at nine meters per second [11]. The vehicle model used to examine the collision sequence is shown in Figure 2.1. Using that model, it was possible to analyze in detail the state of stress of any vehicle or barrier component in order to determine the actual failure mechanisms involved in the collision. Qualitative comparisons of the deformed vehicle shape and acceleration-time histories were used to assess the accuracy of the model.



**Figure 2.1: Photograph (left) and corresponding FE model (right) of the pre-collision with a sign post [11].**

In 1997 the Federal Highway Administration (FHWA) developed the model of a weak post W-beam G2 guardrail [12]. Modeling details, such as post-soil interaction, W-beam end anchorage, post-to-W-beam connection were explained. The results of a full-scale crash test were qualitatively compared to the simulation results. Later, Martin and Wekezer continued to work on the development of a finite element model of the G2 weak post W-beam guardrail [13] using the National Crash Agency Center (NCAC) model of a 1994 Chevrolet pickup truck to simulate an impact with the barrier at 100 km/hr and 25 degrees. In this study, acceleration time histories obtained from the crash test and simulation study were used to make both qualitative and quantitative comparisons. Two comparison metrics which will be discussed later in this chapter,

the NARD validation metrics and the ANOVA metrics proposed by Ray were used to compare the simulated and experimental responses.

Ray and Plaxico continued the work on weak-post W-beam guardrails in order to solve guardrail over-riding and penetration problems occurred in tests involving the 2000-kg pickup truck [14]. Design modifications to the weak-post W-beam guardrail were explored using both finite element simulations and full-scale crash tests. Also in this case, qualitative and quantitative comparisons were made between the crash test results and simulation results to validate the accuracy of the finite element model.

Eskandarian *et al.* developed a model of a slip-base sign support system to investigate the crash performance with vehicles using DYNA3D [15]. A model of a bogie with honeycomb material was used as the impacting vehicle and the FE crash simulations of the bogie impacting the slip-base sign support were validated using the corresponding instrumented crash tests. In a similar study, finite element simulation and its application to crashworthiness evaluation and safety analysis of roadside hardware appurtenances were presented [16]. Also in this case, only qualitative comparisons, such as acceleration, velocity and displacement vs. time graphs were compared.

Reid *et al.* analyzed a turned-down guardrail terminal using LS-DYNA and full-scale crash tests [17]. Finite element simulations were performed on the existing turned-down approach terminal section as well as on various retrofit options to understand the crash performance of end terminals and to evaluate the performance of design alternatives. Deformations and crash test pictures were used to validate the accuracy of the finite element models.

In another study, Paulsen and Reid modeled a dual support breakaway sign system using LS-DYNA [18]. Component models were first constructed on critical parts of the breakaway sign system and the models were compared with physical component tests to aid in the development process, as well as to validate the component simulation results. The components were then assembled into a complete system model. Qualitative comparisons were made between the simulation results and two full-scale vehicle crash tests were used to validate the model.

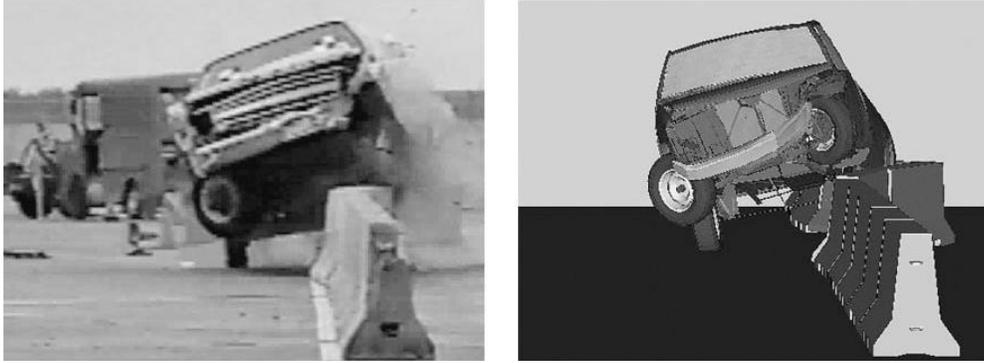
Ray and Patzner developed an LS-DYNA model of a MELT guardrail terminal to investigate the performance of this type of guardrail terminal [19]. Results of the analysis were compared to data from a full-scale crash test involving a small passenger car. Qualitative comparisons of the

acceleration and velocity time histories were made and also quantitative comparisons of the occupant risk criteria and a statistical method were used to illustrate the validity of the models. In particular, the quantitative comparison was performed using the Test Report Analysis Program (TRAP) [20], which automatically evaluates the occupant risk criteria defined in NCHRP Report 350 [21] using the acceleration curves obtained from the simulation data.

In a similar study, Plaxico provided a description of the development of a model of a breakaway timber post and soil system used in the breakaway cable terminal (BCT) and the modified eccentric loader terminal (MELT) [22]. Also in this case, the simulation results were qualitatively compared with data from physical tests of BCT/MELT posts.

Reid *et al.* simulated with LS-DYNA the impact performance of the new barrier called Buffalo Rail capable of capturing and redirecting a larger range of vehicle types and sizes thanks to an innovative design of the W-beam cross-sectional shape, rail thickness and post spacing [23]. Qualitative and quantitative comparisons were made to validate the finite element model of the Buffalo Rail. Similarly, the same authors also used LS-DYNA simulations to analyze a model of the sequential kinking process for the energy dissipation in a new guardrail terminal concept [24]. Qualitative comparisons between the full-scale tests and finite element results were used to validate the model. In particular, the full-scale crash tests showed that predictions of the energy dissipation for the sequential kinking were only seven percent below values obtained from dynamic impact tests.

Marzougui *et al.* used LS-DYNA simulations to evaluate the safety of portable concrete barriers (PCB) [25]. To create accurate models of PCBs, a model of an F-shape PCB design was first created and full-scale crash test data were used to validate the model. Qualitative comparisons were made to validate the model (see Figure 2.2). Once the model had been validated, models of two modified PCB designs were created and their safety performance was evaluated. Based on the simulation results, a third design was developed and its performance was analyzed as well. The safety performances of the three designs were compared.



**Figure 2.2: Qualitative comparison of pickup truck impact with an F-shape portable concrete barrier [21].**

Bielenberg discussed the development of barriers for race venues. He developed a barrier with Polystyrene foam blocks placed between an outer steel tube structure and the race track concrete wall [26]. After successfully modeling a bogie tests, the component model of the foam was placed in the full-scale model of the barrier. Sequential pictures and time history data were used to validate the finite element models.

Ray *et al.* described the design and analysis of an extruded aluminum truss-work bridge railing [27]. In this case there were no crash tests available to assess the performance of the model and LS-DYNA was used to determine if the bridge railing would be likely to result in successful crash tests. The FHWA issued an acceptance letter for this new type of bridge railing based only on the computational analysis.

Mohan *et al.* developed a detailed finite element model of a three-strand cable barrier which was capable of simulating the dynamic interactions of the soil and post, post and hook bolts, cable and hook bolts and cable to truck [28]. The accuracy of the model was validated against a previously conducted full-scale crash test.

Finite element computer simulations coupled with experimental testing were used to investigate the safety of mailbox supports and establish some guidelines on their use and installation [29]. Initially, the mailbox model was validated against some pendulum crash tests, and then a parametric finite element analysis was performed with various mailbox sizes, heights, mounting configurations and post sizes in order to evaluate the mailbox support crashworthiness performance. Eventually, a full-scale crash test was performed using the most critical impact scenario and compared with the corresponding simulation.

Plaxico investigated the performance of a 50-inch high portable concrete barrier which was tall enough to serve as its own glare-shield [30]. Finite element analysis was used to investigate various barrier shapes and connection schemes to identify a successful crashworthy design that would meet the requirements of Report 350 for test level three. The results of a full-scale crash test were used to validate the simulation outcomes using qualitative comparisons of the time histories and sequential pictures, as well as quantitative comparisons of the TRAP domain-specific metrics.

Anghileri investigated the influence of the output frequency and the location of the accelerometer sensor on the computation of the acceleration-time histories and occupant risk factors in numerical simulations [31]. His work was used to explain some of the variation in crash test results that were observed in a round-robin series of crash test in which the same test was performed by five different crash test laboratories.

Table 2.1 summarizes the methods used to validate the models in the previously reviewed publications. In most of the cases found in the literature, the validation of the numerical models used to assess the roadside hardware was performed by means of qualitative methods.

Table 2.1. Examples of domain-specific metrics in roadside safety

Publication	VALIDATION METHOD					
	QUALITATIVE		QUANTITATIVE			
	Time History	Deformation	Geer's Parameters	NARD	TRAP	Hypothesis Testing
Ray (1994)	√	√				
Hendricks and Wekezer (1996)	√	√				
Reid et al (1996)		√				
Paulsen and Reid (1996)	√	√				
Ray and Patzner, (1997)	√	√			√	√
Reid et al (1997)	√	√			√	
Plaxico et al (1998)	√	√				
Martin and Wekezer (1998)		√		√		√
Reid and Sicking, (1998)	√	√				
Marzougui et al (2000a)		√				
Marzougui et al (2000b)		√				
Eskandarian et al (2000)	√	√		√		
Ray et al (2001)		√			√	
Ray et al (2004)		√			√	
Bielenberg, (2004)	√	√				
Mohan et al (2005)	√	√				
Tahan et al (2005)	√	√				
Plaxico et al (2006b)	√	√			√	
Anghileri (2006)	√	√			√	

### 2.1.1 The issue of verification and validation

Obtaining accurate and reliable numerical simulations requires careful attention to modeling in detail the physics, geometry, material properties and several other characteristics of the particular phenomenon which is studied. In order to provide reliable results, the numerical model has to be verified and validated.

As previously mentioned in the Introduction, verification and validation are different concepts. Although both verification and validation are related to a comparison process which involves the numerical solution, they differ from each other for the type of results against which the comparison is carried out. In fact, the verification consists in comparing the numerical results against a known theoretical solution of the specific problem under investigation, while the validation refers to the comparison of the numerical solution with a physical experiment which reproduces the particular phenomenon under investigation. Hence, verification ensures that the model has been correctly developed according to the physics of the problem (e.g., boundary

conditions and constraints have been correctly modeled, the kinematics of the vehicle and other mechanical components is correctly reproduced, etc.) while validation ensures that the numerical model gives results consistent with the real behavior of the system which is modeled. In other words, verification can be seen as a necessary condition, or “*conditio sine qua non*”, for which a numerical model might generate correct results: if a model has been verified it means that potentially it is capable of simulating correct results. Unfortunately, verification is not also a condition sufficient to guarantee that results obtained from the numerical model are correct. In fact, although a model is verified, some of the assumptions or simplifications which inevitably characterize the development of a numerical model could still be wrong or unacceptable for the necessary level of accuracy. Hence, once a model has been verified, it is also necessary to validate the results against one or more experimental tests. Without a careful verification and validation of the numerical model, the results of the simulation cannot be depended on.

## 2.2 Comparison metrics

According to the American Society of Mechanical Engineers (ASME) a comparison metric can be defined as a “measure that quantifies the level of agreement between simulation outcomes and experimental outcomes.” [4].

In the literature it is possible to find various comparison metrics. Most of the metrics available in the literature have been developed in engineering fields, but some of them have also been developed in other research fields, like economics and statistics. Essentially, comparison metrics can be grouped into two main categories: (i) **deterministic** and (ii) **stochastic** metrics (Figure 2.3). Deterministic metrics do not specifically address the probabilistic variation of either experiments or calculation as they imply that, given the same input, the calculation results are the same result every time. On the other side, stochastic metrics involve computing the likely variation in both the simulation and the experiment response due to parameter variations. The formulation and a comprehensive description of the characteristics of the metrics within each group are presented in this subchapter.

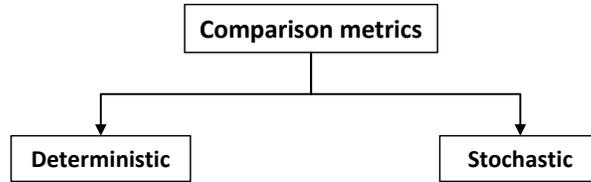


Figure 2.3: Main classification of comparison metrics.

### 2.2.1 Deterministic metrics

Deterministic metrics can basically be classified into two main subgroups: (a) **domain-specific** and (b) **shape-comparison** metrics.

#### 2.2.1.1 Domain-specific metrics

Domain-specific metrics are quantities specific to a particular application. The intended use of the model helps to identify the domain-specific metrics and, generally, domain-specific metrics are chosen among the parameters/indexes required by the testing standard in a particular field.

For example, in the case of roadside safety, tests are performed and evaluated according to Report 350 [21] in the US or EN 1317 [33] in Europe. These crash testing guidelines define various severity indexes like the Occupant Impact Velocity (OIV), Occupant Ridedown Acceleration (ORA), Theoretical Head Impact Velocity (THIV), Acceleration Severity Index (ASI), Post Impact Head Deceleration (PHD) and a range of exit conditions. Each of these severity indexes, which are calculated based on time history data collected in a crash test or from a numerical simulation, might be considered also useful domain-specific metrics in roadside safety. Similarly, in railroad safety the axial crush of a car and the energy absorption during the impact [34] or, in aerospace safety, the computation of the Head Injury Criteria (HIC) [35] for the testing of aircraft seats can represent domain-specific metrics.

Evidently, either the specific severity indexes from the tests in roadside and aerospace safety or the total energy absorption of a railcar may not be relevant if applied in other design situations. As each domain-specific metric is specifically “tailored” to be used in a specific field and for a particular application it may be difficult, if not impossible, to apply the same metric to fields different than those for which it was original intended. Once a domain-specific metric has been calculated for either a test and a numerical simulation or two experimental tests, it is possible to

assess the level of comparison between them by simply comparing the values of the domain-specific metrics.

Since the purpose of performing computer simulations in roadside safety is generally to design hardware such that the Report 350 or EN 1317 test response can be predicted, it makes sense to include these test-based evaluation parameters in any discussion of validation. Table 2.2 for example, shows a comparison between the NCHRP Report 350 evaluation table and the results of an LS-DYNA simulation of the Minnesota Type 3 Combination Bridge Railing [36]. The LS-DYNA model was constructed to exactly replicate the geometry, material properties and impact conditions in the actual experiment so the comparison is a validation exercise. The Report 350 evaluation table contains 14 specific evaluation metrics, eight of which apply to test 3-10. Of the eight evaluation criteria, five are pass/fail qualitative assessments (i.e., criteria A, D F, K and L) based on the global performance of the system. The three remaining criteria (i.e., H, I and M) are calculated quantities based on the time histories.

As shown in Table 2.2, the qualitative criteria like containment and redirection (i.e., criterion A) and the detached fragment criterion (i.e., criterion D) can be used to compare results of crash tests and simulations. It is best if these criteria are unambiguous but the simulation results can be judged in the same way as the experimental results. Also shown in Table 2.2 are the deterministic domain-specific metrics OIV and ORA. In this case, both OIV values (i.e., lateral and longitudinal) are less than 15 percent; so, if 15 percent were the allowable acceptance criteria, both would be acceptable. If the acceptance criteria were 10 percent, however, the longitudinal value would be judged not acceptable. This has certain diagnostic value since a higher experimental value indicates the vehicle may be snagging more, or the coefficient of friction may be higher in the experiment than in the simulation. The analyst could then return to the simulation and attempt to discover the reason for the discrepancy. In this particular case, the simulation ORA values are both much higher than the experiments and fall outside of any likely acceptance criteria. Accordingly, the comparison indicates that the model is not validated. The point of Table 2.2 is to show that the same experimental evaluation metrics can be used as deterministic domain-specific metrics when comparing full-scale crash test experiments to simulations. The EN 1317 values like THIV and ASI could likewise be used as deterministic domain-specific validation metrics.

**Table 2.2. Report 350 criteria for test 4-10 on the Minnesota Type Three Combination Bridge Railing [36]**

<b>Evaluation Factors</b>	<b>Evaluation Criteria</b>	<b>Crash Test</b>	<b>FE Simulation</b>	<b>Difference %</b>		
Structural Adequacy	A. Test article should contain and redirect the vehicle; the vehicle should not penetrate, underride, or override the installation although controlled lateral deflection of the test article is acceptable.	Passed	Passed	-		
	B. The test article should readily activate in a predictable manner by breaking away, fracturing or yielding.	NA	NA	-		
	C. Acceptable test article performance may be by redirection, controlled penetration or controlled stopping of the vehicle.	NA	NA	-		
Occupant Risks	D. Detached elements, fragments or other debris from the test article should not penetrate or show potential for penetrating the occupant compartment, or present an undue hazard to other traffic, pedestrians or personnel in a work zone.	Passed	Passed	-		
	E. Detached elements, fragments or other debris from the test article, or vehicular damage should not block the driver's vision or otherwise cause the driver to lose control of the vehicle.	NA	NA	-		
	F. The vehicle should remain upright during and after the collision although moderate roll, pitching and yawing are acceptable.	Passed	Passed	-		
	G. It is preferable, although not essential, that the vehicle remain upright during and after collision.	NA	NA	-		
	H. Occupant impact velocities should satisfy the following:					
	Occupant Impact Velocity Limits (ft/s)					
	Component	Preferred	Maximum			
	Longitudinal and Lateral	30	40	16.4 ft/s 27.82 ft/s	14.1 ft/s 25.9 ft/s	14 7
	Longitudinal	10	15	NA	NA	-
	I. Occupant ridedown accelerations should satisfy the following:					
Occupant Ridedown Acceleration Limits (g's)						
Component	Preferred	Maximum				
Longitudinal and Lateral	15	20	2.6 g's 10.6 g's	5.8 g's 15.2 g's	-123 -43	
J. (Optional) Hybrid III dummy responses.		NA	NA	-		
Vehicle Trajectory	K. After collision it is preferable that the vehicle's trajectory not intrude into adjacent traffic lanes.	Passed	Passed	-		
	L. The occupant impact velocity in the longitudinal direction should not exceed 40 ft/sec and the occupant ridedown acceleration in the longitudinal direction should not exceed 20 G's.	Passed	Passed	-		
	M. The exit angle from the test article preferable should be less than 60 percent of test impact angle, measured at the time of vehicle loss of contact with test device.	7.5° (Passed)	4.7° (Passed)	37		
	N. Vehicle trajectory behind the test article is acceptable.	NA	NA	-		

### **2.2.1.2 Shape-comparison metrics**

Shape-comparison metrics measure the “difference” in shape between a pair of curves, independently of their specific nature. Unlike domain-specific metrics, shape-comparison metrics directly involve a comparison of the two curves with no further need to compare the metric value from each curve. In fact, a shape-comparison metric is evaluated using simultaneously both the two curves, and its value is already a quantitative measure of the agreement between the curves. Also, these metrics assess the degree of similarity between any two curves in general and, therefore, do not depend on the particular application domain unlike domain-specific metrics. Because of their formulation, shape-comparison metrics can be applied independently to the particular nature of the curves, which may be time histories, force-deflection plots, stress-strain plots, etc.

Although a variety of shape-comparison metrics based on various and different formulations are available in the literature, it is possible to identify two main groups: (a) single-value metrics and (b) multiple-value metrics. The following sections describe in detail the metrics of each of these two subcategories of the shape-comparison group. When possible, the original derivation of the metric is also presented. If available in the literature, a description of practical applications of the metric is described. In all the following sections, the terms  $m_i$  and  $c_i$  refer to the measured and computed quantities respectively, and the “i” subscribe indicates the measurement at a specific instant in time.

#### **2.2.1.2.1 Single-value metrics**

Single-value metrics are characterized by a single element, which gives an overall assessment of the comparison. This group is composed by various metrics based of different concepts.

##### **2.2.1.2.1.1 NARD metrics**

The NARD Validation Manual was published by the FHWA in 1998 to provide several techniques for comparing full-scale tests and simulations [37]. The comparison metrics proposed in the NARD manual can be classified into two groups according to the type of domain in which the original curves are analyzed:

- Frequency domain and
- Time domain

Three validation metrics are defined in the frequency domain and further three metrics are defined in the time domain.

### *Frequency analysis*

The validation metrics in the frequency domain are based on the transformed signals  $F(\omega)$  and  $G(\omega)$ . The following three metrics in the frequency domain are defined in the NARD manual:

- The relative absolute difference of amplitude of two signals,
- The point-wise absolute difference of amplitudes of two signals and
- The root-mean-squared (RMS) log spectral difference between two signals.

The time domain signal is transformed into its corresponding frequency domain signal using a Fourier transformation. Any time domain signal can be expressed in the form:

$$f(t) = \sum_{-\infty}^{+\infty} A^{(i)} \cos(i\omega t) + B^{(i)} \sin(i\omega t) \quad (2.1)$$

If  $A_m^{(i)}$  and  $B_m^{(i)}$  are coefficients of  $m(t)$  and  $A_c^{(i)}$  and  $B_c^{(i)}$  are coefficients of  $c(t)$ , the ***point-wise absolute differences*** are defined as:

$$\Delta_A^{(i)} = |A_m^{(i)} - A_c^{(i)}| \quad ; \quad \Delta_B^{(i)} = |B_m^{(i)} - B_c^{(i)}| \quad (2.2)$$

The ***relative absolute differences*** are defined as:

$$\Delta AB_{rel}^{(i)} = \left| \sqrt{(A_m^{(i)})^2 + (B_m^{(i)})^2} - \sqrt{(A_c^{(i)})^2 + (B_c^{(i)})^2} \right| \quad (2.3)$$

For both the point-wise absolute differences and the relative absolute difference, the NARD Validation Manual considers the measured and computed curve to be close to one another if the difference is less than 20 percent.

The **RMS log spectral distance** measures the distance between the smooth power spectra of the measured and computed signals. In order to define the smooth power spectrum, it is first necessary to define the auto covariance functions associated with the measured and computed signals:

$$\begin{cases} C_m(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{-T/2+\tau} m(t)m(t+\tau)dt \\ C_c(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{-T/2+\tau} c(t)c(t+\tau)dt \end{cases} \quad (2.4)$$

The smooth power spectrum of the measured and calculated functions is defined as:

$$\begin{cases} \Gamma_m(\omega) = \int_{-\infty}^{+\infty} w(\omega)C_m(\tau)e^{-i\omega\tau}d\tau \\ \Gamma_c(\omega) = \int_{-\infty}^{+\infty} w(\omega)C_c(\tau)e^{-i\omega\tau}d\tau \end{cases} \quad (2.5)$$

Where  $w(\omega)$  is the spectral window.

Eventually, the RMS log spectral distance in units of decibels (db) is given by:

$$D = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left| \ln \left( \frac{\Gamma_m(\omega)}{\Gamma_c(\omega)} \right) \right|^2 d\omega \quad (2.6)$$

The smaller the RMS log spectral distance between the signals is, the closer the signals are. According to the NARD Validation manual, a distance of 20 db or less indicates that the difference between the signals is not more than 20 percent. Both the RMS log spectral distance and the relative absolute RMS are deterministic metrics with an *ad hoc* acceptance criterion of 20 percent.

In the Roadside Safety literature no application of the NARD frequency analysis metrics was found. In fact, no example of the use of frequency domain metrics was found in any of the solid mechanics literature. This may be due to the difficulties which arise when trying to apply these particular metrics to the very short time histories of a crash event.

### *Time Domain*

The NARD Validation Manual defines also comparison metrics based on time-domain. In this case, a point-to-point measure is used to evaluate the metrics value.

The following three metrics in the time domain are defined in the NARD manual:

- Relative Absolute difference of Moments of the two signals,
- Root Mean Square (RMS) log measure of the difference between two signals and,
- Correlation Coefficient.

One of the simplest ways to compare two signals is comparing the moments of the shapes as proposed by the NARD Validation manual. The **relative absolute difference of the moments** is based on the comparison of the moments of the area under the time history curve. Moments are mathematical characteristics of a shape (e.g., moments of inertia) and can be defined by the following general expression:

$$M_n = \int_0^T t_i^n m_i dt \quad (2.7)$$

In discrete terms, Equation (2.7) becomes:

$$\begin{aligned} M_{j,m} &= \sum_{i=1}^n t_i^j m_i \cdot \Delta t \\ M_{j,c} &= \sum_{i=1}^n t_i^j c_i \cdot \Delta t \end{aligned} \quad (2.8)$$

The lower order moments have some physical meaning. For example, the zero order moment ( $j=0$ ) when divided by the number of samples is the average acceleration. The 1<sup>st</sup> order moment ( $j=1$ ) divided by the 0<sup>th</sup> order moment locates the time at the centroid of the time history. Moments of order greater than one have little physical meaning when comparing time histories and are simply mathematical characteristics of the shapes. The more moments (i.e., shape characteristics) that two shapes have in common the more likely, in a general sense, they are to represent the same shape. If enough characteristics of the measured acceleration history shape match the characteristics of the calculated acceleration history, the shapes should be similar. The ratio  $R$  between the difference of the  $n^{\text{th}}$  moment of a measured ( $m_i$ ) and a calculated ( $c_i$ ) signal and the  $n^{\text{th}}$  moment of the calculated signal is given by:

$$\frac{|M_n(m_i) - M_n(c_i)|}{M_n(c_i)} \quad (2.9)$$

In discrete formulation, Equation (2.9) becomes:

$$R = \frac{|\sum_{i=1}^n t_i^n m_i - \sum_{i=1}^n t_i^n c_i|}{\sum_{i=1}^n t_i^n c_i} \quad (2.10)$$

The NARD validation procedure recommends that the 0<sup>th</sup> through 5<sup>th</sup> relative differences of the moments defined by equation (2.7) be calculated. The NARD Validation Manual arbitrarily considers the measured and calculated moments to be similar if the absolute difference between

the respective moments of order  $n$   $M_n(m_i)$  and  $M_n(c_i)$  is less than 0.2. All the relative moment metrics are, therefore, deterministic shape metrics with an *ad hoc* acceptance criterion of 20 percent.

Interestingly, the NARD validation procedure makes the comparison with respect to the calculated value rather than the experimental value. It is more appropriate to make the comparison with the experimental value since, from a validation point of view, the experimental value is the “true” response.

The mean value of a signal is simply the algebraic sum of the values divided by the number of values. Similarly, the mean squared is the algebraic sum of the square of the values divided by the number of values. If the square root of the mean squared is taken, the **root mean square (RMS)** of the measured and computed signals are obtained as shown below:

$$\begin{aligned} RMS_m &= \sqrt{\frac{1}{n} \sum_{i=1}^N m_i^2} \\ RMS_c &= \sqrt{\frac{1}{n} \sum_{i=1}^N c_i^2} \end{aligned} \quad (2.11)$$

The RMS is the average value of the signal without respect to its sign. The RMS of two signals can be compared by taking the difference of the two RMS and dividing by the average of the two RMS as follows:

$$\Delta RMS^r = \sqrt{\frac{\sum_{i=1}^N m_i^2 - \sum_{i=1}^N c_i^2}{(\sum_{i=1}^N m_i^2 + \sum_{i=1}^N c_i^2)/2}} \quad (2.12)$$

As with the relative moments, the choice of denominator is ambiguous. For moments, the difference is calculated with respect to the calculated signal whereas for the relative RMS, the difference is calculated with respect to the average. Again, since these are validation metrics, the “true” experimental solution should be the reference value in the denominator.

Like the relative moments, the RMS is simply a characteristic of a particular shape. In the case of an acceleration time history, it is the average value of the accelerations without respect to the sign.

The logarithmic form of the RMS difference can also be considered, as suggested by the NARD Validation Manual:

$$\Delta RMS_{log}^r = \sqrt{\frac{2 \sum_{i=1}^N 10 \log\left(\frac{m_i^2}{c_i^2}\right)^2}{\sum_{i=1}^N (10 \log(m_i^2)^2 + 10 \log(c_i^2)^2)}} \quad (2.13)$$

Both the relative RMS and the logarithmic relative RMS are deterministic shape metrics with an *ad hoc* acceptance criterion of 20 percent.

The **correlation coefficient**, proposed in the NARD Validation Manual, measures the correlation between two signals. Correlation in this context does not mean that the signals are identical but only that one can be linearly transformed into the other. The correlation coefficient is, therefore, a measure of the relative phasing of the two signals. The correlation coefficient of two signals is given in the NARD Validation Manual as:

$$\rho = \frac{\int_0^T c_i m_i \cdot dt}{\sqrt{\int_0^T c_i^2 \cdot dt} \cdot \sqrt{\int_0^T m_i^2 \cdot dt}} \quad (2.14)$$

In a discrete formulation, Equation (2.14)

$$\rho = \frac{\sum_{i=1}^N c_i m_i}{\sqrt{\sum_{i=1}^N c_i^2} \sqrt{\sum_{i=1}^N m_i^2}} \quad (2.15)$$

The closer the correlation ratio is to unity, the more the calculated and measured signals can be linearly transformed into each other.

Several applications of the NARD validation metrics in the time domain were found in the literature for roadside safety finite element simulations. For example Ray, in an unpublished paper quantitatively evaluated four finite element models using the NARD metrics in the time domain among others [38]. In 2000, Tabiei and Wu used the RMS log measure of difference and correlation coefficient to quantitatively validate the results obtained from a finite element model of a strong-post W-beam guardrail system and a pick-up truck [39]. In 2004 Plaxico, among various comparison metrics, computed also the NARD metrics to quantify the level of validation of finite element models used to study impacts against curb-barrier systems [40] and, in 2005,

Atahan and Cansiz used the relative absolute difference of moments between two signals to quantitatively evaluate the accuracy of the results from a baseline simulation of a full-scale test between a bridge rail-to-guardrail transition structure and a pick-up truck [41].

#### **2.2.1.2.1.2 Analysis-of-Variance (ANOVA) metrics**

ANOVA is a set of standard statistical techniques that is commonly used in the analysis of statistical data and for building regression models. ANOVA metrics are based on the assumption that if two curves do, in fact, represent the same event then any differences between the curves must be attributable only to random experimental error [42,43]. The analysis of variance is a collection of statistical models and their associated procedures that assesses whether the variance between two curves can be attributed to random error only.

#### *Ray*

Ray suggested a series of simple statistical tests based on an analysis of the variance of the residuals (i.e., differences) between repeated crash test acceleration histories [42]. Conceptually, if two curves represent the same physical event, the mean residual error and the corresponding standard deviation should be both null but, in practice, due to the presence of random experimental or numerical errors these two quantities are not exactly equal to zero. Hence, if the residuals are truly random, then they should be normally distributed around a mean error of zero (i.e., typical bell-shaped Gaussian distribution). The assumption that residuals are normally distributed about a mean of zero can be examined by means of a paired t-tail test performed with the mean and standard deviation:

$$T = \frac{\bar{e}}{\sigma/\sqrt{n}} \quad (2.16)$$

where  $\bar{e}$  is the average residual between the two curves,  $\sigma$  is the standard deviation of the residuals and  $n$  is the number of paired samples. For convenience in comparing different types of impacts, the average residual  $\bar{e}$  and the standard deviation of residuals  $\sigma$  may be divided by the maximum observed experimental value (e.g., the peak measured acceleration) to obtain the relative average residual error,  $\bar{e}^r$ , and the relative standard deviation of the residual errors,  $\sigma^r$ .

The terms  $\bar{e}^r$  and  $\sigma^r$  are calculated as follows:

$$\begin{aligned}\bar{e}^r &= \frac{\sum_{i=1}^n (m_i - c_i)}{m_{max}} \\ \sigma^r &= \sqrt{\frac{1}{n} \sum_{i=1}^n (m_i - c_i - \bar{e}^r)^2}\end{aligned}\tag{2.17}$$

When two time histories represent the same physical event, both time histories should be identical such that  $\bar{e}$  and  $\sigma$  are zero, but this is almost never the case in practical situations since experimental error causes small variations between tested responses. The conventional T statistic provides an effective method for testing the assumption that the observed  $\bar{e}$  is close enough to zero to represent only experimental error. In fact, the *t*-test indicates if the differences between the two responses can be reasonably attributed to normal experimental error without having a series of repeated tests. One of the biggest advantages of the *t*-test is that it requires only two curves: a test curve and a simulation curve.

In order to correctly evaluate the residuals, it is important that the two time histories are correctly paired. In case there is a random offset between the two time histories, the most probable starting point can be obtained using the method of the least squares, analogous to the standard use of least squares in surveying to balance a traverse so that it closes. Although synchronizing the two signals was discussed in the previous section regarding the Knowles and Gear metrics, the method of least squares is a better approach because it is not based on an arbitrary point in the curve as Knowles and Gear assumed for the TOA metric. With the method of least squares, the residual area is calculated and the curves are shifted in time with respect to each other until the error (i.e., the area of the residuals) is at a minimum. This point is the statistically most likely point of synchronization. Ray implemented the least squares method to find the most likely synchronization point in a computer program called “ctrp”.

The analysis of residuals should be performed only on measured time histories and not on time histories mathematically derived from primary measurements (e.g., velocity obtained from the integration of the acceleration). In fact, certain numerical operations such as integration cause an accumulation of the residuals that are supposed to be independent from one instant to another. While Ray discusses this explicitly, this is really the case for all the metrics discussed in this section in which sensor data is used to compare curves. Comparisons (and therefore also validations) should always be made using the original data from the sensor (e.g., accelerations

from an accelerometer, rotation rates from a rate transducer or displacements from a string-pot displacement transducer).

Ray proposed the following acceptance criteria:

- The average relative residual (i.e.,  $\bar{\epsilon}^r$ ) should be less than 5 percent,
- The standard deviation of the residuals (i.e.,  $\sigma^r$ ) should be less than 20 percent and the  $t$ -statistic should be calculated between the test and simulation curve. The absolute value of the calculated  $t$  statistic should be less than the critical  $t$ -statistic for a two-tailed  $t$ -test at the 5-percent level,  $t_{0.005, \infty}$  (90<sup>th</sup> percentile).

Once the mean and variance of the residual distribution are known, they can be used to plot an envelope around the average response (i.e., the admissible error corridor). In order to be considered the same events, the curve obtained from the simulation should be always inside the above defined corridor. The analysis of variance method involves three deterministic metrics with acceptance criteria that are based on the probabilistic distribution of expected variation between crash tests.

Apart from the work in which Ray proposed the ANOVA metrics [42], he also applied this method as a validation procedure in another project [40]. Some other authors in the roadside safety literature applied the ANOVA metrics as well. In 1998, Sean and Wekezer applied this metric to compare the results from a finite element simulation and a full crash of a pick-up truck against a G2 guardrail [44]. In 2005, Atahan and Cansiz applied the analysis of variance metrics to compare a baseline finite element model of a full-scale test between a bridge rail-to-guardrail transition and a pick-up truck [41].

#### *Oberkampf and Barone*

Oberkampf and Barone developed validation metrics based on the same concept of statistical confidence intervals [45]. They developed two specific metrics: one requiring interpolation of experimental data and another requiring regression (i.e., curve fitting) of experimental data. Though they developed it independently, Oberkampf and Barone's method, as will be shown shortly, is identical to the method proposed by Ray and discussed in the previous paragraphs.

As Oberkampf and Barone were interested in an error measure between a deterministic computational result and the mean of the population of experimental measurements, their key issue was the statistical nature of the sample mean of the measured response of the system. In other words, they were particularly concerned with a statistical estimate of a confidence interval for the true mean of the residuals.

They first defined and constructed a statistical confidence interval for the population mean using sampled quantities for the mean and standard deviation:

$$\mu \sim \left( \bar{y} - z_{\alpha/2} \cdot \frac{s}{\sqrt{n}}, \bar{y} + z_{\alpha/2} \cdot \frac{s}{\sqrt{n}} \right) \quad (2.18)$$

where  $\bar{y}$  and  $s$  are respectively the sample mean and standard deviation based on  $n$  observations and  $z_{\alpha/2}$  is the value of the random variable  $z$  (i.e., standardized random variable) for which the integral of  $Z$  from  $z_{\alpha/2}$  to  $+\infty$  is equal to  $\alpha$ . From standard statistics, the level of confidence that  $\mu$  is in the interval given by the Equation (2.18) is  $100(1/\alpha)$  percent.

As the number of observations in an experiment is usually limited, they used a  $t$  distribution instead of a normal distribution resulting in the following test:

$$\mu \sim \left( \bar{y} - t_{\alpha/2, \nu} \cdot \frac{s}{\sqrt{n}}, \bar{y} + t_{\alpha/2, \nu} \cdot \frac{s}{\sqrt{n}} \right) \quad (2.19)$$

where  $t_{\alpha/2, \nu}$  is the  $1-\alpha/2$  quantile of the  $t$  distribution for  $\nu = n-1$ :degrees of freedom

The validation metric developed by Oberkampf and Barone was initially applied to the case of a scalar value and then extended to the case of a vector of values (e.g., functions of time or space).

The main idea of Oberkampf and Barone metric was to estimate an error of the computational result based on the difference  $\tilde{E}$  between the numerical solution,  $y_c$ , and the *estimated* mean of the population of the experimental results,  $\bar{y}_m$ .

$$\tilde{E} = y_c - \bar{y}_m \quad (2.20)$$

The second step to build their metric was to compute an interval containing the true error at a specified level of confidence. In order to achieve this target, the confidence interval expression

defined by Equation (2.19) was rewritten as an inequality relation using the above mentioned notation:

$$\bar{y}_m - t_{\alpha/2, v} \cdot \frac{s}{\sqrt{n}} < \mu < \bar{y}_m + t_{\alpha/2, v} \cdot \frac{s}{\sqrt{n}} \quad (2.21)$$

where  $\mu$  is the true mean and  $s$  is the sample standard deviation given by:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_m^i - \bar{y}_m)^2} \quad (2.22)$$

Multiplying Equation (2.21) by (-1) and adding  $y_c$  to each term, it becomes:

$$y_c - \bar{y}_m + t_{\alpha/2, v} \cdot \frac{s}{\sqrt{n}} > y_c - \mu > y_c - \bar{y}_m - t_{\alpha/2, v} \cdot \frac{s}{\sqrt{n}} \quad (2.23)$$

Defining the true error as:

$$E = y_c - \mu \quad (2.24)$$

the inequality relation in Equation (2.23) can be further rewritten as:

$$\tilde{E} + t_{\alpha/2, v} \cdot \frac{s}{\sqrt{n}} > E > \tilde{E} - t_{\alpha/2, v} \cdot \frac{s}{\sqrt{n}} \quad (2.25)$$

The inequality in Equation (2.25) represents an interval containing the true error with a level of confidence of  $100(1 - \alpha)\%$ .

Using a traditional level of confidence of 90%, the metric described by Equation (2.25) becomes:

$$\tilde{E} + t_{0.05, v} \cdot \frac{s}{\sqrt{n}} > E > \tilde{E} - t_{0.05, v} \cdot \frac{s}{\sqrt{n}} \quad (2.26)$$

Considering now the case for which the measured and the computed values are function of an input variable  $x$  (e.g., acceleration vs. time), the following assumptions are necessary:

- The mean value of both the computed and measured results is obtained using a sufficient number of values over the range of the input variables.
- The input variables from the experiment are measured much more accurately than the measured experimental values.

- Two or more experimental replications have been obtained and each replication has multiple measurements of the variable of interest over the range of the input values.
- The measurement uncertainty is given by a normal distribution.
- There is no correlation or dependence between one experimental replication and another.

With the previous assumptions, the metric described by Equation (2.26) can be easily rewritten as

$$\tilde{E}(x) + t_{0.05,v} \cdot \frac{s(x)}{\sqrt{n}} > E > \tilde{E}(x) - t_{0.05,v} \cdot \frac{s(x)}{\sqrt{n}} \quad (2.27)$$

where the standard deviation  $s(x)$  is defined as:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_m^i(x) - \bar{y}_m(x))^2} \quad (2.28)$$

Examination of Oberkampf and Barone's method and the ANOVA method presented by Ray and discussed in the previous section shows that they are, in fact, identical. Barone and Oberkampf's term  $\bar{y}_m$  is identical to Ray's  $\bar{\epsilon}^r$ ; they are both the mean value of the residuals between the computed and experimental curve. Likewise, Barone and Oberkampf's  $s$  is identical to Ray's  $\sigma^r$ ; both represent the standard deviation of the residuals. Barone and Oberkampf and Ray then use the same standard statistical test, the t-test to test the hypothesis that the experimental and computational curves are the same within the expected variation of the residuals.

### 2.2.1.2.1.3 Other single-value metrics

Apart from developing a multiple-value metric which is described in the next section, Russell recently collected and examined various comparison metrics in order to underline the strength and weakness of each of them [46]. He listed the following single-value metrics characterized by different formulation and properties: (a) Correlation Coefficient metric [47], (b) Weighted Integrated Factor [48], (c) Zilliacus error metric [49], (d) RSS error metric [49], (e) Theil's inequality metric [50], (f) Whang's inequality metric [49] and (g) Regression Coefficient metric [47]. The formulation of these metrics is listed in Table 2.3. The first two metrics are based on integral comparisons while the others are based on a point-to-point comparison (i.e., they are calculated based on the residuals between the two curves at each time step).

**Table 2.3. Definition of various single-value metrics**

<i>Integral comparison metrics</i>			
<i>Correlation Coefficient</i>	$\frac{n \sum c_i m_i - \sum c_i \sum m_i}{\sqrt{n \sum c_i^2 - (\sum c_i)^2} \sqrt{n \sum m_i^2 - (\sum m_i)^2}}$		
<i>Weighted Integrated Factor</i>	$\sqrt{\frac{\sum \max(m_i^2, c_i^2) \cdot \left(1 - \frac{\max(0, m_i \cdot c_i)}{\max(m_i^2, c_i^2)}\right)^2}{\sum \max(m_i^2, c_i^2)}}$		
<i>Point-to-point comparison metrics</i>			
<i>Zilliacus error</i>	$\frac{\sum  c_i - m_i }{\sum  m_i }$	<i>Whang's inequality</i>	$\frac{\sum  c_i - m_i }{\sum  c_i  + \sum  m_i }$
<i>RSS error</i>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum m_i^2}}$	<i>Regression coefficient</i>	$\sqrt{1 - \frac{(n-1) \sum (c_i - m_i)^2}{n \sum (m_i - \bar{m})^2}}$
<i>Theil's inequality</i>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum c_i^2} + \sqrt{\sum m_i^2}}$		

### 2.2.1.2.2 Multiple-value metrics

Multiple-value metrics are composed by multiple components which are capable to analyze separately different aspects of the comparison between the two responses. In this sense, multiple-value metrics may distinguish the contribution of the difference in magnitude or phase between the two curves or assess the level of agreement at different topological levels (absolute or relative peaks, edges, etc.).

#### 2.2.1.2.2.1 MPC metrics

Different formulation of MPC metrics can be found in the literature. Apart from the specific mathematical formulation which distinguishes one metric from the other, all the MPC-type metrics are composed by three components: (i) Magnitude, (ii) Phase and (iii) Comprehensive component. The first two components, M and P, compare respectively the relative magnitude and phasing of the two curves. In fact, the M component is sensitive to differences in magnitude, but relatively insensitive to difference in phasing and vice versa for the P component. The M and P components are then combined into a single comprehensive metric, C, which represents the combined effect of both magnitude and phase. These characteristics give the MPC metrics a good diagnostic value since they allow to identify the aspects of the curves that do not agree. For

example, if the phase metric is acceptable but the magnitude metric is not, the analyst can examine the stiffness and strength of the model to make sure it is correct.

The following five different formulations of MPC metrics can be found in the literature:

- Geers Metric
- Sprague & Geers Metric
- Geers CSA
- Knowles and Gear Metric
- Russell Metric

Some of the MPC metrics are non-symmetric (i.e., they produce different values when the measurement and calculated response are interchanged). Nonetheless, when comparing an experimental curve to a numerical result, this does not represent a problem as the measured curve is always the one which is considered as the basis.

A detailed description of the formulation for each of the five MPC metrics is given in the following sections.

### *Geers Metrics*

Geers developed the first MPC metric that includes quantitative assessments of the magnitude and phase which are then combined into a single value [51]. The M and P components of the Geers metric are based on the following time integrals:

$$\begin{cases} \vartheta_{mm} = (t_2 - t_1)^{-1} \int_{t_1}^{t_2} m^2(t) dt \\ \vartheta_{cc} = (t_2 - t_1)^{-1} \int_{t_1}^{t_2} c^2(t) dt \\ \vartheta_{mc} = (t_2 - t_1)^{-1} \int_{t_1}^{t_2} m(t)c(t) dt \end{cases} \quad (2.29)$$

where,  $m(t)$  and  $c(t)$  are, respectively, the measured and the computed time histories and  $t_1$  and  $t_2$  are the initial and final time, respectively.

The phase component  $P_G$  is computed as follows:

$$P_G = 1 - \left( \vartheta_{mc} / \sqrt{\vartheta_{mm}\vartheta_{cc}} \right) \quad (2.30)$$

Geers showed how this component metric is insensitive to magnitude differences but is sensitive to differences in phasing or timing between the two time histories.

The magnitude component  $M_G$  of the Geers metric is defined as:

$$M_G = g_{cc}/g_{mm} - 1 \quad (2.31)$$

Note that both the metric components  $P_G$  and  $M_G$  result in a zero value when the two histories are identical.

Once the magnitude component and the phase component have been evaluated, the combined Sprague and Geers metric is evaluated by combining the two component metrics into a single value:

$$C_G = \sqrt{M_G^2 + P_G^2} \quad (2.32)$$

It can be observed that the magnitude and phase components of the Geers metric can be thought of as coordinates of a circle where the  $P_G$  and  $M_G$  values represent the coordinates of a point on the circle and C defines the radius.

As the curves measure from experimental tests or computed by numerical simulations are discrete, the previous analytical formulation for the  $P_G$  and  $M_G$  components can be reduced to the following discrete formulation:

$$P_G = 1 - \left[ \frac{\sum_{i=1}^N m_i c_i}{\sqrt{\sum_{i=1}^N m_i^2 \sum_{i=1}^N c_i^2}} \right] \quad (2.33)$$

$$M_G = \frac{\sum_{i=1}^N c_i^2}{\sum_{i=1}^N m_i^2} - 1 \quad (2.34)$$

A slight variation of the phase component respect to the original Geers metric was developed during the SEAWOLF Main Propulsion Unit Comparative Shock Analysis Program (CSA). In order to provide additional information about possible discrepancies between the two curves, the absolute value of the numerator in Equation (2.33) was considered. Equation (2.35) and Equation (2.36) show respectively the analytical and discrete formulation of the M component for the Geers CSA variation.

$$P_{G\_CSA} = 1 - \left( \frac{|\vartheta_{mc}|}{\sqrt{\vartheta_{mm}\vartheta_{cc}}} \right) \quad (2.35)$$

$$P_{G\_CSA} = 1 - \left[ \frac{|\sum_{i=1}^N m_i c_i|}{\sqrt{\sum_{i=1}^N m_i^2 \sum_{i=1}^N c_i^2}} \right] \quad (2.36)$$

Apart for the modification of the phase component, also the computation of the comprehensive component was changed as it is multiplied by a positive or negative sign according to Equation (2.37).

$$\text{sign}(\sum c_i m_i) \sqrt{M_{CSA}^2 + P_{CSA}^2} \quad (2.37)$$

Recently, Geers has developed a variation of his original metric in collaboration with Sprague. The magnitude and phase components of the Sprague-and-Geers modified version of the metric are based on the same time integrals  $\mathcal{G}_{mm}$ ,  $\mathcal{G}_{cc}$  and  $\mathcal{G}_{mc}$  of Equation (2.29) which were initially defined for the original Geers metric; however the phase component has been slightly modified. In fact, Sprague and Geers noticed that the phase component of the original Geers metric did not scale similar to the magnitude component [43]. A magnitude component of 10 percent, for example, did not reflect the same degree of comparability as a phase component of 10 percent. Sprague and Geers modified the original phase component of the metric to include a trigonometric term that helped the two components scale more similarly. Sprague and Geers used a phase formulation based on Russell [52], a metric discussed later in this section. The Sprague and Geers metric is structured in the same way as the original version with magnitude, phase and combined metrics. The three components of the Sprague and Geer metric are given by the following equations:

$$P_{SG} = \frac{1}{\pi} \cos^{-1} \left( \frac{\vartheta_{mc}}{\sqrt{\vartheta_{mm}\vartheta_{cc}}} \right) \quad (2.38)$$

$$M_{SG} = \sqrt{\mathcal{G}_{cc} / \mathcal{G}_{mm}} - 1 \quad (2.39)$$

$$C_{SG} = \sqrt{M_{SG}^2 + P_{SG}^2} \quad (2.40)$$

Also in this case, the phase-component metric is insensitive to magnitude differences but is sensitive to differences in phasing or timing between the two time histories while the magnitude component metric is insensitive to phase discrepancies, as it is based upon the area under the squared response histories.

The discrete formulation for the  $P_{SG}$  and  $M_{SG}$  components respectively shown in Equation (2.38) and Equation (2.39) can be reduced to the following discrete formulation:

$$P_{SG} = \frac{1}{\pi} \cos^{-1} \left[ \frac{\sum_{i=1}^N m_i c_i}{\sqrt{\sum_{i=1}^N m_i^2 \sum_{i=1}^N c_i^2}} \right] \quad (2.41)$$

$$M_{SG} = \sqrt{\frac{\sum_{i=1}^N c_i^2}{\sum_{i=1}^N m_i^2}} - 1 \quad (2.42)$$

### *Knowles and Gear Metric*

The Knowles and Gear metric is the most recent variation of the MPC-type metrics [43]. Like Geers, Geers CSA and Sprague and Geers metrics, also the Knowles and Gear metric is composed of three parts: a phase component, a magnitude component and a comprehensive combination of both. As in the other versions of the MPC metrics, a value equal to zero represents an exact agreement between the curves. Knowles and Gear recognized that, if two shapes were arbitrarily shifted from each other, the phase component may erroneously indicate poor phase correlation that was simply due to not synchronizing the two signals. For example, if the time of impact was not precisely defined in the experimental curve, some of the phase error might actually be due to the poor identification of the impact point. Before a good comparison of the phasing can be performed, the two signals must be synchronized so they start at the same point.

In the Knowles and Gear metric, the synchronization of the signals is accomplished by defining the time of arrival (*TOA*). The *TOA* of a time history is the time at which the time history attains some percentage of the maximum wave form value. Typically, for time histories with relatively fast rise times, a range of 5–10% is recommended, but this range may be changed in case of slower rise times. Defining  $TOA_c$  and  $TOA_m$  as the times of arrival of the measured and simulation time histories, then the *TOA* metric is defined as:

$$M_{TOA} = \frac{|TOA_c - TOA_m|}{TOA_m} \quad (2.43)$$

The magnitude component  $M_{KG}$  of the Knowles and Gear metric is defined as a weighted sum-of-squared differences between the simulated and measure time histories. Considering a discrete time history characterized by  $N$  time samples,  $M_{KG}$  is defined as:

$$M_{KG} = \sqrt{\sum_{i=1}^N Q_i (\tilde{c}_i - m_i)^2 / QS} \quad (2.44)$$

where,  $m_i$  is the measured time history at the  $i^{\text{th}}$  sample and  $\tilde{c}(t) = c(t - \tau)$  is the time-of-arrival shifted simulation history (i.e., if  $TOA_c$  and  $TOA_m$  are the times of arrival of the measured and simulation time histories and  $TOA_c > TOA_m$ , then  $\tau = TOA_c - TOA_m$ ). Shifting the simulation time history using the function  $\tilde{c}(t)$  instead of  $c(t)$  allows the metric to focus only on the magnitude comparison between the curves without complications arising from the asynchronous signals.  $Q_i$  and  $QS$  represent respectively the weighting and normalization factors.

The weighting factor is designed to scale the sum-of-squares differences to the maximum value of the measurement to the maximum value of the measurement:

$$Q_i = \left( \frac{|m_i|}{m_{max}} \right)^p (t_{i+1} - t_{i-1}) \quad (2.45)$$

where, a unit value of  $p$  is recommended to place more weight on the large values of  $m(t)$ .

In order to avoid creating a gap between time histories characterized by a large magnitude and those characterized by a smaller one, the magnitude has to be normalized. In this metric, the normalization factor  $QS$  is chosen to define a value of unity when the magnitude of the time histories differs by 100 percent:

$$QS = \sum_{i=1}^N \left( \frac{|m_i|}{m_{max}} \right)^p m_i^2 (t_{i+1} - t_{i-1}) \quad (2.46)$$

If a uniform time sampling is chosen, the magnitude component  $M_{KG}$  defined by Equation (2.44) simplifies to the following form:

$$M_{KG} = \sqrt{\frac{\sum_{i=1}^N \left( \frac{|m_i|}{m_{max}} \right)^p (\tilde{c}_i - m_i)^2}{\sum_{i=1}^N \left( \frac{|m_i|}{m_{max}} \right)^p (m_i)^2}} \quad (2.47)$$

In an analogous case as for the Geers metrics, once the magnitude component and the phase component have been evaluated, the Knowles and Gear combination metric is evaluated by combining the two component metrics by mean of a weighted average:

$$C_{KG} = \sqrt{\frac{10M_{KG}^2 + 2M_{TOA}^2}{12}} \quad (2.48)$$

In the Knowles and Geer combined metric, the magnitude and phase factors are weighted such that the phase value does not dominate the combined metric. The main limitation of the Knowles and Gear metric is that it cannot differentiate between an under or over prediction because it is based on the sum of the square differences between the measured and the simulation curve. The Knowles and Gear metric can be also applied in a more general case when different system response quantities are considered at the same time.

#### *Russell Metric*

Another metric based on the concept of magnitude and phase differences between two curves was developed by Russell in 1997 [52]. Russell defined the relative magnitude error between the measured and computed curve as:

$$m = \frac{\sum_{i=1}^N m_i^2 - \sum_{i=1}^N c_i^2}{\sqrt{\sum_{i=1}^N m_i^2 \cdot \sum_{i=1}^N c_i^2}} \quad (2.49)$$

The phase correlation between the measured and computed curve is:

$$p = \frac{\sum_{i=1}^N m_i c_i}{\sqrt{\sum_{i=1}^N m_i^2 \cdot \sum_{i=1}^N c_i^2}} \quad (2.50)$$

Hence Russell derived the magnitude and phase error respectively from the corresponding relative magnitude error and phase correlation. The resulting form for the magnitude error is:

$$M_R = \sin(m) \log_{10}(1 + |m|) \quad (2.51)$$

The phase error is computed as:

$$P_R = \frac{\cos^{-1}(p)}{\pi} \quad (2.52)$$

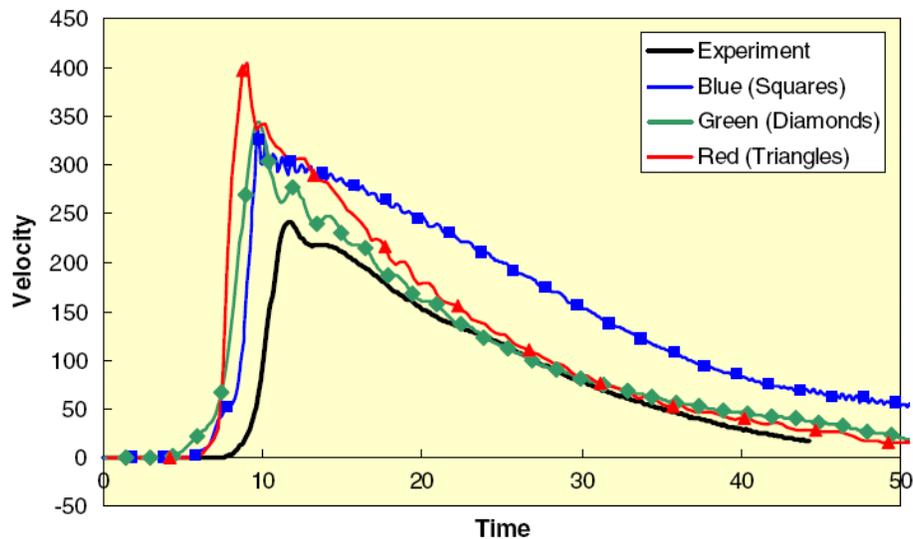
The comprehensive error of the Russell's metric is defined as:

$$C_R = \sqrt{\frac{\pi}{4}(M_R^2 + P_R^2)} \quad (2.53)$$

### *Applications of MPC metrics*

This section presents the applications of some of the above described MPC metrics.

Recently, Schwer [43] showed the worthiness of the Sprague and Geers and the Knowles and Gear metrics both comparing some pairs of simple analytical curves differing by a known time delay or magnitude factor and comparing the experimental and numerical velocity wave form in a geological medium due to the application of an energy from a nearby source (Figure 2.4). The results obtained using both Sprague and Geers and Knowles and Gear metrics to compare each of the three numerical curves with the experimental one are presented in Table 2.4.

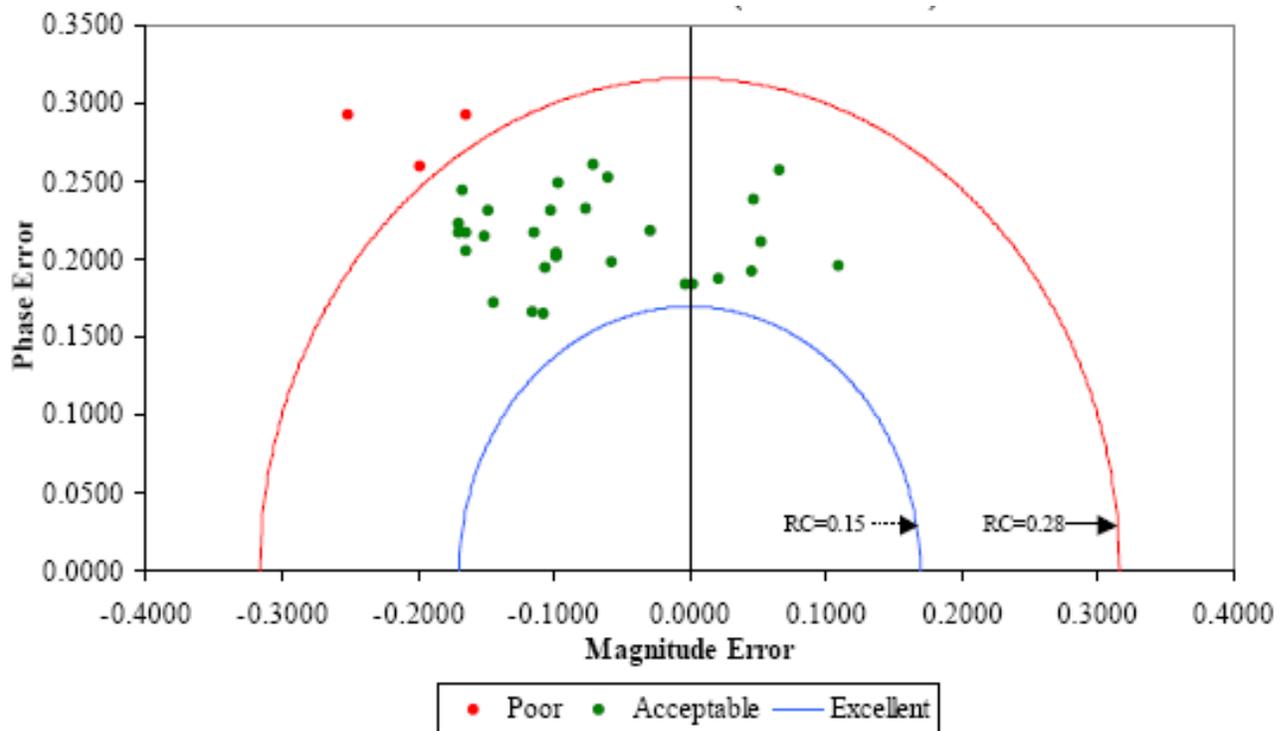


**Figure 2.4:** Comparison of measured velocity wave form with three simulation results [43].

**Table 2.4. Metric components for the three simulation curve of Figure 2.4**

	Sprague and Geers			Knowles and Gear		
	<i>M</i>	<i>P</i>	<i>C</i>	<i>M</i>	<i>P</i>	<i>C</i>
<b>Blue (Squares)</b>	0.60	0.08	0.61	0.54	0.17	0.50
<b>Green (Diamonds)</b>	0.26	0.13	0.29	0.27	0.23	0.26
<b>Red (Triangles)</b>	0.45	0.15	0.47	0.48	0.21	0.45

Shin and Schneider used the Russell metrics to evaluate the blastworthiness of a naval ship. An experiment using the DDG51 class vessel USS *Winston Churchill* was replicated with an LS-DYNA model [53]. The ship was instrumented with 30 accelerometers at various locations to measure the local accelerations in the ship hull when the vessel was in the vicinity of a blast.

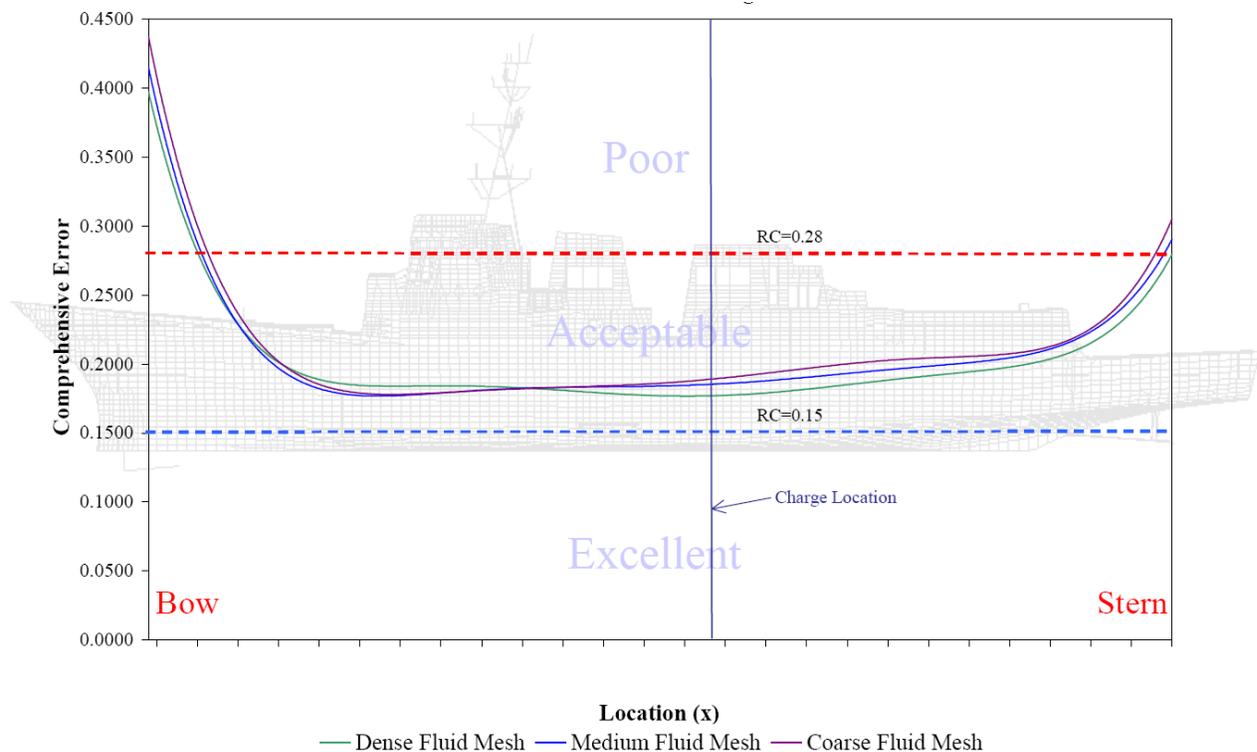


**Figure 2.5: Russell metrics for 30 accelerometers in a ship blast model validation [53].**

The magnitude and phase components of the Russell metrics were computed and then the results for all the sensors were plotted as shown in Figure 2.5. The two component metrics can be considered coordinates of a point with the magnitude component plotted on the x axis and the phase component on the y axis. The radial distance to the origin represents the combined metric. The RC (Russell Combined) values shown in Figure 2.5 represent the values of the combined metric for two different acceptance levels. If the combined metric is less than 0.15 (i.e., 15

percent) the comparison is excellent. If the combined metric is between 0.15 and 0.28, the comparison is acceptable whereas if the combined metric is greater than 0.28, the comparison is unacceptable. Figure 2.5 shows not only of the use of the Russell metrics but the combination of sensor data from a number of different sensors on the same plot to assess the overall utility of a model. For example, the three data points outside the acceptable range in Figure 2.5 might not invalidate the whole model but they call the analyst's attention to regions in which the experiment and calculation did not agree.

Shin and Schneider also plotted the Russell combined metric against the longitudinal position of the sensor as shown in Figure 2.6. The figure clearly shows that the results of the comparison degraded at sensors located at the extreme ends of the vessel. This plot helps the analyst and the experimenter to identify problems with the model or with the location and mounting of sensors that can be used to improve both subsequent experiments and model development.



**Figure 2.6: Russell combined metric plotted versus the longitudinal position of the sensor in a ship blast validation activity [53].**

In roadside safety, the only MPC metric which appears to have been used in rare occasions is the original Geer metric. Plaxico and Ray computed the Geer metric to quantitatively compare the

acceleration time histories computed from numerical simulations with the histories measured from the corresponding experimental tests [40].

#### 2.2.1.2.2 Global Evaluation Method (GEM)

The GEM is a complex comparison metric based on three components which was developed by Jacob *et al.* [54] to compare two time histories in biomechanics. This metric takes into account both the local characteristics and the global characteristics of the two curves. In particular, the local characteristics are divided into two different typologies: (1) **point class** and (2) **shape class**. The point class is related to the local minima and the maxima, while the shape class is related to the local rising/falling edges. Global characteristics are also defined as the third and last class: the **global class**. The main concept is that a score is evaluated for each of the three classes and these partial scores are then combined together using a weighted mean to obtain a final score for the couple of curves.

The score for the **point class** is evaluated by first identifying in the true curve up to four local minima/maxima. Once these local characteristics have been chosen in the true curve they are searched in the test curve. For each point found in the test curve two different sub-scores are computed: (i) a time score comparing the time at which the two extrema occur and (ii) a value score comparing the value of the two extrema. The analytical expression of these two sub-scores is shown in Equation (2.54) and Equation (2.55), respectively. If the extremum is not found in the test curve, its time and value scores are set to zero.

$$Time_{Score} = \left(1 - \frac{|Time_{True\_Extremum} - Time_{Test\_Extremum}|}{Reference\_Period}\right) \cdot 100 \quad (2.54)$$

$$Reference_{Period} = 40\% \text{ Evaluation Period}$$

$$Value\_Score = \left(1 - \frac{|Value_{True\_Extremum} - Value_{Test\_Extremum}|}{Value_{True_{Extremum}}}\right) \cdot 100 \quad (2.55)$$

All the sub-scores obtained using these formulas are then weighted and a single point class score is computed ranging from 0% (extremely bad) to 100% (perfect).

In order to evaluate the score for the **shape class** both the test and true curves of each channel are filtered two times: once with a SAE J211 compliant filter and a second time with a low pass Butterworth filter with a corner frequency of 40 Hz. The main concept is that the shape of a curve can be decomposed into one or two rising/falling edges and one or two local extrema.

Time scores for the extrema are calculated using Equation (2.54), while time and value scores for the edges use Equation (2.56) and Equation (2.57) respectively. Also in this case, the shape class score is obtained by taking the average of all the time and value scores.

$$Time\_Score_{Edge} = \left(1 - \frac{|Time_{True\_Edge} - Time_{Test\_Edge}|}{Reference\_Period}\right) \times 100\% \quad (2.56)$$

$$Value\_Score_{Edge} = \left(1 - \frac{\min(Slope_{True}, Slope_{Test})}{\max(Slope_{True}, Slope_{Test})}\right) \times 100\% \quad (2.57)$$

As for the point characteristics, if a feature is not found in the simulation channel, both the time and value scores are set to zero.

The scores for the **global class** consists of two factors: (i) the normalized cross-correlation function (NCCF) and (ii) a shape corridor.

The NCCF defines the correlation between the test and true curves as a function of the time shift by which the test curve has been arbitrarily moved. The time at which the maximum value of NCCF is reached represents the time offset (phasing) between the curves. For the shape corridor, the input curves are filtered twice as in the shape group, and a corridor is defined based on the test data surrounded by a square rectangle with dimensions of +/- 5%. Similarly to the previous two classes, a time and a value score are computed for both the NCCF and shape corridor and then are combined to give the global score.

Eventually, the sub-scores for the three classes (point, shape and global) are combined together in a final score computing a weighted average. Usually the weight of the point class counts twice the weight of the shape and global class.

### 2.2.2 Stochastic metrics

In the previous section describing ANOVA techniques and metrics, there was an implicit assumption that the computational results were deterministic (i.e., the same result occurs given the same input) but the experimental results were probabilistic (i.e., substantially identical experiments can result in a range of outputs due to random experimental error). Barone and Oberkampf, Oberkampf and Trucano, and Ray all developed metrics that examined the computational results to see if they fall within the expected probabilistic range of experimental results.

Computational results, however, need not be deterministic. Every computed result is based on input data like material properties, geometry and initial conditions. If these input parameters are varied, the results of the computation will likewise vary. For example, an analyst may use the yield stress reported in an engineering handbook to develop a computational model. Although the real physical material exhibits random variation in its properties, the analyst usually just assumes the mean value of the parameter (i.e., the handbook value). Stochastic methods, on the other hand, treat the input to the finite element model as parameters that can experience random variation. If mean values are used to perform the simulation, as is usually the case, the result is a deterministic mean response. If the input variables are allowed to vary randomly as they do in the physical world, the simulation response would vary as well. The idea of stochastic variation of parameters is a key component of computational optimization. If three computations were performed at, say, the minimum, mean and maximum yield stress, the response of the simulation would likewise vary. If the variation of the computation is compared with the expected variation in the experiments a stochastic comparison technique is being used.

The brute-force way of performing a Bayesian or stochastic analysis is to first characterize the probabilistic distributions of all the variables in the model (i.e., Young's moduli, yield strains, rate affects, densities, etc.). Next, specific parameter values can be randomly selected using a Monte Carlo technique and then a trial simulation is performed and the response of interest is generated (e.g., acceleration time history). The process is repeated with another set of randomly selected variations and the simulation is performed again. This process continues until the analyst has adequately characterized the response. The result will be a variety of acceleration time histories that are similar but vary in random ways much as experiments do. Obviously, this method requires a great deal of computer run time and a great deal of analyst intervention so it is not very practical except for small problems.

Techniques for estimating the response under Bayesian parameters variations have been developed in the field of structural optimization that can reduce the computational demand somewhat. Several authors have proposed techniques and processes for performing such analyses for crashworthiness problems but there are relatively few examples of practical applications and none dealing with large models as would be typical of roadside safety applications [55,56]. Faravelli used this technique to perform a stochastic analysis of the front frame of a vehicle [57]. In her case she only varied the angles on the frame horn to observe the

change in responses. In fact, also Patzner did something like this when he examined the response variation due to changes in soil parameters and timber materials properties for guardrail posts [19]. He varied the soil and wood properties in a model of the MELT guardrail terminal and computed the range of responses. The result, summarized in Figure 2.7, shows a plot of the soil density versus the maximum guardrail deflection at the rail height. He found that certain vehicle responses were associated with different combinations of the soil and wood material properties. If, for example, Grade 1 dense wood post were used in a poorly compacted soil (i.e., low density), the vehicle tended to snag on the guardrail post. If weak Grade 2 wood posts were used in over consolidated soils (i.e., high density), the chance of guardrail rupture was maximize. In Patzner's case only a half dozen or so material parameters were being varied but the run time to produce the plot in Figure 2.7 was significant.

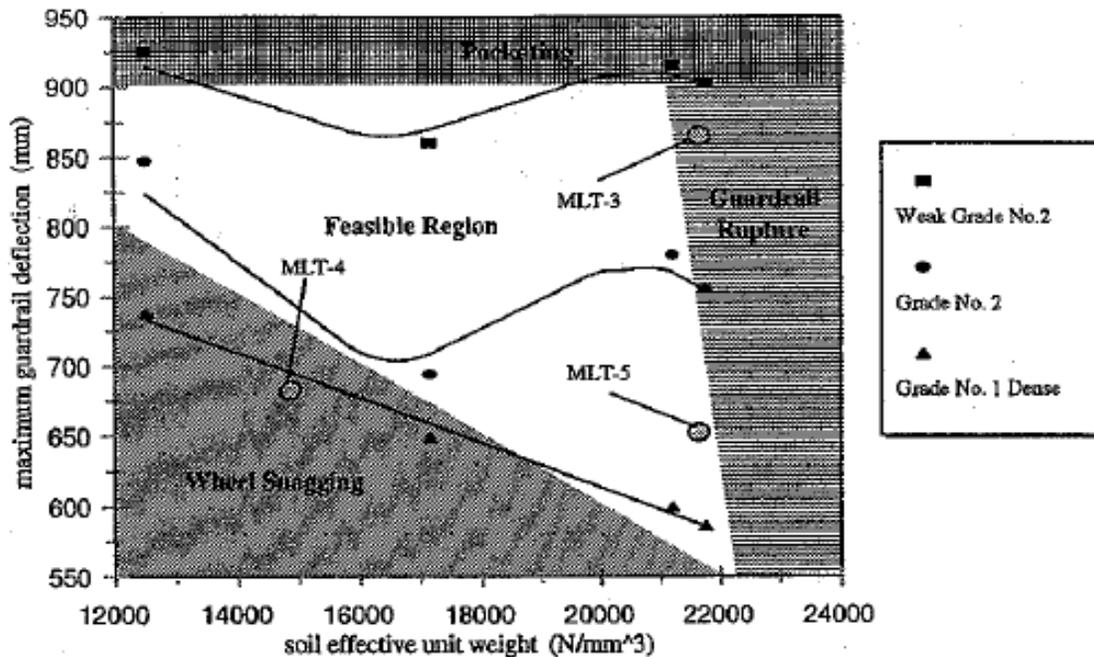


Figure 2.7: Effect of soil and wood material parameter variation on the performance of MELT guardrail terminal [19].

### 2.3 Computer programs for the evaluation of comparison metrics

A brief description of the computer codes previously developed to automatically assess how well two curves compare to each other is presented in this section. The following four softwares have been found in the literature:

- Test Report Assessment Program
- Curve Analyzer
- TNO software
- ModEval.

### **2.3.1 Test Report Assessment Program (TRAP)**

TRAP is a software developed by Texas Transportation Institute [20] to automatically evaluate the domain-specific metrics described in Report 350 (OIV and ORA) [21] and EN1317 (THIV, PHD, and ASI) [33] to perform roadside crash tests. As previously mentioned, domain-specific metrics cannot be directly considered as comparison metrics, but can be used in such a manner by comparing the metric value obtained from each curve.

The domain-specific metrics can be computed using up to six input channels (three accelerations and three rotational rates). The program automatically performs two basic preprocessing operations: (i) resampling of the input curves in order to have the same sampling rate and (ii) the filtering using a SAE 60 filter.

The metric values are output through a table and, for the ASI metric, it is possible to plot a time history. Both the input and the output operations are easily managed using a graphical user interface.

After the user has input the time histories, the software automatically evaluates the metrics without any need for the user to take any subjective choice.

### **2.3.2 Curve Analyzer**

The program Curve Analyzer created at the University of Chalmers in Sweden is an example of code developed to analyze and compare two different curves [58]. This program performs both tests based on the simple curve characteristics like maximum and minimum value, and statistical tests on the residuals. The code was written using the widely used Java language in collaboration with Autoliv, a company involved in developing automotive safety systems. The activity performed by Curve Analyzer can be divided into two different steps: (1) preprocessing and (2) evaluation of different statistical tests.

In the preprocessing phase, after two or more curves have been loaded, the user is given the possibility to filter the time histories using different filter classes, modify the units of either the time or data sets through the definition of scale factors or offset the curves in either the time or data direction. In particular, the offset option can be very useful when the loaded curves have been acquired using different zero settings or they do not started at the same time. Also, the user has the possibility to specify which curves to consider and to compare the selected curves just over a segment of the total curve length.

In the second phase, the program can evaluate up to seven different tests on the selected curves. Some of these test are simply based on the curves values (like minimum and maximum) while others are based on the residuals between the two curves. Following is a list of all the seven different tests evaluated by Curve analyzer:

- Mean value of the curves
- Coefficient of correlation of the curves
- Maximum value and minimum curve values (Box-test)
- Maximum and minimum residuals
- Standard deviation of residuals
- Paired t-test of the residuals
- Non-parametric test of the residuals(signed rank test)

Although the program recommends one or more parameters/criteria for each test, these values can be modified by the user in order to satisfy specific needs.

The program is characterized by an easy graphical interface.

### **2.3.3 TNO software**

TNO Automotive Safety Solutions has recently implemented an in-house software to perform the validation of numerical dummy models used to test aircraft seating systems [48]. The software is implemented in XML language and is able to calculate a single-value metric from multiple channels. Moreover, this program can combine together the metric results obtained from many component validation tests and simulations by organizing them in a hierarchical manner. The

hierarchy is created by combining together metric values obtained from similar tests in order to form subscores for each group (e.g., one group for the component validation tests involving the head and neck system, another group for the tests involving the chest and abdomen, etc.). Having separate subscore values for groups of similar test allows to better show the weak and strong points of a numerical model.

Basically, for each channel of an experimental test and the corresponding numerical simulation three different metrics are computed: (i) the peak value, (ii) the time at which the peak is reached and (iii) the WIFac metric. These three metrics are combined together in a single value using weighting factors defined by the user. The single-value metrics from each channel are then combined in turn by computing another weighted average. Repeating the same procedure for each component validation test performed on the model, it is possible to obtain a set of scores, one for each specific test. Using the hierarchical grouping previously described and proceeding to combine together the scores obtained at the various levels, the software eventually computes a single value metric to assess the overall validation of the model from all the performed tests.

Although this software has been specifically developed for the validation of dummy models, it can be virtually employed in any other field where it is necessary to compare two time histories.

#### **2.3.4 ModEval**

Another case of a computer program developed to assess the comparison between two curves is ModEval. This software was developed by Jacob *et al.* [54] to compare the experimental and numerical curves in biomechanics but, virtually, it can be used in any other field where it is necessary to compare two time histories. In the paper describing ModEval, the authors stated that the main application of this software was the quantification of the quality of a numerical model, which is indeed exactly the same as a validation assessment. ModEval compares two curves giving the user the possibility to choose between four different analytical methods.

Initially, the user is asked to classify the couple of curves obtained respectively from an experimental test and a simulation selecting between four different categories defined as “channels”. The four channels are defined as: (1) normal (e.g., X and Z head accelerations in a typical frontal crash), (2) poor interest (e.g., Y head accelerations in a typical frontal crash), (3) criterion (e.g., head resultant accelerations in the computation of a head severity index as the

HIC) and (4) corridors (generally used when large scatter occurs, as in human being models). The experiment can be fully characterized using these four types of channels. Once each pair of time histories has been classified into the related channel, ModEval evaluates a specific metric (defined as evaluation method) base on the channel type.

The four metrics evaluated by ModEval are: (1) Global Evaluation Method (GEM), (2) Threshold Evaluation Method (TEM), (3) Criterion Evaluation Method (CEM) and (4) Limit Evaluation method (LEM). The first three metrics (GEM, TEM and CEM) can be used with numerical models, while the fourth metric (i.e., LEM) is specifically designed for comparison of tests involving human beings. The reader interested in any of these metrics can find a description in the previous subchapter and in the cited references.

Eventually, the scores obtained from the various channels are combined together using user defined weighting factors. Although this software can automatically evaluate the metrics, it still needs the initial input of the user to classify the time histories between the four available channels, thus resulting in a final score which is not completely objective because of this initial subjective decision.

## **2.4 Synthesis of the literature review**

A synthesis of the literature review described in the previous sections of this chapter is herein presented. The following two sections synthesize respectively the various comparison metrics and the computer programs developed for the comparison of curves which were found in the literature.

### **2.4.1 Classification of comparison metrics**

The many works recently performed in roadside safety using finite element simulations indicates that numerical simulations are becoming a standard methodology to design new hardware or improve existing safety systems. Although numerical simulations represent an effective tool in the design and analysis of roadside safety systems, if the models used to simulate a full-scale crash test are not correctly verified and validated results may not be correct and reliable. From the analysis of most of the numerical works published in the roadside safety literature appears that, in most cases, the verification and/or validation of the numerical models is carried out using solely subjective methods such as the visual comparison of the acceleration time histories from

an experimental test and the corresponding numerical simulation. Only in few rare occasions, some researchers in roadside safety considered to compute also comparison metrics in order to quantitatively assess the level of validation/verification of their models.

Various reasons for this deficiency of a quantitative assessment in the verification/validation approach can be found. The first and most obvious explanation is the limited knowledge of most of the comparison metrics which, generally speaking, have been rarely applied also in the other engineering fields. Even though the practitioner is aware of the existence of one or more particular comparison metrics, the lack of automatic and easy-to-use codes for the calculation of these metrics may restrain from employing them. Frequently, the quantities collected in roadside safety are time histories and this may create some difficulties in the implementation of a code for computing the metrics.

Another limit to the application of comparison metrics in roadside safety is the difficulty to assess reasonable acceptance criteria. In fact, when a comparison metric is used to validate a numerical model, potentially it has two components: (1) the deterministic metric itself and (2) an acceptance criterion. Considering the typical substantial scatter of the results obtained from crash tests, it may be necessary to base the acceptance criteria on the probabilistic variation of the experiments. This approach is better than a mere deterministic approach since it incorporates the uncertainty in physical experiments but requires some *a priori* knowledge about the level of uncertainty typical in experiments, which may be difficult to obtain as it necessitates the availability of a certain number of tests. These methods generally involve some type of classical hypothesis testing such as a t or z test or the Kolmogorov-Smirnov test [59]. For example, if 10 experiments are performed and the mean HIC is found to be 385 with a standard deviation of 11 HIC units, the analyst can estimate that the 90<sup>th</sup> percentile confidence limit for the HIC experiments is  $385 \pm 1.65 \cdot 11 = 18$  HIC units. Stated another way, if 100 tests were performed, 90 of those should be in the range of 334 to 370 HIC units. If the HIC computed from the simulation is 398, in this case, the value is outside the 90<sup>th</sup> percentile confidence range so the conclusion would be that the simulation is not sufficiently close to the experiment. If the simulation were as good as another experiment, the simulation result would have fallen in the specified range. Sometimes, if not enough data are available to determine the acceptance criterion based on a probabilistic approach, an *ad hoc* criterion may be defined based on community experience or engineering judgment. Considering again the example of the HIC,

according to the Federal Aviation Administration (FAA) experience in the specific domain, a HIC comparison that is within +/- 50 HIC units is reckoned to be sufficiently close. Based on this *ad-hoc* criterion, the results of the simulation in the previous example would be now sufficient.

As originally stated at the beginning of this sub-chapter, in general, comparison metrics can be divided into two main groups: (a) deterministic and (b) stochastic metrics. Although stochastic metrics present some advantages respect to deterministic metrics, like useful information about the range of results of a computation, their application in practical cases in roadside safety would require an excessively large amount of work to vary even a few parameters. As optimization softwares which are based on concepts very similar to the stochastic variation are improving progressively, however, stochastic metrics might become more feasible in the future. For now, stochastic methods do not appear to be practical for roadside safety validation and verification efforts.

The several deterministic metrics found in the literature have been divided in two main sub-groups: (i) domain-specific and (ii) shape-comparison metrics. Because they are specific of a particular field, metrics of the domain-specific group may have a limited application; while shape comparison metrics have a more general applicability.

The group of shape-comparison metrics is very heterogeneous and it is possible to distinguish several characteristics. In fact, from an analysis of the metrics previously described in this literature review, shape-comparison metrics may be classified according to (i) the domain in which they are defined, (ii) the method in which curves are compared, (iii) the computation of various components or a single value, (iv) the symmetry, or (v) the particular mathematical concepts behind the analysis of the residuals.

As for the domain, although most metrics use directly the value of the time histories, some of them instead are based on the equivalent Fourier transformation (frequency domain).

Two main methods to perform the comparison of the curves have emerged in this literature review: (1) point-to-point and (2) integral comparison. Metrics based on a point-to-point comparison method combine in some manner the residuals between the two curves at each sampling point; while metrics founded on the integral method compute some sort of integrals of the curves, either separately or combining them together.

Some metrics distinguish either the differences in phase or in magnitude between the two curves by calculating two components which are then combined together, whereas other metrics give purely an overall assessment of the comparison. It is worth noting that, among the reviewed metrics, the GEM metric is the only capable of comparing the curves based on both their local and global characteristics. Unfortunately, the metric complexity and the need to specify numerous weighting factors make the GEM metric difficult to be applied and sensitive to the user subjectivity.

Another possible classification is according to the symmetry of the comparison: some metrics give the same result independently by which curve is considered to be the measured or the computed one.

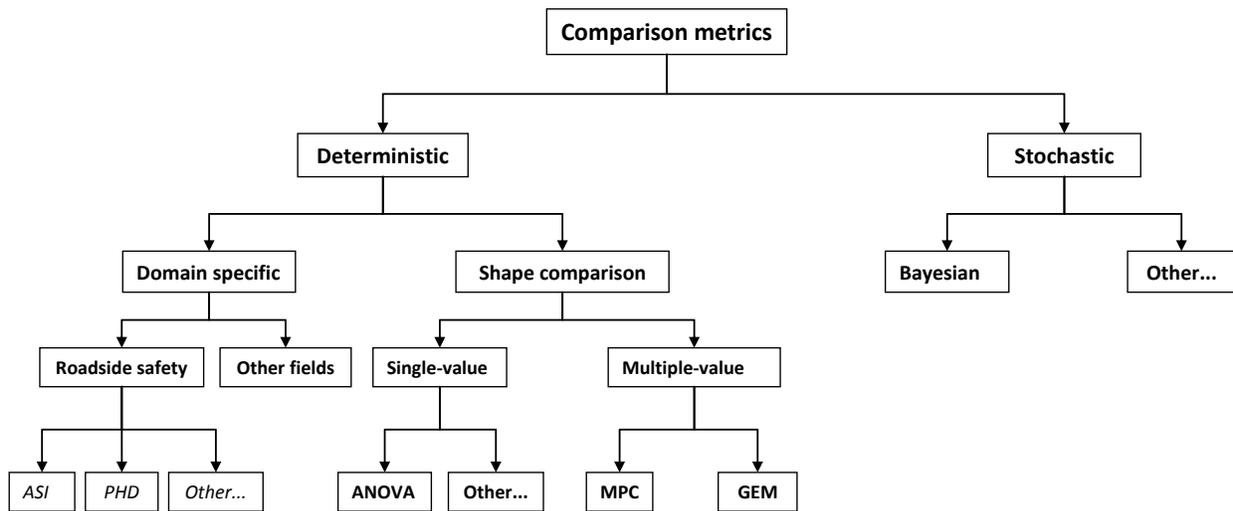
A particular type of metrics is the ANOVA group. These metrics provide useful information about the statistical distribution of the residuals. It is worth noting that, although these metrics are based on simple statistical measures, the values of the two curves which are compared are deterministic.

Although it is difficult to strictly classify shape-comparison metrics taking into consideration simultaneously all of the above mentioned characteristics, however, it is possible to define a general logical classification according to the most important attributes. The diagram in Figure 2.8 proposes a logical classification of the various comparison metrics found in the literature. Both the deterministic and stochastic groups can be classified into subcategories. As can be seen, the group of deterministic metrics can be classified into two subgroups: (i) domain specific and (ii) shape comparison metrics. In a similar manner, also the group of the stochastic metrics is composed of two principal subgroups: (i) metrics based on Bayesian statistics and (ii) metrics based on other statistical methods.

Typically, in roadside safety tests various domain specific metrics are calculated and the comparison of the values obtained from the experiment and the simulation may be used to assess the level of agreement. Shape comparison metrics can be further classified into single and multiple value.

The advantage of multiple-value metrics, MPC-type and GEM, is that they may provide additional information about the nature of disagreement between the curves (Table 2.6). Although the GEM metric can potentially investigate various characteristics of the two compared

curves, it has a strong subjective nature due to the intensive user judgment during the evaluation of the metric.



**Figure 2.8: Logical classification of comparison metrics.**

Unlike MPC metrics which first consider separately the phase and magnitude difference and then assess the general agreement, single-value metrics give immediately a distinct numerical value that represents the level of overall agreement between the two curves (Table 2.5). In the literature several single-value comparison metrics can be found differing for formulation and properties. Two of them are based on integral comparisons while the others are based on a point-to-point comparison (i.e., they are calculated based on the residuals between the two curves at each time step). A particular type of single-metrics is the ANOVA group, which may be useful to investigate if residuals are caused by any systematic problem.

**Table 2.5. Definition of single-value metrics**

<i>Integral comparison metrics</i>			
<i>NARD metrics</i>	<i>Refer to section 2.2.1.2.1</i>		
<i>Correlation Coefficient</i>	$\frac{n \sum c_i m_i - \sum c_i \sum m_i}{\sqrt{n \sum c_i^2 - (\sum c_i)^2} \sqrt{n \sum m_i^2 - (\sum m_i)^2}}$		
<i>Correlation Coefficient (NARD)</i>	$\frac{\sum c_i m_i}{\sqrt{\sum c_i^2} \sqrt{\sum m_i^2}}$		
<i>Weighted Integrated Factor</i>	$\sqrt{\frac{\sum \max(m_i^2, c_i^2) \cdot \left(1 - \frac{\max(0, m_i \cdot c_i)}{\max(m_i^2, c_i^2)}\right)^2}{\sum \max(m_i^2, c_i^2)}}$		
<i>Point-to-point comparison metrics</i>			
<i>Ziliacus error</i>	$\frac{\sum  c_i - m_i }{\sum  m_i }$	<i>Whang's inequality</i>	$\frac{\sum  c_i - m_i }{\sum  c_i  + \sum  m_i }$
<i>RSS error</i>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum m_i^2}}$	<i>Regression coefficient</i>	$\sqrt{1 - \frac{(n-1) \sum (c_i - m_i)^2}{n \sum (m_i - \bar{m})^2}}$
<i>Theil's inequality</i>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum c_i^2} + \sqrt{\sum m_i^2}}$		
<i>ANOVA metrics</i>			
<i>Mean</i>	<i>Standard deviation</i>		<i>t-test</i>
$\bar{e}^r = \frac{\sum_{i=1}^n (m_i - c_i) / m_{max}}{n}$	$\sigma^r = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (m_i - c_i - \bar{e}^r)^2}$		$T = \frac{\bar{e}}{\sigma / \sqrt{n}}$

MPC metrics treat the curve magnitude and phase separately using two different component metrics (i.e., M and P respectively) which are eventually combined together into a single comprehensive metric, C. The phase component (P) should be insensitive to the magnitude differences but sensitive to differences in phasing or timing between the two curves. Similarly, the magnitude component (M) should be sensitive to differences in magnitude but relatively insensitive to differences in phase. For each component of the MPC metrics, zero indicates that the two curves are identical. The following five MPC metrics were found in the literature: (a) Geers, (b) Geers CSA, (c) Sprague-Geers, (d) Russell and (e) Knowles-Gear [51,60,61,52,62]. Each of the MPC metrics differs slightly in its mathematical formulation. The different variations of the MPC metrics are primarily distinguished in the way the phase metric is computed, how it is scaled with respect to the magnitude metrics and how it deals with synchronizing the phase. In particular, the Sprague-Geers metric uses the same phase

component as the Russell metric. Generally, the magnitude component  $M$  of the MPC metrics is sensible to the order the two curves are input (i.e., the  $M$  component is not symmetric). The only symmetric magnitude component found in the literature is that of the Russell metric which is also peculiar as it is based on a base-10 logarithm. The Knowles-Gear metric is the most recent variation of MPC-type metrics. Unlike the previously discussed MPC metrics which are based on the evaluation of integrals involving each of the whole curves, the Knowles-Gear metric is based on a point-to-point comparison. In fact, this metric requires that the two compared curves are first synchronized in time based on the so called Time of Arrival (TOA), which represents the time at which a curve reaches a certain percentage of the peak value. Typically, in the literature the percentage of the peak value used to evaluate the TOA is 5%. Once the curves have been synchronized using the TOA, it is possible to evaluate the magnitude metric. Also, in order to avoid creating a gap between time histories characterized by a large magnitude and those characterized by a smaller one, the magnitude component  $M$  in the Knowles-Gear metric is normalized using the normalization factor  $QS$ .

**Table 2.6. Formulation of Multiple-value metrics**

	<i>Magnitude</i>	<i>Phase</i>	<i>Comprehensive</i>
<b>Integral comparison metrics</b>			
<b>Geers</b>	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2} - 1}$	$P_G = 1 - \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_G^2 + P_G^2}$
<b>Geers CSA</b>	$M_{CSA} = \sqrt{\frac{\sum c_i^2}{\sum m_i^2} - 1}$	$P_{CSA} = 1 - \frac{ \sum c_i m_i }{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\text{sign}(\sum c_i m_i) \sqrt{M_{CSA}^2 + P_{CSA}^2}$
<b>Sprague Geers</b>	$M_{SG} = \sqrt{\frac{\sum c_i^2}{\sum m_i^2} - 1}$	$P_{SG} = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_{SG}^2 + P_{SG}^2}$
<b>Russell</b>	$M_R = \text{sign}(m) \cdot \text{Log}_{10}(1 +  m )$ where $\frac{(\sum c_i^2 - \sum m_i^2)}{\sum c_i^2 \sum m_i^2}$	$P_R = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{\frac{\pi}{4} (M_R^2 + P_R^2)}$
<b>Point-to-point comparison metrics</b>			
<b>Knowles Gear</b>	$M_{KG} = \frac{\sum \left( \frac{ m_i }{m_{max}} \right)^p (\tilde{c}_i - m_i)^2}{\sum \left( \frac{ m_i }{m_{max}} \right)^p (m_i)^2}$ where, $\tilde{c} = c(t - \tau)$ $\tilde{c} = c(t - \tau)p$ is a parameter (generally $p=1$ ) $\tilde{c} = c(t - \tau)$ (with $\tau = TOA_c - TOA_m$ )	$P_{KG} = \frac{ TOA_c - TOA_m }{TOA_m}$	$\sqrt{\frac{10M_{KG}^2 + 2P_{KG}^2}{12}}$
<b>GEM</b>	Refer to section 2.2.1.2.2		

### 2.4.2 Computer codes

This literature review included also a description of the softwares cited in the literature which have been implemented to compute some of the described comparison metrics. In this section, a brief synthesis of the characteristics of these softwares is presented.

One of the programs (TRAP) is implemented for domain-specific metrics in roadside safety and, consequently, its application is limited to particular cases only. Moreover, the domain-specific metrics computed by TRAP usually involve one or, at most, a limited number of channels, thus making difficult to compare simultaneously all the available channels.

As for the computer programs capable of computing shape-comparison metrics, except one (Curve Analyzer), the other two programs cited in the literature (TNO software and ModEval) are able to compare simultaneously multiple pairs of curves. Unfortunately, both the two programs which have the capability to handle multiple channels are based on weighting factors

arbitrarily defined by the user which inevitably introduce a strong subjectivity in the computation of the final value of the metric.

None of the softwares described in the literature computes any of the MPC metrics, which could be useful to analyze separately the effect of either the magnitude or the phase shift between the curves. In fact, although there is a code (Curve Analyzer) which can compute several ANOVA metrics which could be a useful complement to any of the MPC or other metrics, it is limited only to this kind of metrics without the possibility to compute also the MPC metrics.

Moreover, only one of the softwares found in the literature is able to organize in a hierarchical manner various component validation tests (i.e., grouping metric values from similar tests) in order to allow for an easier location of the potential weakness of a numerical model.

# Chapter 3

## Methodology

The objective of this research is to implement into a software a qualitative method to assess the degree of match between one or multiple pairs of curves. The principal application of the implemented computer program is the comparison of curves for the verification and validation of numerical models. Although roadside safety is the main application for the software developed in this research, it can be applied to virtually any pair/s of curves.

The software has to be able to automatically assess how well two curves compare to each other by giving an objective score. Moreover, acceptance criteria for the quantitative metrics computed by the developed software are proposed, based on the typical scatter of experimental results in roadside safety. The software will be applied to compare experimental and numerical results in various typical roadside safety scenarios.

This research consists of the following six main tasks:

- Task 1     Screening of comparison metrics from the literature
- Task 2     Implementation of a software for computing comparison metrics
- Task 3     Code verification
- Task 4     Acceptance criteria and selection of metrics for V&V in roadside safety
- Task 5     Application of the implemented method to three cases
- Task 6     Conclusions and further research

The details for each task are presented in the next sections.

### **3.1 Initial screening of comparison metrics suitable for V&V of numerical models**

A selection is made from all the comparison metrics described in the literature in order to identify which are suitable to be successfully implemented as validation metrics for numerical simulations. The metrics selected in this task are used into the next tasks of the research.

## **3.2 Implementation of a software for calculating comparison metrics**

A computer code which can automatically evaluate the metrics previously selected is developed. In order to ensure a correct comparison of the two curves, the program implements various options which allow the user to perform different preprocessing operations before the metrics are calculated. Generally speaking, the software is designed to perform a comparison of virtually any pair of curves; the comparison metrics calculated by the software can be used for different purposes:

- validate computer simulation models using data obtained from experimental tests
- verify a simulation with another simulation
- assess the repeatability of an experimental test

The program is characterized by intuitive and interactive graphical interfaces which allow the user to input the two curves to be compared and select all the possible preprocessing options. Moreover, a series of automatic warnings are implemented to alert the user about possible mistakes during the preprocessing phase.

The program automatically performs some basic preprocessing operations necessary to correctly compare the input pairs of curves. In addition, the following three optional preprocessing operations have been implemented:

- Filtering,
- Synchronization
- Shift / Drift control

Each of these optional preprocessing controls can be selected independently from the others.

## **3.3 Code verification**

Both the implementation of the general program structure and the specific comparison metrics are verified in order to assure the reliability of the computer program. Particular attention is paid into the verification of the parts of the code which handle the following operations:

- Multichannel evaluation (preprocessing and metrics computation)

- Output of results (screen and file)
- Metrics computation

### **3.4 Acceptance criteria and suggested metrics for V&V in roadside safety**

All the metrics implemented into the computer program are used to compare curves from repeated executions of the same full-scale crash test scenario. Assuming that the tests were repeatable, acceptance criteria are assessed for each metric which appropriately consider the typical scatter of results in roadside safety full-scale tests. Before computing the metrics, curves are appropriately preprocessed and a reference baseline curve is selected for the comparisons.

Moreover, based on the values obtained from the comparison of these curves, some suggestions are made for the metrics which are considered to be the most appropriate in the roadside safety field.

### **3.5 Application of the implemented method to three cases**

In order to assess the robustness of the implemented method, the quantitative assessment of the degree of match for some velocity time histories and the corresponding subjective judgment of eleven experts in the field are compared.

Eventually, the computer code and the acceptance criteria are applied to perform the validation of two different finite element models, respectively in roadside safety and biomechanics.

The results obtained from these two cases are used to assess the applicability of the software to the validation of numerical models.

### **3.6 Conclusions**

A discussion of the results obtained applying the implemented computer program to the three cases analyzed in the previous task is given. Eventually, conclusions related to the applicability of the computer program to the verification and validation process of numerical models are drawn. Also some directions for further research of the work described in this dissertation are suggested.

# **Chapter 4**

## **Computer Program for Comparing Curves**

In this chapter the implementation of a Matlab code for the computation of most of the comparison metrics found in the literature is described.

The principal reason for the implementation of a computer program is that, although many comparison metrics have been proposed in the years, a relatively limited number of computer codes that allow for an easy calculation of any comparison metric have been developed. In fact, one of the biggest limitations in the use of metrics for the comparison of curves in general lies in the lack of automatic and easy-to-use computer programs; even though a practitioner is aware of one or more specific comparison metrics, often, the tediousness and the implementation problems related to a manual evaluation of the metrics themselves may eventually strongly discourage its use.

Although a few computer programs for the evaluation of the comparison metrics have been already implemented in the past, as mentioned at the end of the literature review in Chapter 2, they are characterized by a series of restrictions. In particular, the following two main limitations have been found:

- Computation of a limited number of comparison metrics
- Impossibility or inadequate capacity to objectively compare multiple channels

The computer program developed in this research aims to cover both these two limitations by allowing the user to automatically compute most of the comparison metrics found in the literature and implementing various methods for the concomitant comparison of multiple channels in a completely automatic and, hence, objective manner.

### **4.1 Program overview**

Although the code has been specifically developed to perform the verification and validation of roadside safety simulations and crash tests, it can be used to generally perform a comparison of

virtually any pair of curves. In general, the program implemented in this research computes comparison metrics which can be used to accomplish the following operations:

- Validate computer simulation models using data obtained from experimental tests
- Verify the results of a simulation with another simulation or analytical solution
- Assess the repeatability of a physical experiment by comparing different executions.

In order to ensure the most correct comparison of the curves as possible, various preprocessing options have been implemented and may be performed before the metrics are calculated. All the comparison metrics available in the program are deterministic and do not take into consideration the probabilistic variation of either the experiments or the calculations.

Either a single or multiple pairs of curves can be compared. In the latter case two main options are available: (i) metrics values are computed for each channel and eventually combined together using weighting factors automatically assessed by the program or (ii) the resultant curves are first computed and the corresponding metrics are then calculated.

It is worth noting that, after the preprocessing operations, the code performs the comparison of the couple/s of curves in a complete autonomous manner, hence avoiding the risk of introducing any potentially subjective judgment by the user at any step of the calculation of the metrics.

All the functionalities of the program can be managed through intuitive and interactive Graphical User Interfaces (GUI's) which allow the user to effortlessly input the curves to be compared and easily perform any of the existing preprocessing operations. Also, a series of automatic warnings alert the user about possible mistakes during the preprocessing phase.

The program is completely written in Matlab<sup>®</sup> (version R2009 a) [63]. The choice to use this language is motivated by the following reasons:

- advanced matrix calculation capabilities
  - quick implementation of algorithms and GUI's into the code
  - real-time check of the implemented code (either in its entirety or just for some portions)
- The source code can be either executed directly from the Matlab<sup>®</sup> environment or compiled as an executable application.

Note that, in the case the code is compiled as an executable application it is necessary to have the Matlab<sup>®</sup> Compiler Runtime (MCR) component installed on the machine on which it is desired to run the executable.

#### **4.1.1 Organization of the chapter**

This chapter is divided into two main parts. Section 4.2 gives a description of the main functionalities which are either necessary or useful to be implemented in a program designed for comparing couple/s of curves, while section 4.3 provides some details about the algorithms implemented in the program code.

Also a detailed description of the program and its implementation may be found respectively in the user's and programmer's manuals located respectively in Appendix A and Appendix B.

### **4.2 Description of the tasks**

This section provides a general description of the tasks performed by the program.

In order to create a code for the comparison of couples of curves, the following three main operations are deemed necessary:

- Input and preprocessing
- Computation of metrics
- Output of results

In the following, a detailed description of each of these operations is given.

#### **4.2.1 Input and preprocessing**

##### ***4.2.1.1 Input of data***

The input of the curves which are to be compared is the first task that has to be accomplished by the program. As typically both experimental and numerical curves are not characterized by an analytical formulation, they can be defined only using a discrete series of data points which are usually tabulated as pairs of columns (e.g., the first column for the time values and the second column for the corresponding measured values). Generally, the most common electronic format to store the data points is as plain text (i.e., ASCII standard).

#### **4.2.1.2 Preprocessing**

Usually, the originally input curves are not directly comparable to each other because of one or more of the following reasons: (i) different sampling rates, (ii) different lengths, (iii) shift along either the abscissa or the ordinate directions and (iv) occurrence of high frequency noise. Hence, in order to properly compare the curves, in most cases it is necessary to prepare them through a careful preprocessing procedure.

The following preprocessing options have been implemented into the code:

- Scaling
- Manual trimming and/or Shift of the curve origin
- Filtering
- Shift/Drift
- Re-sampling and Trimming
- Synchronization

Some of these preprocessing operations are necessary and automatically executed, while others are optional and the user can decide whether to accomplish them or not based on the specific case. In particular, the re-sampling and trimming operations have to be always performed by default as they are necessary to correctly compare any generic couple/s of curves. In fact, both curves must have a point-to-point match in order to be possible to calculate most of the comparison metrics.

In the next, a brief description of each preprocessing operation and the theory and/or the method implemented is given. For a description of the algorithms used to implement each preprocessing feature, see section 4.3 and Appendix B.

##### **4.2.1.2.1 Curve scaling**

In certain occasions, it may be necessary to scale the original input curves by an arbitrary user defined factor. For example, this operation may be useful when the original curves have been collected using different units (e.g., accelerations can be converted from  $m/s^2$  to  $g$ 's) or to invert the sign of any of the time histories. In any of these cases, the scaling of the original

curves can be easily performed multiplying the vector containing the data points of the desired curves by the user-defined scale factors.

#### **4.2.1.2.2 Manual trimming and curve shift**

The manual trimming of the original curves represents another typical operation which often needs to be performed when working with both experimental and numerical data. In fact, the option to manually trim the input curves before applying any pre-processing operation allows the user to trim the initial ‘flat segment’ which typically characterizes the time histories of both experiments and simulations before the desired event of interest occurs (i.e., the impact in the case of roadside safety tests). Also, in case the simulation and experimental curves start diverging at a certain time, this option may be used to allow for a better synchronization of the curves by trimming them immediately before the diverging point.

As previously mentioned, the data points of each curve are stored in column vectors. Hence, after the user has defined the boundary values for either one or both the two curves, it is necessary to determine the indexes corresponding to (or approximating, in case the sampling rate does not allow the exact fit) these values. The vectors containing the data points for each curve are then trimmed using the index values previously found. Also, after being trimmed, the time vectors have to be shifted in order to ensure that the beginning of the abscissa vector is coincident with the origin. This shift of the trimmed curves along the abscissa axis can be easily achieved by subtracting from the time vectors the corresponding initial values. Note that, in this manner, the shifting of the curves works for initial values that are either positive or negative.

#### **4.2.1.2.3 Filtering**

Usually both numerical and experimental raw curves are characterized by a considerable level of noise, which may be respectively created by the transducer and then increased by the amplifier placed in the measurement chain in the case of experimental tests or, introduced by some sort of numerical noise, in the case of numerical simulations. In any case, both of these circumstances cause undesired high-frequency noise.

Filtering is a very common operation in the preprocessing of time history data as it represents a very useful option for limiting the noise embedded in the original input curves. In order to obtain a value of the comparison metrics that is as much reliable as possible, it is very important

to remove the noise from both the couple of curves. While noise derives from different sources in either physical experiments or numerical simulations, both the experimental and numerical curves should be filtered using the same filter option to ensure that differences in the metric evaluation are not based on the difference in frequency content in the two signals.

The international standard for filtering curves related to impact events is specified by the SAE J211/1 normative [64] which is the reference in matter of filtering for the NCHRP Report 350 [21] and EN1317 [33]. The SAE J211 filtering operation is achieved through a digital filter which, in particular, is implemented as a four-pole Butterworth low-pass filter using an algorithm that is based on a double-pass filtering option (i.e., forward/backward). In practice, data are filtered twice, once forward and once backward using the following difference equation in the time domain:

$$Y(t) = a_0 \cdot X(t) + a_1 \cdot X(t-1) + a_2 \cdot X(t-2) + b_1 \cdot Y(t-1) + b_2 \cdot Y(t-2) \quad (4.1)$$

Where:  $X(t)$  is the input data sequence and  $Y(t)$  is the filtered output sequence.

The filter coefficients vary with the CFC value and are calculated using the following formulas:

$$a_0 = \frac{\omega_i^2}{1 + \sqrt{2}\omega_i + \omega_i^2} \quad (4.2)$$

$$a_1 = 2a_0 \quad (4.3)$$

$$a_2 = a_0 \quad (4.4)$$

$$b = \frac{-2(\omega_a^2 - 1)}{1 + \sqrt{2}\omega_a + \omega_a^2} \quad (4.5)$$

$$b_2 = \frac{-1 + \sqrt{2}\omega_i - \omega_i^2}{1 + \sqrt{2}\omega_i + \omega_i^2} \quad (4.6)$$

where,

$$\omega_i = 2\pi \cdot CFC \cdot 2.0775 \quad (4.7)$$

$$\omega_a = \frac{\sin(\omega_d \cdot T/2)}{\cos(\omega_d \cdot T/2)} \quad (4.8)$$

These filter coefficients presume that the curves are time histories with time values expressed in units of seconds. The following values for the Channel Frequency Class (CFC) are suggested: 60, 180, 600 and 1000. Table 4.1 shows the specifications of each particular CFC value as defined by SAE J211/1.

**Table 4.1. Specifications for the most usual CFC values of SAE J211 compliant filters [64]**

<b>CFC value</b>	<b>3 dB limit frequency [Hz]</b>	<b>Stop damping [dB]</b>
60	100	-30
180	300	-30
600	1000	-40
1000	1650	-40

Note that, while it is not recommended, if the user wanted to use CFC values different from those suggested by the standard SAE J211, also user defined CFC parameters could be specified.

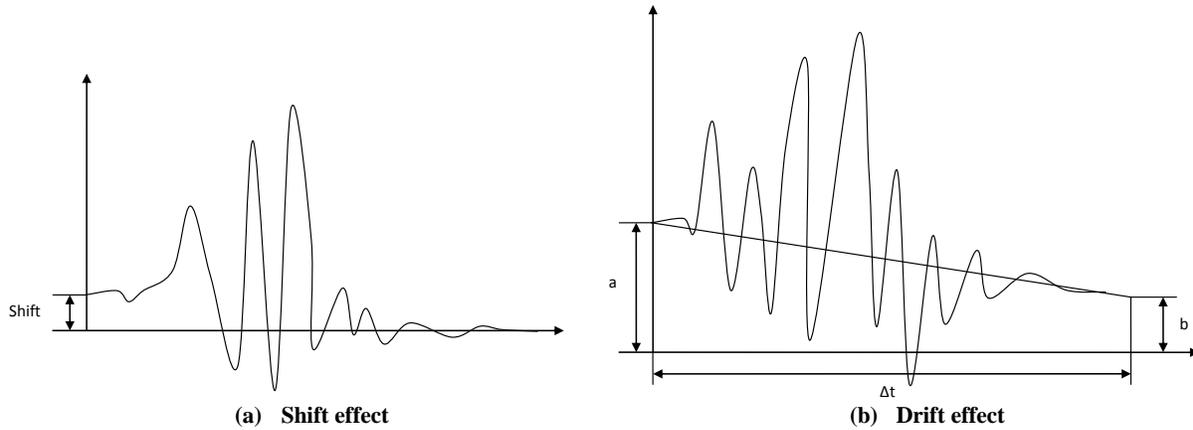
#### **4.2.1.2.4 Shift and/or drift correction**

Another preprocessing option which may result very useful when handling experimental data is the possibility to correct any initial vertical shift and/or drift in the curves which may be caused by the heating of sensors during an experiment. In fact, experimental data sometimes contain shift and/or drift effects which are due to the change of temperature respectively before or during the test.

In particular, the shift effect is a homogeneous vertical translation of the entire experimental curve due to the change of the sensor temperature in the period of time between the measurement system has been zeroed and the data are started to be collected at the beginning of the test (Figure 4.1 a). Instead, the drift effect is a linearly increasing translation of the experimental curve due to a change in the temperature of the sensor between the beginning and the end of the test (Figure 4.1 b).

The implementation of procedures to control the shift and drift effects may result very useful in case one or both the two input curves have been recorded from experimental tests and present either or both these data acquisition problems. In fact, as either the initial shift or drift of the curves are caused by an incorrect acquisition of the experimental data, these pre-processing options may play an important role for an accurate evaluation of the comparison metrics. In generally, curves resulting from numerical analyses should not need to use these options since

shift and drift are side effects which purely characterize physical sensors and do not appear in numerical results.



**Figure 4.1: Representation of (a) Shift and (b) drift effects.**

The drifted curve is basically shifted by a value which linearly increases/decreases in the time according to the following simple formula:

$$\Delta = m \cdot t \quad (4.9)$$

where  $m$  is the slope of the linear drift and can be easily computed as:

$$m = (b - a) / \Delta t \quad (4.10)$$

with  $a$  and  $b$  representing respectively the initial and final values of the linear drift function and  $\Delta t$  its total length as shown in Figure 4.1 b.

The correction of the shift effect can be easily achieved by translating the whole curve by the shift value  $a$ . As for the drift effect, once the value of the slope  $m$  has been calculated from Equation (4.10), it is possible to correct it by translating each point of the curve by the opposite of the corresponding values obtained from the drift baseline function (Equation (4.9)). Note that, as the drift baseline is not a constant function, the correction value is different for each sample point of the original curve.

In order to calculate the shift or drift values by which the input curves have to be translated, it is necessary to consider the mean values at the beginning and the end of each curve. In particular, it is important to ensure that the values at the beginning and the end of the vector containing the data points (i.e., head and tail) are sufficiently constant to guarantee that the mean values of these

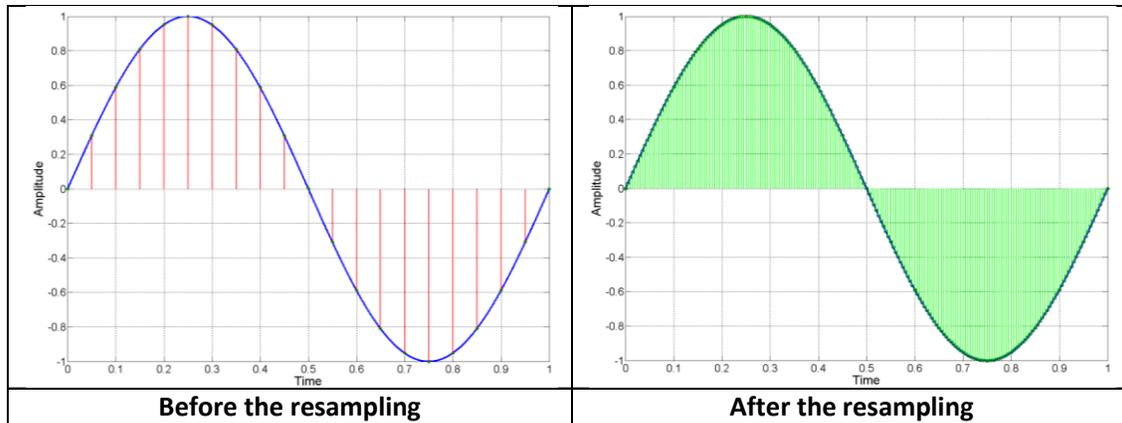
sub-vectors effectively represent the shift/drift values. An initial guess for the point until the head of the curve can be considered as constant (or from which the tail of the curve is considered constant) is the time at which the curve reaches five percent of the peak value (or from which the curve is less than five percent of the curve's peak). In order to check that the initially guessed head or tail sub-vectors are constant, their standard deviation must be under a critical value, which is defined to be 0.1. In case the initial sub-vectors do not satisfy the above mentioned criterion, the algorithm iteratively decreases the initial length of the head and tail sub-vectors till the standard deviation of the new reduced sub-vectors is less than the proposed critical value.

#### **4.2.1.2.5 Resampling and automatic trimming**

As most of the metrics which quantify the agreement between the test and simulation curves are based on the evaluation of the point-to-point error, in order to correctly calculate the residuals, it is mandatory that the two curves have the same sampling rate. The original curves, however, may have been sampled at different frequencies so it could be necessary to re-sample them in order to compute a point-to-point difference (Figure 4.2).

In case the two input curves which have to be compared have not been collected at the same sampling rate, it may result very convenient to resample the curve which has the lower sampling rate (i.e., the bigger difference in time between two contiguous data points) at the higher rate of the other curve. An effective and yet simple manner to achieve the resampling is represented by the use a linear interpolation of the original data. In fact, once the interpolating polynomial has been defined, it would be possible to move on and add the extra data points by interpolating between the originally collected points.

During the resampling operation, it is also possible to trim the curves in order to have them at the same length (i.e., trim their initial and final abscissa points respectively to the same values).



**Figure 4.2: Curves with different sampling rates before (left) and after (right) the resampling.**

#### 4.2.1.2.6 Curve synchronization

The least but certainly not the last preprocessing operation which may be necessary to perform before the evaluation of the metrics is the synchronization of the two curves. Indeed, this option can be very useful if the original curves have not been acquired starting at exactly the same instant (e.g., they represent respectively a numerical simulation and an experimental test of the same crash test but the instant at which data collection is started does not match).

The synchronization of the two input curves is very important as any initial shift in the acquisition between the two compared curves could seriously affect the final value of the comparison metrics. In fact, as the comparison metrics are mostly based on the evaluation of the residual error, it is necessary to delete or at least reduce as much as possible any time shift between the two curves which are compared.

For example, although identical, two input curves with an initial phase difference due to a different starting point in the acquisition process would probably lead to poor results of most of the comparison metrics. In this case, the curves should be shifted back or forth so that the impact time in both of them is matched and only after that they could be compared.

Two different synchronization options are considered in this research. Both options are based on the minimization of a target function which is respectively: (1) the absolute area between the two curves (i.e., the area of the residuals) and (2) the squared error between the two curves. In order to implement these two methods of synchronization, it is first necessary to shift by an arbitrary value the two curves respect to each other. The code can then assess the shift value which

minimizes either the absolute area of residuals (method 1) or the sum of squared residuals (method 2).

Although these two methods are similar, they may provide slightly different results. In any case, selecting one of these two methods should result in the most probable pairing point for the two curves.

#### **4.2.2 Metrics calculation**

Once the curves have been properly preprocessed, it is then possible to compute the comparison metrics. Fourteen different deterministic shape-comparison metrics have been implemented in the program created during this research. The metrics implemented are listed in Table 4.2. Each of them is characterized by a different formulation and, as indicated in the summary of the literature review in Chapter 2, conceptually they can be classified into two main categories: (i) multiple-value and (ii) single-value metrics. In particular, the multiple value metrics chosen to be implemented in the code are the so called MPC (Magnitude-Phase-Composite) metrics.

Details about the mathematical formulation can be found in Table 4.3 and Table 4.4 for respectively the single-value and multiple-value metrics. In both the cited tables the measured and computed data points are indicated respectively as  $m_i$  and  $c_i$ . Note that the formulation for each of the metrics assumes that both the two curves have a common and constant sampling period.

For more details about the characteristics of these metrics, refer to Chapter 2 and to the cited literature.

**Table 4.2. Metrics implemented in the computer program**

<b>Multiple-value metrics</b>	<b>Single-value metrics</b>	
Geers	Correlation Coefficient	Whang's inequality
Geers CSA	Correlation Coefficient (NARD)	RMS error
Sprague&Geers	Theil's inequality	Regression coefficient
Russell	Weighted Integrated Factor	Average residual error
Knowles&Gear	Zilliacus error	Standard deviation of the residuals

**Table 4.3. Mathematical formulation of the single-value metrics implemented in the computer program<sup>§</sup>**

<b>Correlation Coefficient</b>	$\frac{n \sum c_i m_i - \sum c_i \sum m_i}{\sqrt{n \sum c_i^2 - (\sum c_i)^2} \sqrt{n \sum m_i^2 - (\sum m_i)^2}}$		
<b>Correlation Coefficient (NARD)</b>	$\frac{\sum c_i m_i}{\sqrt{\sum c_i^2} \sqrt{\sum m_i^2}}$	<b>Theil's inequality</b>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum c_i^2} + \sqrt{\sum m_i^2}}$
<b>Weighted Integrated Factor</b>	$\sqrt{\frac{\sum \max(m_i^2, c_i^2) \cdot \left(1 - \frac{\max(0, m_i \cdot c_i)}{\max(m_i^2, c_i^2)}\right)^2}{\sum \max(m_i^2, c_i^2)}}$		
<b>Zilliacus error</b>	$\frac{\sum  c_i - m_i }{\sum  m_i }$	<b>Whang's inequality</b>	$\frac{\sum  c_i - m_i }{\sum  c_i  + \sum  m_i }$
<b>RMS error</b>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum m_i^2}}$	<b>Regression coefficient</b>	$\sqrt{1 - \frac{(n-1) \sum (c_i - \bar{m})^2}{n \sum (m_i - \bar{m})^2}}$
<b>ANOVA metrics</b>			
<b>Average residual error*</b>		<b>Standard deviation of the residuals*</b>	
$\bar{e}^r = \frac{\sum_{i=1}^n (m_i - c_i) / m_{max}}{n}$		$\sigma^r = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (e^r - \bar{e}^r)^2}$	
(*) normalized to the peak of the measured values			

<sup>§</sup>For all the metrics it is assumed that both the two curves have the same constant sampling rate.

**Table 4.4. Mathematical formulation of the multiple-value metrics implemented in the computer program<sup>§</sup>**

	<i>Magnitude</i>	<i>Phase</i>	<i>Comprehensive</i>
<b>Geers</b>	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2} - 1}$	$P_G = 1 - \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_G^2 + P_G^2}$
<b>Geers CSA</b>	$M_{CSA} = \sqrt{\frac{\sum c_i^2}{\sum m_i^2} - 1}$	$P_{CSA} = 1 - \frac{ \sum c_i m_i }{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\text{sign}(\sum c_i m_i) \sqrt{M_{CSA}^2 + P_{CSA}^2}$
<b>Sprague&amp;Geers</b>	$M_{SG} = \sqrt{\frac{\sum c_i^2}{\sum m_i^2} - 1}$	$P_{SG} = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_{SG}^2 + P_{SG}^2}$
<b>Russell</b>	$M_R = \text{sign}(m) \cdot \text{Log}_{10}(1 +  m )$ where $m = \frac{(\sum c_i^2 - \sum m_i^2)}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$P_R = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{\frac{\pi}{4} (M_R^2 + P_R^2)}$
<b>Knowles&amp;Gear</b>	$M_{KG} = \sqrt{\frac{\sum \left(\frac{ m_i }{m_{max}}\right) (\tilde{c}_i - m_i)^2}{\sum \left(\frac{ m_i }{m_{max}}\right) (m_i)^2}}$ where $\tilde{c} = c(t - \tau)\tilde{c} = c(t - \tau)\tilde{c} = c(t - \tau)$ (with $\tau = TOA_c - TOA_m$ )	$P_{KG} = \frac{ TOA_c - TOA_m }{TOA_m}$	$\sqrt{\frac{10M_{KG}^2 + 2P_{KG}^2}{12}}$

<sup>§</sup>For all the metrics it is assumed that both the two curves have the same constant sampling rate.

#### 4.2.2.1 Metrics history and comparison on user-defined sub-intervals

Note that, sometimes, in order to analyze any potential problem occurring when comparing the curves on their whole domain, it may be also necessary to compare them on smaller intervals to diagnose the cause of a bad match (i.e., localize the segments on which the curves show an unsatisfactory agreement in order to allow for a better investigation of the causes of the problem). For this reason, although by definition they provide a scalar value (i.e., a simple number), the above listed metrics are implemented in such a manner that they are evaluated on various time intervals characterized by an increasing size, hence giving as output a sort of time history of the metrics. The smaller time interval on which the input curves are compared is set to be ten percent of the total time and, at each step, it is increased in size till it becomes the total interval. In this way, it is possible to reconstruct a time history of the metrics indicating how the values evolve comparing the curves on increasing portions of their total time interval.

Moreover, sometimes it is necessary to focus the attention of the comparison on an interval smaller than the initial whole domain in which the original curves are defined. For this reason

the option to restrict the computation of the metrics on a sub-interval may result to be very useful. The capability to define a user-defined time interval out of the original domain in which the couple/s of curves are defined, is implemented as well in the code of the computer program.

#### **4.2.2.2 Multiple channels**

In cases when multiple channels are collected, it may be necessary to compare the various channels concurrently in order to obtain an overall assessment of the degree of match. In fact, although theoretically the couples of curves for all the channels are supposed to be in good agreement, in practice there may be some channels which are less relevant than others as they do not represent significant measurements for that specific type of test. In this case, for these channels, a good agreement might be not as important as for the principal channels or vice versa, a potential disagreement would not seriously affect the overall verification and validation process for the numerical model. As an example, in case of a test involving an impact against a barrier which redirects the vehicle by keeping its asset horizontal during all the crash event, the acceleration collected along the vehicle vertical axis and also the vehicle roll and pitch motions may be both considered irrelevant channels respect to the other two acceleration components measured in the vehicle plane and the yaw angle.

Typically in roadside safety multiple channels are simultaneously collected during a full-scale crash test (e.g., three accelerations and three rotational rates) and the verification and validation of numerical models aiming to reproduce these types of tests should take account of the comparison of numerical and experimental curves for each of the six channels. For the reasons previously mentioned, the comparison of the various channels each considered singularly may not provide an effective assessment of the level of verification and validation of a numerical model as much as the concurrent comparison of all the available channels as a group. Hence, in order to allow a comprehensive comparison of the model, two different methods to compare contemporaneously multiple channels are investigated, based respectively on:

- Weighted average of the metrics from each channel
- Resultant of the accelerations and rotational rates.

In the former method (weighted average), the comparison metrics calculated using the pairs of curves from each channel are combined together using weighting factors; while, in the other

method (resultant), the comparison metrics are computed based on the resultants of respectively the acceleration and rotational rate channels instead of considering their components singularly. In either way, the metric values obtained using either of the two methods are intended to provide a general assessment of the comparison using all the available channels simultaneously.

In particular, 3 different methods are examined for the assessment of the weighting factors used to compute the weighted average of the metrics values. Each of these methods proposed to compute the weighting factors is based on a peculiar approach, based respectively on:

- Peak values of the curves
- Area of the curves
- Momentum of the vehicle

Following a description of the general characteristics for each of the proposed methods which are investigated is given. Also, the advantages/disadvantages of the various methods are discussed.

#### **4.2.2.2.1 Peak values of the curves**

The first method considered for the assessment of the weighting factors is based on the peak value of the baseline curve for each channel. In particular, in order to be able to compare them, the peaks obtained from the three acceleration and the three rotational rate channels, are normalized by the sum of the values from their respective group. These normalized values are then used to compute the final weighting factors.

#### **4.2.2.2.2 Area of the curves**

This second method to compute the weighting factors is based on a similar methodology as the previous, but it uses the area of the baseline curve from each channel instead of the peak value. Moreover, a variation of the former method based on the area is also proposed which implicitly assigns to each of the two groups of channels (i.e., linear accelerations and rotational rates) half of the total weight.

#### **4.2.2.2.3 Momentum of the vehicle**

A third different method to compute the weighting factors which has been considered is based on the linear and rotational momentum of the vehicle. In order to calculate these quantities, it is

necessary to know both the linear and rotational velocities and also the vehicle's inertial properties (i.e., respectively the value of the vehicle mass and principal component of the moment of inertia). In particular, it is chosen to use as the values of the linear velocities the area of the respective acceleration time histories till their peaks and as the values of the rotational velocities the peaks of the corresponding rotational velocities. The vehicle inertial properties have to be manually defined by user for each specific case. Note that, it is also necessary to convert the rotational momentum of the vehicle to an equivalent linear momentum. To perform this conversion it is decided to define an arbitrary value of the arm for the rotational momentum which is equal to the unity.

#### **4.2.2.2.4 Advantages and disadvantages of each weighting method**

This section presents a brief discussion of the advantages and disadvantages of the above described methods to compute the weighting factors which are investigated in the case of multiple channels.

The method based on the peak values of the curves is simple in its nature but, because it is based on the local characteristics of each channel it may be drastically influenced by the way the curves have been preprocessed. In particular, one of the aspects which may affect most the computed values of the weighting factors is the way curves are filtered, as this can distinctly influence the peak values. In order to make the computation of the weighting factors less sensitive to the filtering process, a similar method, but based on the area of the curves instead of the peaks, is investigated as well.

Both the two methods based respectively on the peak values and area of the curves are characterized by the implicit condition that the same weighting factor is assigned to the curves with the highest peak/area in each of the two groups (i.e., accelerations and rotational rates). This could give too much weight to one of the rotational rate channels respect to the other counterpart in the acceleration group or vice-versa (e.g., if there is no important rotation there may be an over estimation of the importance of the rotation channels).

As any of the first two methods did not imply any particular relationship between the acceleration channels and the rotation rate channels, it is decided to investigate an alternative

formulation of the second method based on the area of the curves, which implied the weighting factors be distributed in equal portion between the acceleration and rotational rate channels.

A potential problem with this alternative method could be an over assessment of the weighting factors in case only one channel is input for one of the two groups and two or more channels are added in the other group. In fact, in this particular case, the only channel of the first group would be assigned a weight equal to 0.5.

Moreover, another method which implied a relation between accelerations and velocities is investigated based on the inertial properties of the vehicle used in the actual crash test. The main limitations of this method are: (i) the difficulty to have access to the inertial properties of the vehicles used during the test and (ii) the fact that this particular method assumes a particular way of collecting data (e.g., using three accelerometers and three rate gyros) which may not always be the case.

The averaged values of the metrics based on the weighting factors computed using each of the various methods above described and also the values of the metrics based on the resultant method have been calculated in the application case which is described in Chapter 8.

### **4.2.3 Results output**

After the curves have been compared, the program has to provide a convenient and efficient way to output all the information created during the execution. As the results are in the form of numbers (i.e., the values of the various comparison metrics computed), the most convenient and organized way to store the information is through tables and graphs.

In particular, the tabulated format is essential to organize the values of the various metrics according to the following criteria:

- Time interval on which curves have been compared
- Values computed for each channels (in case of multiple channels)

Two principal types of output are implemented in the software: (i) on screen and (ii) on files. The display of the values on the screen is useful to provide the user with an immediate visualization of the results during the execution of the program and, in case, proceed to analyze potential problems by comparing the curves on one or more specific restricted intervals of

interest. On the other hand, the storage of results in files allows to easily access them after the program has been terminated.

The following information is displayed on the screen:

- Graphs of the preprocessed curves and their integrals
- Metrics values
- Graphs of the residual history, distribution and cumulative distribution

In particular, because of the characteristics of the results, it is chosen to save the output values in one or more spreadsheet files which are compatible with the largely diffused Microsoft Excel format. In fact, the use of a spreadsheet format allowed the possibility to both visually organizing the data in a tabular configuration and also dividing the tables in different sheets of the same Excel file if necessary to consider separately the values of the comparison metrics for each of the multiple channels.

Moreover, another feature that may result to be very useful for a program designed to compare curves as that developed in this research is the possibility to save the preprocessing options made during the execution of the program and the selection for the input curves in a configuration file. This would allow the user to have a convenient way to keep record of a specific comparison and, in particular, to allow any certifying agency to independently repeat all the calculations to confirm the authenticity of the results. The possibility to load the configuration may result very convenient also in the case it is desired to apply some minor modifications to a previous comparison.

### **4.3 Program implementation**

This section gives a description of the programming concepts behind the code of the software which has been developed in this research. As the main objective of this dissertation is to explain the various choices made in the implementation of the code, only a general description of the main algorithms is given. For more details about the algorithms, refer to the programmer's manual of the software which is located in Appendix B.

The code has a modular structure: the core of the program is contained into a principal script (*Main.m*) which is started at the execution of the program. On their turn, each of the parts of this

principal script invokes one or more specific secondary scripts to manage the various particular tasks. Both the principal and secondary scripts may recall either specific functions from various Matlab libraries (toolboxes) or user-defined functions programmed *ad hoc* to perform some specific operations. If necessary, a secondary script may also recall one or more subscripts.

As each of the scripts/functions is saved in a distinct file, such a modular organization of the code allows to easily locating the lines which implement a specific operation and makes the interpretation of the code more intuitive.

### **4.3.1 Programming structures used in the code**

The code has been implemented making an extensive use of GUI's, advanced data structures and loops. In the following, these three main characteristics used in the creation of the code are described.

#### **4.3.1.1 GUI's and data structures**

In order to facilitate the interaction between the user and the program, GUI's are implemented to handle the various features of the program. A graphical interface in Matlab is regarded as a function, which means it is possible to define both input and output variables. In particular, the possibility to input variables allows a GUI to load information about the configuration saved during the previous instance or a default configuration, in case the GUI is opened for the first time. In fact, some of the GUIs are implemented into iterative loops and, after being invoked for the first time, they may be opened again. In this case, the new instance of a GUI may be given as input a variable which contains all the information about the previous configuration. In order to organize all the configuration information in a single variable which is easy to pass through the GUI functions, the Matlab structure type is adopted. This particular type of Matlab variable is described in the next section together with the other data structures used in the implementation of the code.

#### **4.3.1.2 Data structures used in the code**

Three different types of data structure are used in the implementation of the code:

- Matrices and arrays (floating-point/integer, characters and strings)

- Structures
- Cell arrays

Each of them is applied based upon the specific characteristics of the data that are necessary to manipulate at each instance. This section summarizes the properties of each of the three data structures and indicates for what purpose each of them are considered.

Matrices and arrays are the most common and basic type of variable available in Matlab. They are used in the code to store information organized in a homogeneous manner (e.g., a matrix of data vectors which are all numbers or text characters and have the same length). These data structures can be used to store both numbers and text characters. Numbers can be either stored in form of floating-point or integer. Scalar numbers or single characters are a particular case in which the matrix is reduced to be a 1x1 array.

Although matrices and arrays are widely used in the code, they are not always appropriate to store the information. In particular, the main limitation of this type of data structure is represented by the impossibility to store heterogeneous information in the same variable, such as numerical vectors characterized by a different length or even text strings and numerical vectors. For various reasons, in some cases, it is necessary to store heterogeneous data in the same variable. For example, the need to use a single variable in which data are organized in an indexed manner is compulsory to allow retrieving this information or writing new data during the iterations of a loop cycling over the different input channels. Also, another application of the use of indexed heterogeneous information is in the loop for the definition of an arbitrary number of intervals on which to compare the various channels.

In other occasions, although the use of a single variable which contained multiple data is not strictly required, it represents a very convenient option to store the various pieces of information into separate homogeneous variables. A typical application in which the use of a heterogeneous single variable is preferred to the use of numerous homogeneous variables is when data need to be passed to and received back from function calls.

Two different types of data structures which allow storing heterogeneous information are used in the implementation of the code: (a) structures and (b) cell arrays.

A Matlab structure is a data type that provides the mean to store selected data together in a single entity. A structure consists mainly of data containers, called fields, and each of these fields stores an array of any available Matlab data type. The fields of a structure variable can be heterogeneous, thus allowing to store in the same variable different type of information. Figure 4.3 shows an example of structure variable, *s*, which has three fields: *a*, *b*, and *c*.

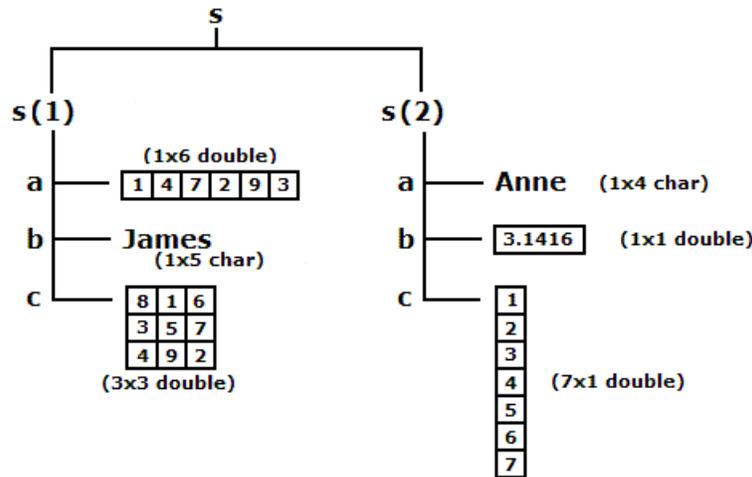


Figure 4.3: Example of a Matlab structure variable [63].

Another type of Matlab data structure frequently used in the implementation of the program is the cell array. A Matlab cell array is a collection of data containers, called cells, in each of which different types of data can be stored. As an example, Figure 4.4 shows a 2-by-3 cell array in which the cells in the first row one hold an array of unsigned integers, an array of strings, and an array of complex numbers, while the second row holds three other types of arrays, the last being a second cell array nested in the outer one.

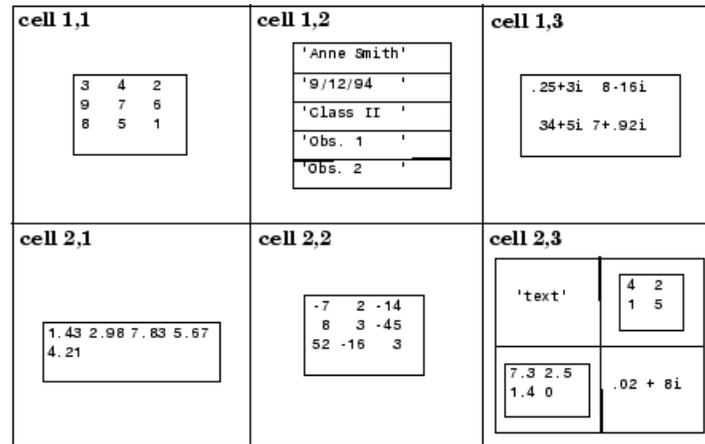


Figure 4.4: Example of a Matlab structure variable [63].

As a general rule, structure-type variables are used to store in an organized manner the several options selected by the user in the various GUI's, while cell-type variables are used to conveniently store in a single variable all the data vectors (or matrices) which are characterized by different dimensions and would otherwise require singular specific array (or matrix) variables.

It is worth noting that structures and cell arrays are used to store the originally input curves and their initially preprocessed counterparts. At the end of the preprocessing, when the curves are characterized by the same length and number of data points, it is possible to use a simple matrix structure to store the data.

#### 4.3.1.3 Loops implemented in the code

Various loops are implemented into the code. In general, there are two different reasons for which it is necessary to cycle some parts of the code through various iterations: (i) repeat the same operation for a certain number of iterations or (ii) allow to cycle back to a previous part of the code which is already executed.

The former case is used to accomplish the computation of the metrics over each of the channels or the arbitrarily number of user-defined time intervals, while the latter is implemented to allow the user to go back to any stage of the preprocessing operation.

In particular, in order to allow cycling over an initially undefined number of iterations (i.e., till the user is satisfied of the results obtained) like in the case of the computation of metrics over an arbitrarily number of user-defined time intervals or the cycle to go back to previously performed

operations, a ‘while’ loop is implemented based on the value of a flag variable which changed only when the desired objective is reached.

#### 4.3.1.4 *Sign convention for flow charts*

The flowchart diagrams presented in the next sections have been created using a set of standard symbols whose aspect and meaning are displayed in Figure 4.5.

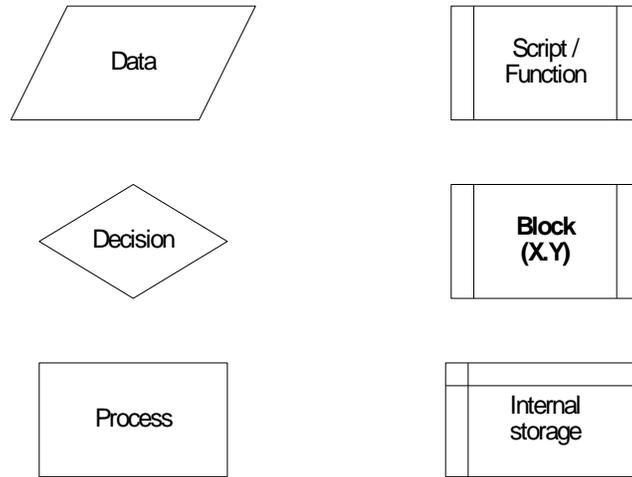


Figure 4.5: Symbols used for the flowcharts in this manual.

#### 4.3.2 **Input and preprocessing operations**

GUI's have been implemented with the purpose of facilitating the various interactions with the program.

In particular, the configuration and sequence of the GUI's which manage the input and preprocessing of the curves are specifically designed in order to obtain the following characteristics:

- Real-time feedback
- Interactivity
- Ease of process

The real-time feedback is obtained by allowing the user the possibility to see immediately how both the original and preprocessed curves look like as soon as they are input and the preprocessing options are selected. The opportunity to preprocess the original input curves in a real-time manner represents a very powerful and effective tool to obtain an immediate feedback

of how the selected preprocessing options affect the original curves. In fact, it allows the user to assess immediately if the selected options and values give the desired results and, in case, to apply immediately different preprocessing options or modify the values previously defined in order to improve the final aspect of the preprocessed curves. In particular, in order to allow the user to make a direct comparison, the graphs of both the original and preprocessed curves are plotted side by side in the same window.

The interactivity of the input/preprocessing operation allows guiding the user to during the whole process. In order to achieve this target, some of the options/values are automatically detected by the GUI's as soon as they are selected / input by the user. This makes the input and preprocessing process very interactive and quick as there is no need to either hit the Enter key or pres any button to apply the choices. Some examples of this immediate selection are the options to scale the curves or the definition of manual boundaries to trim the curves. Also, in most cases, when a selected option requires the input of additional information, a secondary GUI containing the further options pops up. This allows limiting the number of options/fields which are shown in the main GUI's in order to keep them as much simple and clear as possible.

The least, but certainly not the last characteristic which the GUI's had to accomplish is making the process as much easy and controlled as possible. In order to obtain this target, the GUI's are designed in such a way that the user is guided through the whole process step by step.

In particular, the user is forced to perform any of the mandatory steps in order to further proceed and, warning messages indicate those unselected operations which may be important but are not strictly mandatory.

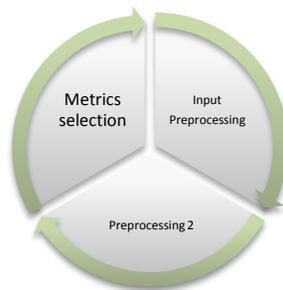
Each channel is input and preprocessed in a sequential manner so that the user is forced to go through any of the six possible channels. Note that, in case a channel is not available, it would be possible to skip it. Also, to avoid the problem of passing through all the channels in case it is necessary to make any further modification, a specific button is implemented which allows the user to go back directly to any of the previously defined channels. The list of the available buttons is automatically updated while the user moves to the next available channel.

Differently from the case in which only a single channel is input, in case of multiple channels, it is decided to implement the synchronization of the curves from the various channels into a second separate GUI.

Assuming that the all the experimental and all the numerical curves have been respectively collected/computed starting at the same time, an independent synchronization between the couples of curves of each single channel may create an undesired phase difference between the channels themselves. Although this phase difference between the various channels is irrelevant when the couples of curves from each channel are independently compared to each other (i.e., weighting factors are used to combine the results from each specific channel), it may create some problems if the resultant of the accelerations and rotational rates is computed. In fact, in order to correctly compute the resultant, the couple of curves from all the channels are supposed to be in perfect phase. For this reason, when multiple channels are compared, the synchronization task is performed only after the user has selected whether to calculate the resultant or combine the values from each channel using weighting factors. In the former case, this would allow to compute the resultants using the originally input components for both the couples of curves and apply the synchronization only after the resultants have been computed.

Overall, the main GUI's of the program are designed in order to guide the user throughout the input and preprocessing of the curves in a sequential and interactive manner.

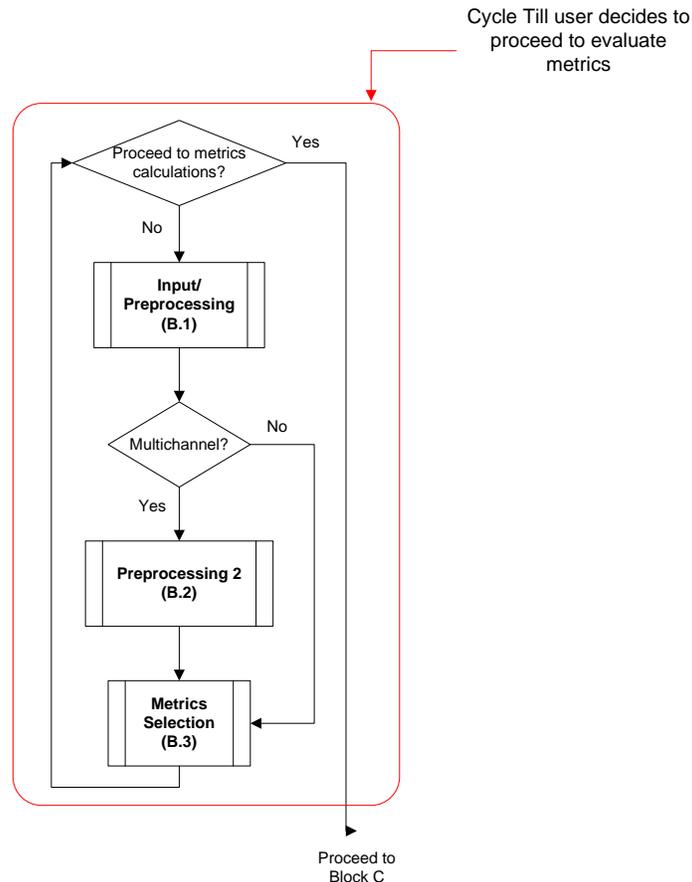
Each of the three GUI's which guide the user during the input and preprocessing of the curves and the following selection of the metrics is implemented into a specific block of the code. These three blocks are then sequentially implemented in a loop (Figure 4.6) which terminates only when the user decides to proceed to the evaluation of metrics.



**Figure 4.6: Schematic representation of the loop enclosing the GUI's for the input, preprocessing and selection of metrics.**

The implementation of the three blocks into a loop allows the possibility to go back to the previous block to apply any further change to the options selected in the previous instance. In fact, by skipping directly to the next iteration of the loop, it is possible to go back to the first block (i.e., Input/Preprocessing). Also, in case the user is executing operations implemented in the third block (i.e., Metrics selection), it would be possible to skip directly to the second block, Preprocessing 2, during the new iteration of the loop by defining a flag variable and a conditional statement at the beginning of the first block (see Appendix B for the details). In this case, given a particular value of the flag variable, the conditional statement would skip the first block and start the new iteration of the loop directly from the second block.

Figure 4.7 shows the main algorithm of the block of the code, which manages most of the input and preprocessing operations.



**Figure 4.7: Flow chart of the main algorithm for the preprocessing block of the code.**

Any time one the GUI's of each of the three sub-blocks opens, it load the options and the various data input by the user during the previous iteration of the main loop. This information is read

from variables whose structure and content will be specifically discussed ahead in this section. Using such a structure guarantees that, whenever the user goes back to the previous block, the GUI which is reopened shows the same information of the previous iteration.

Note that, in the case a configuration file has been loaded, the input/preprocessing operations performed by the main preprocessing block are skipped as they have already been performed in a previous part of the code. For detailed information about the algorithms implemented in the code and the program structure, see Appendix B.

The next sections describe the algorithms and the choices made for the implementation of the following preprocessing features:

- Filtering
- Shift/drift
- Resampling & trimming
- Synchronization

#### **4.3.2.1 Filtering**

The filter process is implemented in a user-defined function (*sae\_filter*), whose algorithm is shown in Figure 4.8. The function receives as input three variables: (i) *CFC*, (ii) *T* and (iii) *X*. The first two variables contain respectively the value of the filter class and the sampling period of the input curves, both as are scalar values; while the third variable is a vector containing the data points of the curve. The algorithm assumes that the sampling period is constant and, hence, it is not necessary to pass to the function also the time vector, as it can be recreated using the available information. This increases the efficiency of the algorithm. The theoretical background of the SAE filter has already been described in section 4.2.1.2.3.

In order to avoid the typical scatter at both the beginning and the end of the filtered time histories due to the application of the difference Equation (4.1), a dummy head and tail are added to the original data sets consisting respectively of a simple repetition of the first and last data value. The length of the head and tail is equal to the closest integer approximation of the curve frequency divided by 10.

The modified vector is then filtered by applying equation (4.1) a first time frontward and a second time backward in order to obtain a two-pass filter. The filter coefficients are calculated

using the formulas indicated in Equation (4.2) through Equation (4.6) described in section 4.2.1.2.3 of this chapter. Once the modified data sets are filtered, the previously added dummy head and tail are trimmed from the final filtered curve. Eventually, the vector containing the filtered data points is given as the function output.

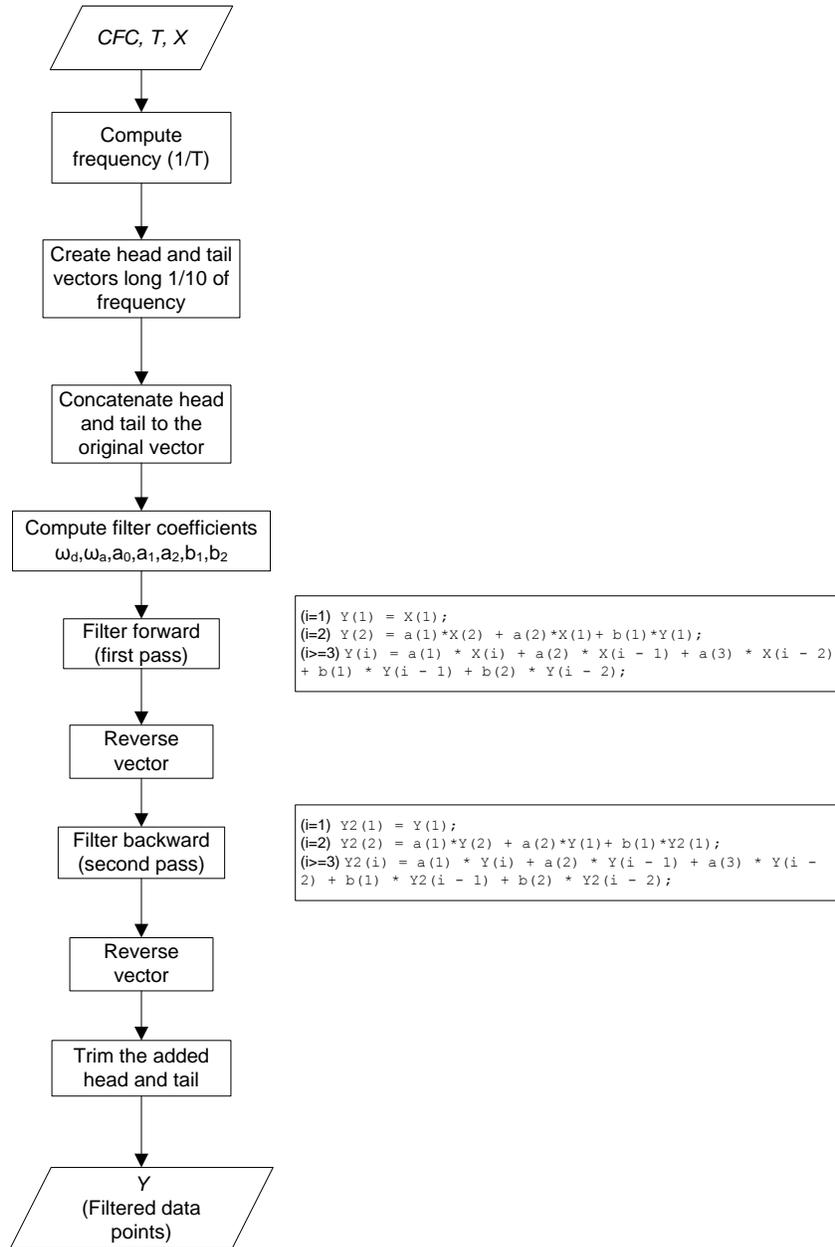


Figure 4.8: Algorithm implemented for the SAE filtering.

### 4.3.2.2 Shift/drift

The shift and drift corrections are implemented in a Matlab script (*Shift\_drift*). The main steps followed to perform this preprocessing task are shown in the algorithm in Figure 4.9. The steps described in the algorithm are performed on either one of the two curves or both of them, according to the user selection.

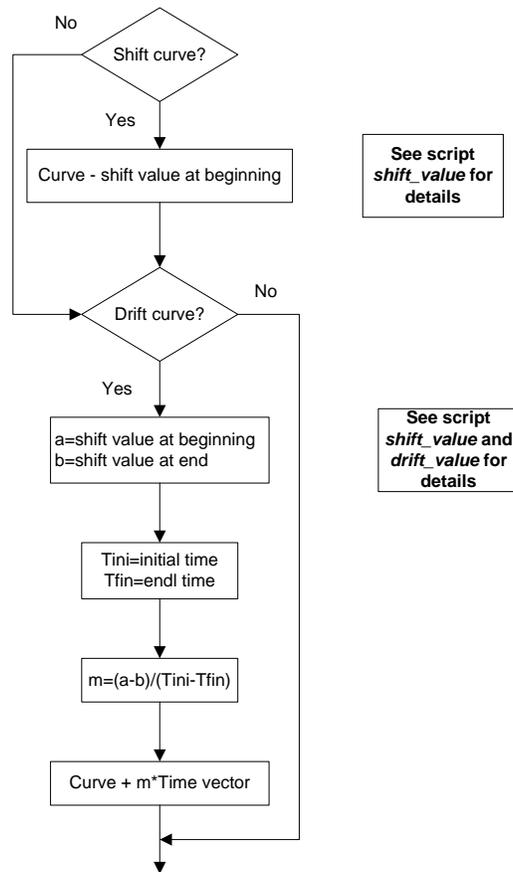


Figure 4.9: Main algorithm of the script *Shift\_drift*.

As indicated in the flowchart in Figure 4.9, the shift values at the beginning and the end of the curve are computed respectively by the user-defined functions *shit\_value* and *drift\_value*. The flowchart of the algorithms for these two functions are shown in Figure 4.10. The former function evaluates the vertical shift at the beginning of the curve, while the *drift* function assesses the value of the shift at the end of the curve.

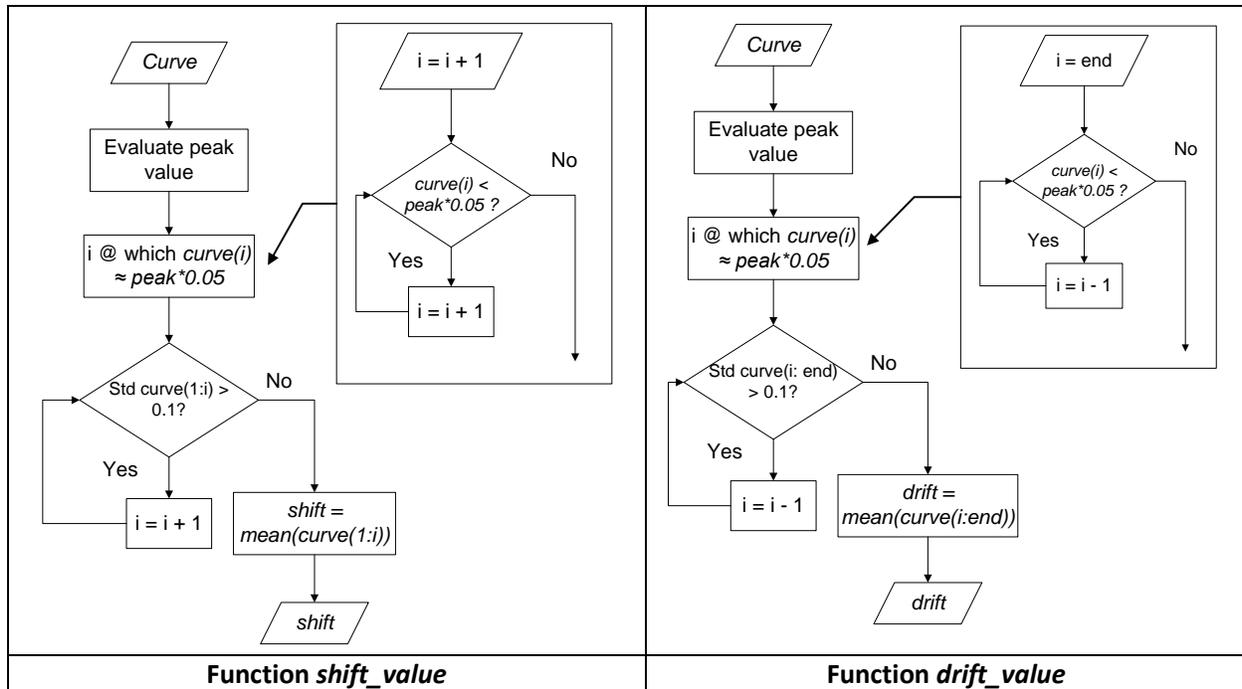


Figure 4.10: Algorithm of the user-defined functions *shift\_value* (left) and *drift\_value* (right).

The algorithms of both the two functions are very similar; in both of them the shift values are computed as the mean of respectively the initial and final portion of the original data vector obtain considering the data points till/from five percent of the peak value (Figure 4.11).

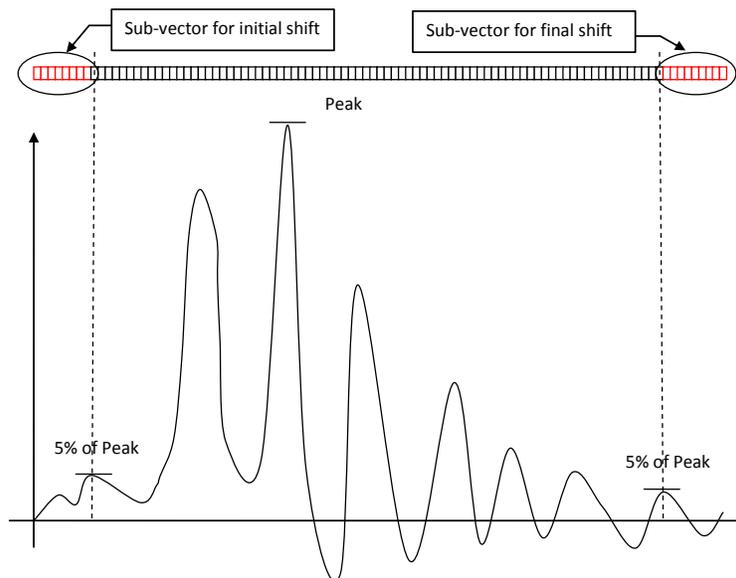


Figure 4.11: Sketch of the sub-vectors derived from the original data vector.

Also, if the standard deviation computed from the values of the data point contained in the sub-vector is greater than 0.1, the algorithm iteratively decreases the size of this sub-vector till the standard deviation drops within a value equal or smaller than that.

#### ***4.3.2.3 Resampling and trimming***

The program automatically checks if the two sets of data have the same sampling period within a tolerance of 5E-6 sec. If the two curves do not have the same sampling frequency, RSSVP proceeds to re-sample the curve which has the lower sampling rate (i.e., the bigger difference in time between two contiguous data points) at the higher rate of the other curve.

The resampling and trimming operations are implemented simultaneously in the same Matlab script (*Resampling\_trimming*) whose algorithm is shown in Figure 4.12.

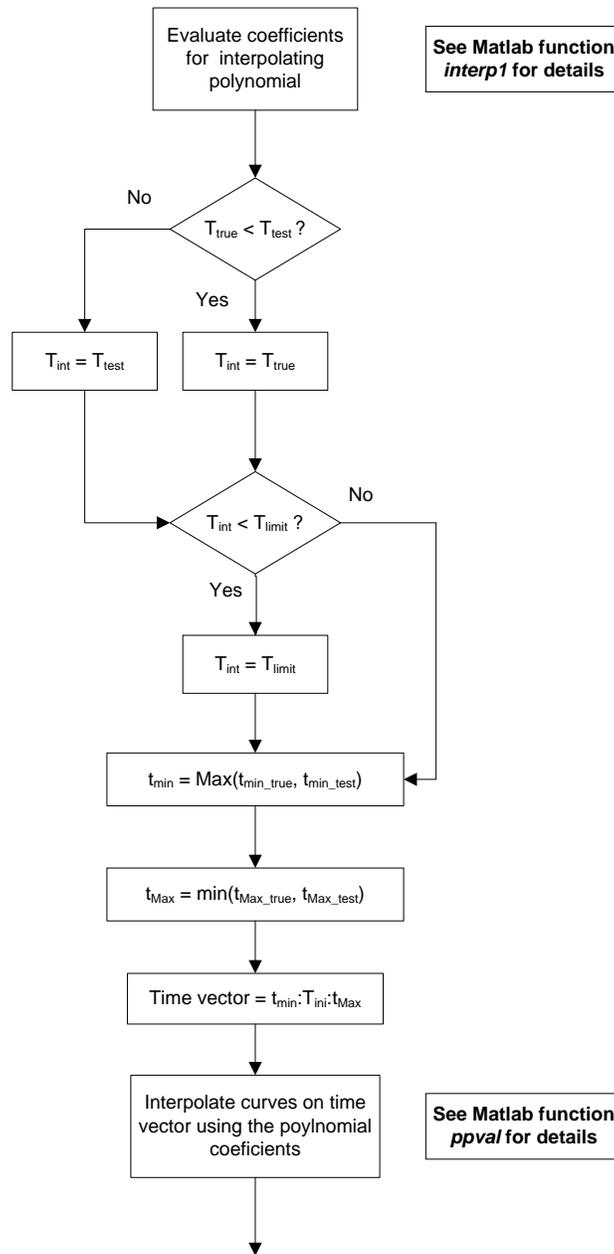


Figure 4.12: Algorithm of the script *Resampling\_trimming*.

The re-sampling is performed by mean of a linear interpolation, assuming that the time vector starts from zero. In particular, the interpolation uses a linear polynomial whose coefficients are evaluated using the predefined Matlab function *interp1*. Also, when the two curves are re-sampled, the maximum initial time values and the smaller final values between the two original time vectors are considered, in order to trim them to the same interval. Note that, because of the new sampling rate the end value of the new time vector may be approximated by defecting the maximum time of the original curve.

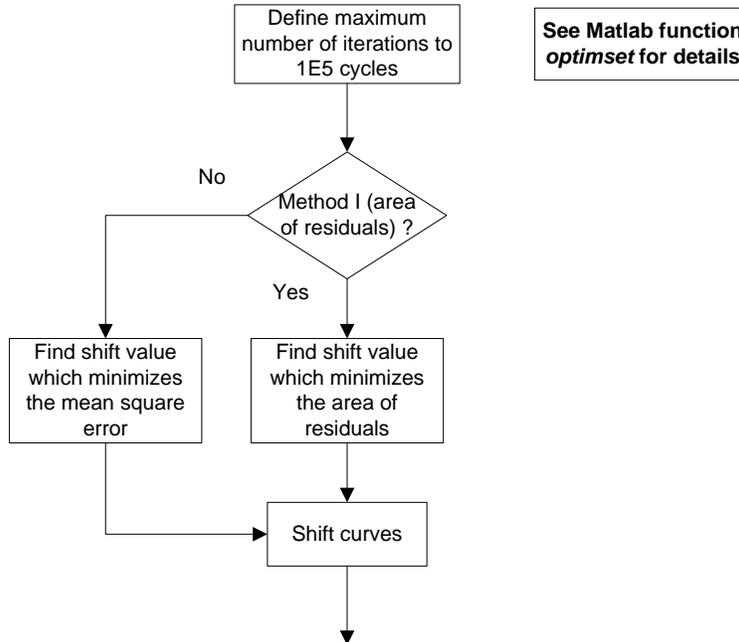
The sampling period (i.e., the minimum time between the collection of two consecutive data points) is defined as the minimum between that of the two curves and, in any case, it cannot be smaller than a minimum limit value (defined by the user at the beginning of the calculation). In case of multiple channels, the minimum sampling period is determined by considering the first input channel; the same period is then used to resample also any of the other channels.

Eventually, both the two curves of each channel are interpolated on the trimmed time interval using their respective polynomial coefficients through the Matlab function *ppval*. In case of multiple channels, the same procedure is repeated for each of channel.

In the multichannel mode, after each pair of curves has been trimmed, the various channels are further trimmed to the length of the shortest channel.

#### **4.3.2.4 Synchronization**

The automatic synchronization of the curves is implemented in a specifically implemented Matlab script (*Curve\_synchronizing*) whose algorithm is shown in Figure 4.13. This script calculates the shift value between a couple of curves which minimizes a target function. Specifically, two different target functions are considered in the implementation: (i) the area between the two curves or (ii) the square error. These two methods are respectively defined as area of the residuals method and least square error method in section 4.2.1.2.6. The corresponding two target functions are recalled by user-defined Matlab functions, respectively *area\_res* and *rse*. In both the two functions, the couple of curves can be shifted respect to each other by an arbitrary value  $s$ . The minimization process is performed using the available Matlab function *fminsearch* which iteratively calculates the values of the selected target function (i.e., area of residual or square-root error in this case) varying the value of the shift variable in order to find the optimal solution. Eventually, the algorithm shifts the two original curves by the optimal value obtained from the minimization process.



**Figure 4.13: Algorithm of the script *Curve\_synchronizing*.**

Both the user-defined functions *are-res* and *sre* require as input only the value by which to shift the two curves before calculating respectively the area between them or the square-root error. The algorithms of these two functions are shown in Figure 4.14. For either of these two functions, the shifting of the two curves is performed by invoking the user-defined function *shift* whose algorithm is shown in Figure 4.15.

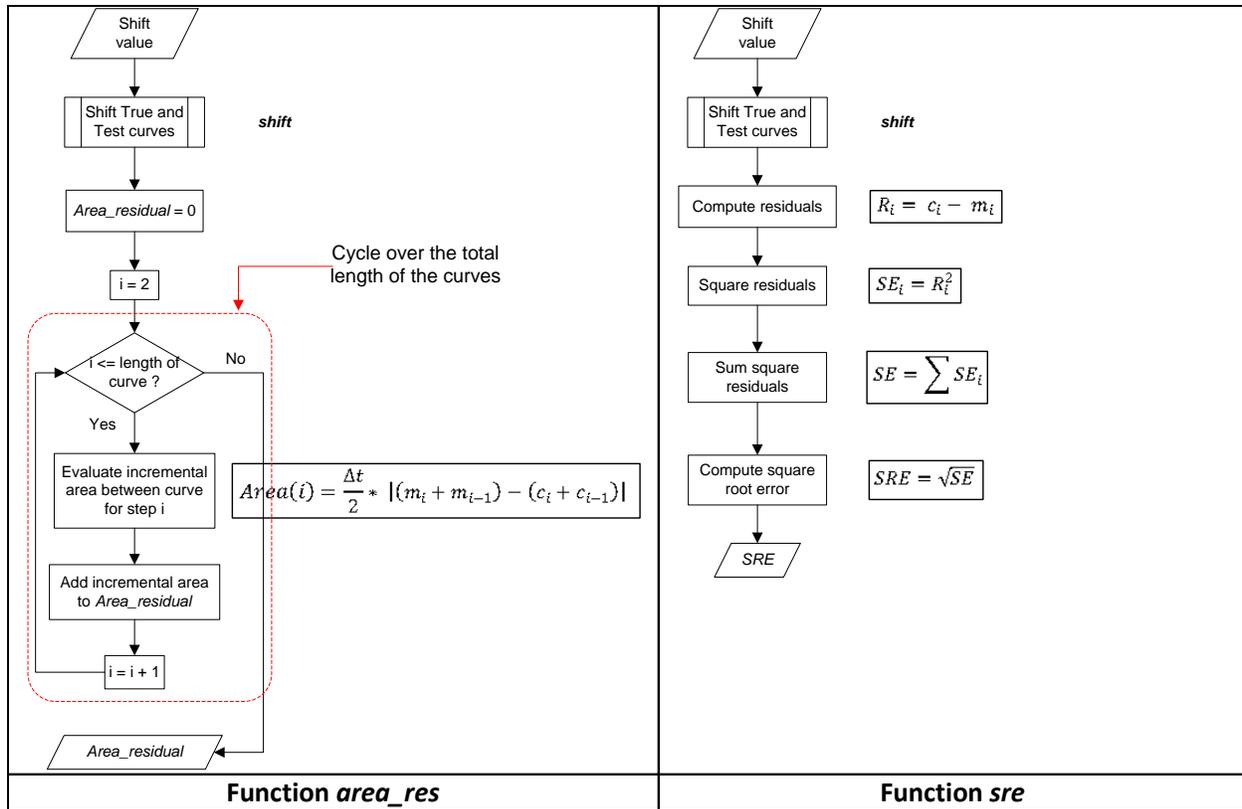
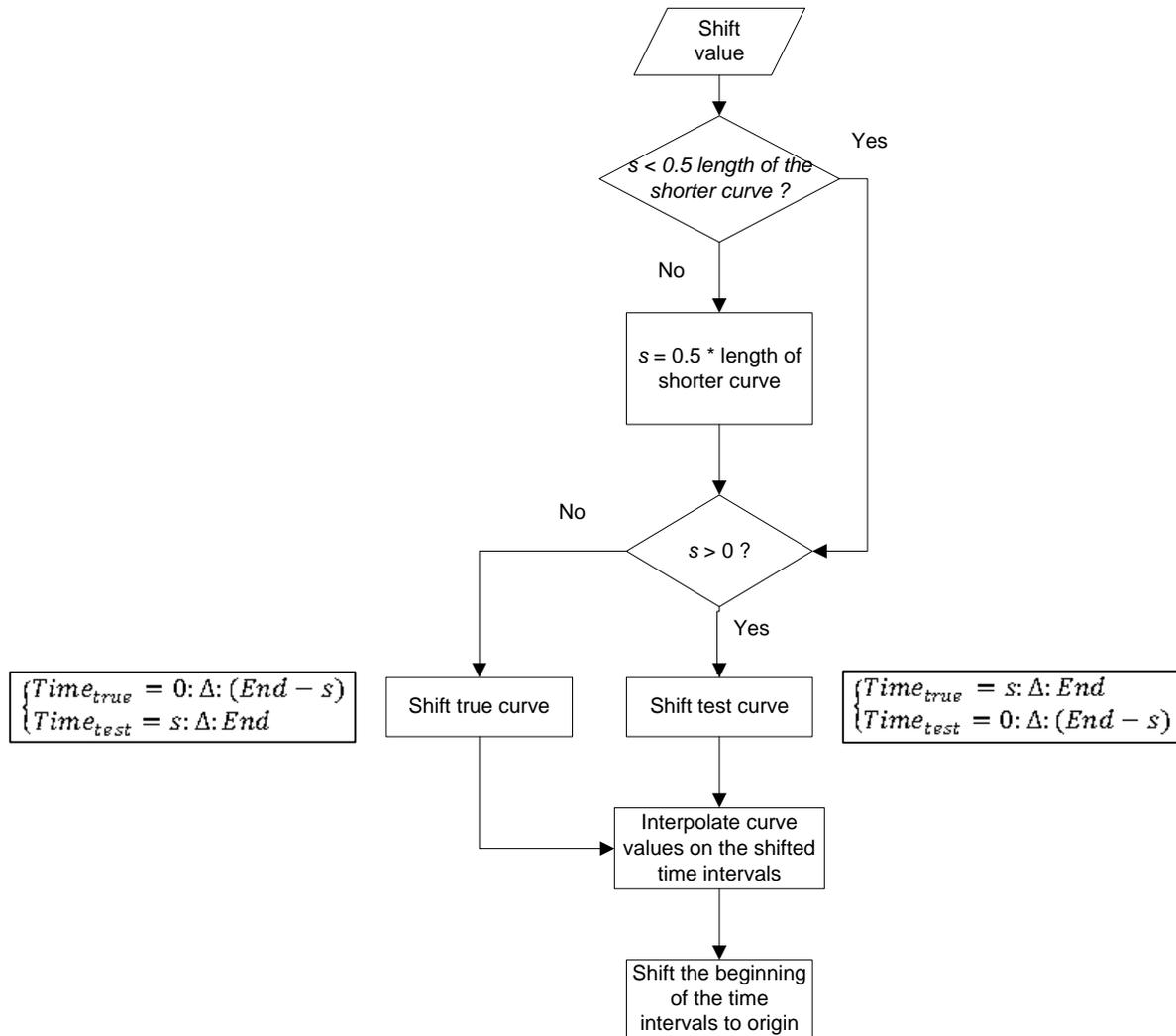


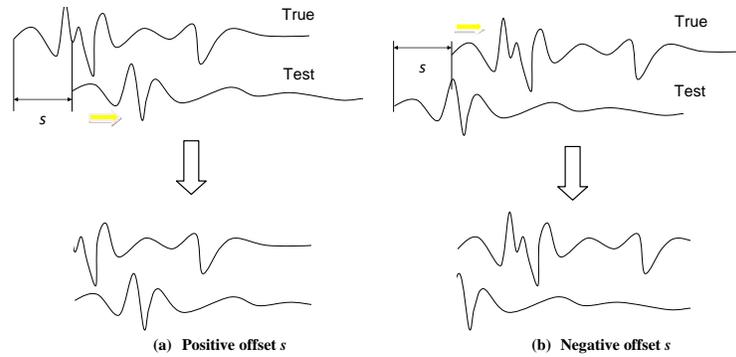
Figure 4.14: Algorithm of the user-defined functions *area\_res* (left) and *sre* (right).

The function *shift* has only one input, the value by which to shift the two curves respect to each other. For this function the data points of the two curves are read directly from the global space of Matlab, without the need to pass the information to the local space of the function using input variables. This is a necessary step as it allows the shift function to be recalled also within the local space *area\_res* and *sre*, where the data points of the input curves would not otherwise be available. Based on the sign of the input shift value, the algorithm of the function *shift* creates a shifted time vector for either one of the curves and a vector trimmed at the end for the other curve.



**Figure 4.15: Algorithm of the user-defined function *shift*.**

In particular, in order to have two vectors with the same length, a positive shift value ( $s$ ) corresponds to a time vector for the baseline curve which starts at the shift value and a time vector for the other curve which is trimmed at the end by the same shift value. Vice versa happens in case of a negative shift value. Figure 4.16 shows a sketch of the process implemented to shift the couple of curves.



**Figure 4.16: Sketch of the behavior of the shift subroutine for a (a) positive or (b) negative input.**

Once the appropriate time vectors have been defined, the shifted curves are obtained by computing the corresponding interpolating polynomials on these time points. Eventually, a common time vector is used for both the interpolated curves, which starts at time zero and is trimmed at the end by a value equal to the shift.

In case the result is not satisfactory, the user can repeat the synchronization procedure either using a different initial shift value at the beginning of the minimization algorithm or using the other minimization method.

### 4.3.3 Single and multiple channel computation (comparison)

As already mentioned in section 4.2.2.2, different methods to assess the weighting factors are considered in this research. This section describes the details of the implementation of the three weighting schemes investigated:

- Peak values
- Area of curves
- Inertial properties

The basic concept for all the three weighting scheme is to calculate a sort of local index for each channel which could be representative of the importance (or weight) of the specific channel respect to the others of the same group. Once these indexes are computed, the weighting factors are calculated by simply dividing the index from each channel by their total sum. In this way, the summation of the obtained weighting factors is the unity.

The weighting factors are evaluated for both the acceleration and rotational rate channels as both the groups of channels are used to calculate the final metrics. Note that, in order to allow a

correct computation of the weighting factors when both the acceleration and velocity channels are used, the definition of the local indexes has to take into account the different magnitude and units of these two groups of channels.

The difference between each of the three implemented weighting schemes is in the way the initial local indexes are evaluated to compute the final weighting factors. Following is a description of how these local indexes are computed for each of the four weighting schemes. Note that all the calculations are made based on the baseline curve of each channel.

#### ***4.3.3.1 Peak values***

The absolute peak value of the acceleration is evaluated for each of the three acceleration channels. The local indexes are calculated dividing each peak value by the maximum peak for all the three acceleration channels.

A similar procedure is performed for the three rotational rate channels. Normalizing the peak values for both these two groups ensures that the acceleration and velocity channels and avoids the problem of having different units (i.e., accelerations and rotational velocities).

#### ***4.3.3.2 Area of curves***

This method is similar to the method of the Peak Values, but in this case the absolute value of the area of the acceleration (rotational rate) channels is computed instead of the absolute peaks. Also in this case, for the calculation of the local indexes, the maximum of the peaks from each group of channels is picked and each acceleration/velocity channel is divided by the maximum peak from the respective group.

In the alternative formulation previously discussed, the area obtained from each channel is divided by the sum of the areas (instead of the maximum of the peaks) from the respective channel group. Note that this solution implies that the total weight of the acceleration channels has to be 50 percent and that the remaining 50 percent is assigned to the cumulative weighting factors of the rotational rate channels.

#### ***4.3.3.3 Inertial properties***

The linear and rotational momentum are computed by respectively multiplying the vehicle mass by the area of the corresponding acceleration channel (i.e. the velocity) and each momentum of

inertia of the vehicle by the area of the corresponding rotational rate channel. Note that, in case the rotational moment is evaluated, it is assumed that the arm of this momentum is equal to the unity; this assumption allows to compare directly the linear and rotational momentum.

In this case, the local indexes are evaluated by dividing the momentum computed for each channel by the value of the maximum of all the momentum.

#### **4.3.4 Output of results**

The developed software creates two types of output files: (i) ASCII files and (ii) Excel spreadsheets. The ASCII files are used to save the histories of both the original and the preprocessed curves for the various channels input at the beginning of the program execution; while the Excel spreadsheets store in a logical and organized manner the various metrics values.

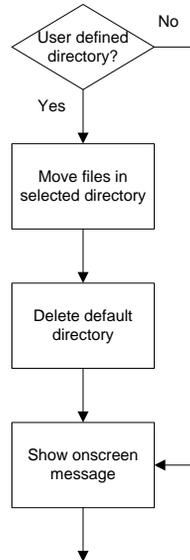
Due to the simple tabular nature of the curves time histories, the creation of the ASCII files is easily implemented by saving the vectors containing the data points of the curves in text format. A specific file is created for each of the curves.

As for the output of the metrics values, for the reasons already described in section 4.2.3 it is decided to arrange them in the form of spreadsheet tables. In fact, according to the choices made by the user during the execution of the program, it may be necessary to organize the metrics values according to one or more of the following three criteria:

- Metrics selected
- Interval on which curves are compared
- Channels compared

Also, as mentioned in section 4.2.3 the output of the results is implemented using two different approaches: (i) on the screen and (ii) using files. The former method is used to provide a real time interaction with the user during the execution of the program, while the latter is a convenient mean to store the results at the end of the calculation.

The program offers the user the possibility to define a folder in which to save all the output files created during its execution. Also, in case this folder is not created by the user, it creates a defaults folder in the program directory which is sequentially numbered considering the latest result folder already present in that directory (Figure 4.17).



**Figure 4.17: Flowchart of the algorithm for creation of the results folder.**

The following sections describe the data structures used to organize the various output information and the algorithms implemented to output the results using either of these two above mentioned methods.

#### **4.3.4.1 Metrics time histories**

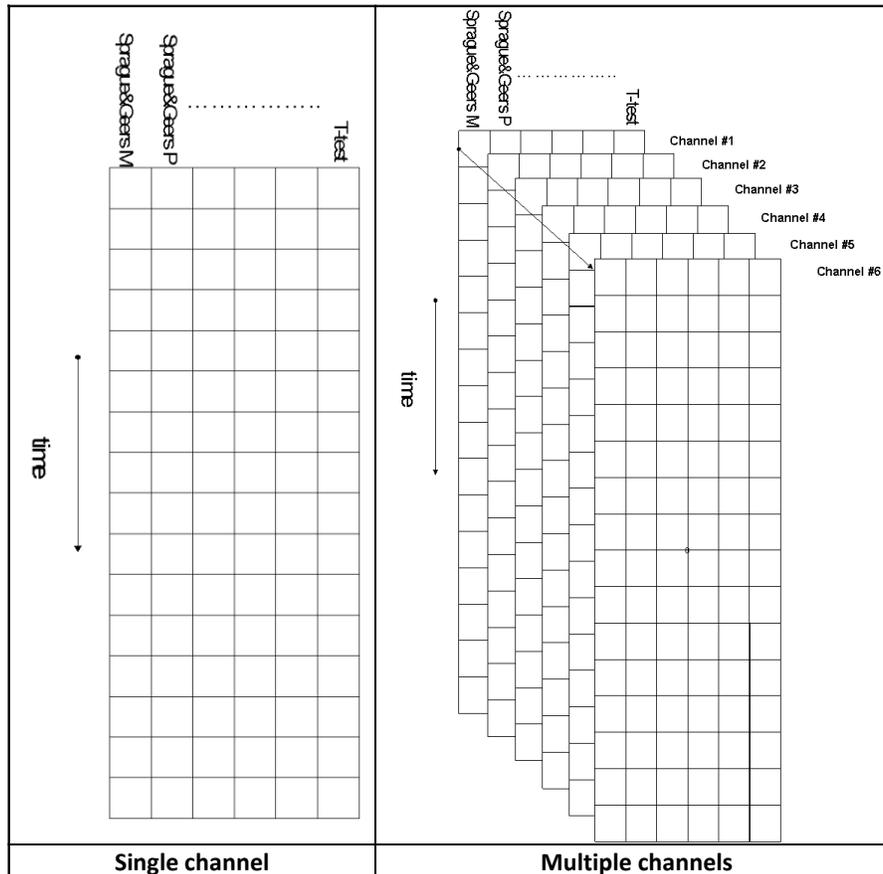
The values of the comparison metrics computed during the execution of the program are saved in various variables; each of them is characterized by a typical organization of the data. In general, the variables containing the output results could be distinguished into two different categories, according to the type of information they contained: (i) time histories of the metrics and (ii) final values of the metrics.

Each of these two typologies of variables are characterized by two different data structures, containing respectively the results for either a single channel or the weighted average and the results from each of the channels in case multiple channels are input.

##### **4.3.4.1.1 Data structure of the variable containing the results**

As described in section 4.2.2.1, it may be useful to assess how the values of the metrics evolve in the domain in which the curves are defined. For this reason, apart from the final values of the comparison metrics computed considering the entire interval, also a sort of history of each metric is created considering increasing intervals till the entire domain is reached.

The data structures which characterize the two different types of variables used to store the time histories of the metrics are shown in Figure 4.18. In both the two structures, the time history of each computed metric is organized as the column of a matrix. In case of multiple channels, the structure becomes a 3-D matrix in which the results for each of the input channels are stored along the third dimension of the variable.



**Figure 4.18: Sketch of the data structure used to store the metrics time histories for a single input channel (left) and multiple channels (right).**

#### 4.3.4.1.2 Creation of the output file

The program saves the histories of the metrics in an organized manner using Excel files. Figure 4.19 shows the flowchart of the algorithm used to store the information contained in the data structures described in the previous section.

The first step performed is the selection of the name to be given to the Excel sheet containing the metrics time histories for either the single channel or the weighted mean. The algorithm then creates an Excel file containing the results for the comparison between the curves on the whole

time interval and/or separate Excel files for each of the user-defined time intervals, depending on which selection is made during the execution of the program. In the latter case, a loop cycles over the number of time intervals defined by the user and creates an Excel file during each iteration. Also, if multiple channels are input, in either the case the comparison is performed on the whole time interval or user defined time interval/s, the algorithm cycles over each of the input channels in order to save them in separate sheets of the same excel file.

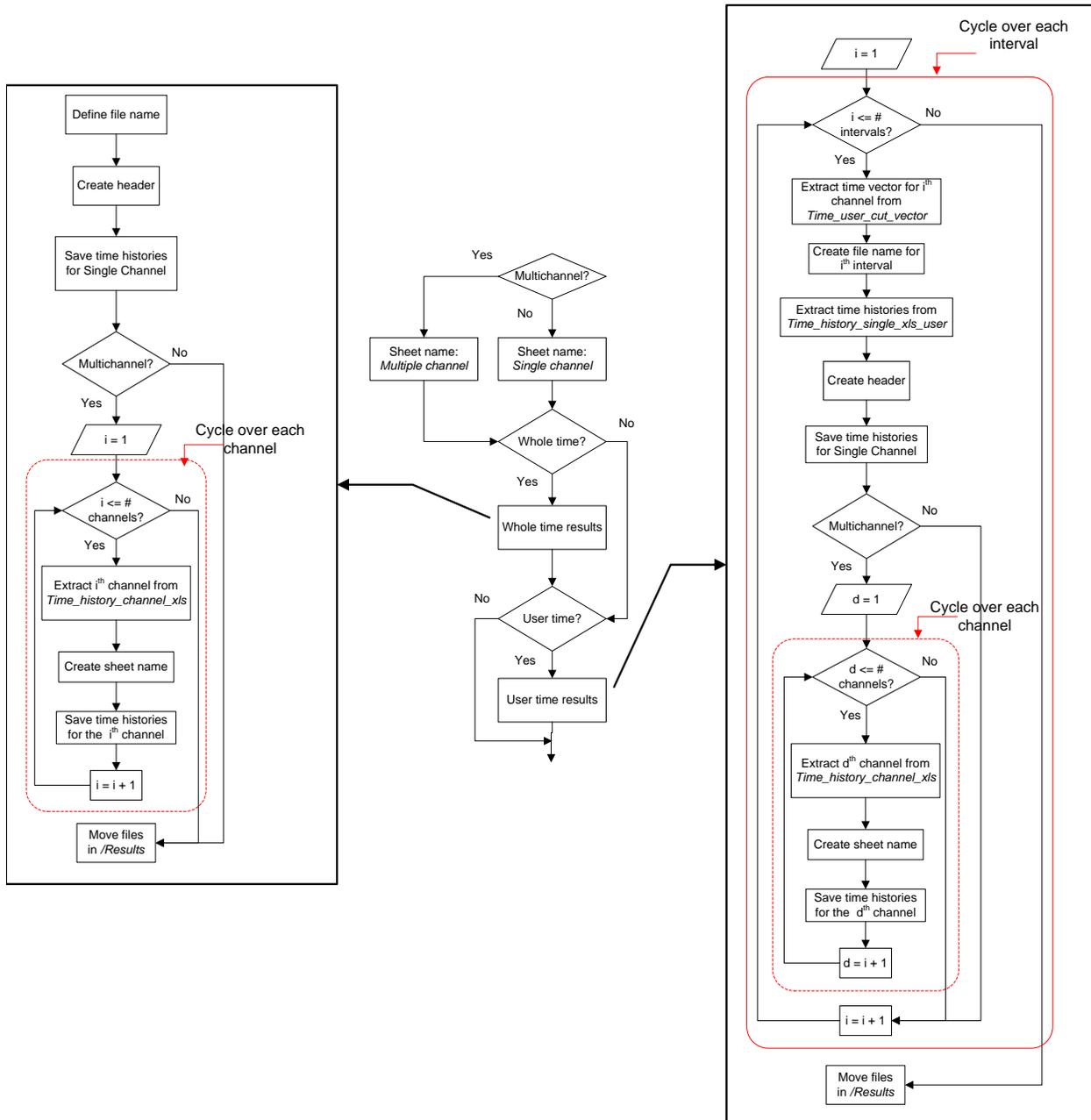
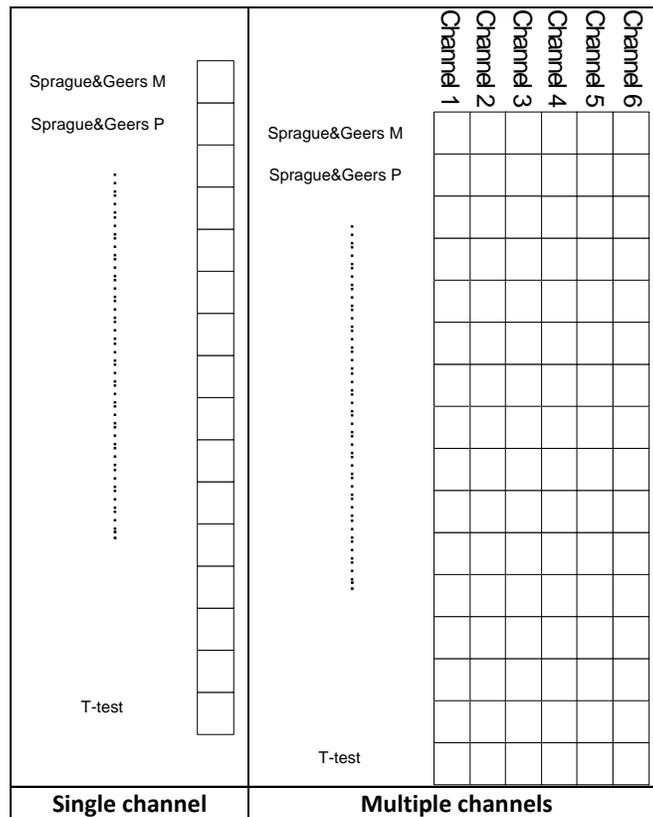


Figure 4.19: Flowchart of the algorithm for the output of the histories of the metrics in Excel format.

### 4.3.4.2 Metrics final values

#### 4.3.4.2.1 Data structure of the variable containing the results

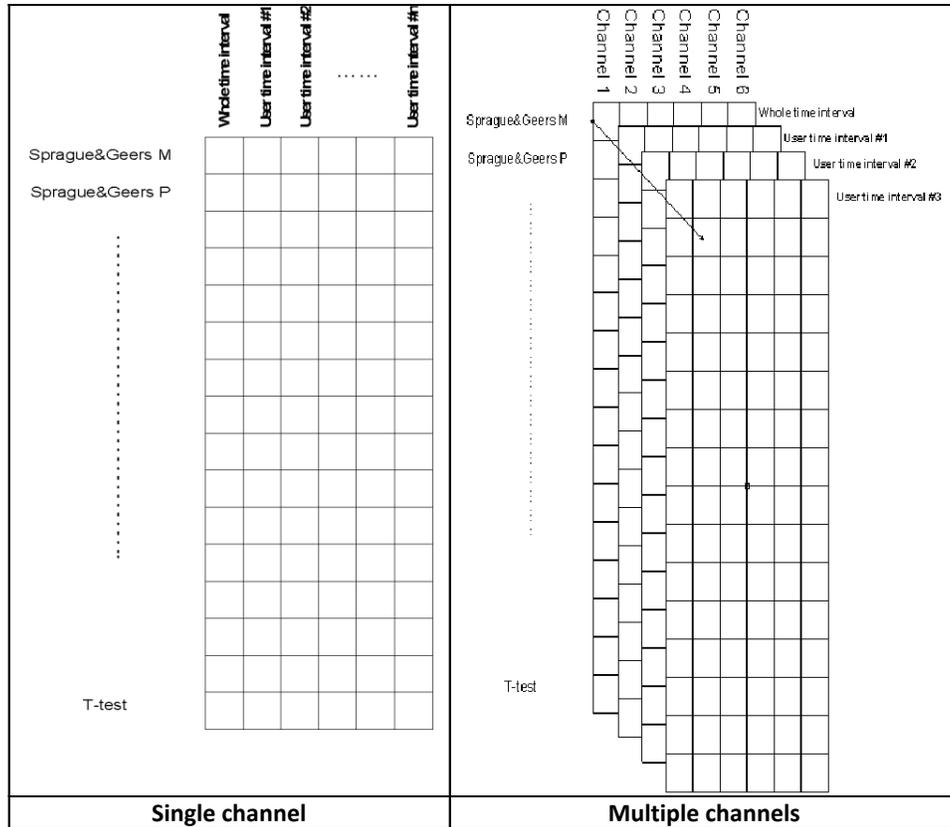
While it is necessary to allocate an entire vector of data points to store the time history of each computed metric, the memorization of their final values require simple scalar numbers. This means that the data structures necessary to store the metrics final values are one dimension smaller than the corresponding structures used for the time histories. Figure 4.20 shows a representation of the data structures used to store the final values of the metrics for each of the sub-intervals considered during the execution of the program. The values are rounded at the first decimal digit for the MPC and Single-value metrics, and to the second decimal digit for the ANOVA metrics.



**Figure 4.20: Sketch of the data structure used to store the metrics final values for a single input channel (left) and multiple channels (right).**

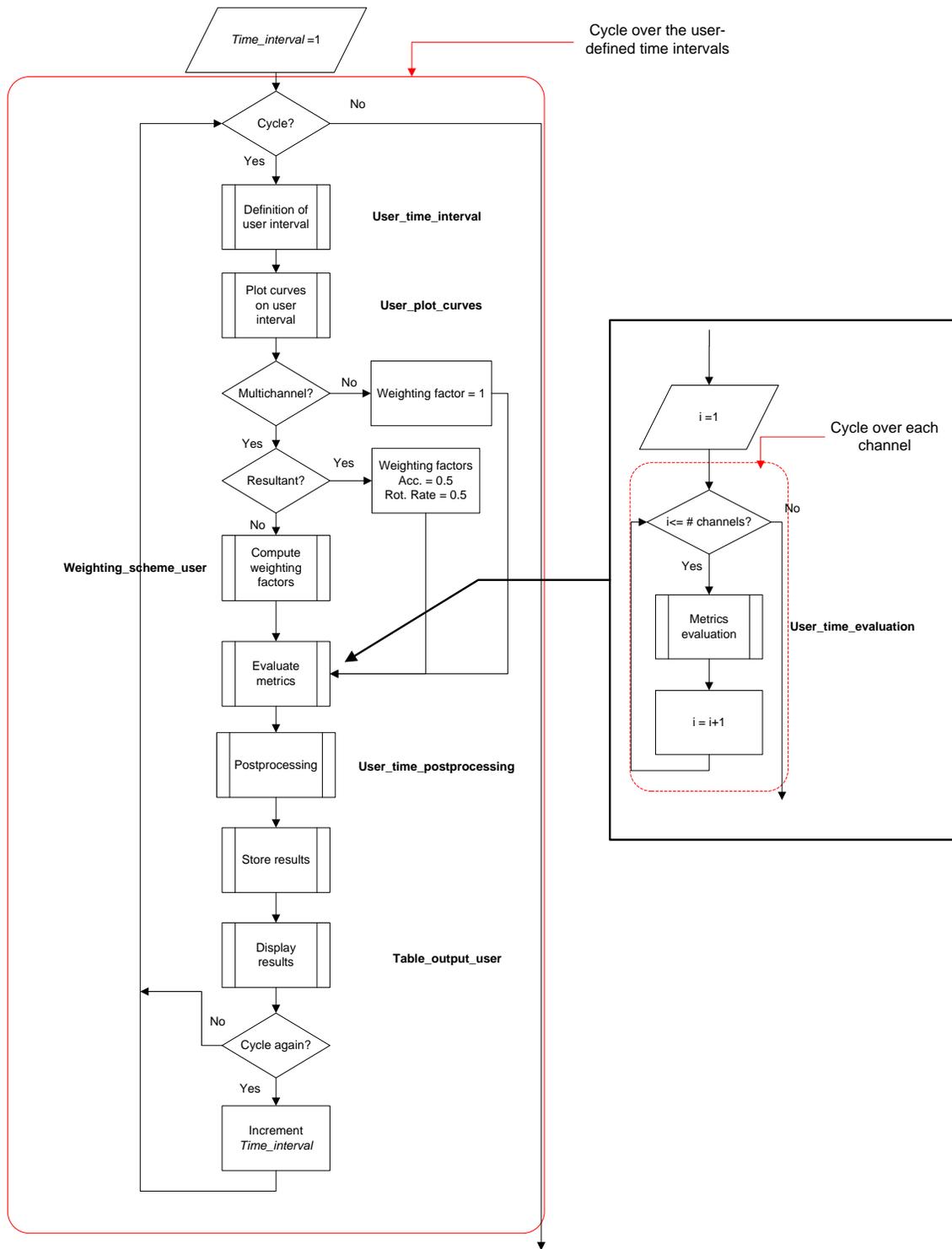
In case the metrics are computed on more than one time interval (i.e., the user decided to compare the couple/s of curves on one or more portion/s of the original domain in which the curves are defined), it is necessary to increase the data structures used to organize the final

metrics for both the single and multiple channels by one dimension as shown in Figure 4.21. It is worth noting that the structure chosen to organize the results has the main advantage of allowing to sequentially retrieve the values through the use of a loop.



**Figure 4.21: Sketch of the data structure used to store the metrics final values computed on different sub-intervals for a single input channel (left) and multiple channels (right).**

In order to compare the curves on an arbitrary number of user-defined sub-intervals, the script for the computation of the metrics and the following storage of the results in the appropriate variables are enclosed within a loop which terminates only when the user is satisfied by the results and decides to quit the program. The algorithm of this loop is shown in Figure 4.22.



**Figure 4.22: Flow chart of the algorithm for the computation of the metrics over user-defined intervals.**

The structures of the variables used to store the results at each iteration of this loop (i.e., each computation of the metrics over a new user-defined subinterval) is the same shown in Figure

4.20. Once the values of the metrics for each sub-interval are computed, they are appended to the corresponding variables initially used to store the results of the comparison over the entire domain of the curves. In particular, the results from each iteration of the loop are appended as either a new column, in case of a single channel comparison, or as a new matrix of data along the third dimension, in case of a multichannel comparison.

Note that it is necessary to append the various results for each user-defined time interval (i.e., each iteration of the loop) to the corresponding variable. In fact, as multiple user time intervals may be defined, at the beginning of each new iteration the matrices containing the results for the specific time interval considered are overwritten.

#### **4.3.4.2.2 Creation of the output file**

The flowchart of the algorithm implemented to output the values of the comparison metrics is shown in Figure 4.23. The algorithm manages the creation of the Excel file containing the metrics values for the various user-defined intervals considered during the execution of the program. In case of multiple channels the algorithm also cycles over the total number of the input channels and creates specific sheets containing the results for each of them.

The first step performed by the algorithm is to create the headers for each of the time interval considered during the run of the program and write them in the main sheet of the Excel file. In case of multiple channels, a new sheet is written for each channel and the same headers are written also for any of the channel sheets. This operation is implemented in a loop which cycles over the number of input channels.

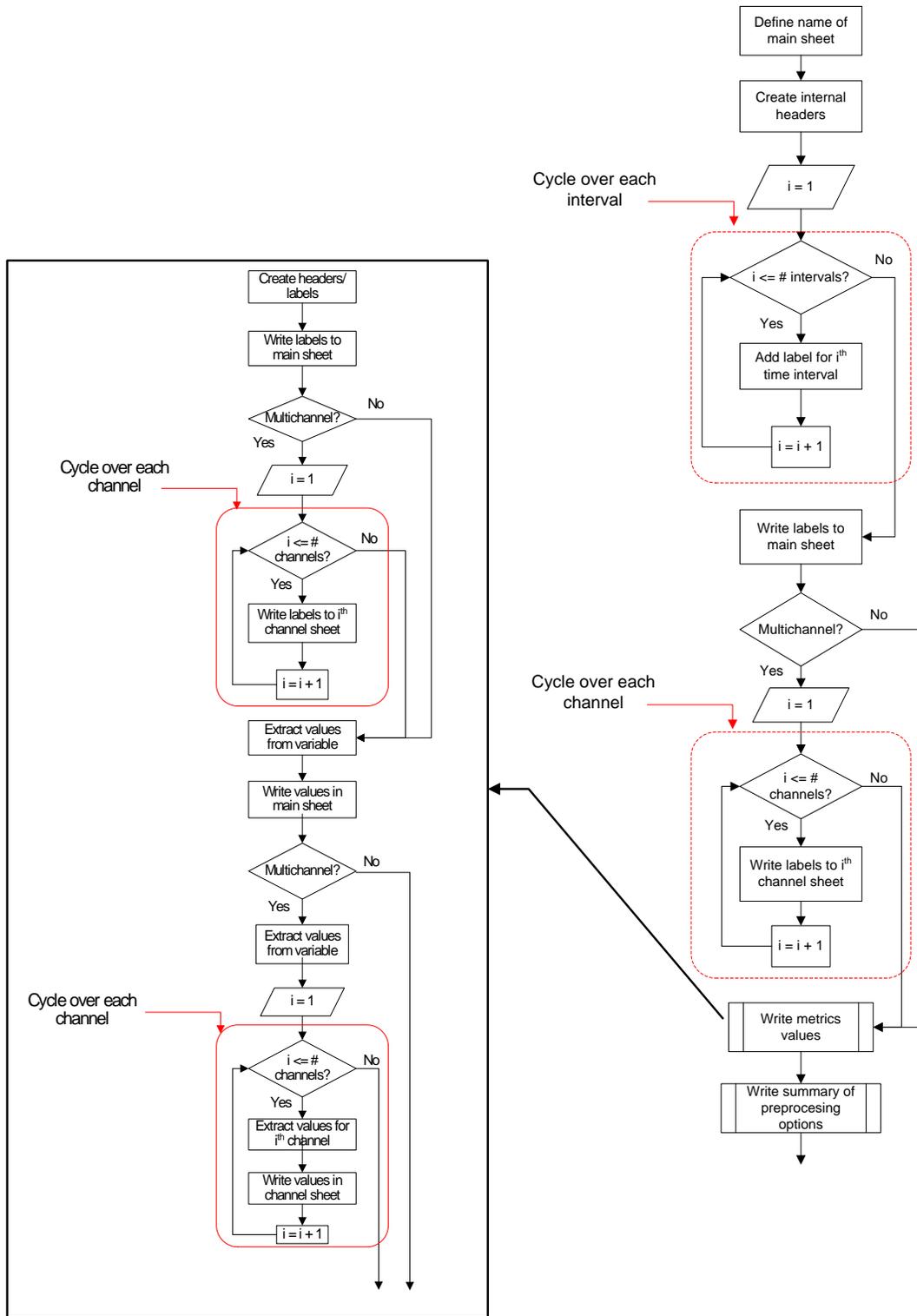


Figure 4.23: Flowchart of the algorithm for the output of the final values of the metrics in Excel format.

The results created during the computation of the metrics are logically stored in different variables and, as discussed in the previous section, each of them is characterized by a specific data structure which allows to easily retrieve the data that have to be written in the Excel files.

In order to divide each of the three groups of metrics that the program is capable to compute with a specific header in the Excel file, after the header for a different group has been created, the script proceeds to extract the values from the corresponding variable. The values corresponding to a specific group of metrics are extracted by considering only a certain range of rows of the matrix where the results are stored. Note that, number of rows extracted varies according to the corresponding number of computed metrics.

For the results computed considering only one channel or the weighted average, the variable containing the values is a bi-dimensional matrix in which each column represents a specific interval; while for the results of multiple channels, a 3-D matrix is used (for details about the structure of this variable refer to section 4.3.4.1.2). In the latter case, a similar extraction procedure is applied on the 3-D matrix in order to extract only the desired metrics values. Also, in this particular case, the results for every channel are extracted by cycling over each of them. The process followed to extract the necessary information from the variable containing the values of the metrics for each channel compared during the execution of the program is graphically shown in Figure 4.24.

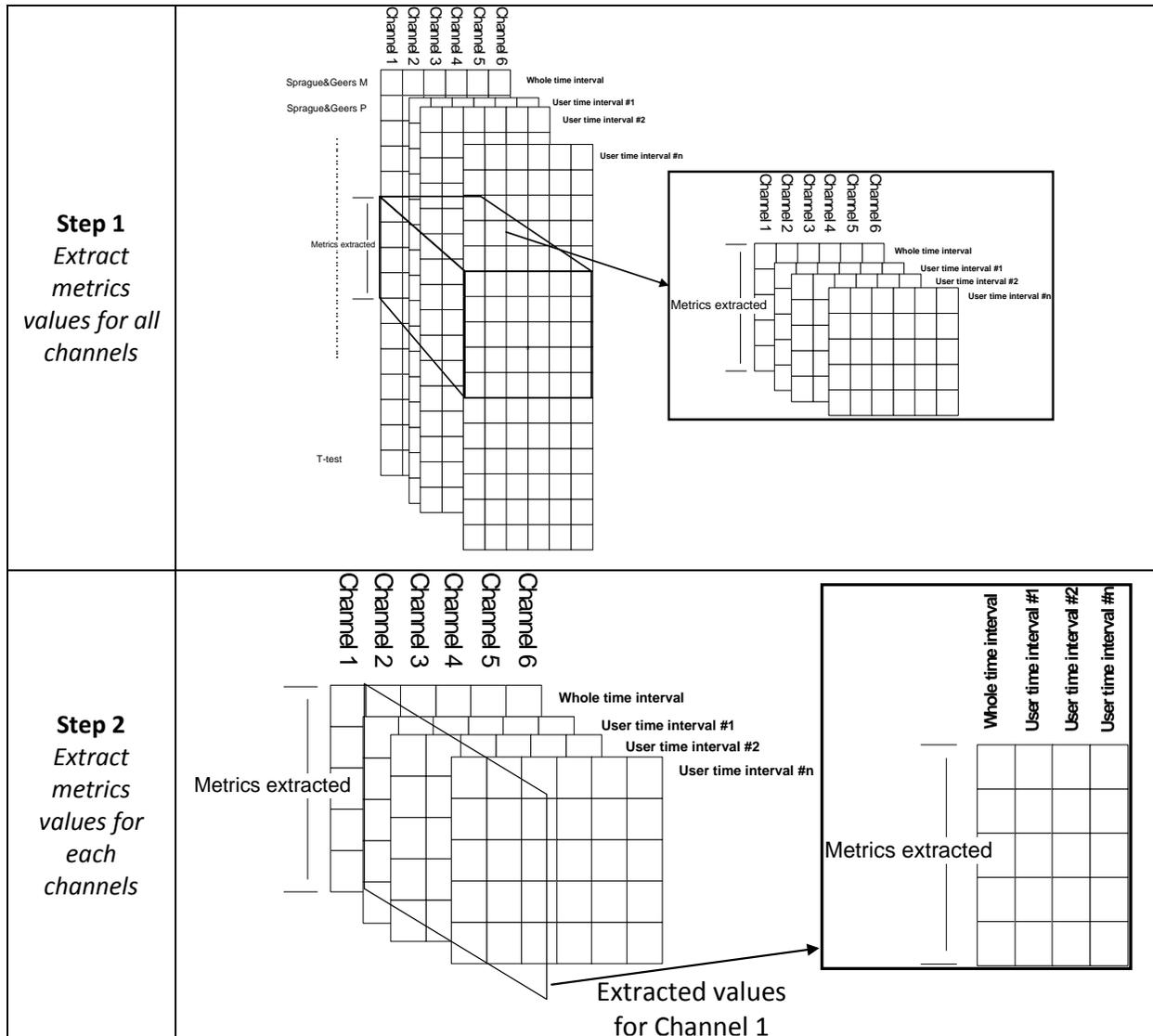


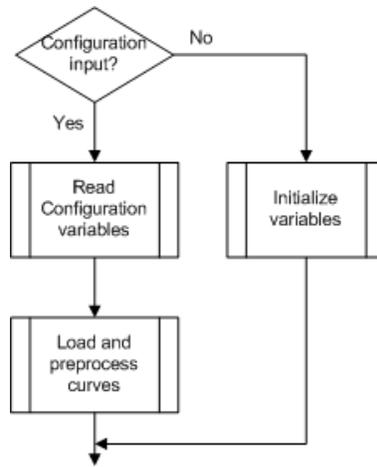
Figure 4.24: Process for the extraction of the data from the variable containing the metrics values for the various intervals considered.

### 4.3.5 Configuration file

At the beginning of the execution, the program offers the user the possibility to load a configuration file. As previously mentioned in section 4.2.3, this file contains all the necessary information to load and preprocess data from a previous execution of the program.

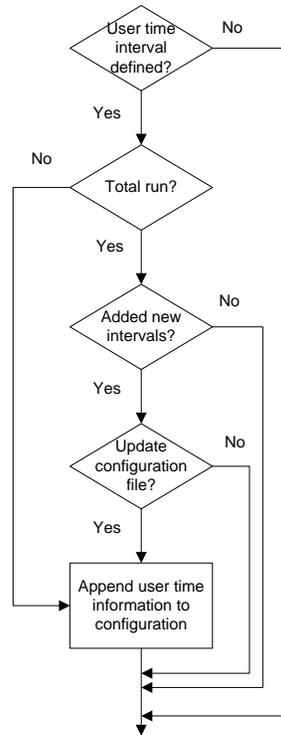
In order to implement this feature, the code searches for the configuration file at the beginning of the run to load the variables containing the configuration information; then it performs the preprocessing accordingly.

In case, no configuration is loaded, the code initializes the corresponding variables with some default options and proceeds normally. Figure 4.25 summarizes this process.



**Figure 4.25: Flowchart of the algorithm for the initialization of the variables at the start of the program.**

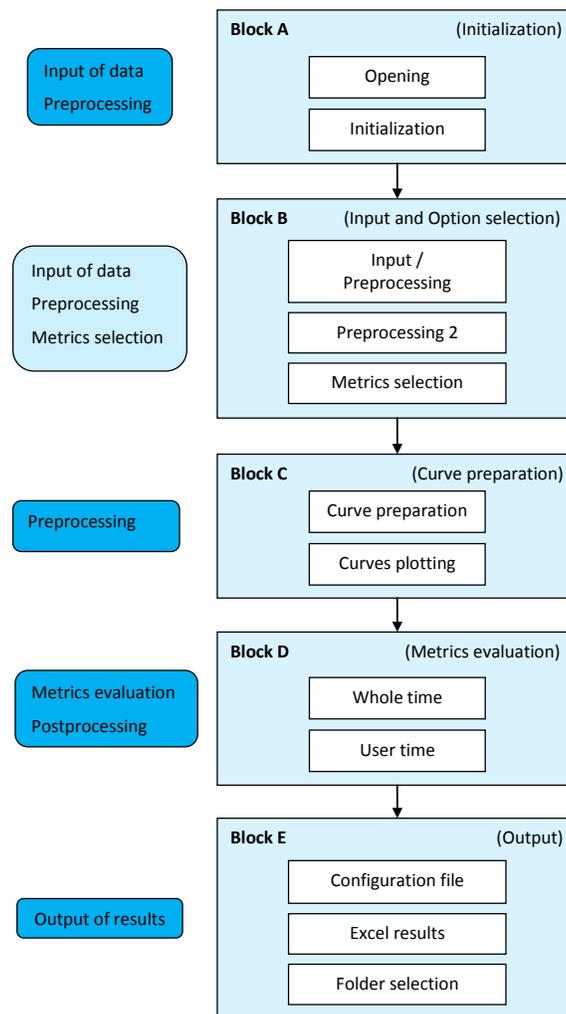
In any case, the code automatically creates a configuration file at the end of the preprocessing operation and, if the curves have been compared over any user-defined interval, it includes this information at the end of the program execution (Figure 4.26).



**Figure 4.26: Flowchart of the algorithm for the update for the configuration file at the end of the program.**

### 4.3.6 Logical Organization of the code

The code is logically divided into five major blocks and each of them performed one or more of the tasks described in section 4.2. Figure 4.27 shows the diagram of the five blocks in which the code can be divided. On the left side of each block are listed the tasks respectively performed. Note that each block may achieve more than one task as, due to various reasons which will be explained later in this chapter, in general, it is not possible to implement each of the tasks described in section 4.2 in a specific corresponding block (i.e., a one to one correspondence between the tasks and the blocks).



**Figure 4.27: Diagram of the five blocks which compose the computer program.**

As can be seen from the diagram in Figure 4.27, the code is composed of the following five blocks:

- A. Initialization
- B. Input & Options selection
- C. Curve preparation
- D. Metrics evaluation
- E. Output

Every block is divided into two or three sub-blocks, each of them performing specific operations.

The first block (Initialization) implements the initial operation necessary to start the program:

- i) definition of the type of comparison (Single/Multiple channels),
- ii) selection of basic choices,
- iii) creation and/or initialization of most of the variables used during the execution of the rest of the code and,
- iv) input of a possible configuration file

In particular, this block manages the input and preprocessing of the curves in case a configuration file from a previous comparison is entered. In this case, the variables containing the information about the input curves and the preprocessing options are loaded into memory from the configuration file. Note that the input curves are always obtained from the original source (i.e., read from the data file) and preprocessed only thereafter according to the options retrieved from the given configuration file; this guarantees that any modification of the original files is always updated in the computation of the metrics. Also, calculating the metrics from the original source files when a previous configuration is loaded, guarantees the authenticity of the results in case they might be used to certify the verification or validation process for a numerical model. The block is divided into two main sub-blocks respectively responsible for (i) the selection of the various options and (ii) the initialization of the variables.

Block B (Input and Option selection) and Block C (Curve preparation) implement the procedure to input and, in particular, preprocess the curves. The details of how these operations have been implemented are described in the following section. Block B is divided into three sub-blocks, each of them respectively handles the following operation: (i) input and preprocessing, (ii) advanced preprocessing for multiple channels and (iii) selection of metrics. Similarly, Block C

is subdivided into two sub-blocks respectively accountable for (i) finalizing the preprocessing and (ii) plotting the curves.

Block D (Metrics evaluation) is the core of the code as it implements the algorithms to compute the various comparison metrics. It is divided into two sub-blocks which handle the computation of the metrics considering respectively (i) the whole domain on which the curves are defined or (ii) a sub interval which can be arbitrarily defined by the user.

The last block in which the code is organized (Output) implements all the necessary operations to output the results. This block is divided into three sub-blocks, each of them creates the following output information: (i) configuration file, (ii) Excel files and (iii) folder for the results.

The logical organization of the code in five blocks and the following sub-blocks makes easier to interpret the meaning of the various parts of the code. Moreover, as previously mentioned, the code has also been physically divided into various files according to a modular structure. The five blocks are located in the principal file of the code and the various tasks and sub-tasks are implemented in the scripts and functions located in the different files in which the code has been divided.

# Chapter 5

## Code Verification

The previous chapter provided a description of the characteristics of the software developed in this research and the principal details of how they have been implemented in the code. This chapter provides a description of the controls performed on the code in order to verify the correct implementation and the robustness of the algorithms used in the various tasks performed by the program during its execution.

The verification of the code was performed in two different phases each focusing on the correct implementation of two essential aspects for a proper comparison of the input curves: (i) the verification of the correct implementation of the metrics algorithms and (ii) the control of the code main structure. Consequently, this chapter is divided in two main sections respectively related to each of these two main verification aspects. In particular, the latter section describes the various verification controls performed on both the implemented input/output and preprocessing operations.

### 5.1 Verification of the metrics implementation

In order to verify the implementation of the metrics, it is necessary to compare the results computed using the implemented algorithms with a known solution. Unfortunately, in the literature, results obtained comparing some forms of waves are available only for a few of the comparison metrics implemented in this computer program. Hence, in order to have a known solution for all the metrics, it is decided to manually compute each of them using a spreadsheet.

Figure 5.1 shows the couple of curves used to perform this benchmark. They represent the lateral acceleration time histories collected during two similar crash tests and were prefiltered using a SAE 60 compliant filter. The vectors containing the data points of these curves were directly given as input to each of the Matlab functions which implement the metrics. In this manner, only the algorithms of the metrics were tested. The metric value obtained using either the Excel spreadsheet or the implemented Matlab functions are summarized in Table 5.1.

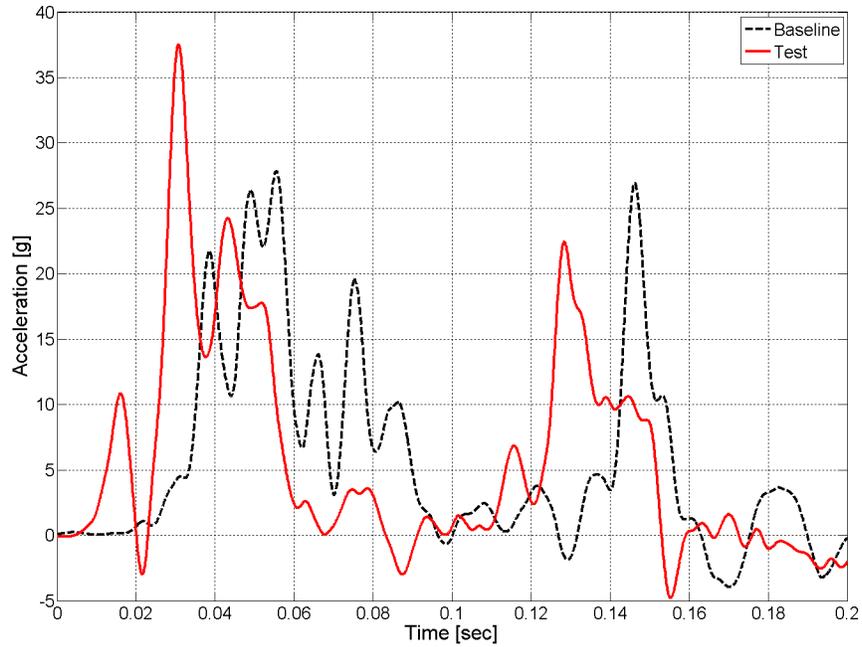


Figure 5.1: Time histories used for the verification of the metrics implementation.

Table 5.1. Metric values computed for the curves used to verify the metric implementation

<b>MPC Metrics</b>	Value [%]	<b>Single Value Metrics</b>	Value [%]
<i>Geers Magnitude</i>	4.9	<i>Whang's inequality metric</i>	51.2
<i>Geers Phase</i>	43.1	<i>Theil's inequality metric</i>	46.5
<i>Geers Comprehensive</i>	43.4	<i>Ziliacus error metric</i>	102.9
<i>Geers CSA Magnitude</i>	4.9	<i>RSS error metric metric</i>	95.2
<i>Geers CSA Phase</i>	43.1	<i>WIFac_Error</i>	70
<i>Geers CSA Comprehensive</i>	43.4	<i>Regression Coefficient</i>	0*
<i>Sprague-Geers Magnitude</i>	4.9	<i>Correlation Coefficient</i>	35.4
<i>Sprague-Geers Phase</i>	30.7	<i>Correlation Coefficient(NARD)</i>	56.9
<i>Sprague-Geers Comprehensive</i>	31.1	<b>ANOVA Metrics</b>	Value [%] (normalized to peak)
<i>Russell Magnitude</i>	4	<i>Average</i>	-0.2
<i>Russell Phase</i>	30.7	<i>Std</i>	32.8
<i>Russell Comprehensive</i>	27.5	*Imaginary number obtained	
<i>Knowles-Gear Magnitude</i>	56.6		
<i>Knowles-Gear Phase</i>	60.9		
<i>Knowles-Gear Comprehensive</i>	57.4		

The values computed using both the implemented Matlab functions and the Excel<sup>®</sup> spreadsheet were exactly the same for any of the metrics. The perfect match confirms the correct implementation of all the metrics algorithms into the code. Note that, because of a limitation in the domain in which the regression coefficient is defined, in this case, the value computed using both the two methods was equal to zero.

## **5.2 Verification of the code main frame**

This section illustrates the verification of the program main structure, which manages the various steps necessary to compute the metrics and, eventually output the results. In particular, the correct implementation of the following operations has been verified:

- Input of the curves
- Preprocessing
- Output of results

### **5.2.1 Input and storage of the curves**

Although the input of the curves may appear to be a trivial operation, it is essential to ensure that the selected files containing the data points of the curves are correctly read by the code; in particular, a crucial aspect of this verification is that the data points are appropriately saved in the corresponding variable. In fact, in order to be able to handle multiple channels, the data points corresponding to the various curves are stored in elaborate data structures which are created cycling over the different channels and, hence, it is important that all the input curves are stored in the appropriate location.

The correct implementation of the part of the code dedicated to the input of the couples of curves for the various channels was verified by manually plotting the time histories located in the variables used by the program to store the data points of the various input curves.

### **5.2.2 Preprocessing**

The following preprocessing operations were verified:

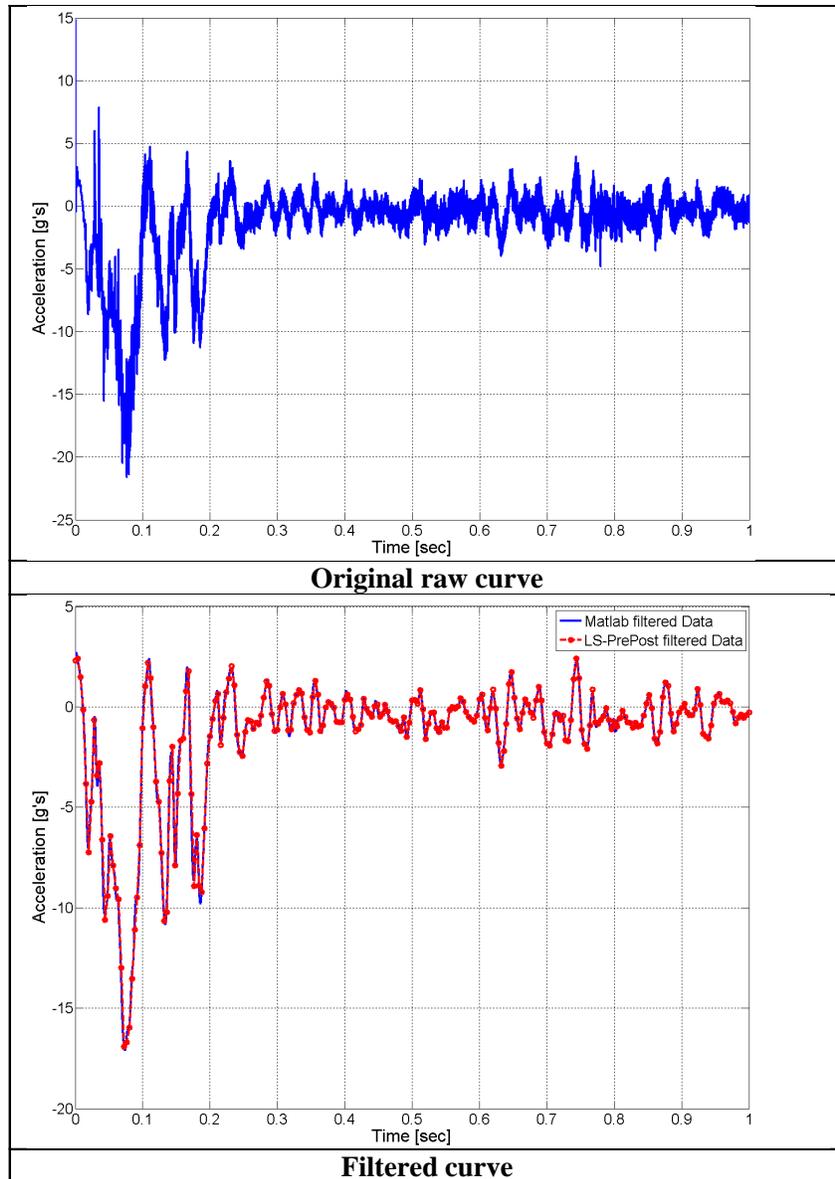
- Filtering
- Resampling/trimming
- Synchronization
- Shift/drift correction

The algorithms of some of these preprocessing operations were verified considering some simple analytical curves created by modifying on purpose the original baseline curve in order to add to it the specific disturbance/perturbation that the corresponding preprocessing operation was supposed to adjust. When the use of analytical curves was not possible, or to have a better investigation of the performance for a specific preprocessing algorithm, a couple of curves containing data measured from an experimental test in roadside safety were used to verify the implementation of the algorithm for some preprocessing operations. In both cases, the verification of each preprocessing algorithm was considered successful if it was capable to either recreate the originally defined analytical curve before it was manipulated or to appropriately modify the two given experimental curves.

#### **5.2.2.1 Filtering**

The verification of the code which manages the filter operation was conducted by filtering the same raw curve with both the implemented algorithm and a third-part program which used the same type of filter (i.e., SAE J211/1 compliant filter) and comparing the filtered curves obtained with the two programs. The external program used to filter the curve was LS/PrePost, which is the preprocessor of LS-DYNA [9]. In particular, as the program was specifically developed to compare numerical and experimental curves, a comparative test was performed considering a raw curve representing the acceleration time history of a node of a numerical model (i.e., a curve computed in a numerical simulation).

The original raw curve was filtered using a value the Channel Filter Channel (CFC) equal to 60 in both the implemented algorithm and LS/Prepost. Both the original and filtered curves are shown in Figure 5.2. As can be seen, the curve filtered with the implemented algorithm and that obtained using the commercial software LS/Prepost were smoother than the original unfiltered curve and looked exactly the same. This confirmed the correct implementation of the filter algorithm.

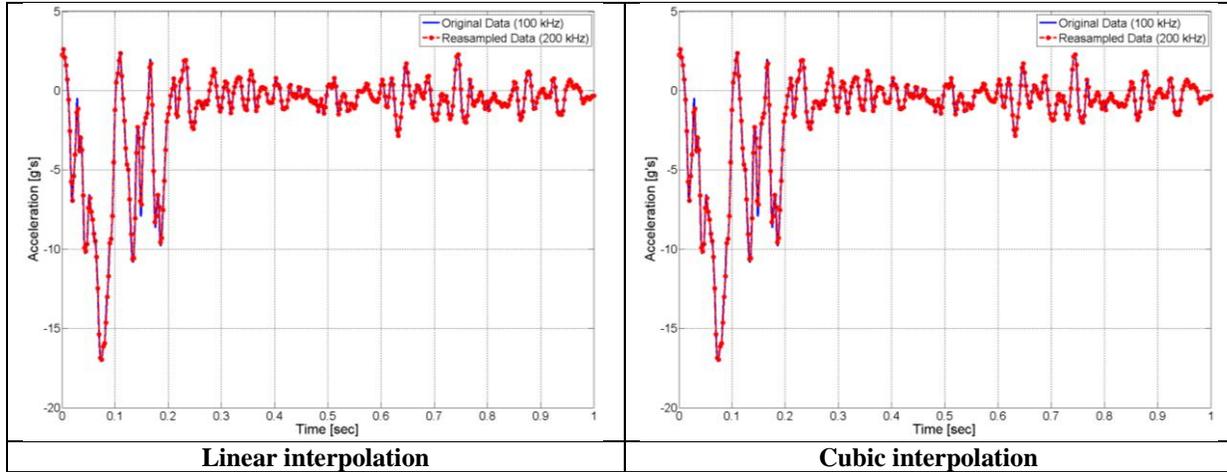


**Figure 5.2: Original curve (top) and curves filtered using both the implemented algorithm and the commercial program LS/Prepost (bottom).**

### **5.2.2.2 Resampling/Trimming**

The resampling subroutine is based on a linear interpolation. In order to check the accuracy of the re-sampled curves using this method, the filtered acceleration curve obtained in the previous section was re-sampled from its original sampling rate of 100 kHz (i.e., equivalent to a sampling period of  $1e-5$  sec) to a new sampling rate of 200 kHz which was equivalent to a new sampling period of  $0.5e-5$  sec). In order to investigate the potential improvement which could be achieved, also a cubic interpolation was considered for the same curve. From the comparison

shown in Figure 5.3, it is possible to see how close the original and the re-sampled curves were in both cases of a linear or a cubic interpolation. In particular, the use of a linear interpolation proved to be optimal for the resampling of the curves.



**Figure 5.3. Comparison of an acceleration curve and the equivalent resampled curve using either a linear (left) or a cubic (right) interpolation.**

As for the trimming option, to verify the correct implementation of this part of the resampling algorithm which ensures the same final length for the input curves, the filtered acceleration curve already used in the previous two tests was manually trimmed at 0.7 sec. The length of the original and shorter curve therefore obtained was then compared. This verification confirmed that the implemented algorithm was able to correctly reduce the original curve to the same length of the shorter curve (i.e. the final curve length).

### 5.2.2.3 Synchronization

In order to verify the correct implementation of the corresponding algorithm, the synchronization procedure was applied to two identical analytical sine waves which were simply shifted respect to each other along the time direction.

The ideal analytical wave form for the baseline curve was given by the following decayed sinusoidal curve:

$$m(t) = e^{-t} \sin 2\pi(t - \tau) \quad (5.1)$$

where the parameter  $\tau$  was used to create a phase in time or “time of arrival”.

Two different tests were performed considering either a positive or negative bias, computing the phased curves respectively as:

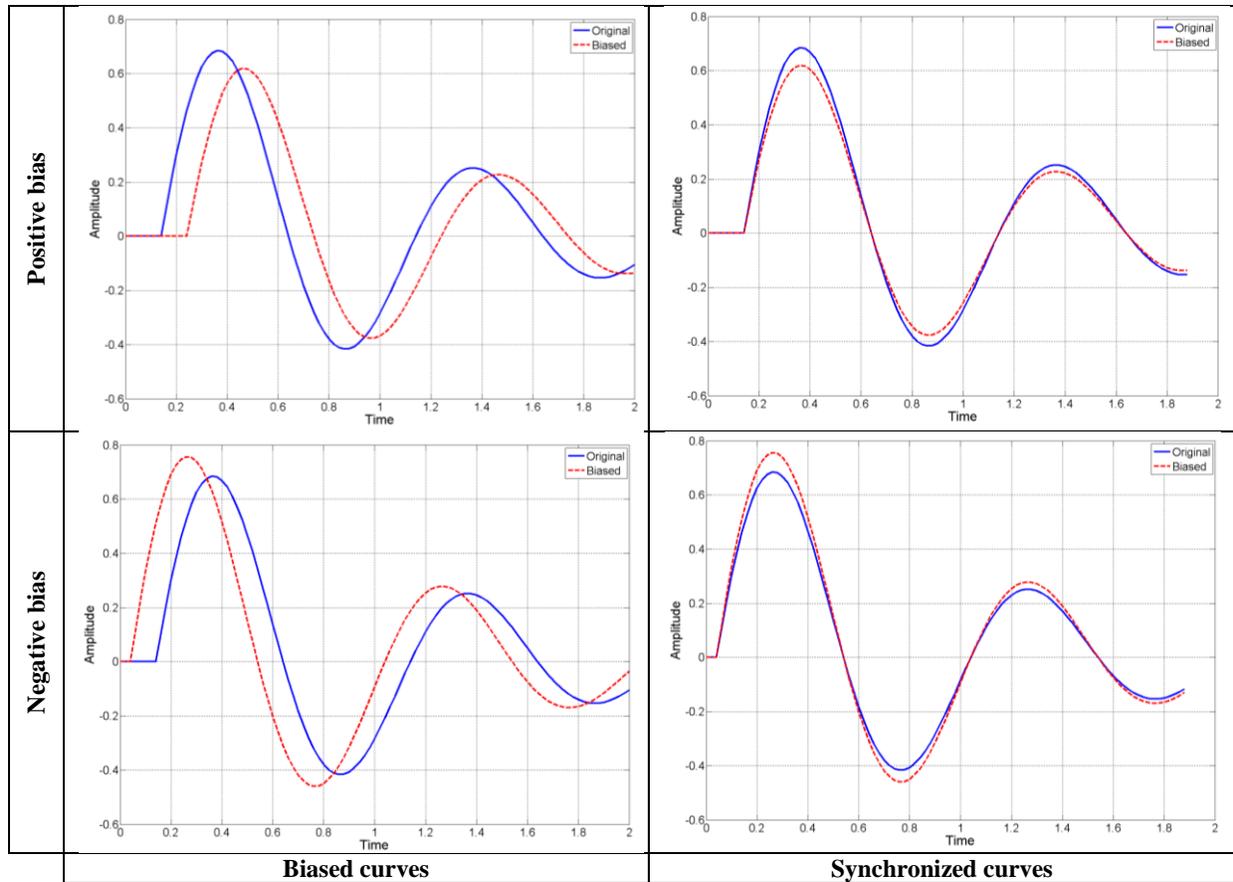
$$\begin{cases} m(t) = e^{-(t-0.14)} \sin 2\pi(t-0.14) \\ c(t) = e^{-(t-0.04)} \sin 2\pi(t-0.04) \end{cases} \quad (5.2)$$

and,

$$\begin{cases} m(t) = e^{-(t-0.14)} \sin 2\pi(t-0.14) \\ c(t) = e^{-(t-0.24)} \sin 2\pi(t-0.24) \end{cases} \quad (5.3)$$

In both cases, the sampling period was  $\Delta t = 0.02 \text{sec}$  and  $0 \text{sec} \leq t \leq 2 \text{sec}$ . Figure 5.4 shows both the originally biased curves and the results obtained after the synchronization process was applied. Note that the couples of analytical curves were defined in order to be not only biased respect to each other, but also to present a different magnitude at the corresponding peaks. In fact, real curves often present simultaneously both these two characteristics and, hence, it was paramount to verify that the algorithm could handle both of them.

After being synchronized, the curves showed a good match for both the case with a positive or negative bias, thus confirming the correct implementation of the synchronization algorithm. It is worth noting that, the algorithm also proved to correctly trim the curves according to the specific phase correction.



**Figure 5.4: Original biased waves (top) and synchronized waves (bottom).**

Moreover, in order to verify the robustness of the implemented algorithm for the synchronization in case of real curves where a random scatter of the results may appear, the time histories representing the lateral accelerations measured at the center of gravity of two vehicles during similar full-scale crash tests were synchronized. In fact, although the two experimental tests were performed using the same type of vehicle (brand, model and year) and barrier and under the same initial conditions, the two original curves not only were biased respect to each other, but also presented some differences.

Figure 5.5 shows the couple of experimental curves before and after being synchronized. In the original pair of curves, the first peak was not matching exactly while, after applying the synchronization, a better match between the curves occurred.

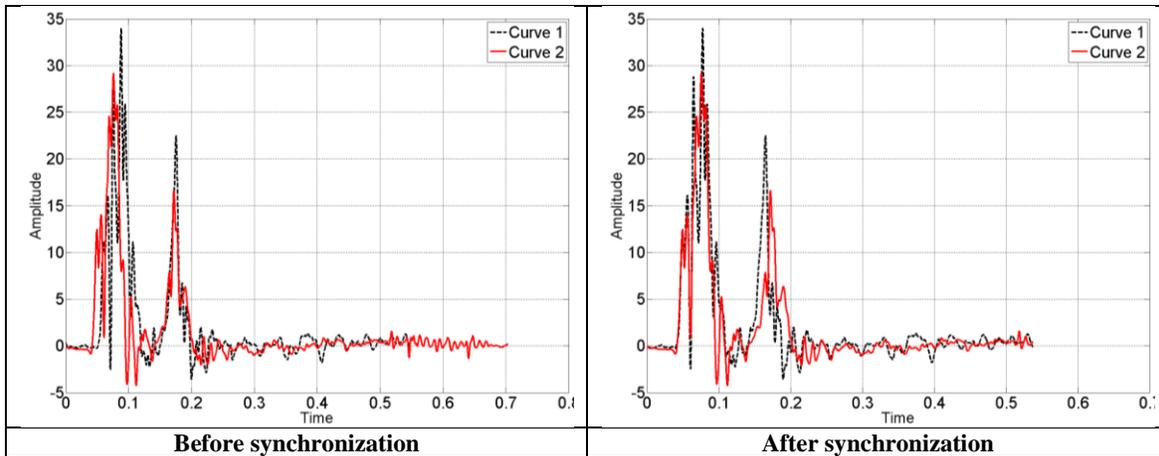


Figure 5.5: Experimental curves before (top) and after (bottom) being synchronized.

#### 5.2.2.4 Shift/Drift correction

In order to verify the implementation and the robustness of the algorithm for the correction of the shift and/or drift effect, two simple analytical cases were considered, respectively characterized by shift only and both shift and drift effect. In each of the two cases, the curves were modified starting from the following baseline curve:

$$\begin{cases} y_0(t) = 0 & t \leq 0.14 \\ y_0(t) = 0.5 e^{-(t-0.14)} \sin 2\pi(t-0.14) & t > 0.14 \end{cases} \quad (5.4)$$

The shift effect was created by simply translating along the vertical axis the original baseline curve by a specific value, which in this case was 0.1; while the drift effect was obtained by subtracting point-by-point a linear straight line from the previously shifted curve:

$$y_{shifted}(t) = 0.1 + y_0(t) \quad (5.5)$$

$$y_{shifted\&drifted}(t) = y_{shifted}(t) - 0.05t \quad (5.6)$$

The original baseline and the shifted and shifted/drifted curves are so computed shown in the top of Figure 5.6.

Both the baseline and the modified curves were preprocessed and, in any of the two cases, the shift/drift effect was completely reduced, thus confirming the correct implementation of the algorithm for this preprocessing feature. The curves obtained after the preprocessing are shown in the bottom of Figure 5.6. As can be seen, the algorithm slightly translated also the baseline

curve along the vertical direction, but this could be considered negligible respect to the initial shift of the modified curves.

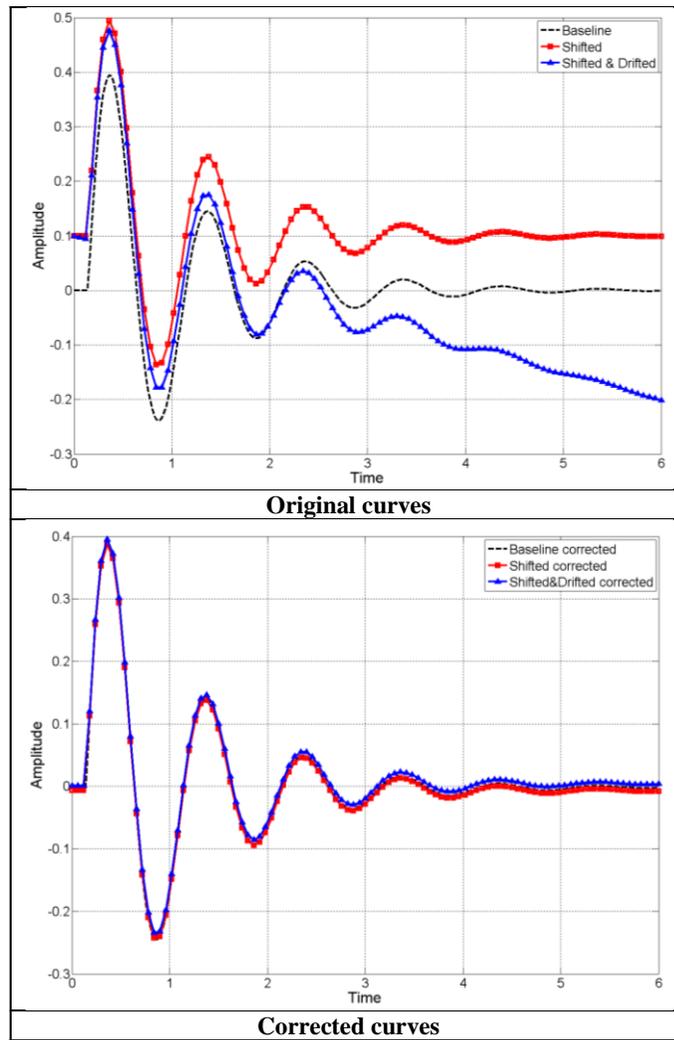


Figure 5.6: Baseline and shifted/drifted curves before (top) and after (bottom) the preprocessing.

Similarly to the previously described verification of the correct storage for the original curves, also for the preprocessed curves it was verified that the data points were correctly saved into the corresponding data structure. In this case, as after being preprocessed the couples of curves from each channel are characterized by the same number of points, the variable in which the data are stored is simply a three-dimensional matrix in which the third dimension corresponds to the specific channel. It is important to verify that the data retrieved during the computation of the metrics correspond to the desired curve and they have not been misplaced in the data matrix.

### 5.2.3 Computation of metrics using multiple channels

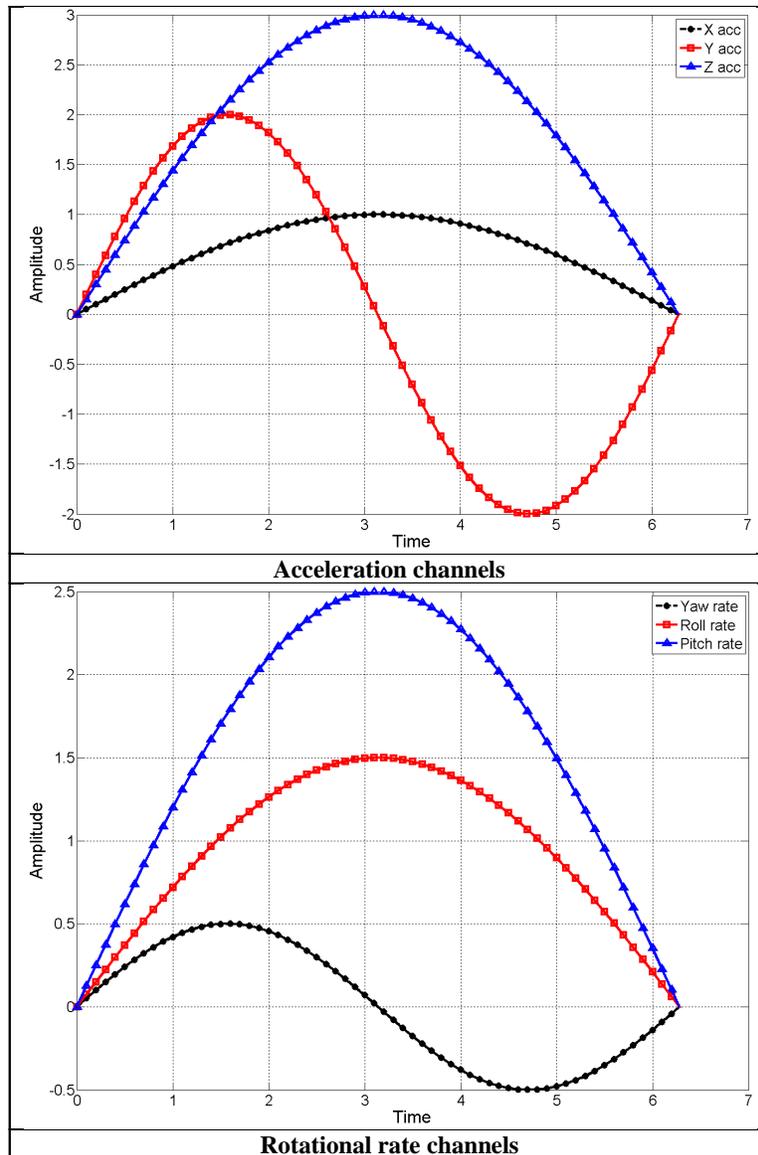
As described in the previous chapter, in case multiple channels are input the metrics corresponding to each specific channel are first computed and then a weighted average is calculated. In the following, the verification of the part of the code related to the computation of the metrics using multiple channels is described.

#### 5.2.3.1 Weighting factors

The first code verification for the multichannel computation regarded the correct implementation of the algorithm used to calculate the weighting factors. In particular, the implementation of each of the three proposed methods was verified. The weighting factors corresponding to six sinusoidal curves were computed both by using the Matlab scripts implemented into the computer program and an Excel<sup>®</sup> spreadsheet. The analytical definition and other important characteristics of the curves are shown in Table 5.2, while the plots of the curves are shown in Figure 5.7.

**Table 5.2. Characteristics of the curves used for the code verification of the weighting factors**

	<b>X acc</b>	<b>Y acc</b>	<b>Z acc</b>	<b>Yaw rate</b>	<b>Roll rate</b>	<b>Pitch rate</b>
<b>Curve definition</b>	$\sin(x)$	$2 \sin\left(\frac{x}{2}\right)$	$3 \sin(x)$	$0.5 \sin\left(\frac{x}{2}\right)$	$1.5 \sin(x)$	$2.5 \sin(x)$
<b>Peak value</b>	1	2	3	0.5	1.5	2.5
<b>Area value</b>	4	0	12	0	6	10
<b>Area (till peak)</b>	2	2	6			



**Figure 5.7: Plots of the curves used for the verification of the weighting factor implementation.**

The choice of simple analytical curves allowed to easily control and define the curves characteristics such as the peak values and the area. In particular, given their known analytical formulation it was possible to precisely calculate the area of each of the curves.

The values of the weighting factors computed with the Excel<sup>®</sup> spreadsheet confirmed the correct implementation of the Matlab scripts used in the computer program. The results obtained for each of the three suggested methods are shown in Table 5.3. As expected, the sum of the weighting factors was equal to the unity in any of the considered cases.

**Table 5.3. Values of the weighting factors computed for each of the three proposed methods**

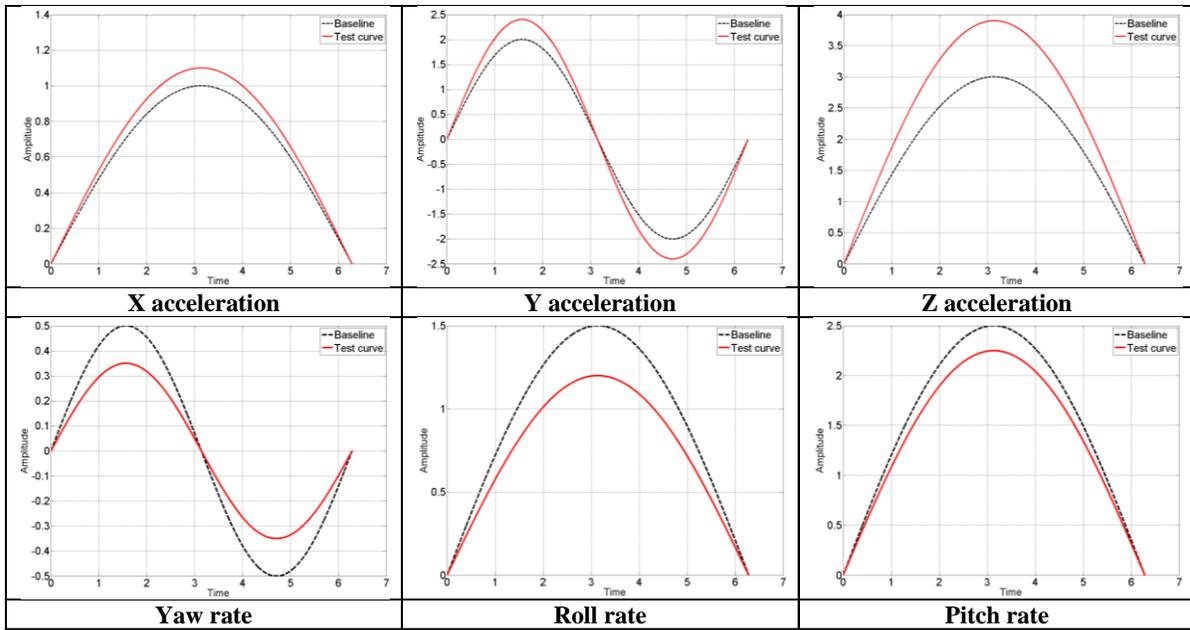
	<b>X acc</b>	<b>Y acc</b>	<b>Z acc</b>	<b>Yaw rate</b>	<b>Roll rate</b>	<b>Pitch rate</b>
<b>Peak method</b>	0.088	0.175	0.263	0.053	0.158	0.263
<b>Area methods</b>						
Area I	0.114	0.000	0.341	0.000	0.205	0.341
Area II	0.125	0.000	0.375	0.000	0.188	0.313
<b>Momentum method</b>	0.086	0.086	0.258	0.095	0.060	0.415

### 5.2.3.2 *Weighted average*

The computation of the weighted average of the metrics from each input channel is a complex process which involves various loops considering only effectively input channels and the storage of data in multiple variables characterized by somehow elaborate data structures.

In order to verify the correct implementation of this part of the code, the results automatically obtained with the multichannel option of the program were compared with those computed by considering the metrics of each single channel and the corresponding weighting factors. The same curves used in the previous section were adopted as baseline for each respective channel and the other curves of each pair were created multiplying the corresponding baseline by a scale factor equal. In particular, the three acceleration curves were multiplied by respectively a scale factor equal to 1.1, 1.2 and 1.3, while the factors used for the rotational rates were respectively 0.7, 0.8 and 0.9.

The pairs of curves obtained for each of the three acceleration and rotational rate channels are shown in Figure 5.8, while the weighted metric values are shown in Table 5.4. The results computed with the spreadsheet calculations matched exactly those obtained using the implemented computer program, thus confirming the correct implementation of this part of the code as well.



**Figure 5.8: Pairs of curves used in the code verification of the implemented algorithm for the weighted average.**

**Table 5.4. Channels and weighted metric values obtained from the verification of the multichannel calculations**

	<b>X acc</b>	<b>Y acc</b>	<b>Z acc</b>	<b>Yaw rate</b>	<b>Roll rate</b>	<b>Pitch rate</b>	<b>Weighted average</b>
<b>MPC Metrics</b>	<b>Value [%]</b>	<b>Value [%]</b>					
Geers Magnitude	10	20	30	30	20	10	19.4
Geers Phase	0	0	0	0	0	0	0
Geers Comp.	10	20	30	30	20	10	19.4
Geers CSA Magnitude	10	20	30	30	20	10	19.4
Geers CSA Phase	0	0	0	0	0	0	0
Geers CSA Comp.	10	20	30	30	20	10	19.4
Sprague-Geers Magnitude	10	20	30	30	20	10	19.4
Sprague-Geers Phase	0	0	0	0	0	0	0
Sprague-Geers Comp.	10	20	30	30	20	10	19.4
Russell Magnitude	7.6	13.6	18.5	-23.8	-16.1	-8.3	2.3
Russell Phase	0	0	0	0	0	0	0
Russell Comp.	6.7	12	16.4	21.1	14.3	7.4	12
Knowles-Gear Magnitude	10	20	30	30	20	10	19.4
Knowles-Gear Phase	0	0	0	0	0	0	0
Knowles-Gear Comp.	9.1	18.3	27.4	27.4	18.3	9.1	17.7
<b>Single Value Metrics</b>							
Whang's inequality metric	4.8	9.1	13	17.6	11.1	5.3	9.2
Theil's inequality metric	4.8	9.1	13	17.6	11.1	5.3	9.2
Zilliacus error metric	10	20	30	30	20	10	19.4
RSS error metric metric	10	20	30	30	20	10	19.4
WIFac_Error	9.1	16.7	23.1	30	20	10	16.7
Regression Coef.	97.3	98	72.5	95.4	88.8	97.3	86.4
Correlation Coef.	100	100	100	100	100	100	100
Correlation Coef.	100	100	100	100	100	100	100
<b>ANOVA Metrics</b>							
Average	6.37	0	19.1	0	-12.73	-6.37	3.6
Standard deviation	3.08	14.14	9.24	21.21	6.16	3.08	6

#### 5.2.4 Output of results

The proper implementation of the algorithms for the output of the results is another critical aspect of the program, as the user will eventually rely on these results to compare the couple/s of input curves.

The output of the results is provided in two different ways: (i) through tables shown on the screen during the execution of the program and (ii) saved as tables in a spreadsheet file (in Excel<sup>®</sup> format) at the end of the execution.

In both these two cases, the metrics scores are obtained after some postprocessing steps which extract and round the final value of the metrics from the corresponding time histories. Also, in case, the same procedure is applied for any of the intervals defined by the user, leading to the creation of elaborate variables.

The analysis of the results obtained in the previous comparison of the six couples of analytical curves was used as a confirmation of the correct implementation of the output of the results as well. In particular, the following aspects were verified:

- Output of results for each of the single channels
- Output of results for the weighted average of the multiple channels
- Consistent output of screen and Excel<sup>®</sup> tables

The match with the spreadsheet calculations confirms the correct implementation of the whole procedure to output the results.

## **Chapter 6**

### **Acceptance Criteria for Roadside Safety**

This chapter describes the method followed in the research to identify the comparison metrics and the corresponding acceptance criteria which are most appropriate for the purpose of the verification and validation process of numerical models in the field of roadside safety.

When comparing two curves during the validation or verification process of a numerical model it is important to consider the level of repeatability of the experimental data in that particular field of application. In other words, the criteria used to assess if the level of agreement between two curves is acceptable or not should be related to the scatter of data which characterizes the experimental tests in that particular field.

In fact, in fields where experimental tests performed under the same conditions give very close results, it should be expected that also the correlation between the outcomes of an experimental test and a numerical simulation is at a similar level of agreement. On the contrary, if experimental results obtained under the same conditions are characterized by a considerable level of scatter, assuming that the experimental outcomes can be considered all equivalent, also the expected level of agreement between numerical and experimental results should be less tight in this case. The reason for a greater tolerance on the acceptance criteria is implied in the fact that the same numerical results may compare fair or poor with one test and good or, even excellent, with another test only because the experimental results themselves are characterized by a sensible level of scatter.

In roadside safety, typically experiments involve full-scale crash tests. Because of the hundreds, if not thousands, of parameters involved in such a complex event like a crash between a vehicle and a restraint system, the experimental outcomes may be characterized by a considerable level of scatter respect to other fields. In order to assess acceptance criteria which can reasonably reflect the typical scatter of data of full-scale tests in roadside safety, a series of repeated full-scale crash tests have been analyzed.

In fact, assuming that these experimental tests could be considered essentially identical, the variation of the metric values obtained comparing the curves can be eventually used to propose acceptance criteria. These criteria would be based on the typical scatter of data that characterizes full-scale tests in roadside safety.

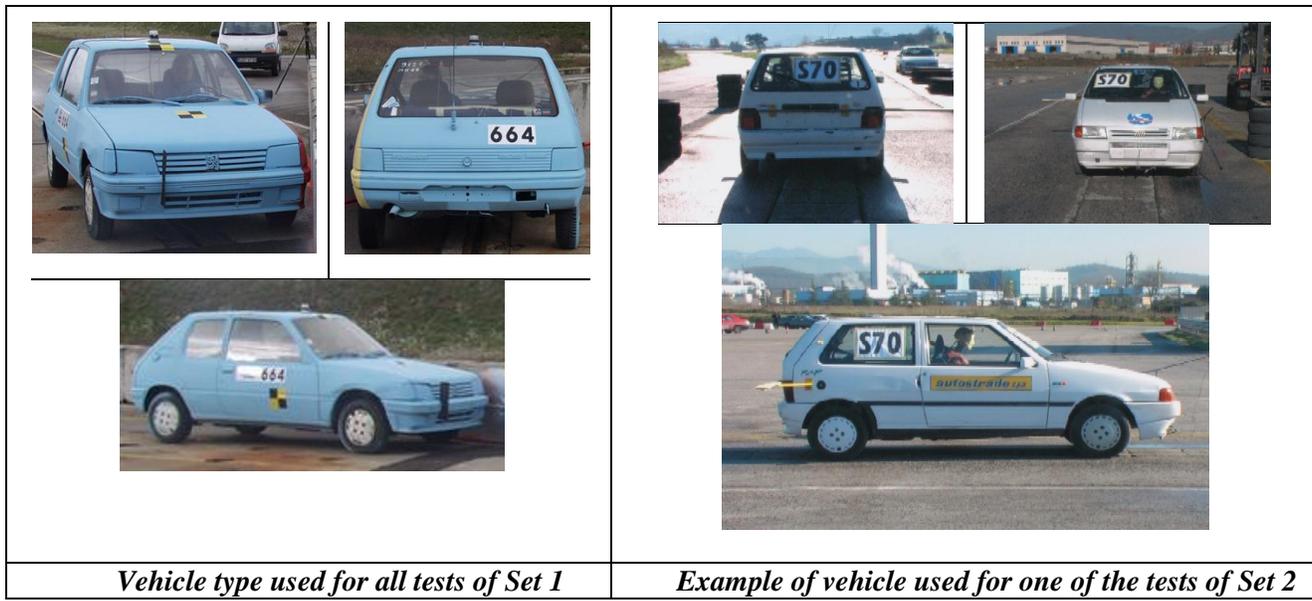
In particular, ten distinct full-scale crash tests performed using the same barrier and impact conditions have been analyzed. The tests have been subdivided into two series of five experiments each; the first series (Set 1) consisting of five crash tests with new vehicles of exactly the same brand and model and the second series (Set 2) including other five tests performed with various used vehicles of a category equivalent to the vehicles employed in the first set.

The following sections describe how the original curves are first preprocessed and the results then obtained from their comparative analysis. Based on the analysis of the results obtained from the comparisons of the curves, the most appropriate metrics for the application in roadside safety and the corresponding acceptance criteria are proposed at the end of this chapter.

## **6.1 Description of full-scale crash tests**

This section illustrates the experimental setup for the series of repeated tests considered in this comparative study. The tests involved a 900-kg small car impacting a rigid concrete barrier with an initial impact velocity and angle of respectively 100 km/h (62 mph) and 20 degrees.

All the ten tests were performed as a part of the ROBUST project [66] and were independently carried out by five different test laboratories in Europe, herein called laboratories one through five with the original purpose of assessing the repeatability of crash tests. As the main intent was to see if experimental curves obtained from identical or similar tests resulted in similar responses, a rigid barrier was intentionally chosen in order to limit the scatter of the results, which is typically greater in the case of deformable barriers. In particular, the second set of tests was performed to investigate influences arising from the use of similar but not identical vehicle models on the repeatability of crash tests.



**Figure 6.1: Vehicle model used in the first set of crash tests (left) and a similar vehicle used in one of the tests of the second set (right) [66].**

Any of the vehicles used in both the two series of tests complied with the specifications for a small car vehicle as required by the European crash test standards, EN 1317 [33]. As previously mentioned, the same vehicle model was used in any of the tests for the first set and various but similar vehicle models were used in the second set of tests. Figure 6.1 shows both one of the *Peugeot 106* cars used in the first set of tests and a *Fiat UNO* car used for one of the tests in the second set. As can be seen the two vehicles fall in the same category. In all the tests, the three components of the acceleration and the rotational rates were measured at the center of gravity of the vehicles.

The barriers used in all the tests were composed by simple concrete blocks rigidly interconnected to each other (Figure 2.3). This restraint system is produced by the British company Marshall RC Products Company and is usually employed as a temporary modular lane divider. In particular, it is constructed with independent, precast concrete units each having a mass of 2,600 kg (5,732 lb) and being 3.15 m (10.3 ft) long, 0.45 m (1.5 ft) wide and 0.816 m (2.7 ft) high. During the tests, the modules were placed end to end on the track and were connected by two bolts (M24) passing through holes in the vertical scarf-joints at the ends of the units. To prevent damage to the joints, a non-cellular rubber gasket (5 mm thick) was placed in the gap between the ends of the units. The total length of the barrier installation for all the ten tests was about 30 m (100 ft).

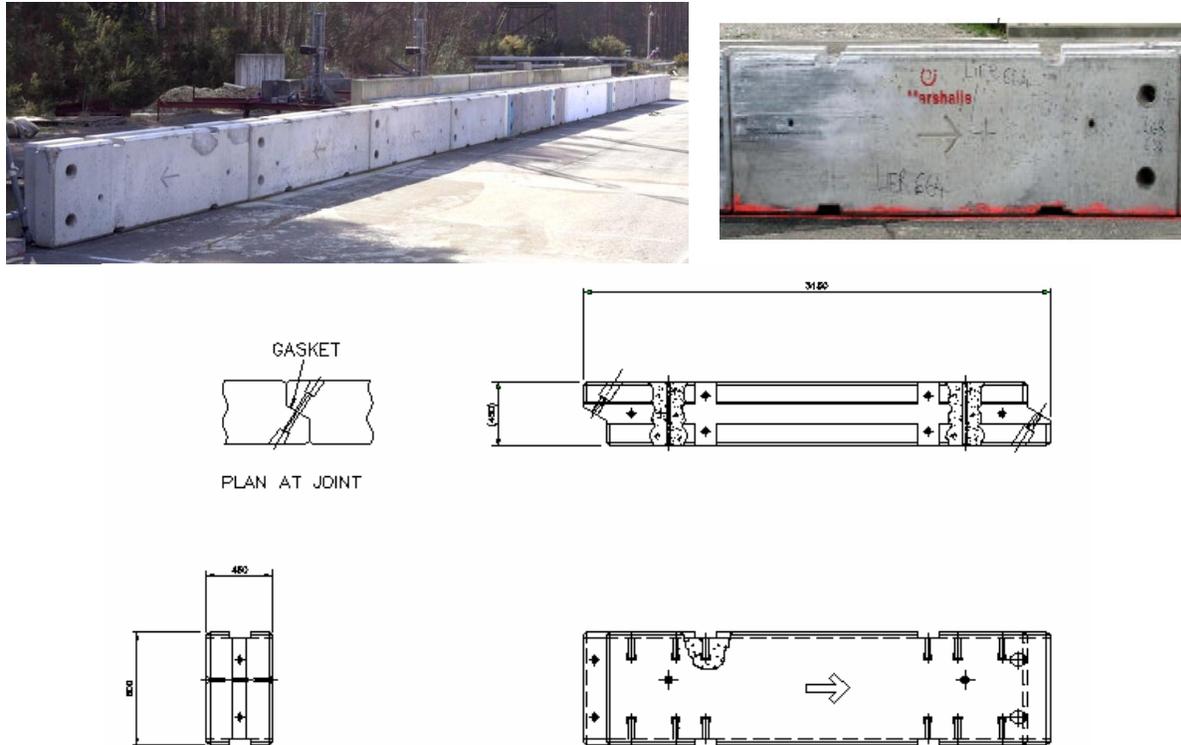


Figure 6.2: Temporary vertical concrete barrier used in the repeated full-scale crash tests.

## 6.2 Analysis of the repeated tests

As previously mentioned, for each test four different channels were collected: the three components of the acceleration (measured along the local coordinate system of an accelerometer moving with the center of mass of the vehicles) and the yaw rate of the vehicle at its center of mass. Considering the particular nature of the barrier, which is rigid, and the initial impact angle ( $20\text{ deg}$ ), it is decided to focus the analysis primarily on the lateral acceleration time histories (i.e., the acceleration along the local Y axis of the accelerometers). In fact, in this type of collision, the lateral acceleration is generally the best measure of the stiffness of the barrier response and is most directly related to the measurement of the occupant impact severity.

Acceleration time histories are used rather than velocity or displacement time histories because the accelerations are the experimentally observed quantities whereas the velocity and displacement are back-calculated by integrating and double integrating the acceleration response, respectively. Anyhow, in order to investigate the effect of using derived time histories instead of the data originally collected during the tests, also a comparison involving velocity time histories obtained from the integration of the original accelerations is analyzed.

The residuals between two time histories are by definition the difference at each instant in time between two curves, one of which is considered to be the baseline or “correct” curve. Hence, it is necessary to define a baseline curve for each of the two sets of tests which could be used for the calculation of the residuals. In this research two different approaches are considered, consisting respectively of using as the baseline curve either the average response computed from the curves of each set of tests or the combination of both, or one curve from each set which is considered to be the “correct” one. In the latter case, considering the lateral accelerations, for each of the two sets of tests the baseline curve is chosen as the time history which is the closest to the average response among all the curves of the respective set (i.e., tests Lab#1 (Set 1) and Lab#1 (Set 2)).

### **6.2.1 Preprocessing of curves**

Before being able to compare the data from the various experimental tests, it is necessary to preprocess the original time histories. In fact, without a proper preprocessing, the curves would have been otherwise characterized by noise, different sampling rates, units and sign, diverse lengths, etc. These dissimilarities between the curves would either make impossible to compare them or, even in the case the curves can still be compared, they could lead to misleading results. Furthermore, the original curves from the various tests might probably be shifted in time respect to each other as the impact point did not happen at the same time in each test, thus requiring to synchronize them.

In order to make the curves comparable, the following preprocessing operations are applied:

- Filtering
- Shift correction
- Resampling
- Synchronization
- Trimming

The preprocessing, as also the subsequent comparison of the curves, is completely performed using Matlab.

### **6.2.1.1 Filtering**

The first step of the preprocessing operation consisted in digital filtering them using a SAE Class 60 filter algorithm [64]. This operation is deemed necessary to smooth the original curves by eliminating the noise component collected by the acceleration transducers during the acquisition of the data points. After the filtering process, for some curves it is necessary to change the original units and/or sign by using appropriate scale factors. In particular, as for the sign, although the accelerometers are placed in the vehicle's center of gravity in all the tests, the lateral axes were not always defined in the same direction hence requiring that their sign be inverted in some cases.

### **6.2.1.2 Shift correction**

After the time histories of each test are filtered, they are also shifted down by a value equal to the average of the first ten data points of the respective curve in order to correct any shift effect during the collection of the data.

### **6.2.1.3 Resampling**

Residuals are, by definition, the difference at each sample point between two curves and hence, in order to be able to compute them, it is necessary that any of the time histories have the data collected at the same sampling rate. As the original sampling rate is not the same for all the time histories, it is then necessary to resample each curve to a common rate, which is chosen to be the highest sampling rate among all the tests (i.e., 20 kHz).

### **6.2.1.4 Synchronization**

Another important step in the preprocessing of the curves consisted in synchronizing them in order to ensure that all the experimental tests started at an equivalent instant of the event. Specifically, the match of the impact point of the time histories from the ten experimental tests is based on the minimization of the area of the residuals between the lateral acceleration for each particular test and a baseline curve, which is the same for the synchronization process of any test. It is decided to use the curve of the lateral acceleration from Lab# 5 of the first set of tests (Set 1).

As discussed in Chapter 4, the implemented synchronization procedure based on a minimization procedure should guarantee the most probable matching between each curve and the selected baseline curve.

After the shift values which optimize the synchronization of the curves between each test have been calculated using the minimization algorithm, all the curves are then shifted respect to each other. The optimal shift value may be different for the curves of each specific test; hence, in order to achieve a consistent synchronization respect to the same baseline curve, it is necessary to consider the differences between the various shifts. The diagram of the procedure used to synchronize consistently all the ten curves from a specific channel is shown in Figure 6.3. The final value of the shift for each test is assessed based to the maximum value of all the shifts,  $S_{max}$  (note that, if all the shift values are negative,  $S_{max}$  would be assigned a value equal to zero). As can be seen from the flowchart in Figure 6.3, the final value of the shift for each test is computed as the difference between  $S_{max}$  and the shift for the specific test if the latter is positive and the sum of  $S_{max}$ , and the absolute value of the specific shift in case the latter is negative.

In fact, the function used to shift the curves respect to each other is implemented in a way that a positive value of  $S$  means a forward translation of the test curve respect to the baseline curve respect to which the synchronization is performed). Also, the head of the baseline curve is then trimmed by a value corresponding to the shift of the test curve. As in the worst case, the head of the baseline curve is trimmed by the value  $S_{max}$ , it is necessary to take into account the difference between the shift of the particular test curve and this maximum translation. As shown in the diagram of Figure 6.4, this difference can be balanced by cutting the test curve of a further value equal to the difference  $S_{max} - S_i$ . On the contrary, in case the shift value is negative, the test curve is shifted backward and its head is trimmed by an equivalent value as well. Hence, for negative values of the shift, the difference is balanced by further trimming the head of the test curve by the value  $S_{max}$  (i.e., the total trimming is equal to  $S_{max} + |S_i|$ ). Note that, in the particular case all the curves have a negative shift, each of them is simply shifted backward (and trimmed) by the corresponding shift value,  $S_i$  as shown in Figure 6.5.

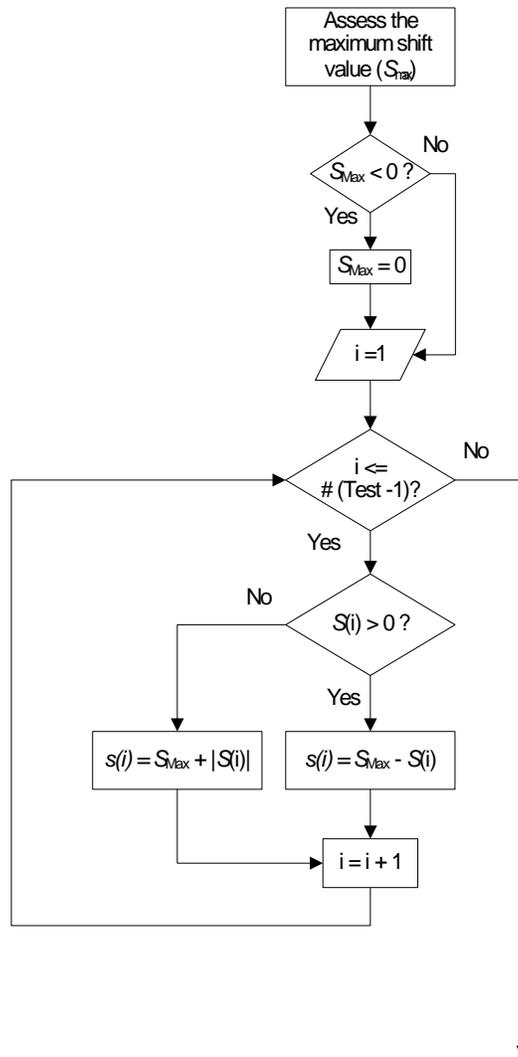
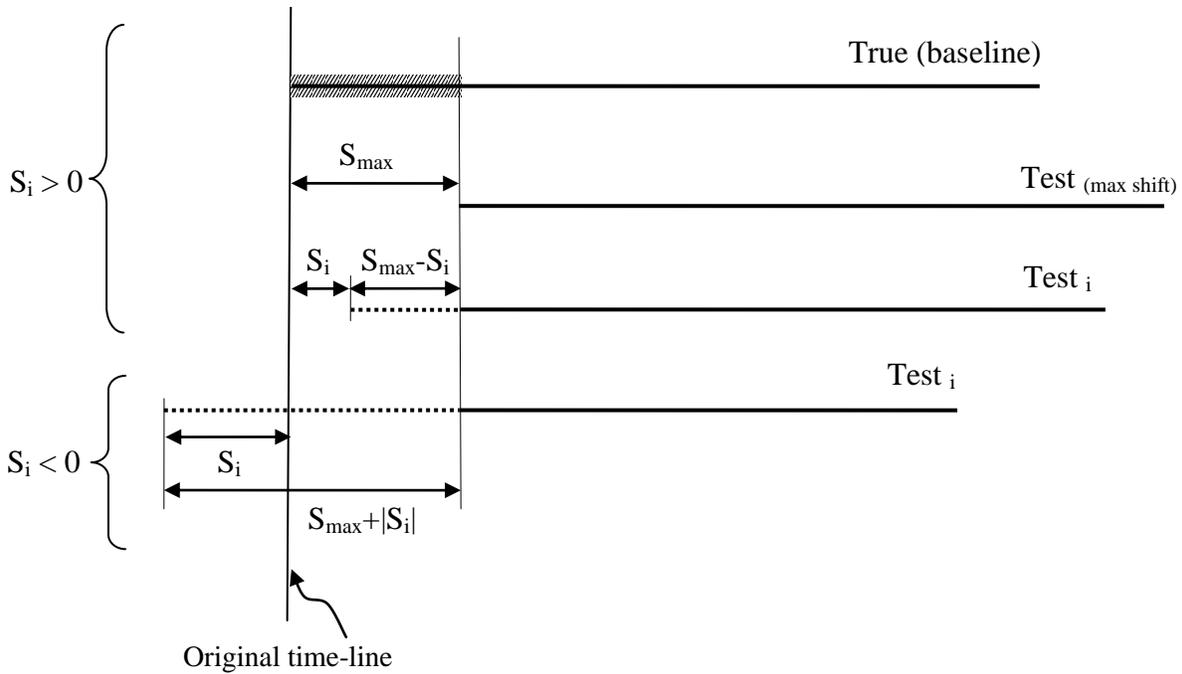
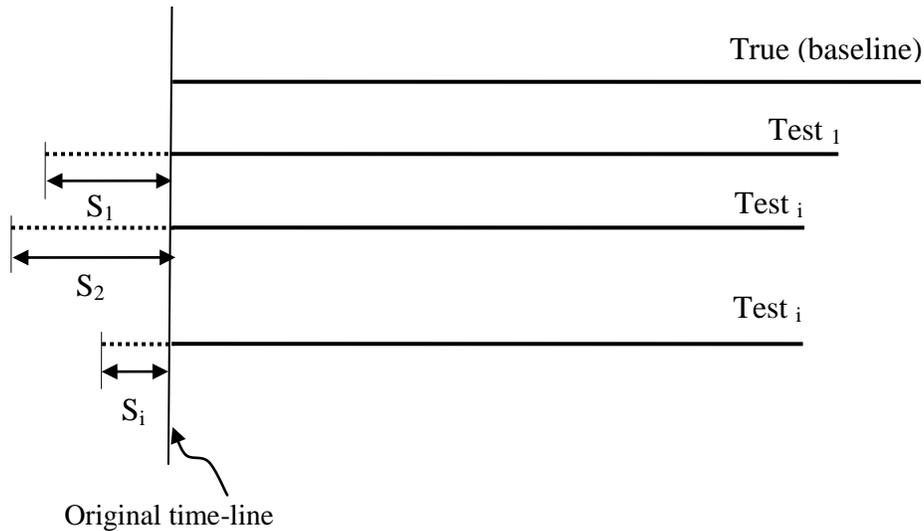


Figure 6.3: Flow chart of the procedure for the assessment of the final shift values  $s(i)$  for each test.



**Figure 6.4: Graphical representation of the shift routine.**



**Figure 6.5: Graphical representation of the shift routine for the case of negative shift values  $s(i)$  of all the tests.**

Eventually, the curves are synchronized by translating them according to the final shift values obtained using the previously described procedure. Figure 6.6 shows the algorithm used to translate the curves of each test. Basically, three main steps are followed to synchronize the channels of all the ten tests:

- 1) Creation of a shifted time vector for the curves of each test

- 2) Interpolation of the curves using linear interpolating polynomials calculated over the shifted time vectors
- 3) Shifting of the interpolated curves to the origin (shift time vector)

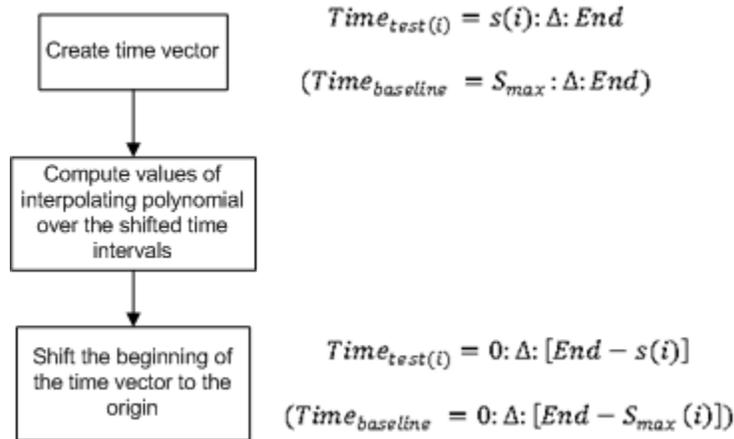


Figure 6.6: Flow chart of the procedure for the translation of the curves of the various tests.

### 6.2.1.5 *Trimming*

The last step of the preprocessing consisted in trimming the curves from all the tests in order to have them characterized by the same final length.

## 6.2.2 **Statistical analysis of the residuals**

Once the curves from the ten experimental tests have been preprocessed, it is possible to perform a statistical analysis of the residuals in order to investigate the behavior of the typical errors that can be expected in full-scale roadside safety crash tests. For this purpose, the 90<sup>th</sup> percentile interval of confidence is created based on two different approaches. Moreover, the distribution and the values of the mean and standard deviation of the residual time histories are analyzed.

### 6.2.2.1 *90th percentile confidence envelope*

The first step in the analysis of the residual consisted in the evaluation of the corridor corresponding to the 90<sup>th</sup> percentile interval of confidence (i.e., the envelope in which fall at least 90 percent of the curves). The width of the corridor is computed based on either the values of the curves or the residuals, which are the difference of the values between two curves at each time point. In particular, the two different approaches considered are respectively based on: (i) values of all the available curves calculated at each sampling point or (ii) time history of the

residuals between each curve and the baseline curve. In the following two sections, the method and assumption made for each of these two approaches are described in details.

### 6.2.2.1.1 Statistical analysis of values at each sampling point

This first approach to compute the confidence interval is based on the statistical analysis of the values of the time histories at each time point. Two steps are followed at each sample point:

1. Computation of the average of the values from each curve
2. Computation of the standard deviation of the values from each curve

Once the standard deviation is computed for each time point, the 90<sup>th</sup> percentile interval is created by adding and summing to the average curve of the specific set the previously calculated time history of the standard deviation multiplied by the factor 1.6449 (i.e., the 90<sup>th</sup> percentile for a normal distribution).

Figure 6.7 and Figure 6.8 show the curves and the corresponding confidence intervals for respectively each of the two sets of tests and their combination.

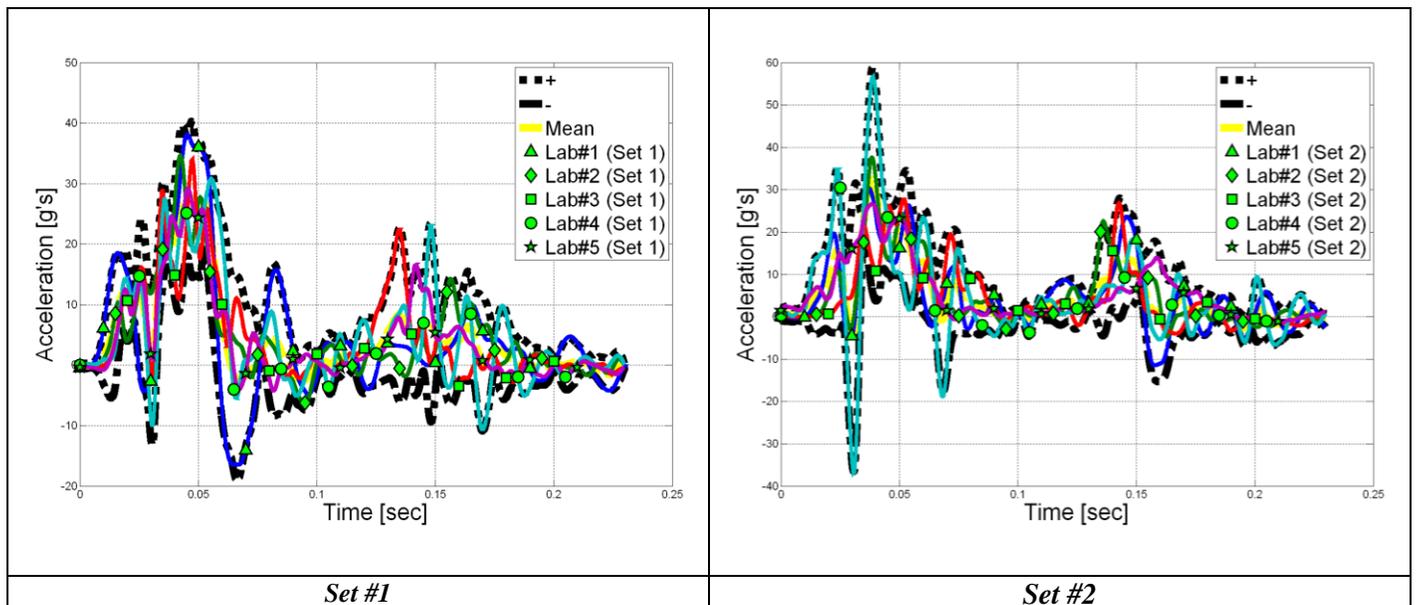
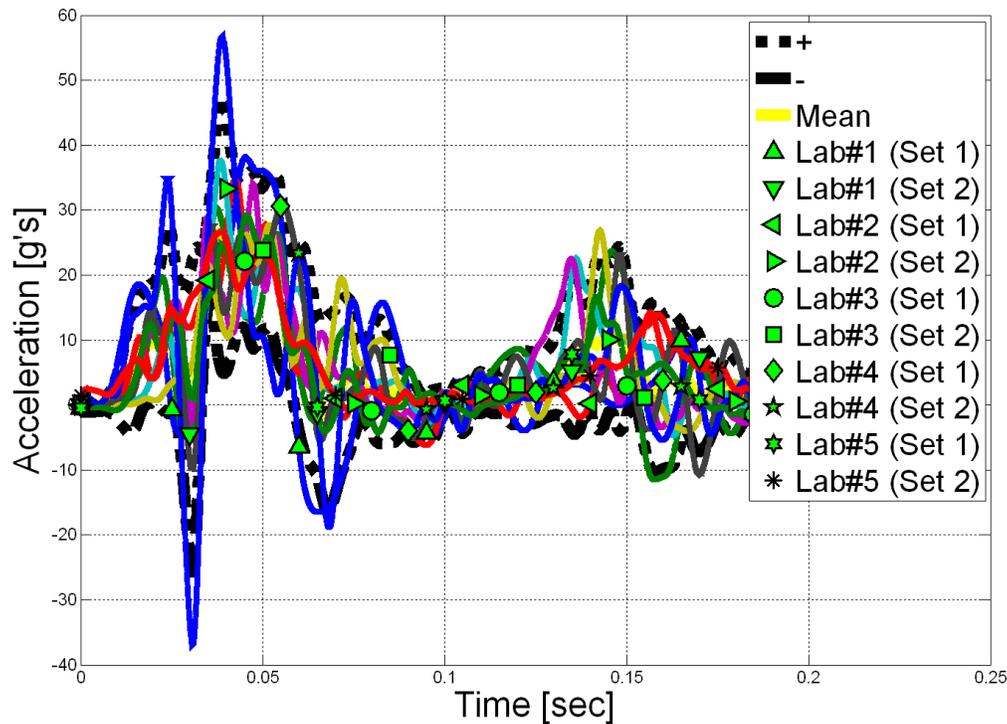


Figure 6.7: Acceleration time histories and 90<sup>th</sup> percentile envelope based on the average curve for Set 1 (left) and Set 2 (right).



**Figure 6.8: Acceleration time histories and 90<sup>th</sup> percentile envelope based on the average curve considering all tests (combination of Set 1 and Set 2).**

This approach required the assumption that the number of curves is large enough to consider the population of the values collected at each sampling point as normal. In the most favorable case (i.e., considering both the two sets of tests) the maximum number of available curves is equal to ten. Moreover, it is worth noting that, the level of confidence (i.e., the width of the corridor) becomes wider the smaller is the number of samples, as the standard deviation of the sampled values tends to be larger with a smaller population of samples. This effect is clearly visible if the width of the confidence interval obtained using the combination of both the two sets of tests is compared to that computed considering only one set (i.e., five tests instead of ten).

#### **6.2.2.1.2 Time history of the residuals**

The previous method to compute the 90<sup>th</sup> percentile corridor is based on the assumption that the population of the values of the curves at each time-step is normal. Because the number of samples at each time-step is restricted to the number of the available curves (i.e., no more than ten), it is quite improbable that the average and standard deviation computed from such a limited population could effectively reflect the actual statistical distribution of the residuals. This

problem is even more actual in roadside safety where, usually, the same full-scale crash test is never performed more than a couple of instances if not only a single time.

Hence, in order to build a confidence corridor which could be based on a greater number of residual samples, a second approach is performed considering the time history of the residuals between each curve and the baseline curve, instead. This second approach is based on the assumption that all the error is homogeneously and randomly distributed on the entire time domain of the curves and, therefore, the variance between the curves at any time point can be accurately represented by the standard deviation of the whole residual time history.

In particular, the following steps are followed for each of the two sets of tests and also for the combination of both of them:

1. Computation of the residuals time histories from the comparison of each curve with the selected baseline curve
2. Computation of the standard deviation for each residual time history
3. Average of the standard deviation

Once the value representing the average standard deviation of the residuals for the specific set of tests (or the combination) is assessed, the 90<sup>th</sup> percentile envelope is created by adding and subtracting its value multiplied by the factor 1.6449 to the respective baseline curve. Note that it is assumed that the residual time distribution is normal and the average is null.

As previously mentioned, in this case, the computation of the 90<sup>th</sup> percentile corridor required the selection of a baseline curve respect to which the residuals are calculated. Initially, the average responses of each of the two single sets of tests and their combination are considered as the baseline curve. Figure 6.9 and Figure 6.10 show the curves and the 90<sup>th</sup> percentile confidence interval for respectively each of the two sets of tests and their combination. As already mentioned at the end of the previous section, it clearly appears that the confidence envelopes calculated using this second method are narrower than the corresponding envelopes which are computed considering the average and standard deviation of the more limited population of samples obtained considering the value of the curves at each sampling point.

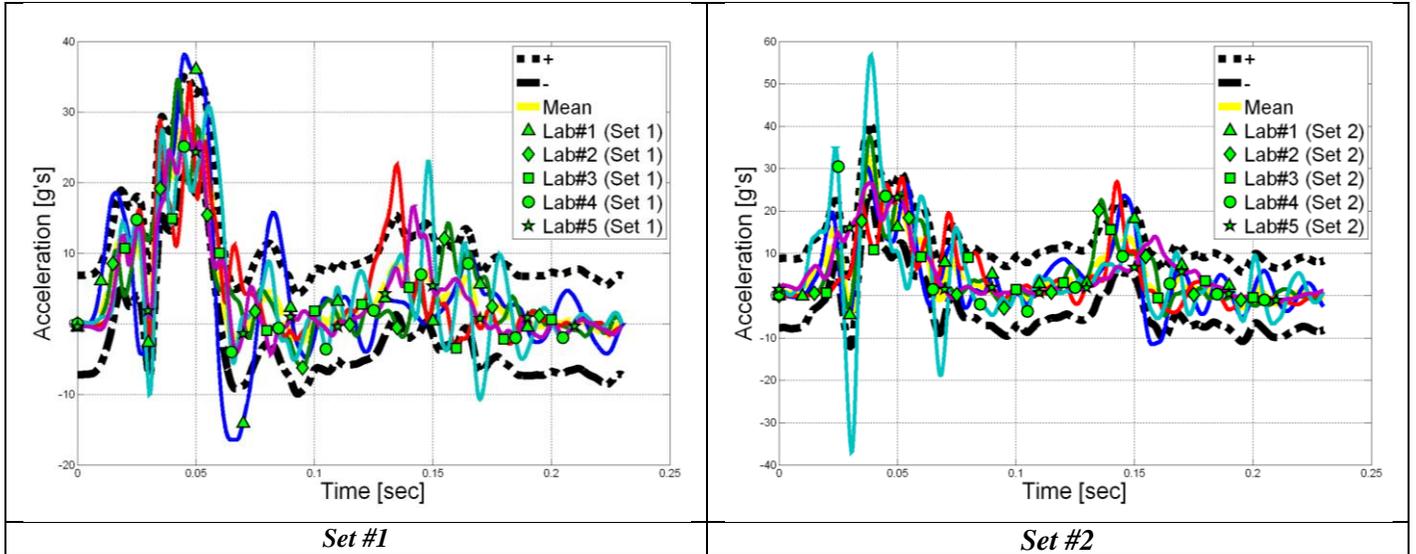


Figure 6.9: Acceleration time histories and 90<sup>th</sup> percentile envelope based on the time history of the residuals respect to the average curve for Set 1 (left) and Set 2 (right).

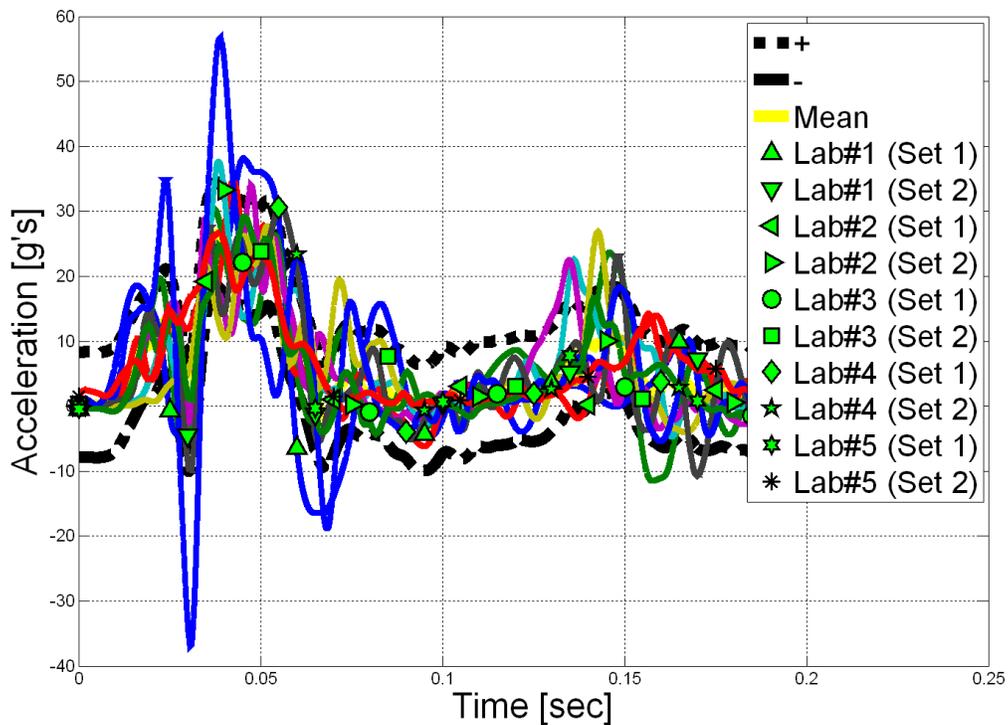


Figure 6.10: Acceleration time histories and 90<sup>th</sup> percentile envelope based on the time history of the residuals respect to the average curve for the combination of all tests (Set 1 and Set 2).

Although the 90<sup>th</sup> percentile envelopes based on the average response of the specific set of tests follow the general behavior of the test curves, they do not completely include all the test curves

in any of the considered cases. In particular, the maximum positive peak, which for a lateral impact represents one of the most important information, sometimes is not included in the confidence interval. This denotes that the use of the average response for the computation of the residuals may be not adequate to build a confidence envelope which could contain most of the curves.

In order to investigate how the selection of different baseline curves affected the final shape and width of the corresponding confidence interval, various test curves are chosen as the baseline as well. In particular, for Set 1 and Set 2 any of the corresponding tests is considered as the baseline for the construction of the 90<sup>th</sup> percentile envelope while, for the combination of both the two sets, only the first five curves are examined. Table 6.1 summarizes the various cases considered (i.e., what baseline tests are considered for each of the sets and also their combination), while the plots of the curves and the corresponding confidence envelopes are shown in Figure 6.11 through Figure 6.13.

**Table 6.1. Summary of the baseline curves considered for the creation of the 90<sup>th</sup> percentile envelopes**

	Considered baseline tests										
	<i>Set 1</i>						<i>Set 2</i>				
	Average	Lab#1	Lab#2	Lab#3	Lab#4	Lab#5	Lab#1	Lab#2	Lab#3	Lab#4	Lab#5
<b>Set 1</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>					
<b>Set 2</b>	<b>X</b>						<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
<b>Set 1+Set2</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>					

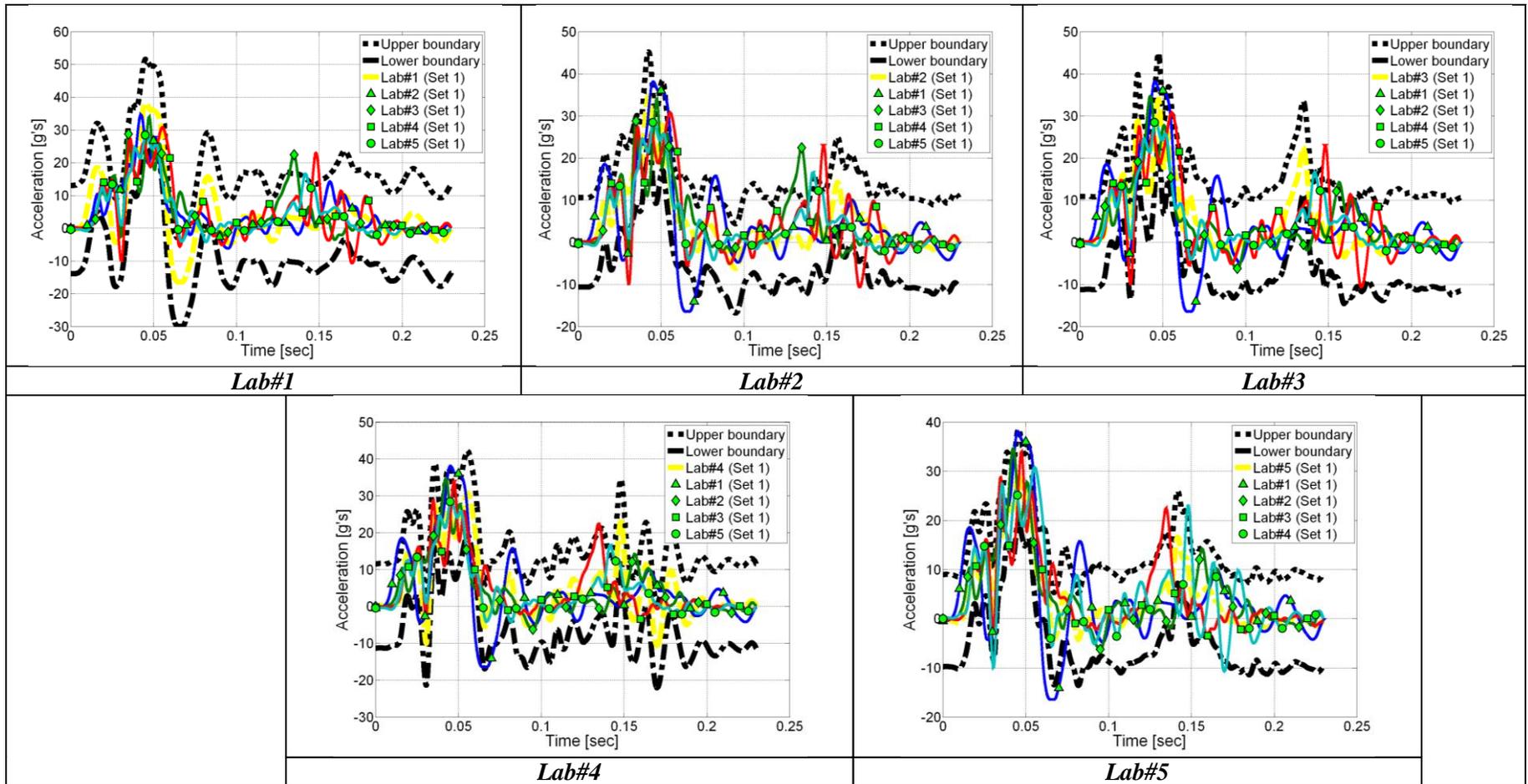


Figure 6.11: Acceleration time histories and 90<sup>th</sup> percentile envelope based on the time history of the residuals respect to the indicated baseline curve for Set 1.

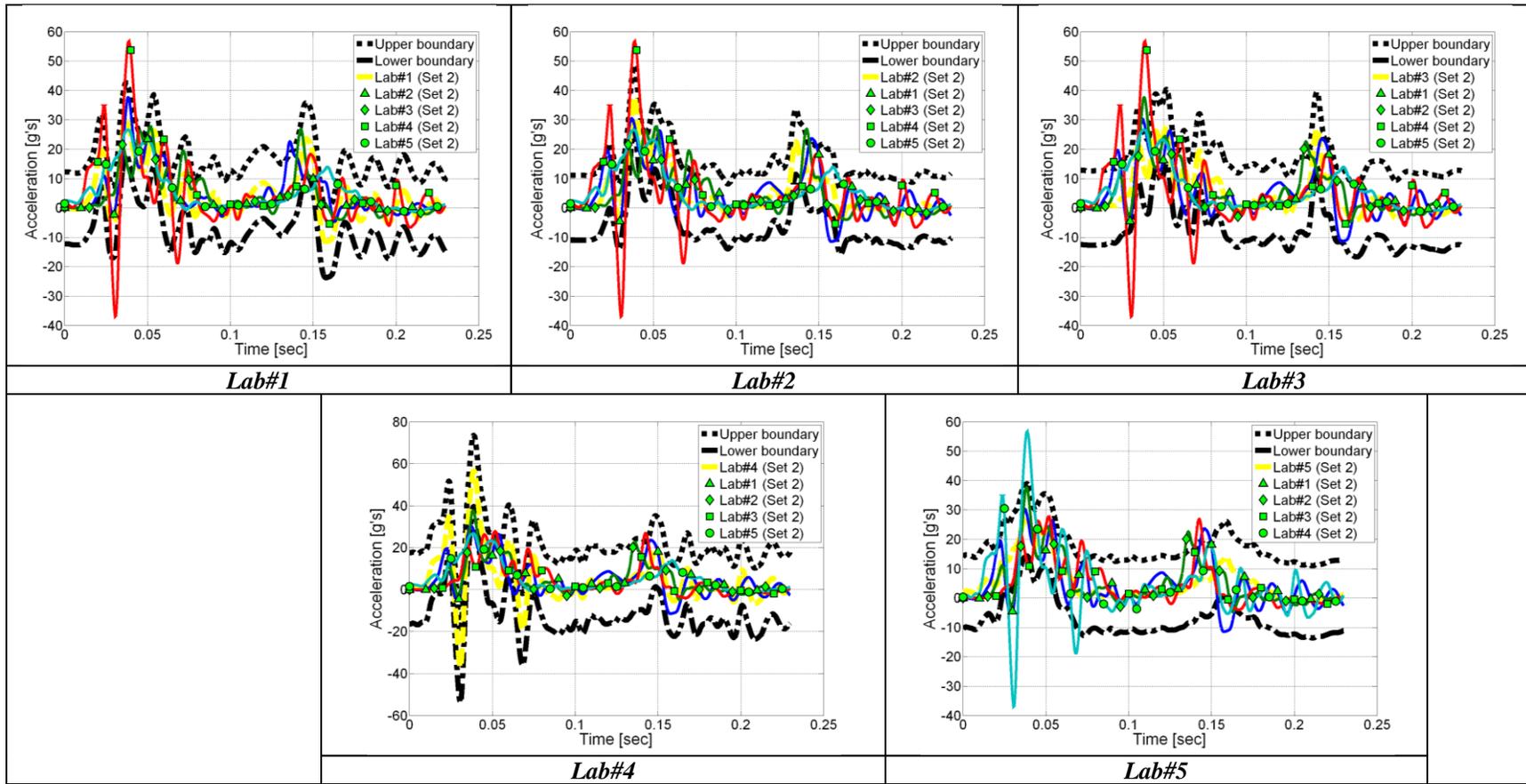


Figure 6.12: Acceleration time histories and 90<sup>th</sup> percentile envelope based on the time history of the residuals respect to the indicated baseline curve for Set 2.

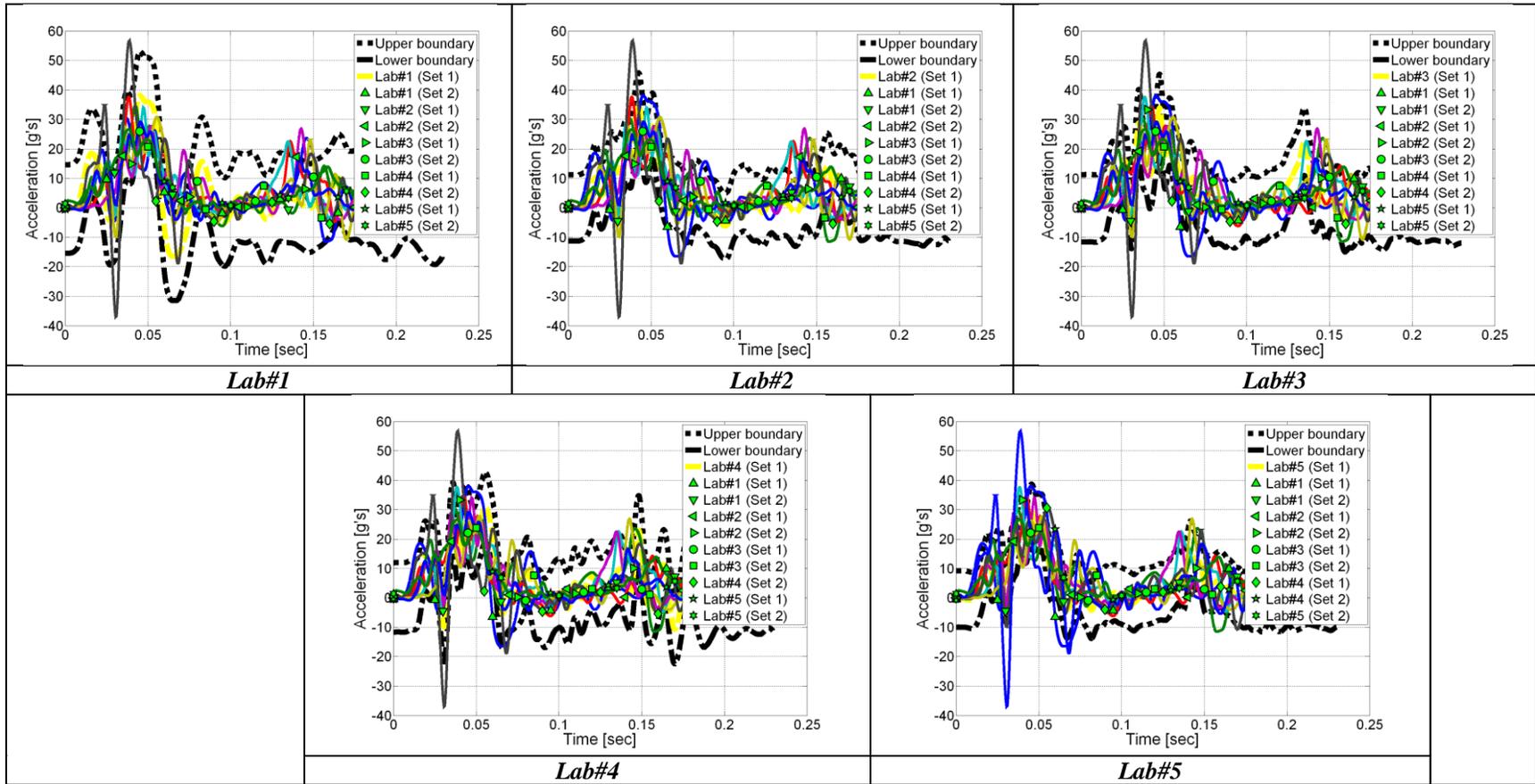


Figure 6.13: Acceleration time histories and 90th percentile envelope based on the time history of the residuals respect to the indicated baseline curve for the combination of all tests (Set 1 and Set 2)

From the analysis of the graphs of the confidence intervals, it is possible to draw some interesting observations which are discussed in the following paragraphs.

In any of the cases one of the curves from the specific set is used as baseline, the corresponding confidence envelopes are wider respect to the case in with the average curves.

When considering all the ten tests (Set1 + Set2), the confidence envelope obtained taking as baseline the curve which is also used as baseline for the synchronization (Lab#5 of Set 1) is very similar to that previously obtained with the average response as the baseline. This confirms that the curve selected as a reference for the synchronization represents the average response of all the other tests.

### **6.2.2.2 Residual distribution**

In order to have a better understanding of the random nature of the residuals, their statistical distribution is analyzed. In fact, since the time histories for all the crash tests represented essentially identical physical events, the residuals for each curve should be attributable only to random experimental error or noise. Statistically speaking, this means that the residual time history should be normally distributed around a mean error equal to zero. Table 6.2 shows the values of the average and standard deviation of the residuals for each time history of the two sets of tests, while Table 6.3 refers to the case with the combination of both the two sets. For all the three different combinations of tests (Set 1, Set2 and their combination) the residuals are computed considering as baseline curves all the cases analyzed for the creation of the 90<sup>th</sup> percentile envelopes in the previous section.

As can be seen, the mean value of the averages from each of the considered cases is relatively small (considering that the peak values for each of the cases are around 40-50 g's), thus confirming the random distribution of the residuals. In particular, this confirms the assumption of a null (or almost null) mean value of the residuals history which was previously made for the creation of the confidence envelopes.

Also the mean value of the standard deviations is limited for any of these three cases.

**Table 6.2. Summary of the average and standard deviation of the residual errors for tests of Set 1 and Set 2**

Baseline		Curves							
		Lab#1	Lab#2	Lab#3	Lab#4	Lab#5	Mean value		
Set 1	Lab#1	Average		0.16	0.25	0.33	-0.15	0.15	
		Std		7.37	9.13	8.66	7.68	8.21	
	Lab#2	Average	-0.16		0.09	0.16	-0.31	-0.06	
		Std	7.37		6.58	7.24	4.59	6.45	
	Lab#3	Average	-0.25	-0.09		0.08	-0.40	-0.17	
		Std	9.13	6.58		6.25	4.88	6.71	
	Lab#4	Average	-0.33	-0.16	-0.08		-0.48	-0.26	
		Std	8.66	7.24	6.25		5.66	6.95	
	Lab#5	Average	0.15	0.31	0.40	0.48		0.33	
		Std	7.68	4.59	4.88	5.66		5.70	
	Average response	Average	-0.12	0.04	0.13	0.21	-0.27	0.00	
		Std	5.91	3.86	4.31	4.51	2.80	4.28	
	Set 2	Lab#1	Average		0.21	-0.06	0.03	0.70	0.22
			Std		6.25	6.56	9.43	7.54	7.44
Lab#2		Average	-0.21		-0.28	-0.18	0.49	-0.05	
		Std	6.25		6.11	9.16	5.35	6.72	
Lab#3		Average	0.06	0.28		0.10	0.76	0.30	
		Std	6.56	6.11		11.90	6.23	7.70	
Lab#4		Average	-0.03	0.18	-0.10		0.66	0.18	
		Std	9.43	9.16	11.90		10.75	10.31	
Lab#5		Average	-0.70	-0.49	-0.76	-0.66		-0.65	
		Std	7.54	5.35	6.23	10.75		7.47	
Average response		Average	-0.18	0.04	-0.24	-0.14	0.52	0.00	
		Std	4.31	3.29	5.02	7.68	4.59	4.98	

**Table 6.3. Summary of the average and standard deviation of the residual errors for tests of the combination of Set 1 and Set 2**

Baseline		Curves										Mean Value
		Lab#1 (Set 1)	Lab#2 (Set 1)	Lab#3 (Set 1)	Lab#4 (Set 1)	Lab#5 (Set 1)	Lab#1 (Set 2)	Lab#2 (Set 2)	Lab#3 (Set 2)	Lab#4 (Set 2)	Lab#5 (Set 2)	
Lab#1 (Set 1)	Average		0.27	0.16	0.48	0.25	0.20	0.33	0.30	-0.15	0.96	0.31
	Std		9.85	7.37	9.42	9.13	9.78	8.66	12.14	7.68	8.18	9.13
Lab#2 (Set 1)	Average	-0.16		0.10	0.32	0.09	0.04	0.16	0.14	-0.31	0.80	0.13
	Std	7.37		8.23	6.54	6.58	6.87	7.24	10.94	4.59	3.04	6.82
Lab#3 (Set 1)	Average	-0.25	0.02		-0.09	0.23	-0.05	0.08	0.05	-0.40	0.71	0.03
	Std	9.13	6.83		6.58	5.50	6.48	6.25	11.06	4.88	5.90	6.96
Lab#4 (Set 1)	Average	-0.33	-0.06	-0.16		0.15	-0.08	-0.12	-0.03	-0.48	0.64	-0.05
	Std	8.66	7.09	7.24		6.27	6.25	7.12	9.66	5.66	6.78	7.19
Lab#5 (Set 1)	Average	0.15	0.42	0.31	0.63		0.40	0.35	0.48	0.45	1.11	0.48
	Std	7.68	6.17	4.59	4.72		4.88	5.22	5.66	9.57	4.00	5.83
Average response	Average	-0.28	-0.01	-0.12	0.20	-0.03	-0.08	0.05	0.02	-0.43	0.68	0.00
	Std	7.06	5.09	4.34	3.79	4.43	5.04	4.57	8.57	2.49	3.79	4.92

Moreover, a visual confirmation of the randomness of the residuals for all the considered baseline curves can be obtained from the graphs of their (i) probability density and (ii) cumulative distribution functions which are shown respectively in Figure 6.14 through Figure 6.16 and Figure 6.17 through Figure 6.19. Both these two functions reproduce the typical behavior of a normal distribution (i.e., a bell-shaped curve with a peak at the mean for the probability functions and an 'S' shape for the cumulative distributions).

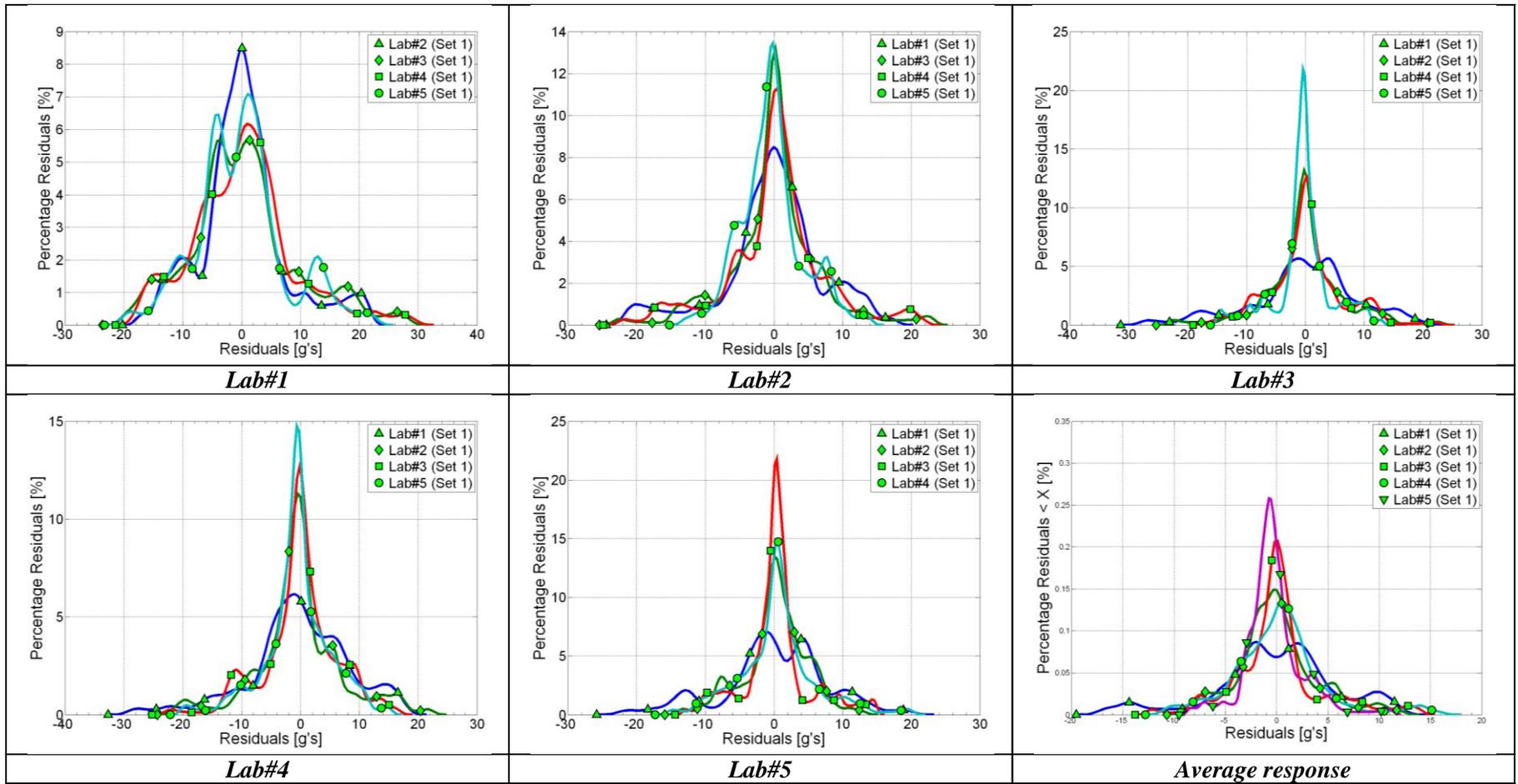


Figure 6.14: Probability density functions of the residuals considering the indicated baseline curves for Set 1.

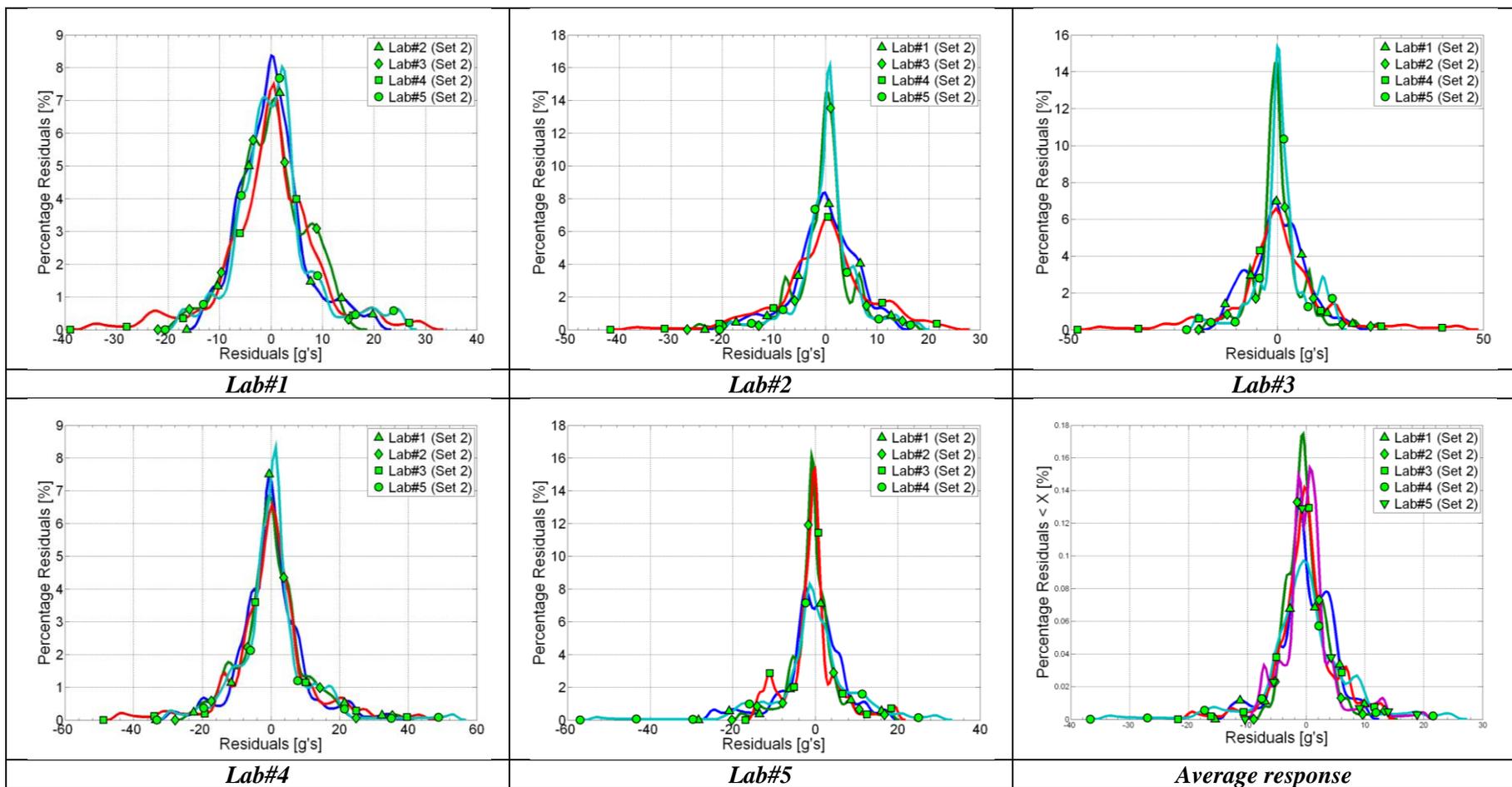


Figure 6.15: Probability density functions of the residuals considering the indicated baseline curves for Set 2.

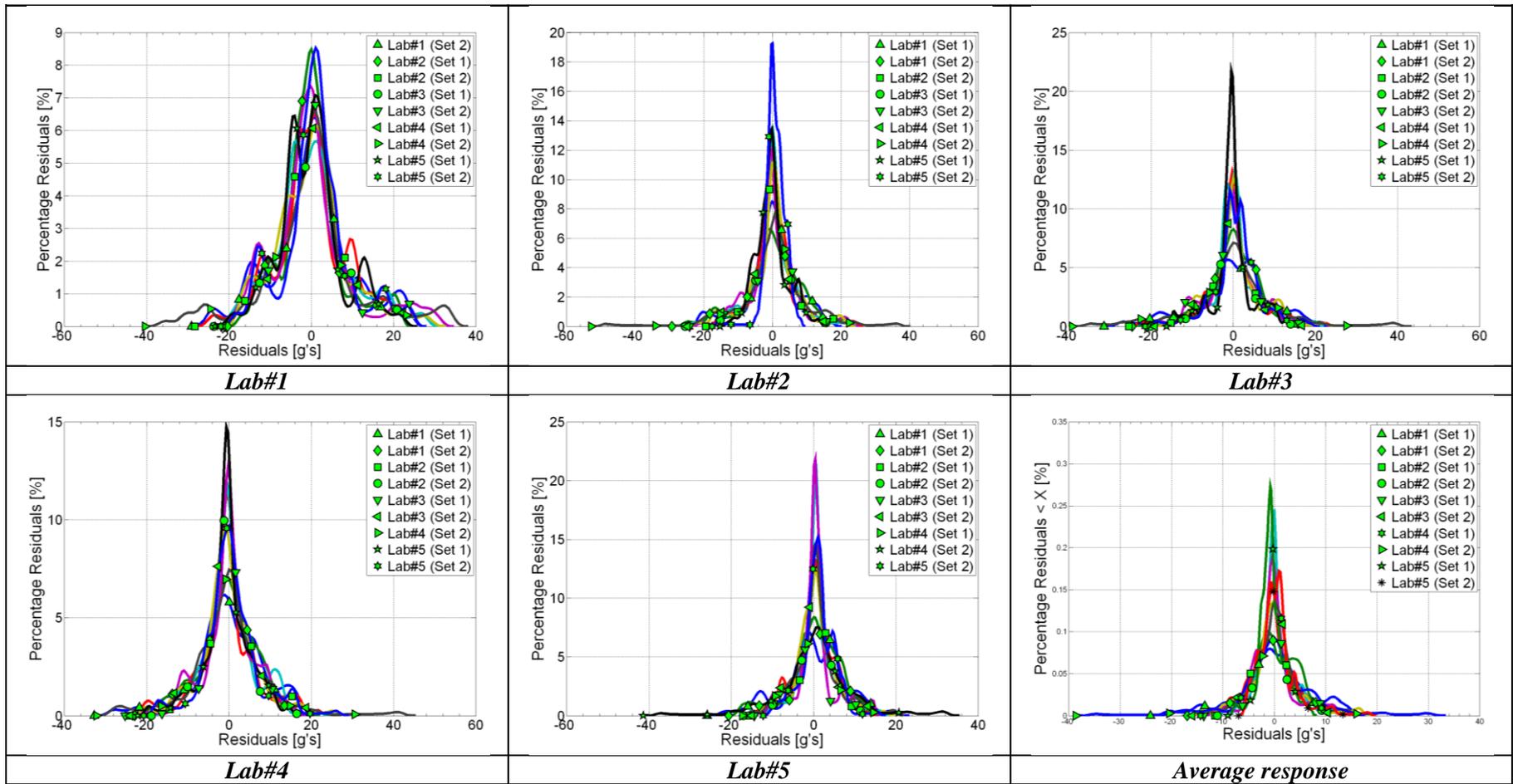


Figure 6.16: Probability density functions of the residuals considering the indicated baseline curves for combination of Set 1 and Set 2.

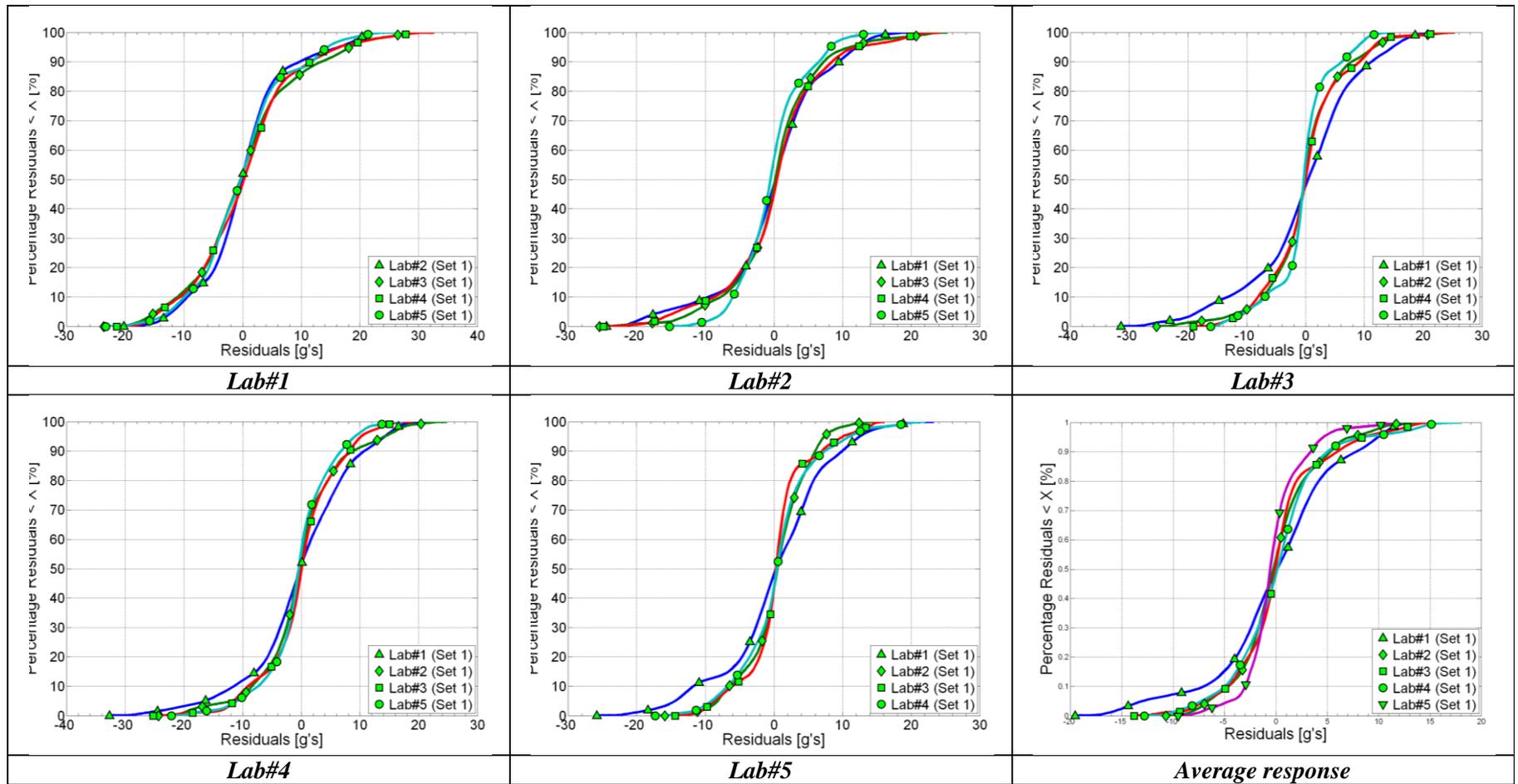


Figure 6.17: Cumulative distribution functions of the residuals considering the indicated baseline curves for Set 1.

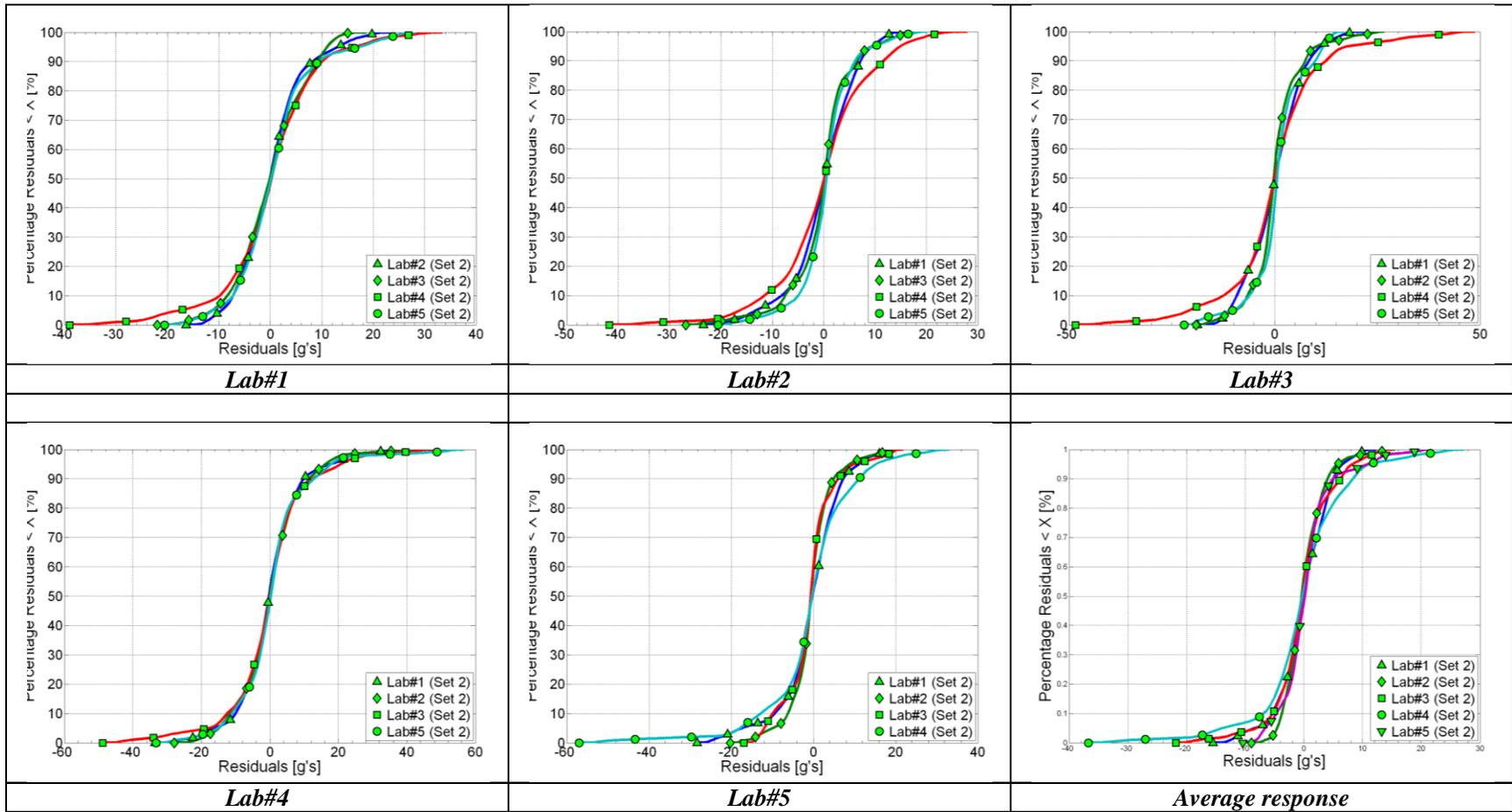


Figure 6.18: Cumulative distribution functions of the residuals considering the indicated baseline curves for Set 2.

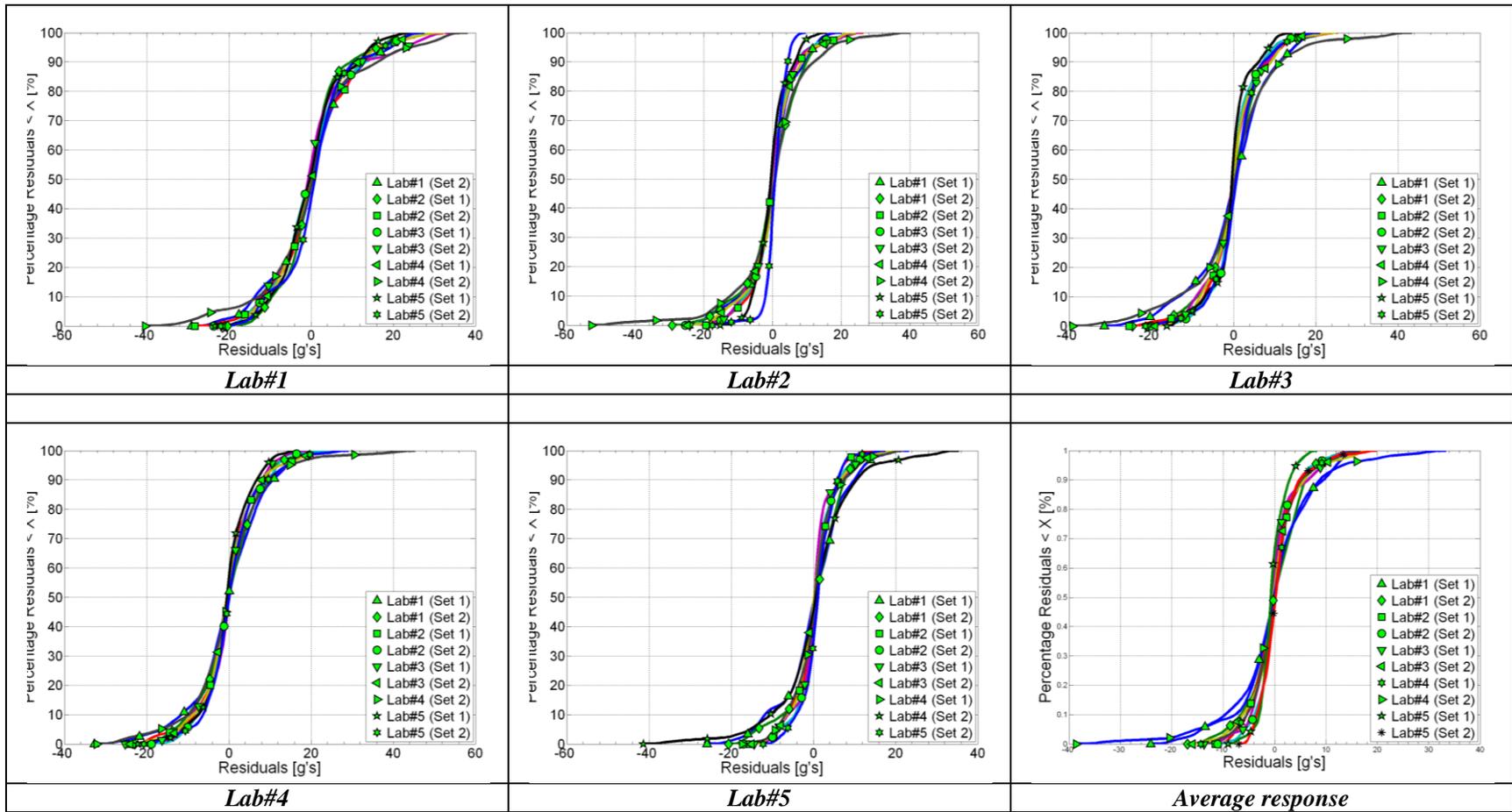


Figure 6.19: Cumulative distribution functions of the residuals considering the indicated baseline curves for combination of Set 1 and Set 2.

### 6.2.3 Comparison of curves

The previous sections discussed the preprocessing of the curves and the statistical analysis of the residuals. This section now describes the results obtained computing the comparison metrics for the preprocessed curves of Set 1, Set 2 and the combination of both of them.

As mentioned before, two different types of comparison are considered, differing for the baseline curve selected:

- Average curve
- A specific curve of the set

#### 6.2.3.1 Case I: Comparison versus the average curve

This first group of comparisons is based on the average response of each specific set of tests.

Initially, the two sets of tests, Set 1 with the same new vehicle and Set 2 with similar vehicles, are considered separately. Table 6.4 and Table 6.5 show the values of the comparison metrics obtained respectively for each of the two singular sets of test data. The values of the metrics for each of the laboratories (i.e., Lab#1 through Lab#5) are shown in each table along with the mean and standard deviation computed considering the scores from all the laboratories.

As for most metrics the sign of the final score depends solely on the order in which the two curves are input (i.e., they are characterized by an asymmetrical behavior), in the calculation of the mean values and the corresponding standard deviation, the value of the metrics is taken without considering the sign. In this manner it is possible to assess the maximum average value that could be observed for the considered curves independently from the order in which they are compared.

It is worth to remind that, in these comparison cases, the order in which curves are considered is completely arbitrary. As the main objective of this study is that of assessing the values of the metrics which represented the typical scatter of similar tests, it is paramount to consider the tests independently by the order in which they are compared (i.e., examining the maximum absolute values of the metrics).

Table 6.4. Values of the comparison metrics for Set#1 (baseline curve: average response)

	Lab#1 (Set 1)	Lab#2 (Set 1)	Lab#3 (Set 1)	Lab#4 (Set 1)	Lab#5 (Set 1)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
	Value [%]	Value [%]					
<b>MPC Metrics</b>							
<i>Geers Magnitude</i>	34.1	3.2	5.5	14.8	-0.5	11.6	13.67
<i>Geers Phase</i>	13	9.5	11.6	10.9	5.3	10.1	2.94
<i>Geers Comprehensive</i>	36.5	10.1	12.8	18.4	5.3	16.6	12.08
<i>Geers CSA Magnitude</i>	34.1	3.2	5.5	14.8	-0.5	11.6	13.67
<i>Geers CSA Phase</i>	13	9.5	11.6	10.9	5.3	10.1	2.94
<i>Geers CSA Comprehensive</i>	36.5	10.1	12.8	18.4	5.3	16.6	12.08
<i>Sprague-Geers Magnitude</i>	34.1	3.2	5.5	14.8	-0.5	11.6	13.67
<i>Sprague-Geers Phase</i>	16.4	14	15.5	15	10.4	14.3	2.33
<i>Sprague-Geers Comprehensive</i>	37.9	14.4	16.4	21.1	10.4	20.0	10.70
<i>Russell Magnitude</i>	20.3	2.7	4.4	10.6	-0.4	7.7	8.01
<i>Russell Phase</i>	16.4	14	15.5	15	10.4	14.3	2.33
<i>Russell Comprehensive</i>	23.1	12.6	14.3	16.3	9.2	15.1	5.18
<i>Knowles-Gear Magnitude</i>	43	33.3	38.6	36.4	33	36.9	4.14
<i>Knowles-Gear Phase</i>	20.9	11.4	42.3	42.3	32.8	29.9	13.59
<i>Knowles-Gear Comprehensive</i>	40.2	30.8	39.2	37.5	33	36.1	4.06
<b>Single Value Metrics</b>	Value [%]	Value [%]					
<i>Whang's inequality metric</i>	35.7	27	27.6	27.6	19.4	27.5	5.77
<i>Theil's inequality metric</i>	29.1	21.9	24.2	24.2	16.2	23.1	4.68
<i>Zilliacus error metric</i>	88.1	56.1	57.4	63.4	39.2	60.8	17.69
<i>RSS error metric metric</i>	68.2	44.5	49.7	52.1	32.4	49.4	12.98
<i>WIFac_Error</i>	46.4	37.4	39.5	40.5	27.2	38.2	7.00
<i>Regression Coefficient</i>	59	85	80.9	78.7	92.4	79.2	12.44
<i>Correlation Coefficient</i>	85.1	86.8	84	85.5	92.8	86.8	3.48
<i>Correlation Coefficient(NARD)</i>	87	90.5	88.4	89.1	94.7	89.9	2.94
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value	Value
<i>Average</i>	0	0	0	-0.01	0.01	0.00	0.01
<i>Std</i>	0.21	0.14	0.15	0.16	0.1	0.15	0.04
<i>T-test</i>	1.36	-0.79	-2.09	-3.13	6.47		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Table 6.5. Values of the comparison metrics for Set#2 (baseline curve: average response)

	Lab#1 (Set 2)	Lab#2 (Set 2)	Lab#3 (Set 2)	Lab#4 (Set 2)	Lab#5 (Set 2)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
<b>MPC Metrics</b>	Value [%]	Value [%]					
<i>Geers Magnitude</i>	11.6	8.7	2.5	51.4	1.2	15.1	20.75
<i>Geers Phase</i>	10.5	6.3	16.3	17.1	14	12.8	4.46
<i>Geers Comprehensive</i>	15.6	10.7	16.5	54.1	14	22.2	17.98
<i>Geers CSA Magnitude</i>	11.6	8.7	2.5	51.4	1.2	15.1	20.75
<i>Geers CSA Phase</i>	10.5	6.3	16.3	17.1	14	12.8	4.46
<i>Geers CSA Comprehensive</i>	15.6	10.7	16.5	54.1	14	22.2	17.98
<i>Sprague-Geers Magnitude</i>	11.6	8.7	2.5	51.4	1.2	15.1	20.75
<i>Sprague-Geers Phase</i>	14.7	11.3	18.4	18.9	17	16.1	3.12
<i>Sprague-Geers Comprehensive</i>	18.7	14.3	18.6	54.7	17.1	24.7	16.88
<i>Russell Magnitude</i>	8.6	6.7	2.1	26.8	1	9.0	10.42
<i>Russell Phase</i>	14.7	11.3	18.4	18.9	17	16.1	3.12
<i>Russell Comprehensive</i>	15.1	11.7	16.4	29.1	15.1	17.5	6.73
<i>Knowles-Gear Magnitude</i>	42.4	51.9	47.4	69	53.2	52.8	10.01
<i>Knowles-Gear Phase</i>	16.2	60.4	102.3	3.2	100	56.4	46.02
<i>Knowles-Gear Comprehensive</i>	39.2	53.4	60.1	63	63.4	55.8	10.12
<b>Single Value Metrics</b>	Value [%]	Value [%]					
<i>Whang's inequality metric</i>	27.4	22	31.7	37.2	27.7	29.2	5.65
<i>Theil's inequality metric</i>	23.5	18.2	28.6	35.2	26.4	26.4	6.28
<i>Zilliacus error metric</i>	61.5	45	64.2	93.8	56.7	64.2	18.09
<i>RSS error metric metric</i>	49.7	37.9	57.8	88.5	53.2	57.4	18.87
<i>WIFac_Error</i>	38.7	31.9	46.3	54.3	44.2	43.1	8.39
<i>Regression Coefficient</i>	79.6	88.7	71	N/A	76.2	78.9	7.44
<i>Correlation Coefficient</i>	85.9	91.2	76.7	81.1	78.6	82.7	5.87
<i>Correlation Coefficient(NARD)</i>	89.5	93.7	83.7	82.9	86	87.2	4.46
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value	Value
<i>Average</i>	0.01	0	0.01	0	-0.02	0.01	0.01
<i>Std</i>	0.13	0.1	0.15	0.24	0.14	0.15	0.05
<i>T-test</i>	2.76	-0.74	3.23	1.26	-7.7		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Next, all ten tests from both sets are compared together considering the corresponding average response. Table 6.6 shows the results obtained in this case.

Table 6.6. Values of the comparison metrics considering all tests (baseline curve: average response)

	Lab#1 (Set 1)	Lab#1 (Set 2)	Lab#2 (Set 1)	Lab#2 (Set 2)	Lab#3 (Set 1)	Lab#3 (Set 2)	Lab#4 (Set 1)	Lab#4 (Set 2)	Lab#5 (Set 1)	Lab#5 (Set 2)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
<b>MPC Metrics</b>	Value [%]	Value [%]										
<i>Geers Magnitude</i>	37.5	14.6	5.8	11.6	8.2	5.2	17.7	55.4	2	3.9	16.2	17.20
<i>Geers Phase</i>	20.2	14.8	12.3	8.4	12.4	16.8	11.1	23.2	4.3	9.9	13.3	5.61
<i>Geers Comprehensive</i>	42.6	20.8	13.6	14.3	14.8	17.6	20.9	60.1	4.8	10.7	22.0	16.68
<i>Geers CSA Magnitude</i>	37.5	14.6	5.8	11.6	8.2	5.2	17.7	55.4	2	3.9	16.2	17.20
<i>Geers CSA Phase</i>	20.2	14.8	12.3	8.4	12.4	16.8	11.1	23.2	4.3	9.9	13.3	5.61
<i>Geers CSA Comprehensive</i>	42.6	20.8	13.6	14.3	14.8	17.6	20.9	60.1	4.8	10.7	22.0	16.68
<i>Sprague-Geers Magnitude</i>	37.5	14.6	5.8	11.6	8.2	5.2	17.7	55.4	2	3.9	16.2	17.20
<i>Sprague-Geers Phase</i>	20.6	17.6	16	13.1	16	18.7	15.1	22.1	9.4	14.3	16.3	3.70
<i>Sprague-Geers Comprehensive</i>	42.8	22.8	17	17.5	18	19.4	23.3	59.7	9.6	14.8	24.5	15.16
<i>Russell Magnitude</i>	21.7	10.5	4.7	8.6	6.3	4.2	12.3	28.1	1.7	3.2	10.1	8.58
<i>Russell Phase</i>	20.6	17.6	16	13.1	16	18.7	15.1	22.1	9.4	14.3	16.3	3.70
<i>Russell Comprehensive</i>	26.5	18.1	14.7	13.9	15.3	17	17.3	31.7	8.5	13	17.6	6.75
<i>Knowles-Gear Magnitude</i>	57.4	46.6	32.9	42.5	42.7	54.1	40.7	78.6	28.6	40.4	46.5	14.20
<i>Knowles-Gear Phase</i>	22.8	25.2	8.7	72.8	38.8	118	38.8	4.4	29.6	100	45.9	38.42
<i>Knowles-Gear Comprehensive</i>	53.3	43.7	30.2	48.9	42.1	69	40.4	71.8	28.7	55	48.3	14.47
<b>Single Value Metrics</b>	Value [%]	Value [%]										
<i>Whang's inequality metric</i>	41.8	33.2	30.3	23.7	28.9	32.6	28	42	18.7	24.1	30.3	7.50
<i>Theil's inequality metric</i>	35.2	28	25	21.2	25.2	29	24.8	39.7	14.8	22.3	26.5	7.05
<i>Ziliacus error metric</i>	102.7	77.1	62.7	50	59.8	68.1	64.1	109.6	37.7	50.9	68.3	22.76
<i>RSS error metric metric</i>	83.5	60.1	51.4	44.8	52.4	59.6	54	101.4	29.9	45.6	58.3	20.41
<i>WIFac_Error</i>	52.8	44.3	41.9	36.2	41.7	48	41	59.8	24.3	39.4	42.9	9.55
<i>Regression Coefficient</i>	N/A	68.3	78.1	83.9	77.1	69	75.5	N/A	93.2	83.3	78.6	8.23
<i>Correlation Coefficient</i>	76.1	79.8	82.7	88.1	82.6	76.1	85.2	73.1	94.2	85	82.3	6.32
<i>Correlation Coefficient(NARD)</i>	79.8	85.2	87.7	91.6	87.6	83.2	88.9	76.8	95.7	90.1	86.7	5.61
<b>ANOVA Metrics</b>	Value	Value										
<i>Average</i>	0.01	0	0	-0.01	0	0	0	0	0.02	-0.03	0.01	0.01
<i>Std</i>	0.27	0.19	0.16	0.14	0.17	0.19	0.17	0.32	0.09	0.14	0.18	0.07
<i>T-test</i>	2.7	0.18	1.83	-3.55	0.45	1.04	-0.68	-0.15	11.69	-12.22		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Table 6.7 summarizes the values of the mean and the standard deviation evaluated for each one of the three cases analyzed (i.e., Set 1, Set 2 and their combination). The average values of the mean and standard deviation from these three cases are computed and, ultimately, used to create

an interval of confidence for each specific metric. In fact, assuming that distribution of the metric values is normal, the 90 percent of the values should be within 1.6449 standard deviations of the mean. Potential acceptance criteria are listed in the last column of Table 6.7 by calculating the 90<sup>th</sup> percentile limit of the observed variation of the metric.

**Table 6.7. Average values of the means and standard deviations, and the corresponding acceptance criteria**

	Mean			STD			Average Mean	Average STD	Acceptable value	
	Set#1	Set#2	All	Set#1	Set#2	All			min [%]	max [%]
<b>MPC Metrics</b>	Value [%]	Value [%]	min [%]	max [%]						
<i>Geers Magnitude</i>	11.6	15.1	16.2	13.7	20.8	17.2	14.3	17.2	-14.3	42.9
<i>Geers Phase</i>	10.1	12.8	13.3	2.9	4.5	5.6	12.1	4.3	4.9	19.3
<i>Geers Comprehensive</i>	16.6	22.2	22.0	12.1	18.0	16.7	20.3	15.6	-5.6	46.1
<i>Geers CSA Magnitude</i>	11.6	15.1	16.2	13.7	20.8	17.2	14.3	17.2	-14.3	42.9
<i>Geers CSA Phase</i>	10.1	12.8	13.3	2.9	4.5	5.6	12.1	4.3	4.9	19.3
<i>Geers CSA Comprehensive</i>	16.6	22.2	22.0	12.1	18.0	16.7	20.3	15.6	-5.6	46.1
<i>Sprague-Geers Magnitude</i>	11.6	15.1	16.2	13.7	20.8	17.2	14.3	17.2	-14.3	42.9
<i>Sprague-Geers Phase</i>	14.3	16.1	16.3	2.3	3.1	3.7	15.5	3.0	10.5	20.6
<i>Sprague-Geers Comprehensive</i>	20.0	24.7	24.5	10.7	16.9	15.2	23.1	14.2	-0.6	46.7
<i>Russell Magnitude</i>	7.7	9.0	10.1	8.0	10.4	8.6	9.0	9.0	-6.0	23.9
<i>Russell Phase</i>	14.3	16.1	16.3	2.3	3.1	3.7	15.5	3.0	10.5	20.6
<i>Russell Comprehensive</i>	15.1	17.5	17.6	5.2	6.7	6.8	16.7	6.2	6.4	27.0
<i>Knowles-Gear Magnitude</i>	36.9	52.8	46.5	4.1	10.0	14.2	45.4	9.4	29.7	61.0
<i>Knowles-Gear Phase</i>	29.9	56.4	45.9	13.6	46.0	38.4	44.1	32.7	-10.2	98.3
<i>Knowles-Gear Comprehensive</i>	36.1	55.8	48.3	4.1	10.1	14.5	46.8	9.6	30.9	62.6
<b>Single Value Metrics</b>	Value [%]	Value [%]	min [%]	max [%]						
<i>Whang's inequality metric</i>	27.5	29.2	30.3	5.8	5.6	7.5	29.0	6.3	18.5	39.5
<i>Theil's inequality metric</i>	23.1	26.4	26.5	4.7	6.3	7.1	25.3	6.0	15.4	35.3
<i>Zilliacus error metric</i>	60.8	64.2	68.3	17.7	18.1	22.8	64.5	19.5	32.1	96.8
<i>RSS error metric</i>	49.4	57.4	58.3	13.0	18.9	20.4	55.0	17.4	26.1	83.9
<i>WIFac_Error</i>	38.2	43.1	42.9	7.0	8.4	9.5	41.4	8.3	27.6	55.2
<i>Regression Coefficient</i>	79.2	78.9	78.6	12.4	7.4	8.2	78.9	9.4	63.3	94.4
<i>Correlation Coefficient</i>	86.8	82.7	82.3	3.5	5.9	6.3	83.9	5.2	75.3	92.6
<i>Correlation Coefficient(NARD)</i>	89.9	87.2	86.7	2.9	4.5	5.6	87.9	4.3	80.7	95.1
<b>ANOVA Metrics</b>	Value	Value	min	max						
<i>Average</i>	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.01	0.02
<i>Std</i>	0.15	0.15	0.18	0.04	0.05	0.07	0.16	0.05	0.07	0.25

### 6.2.3.2 Comparison versus specific curves of the set

In this case, the curves are compared respect to a baseline curve which is selected from the respective set. Similarly to the previous analysis, also the combination of both the two sets is considered.

While the 90<sup>th</sup> percentile envelope based on the average response contained most of the curves from the ten tests, one test had peaks clearly falling out of the confidence envelope in the case with the combination of Set1 and Set 2 (Figure 6.10). Since these mentioned peaks occur during the first and most important part of the impact, it is decided to investigate also the case in which the curve of one of the test would be considered as the baseline curve.

Table 6.8 summarizes the baseline curves selected for each specific set and combination of tests. As can be seen from the table, two different baseline curves are investigated for each case. The response from the tests Lab#1 and Lab#5 belonging to the respective sets are used as reference curves for the comparison of the corresponding curves of those sets while, in the case of the combination of the two sets, Lab#1 and Lab#5 from Set 1 are considered.

**Table 6.8. Summary of the baseline curves considered for the comparison of each set of tests**

	Considered baseline tests									
	Set 1					Set 2				
	Lab#1	Lab#2	Lab#3	Lab#4	Lab#5	Lab#1	Lab#2	Lab#3	Lab#4	Lab#5
<b>Set 1</b>	<b>X</b>				<b>X</b>					
<b>Set 2</b>						<b>X</b>				<b>X</b>
<b>Set 1+Set2</b>	<b>X</b>				<b>X</b>					

Although theoretically any of the tests could have been selected as the baseline for the comparison, in this study the investigation is focused on some specific tests of particular interests. In fact, as mentioned, the selection of the tests of Table 6.8 as the baseline curves is made based on considerations on the corresponding 90<sup>th</sup> percentile envelopes. The discussion of the cases considered and the obtained results are shown in the following two sections

### **6.2.3.2.1 Case II: First curve of each sets and combination of sets**

The selection of the first test for each of the two sets (i.e., respectively Lab#1 of Set1 and Lab#1 of Set2) and Lab#1 of Set1 for the combination of the two sets is motivated by the fact that the 90<sup>th</sup> percentile envelopes based on these curves included most of the curves of their respective set/combination of sets. In particular, in the case the two sets are combined together, the confidence envelope based on Lab#1 of Set1 enclosed almost completely the maximum positive peak values of the curves which, as previously mentioned, represent one of the most important phenomena of the lateral acceleration time history in this kind of crash tests. It is worth noting that, for both the two single sets, the choice of the curve from the corresponding Lab#1 is preferred to that of the Lab#4 as in the latter cases, although the confidence envelopes completely contained the curves, they are also characterized by an excessive negative peak at the beginning of the impact.

The values of the metrics obtained from the comparison of each of the two sets of tests considered separately are shown in Table 6.9 and Table 6.10, while the results from the combination of the two sets are listed in Table 6.11.

Table 6.9. Values of the comparison metrics for Set#1 (baseline curve: Lab 1)

	Lab#2 (Set 1)	Lab#3 (Set 1)	Lab#4 (Set 1)	Lab#5 (Set 1)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
<b>MPC Metrics</b>	Value [%]	Value [%]				
<i>Geers Magnitude</i>	-23	-21.4	-14.4	-25.8	21.2	4.85
<i>Geers Phase</i>	22.6	36.3	31.2	24.9	28.8	6.21
<i>Geers Comprehensive</i>	32.3	42.1	34.3	35.8	36.1	4.23
<i>Geers CSA Magnitude</i>	-23	-21.4	-14.4	-25.8	21.2	4.85
<i>Geers CSA Phase</i>	22.6	36.3	31.2	24.9	28.8	6.21
<i>Geers CSA Comprehensive</i>	32.3	42.1	34.3	35.8	36.1	4.23
<i>Sprague-Geers Magnitude</i>	-23	-21.4	-14.4	-25.8	21.2	4.85
<i>Sprague-Geers Phase</i>	21.8	28	25.8	22.9	24.6	2.81
<i>Sprague-Geers Comprehensive</i>	31.7	35.2	29.6	34.5	32.8	2.59
<i>Russell Magnitude</i>	-18.5	-17.2	-11.8	-20.6	17.0	3.75
<i>Russell Phase</i>	21.8	28	25.8	22.9	24.6	2.81
<i>Russell Comprehensive</i>	25.3	29.1	25.2	27.3	26.7	1.86
<i>Knowles-Gear Magnitude</i>	55.2	60.6	48.6	56.8	55.3	5.01
<i>Knowles-Gear Phase</i>	40.9	79.9	79.9	67.9	67.2	18.39
<i>Knowles-Gear Comprehensive</i>	53.1	64.2	55.1	58.8	57.8	4.88
<b>Single Value Metrics</b>	Value [%]	Value [%]				
<i>Whang's inequality metric</i>	40.9	53.2	46.3	46	46.6	5.05
<i>Theil's inequality metric</i>	35.8	43.9	40.1	37.9	39.4	3.46
<i>Ziliacus error metric</i>	70.8	92.2	87.1	77.9	82.0	9.53
<i>RSS error metric metric</i>	63.3	78.5	74.4	66	70.6	7.10
<i>WIFac_Error</i>	53.3	61.5	58.2	55.5	57.1	3.54
<i>Regression Coefficient</i>	72.6	52.4	58.9	69.8	63.4	9.43
<i>Correlation Coefficient</i>	72.7	55.3	62	69.8	65.0	7.86
<i>Correlation Coefficient(NARD)</i>	77.4	63.7	68.8	75.1	71.3	6.21
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value
<i>Average</i>	0	-0.01	-0.01	0	0.01	0.01
<i>Std</i>	0.19	0.24	0.23	0.2	0.22	0.02
<i>T-test</i>	-1.51	-1.87	-2.56	1.31		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Table 6.10. Values of the comparison metrics for Set#2 (baseline curve: Lab 1)

	Lab#2 (Set 2)	Lab#3 (Set 2)	Lab#4 (Set 2)	Lab#5 (Set 2)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
	Value [%]	Value [%]				
<b>MPC Metrics</b>						
<i>Geers Magnitude</i>	-2.6	-8.2	35.6	-9.3	13.9	14.74
<i>Geers Phase</i>	21.3	24.5	30.2	33.2	27.3	5.39
<i>Geers Comprehensive</i>	21.5	25.9	46.7	34.5	32.2	11.10
<i>Geers CSA Magnitude</i>	-2.6	-8.2	35.6	-9.3	13.9	14.74
<i>Geers CSA Phase</i>	21.3	24.5	30.2	33.2	27.3	5.39
<i>Geers CSA Comprehensive</i>	21.5	25.9	46.7	34.5	32.2	11.10
<i>Sprague-Geers Magnitude</i>	-2.6	-8.2	35.6	-9.3	13.9	14.74
<i>Sprague-Geers Phase</i>	21.2	22.8	25.4	26.7	24.0	2.49
<i>Sprague-Geers Comprehensive</i>	21.3	24.2	43.8	28.3	29.4	10.02
<i>Russell Magnitude</i>	-2.3	-6.9	20.9	-7.8	9.5	7.99
<i>Russell Phase</i>	21.2	22.8	25.4	26.7	24.0	2.49
<i>Russell Comprehensive</i>	18.9	21.1	29.2	24.6	23.5	4.49
<i>Knowles-Gear Magnitude</i>	54.2	60.7	97.3	68.4	70.2	19.01
<i>Knowles-Gear Phase</i>	38	74	16.7	100	57.2	37.07
<i>Knowles-Gear Comprehensive</i>	51.9	63.2	89.1	74.6	69.7	15.91
<b>Single Value Metrics</b>	Value [%]	Value [%]				
<i>Whang's inequality metric</i>	37.7	42.2	43.8	43.2	41.7	2.76
<i>Theil's inequality metric</i>	32.7	35.3	41.3	41	37.6	4.26
<i>Zilliacus error metric</i>	69.3	77	97.3	79.5	80.8	11.84
<i>RSS error metric</i>	64.5	67.6	97.2	78.1	76.9	14.76
<i>WIFac_Error</i>	49.6	54	61.1	56.8	55.4	4.83
<i>Regression Coefficient</i>	67.1	62.8	N/A0	43.9	57.9	12.34
<i>Correlation Coefficient</i>	71.2	66.8	63.9	53	63.7	7.75
<i>Correlation Coefficient(NARD)</i>	78.7	75.4	69.8	66.8	72.7	5.37
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value
<i>Average</i>	-0.01	0	0	-0.02	0.01	0.01
<i>Std</i>	0.21	0.22	0.31	0.25	0.25	0.05
<i>T-test</i>	-2.3	0.66	-0.23	-6.27		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Table 6.11. Values of the comparison metrics considering all tests (baseline curve: Lab 1 Set#1)

	Lab#1 (Set 2)	Lab#2 (Set 1)	Lab#2 (Set 2)	Lab#3 (Set 1)	Lab#3 (Set 2)	Lab#4 (Set 1)	Lab#4 (Set 2)	Lab#5 (Set 1)	Lab#5 (Set 2)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
<b>MPC Metrics</b>	Value [%]	Value [%]									
Geers Magnitude	-16.7	-23	-18.9	-21.4	-23.5	-14.4	13	-25.8	-24.5	20.1	4.60
Geers Phase	41.3	22.6	38.3	36.3	42.6	31.2	47.5	24.9	29.2	34.9	8.44
Geers Comprehensive	44.6	32.3	42.7	42.1	48.6	34.3	49.2	35.8	38.1	40.9	6.10
Geers CSA Magnitude	-16.7	-23	-18.9	-21.4	-23.5	-14.4	13	-25.8	-24.5	20.1	4.60
Geers CSA Phase	41.3	22.6	38.3	36.3	42.6	31.2	47.5	24.9	29.2	34.9	8.44
Geers CSA Comprehensive	44.6	32.3	42.7	42.1	48.6	34.3	49.2	35.8	38.1	40.9	6.10
Sprague-Geers Magnitude	-16.7	-23	-18.9	-21.4	-23.5	-14.4	13	-25.8	-24.5	20.1	4.60
Sprague-Geers Phase	30	21.8	28.8	28	30.5	25.8	32.4	22.9	25	27.2	3.60
Sprague-Geers Comprehensive	34.4	31.7	34.5	35.2	38.5	29.6	34.9	34.5	34.9	34.2	2.45
Russell Magnitude	-13.6	-18.5	-15.3	-17.2	-18.8	-11.8	9.5	-20.6	-19.5	16.1	3.79
Russell Phase	30	21.8	28.8	28	30.5	25.8	32.4	22.9	25	27.2	3.60
Russell Comprehensive	29.2	25.3	28.9	29.1	31.8	25.2	29.9	27.3	28.1	28.3	2.13
Knowles-Gear Magnitude	63.3	55.2	63.9	60.6	78.9	48.6	83.4	56.8	59.1	63.3	11.16
Knowles-Gear Phase	62.3	40.9	123.9	79.9	182.4	79.9	35.2	67.9	100	85.8	45.42
Knowles-Gear Comprehensive	63.1	53.1	77.2	64.2	103.6	55.1	77.5	58.8	67.6	68.9	15.60
<b>Single Value Metrics</b>	Value [%]	Value [%]									
Whang's inequality metric	51.6	40.9	53.3	53.2	52.7	46.3	53.3	46	44.9	49.1	4.66
Theil's inequality metric	46.2	35.8	44.8	43.9	47.6	40.1	49	37.9	40.3	42.8	4.54
Ziliacus error metric	98.3	70.8	93.6	92.2	92	87.1	112.1	77.9	79.1	89.2	12.33
RSS error metric metric	84.7	63.3	81.1	78.5	84	74.4	104.4	66	70.8	78.6	12.30
WIFac_Error	65.9	53.3	63.9	61.5	63.7	58.2	69.5	55.5	58	61.1	5.23
Regression Coefficient	39.5	72.6	47.6	52.4	41	58.9	N/A	69.8	64	55.7	12.64
Correlation Coefficient	49.3	72.7	52.6	55.3	47.2	62	44.6	69.8	64.6	57.6	10.13
Correlation Coefficient(NARD)	58.7	77.4	61.7	63.7	57.4	68.8	52.5	75.1	70.8	65.1	8.44
<b>ANOVA Metrics</b>	Value	Value									
Average	-0.01	0	-0.01	-0.01	-0.01	-0.01	-0.01	0	-0.03	0.01	0.01
Std	0.26	0.19	0.25	0.24	0.26	0.23	0.32	0.2	0.21	0.24	0.04
T-test	-1.84	-1.51	-3.45	-1.87	-1.41	-2.56	-1.68	1.31	-7.99		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Similarly to the previous case, the mean and the standard deviation of the values are evaluated for each of the two sets of tests and also for their combination eventually an interval of confidence for each specific metric is assessed. Table 6.12 summarizes the results obtained.

**Table 6.12. Average values of the means and standard deviations, and the corresponding acceptance criteria (baseline curve: Lab 1 of each set and Lab 1 Set#1 for the combination of the two sets)**

	Mean			STD			Average Mean	Average STD	Acceptable value	
	Set#1	Set#2	All	Set#1	Set#2	All			min [%]	max [%]
<b>MPC Metrics</b>	Value [%]	Value [%]	min [%]	max [%]						
<i>Geers Magnitude</i>	21.2	13.9	20.1	4.9	14.7	4.6	18.4	8.1	5.0	31.8
<i>Geers Phase</i>	28.8	27.3	34.9	6.2	5.4	8.4	30.3	6.7	19.2	41.4
<i>Geers Comprehensive</i>	36.1	32.2	40.9	4.2	11.1	6.1	36.4	7.1	24.5	48.2
<i>Geers CSA Magnitude</i>	21.2	13.9	20.1	4.9	14.7	4.6	18.4	8.1	5.0	31.8
<i>Geers CSA Phase</i>	28.8	27.3	34.9	6.2	5.4	8.4	30.3	6.7	19.2	41.4
<i>Geers CSA Comprehensive</i>	36.1	32.2	40.9	4.2	11.1	6.1	36.4	7.1	24.5	48.2
<i>Sprague-Geers Magnitude</i>	21.2	13.9	20.1	4.9	14.7	4.6	18.4	8.1	5.0	31.8
<i>Sprague-Geers Phase</i>	24.6	24.0	27.2	2.8	2.5	3.6	25.3	3.0	20.4	30.2
<i>Sprague-Geers Comprehensive</i>	32.8	29.4	34.2	2.6	10.0	2.5	32.1	5.0	23.8	40.5
<i>Russell Magnitude</i>	17.0	9.5	16.1	3.8	8.0	3.8	14.2	5.2	5.6	22.8
<i>Russell Phase</i>	24.6	24.0	27.2	2.8	2.5	3.6	25.3	3.0	20.4	30.2
<i>Russell Comprehensive</i>	26.7	23.5	28.3	1.9	4.5	2.1	26.2	2.8	21.5	30.9
<i>Knowles-Gear Magnitude</i>	55.3	70.2	63.3	5.0	19.0	11.2	62.9	11.7	43.5	82.4
<i>Knowles-Gear Phase</i>	67.2	57.2	85.8	18.4	37.1	45.4	70.0	33.6	14.2	125.9
<i>Knowles-Gear Comprehensive</i>	57.8	69.7	68.9	4.9	15.9	15.6	65.5	12.1	45.3	85.6
<b>Single Value Metrics</b>	Value [%]	Value [%]	min [%]	max [%]						
<i>Whang's inequality metric</i>	46.6	41.7	49.1	5.0	2.8	4.7	45.8	4.2	38.9	52.7
<i>Theil's inequality metric</i>	39.4	37.6	42.8	3.5	4.3	4.5	39.9	4.1	33.2	46.7
<i>Zilliacus error metric</i>	82.0	80.8	89.2	9.5	11.8	12.3	84.0	11.2	65.4	102.6
<i>RSS error metric metric</i>	70.6	76.9	78.6	7.1	14.8	12.3	75.3	11.4	56.4	94.2
<i>WIFac_Error</i>	57.1	55.4	61.1	3.5	4.8	5.2	57.9	4.5	50.3	65.4
<i>Regression Coefficient</i>	63.4	57.9	55.7	9.4	12.3	12.6	59.0	11.5	40.0	78.1
<i>Correlation Coefficient</i>	65.0	63.7	57.6	7.9	7.8	10.1	62.1	8.6	47.8	76.3
<i>Correlation Coefficient(NARD)</i>	71.3	72.7	65.1	6.2	5.4	8.4	69.7	6.7	58.6	80.8
<b>ANOVA Metrics</b>	Value	Value	min	max						
<i>Average</i>	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	-0.01	0.02
<i>Std</i>	0.22	0.25	0.24	0.02	0.05	0.04	0.23	0.04	0.17	0.29

**6.2.3.2.2 Case III: Last curve of each sets and the combination of sets**

An accurate analysis of 90<sup>th</sup> percentile envelopes created in section 6.2.2.1 has shown that the confidence envelopes based on the average responses of the respective sets/combination of tests are very similar and comparable to the corresponding confidence corridors based on the

respective Lab#5 for each of the two sets and also Lab#5 of Set1 for the combination of the two sets.

In order to compare them to the results obtained using as baseline curves the corresponding average responses, also the results obtained considering as baseline the latter mentioned curves are investigated.

The values of the comparison metrics obtained in this case are shown in Table 6.13 through Table 6.15. In particular, in the case with Set, the phase component (P) of the Knowles & Gear metric is characterized by high values compared to the score of the same component computed for Set 1 and also for the combination of both sets. Moreover, as in the cases with the previous baseline curves, the value of the Regression Coefficient is null due to the computation of an imaginary number which the program automatically considered as zero. This latter problem is due to a limitation of the Regression Coefficient as it is defined only for a limited domain. A further discussion of this aspect will be present at the end of the chapter.

Table 6.13. Values of the comparison metrics for Set#1 (baseline curve: Lab 5)

	Lab#1 (Set 1)	Lab#2 (Set 1)	Lab#3 (Set 1)	Lab#4 (Set 1)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
<b>MPC Metrics</b>	Value [%]	Value [%]				
<i>Geers Magnitude</i>	34.8	3.7	6	15.4	15.0	14.15
<i>Geers Phase</i>	24.9	13.6	15	17.7	17.8	5.03
<i>Geers Comprehensive</i>	42.7	14.1	16.1	23.4	24.1	13.04
<i>Geers CSA Magnitude</i>	34.8	3.7	6	15.4	15.0	14.15
<i>Geers CSA Phase</i>	24.9	13.6	15	17.7	17.8	5.03
<i>Geers CSA Comprehensive</i>	42.7	14.1	16.1	23.4	24.1	13.04
<i>Sprague-Geers Magnitude</i>	34.8	3.7	6	15.4	15.0	14.15
<i>Sprague-Geers Phase</i>	22.9	16.8	17.7	19.2	19.2	2.69
<i>Sprague-Geers Comprehensive</i>	41.7	17.2	18.6	24.6	25.5	11.25
<i>Russell Magnitude</i>	20.6	3.1	4.8	11	9.9	7.92
<i>Russell Phase</i>	22.9	16.8	17.7	19.2	19.2	2.69
<i>Russell Comprehensive</i>	27.3	15.1	16.2	19.6	19.6	5.51
<i>Knowles-Gear Magnitude</i>	64	35.2	39.1	42.7	45.3	12.87
<i>Knowles-Gear Phase</i>	40.4	16.1	7.1	7.1	17.7	15.73
<i>Knowles-Gear Comprehensive</i>	60.7	32.8	35.8	39.1	42.1	12.66
<b>Single Value Metrics</b>	Value [%]	Value [%]				
<i>Whang's inequality metric</i>	46	32.5	30.4	34	35.7	7.01
<i>Theil's inequality metric</i>	37.9	26.2	27.5	30.5	30.5	5.24
<i>Ziliacus error metric</i>	112	66.5	62.4	77.1	79.5	22.53
<i>RSS error metric</i>	88.9	53.3	56.7	65.7	66.2	16.04
<i>WIFac_Error</i>	55.5	42.5	44.2	48.4	47.7	5.79
<i>Regression Coefficient</i>	N/A	78.6	75.4	64.7	72.9	7.28
<i>Correlation Coefficient</i>	69.8	81.5	79.6	76.5	76.9	5.13
<i>Correlation Coefficient(NARD)</i>	75.1	86.4	85	82.3	82.2	5.03
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value
<i>Average</i>	-0.01	-0.01	-0.01	-0.02	0.01	0.01
<i>Std</i>	0.26	0.16	0.17	0.19	0.20	0.05
<i>T-test</i>	-1.31	-4.61	-5.56	-5.7		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Table 6.14. Values of the comparison metrics for Set#2 (baseline curve: Lab 5)

	Lab#1 (Set 2)	Lab#2 (Set 2)	Lab#3 (Set 2)	Lab#4 (Set 2)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
	Value [%]	Value [%]				
<b>MPC Metrics</b>						
<i>Geers Magnitude</i>	10.3	7.4	1.3	49.6	17.2	21.96
<i>Geers Phase</i>	33.2	17.2	25.2	42	29.4	10.64
<i>Geers Comprehensive</i>	34.7	18.7	25.2	65	35.9	20.48
<i>Geers CSA Magnitude</i>	10.3	7.4	1.3	49.6	17.2	21.96
<i>Geers CSA Phase</i>	33.2	17.2	25.2	42	29.4	10.64
<i>Geers CSA Comprehensive</i>	34.7	18.7	25.2	65	35.9	20.48
<i>Sprague-Geers Magnitude</i>	10.3	7.4	1.3	49.6	17.2	21.96
<i>Sprague-Geers Phase</i>	26.7	18.9	23.1	30.3	24.8	4.88
<i>Sprague-Geers Comprehensive</i>	28.6	20.3	23.1	58.1	32.5	17.40
<i>Russell Magnitude</i>	7.8	5.8	1.1	26.2	10.2	11.01
<i>Russell Phase</i>	26.7	18.9	23.1	30.3	24.8	4.88
<i>Russell Comprehensive</i>	24.6	17.5	20.5	35.5	24.5	7.87
<i>Knowles-Gear Magnitude</i>	72.8	67.4	63.9	96.8	75.2	14.84
<i>Knowles-Gear Phase</i>	12800	17700	22350	10650	15875.0	5228.53
<i>Knowles-Gear Comprehensive</i>	5226	7226	9125	4349	6481.4	2134.27
<b>Single Value Metrics</b>	Value [%]	Value [%]				
<i>Whang's inequality metric</i>	43.2	32.1	37.9	50.1	40.8	7.67
<i>Theil's inequality metric</i>	41	29.5	35.5	49.1	38.8	8.33
<i>Zilliacus error metric</i>	94.4	64	74.9	122.6	89.0	25.70
<i>RSS error metric metric</i>	86.2	61.2	71.4	122.6	85.4	26.87
<i>WIFac_Error</i>	56.8	48.2	56.7	64.3	56.5	6.58
<i>Regression Coefficient</i>	N/A	62.2	40.5	N/A	51.4	15.34
<i>Correlation Coefficient</i>	53	75.1	62.9	48.3	59.8	11.86
<i>Correlation Coefficient(NARD)</i>	66.8	82.8	74.8	58	70.6	10.64
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value
<i>Average</i>	0.03	0.02	0.03	0.02	0.03	0.01
<i>Std</i>	0.28	0.2	0.23	0.4	0.28	0.09
<i>T-test</i>	6.27	6.14	8.27	4.19		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Table 6.15. Values of the comparison metrics considering all tests (baseline curve: Lab 5 Set#1)

	Lab#1 (Set 1)	Lab#1 (Set 2)	Lab#2 (Set 1)	Lab#2 (Set 2)	Lab#3 (Set 1)	Lab#3 (Set 2)	Lab#4 (Set 1)	Lab#4 (Set 2)	Lab#5 (Set 2)	Mean <sup>§</sup>	Standard deviation <sup>§</sup>
<b>MPC Metrics</b>	Value [%]	Value [%]									
Geers Magnitude	34.8	12.3	3.7	9.4	6	3.1	15.4	52.3	1.8	15.4	17.12
Geers Phase	24.9	22.2	13.6	13.5	15	17.7	17.7	31.4	11.3	18.6	6.47
Geers Comprehensive	42.7	25.4	14.1	16.4	16.1	18	23.4	61	11.5	25.4	16.25
Geers CSA Magnitude	34.8	12.3	3.7	9.4	6	3.1	15.4	52.3	1.8	15.4	17.12
Geers CSA Phase	24.9	22.2	13.6	13.5	15	17.7	17.7	31.4	11.3	18.6	6.47
Geers CSA Comprehensive	42.7	25.4	14.1	16.4	16.1	18	23.4	61	11.5	25.4	16.25
Sprague-Geers Magnitude	34.8	12.3	3.7	9.4	6	3.1	15.4	52.3	1.8	15.4	17.12
Sprague-Geers Phase	22.9	21.6	16.8	16.7	17.7	19.2	19.2	26	15.3	19.5	3.44
Sprague-Geers Comprehensive	41.7	24.9	17.2	19.2	18.6	19.5	24.6	58.4	15.4	26.6	14.27
Russell Magnitude	20.6	9.1	3.1	7.2	4.8	2.6	11	27.1	1.5	9.7	8.77
Russell Phase	22.9	21.6	16.8	16.7	17.7	19.2	19.2	26	15.3	19.5	3.44
Russell Comprehensive	27.3	20.8	15.1	16.1	16.2	17.2	19.6	33.3	13.6	19.9	6.46
Knowles-Gear Magnitude	64	49.2	35.2	46	39.1	53.3	42.7	90.7	48.4	52.1	16.72
Knowles-Gear Phase	40.4	3.4	16.1	33.3	7.1	68.2	7.1	19.5	100	32.8	32.53
Knowles-Gear Comprehensive	60.7	44.9	32.8	44.2	35.8	56	39.1	83.2	60.2	50.8	15.92
<b>Single Value Metrics</b>	Value [%]	Value [%]									
Whang's inequality metric	46	39	32.5	29.9	30.4	35.2	34	48.4	28.6	36.0	7.10
Theil's inequality metric	37.9	33.8	26.2	26.4	27.5	29.8	30.5	44	23.8	31.1	6.46
Ziliacus error metric	112	89.7	66.5	62.5	62.4	73	77.1	125	59.9	80.9	23.44
RSS error metric metric	88.9	71.7	53.3	55.2	56.7	60.6	65.7	111	48	67.9	20.21
WIFac_Error	55.5	49.5	42.5	42.3	44.2	47.2	48.4	63.7	39.7	48.1	7.51
Regression Coefficient	N/A	55.6	78.6	76.9	75.4	71.2	64.7	N/A	83	55.7	12.64
Correlation Coefficient	69.8	70.4	81.5	81.6	79.6	75.8	76.5	62.3	84.7	75.8	7.14
Correlation Coefficient(NARD)	75.1	77.8	86.4	86.5	85	82.3	82.3	68.6	88.7	81.4	6.47
<b>ANOVA Metrics</b>	Value	Value									
Average	-0.01	-0.01	-0.01	-0.02	-0.01	-0.01	-0.02	-0.02	-0.04	0.02	0.01
Std	0.26	0.21	0.16	0.16	0.17	0.18	0.19	0.33	0.14	0.20	0.06
T-test	-1.31	-4.57	-4.61	-9.01	-5.56	-4.58	-5.7	-3.18	-18.89		

<sup>§</sup> The mean and standard deviation are computed considering the absolute values of the each metric.

Table 6.16 summarizes the results obtained from the analysis of the mean and the standard deviation of the metric values evaluated for each of the two sets of tests and also for their combination.

**Table 6.16. Average values of the means and standard deviations, and the corresponding acceptance criteria (baseline curve: Lab 5 of each set and Lab 5 Set#1 for the combination of the two sets)**

	Mean			STD			Average Mean	Average STD	Acceptable value	
	Set#1	Set#2	All	Set#1	Set#2	All			min	max
<b>MPC Metrics</b>	Value [%]	Value [%]	min [%]	max [%]						
<i>Geers Magnitude</i>	15.0	17.2	15.4	14.2	22.0	17.1	15.8	17.7	-13.6	45.3
<i>Geers Phase</i>	17.8	29.4	18.6	5.0	10.6	6.5	21.9	7.4	9.7	34.2
<i>Geers Comprehensive</i>	24.1	35.9	25.4	13.0	20.5	16.3	28.5	16.6	0.9	56.0
<i>Geers CSA Magnitude</i>	15.0	17.2	15.4	14.2	22.0	17.1	15.8	17.7	-13.6	45.3
<i>Geers CSA Phase</i>	17.8	29.4	18.6	5.0	10.6	6.5	21.9	7.4	9.7	34.2
<i>Geers CSA Comprehensive</i>	24.1	35.9	25.4	13.0	20.5	16.3	28.5	16.6	0.9	56.0
<i>Sprague-Geers Magnitude</i>	15.0	17.2	15.4	14.2	22.0	17.1	15.8	17.7	-13.6	45.3
<i>Sprague-Geers Phase</i>	19.2	24.8	19.5	2.7	4.9	3.4	21.1	3.7	15.0	27.2
<i>Sprague-Geers Comprehensive</i>	25.5	32.5	26.6	11.3	17.4	14.3	28.2	14.3	4.5	52.0
<i>Russell Magnitude</i>	9.9	10.2	9.7	7.9	11.0	8.8	9.9	9.2	-5.4	25.2
<i>Russell Phase</i>	19.2	24.8	19.5	2.7	4.9	3.4	21.1	3.7	15.0	27.2
<i>Russell Comprehensive</i>	19.6	24.5	19.9	5.5	7.9	6.5	21.3	6.6	10.4	32.3
<i>Knowles-Gear Magnitude</i>	45.3	75.2	52.1	12.9	14.8	16.7	57.5	14.8	32.9	82.1
<i>Knowles-Gear Phase</i>	17.7	15875.0	32.8	15.7	5228.5	32.5	5308.5	1758.9	2388.7	8228.3
<i>Knowles-Gear Comprehensive</i>	42.1	6481.4	50.8	12.7	2134.3	15.9	2191.4	721.0	994.6	3388.2
<b>Single Value Metrics</b>	Value [%]	Value [%]	min [%]	max [%]						
<i>Whang's inequality metric</i>	35.7	40.8	36.0	7.0	7.7	7.1	37.5	7.3	25.5	49.6
<i>Theil's inequality metric</i>	30.5	38.8	31.1	5.2	8.3	6.5	33.5	6.7	22.4	44.6
<i>Ziliacus error metric</i>	79.5	89.0	80.9	22.5	25.7	23.4	83.1	23.9	43.5	122.8
<i>RSS error metric metric</i>	66.2	85.4	67.9	16.0	26.9	20.2	73.1	21.0	38.2	108.1
<i>WIFac_Error</i>	47.7	56.5	48.1	5.8	6.6	7.5	50.8	6.6	39.8	61.8
<i>Regression Coefficient</i>	72.9	51.4	72.2	7.3	15.3	9.3	65.5	10.7	47.8	83.2
<i>Correlation Coefficient</i>	76.9	59.8	75.8	5.1	11.9	7.1	70.8	8.0	57.5	84.2
<i>Correlation Coefficient(NARD)</i>	82.2	70.6	81.4	5.0	10.6	6.5	78.1	7.4	65.8	90.3
<b>ANOVA Metrics</b>	Value	Value	min	max						
<i>Average</i>	0.01	0.03	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.03
<i>Std</i>	0.20	0.28	0.20	0.05	0.09	0.06	0.22	0.06	0.12	0.33

## **6.2.4 Analysis of the velocity time histories**

As earlier mentioned in this chapter, the use of back-calculated data (e.g., derived using either integration or differentiation) instead of the originally collected experimental or numerical data may cause some problems due to the inevitable accumulation of errors created by the particular numerical method used to manipulate the original curves. Hence, if accelerations are measured experimentally, then accelerations should be the basis of comparison; using velocities and displacements obtained by integrating the acceleration curve would otherwise accumulate error.

In order to investigate how these errors introduced by the manipulation of the original curves may affect the values of the various comparison metrics considered in this research, the integrals of the acceleration time histories are also compared.

Physically speaking, the integrated acceleration time histories represented the change in velocity of the vehicle's center of gravity along the direction of the corresponding components. In particular, only the cases with the first two baseline curves previously analyzed (i.e., the average response and first test of each set/combination of sets) are considered. The case with the third baseline curve is not considered for the analysis of the velocities as, from the results obtained using the acceleration time histories, it resulted to be very similar to that involving the average response.

As in the previous case involving the accelerations, for each of the two selected baseline curves, every one of the two sets and also their combination are considered.

### **6.2.4.1 90th percentile confidence envelope**

Similar to the previous case with the acceleration curves, the 90<sup>th</sup> percentile confidence envelope is created. For the reasons already described in section 6.2.2.1.2 the confidence envelopes are created based upon the time histories of the residuals.

In this case, the 90<sup>th</sup> percentile corridors are computed using only the average response and first test of respectively each set/combination of sets as baseline.

Figure 6.20 through Figure 6.22 show the velocity time histories and the confidence envelopes for respectively Set 1, Set 2 and their combination, considering each of the two types of baseline curve examined.

In all the analyzed cases, the velocity curves fitted in the corresponding confidence intervals better than in the case with the acceleration time histories. Also, similar to the case with the accelerations, the confidence intervals based on the average curves are narrower.

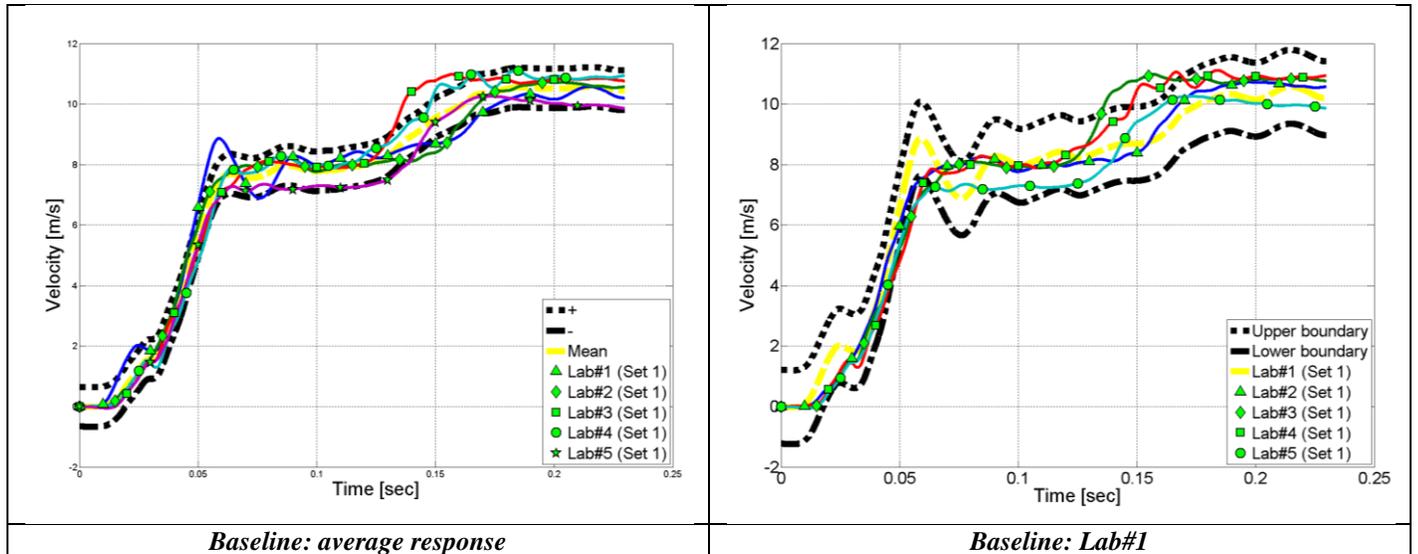


Figure 6.20: Velocity time histories and 90<sup>th</sup> percentile envelope for Set 1 based on the corresponding average curve (left) and the curve from test Lab#1 (right).

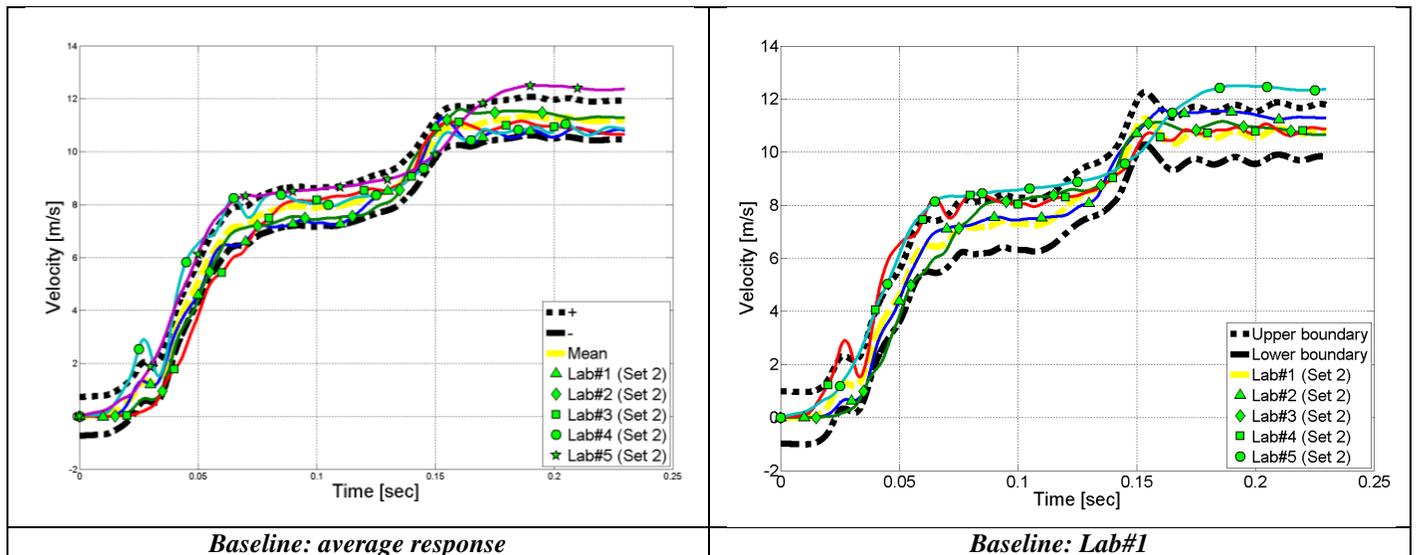


Figure 6.21: Velocity time histories and 90<sup>th</sup> percentile envelope for Set 2 based on the corresponding average curve (left) and the curve from test Lab#1 (right).

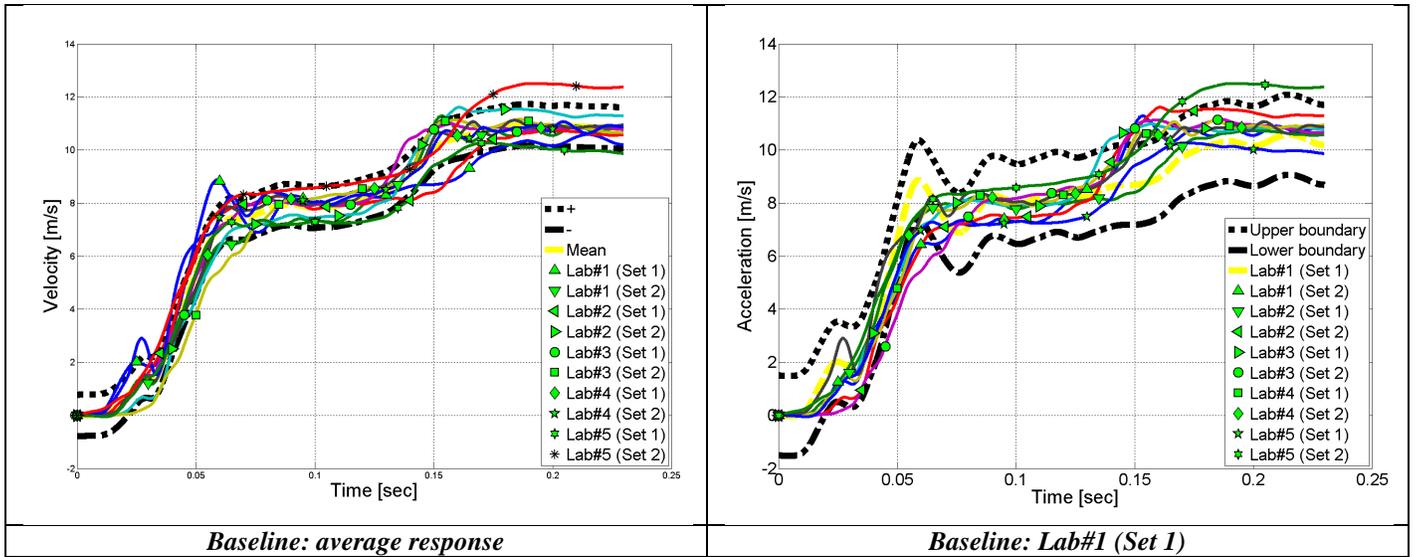


Figure 6.22: Velocity time histories and 90<sup>th</sup> percentile envelope for the combination of Set 1 and Set 2 based on the corresponding average curve (left) and the curve from test Lab#1 of Set 1(right).

#### 6.2.4.2 Comparison of curves

Table 6.17 summarizes the mean values of the comparison metrics computed considering both acceleration and velocity time histories for each of the two sets and also for their combination using the average response as baseline curve. Analogous results obtained considering the second type of baseline curves (i.e., Lab#1 of each set and Lab#1 from set 1 for the combination) are shown in Table 6.18. The detailed results obtained for each curve are shown in Appendix C.

**Table 6.17. Metrics mean values for each set/combination computed considering acceleration and velocity time histories (baseline: average response)**

	Set#1		Set#2		Combination	
	Acc.	Vel.	Acc.	Vel.	Acc.	Vel.
<b>MPC Metrics</b>	Value [%]	Value [%]				
<i>Geers Magnitude</i>	11.6	2.8	15.1	3.0	16.2	3.2
<i>Geers Phase</i>	10.1	0.1	12.8	0.1	13.3	0.2
<i>Geers Comprehensive</i>	16.6	2.8	22.2	3.0	22.0	3.2
<i>Geers CSA Magnitude</i>	11.6	2.8	15.1	3.0	16.2	3.2
<i>Geers CSA Phase</i>	10.1	0.1	12.8	0.1	13.3	0.2
<i>Geers CSA Comprehensive</i>	16.6	2.8	22.2	3.0	22.0	3.2
<i>Sprague-Geers Magnitude</i>	11.6	2.8	15.1	3.0	16.2	3.2
<i>Sprague-Geers Phase</i>	14.3	1.5	16.1	1.7	16.3	1.7
<i>Sprague-Geers Comprehensive</i>	20.0	3.4	24.7	3.8	24.5	3.9
<i>Russell Magnitude</i>	7.7	2.4	9.0	2.4	10.1	2.6
<i>Russell Phase</i>	14.3	1.5	16.1	1.7	16.3	1.7
<i>Russell Comprehensive</i>	15.1	2.6	17.5	3.0	17.6	3.0
<i>Knowles-Gear Magnitude</i>	36.9	5.6	52.8	6.0	46.5	6.4
<i>Knowles-Gear Phase</i>	29.9	12.7	56.4	24.2	45.9	18.7
<i>Knowles-Gear Comprehensive</i>	36.1	7.4	55.8	12.1	48.3	10.4
<b>Single Value Metrics</b>	Value [%]	Value [%]				
<i>Whang's inequality metric</i>	27.5	2.4	29.2	3.1	30.3	2.9
<i>Theil's inequality metric</i>	23.1	2.9	26.4	3.3	26.5	3.4
<i>Zilliacus error metric</i>	60.8	4.8	64.2	6.1	68.3	5.8
<i>RSS error metric metric</i>	49.4	5.7	57.4	6.7	58.3	6.8
<i>WIFac_Error</i>	38.2	5.6	43.1	6.5	42.9	6.6
<i>Regression Coefficient</i>	79.2	99.0	78.9	98.7	78.6	98.5
<i>Correlation Coefficient</i>	86.8	99.4	82.7	99.4	82.3	99.3
<i>Correlation Coefficient(NARD)</i>	89.9	99.9	87.2	99.9	86.7	99.8
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value
<i>Average</i>	0.00	0.02	0.01	0.03	0.01	0.02
<i>Std</i>	0.15	0.04	0.15	0.04	0.18	0.04

**Table 6.18. Metrics mean values for each set/combination computed considering acceleration and velocity time histories (baseline: Lab#1)**

	Set#1		Set#2		Combination	
	Acc.	Vel.	Acc.	Vel.	Acc.	Vel.
<b>MPC Metrics</b>	Value [%]	Value [%]				
<i>Geers Magnitude</i>	21.2	3.5	13.9	5.7	20.1	4.7
<i>Geers Phase</i>	28.8	0.4	27.3	0.3	34.9	0.6
<i>Geers Comprehensive</i>	36.1	3.5	32.2	5.7	40.9	4.8
<i>Geers CSA Magnitude</i>	21.2	3.5	13.9	5.7	20.1	4.7
<i>Geers CSA Phase</i>	28.8	0.4	27.3	0.3	34.9	0.6
<i>Geers CSA Comprehensive</i>	36.1	3.5	32.2	5.7	40.9	4.8
<i>Sprague-Geers Magnitude</i>	21.2	3.5	13.9	5.7	20.1	4.7
<i>Sprague-Geers Phase</i>	24.6	2.9	24.0	2.2	27.2	3.4
<i>Sprague-Geers Comprehensive</i>	32.8	4.6	29.4	6.2	34.2	6.1
<i>Russell Magnitude</i>	17.0	2.9	9.5	4.4	16.1	3.7
<i>Russell Phase</i>	24.6	2.9	24.0	2.2	27.2	3.4
<i>Russell Comprehensive</i>	26.7	3.7	23.5	4.5	28.3	4.7
<i>Knowles-Gear Magnitude</i>	55.3	9.3	70.2	9.1	63.3	11.9
<i>Knowles-Gear Phase</i>	67.2	44.9	57.2	30.0	85.8	53.9
<i>Knowles-Gear Comprehensive</i>	57.8	20.2	69.7	15.5	68.9	25.2
<b>Single Value Metrics</b>	Value [%]	Value [%]				
<i>Whang's inequality metric</i>	46.6	4.2	41.7	4.2	49.1	5.2
<i>Theil's inequality metric</i>	39.4	4.8	37.6	4.6	42.8	6.0
<i>Ziliacus error metric</i>	82.0	8.4	80.8	8.7	89.2	10.4
<i>RSS error metric metric</i>	70.6	9.8	76.9	9.6	78.6	12.2
<i>WIFac_Error</i>	57.1	9.3	55.4	8.9	61.1	11.4
<i>Regression Coefficient</i>	63.4	96.7	57.95	97.3	55.7	94.5
<i>Correlation Coefficient</i>	65.0	97.9	63.7	99.0	57.6	97.2
<i>Correlation Coefficient(NARD)</i>	71.3	99.6	72.7	99.8	65.1	99.4
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value
<i>Average</i>	0.01	0.02	0.01	0.04	0.01	0.02
<i>Std</i>	0.22	0.07	0.25	0.05	0.24	0.09

As can be seen, for both the cases with the two considered baseline curves, the values of the metrics computed using the velocity time histories obtained by integrating the original curves are always better than the corresponding values computed considering the accelerations. The only exception to this improvement of the metrics score is represented by the Average of the residuals

(group of the ANOVA metrics). In fact, the average value of the residuals is always greater in the case velocities are used. In order to further investigate the reasons of this behavior, the residuals distribution is analyzed in detail as described in the next section.

#### **6.2.4.3 Residual distribution**

As previously arranged for the analysis of the residuals in the case of the acceleration time histories, the probability density and the cumulative distribution functions are plotted for each of the considered cases. Figure 6.23 and Figure 6.24 show respectively the graphs of the probability density and cumulative distribution functions for Set 1, Set 2 and their combination.

Different from what was obtained with the accelerations, when comparing velocities, the distributions of the residuals show an anomalous behavior. The plots of the probability density functions are not centered respect to the null value (as it would be expected for a distribution with a zero mean) and in most cases their shape is not the typical bell-like of a normal distribution. A further confirmation of the fact that the mean residual is no longer null can be found also in the analysis of the cumulative distribution functions where the residual error corresponding to the 50<sup>th</sup> percentile is different than zero. Also, the cumulative distribution functions are characterized by a larger scatter of the curves than in the case with the accelerations.

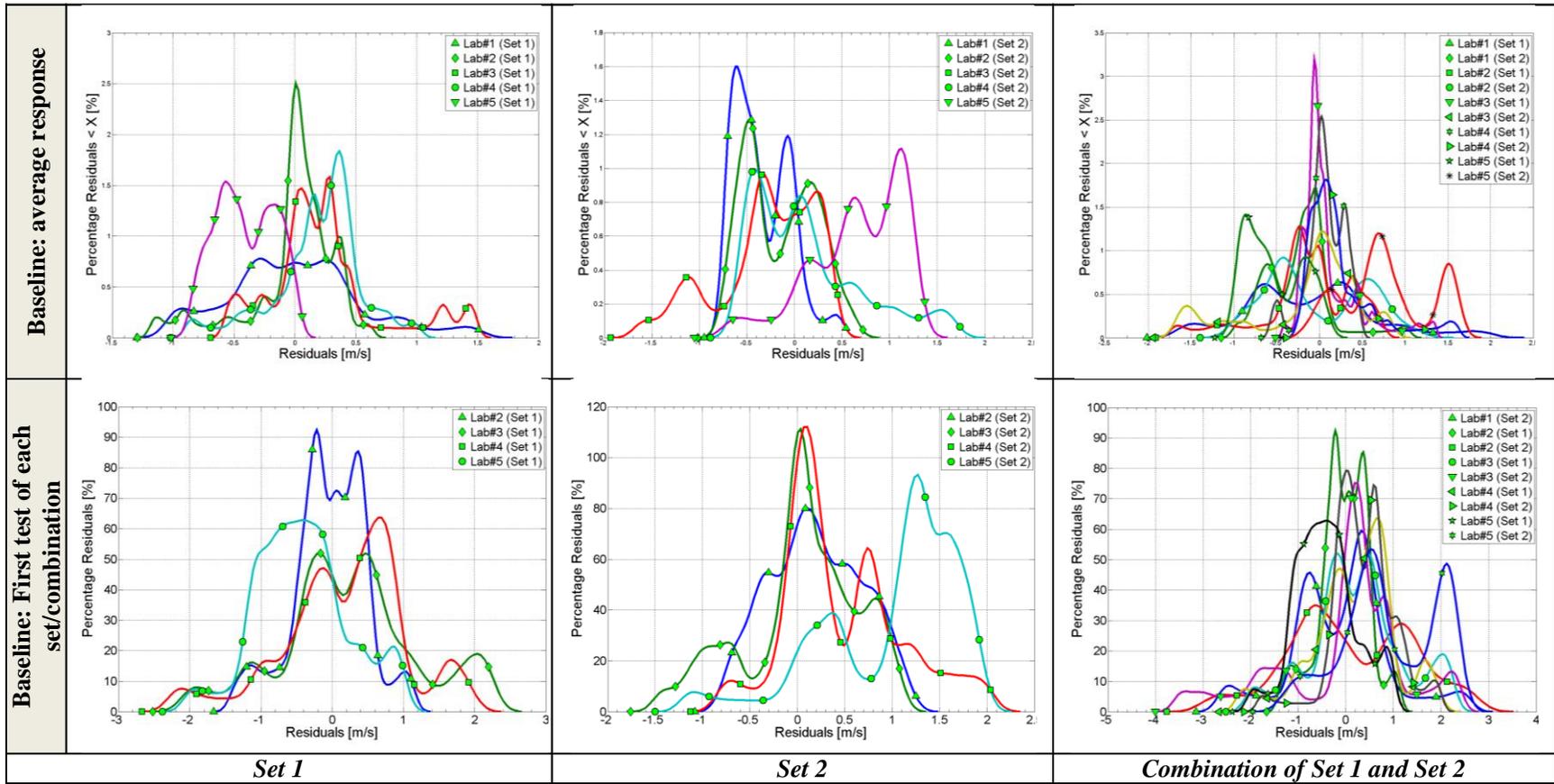


Figure 6.23: Probability density functions considering as baseline curve the average response (top) and the first test (bottom) of each set/combination.

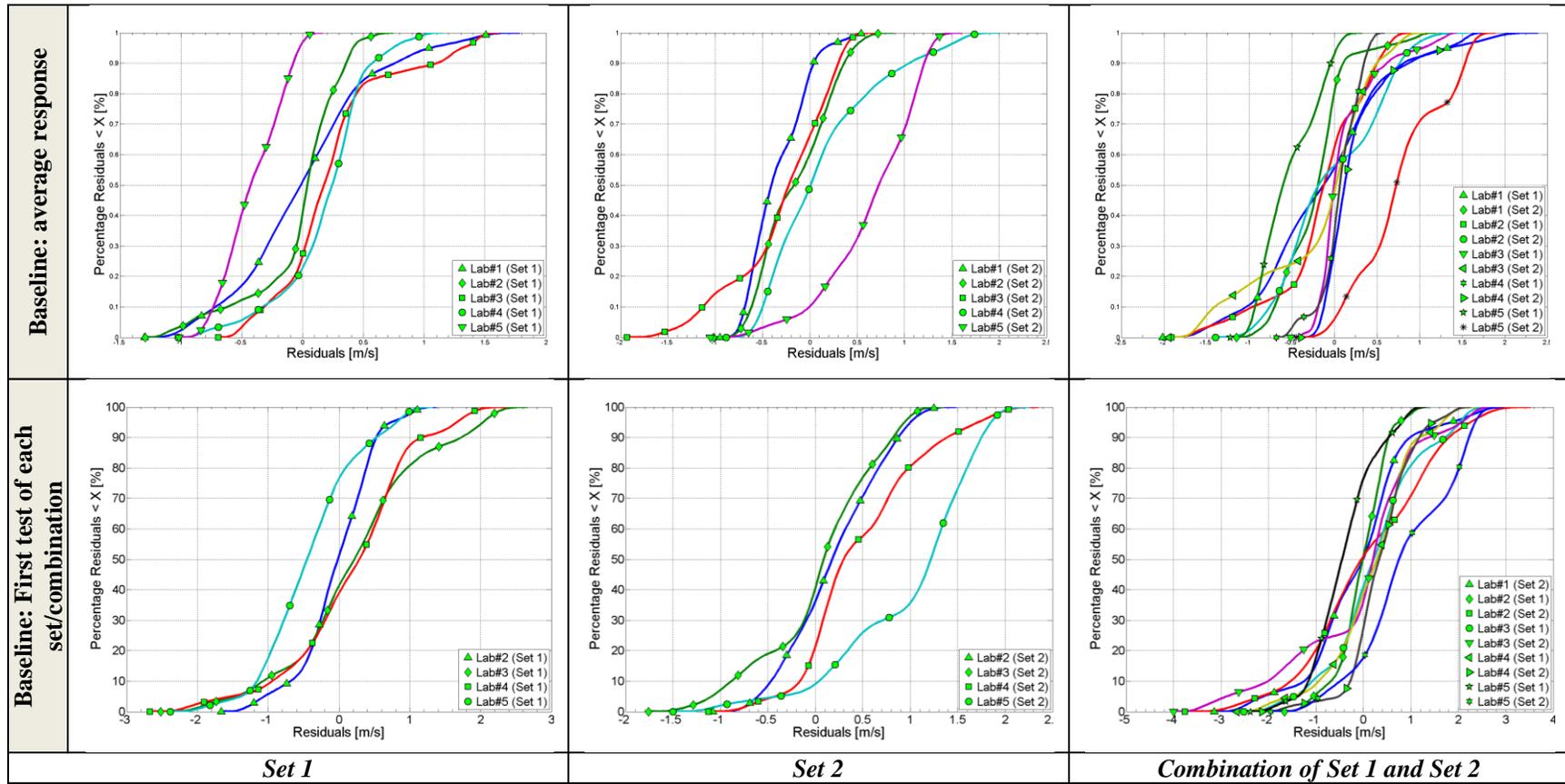


Figure 6.24: Cumulative distribution functions considering as baseline curve the average response (top) and the first test (bottom) of each set/combination.

### 6.2.5 Discussion of the results

From the analysis of the results obtained comparing the curves of the ten repeated full-scale crash tests four main considerations can be drawn. The first three of these considerations regard the analysis of the three cases involving the acceleration time histories, while the fourth consideration is related to the analysis of the velocity time histories.

- i. In all three cases involving the acceleration time histories (i.e., comparison versus the average response, the curves of the first test and the curve of the last test of the respective set/combination) the mean values of the metrics computed for each set/combination are similar or, at least the same order of magnitude. The only exception is for the Knowles&Gear metric in the case the curves of Set#2 are compared with respect to the curve from Lab 5. In this particular case, the values of the phase component are characterized by anomalously large numbers for each of the four different tests (Lab 1 through Lab 4). This eventually also affected the comprehensive value of the Knowles&Gear metric as the computation of the latter is directly related to the phase component.

The reason for these extremely bad scores for the phase components is not clear, especially considering that the same metric did not give any problem for the case involving either Set#1 or the combination of both the two sets.

- ii. In some comparisons, the Regression Coefficient resulted in imaginary numbers which the automatically software marked as null values (i.e., worst score possible for this type of metrics). As previously mentioned in Chapter 5, the reason for this imaginary score is a limitation in its formulation. In fact, according to a study performed by Russell [46], this metric can only be defined on a limited domain. In case the metric is computed outside this domain, the square root of a negative number is produced causing an imaginary score. In these instances, the corresponding values in the tables are marked as *N/A* and are not considered in the computation of the mean and the standard deviation.
- iii. In all three cases considered in this research, the values of the ANOVA metrics between each of the three considered groups of tests (Set#1, Set#2 and their combination) produced very similar results. This confirms the stable and reliable behavior of these

metrics in the comparison of curves characterized by the typical scatter of roadside safety tests.

- iv. Also, a comment can be drawn from the comparison of the mean values of the metrics computed for each set/combination for any of the three considered cases involving the acceleration time histories. A summary of the mean values computed for each set/combination in all the three cases is shown in Table 6.19. In fact, there is an evident analogy of the results for both Set 1 and the combination of Set 1 and Set 2 considering as baseline either the average response or the curve from Lab#5 (of Set 1). The very similar mean values obtained considering as baseline curves respectively the average response and the curve from Lab#5 of Set 1 can be motivated by the fact that the original curves are synchronized with respect to the latter cited baseline curve. This eventually resulted in an average response of the synchronized curves which is similar to that chosen as the reference in the synchronization (i.e., Lab 5 of Set#1). A further confirmation of this can be found in the fact that, when comparing the results for Set#2 in any of the three different cases, the correlation between the results obtained considering as baseline respectively the average response and the curve Lab 5 of Set#2 is not as good as for Set#1, for which the baseline curve is Lab 5 of Set#1 (which is the reference curve used during the synchronization procedure).

**Table 6.19. Mean values of the metrics of each set/combination computed for any of the three different considered cases (Case I: average response; Case II: Lab#1; Case III: Lab#5)**

	Set#1			Set#2			Combination (Set#1 + Set#2)		
	Case I	Case II	Case III	Case I	Case II	Case III	Case I	Case II	Case III
<b>MPC Metrics</b>	Value [%]	Value [%]	Value [%]						
<i>Geers Magnitude</i>	11.6	21.2	15.0	15.1	13.9	17.2	16.2	20.1	15.4
<i>Geers Phase</i>	10.1	28.8	17.8	12.8	27.3	29.4	13.3	34.9	18.6
<i>Geers Comprehensive</i>	16.6	36.1	24.1	22.2	32.2	35.9	22.0	40.9	25.4
<i>Geers CSA Magnitude</i>	11.6	21.2	15.0	15.1	13.9	17.2	16.2	20.1	15.4
<i>Geers CSA Phase</i>	10.1	28.8	17.8	12.8	27.3	29.4	13.3	34.9	18.6
<i>Geers CSA Comprehensive</i>	16.6	36.1	24.1	22.2	32.2	35.9	22.0	40.9	25.4
<i>Sprague-Geers Magnitude</i>	11.6	21.2	15.0	15.1	13.9	17.2	16.2	20.1	15.4
<i>Sprague-Geers Phase</i>	14.3	24.6	19.2	16.1	24.0	24.8	16.3	27.2	19.5
<i>Sprague-Geers Comprehensive</i>	20.0	32.8	25.5	24.7	29.4	32.5	24.5	34.2	26.6
<i>Russell Magnitude</i>	7.7	17.0	9.9	9.0	9.5	10.2	10.1	16.1	9.7
<i>Russell Phase</i>	14.3	24.6	19.2	16.1	24.0	24.8	16.3	27.2	19.5
<i>Russell Comprehensive</i>	15.1	26.7	19.6	17.5	23.5	24.5	17.6	28.3	19.9
<i>Knowles-Gear Magnitude</i>	36.9	55.3	45.3	52.8	70.2	75.2	46.5	63.3	52.1
<i>Knowles-Gear Phase</i>	29.9	67.2	17.7	56.4	57.2	15875.0	45.9	85.8	32.8
<i>Knowles-Gear Comprehensive</i>	36.1	57.8	42.1	55.8	69.7	6481.4	48.3	68.9	50.8
<b>Single Value Metrics</b>	Value [%]	Value [%]	Value [%]						
<i>Whang's inequality metric</i>	27.5	46.6	35.7	29.2	41.7	40.8	30.3	49.1	36.0
<i>Theil's inequality metric</i>	23.1	39.4	30.5	26.4	37.6	38.8	26.5	42.8	31.1
<i>Ziliacus error metric</i>	60.8	82.0	79.5	64.2	80.8	89.0	68.3	89.2	80.9
<i>RSS error metric metric</i>	49.4	70.6	66.2	57.4	76.9	85.4	58.3	78.6	67.9
<i>WIFac_Error</i>	38.2	57.1	47.7	43.1	55.4	56.5	42.9	61.1	48.1
<i>Regression Coefficient</i>	79.2	63.4	72.9	78.9	57.9	51.4	78.6	55.7	72.2
<i>Correlation Coefficient</i>	86.8	65.0	76.9	82.7	63.7	59.8	82.3	57.6	75.8
<i>Correlation Coefficient(NARD)</i>	89.9	71.3	82.2	87.2	72.7	70.6	86.7	65.1	81.4
<b>ANOVA Metrics</b>	Value	Value	Value						
<i>Average</i>	0.00	0.01	0.01	0.01	0.01	0.03	0.01	0.01	0.02
<i>Std</i>	0.15	0.22	0.20	0.15	0.25	0.28	0.18	0.24	0.20

- v. In order to assess how using data back-calculated from the originally collected time histories can affect the computation of comparison metrics, the use of the velocity time histories instead of the original acceleration time histories is analyzed. As shown in Table 6.17 and Table 6.18, the results obtained comparing each of the two sets of tests

and also their combination using the integrals of the accelerations (i.e., the change in velocity of the vehicle during the tests) indicated that, although the values of the metrics are characterized by smaller numbers, the values of the ANOVA metrics are anomalous. A further analysis of the residuals using both the probability density and cumulative distribution functions clearly showed that their distribution is not normal any longer. This confirmed the initial thesis that the integration of the original time histories implied an inevitable introduction of errors and, hence, only originally calculated/computed curves should be compared in order to avoid any manipulation of the original residuals.

### **6.3 Acceptance criteria and proposed metrics for roadside safety**

The acceptance criteria for each of the metrics computed in this analysis of the ten repeated tests are assessed considering the confidence intervals previously calculated using the average of the absolute values of the metrics. Each of the three cases, herein respectively named as Case I, Case II and Case III, are considered.

In particular, the upper boundary of the respective confidence intervals is considered for all the metrics except for the regression and correlation coefficients for which the lower boundary is used. In fact, for the latter metrics, the perfect match is represented by the highest obtainable values (i.e., 100) and not the null value as for all the other metrics. Table 6.20 summarizes the selected boundary values of the confidence acceptance intervals assessed for every metric during each of the three main comparison cases considered in this research. The last column of this table provides the average values of the selected boundary computed considering all the three cases. These average values are eventually proposed as acceptance criteria for the comparison of curves in roadside safety.

**Table 6.20. Boundary of the confidence intervals computed for each of the three considered comparison cases**

	Case I	Case II	Case III	Average Mean
<b>MPC Metrics</b>	max [%]	max [%]	max [%]	Value [%]
<i>Geers Magnitude</i>	42.9	31.8	45.3	40.0
<i>Geers Phase</i>	19.3	41.4	34.2	31.6
<i>Geers Comprehensive</i>	46.1	48.2	56.0	50.1
<i>Geers CSA Magnitude</i>	42.9	31.8	45.3	40.0
<i>Geers CSA Phase</i>	19.3	41.4	34.2	31.6
<i>Geers CSA Comprehensive</i>	46.1	48.2	56.0	50.1
<i>Sprague-Geers Magnitude</i>	42.9	31.8	45.3	40.0
<i>Sprague-Geers Phase</i>	20.6	30.2	27.2	26.0
<i>Sprague-Geers Comprehensive</i>	46.7	40.5	52.0	46.4
<i>Russell Magnitude</i>	23.9	22.8	25.2	24.0
<i>Russell Phase</i>	20.6	30.2	27.2	26.0
<i>Russell Comprehensive</i>	27.0	30.9	32.3	30.1
<i>Knowles-Gear Magnitude</i>	61.0	82.4	82.1	75.2
<i>Knowles-Gear Phase</i>	98.3	125.9	8228.3	112.1
<i>Knowles-Gear Comprehensive</i>	62.6	85.6	3388.2	74.1
<b>Single Value Metrics</b>	max [%]	max [%]	max [%]	Value [%]
<i>Whang's inequality metric</i>	39.5	52.7	49.6	47.2
<i>Theil's inequality metric</i>	35.3	46.7	44.6	42.2
<i>Ziliacus error metric</i>	96.8	102.6	122.8	107.4
<i>RSS error metric</i>	83.9	94.2	108.1	95.4
<i>WIFac_Error</i>	55.2	65.4	61.8	60.8
<i>Regression Coefficient<sup>§</sup></i>	63.3	40	47.8	50.4
<i>Correlation Coefficient<sup>§</sup></i>	75.3	47.8	57.5	60.2
<i>Correlation Coefficient(NARD)<sup>§</sup></i>	80.7	58.6	65.8	68.4
<b>ANOVA Metrics</b>	max	max	max	Value
<i>Average</i>	0.02	0.02	0.03	0.02
<i>Std</i>	0.25	0.29	0.33	0.29

<sup>§</sup>Lower boundary of the confidence acceptance interval is considered.

Some comments can be drawn from the analysis of the last column of Table 6.20:

- The suggested values for the different variations of the Geers metrics are very comparable. This does not surprise as they had a similar formulation. In particular, in this comparison cases, the original Geers and the Geers CSA formulations resulted in exactly the same scores. As for the other two MPC-type metrics, the Russell metric is characterized by similar values for both the magnitude and phase components.

Considering that the ten curves are properly synchronized, as confirmed by their visual comparison, scaling the magnitude score (which is the one expected to be more affected in these comparison cases) at the same level of the phase value could lead the user to misleading conclusions.

Similarly, also the Knowles-Gear metric is characterized by a confusing different scaling between the magnitude and the phase component. In this case, the phase score is even higher than the magnitude value by about 50 percent. Also, the Knowles-Gear metric resulted in extremely large values for the comparison of Set# 2 in the case of the third baseline considered in this analysis. As this problem is limited to that specific set of curves, this is probably due to a numerical approximation of small numbers during the computation of the metrics.

- In the comparison of some curves for any of the three analyzed baseline cases, the Regression Coefficient resulted in a null value (i.e., the worst possible score). As mentioned before, this problem is due to the computation of imaginary numbers, probably due to rounding errors. Note that the average values for the Regression Coefficient in the last column of Table 6.20 are computed neglecting these comparisons in which imaginary numbers are obtained.
- The average residual error component of the ANOVA metric is generally close to zero, while the standard deviation is close to 30 percent of the maximum peak. Also, with the results obtained in the analysis of the velocity time histories, the ANOVA metrics proved to be very effective in investigating the presence of any systematic error in the collection of data or in the execution of the test.

Because of the above mentioned reasons, the **Sprague&Geers** and the **ANOVA** are proposed as preferred metrics to compare curves in roadside safety. In particular, considering the average of the mean values which are computed taking into account the comparison versus three different representative baseline curves, the following acceptance criteria may be proposed:

- Sprague&Geers
  - Magnitude and Phase components,  $\leq 40$  percent (i.e, 0.4)
  - Comprehensive value,  $\leq 50$  percent (i.e, 0.5)

- ANOVA (values are normalized to the peak of the baseline curve)
  - Average of residuals,  $\leq 5$  percent (i.e., 0.05)
  - Standard deviation of the residuals,  $\leq 35$  percent (i.e., 0.35)

# Chapter 7

## Application Cases

Chapter 2 and Chapter 4 provided respectively a review of the comparison metrics available in the literature and a description of the characteristics implemented in the computer program specifically developed for the quantitative comparison of single or multiple couples of curves; while, in Chapter 6, metrics suitable to compare curves in roadside safety were investigated and their corresponding acceptance criteria were proposed based on the analysis of ten repeated full-scale crash tests. This chapter now describes the application of the implemented computer program for the comparison of curves in various cases.

In order to prove the effectiveness and advantages of the implemented method for comparing curves in various fields, the following cases were considered:

- Comparison of sets of experimental velocity waveforms
- Validation of a vehicle model for crash tests
- Validation of a knee-thigh-hip model for car crash injuries

The results obtained in the first case were compared to the independent subjective judgment of a group of experts, while the other two cases represent a typical application of comparison metrics to the validation process of numerical models. These three cases are described in the following sections of this chapter.

### 7.1 Comparison of velocity waveforms

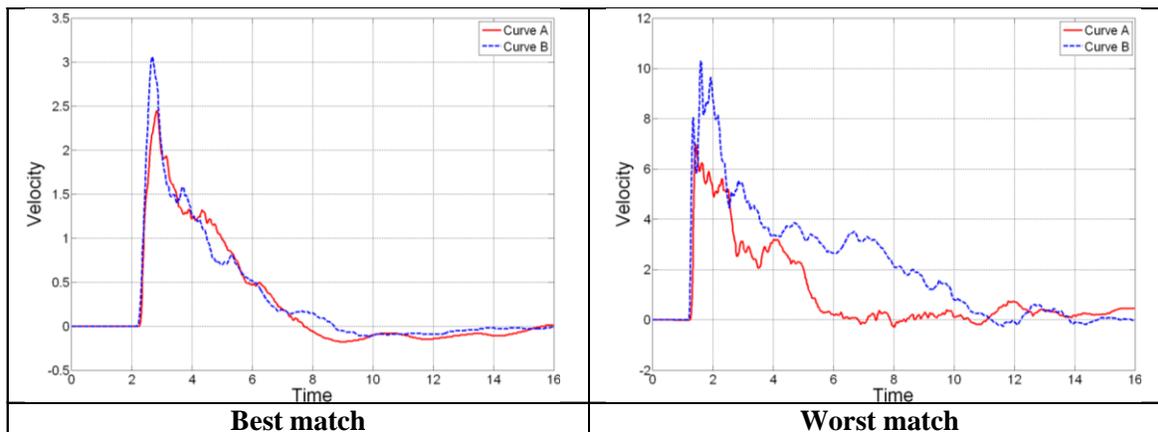
The main intent of using comparison metrics is that of mathematically quantifying the level of agreement in the comparison of a couple of curves. This intent obviously implies that the metrics scores are consistent with the subjective judgment of experts in that specific field which, in other words, means that comparison metrics should provide a quantified mimic expert evaluation. In order to assess the effectiveness of the metrics computed by the program, curves representing velocity waveforms were compared using a series of benchmark tests proposed by Schwer [43]. In particular, the quantitative results obtained using the implemented software

were confronted with the subjective judgment that the original author of the benchmark tests collected from experts in the field.

### 7.1.1 Subject matter expert opinions

The original benchmark tests described by Schwer consisted of five pairs of velocity waveforms obtained from repeated tests or symmetric measurements taken during single tests. Experts in the field in which these curves were collected were asked to rate the agreement of the different pairs of curves with a score ranging from zero to unity, with zero meaning poor agreement and one a perfect match.

In this case, the two pairs of curves which respectively obtained the best and worst judgment according to the subjective opinion have been investigated. In this manner, it was possible to analyze and compare the metrics scores obtained considering both the best and worst possible level of agreement for this type of curve. The two selected pairs of curves are shown in Figure 7.1.



**Figure 7.1: Plot of the velocity waveforms for the best (left) and the worst (right) pair.**

The individual responses of the 11 experts for the pairs of curves which obtained respectively the best and worst average score are summarized in the column graph shown in Figure 7.2. The columns with the error bars shown in the last group of the plot indicate the average response and the corresponding standard deviation for each of the two selected pairs of curves.

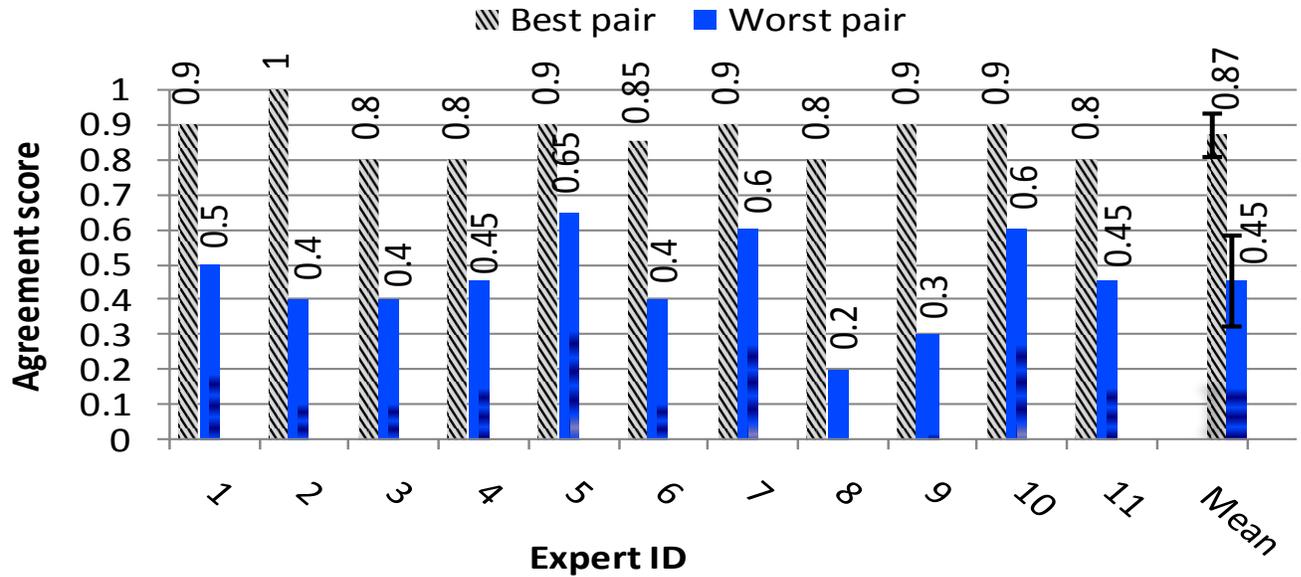


Figure 7.2: Expert evaluation for the best and worst pairs of velocity waveforms proposed by Schwer [43].

### 7.1.2 Metrics values

The two pairs of curves shown in Figure 7.1 are compared using all the metrics implemented in the computer program developed in this research.

Because most metrics are not symmetric, the choice of the baseline curve in each of the two considered pairs of curves may affect the resulting metric values. As in this case it was desired to analyze the correspondence between the quantitative assessment of the metrics with a subjective assessment, which could be based on any of the two curves of the pair, the average of the values obtained using as reference either of the two curves was computed. Table 7.1 shows the values of the comparison metrics computed considering both the possible configurations. The corresponding average values are also shown in the same table.

**Table 7.1. Comparison metrics and corresponding average values for the best and worst pairs**

	Best pair			Worst pair		
	Baseline 1	Baseline 2	Average	Baseline 1	Baseline 2	Average
<b>MPC Metrics</b>	Value[%]	Value[%]	Value[%]	Value[%]	Value[%]	Value[%]
Geers Magnitude	5.2	-4.9	5.05	61.8	-38.2	50
Geers Phase	2.4	2.4	2.4	11.9	11.9	11.9
Geers Comp.	5.7	5.5	5.6	62.9	40	51.45
Geers CSA Magnitude	5.2	-4.9	5.05	61.8	-38.2	50
Geers CSA Phase	2.4	2.4	2.4	11.9	11.9	11.9
Geers CSA Comp.	5.7	5.5	5.6	62.9	40	51.45
Sprague-Geers Magnitude	5.2	-4.9	5.05	61.8	-38.2	50
Sprague-Geers Phase	7	7	7	15.7	15.7	15.7
Sprague-Geers Comp.	8.7	8.6	8.65	63.8	41.3	52.55
Russell Magnitude	4.2	-4.2	4.2	30.1	-30.1	30.1
Russell Phase	7	7	7	15.7	15.7	15.7
Russell Comp.	7.2	7.2	7.2	30.1	30.1	30.1
Knowles-Gear Magnitude	19.6	18.3	18.95	50	44.8	47.4
Knowles-Gear Phase	1.7	1.8	1.75	4.6	4.8	4.7
Knowles-Gear Comp.	17.9	16.7	17.3	45.7	40.9	43.3
<b>Single Value Metrics</b>						
Whang's inequality metric	12.2	12.2	12.2	39.2	39.2	39.2
Theil's inequality metric	11.3	11.3	11.3	33.5	33.5	33.5
Zilliacus error metric	23.9	24.9	24.4	118	58.8	88.4
RSS error metric metric	23.1	21.9	22.5	87.6	54.2	70.9
WIFac_Error	20.3	20.3	20.3	53.5	53.5	53.5
Regression Coef.	96.8	97	96.9	N/A	68.4	68.4
Correlation Coef.	97.2	97.2	97.2	83.2	83.2	83.2
Correlation Coef.	97.6	97.6	97.6	88.1	88.1	88.1
<b>ANOVA Metrics</b>						
Average	-0.01	0.01	-0.01	-15	10	12.5
Standard deviation	0.06	0.05	0.06	19	13	16

### 7.1.3 Comparison of metrics values and experts opinions

Schwer pointed out that the collection of the experts' opinion was solicited asking them to indicate a level of agreement rather than quantifying the disagreement, as the former was considered to be more natural to interpret. Apart from the correlation and the two formulations of the regression coefficients, for which unity represents the best agreement, in all the other metrics the perfect match is indicated by a score equal to zero. In other words, with the exception of the three cited metrics, all the others give a measure of the error between the couple

of curves instead of a measure of their agreement as it was solicited to the experts. Hence, in order to compare the values of these metrics with the subjective evaluations, it was necessary to convert the agreement score into an error score by subtracting the former value to the unity. Table 7.2 shows the average values of the originally collected agreement scores and the corresponding back-calculated error scores for both the two considered pairs of curves. Note that, in order to compare them with the values of the comparison metrics, the agreement and error scores are expressed in percentage.

**Table 7.2. Average agreement and error scores for respectively the best and worst pairs of curves**

	<i>Agreement score [%]</i>	<i>Error score [%]</i>
<i>Best pair</i>	45	55
<i>Worst pair</i>	87	13

The column plots shown in Figure 7.3 summarize the comparison between the metrics values and the corresponding average subjective evaluation for both the agreement and the error scores in the case of the worst pair of curves. Similarly, Figure 7.4 shows the column plots for the comparison of the best pair of velocity curves.

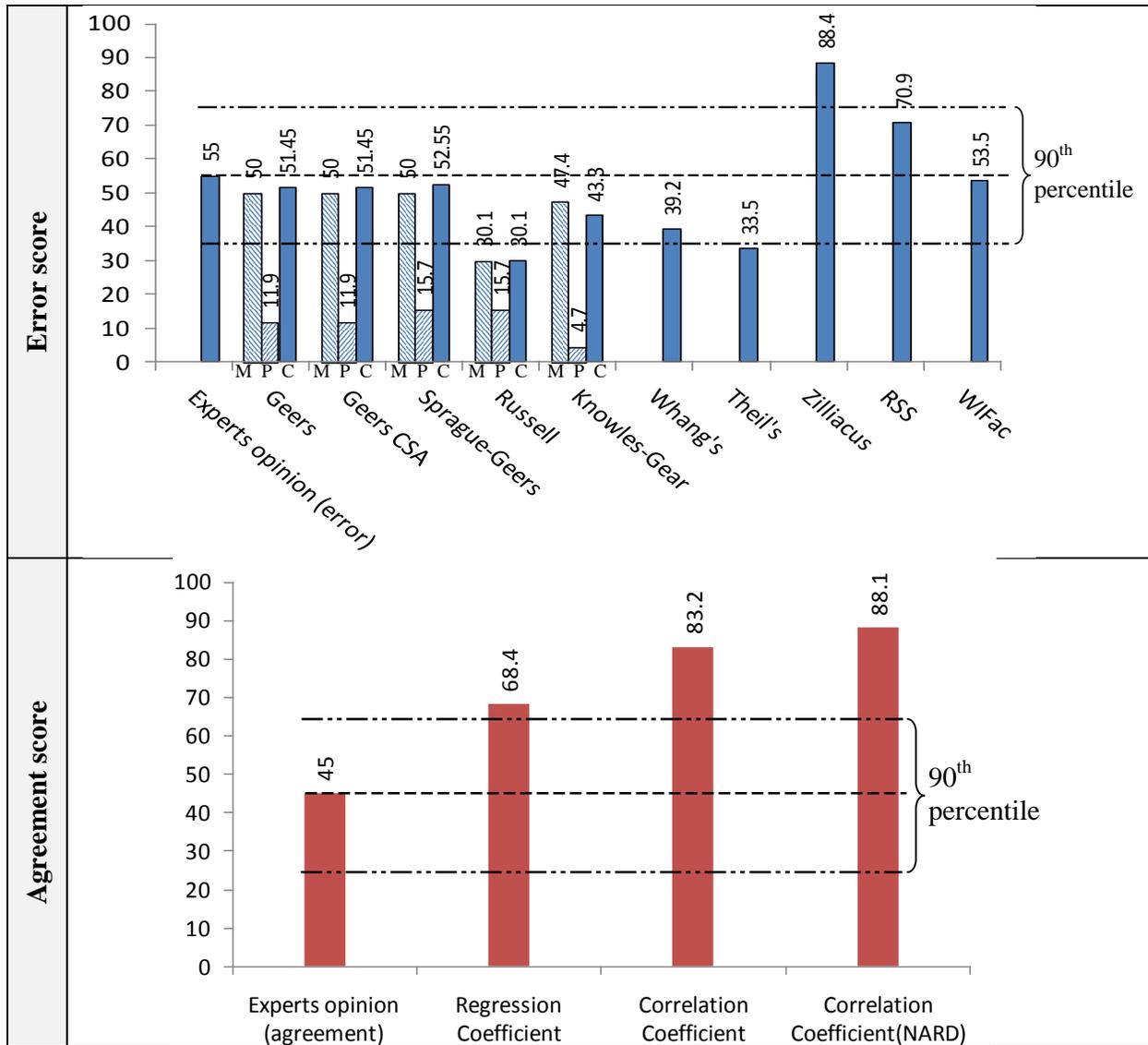


Figure 7.3: Column plots of the metrics values for the agreement (top) and error (bottom) scores in the case of the worst pair of curves.

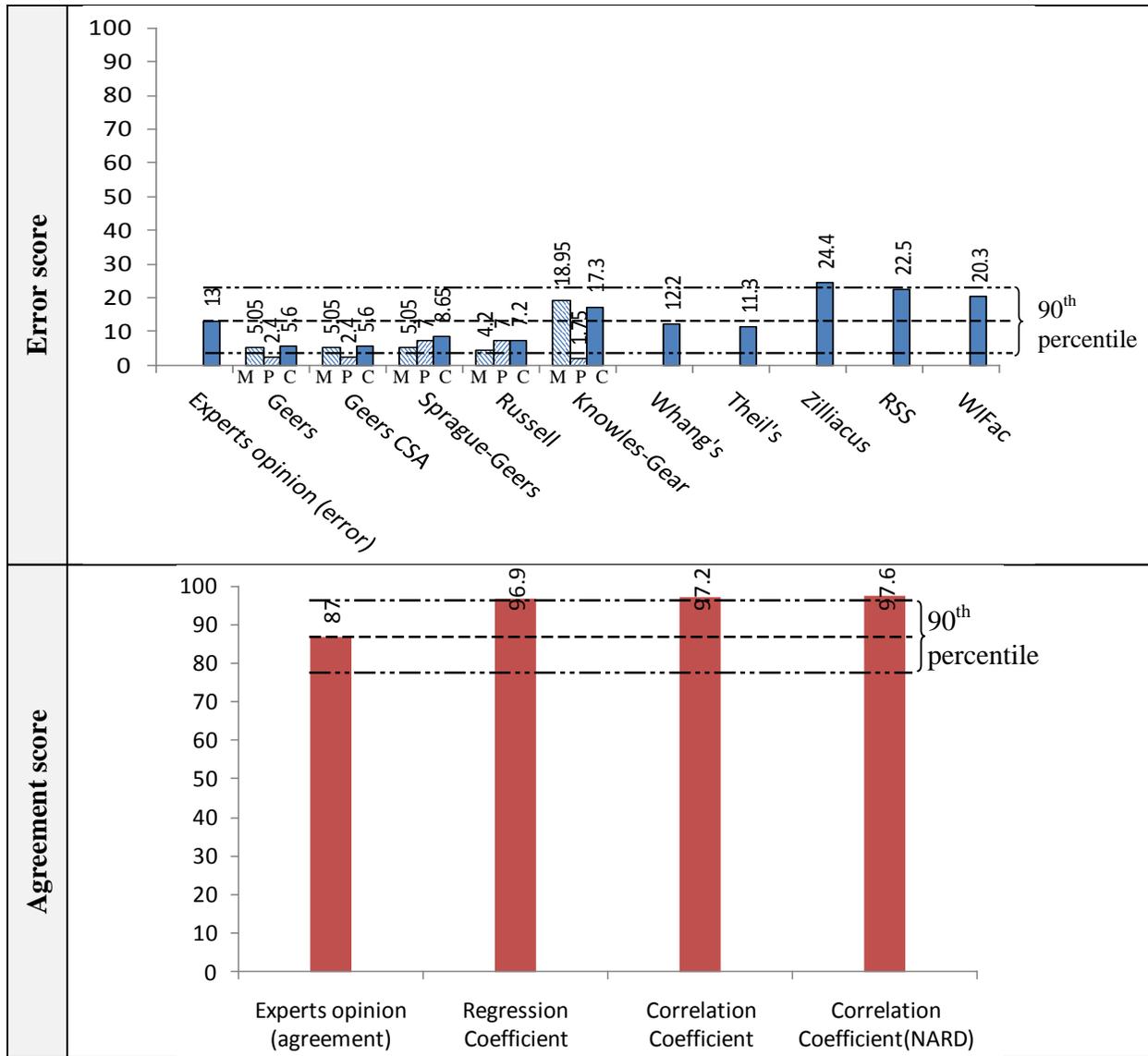


Figure 7.4: Column plots of the metrics values for the agreement (top) and error (bottom) scores in the case of the best pair of curves.

### 7.1.3.1 Error metrics

Considering the comparison of the worst pair of velocity curves, the values of all the MPC metrics with the exception of the Russell metric were in line with the average subjective assessment. In particular, the Russell metric appears to be more forgiving than the other MPC metrics. As for the differentiation between the magnitude and phase error, all the MPC metric were consistent as they all identify most of the error due to magnitude difference between the curves. Although this appears to be consistent with a common sense, as also pointed out by the original author of these benchmark tests, unfortunately no information was collected regarding

the specific manner the experts assessed the level of agreement/disagreement between the curves. Hence, it was not possible to quantify this aspect of the comparison between the subjective opinions and the metric values.

In the case with the comparison of the best pair of curves, all of the metrics except the Knowles&Gear were characterized by values more forgiving than the subjective opinions of the experts. In this instance, however, given the apparent good match of the curves, the low error scores may be considered to be equivalent for both the subjective and quantitative values. Differently from the case with the worst pair, in this case, for the Sprague&Geers and the Russell metrics the phasing difference was given more relevancy than the magnitude correlative.

As for the other single-value metrics which give an indication of the error, it appears evident that the scale they use to differentiate between a good and poor agreement was different from that used in the subjective evaluations. In fact, although in the case of the best couples of curves the single-value metrics give results in line with the subjective expert opinions, it was evident that they were either more demanding (Ziliacus and RSS) or more forgiving (Whang's and Theil's). Only the Weighted Integrated Factor (WIFac) metric use the same scale to differentiate in both the two considered comparisons.

#### **7.1.3.2 Agreement metrics**

As for the remaining three single-value metrics (i.e., the regression coefficient and the two variants of the correlation coefficient), they provide a direct measurement of the agreement between the pairs of curves. As previously mentioned, the original subjective opinions have been solicited as agreement scores as well. Note that, in one of the two comparisons computed for the worst pair of curves, it was not possible to evaluate the regression coefficient value due to the limited domain in which this particular metric is defined. Hence only the value obtained from the case with one baseline curve was considered.

The single-value agreement metrics and the subjective assessments were in good accordance for the case with the best pair of curves while, for the worst pair, the quantitative metrics seem to use a different scale respect to the experts average opinion. In fact, it was evident that the level of agreement assessed by the these metrics was shifted towards values bigger than 60 percent while the subjective score differentiates more between good and poor level of agreement by giving an

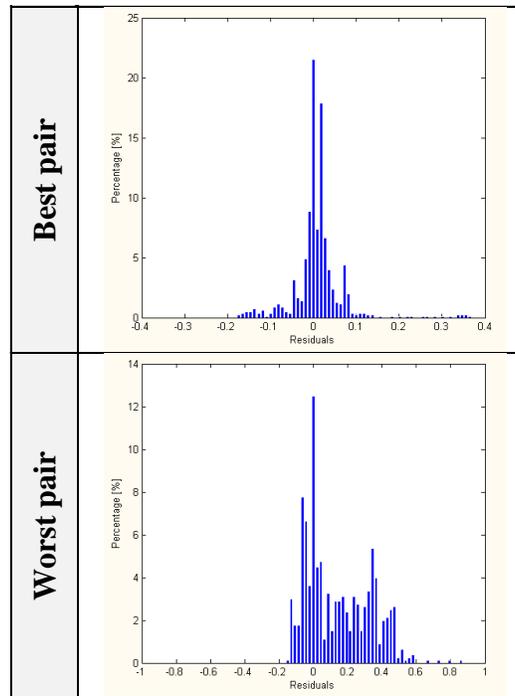
average score below 50 percent for the worst case. This can be interpreted by the fact that the experts expressed their opinion on the agreement between the couples of curves considering only as a reference the worst case available, while the metrics do not take them as a relative reference.

### **7.1.3.3 ANOVA metrics**

In the case of the ANOVA metrics it was not possible to make a direct comparison with the subjective opinions of the experts. In fact, the ANOVA metrics by definition give a measure of how random the residuals are distributed and this is not usually directly assessed in a subjective judgment which is more focused on the common topological aspects of the curves (e.g., common peaks and/or edges, phasing, etc).

Although a direct comparison was not possible, it was still possible to find some interesting analogies between the ANOVA values and the experts' average opinion. In fact, the values of the average and the standard deviation of the residuals were clearly smaller in the case of the couples of curves which were considered to be in the best agreement according to the subjective evaluation.

This was further confirmed by the plot of the residual histograms which, in the case of the pair of curves that obtained the worst subjective score, shows evident problems (Figure 7.5). Note that the histograms shown in Figure 7.5 refer to the case in which the first curve of the pair was used as baseline in the comparison. As inverting the order in which the two curves are compared involve the a change of sign of the residuals, the plots of histograms obtained considering the other curve of the pair as the baseline would simply be reflected around the vertical axis.



**Figure 7.5: Histogram of the residuals for respectively the best (top) and the worst (bottom) pairs of curves considering the first curve of the pair as the baseline.**

## 7.2 Validation of a FE model for full-scale crash test

As the main purpose for which this program has been created is the assessment of the reliability of numerical models (i.e., verification and validation), also a case of model validation was investigated. This section illustrates the application of the implemented method for the comparison of numerical and experimental time histories in order to validate a vehicle model to be used in roadside safety.

In particular, the numerical model simulated the set-up of the repeated crash tests described in section 6.1 of the previous chapter. The validation of the model was carried out by comparing the simulation results against the outcomes of two different full-scale experimental tests performed using the same type of barrier and similar vehicles. In the following sections, the various steps of the validation process are separately described respect to each of the two experimental tests, which will be addressed to as Test 1 and Test 2 accordingly. The vehicle used in the Test 1 and Test 2 were respectively a Fiat UNO and a Peugeot 206. In both the tests the vehicle complied with the requirements of the EN1317 standard for the 900-kg small car category.

Table 7.3 summarizes the vehicle and impact conditions for each of the two experimental cases and also the numerical simulation.

**Table 7.3. Vehicle type and impact conditions for each of the two tests and the simulation**

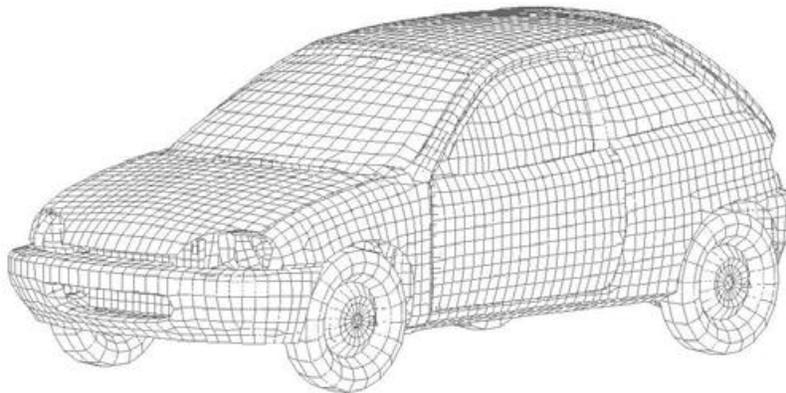
<b>General Information</b>	<b>Test 1 (Lab#1 of Set 2)</b>	<b>Test 2 (Lab#4 of Set 2)</b>	<b>FE simulation</b>
Vehicle:	F i a t U N O	Peugeot 205	Geo Metro (GM_R5)
<b>Impact Conditions</b>			
Vehicle Mass:	922 kg (2,033 lb)	862 kg (1,900 lb)	925 kg (1,896 lb)
Speed:	100.33 km/h (62 mph)	100.4 km/h (62 mph)	100 km/h (62 mph)
Angle:	20deg	20deg	20 deg.
Impact Point:	10 m (33 ft) from beginning	10.7 m (35 ft)from beginning	4.5 m (15 ft) from beginning

### 7.2.1 Numerical model

The simulation was performed using the explicit non-linear finite element code LS-DYNA [9].

#### 7.2.1.1 Vehicle

The vehicle was modeled using an improved version of the publicly available model of the GeoMetro vehicle (Figure 7.6) originally developed by the NCAC [67]. The main changes/improvements to the original vehicle model involved both the front and rear suspensions and the steering system. Another minor improvement on the original model regarded the remeshing of the wheels (both the tires and the rims). Also, the total mass of the model was increased in order to comply with the specifications of the European standard EN 1317 [33] for a small vehicle (i.e., a 900-kg car).



**Figure 7.6: FE model of the Geo-Metro vehicle used to simulate the full-scale crash test.**

The various modifications applied to the original model were verified through a qualitative procedure by comparing the behavior of both the original and modified vehicle models in some simple cases. Table 7.4 shows a list of the phenomena investigated to verify the improvements to the original model. For the details of the modifications made to these components of the model and the corresponding verification, refer to [68].

**Table 7.4. List of phenomena investigated for the verification of improvements to the Geo-Metro model**

<b>Phenomena</b>	<b>Summary</b>
Front suspension system (kinematics and independence from steering system)	Compression load applied to a single wheel w/ vehicle raised from ground. Negligible steering motion and independent compression of the suspension under load.
Steering system properties i) Ackerman principle ii) Steep stop	Steering torque applied to a single wheel w/ vehicle raised from ground. Internal wheel rotation greater than external wheel.
Front suspensions response and steer stability (curb test)	Impact against an asymmetrical curb (left wheel) tilted by 6° at 12 km/h. Independent and stable behavior of front suspensions and prompt realignment of the front wheels.
Rear suspension system (kinematics)	Compression load applied to both rear wheels w/ vehicle raised from ground. Correctness of the basic suspension kinematics.
Rear suspensions response (curb test)	Impact against an asymmetrical curb (left wheel) tilted by 6° at 12 km/h. Independent and stable behavior of rear suspensions.

Although the vehicle numerical model does not represent in full detail the actual vehicle used during any of the two full-scale tests, it has similar geometrical and inertial properties and falls into the same category, which is that of small cars. Table 7.5 summarizes the inertial properties of the vehicle numerical model.

**Table 7.5. Inertial properties of the Geo-Metro FE model used in the simulations**

<b>Mass [kg]</b>	925
<b>I<sub>xx</sub> [kg*m<sup>2</sup>]</b>	285.7
<b>I<sub>yy</sub> [kg*m<sup>2</sup>]</b>	1,194
<b>I<sub>zz</sub> [kg*m<sup>2</sup>]</b>	1,361

### 7.2.1.2 Barrier

The model of the concrete barrier is shown in Figure 7.7. Due to the simplicity of the geometry of the actual design and its strength, the barrier has been modeled as a monolithic wall made of solid elements. In order to improve the contact between the vehicle and the barrier model, the external surface of the solid elements facing the front side of the barrier model were overlapped by shell elements defined with the LS-DYNA material null (Mat\_009). As during the various tests independently carried out by various test agencies the barrier was always firmly hold in its position by placing it against concrete parapets or massive concrete blocks, the corresponding numerical model rigidly anchored to the ground. The total length of the barrier model was 15 m.

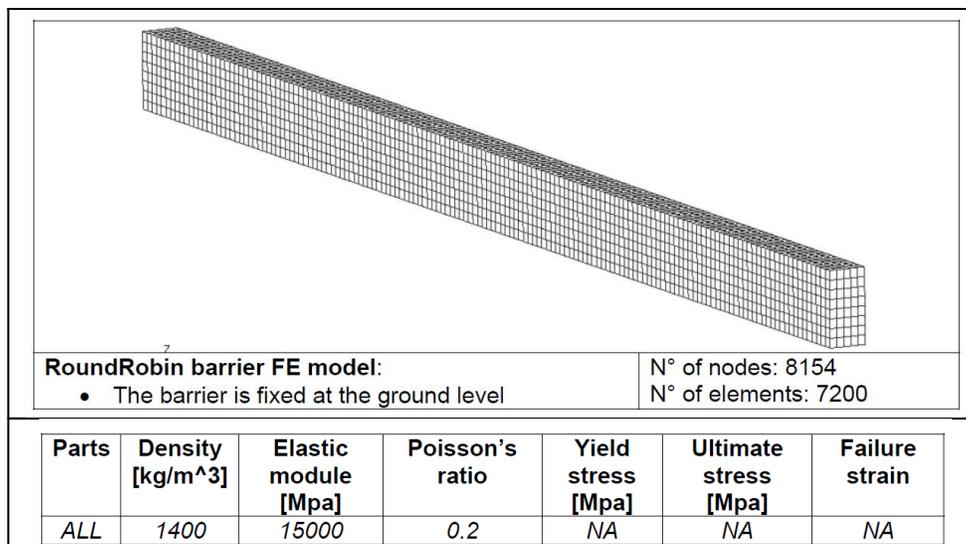


Figure 7.7: Barrier model with a summary of the model and statistics and material properties.

In the case of the barrier, as it can be considered at full extent a rigid wall, no tests were performed to verify any of its components (i.e., concrete blocks and connections).

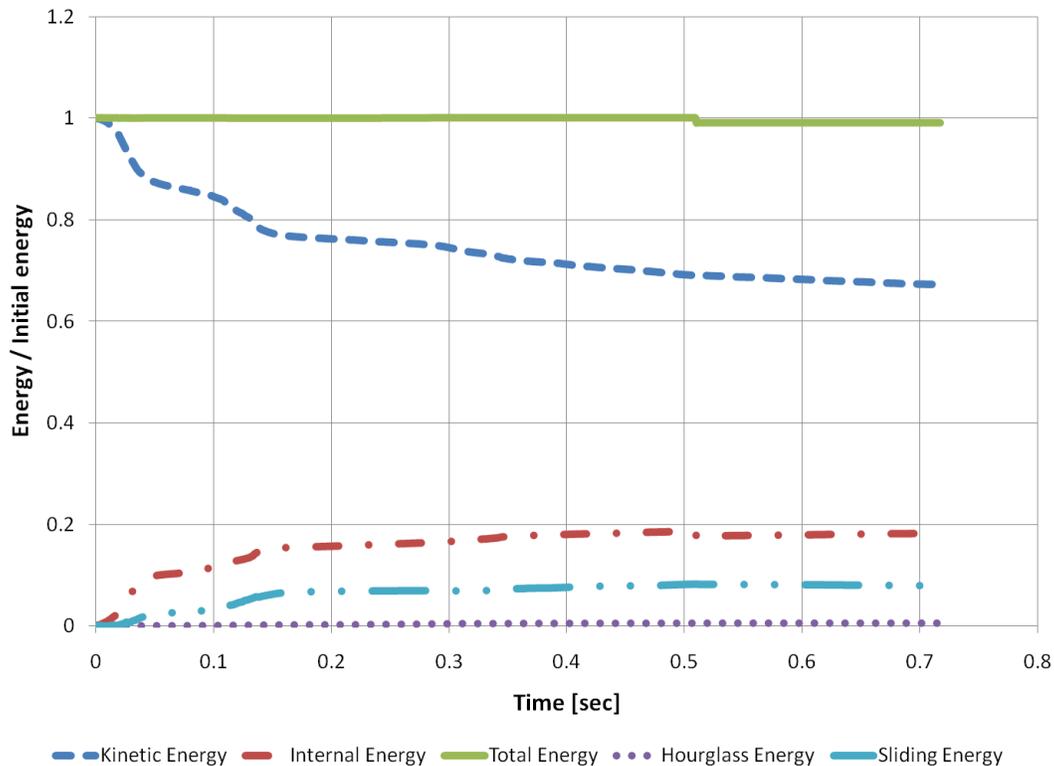
## 7.2.2 Simulation results

### 7.2.2.1 Energy and mass conservation

In order to check that the model was stable and capable to produce physical results, some controls were made considering basic energy and mass balances; in particular what was checked is that the global energies and mass involved in the simulation vary within a reasonable range. These controls serve to make sure that the model does not present any numerical error and it complies with the basic general physics laws; hence, passing these checks has to be considered a

*conditio sine qua non* which is necessary to comply with but is not yet sufficient to consider the model as validated. In fact, these controls were made independently from the results of the experimental tests.

From the energy point of view, as in this case no external energy was added to the system, the value of the total energy should remain constant. The time histories of the global energies (normalized respect to the initial energy) involved in the simulation are shown in Figure 7.8. As can be seen, the total initial energy stays constant for all the duration of the simulation. At the beginning of the simulation, the total energy of the system was given purely by the kinetic energy of the vehicle.



**Figure 7.8: Plot of the global energy time histories (normalized respect to the initial energy).**

The decrease of the kinetic energy during the impact was compensated by an equivalent increase of the internal energy and the energy dissipated by the frictional forces. Moreover, the fictitious hourglass energy could be considered practically null. Similarly, the total mass of the system should stay constant and any increase of the total mass of the model due to the application of the mass-scaling technique during the simulation should be negligible respect to the initial mass.

The verification of the energy and mass conservation was performed according to the criteria shown in Table 7.6.

**Table 7.6. Analysis solution verification table for the numerical model**

<b>Verification Evaluation Criteria</b>	<b>Change (%)</b>	<b>Pass?</b>
Total energy of the analysis solution (i.e., kinetic, potential, contact, etc.) must not vary more than 10 percent from the beginning of the run to the end of the run.	-1	✓
Hourglass Energy of the analysis solution at the end of the run is less than five percent of the total initial energy at the beginning of the run.	0.5	✓
The part/material with the highest amount of hourglass energy at any time during the run is less than five percent of the total initial energy at the beginning of the run.	0.5	✓
Mass added to the total model is less than five percent of the total model mass at the beginning of the run.	0	✓
The part/material with the most mass added had less than 10 percent of its initial mass added.	0	✓
The moving parts/materials in the model have less than five percent of mass added to the initial moving mass of the model.	0	✓
There are no shooting nodes in the solution?	No	✓
There are no solid elements with negative volumes?	No	✓

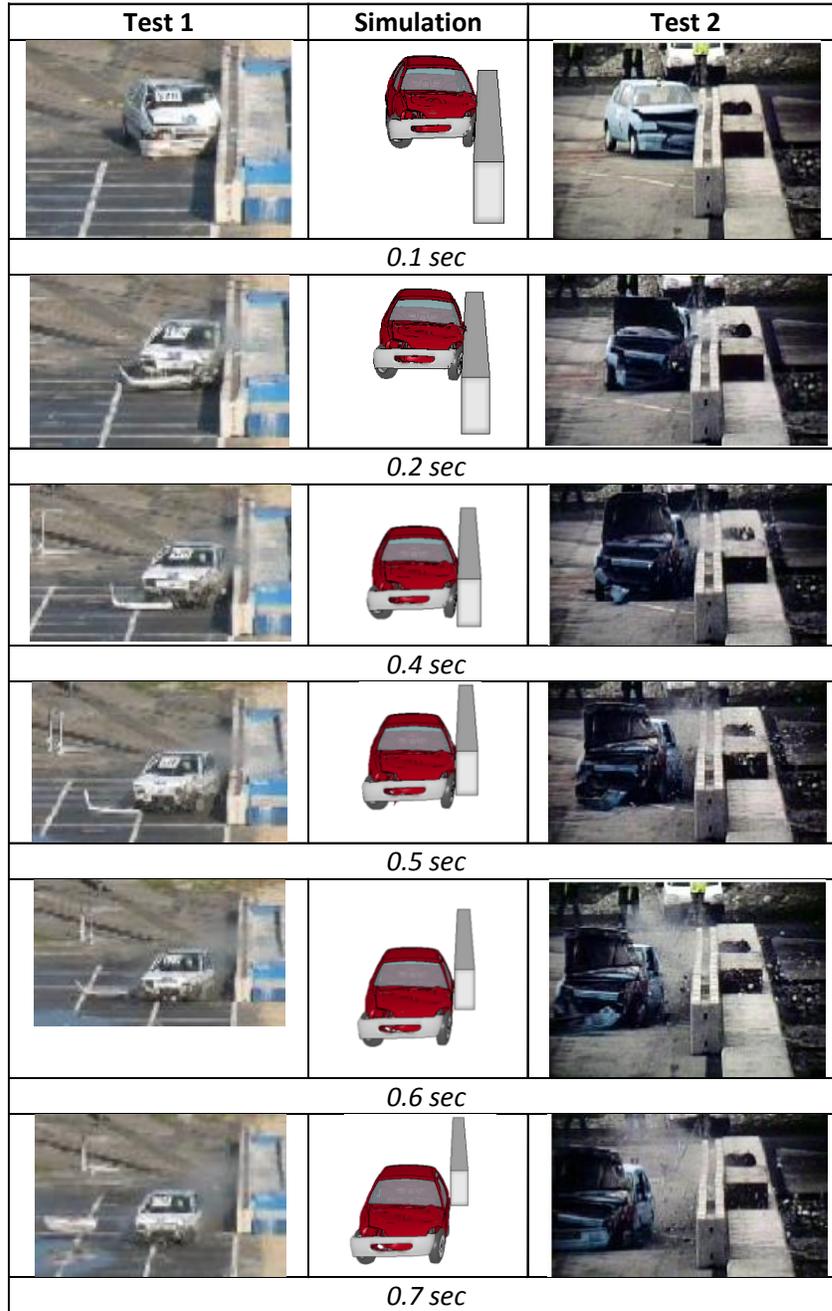
As all the criteria listed in Table 7.6 passed, the model could be considered stable.

### **7.2.2.2 Qualitative validation**

The qualitative assessment of the vehicle response obtained from the numerical simulation compared well with both the two full-scale tests. In both cases, the numerical model was able to replicate the general vehicle kinematics and the timing of the actual experimental test during the first phase of the impact. Figure 7.9 shows a sequential comparison of the vehicle behavior between respectively Test 1 and Test 2, and the numerical analysis.

In the second phase of the impact a different behavior in the vehicle kinematics occurred between the numerical solution and the two experimental tests, as in the numerical model the vehicle tended to remain parallel to the barrier during the entire test while in the actual test it was re-bounced. In Test 1 the vehicle eventually settled back parallel to the barrier. The reason for this difference in the vehicle trajectory of the numerical model was motivated by the turn of the right front wheel towards the barrier which was caused by the failure of the steering system when the opposite wheel hits against the rigid barrier. Although in both these two particular experimental tests it did not happen, this phenomenon is quite usual during a harsh impact in

which the crushing of the suspension arm may set the hitting wheel itself turned in such a way that it pulls the steering rod connected to the opposite front wheel. Note that the comparison of the experimental and numerical curves was performed on a time interval smaller than the entire period which was simulated and is shown in the sequence in Figure 7.9. See the section about the preprocessing of the curves for more details.



**Figure 7.9: Vehicle kinematics for the experimental tests (left and right) and the FE simulation (center).**

### 7.2.2.3 *Quantitative validation*

After the stability of the numerical model was checked, it was possible to proceed to validate it by assessing a quantitative comparison of the time histories using the Sprague&Geers and ANOVA comparison metrics which were computed using the computer program developed in this research and previously described in Chapter 4.

The four time histories available from Test 1 (X, Y, Z accelerations and Yaw rate) were compared with the corresponding curves from the numerical simulation. Similarly was performed also considering the six time histories from Test 2. The data were collected at the vehicle center of gravity in both the tests and the simulation.

The original numerical input curves were initially filtered using a SAE 180 class filter and, before the metrics were calculated, both the experimental and numerical curves were preprocessed using the software preprocessing options. The units and, for certain channels also the sign, of the numerical time histories needed to be adjusted in order to be consistent with the experimental curves. In particular, the numerical acceleration channels were converted from  $mm/s^2$  to  $g$ 's and the sign of the Y, Z acceleration and the Yaw rate and Pitch rate (for Test 2) were inverted due to a different reference system used in the numerical model. The experimental time histories from both Test 1 and Test 2 were manually trimmed as they were characterized by a considerable flat head due an early triggering during the collection of the data and an excessive flat tail which collected unnecessary data also after the impact was practically concluded. After the experimental curves were manually trimmed, the program automatically shifted the time vector to the origin (i.e., the beginning of the impact event started at time zero). Note that, also the tail of the simulation time histories was manually trimmed after 0.4 sec, in order to consider only the effective interval of impact. Eventually, both the experimental and numerical curves were filtered using the SAE 180 Class filter option and each channel was synchronized using the dedicated preprocessing option based on the minimization of the area of the residuals. Figure 7.10 and Figure 7.11 show a comparison of the original and preprocessed curves for each of the input channels considering Test 1, (the units and sign of the original numerical curves were already adjusted to be consistent with the experimental curves). From the graph of the preprocessed Yaw rate in Figure 7.13 it is clearly evident that the time interval selected to compare the curves was large enough to adequately cover all the phases of the

impact. In fact, the selected interval completely contained the curve since the velocity increases from zero till the time it was stabilized back to a null value. Similar considerations can be drawn considering the six channels when using the time histories from Test 2. The comparisons between the original and preprocessed curves for the six channels considering Test 2 are shown in Figure 7.12 and Figure 7.13.

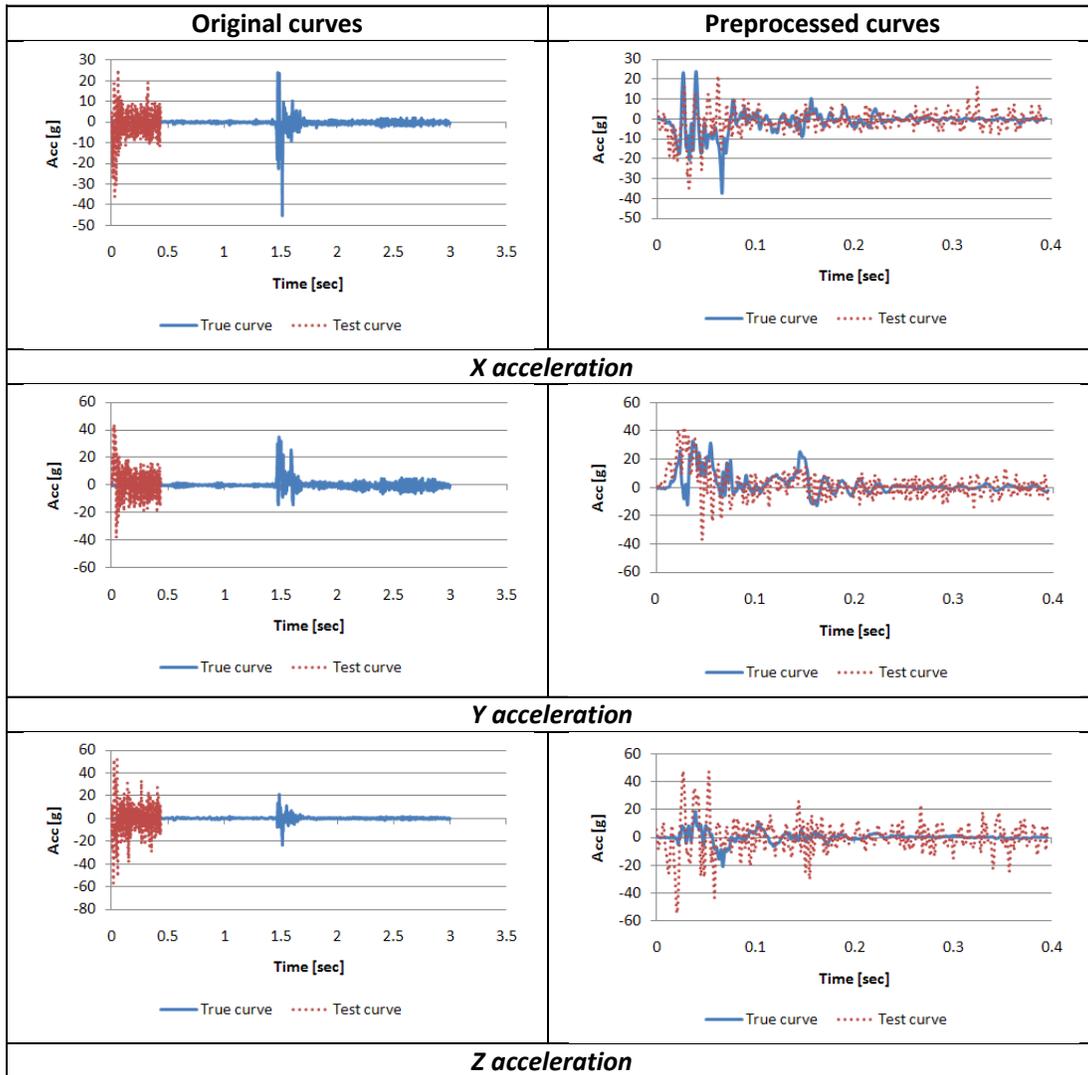


Figure 7.10: Original and preprocessed acceleration curves considering (case with Test 1).

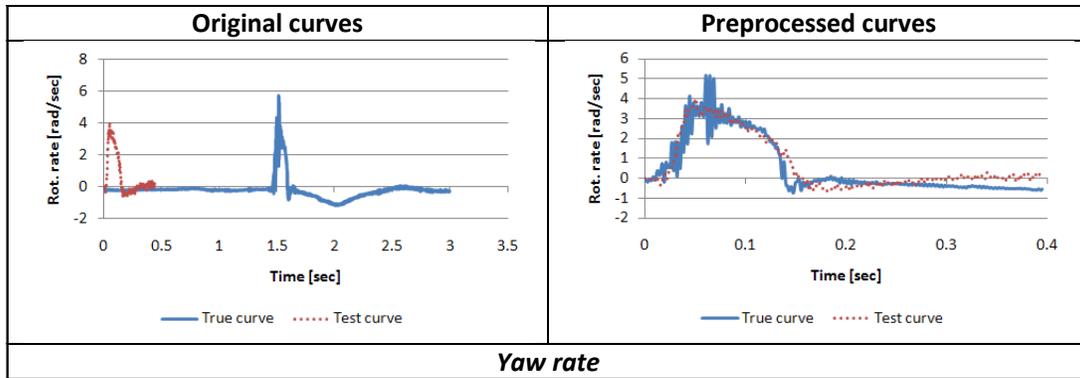


Figure 7.11: Original and preprocessed Yaw rate curve considering (case with Test 1).

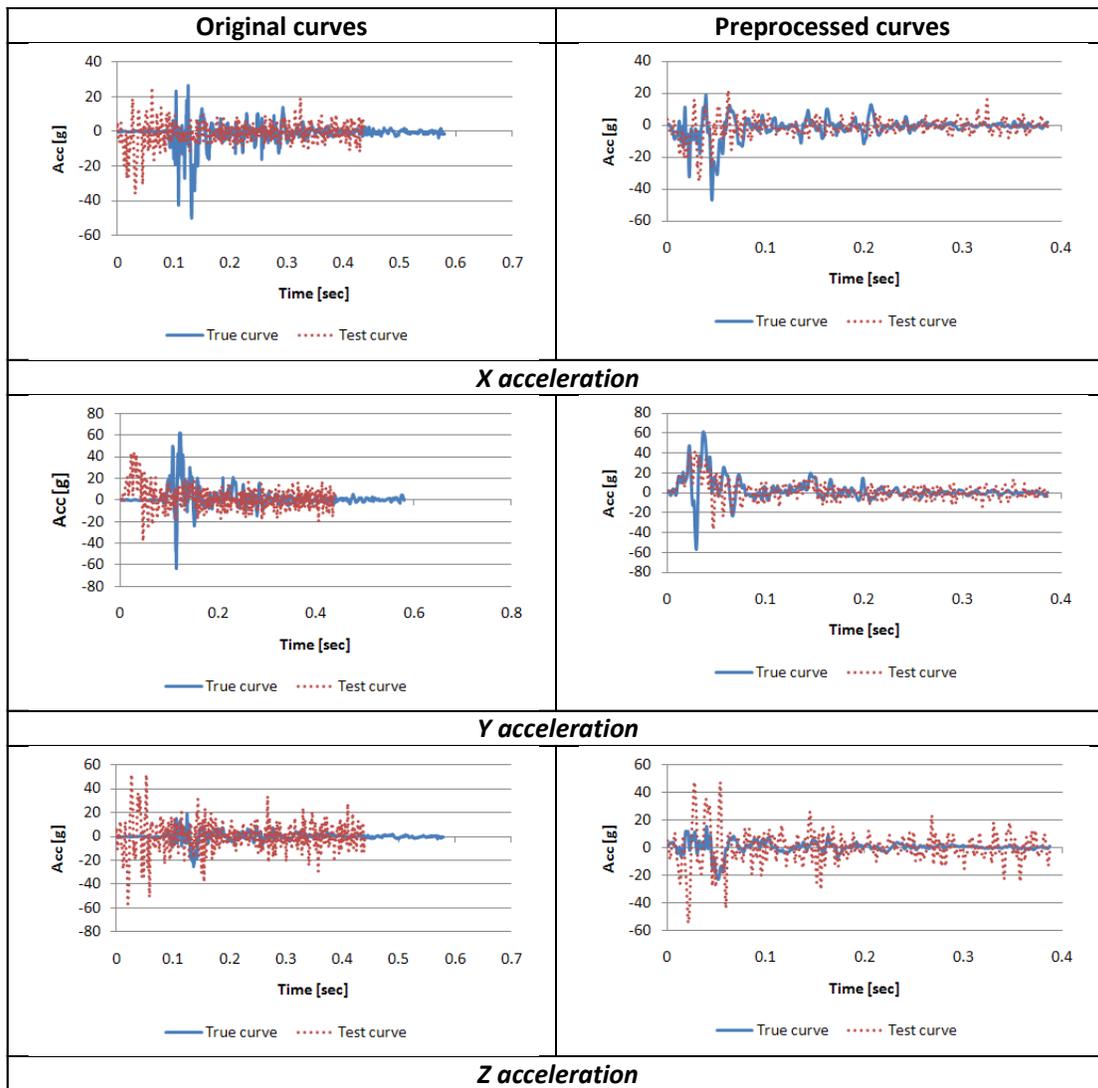
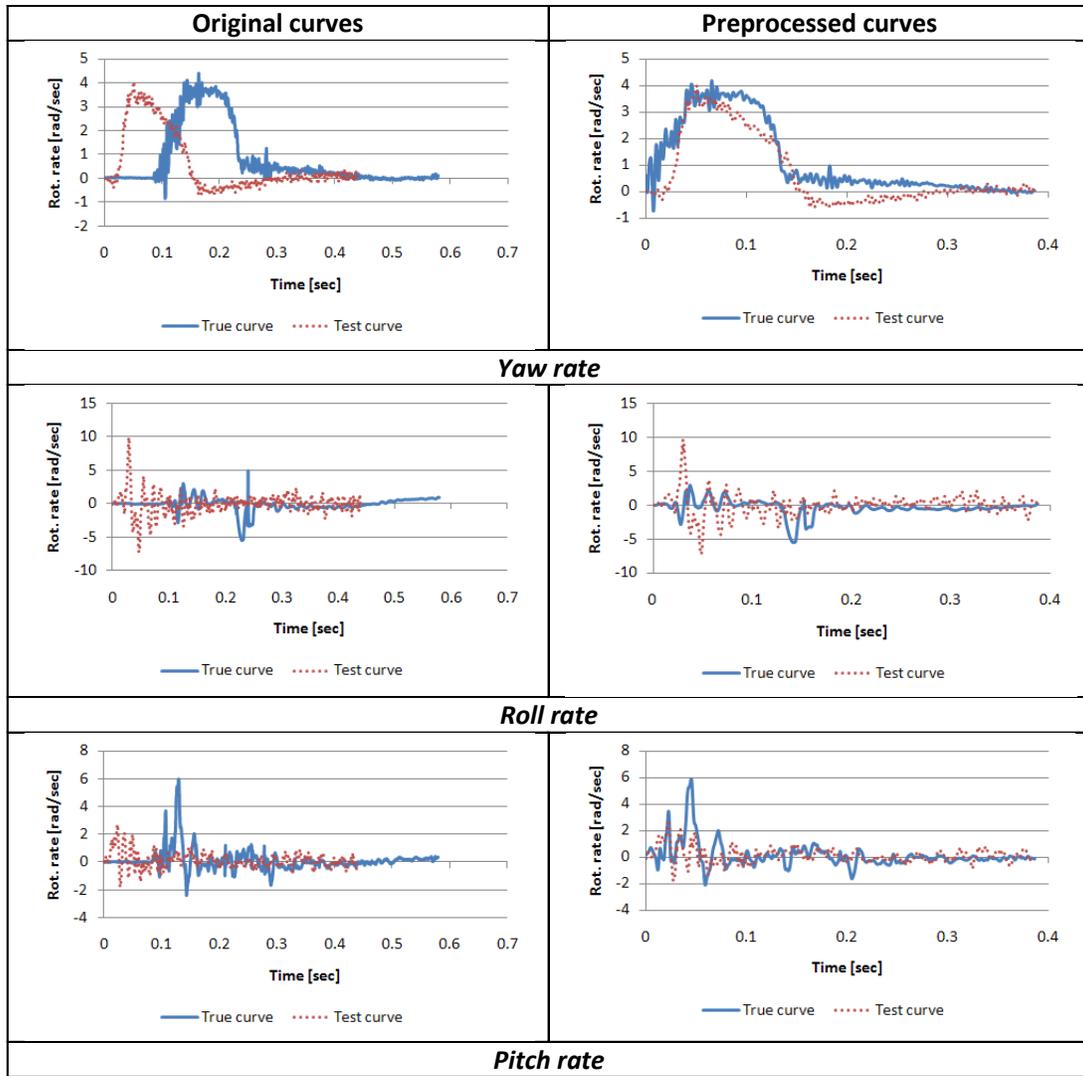


Figure 7.12: Original and preprocessed acceleration curves considering (case with Test 2).



**Figure 7.13: Original and preprocessed rotational rate curves considering (case with Test 2).**

Eventually, the time histories were compared over the complete interval of the preprocessed data, (i.e., 0.4 seconds). The comparison metrics were computed considering each single input channel and also all of them simultaneously by combining the results from each channel using weighting factors or considering the resultants of respectively the accelerations and rotational rates. In particular, in case a weighted average was computed from the score of each single channel, the weighting factors for each channel were assessed automatically by the program.

### 7.2.2.3.1 Metrics computed for each single channel

The results of the evaluation for the individual channels are shown in Table 7.7. As for the comparison versus Test 1, the only channel which was outside the proposed values for both the Sprague & Geers and the ANOVA metrics was the acceleration along the vertical axis Z. In particular, the magnitude component of the Sprague & Geers was greatly over the proposed criterion of 40 percent denoting that, for this channel, the numerical curve had a considerable difference in magnitude respect to the corresponding experimental curve. Nevertheless, it is worth to note that the vertical acceleration can be considered negligible respect to the other channels, in particular the lateral acceleration along the Y axis. All the other input channels (X and Y accelerations and Yaw rate) complied with the proposed acceptance criteria.

The comparison of the six individual channels in the case involving Test 2 confirmed the disagreement between the vertical acceleration time histories from the experimental test and the numerical analysis. In this case, also the phase component of the Sprague&Geers metric for the X-acceleration was slightly above the acceptance value (41.3 %). As for the rotational rates, the yaw rate time histories presented a good level of agreement according to the Sprague & Geers, while the average residual of the ANOVA metrics exceeded the suggested limit of five percent in the case with Test 2. The roll rate slightly exceeded the acceptance values as well.

In any of the channels where the values of either the M or P component of the Sprague&Geers metric were larger than the proposed acceptance criteria (i.e., 40 %), also the corresponding comprehensive part of the metric exceeded the respective maximum value (i.e., 50 %).

**Table 7.7. Roadside safety validation metrics rating table for each of the single channels**

<b>Evaluation Criteria</b> (Time interval [0 sec; 0.4 sec])						
	<b>Sim. vs Test 1</b>			<b>Sim. vs Test 2</b>		
<i>Sprague-Geers</i>	<b>M</b> [%]	<b>P</b> [%]	<b>C</b> [%]	<b>M</b> [%]	<b>P</b> [%]	<b>C</b> [%]
X acceleration	7.7	36.8	37.6	6.8	41.3	41.8
Y acceleration	24.5	38.5	45.6	12.3	39.7	41.6
Z acceleration	218	41.5	222.3	181	47.8	187.5
Yaw rate	0.7	11.1	11.1	16.4	12	20.3
Roll rate	N/A	N/A	N/A	46.2	50.1	68.2
Pitch rate	N/A	N/A	N/A	38.7	40.2	55.8
<i>ANOVA Metrics</i>	<b>Mean Residual</b> [%]	<b>STD of Residual</b> [%]		<b>Mean Residual</b> [%]	<b>STD of Residual</b> [%]	
X acceleration/Peak	0.82	17.4		0.9	16.7	
Y acceleration/Peak	-2.32	30.5		-1	20	
Z acceleration/Peak	-2.84	54.2		-3	53	
Yaw rate	3.3	9.5		-11	11.8	
Roll rate	N/A	N/A		6.2	36.7	
Pitch rate	N/A	N/A		-0.11	16.1	

A visual confirmation of the results obtained from the values of the comparison metrics from each input channel can be found in the analysis of the integrated time histories which are shown in Figure 7.14 and Figure 7.15. Considering the acceleration channels, the integral functions of the Z-acceleration time histories were clearly the ones with the worst match for both the cases with Test 1 and Test 2. Furthermore, the comparison of the integral functions from the accelerations along the X axis denotes that the case with Test 1 has a slightly better agreement than with Test 2, especially as for the magnitude. This is in agreement with the fact that the value of the M component of the Sprague & Geers metric which was computed using the acceleration time histories, in the case with Test 2, presented a slightly worse magnitude correlation between the curves. As for the rotational rate channels, also in this instance, the visual inspection of the corresponding integral functions showed a better agreement in the comparison involving Test 1

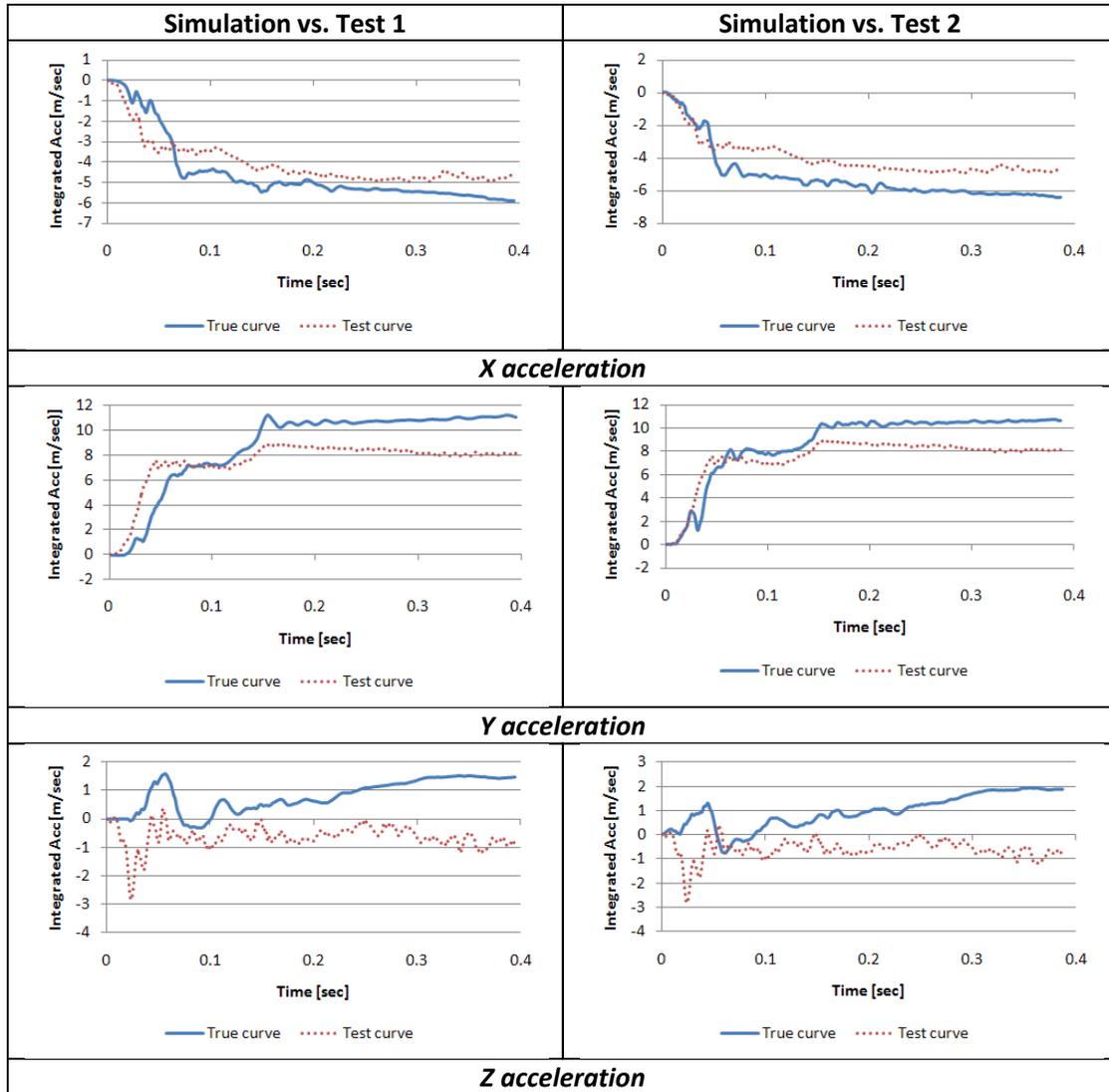


Figure 7.14: Comparison of the integral functions for each of the acceleration channels involving respectively Test 1 (left) and Test 2 (right).

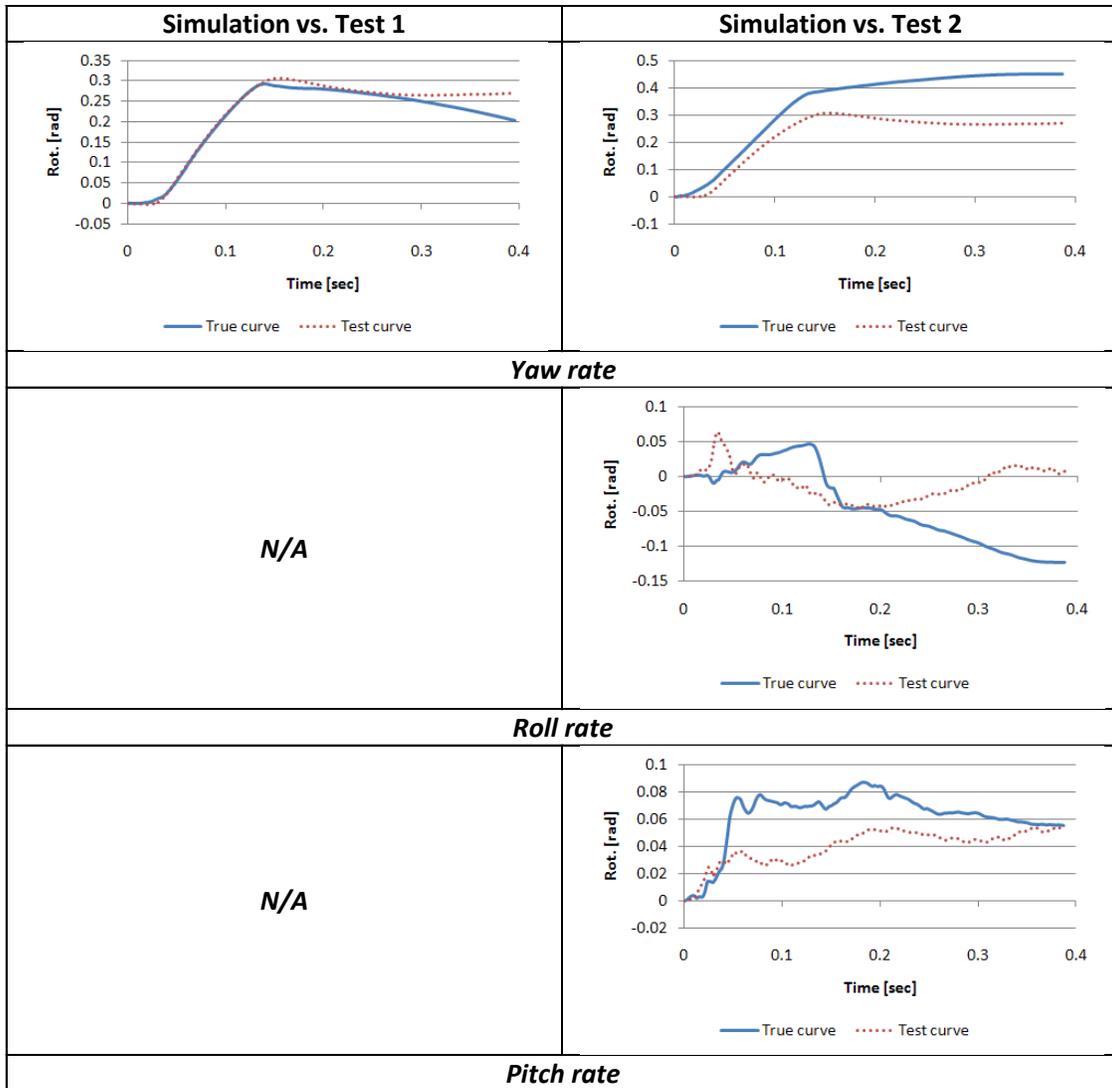


Figure 7.15: Comparison of the integral functions for each of the rotational rate channels involving respectively Test 1 (left) and Test 2 (right).

### 7.2.2.3.2 Metrics computed on multiple channels

Since not all the channels satisfied the acceptance criteria, both the case with Test 1 and Test 2 were further compared using the multi-channel option implemented in the program in order to calculate the weighted Sprague & Geers and ANOVA metrics.

As described in Chapter 4, a total of three different methods for the computation of the weighting factors have been proposed and implemented in this research. One method is based on the peaks of the baseline curves, two of them are based on their respective area and the last proposed method is based on the linear and rotational momentum of the vehicle.

Each of these four methods to compute the weighting factors have been investigated. The weighting factors for each method are summarized in the bar plots in Figure 7.14, which shows the weights computed considering the comparison involving respectively Test 1 and Test 2. The four methods were respectively indicated in the plots as: (i) Peak, (ii) Area I, (iii) Area II and (iv) Inertial properties.

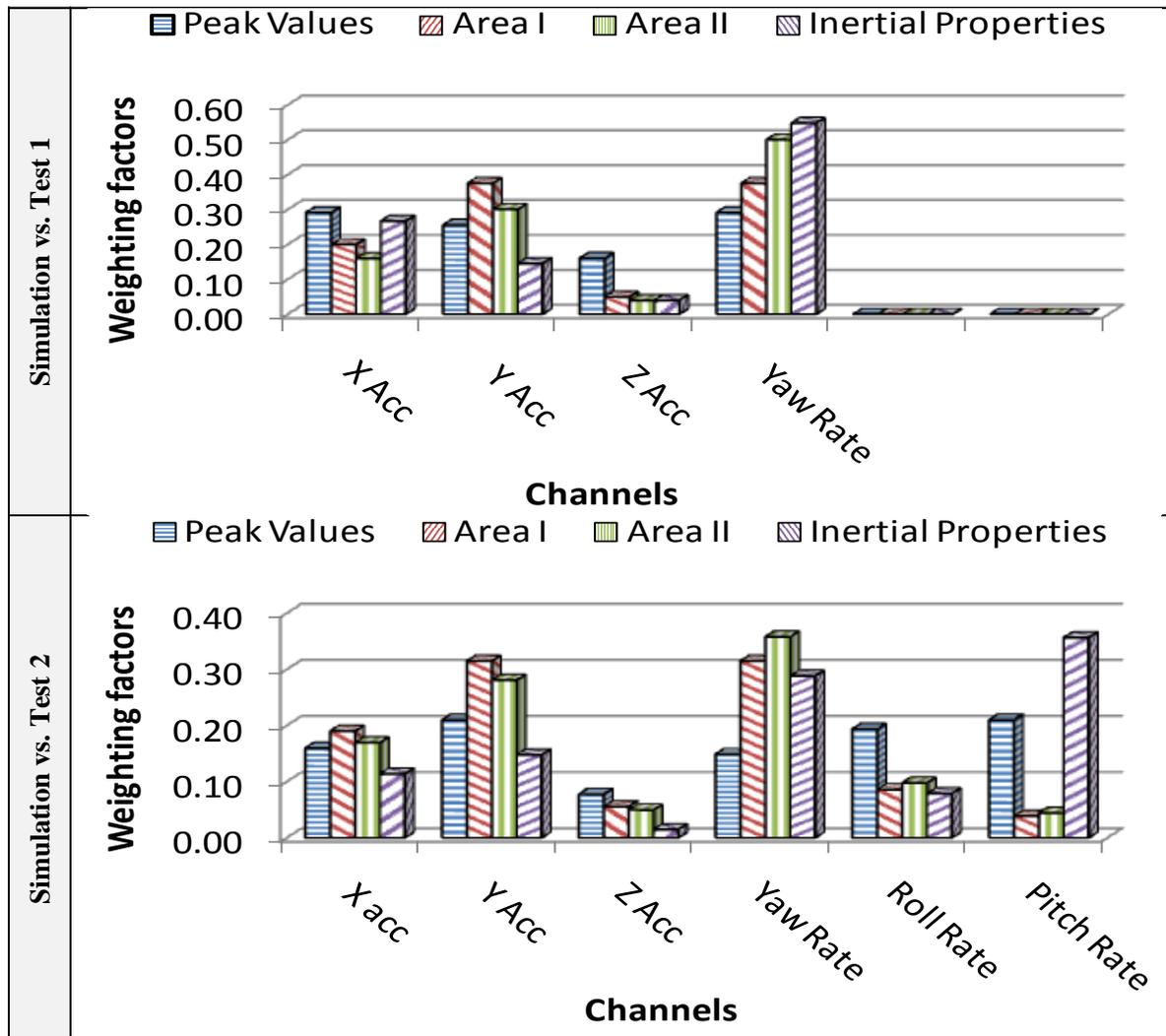


Figure 7.16: Weighting factors computed considering each of the four implemented methods for the comparison involving respectively Test 1 (top) and Test 2 (bottom).

Following is an analysis of the weighting factors computed using each of the four methods proposed in this research and the corresponding weighted average values of the metrics. For the sake of simplicity, the results obtained for each of the four proposed methods are described separately.

#### **7.2.2.3.2.1 Peak values**

Using the method based on the absolute peaks of the baseline curves, in the comparison with Test 1 the X-acceleration channel was given a slightly larger value than the Y-acceleration channel. This was motivated by the fact that the baseline curve for the longitudinal acceleration channel clearly had a negative spike whose absolute value was larger than the peak of the corresponding baseline curve for the lateral acceleration channel. For a similar reason, also the Z-acceleration channels was given an excessively large weighting factor considering that in the experimental test the vehicle basically remained horizontal at all the time during the impact. As for the Yaw-rate channel, the corresponding weighting factor was reasonable considering the actual dynamics of the impact during the real crash test.

In the second case (comparison with Test 2), the weights obtained using the method based on the peaks of the curves gave correctly most of the importance to the Y-acceleration channel in the acceleration group but, in the group of the three rotational rates, too much weight was given to the pitch and roll-rate channels respect to the yaw-rate channel.

In general, the use of the absolute peaks of the curves to evaluate the weighting factors seems to be too related with the spikes that may occur. The weighted average values of both the Sprague&Geers and the ANOVA metrics computed using these weighting factors are shown in Table 7.8.

Table 7.8. Metrics rating table for the Geo-Metro model using the multi-channel option (Peak method)

Simulation vs. Test 1							
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.29						
	Y Channel – 0.26						
Z Channel – 0.16							
Yaw rate Channel – <b>0.29</b>							
Simulation vs. Test 2							
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.16						
	Y Channel – 0.21						
	Z Channel – 0.08						
	Yaw rate Channel – 0.15						
	Roll rate Channel – 0.19						
	Pitch rate Channel – <b>0.21</b>						
Simulation vs. Test 1			Simulation vs. Test 2				
<b>Sprague-Geer Metrics</b>	M [%]	P [%]	C [%]	M [%]	P [%]	C [%]	
	43.8	30.5	61.6	37.3	38.6	57.9	
<b>ANOVA Metrics</b>	Residuals Mean [%]		Residuals STD [%]		Residuals Mean [%]		Residuals STD [%]
	0.1		24.4		0.8		23.2

#### 7.2.2.3.2.2 Area I and Area II

The use of the area of the baseline curves for each channel provided a reasonable distribution of the weights in both the cases with Test 1 and Test 2. In the former case, where only four channels were compared, the largest weight among the acceleration group was given to the Y-acceleration channel and the smallest to the Z-acceleration channel. This is reasonable considering that the impact is a 20-degree lateral impact. Also, a reasonable weight was given to the yaw-rate channel. In particular, the two implemented versions of this method based on the area of the curves, gave very similar results. The only notable difference was that the second version of this method (i.e., Area II) implied an equal distribution of the weights between the acceleration and rotational rates groups which, in the case with only the only yaw rate compared (i.e., Test 1), implicated a weight of 0.5 to be assigned to the yaw rate channel.

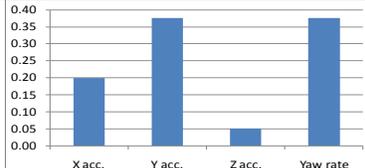
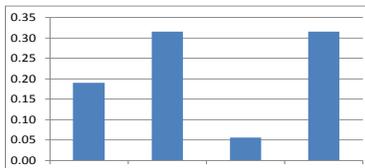
The weights computed in the case with all the six channels available were consistent with the previous comparison with only four channels. In fact, also in this case the distribution of the weight in the acceleration group reflects the actual importance of the channels as expected from the results obtained in the experimental test (i.e., the Y-acceleration channel was given the highest weight and the Z-acceleration channel the lowest). In particular, in this case, the more

realistic distribution of the weights is confirmed also in the values assigned to all the three channels of the rotational-rate group, in which the yaw-rate channel was given the largest weight and the roll-rate and pitch-rate channels are considered negligible. Note that, in the case with the comparison versus Test 2, in general, the values assigned to the weights were smaller as they were redistributed over all the six channels and not only four as in the case involving Test 1.

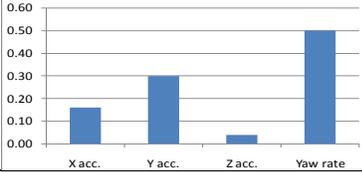
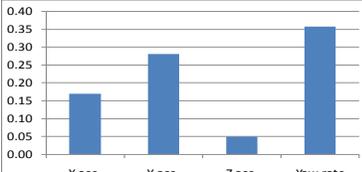
A further confirmation of the low weighting factor assigned to the Z-acceleration channel can be obtained analyzing the integrals of the acceleration time histories for the X, Y, and Z channels in the case involving the comparison versus either Test 1 or Test 2 (Figure 7.14). In fact, it can be clearly seen that the change in velocity along the Z direction (i.e., the integral of the Z-acceleration curve) was insignificant respect to the other two acceleration channels. Similar conclusions can be drawn also for the roll- and pitch-rate channels for the case with Test 2 (Figure 7.15).

The weighted average values of both the Sprague&Geers and the ANOVA metrics computed using the weighting factors based on the method Area I and Area II are shown in respectively in Table 7.9 and Table 7.10.

**Table 7.9. Metrics rating table for the Geo-Metro model using the multi-channel option (Area I method)**

Simulation vs. Test 1						
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.20					
	Y Channel – 0.38					
Simulation vs. Test 2						
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.19					
	Y Channel – 0.31					
Simulation vs. Test 1						
<i>Sprague-Geer Metrics</i>	M [%]		P [%]	C [%]	Simulation vs. Test 2	
	21.8		28	39.8	M [%] P [%] C [%]	
<i>ANOVA Metrics</i>	Residuals Mean [%]		Residuals STD [%]	Residuals Mean [%]		Residuals STD [%]
	0.4		21.2	3.3		19.8

**Table 7.10. Metrics rating table for the Geo-Metro model using the multi-channel option (Area II method)**

Simulation vs. Test 1							
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.16						
	Y Channel – 0.30						
Z Channel – 0.04							
Yaw rate Channel – 0.50							
Simulation vs. Test 2							
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.17						
	Y Channel – 0.28						
Z Channel – 0.05							
Yaw rate Channel – 0.36							
Roll rate Channel – 0.10							
Pitch rate Channel – 0.04							
Simulation vs. Test 1			Simulation vs. Test 2				
<i>Sprague-Geer Metrics</i>	M [%]	P [%]	C [%]	M [%]	P [%]	C [%]	
	17.6	24.7	34.1	25.7	31.5	44.5	
<i>ANOVA Metrics</i>	Residuals Mean [%]		Residuals STD [%]		Residuals Mean [%]		Residuals STD [%]
	1%		18.8%		-3.7		19.6

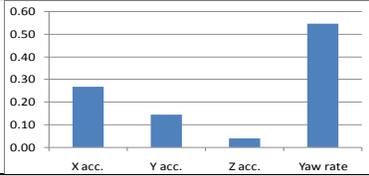
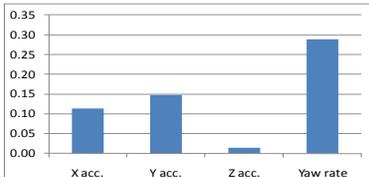
### 7.2.2.3.2.3 *Inertial properties*

In the comparison versus Test 1, the use of the vehicle momentum to compute the weighting factors gave an excessively large weight to the X-acceleration channel compared to the lateral acceleration channel which does not reflect the actual physics of the experimental test. In fact, as the crash test was a lateral impact, the Y-acceleration was supposed to be the most relevant. Considering the overall distribution of the weights, although the yaw-rate was another relevant channel, it was probably given an excessively large weighting factor (i.e., almost 0.6).

This method presents some evident setbacks also in the case all the six channels were compared. In fact, although in this case the distribution of the weighting factors in the acceleration group was realistic, in the rotational rate group an excessive weight was given to the pitch-rate channel (which actually had the biggest weight). An explanation for this may be the fact that the momentum was evaluated till the peak of each specific channel and the pitch-rate presented an evident anomalous spike. Also, the assumption of an arbitrary arm equal to the unity for the conversion of the rotational momentum may probably not be appropriate in these cases.

The weighted average values of the comparison metrics computed using the weighting factors based on the inertial properties of the vehicle are shown in Table 7.11.

**Table 7.11. Metrics rating table for the Geo-Metro model using the multi-channel option (Inertial prop. method)**

Simulation vs. Test 1								
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.27							
	Y Channel – 0.15							
Z Channel – 0.04								
Yaw rate Channel – 0.55								
Simulation vs. Test 2								
<b>Multi-Channel Weights -Peak Method-</b>	X Channel – 0.11							
	Y Channel – 0.15							
Z Channel – 0.01								
Yaw rate Channel – 0.29								
Roll rate Channel – 0.08								
Pitch rate Channel – 0.36								
Simulation vs. Test 1			Simulation vs. Test 2					
<b>Sprague-Geer Metrics</b>	M [%]	P [%]	C [%]	M [%]	P [%]	C [%]		
	14.7	23.2	31.6	27.4	33	44.8		
<b>ANOVA Metrics</b>	Residuals Mean [%]		Residuals STD [%]		Residuals Mean [%]		Residuals STD [%]	
	1.6		16.4		-2.9		17.6	

**7.2.2.3.2.4 Resultant time histories**

As previously described in Chapter 4, an alternative method for comparing simultaneously multiple channels which has been implemented into the computer program is based on the use of the resultant of both the acceleration and rotational velocity time histories. In this case, the program computes the respective resultant histories using the available input channels and proceeds to the comparison. Eventually, the metrics from both the acceleration and rotational resultants were averaged.

The values of the comparison metrics computed using the resultant method are shown in Table 7.12.

**Table 7.12. Metrics rating table for the Geo-Metro model using the multi-channel option (Resultant method)**

Simulation vs. Test 1			Simulation vs. Test 2					
<b>Sprague-Geer Metrics</b>	M [%]	P [%]	C [%]	M [%]	P [%]	C [%]		
	29.4	14.8	35.3	10.7	18.2	23		
<b>ANOVA Metrics</b>	Residuals Mean [%]		Residuals STD [%]		Residuals Mean [%]		Residuals STD [%]	
	6.2		13.9		3.3		16.5	

### 7.2.3 Analysis of results

The analysis of the weighting factors obtained considering all the four proposed methods for both the two comparison investigated cases shows that the two variations based on the area of the baseline curves were the ones which provided the most realistic distribution for the weights. In particular, both the two proposed area methods were capable to give a reasonable distribution of the weights in both the accelerations and rotational rates groups. Although they both provided similar weights, eventually, it is proposed to use the second formulation of the area method. In fact, this formulation implies an equal distribution of the weighting factors between the accelerations and rotational rate groups which seems to be more appropriate.

The column plots in Figure 7.17 and Figure 7.18 summarize the metric values obtained using each of the five different methods to compare the multiple channels simultaneously for respectively the case involving Test 1 and Test 2. With the exception of the first method based on the peaks of the curves, for all the other four proposed methods to compute the metrics using multiple channels altogether, the magnitude (M) and phase (P) components of the Sprague&Geers metric were in accordance with the comprehensive (C) assessment.

The weighted average values of both the Sprague&Geers and the ANOVA metrics computed using the weighting factors based on the area of the curves are shown in Table 7.10. In this case, all the values complied within the proposed acceptance criteria.

Considering the results obtained from the calculation of the metrics with multiple channels for both the two comparison cases analyzed (involving respectively Test 1 and Test 2), the quantitative evaluation of the time histories can be considered as an efficient and reliable method to validate numerical models.

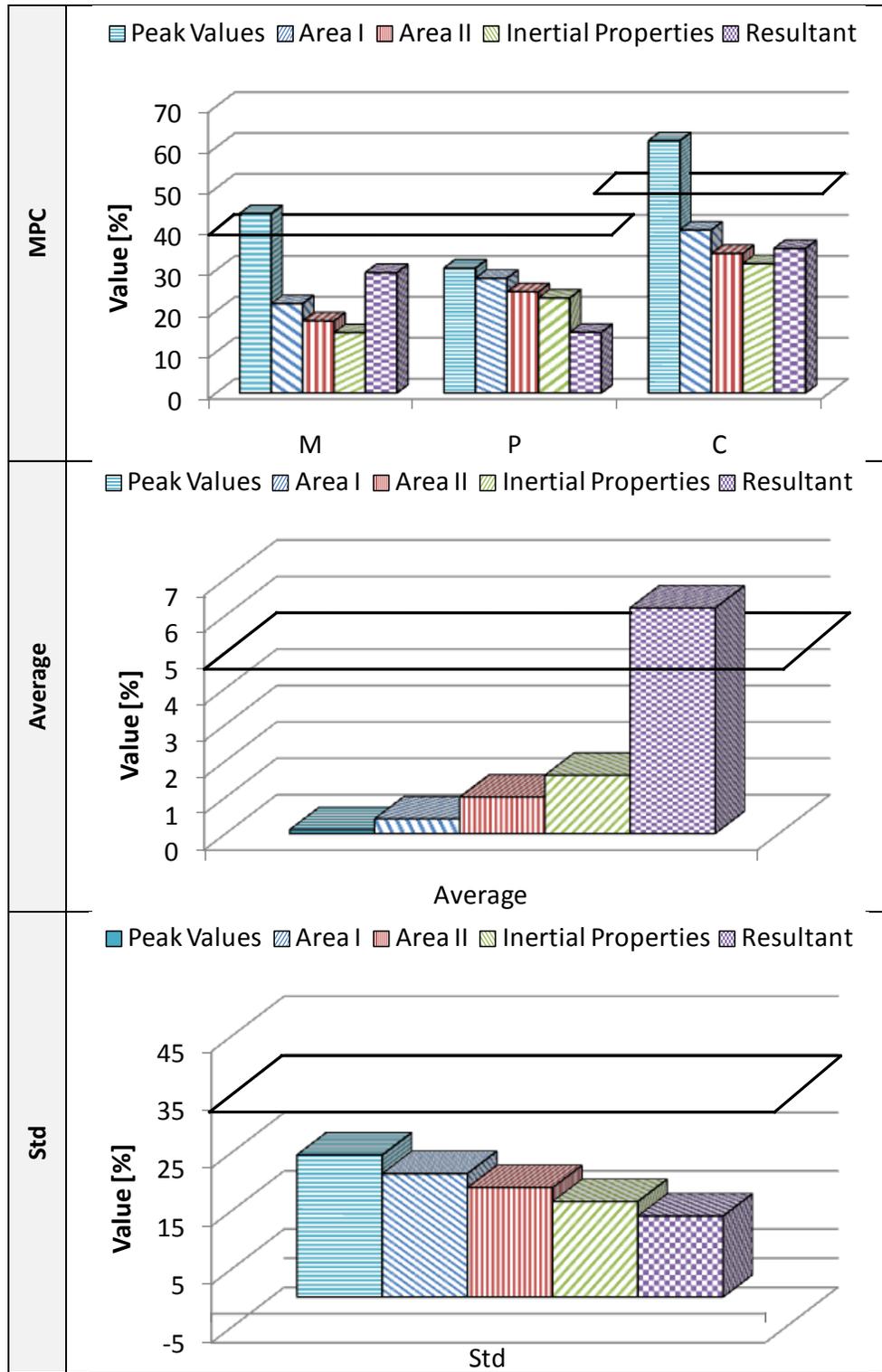


Figure 7.17: Comparison of the weighted values using each of the five proposed methods for respectively the Sprague&Geers (top), Average (middle) and Std (bottom) metrics (case with Test 1).

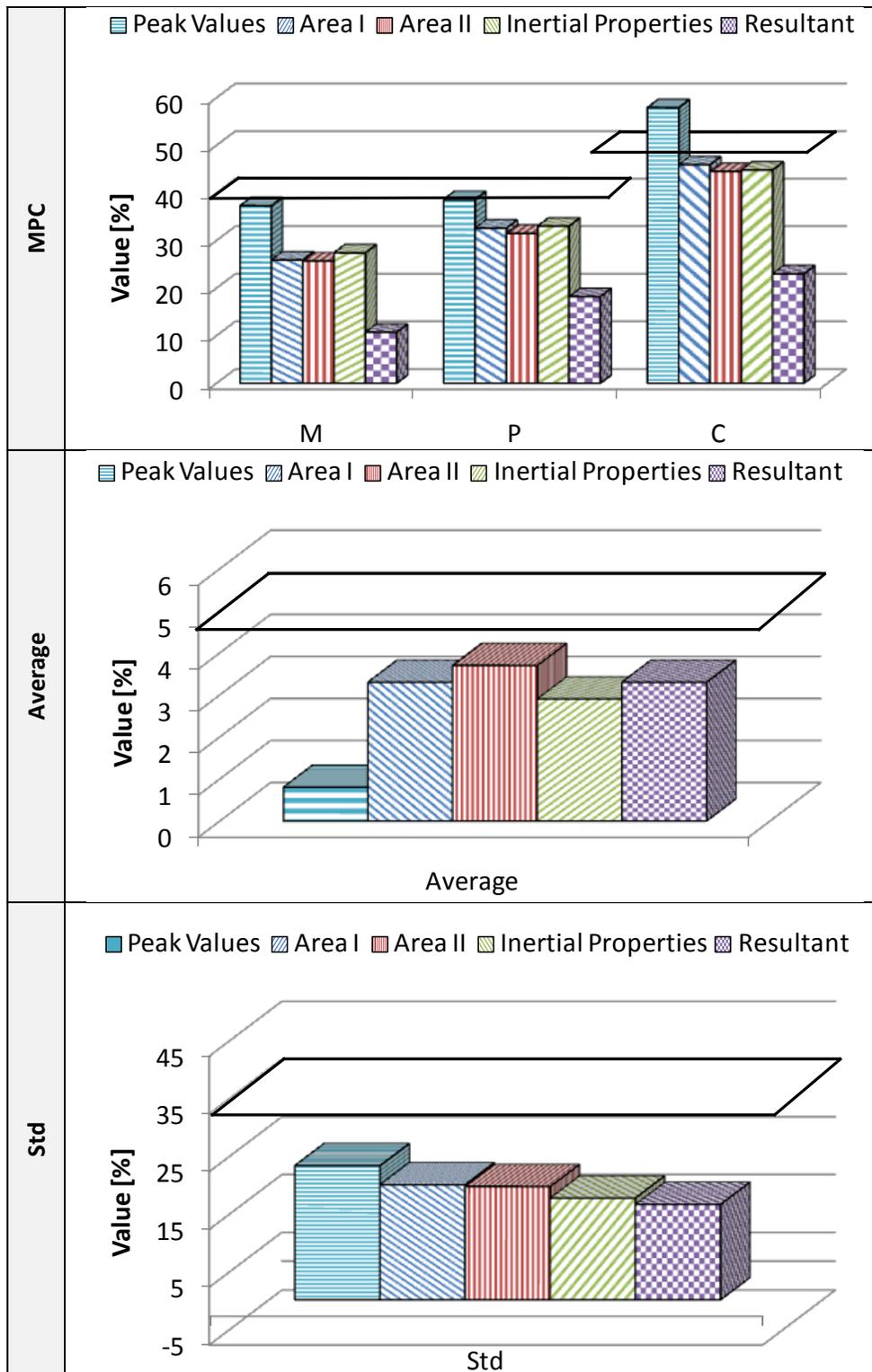


Figure 7.18: Comparison of the weighted values using each of the five proposed methods for respectively the Sprague&Geers (top), Average (middle) and Std (bottom) metrics (case with Test 2).

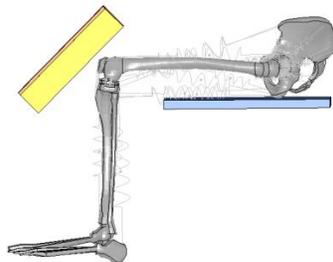
### 7.3 Validation of a Knee-Thigh-Hip FE model

This section presents the results obtained in the validation process of a finite element model reproducing the Knee-Thigh-Hip (KTH) complex. The numerical model was developed by Silvestri and Ray [69] to investigate the consequences of car crashes to the human lower extremities under various conditions.

This case aims to assess a quantitative validation of the FE model of the KTH complex. The three main model components (i.e., condyle, femur and pelvis) were validated separately using comparison metrics. As in the previous case, the Sprague&Geers and the ANOVA metrics were used for the validation of each component of the model. The quantitative comparison of the experimental and numerical curves would eventually guarantee an objective and more rigorous validation of the finite element model.

#### 7.3.1 KTH FE model

Figure 7.19 shows the FE model of the KTH which was developed by Silvestri and Ray to study the passenger injuries to the lower extremities during a car crash. The model carefully reproduces the geometry of a 50<sup>th</sup> percentile male and the biomechanical properties of the bones and soft tissues.



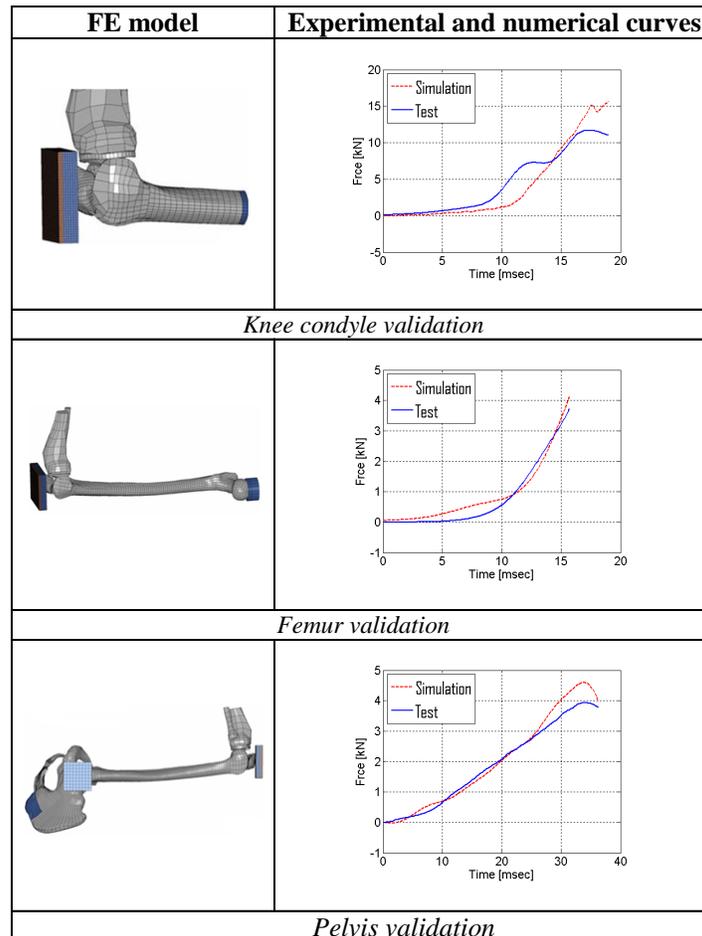
**Figure 7.19: Finite element model of the KTH complex [69].**

To validate each main component of the numerical model, experimental setup tests were considered with respect to three regions of the KTH: (i) knee condyle, (ii) femur and (iii) pelvis [70]. For the validation of each of these three components, a numerical simulation was performed which replicated the same conditions of the corresponding experimental tests [69].

### 7.3.2 Quantitative validation

Each of the three components of the FE model of the KTH complex had already been validated by Silvestri and Ray using a qualitative approach by visually comparing the curves obtained from the numerical simulations and the test results. In this case, an objective validation was performed by comparing the experimental and numerical curves using the method implemented in this research.

Figure 7.20 shows the set-up of the finite element model used to replicate each of the experimental tests along with the plots of the corresponding numerical and experimental curves. The curves for each validation case were filtered and synchronized by minimizing the absolute area of the residuals before the computation of the metrics.



**Figure 7.20:** Plots of the test and numerical curves and the FE model set-up for the knee condyle (top), femur (middle) and pelvis (bottom) validation cases.

Table 7.13 shows the values of the metrics from the comparison of the experimental and numerical curves for each of the three validation cases.

**Table 7.13 S&G, MPC and ANOVA metric values**

<b>MPC Metrics [%]</b>	<b>Condyle</b>	<b>Femur</b>	<b>Pelvis</b>
Sprague-Geers M	-4.7	2.2	9.2
Sprague-Geers P	9.9	5.6	2.5
Sprague-Geers C	10.9	6	9.6
<b>ANOVA metrics [%]</b>			
Average	-7	3	3
Std	13	5	7

For all the validation cases, the comprehensive Sprague&Geers metric was always below 15 percent. In particular, the magnitude and phase components were always smaller than 10 percent.

As for the ANOVA metrics, both the average and the standard deviation of the residuals were always below 15 percent of the peak value of the respective experimental curve. In particular, the femur and pelvis validation showed a residual mean smaller than 5 percent and a standard deviation of the residuals smaller than 10 percent of the peak of the respective experimental curve.

### 7.3.3 Analysis of results

The results of the FE model validation using the comparison metrics were in line with the subjective validation previously performed.

The low values of each of the Sprague&Geers components and also the comprehensive value for the pelvis and femur validation cases indicated a good agreement of the test and experimental curves. In particular, the value of the magnitude and phase components smaller than 10 percent showed a good magnitude and phase correlation for both the experimental and numerical curves. Only in the case of the condyle validation, the Sprague&Geers comprehensive metric (C) was slightly above 10 percent. This could be related to the behavior of the condyle bone during this particular test. A further investigation with respect to other condyle tests outcome and improvements to the FE condyle model could lead to a better match in the future.

# Chapter 8

## Conclusions and Further Research

In this chapter a summary and discussion of the outcomes from this research is given together with some suggestions for a further research.

### 8.1 Summary of the research

The objective of this study was the development of a computer program which could quantitatively assess the degree of match of single or multiple pairs of curves. The possibility to evaluate the match between single or multiple pairs of curves using quantitative measurements could drastically improve the objectivity of the verification and validation process of numerical models compared to the traditional subjective methods.

The created software represents a useful tool for engineers and analysts tasked with the difficult job of judging the appropriateness of simulation techniques in various fields, with a particular focus on roadside safety. In fact, the comparison metrics computed by the program can provide an invaluable support to engineers and managers in assessing the level of reliability of numerical results and in providing guidance on the aspects for which an additional validation may be necessary.

Although some previous attempts to create computer programs for the validation and verification of numerical models can be found in the literature, all these programs were limited to the analysis of some specific aspects of the comparison between the curves only; in particular, most of them required a certain degree of subjectivity to quantify the level of agreement or disagreement. The developed software analyzes the curves considering various aspects of the comparison, such as the differentiation among the magnitude or phase differences and the analysis of the randomness of the residuals. The main feature of the program is that the implemented procedure is completely objective without any introduction of subjective judgments by the user.

Another important characteristic of the computer program which was not implemented in any of the softwares described in the literature is the capability to preprocess the curves. As curves may

often require some type of preprocessing to be properly compared, various preprocessing options have been implemented in the program as well. Some of these preprocessing operations like the resampling and trimming of the various curves are always necessary and automatically performed by the program, while others are directly controlled by the user who can decide whether or not to apply them. Moreover, the interaction with the user is achieved through easy and intuitive graphical interfaces.

All the main aspects implemented in the code such as the input and preprocessing of the curves, the computation of the metrics for both single and multiple channels, and the output of the results have been thoroughly verified.

## **8.2 Comparison metrics and acceptance criteria**

In order to quantify the level of agreement/disagreement between couples of curves, comparison metrics have been used. An extensive review and classification of the comparison metrics available in the literature has been carried out. Eventually, the metrics which consider the curves as deterministically defined have been chosen as the most suitable for the purpose of verification and validation of numerical models in which a large number of parameters may be present. In fact, although the use of metrics which consider the stochastic distribution of the results would indeed provide additional useful information about the match of the curves, their use for highly computationally demanding numerical models with a large number of parameters involved would be practically impossible. The selected deterministic comparison metrics can be divided into two groups: (i) MPC and (ii) single-value metrics.

The next step in the research was the development of acceptance criteria for the selected comparison metrics with the intent of using them for the verification and validation process of computational models in roadside safety. In fact, although the use of comparison metrics allows an objective quantification of the level of agreement, eventually it is necessary to compare the metrics score with acceptance criteria. In particular, suitable criteria have to represent the typical scatter of data which characterizes the experimental tests in the particular field where comparison metrics are applied. In this study, the acceptance criteria have been established by comparing ten essentially identical full-scale vehicle crash tests.

After being properly preprocessed, the lateral accelerations of the repeated tests have been analyzed considering three different relevant baseline curves: (i) the average response (case I), (ii) the curve of the test which gave the best 90<sup>th</sup> percentile envelope (case II) and (iii) the curve of the test whose behavior was closer to the average response (case III). For each of these cases, three groups of curves have been analyzed comprising respectively the results of five tests with the same exact type of vehicle, five test with similar vehicles and, eventually, all the ten tests. The mean value of each metric from these three groups was considered and a confidence interval was eventually created. The analysis of the boundaries of these confidence intervals for all the three main cases lead to the definition of acceptance criteria which should be representative of the scatter of results in roadside safety.

In particular, the results obtained from the three cases denoted a limitation of the regression coefficient due to the fact that, in some cases, imaginary numbers were obtained. Another metric which showed problems is the Knowles&Gear, whose phase component was characterized by anomalously large numbers for each of the four different tests of Set#2 in case III. Eventually this affected the comprehensive value of the metric also. As this problem was limited to that specific set of curves, this was probably due to a numerical approximation of small numbers during the computation of the metrics.

Moreover, the analysis of the velocity time histories instead of the original acceleration time histories has confirmed that using back-calculated data can negatively affect the computation of some comparison metrics. As an example, using the integrals of the accelerations resulted in anomalous values of the ANOVA metrics due to the fact that the integration of the original time histories implied an inevitable introduction of errors. This aspect was also visually confirmed by the analysis of both the residuals probability density and cumulative distribution functions which were not normal any longer.

Eventually, Sprague&Geers and the ANOVA metrics have been indicated as the preferred metrics for the comparison of curves in roadside safety. The reasons of this choice can be summarized in the following:

- similar results between the different variations of the Geers metrics,
- different scaling for the M and P components of Russell and Knowles&Gear metrics,

- potential computation problems with Knowles&Gear metric, and
- effectiveness of ANOVA metrics in the investigation of systematic errors.

### **8.3 Discussion of results**

#### **8.3.1 Effectiveness of comparison metrics**

The capacity of the implemented method to quantitatively mimic an expert evaluation was tested by confronting the results obtained from the computed metrics and the independent subjective opinions of a group of experts regarding the level of agreement or disagreement of two different couples of velocity waveforms. The comparison between the quantitative and subjective assessment of the match was divided into two parts, each of them considering respectively the metrics which gave a measure of the error and those which gave a measurement of the agreement. The two pairs of curves which obtained respectively the worst and best level of agreement according to the experts' evaluation were analyzed.

For both the two pairs, all the error metrics except the Zilliacus and Russell were within the 90<sup>th</sup> percentile interval computed with respect to the experts subjective opinion. As for the three agreement metrics (the regression coefficient and the two variations of the correlation coefficient), in the case of a poor level of agreement they appeared to be more forgiving than the average of the experts opinion.

In particular, the variations of the Geers metric showed to be the metrics of the MPC group which were closest to the average subjective evaluation in case of poor correlation of the two curves. It is worth noting that the capacity to mimic the subjective opinion of the experts for a poor agreement is undoubtedly a characteristic far more important than giving a slightly better score in case of good agreement. Moreover, these types of metrics have demonstrated an excellent capability to distinguish the contribution due to either magnitude or phase differences between the curves in case of a poor agreement.

Although it was not possible to directly compare the ANOVA metrics with the average subjective experts' opinion as no information was collected regarding the experts' opinion about the distribution of residuals, the values obtained for these metrics were consistent with the

average experts' judgment about the best and worst pair of curves. This confirmed the good diagnostic capacity of the ANOVA metrics as well.

Overall, the results obtained from the comparison with the subjective evaluations confirmed the ability of the metrics implemented in the computer program and, in particular the Sprague&Geers and the ANOVA metrics, to mimic an expert evaluation.

### **8.3.2 Validation of numerical models**

The acceptance criteria and the metrics proposed for roadside safety were applied to the validation of a numerical model used to simulate the crash test of a small vehicle impacting against a rigid barrier. After the stability of the numerical simulation were verified thorough considerations on the conservation of mass and energy, the numerical results were compared to two similar experimental tests, one with four channels and the other with all the six channels available.

In both cases, the comparison of each single channel indicated that the most important channel for this type of impact (i.e., the lateral acceleration and the yaw rate) satisfy the acceptance criteria proposed for the Sprague&Geers and ANOVA metrics with the only exception of the residuals average for the yaw rate channel in the comparison versus one test. The Z-acceleration channels, which play a minor role in the impact, exceeded evidently the maximum allowed values; while, for the remaining channels, some metrics were slightly over the limit.

In order to assess an overall comparison of the numerical and experimental outcomes which could consider the effective importance of each of the available channels, four different options were investigated to automatically compute weighting factors without any subjective input by the user. The analysis of the weights obtained using each of the implemented options indicated the two methods based on the area of the baseline curves as those which guaranteed the most realistic distribution of the weighting factors. The other two proposed methods, based respectively on the curve peaks and the momentum of the vehicle showed some limitations due to the excessive sensibility to the presence of spikes in the time histories of some channels. In particular, the latter method could present some practical problems as it requires the input of the vehicle inertial properties which may not be always known.

An alternative solution to the use of weighting factors which was investigated is the computation of the metrics based on the resultant of the accelerations and rotational rates, respectively. The results obtained with the use of the resultants are consistent with those obtained using weighting factors based on the area of the curves. Eventually, either of these two methods is proposed for the comparison of multiple channels in the verification and/or validation process of numerical models in roadside safety.

The implemented method was also used to validate the components of a numerical model of the human lower extremities using data from a limited number of available experimental tests. The encouraging results obtained indicate that the computer program can be a useful tool for the verification/validation of numerical models also outside the roadside safety field.

Eventually, the quantitative method to compare one or multiple pairs of curves which has been implemented into the computer program developed in this research and the suggested acceptance criteria will be incorporated as the core part of the verification and validation procedures for numerical models in roadside safety. In particular, the software described in this dissertation will be used to quantify the comparison of the various time histories collected during crash tests in this field

#### **8.4 Further research**

This work can be extended in a few directions. One improvement could be to propose differentiated acceptance criteria according to each specific channel in order to reflect potentially different levels of scatter for each specific channel (in particular between the groups of accelerations and rotational rates). This may be achieved by considering an analysis of the repeated crash tests as already performed for the lateral acceleration during this research.

For similar reasons, different acceptance criteria may also be proposed for different types of impacts (e.g., lateral, frontal, against rigid or deformable barrier, etc) and, eventually, they could be implemented into the program allowing the user to select a set of criteria for each specific situation.

Another direction in which the research may be further developed is the investigation of new methods for the simultaneous comparison of multiple channels. For example, a possible method

to assess the weighting factors may be based on the various components of the kinetic energy associated to the motion of the vehicle during a crash test.

Moreover, a third direction for further research could be represented by the enhancement of some aspects of the computer program. In fact, as the implemented program can be theoretically applied to the comparison of curves in any field, it may be useful to implement the possibility to compare also curves with a non-monotonically increasing abscissa. This potentiality may allow to compare special types of curves such as loading/unloading cycles.

## References

- [1] Johnson, W., "*Impact Strength of Materials*", Edward Arnold Publishers LTD, London, UK, 1972.
- [2] Kolsky, H., "*Stress Waves in Solids*", Clarendon Press, Oxford, UK, 1953.
- [3] Stone, R., and Ball, J.K., "*Automotive engineering fundamentals*", SAE International, Warrendale, PA, 2004.
- [4] American Society of Mechanical Engineers, "Guide for Verification and Validation in Computational Solid Mechanics", American National Standard V&V 10-2006 (Performance Test Code Comitee, PTC-60), ASME, New York, 2006.
- [5] Belytschko, T., Liu, W.K. and Moran, B., "*Nonlinear Finite Elements for Continua and Structures*", 2000, Wiley, John & Sons.
- [6] Powell, G.H., "*BARRIER VII: A Computer Program for Evaluation of Automobile Barrier Systems*", Prepared for: Federal Highway Administration, April 1973, Report No. FHWA RD-73-51.
- [7] Basu, S., and McHale, G., "*Numerical Analysis of Roadside Design (NARD)*", Report FHWA-RD-88-212, Federal Highway Administration, Washington, D.C., 1988.
- [8] Hallquist, J.O. "*DYNA3D: A Nonlinear, Explicit, Three-Dimensional Finite Element Code for Solid and Structural Mechanics - User Manual*", University of California, Lawrence Livermore National Laboratory, Rept. UCID-1956, 1982.
- [9] Hallquist, J.O. "*LS-DYNA Keyword User's Manual*", Livermore Software Technology Corporation (LSTC), May 2007.
- [10] Wekezer, J.W., Oskard, M.S., Logan, R.W., and Zywicz, E., "Vehicle Impact Simulation", *Journal of Transportation Engineering*, Vol. 199, Issue 4, pp. 598-617, July/August, 1993.
- [11] Ray, M.H., "Using Finite Element Analysis in Designing Roadside Hardware", *Public Roads*, Vol. 57, Issue 4, Federal Highway Administration, Washington, D.C., 1994.

- [12] Hendricks, B.F., O. Sean Martin, and Wekezer, J.W., “*Impact Simulation of the 820C Vehicle with the G2 Guardrail*”, Report FHWA-RD-96-212, FHWA Vehicle Crash Analysis, Federal Highway Administration, Washington, D.C., 1996.
- [13] Martin, O.S., and Wekezer, J.W., “Crash Impact Analysis of the G2 Guardrail: a Validation Study”, *In Proceedings of the 77th Transportation Research Board Meeting*, Transportation Research Board, Paper No. 98-0184, 1998.
- [14] Ray, M.H., Engstrand, K., Plaxico, C.A., and McGinnis, R.G., “Improvements to the Weak-Post W-Beam Guardrail,” *Transportation Research Record 1743*, Transportation Research Board, Washington, D.C., pp 88-96, 2001.
- [15] Eskandarian, A., Marzougui, D., and Bedewi, N.E., "Impact Finite Element Analysis of Slip-Base Sign Support Mechanism", *Journal of Transportation Engineering*, Vol. 126, Issue 2, pp. 143-153, 2000.
- [16] Marzougui, D., Kan, C.D., and Eskandarian, A., "Safety Performance Evaluation of Roadside Hardware Using Finite Element Simulation", *In Proceedings of the ASCE Fourteenth Engineering Mechanics Conference*, Austin, Texas, May 24, 2000.
- [17] Reid, J.D., Sicking, D.L., and Paulsen, G.W., "Design and Analysis of Approach Terminal Sections using Simulation,” *Journal of Transportation Engineering*, Vol. 122, Issue 5, pp. 399-405, 1996.
- [18] Paulsen, G.W., and Reid, J.D., “Nonlinear Finite-Element Analysis of Dual Support Breakaway Sign”, *Transportation Research Record 1528*, Transportation Research Board, Washington D.C., pp 146-154, 1996.
- [19] Ray, M.H., and Patzner, G.S., “Finite Element Model of a Modified Eccentric Loader Breakaway Guardrail Terminal”, *Transportation Research Record 1599*, Transportation Research Board, Washington, D.C., 1997, pp 11-21.
- [20] Epperson, B., Bligh, R., and Ross, H., “USER’S MANUAL – Test Risk Assessment Program (TRAP)”, Texas Transportation Institute, Texas, May 1999.
- [21] Ross, H. E. Jr., Sicking, D.L, Zimmer, R.A., and Michie, J. D., “*Recommended Procedures for the Safety Performance Evaluation of Highway Features*”, National Cooperative Highway

Research Program (NCHRP) Report 350, Transportation Research Board, Washington, D.C., 1993.

[22] Plaxico, C.A., Patzner, C.S., and Ray, M.H., “Finite Element Modeling of Guardrail Timber Posts and the Post-Soil Interaction”, *Transportation Research Record 1647*, Transportation Research Board, Washington D.C., pp 139-146, 1998.

[23] Reid, J.D., Sicking, D.L., Faller, R.K., Pfeifer, B.G., “Development of a New Guardrail System”, *Transportation Research Record 1599*, Transportation Research Board, Washington, D.C., 1997.

[24] Reid, J.D., and Sicking, D.L., “Design and Simulation of a Sequential Kinking Guardrail Terminal”, *International Journal of Impact Engineering*, Vol. 21, Issue 9, pp. 761-772, 1998.

[25] Marzougui, D., Bahouth, G., Eskandarian, A., Meczkowski, L., and Taylor, H., “Evaluation of Portable Concrete Barriers Using Finite Element Simulation”, *Transportation Research Record 1720*, Transportation Research Board, Washington, D.C., pp 1-6, 2000.

[26] Bielenberg, R.W., “Modeling Crushable Foam for the SAFER Racetrack Barrier”, *In Proceedings of the 8th International LS-DYNA User’s Conference*, Dearborn, Michigan, 2004.

[27] Ray, M.H., Oldani, E., and Plaxico, C.A., “Design and Analysis of an Aluminum F-Shape Bridge Railing”, *International Journal of Crashworthiness*, Vol. 9, Issue 4, pp. 349–363, 2004.

[28] Mohan, P., Marzougui, D., Meczkowski, L., and Bedewi, N.E., “Finite Element Modeling and Validation of a 3-Strand Cable Guardrail System”, *International Journal of Crashworthiness*, Vol. 10, Issue 3, pp. 267-273, 2005.

[29] Tahan, F., Marzougui, D., Zaouk, A., Bedewi, N.E., Eskandarian, A., and Meczkowski, L., “Safety Performance Evaluation of Secure Mailboxes Using Finite Element Simulations and Crash Testing”, *International Journal of Crashworthiness*, Vol. 10, Issue 4, pp. 341-349, 2005.

[30] Plaxico, C.A., Kennedy, J.C. Jr., and Miele, C.R., “*Development of an NCHRP Report 350 TL-3 New Jersey Shape 50-inch Portable Concrete Barrier*”, Report No. FHWA/OH- 2006/16, Battelle Memorial Institute, Columbus, Ohio, February 2006.

[31] Anghileri, M., “Numerical Evaluation of Severity Indices for Finite Element Analysis”, *In proceedings of the 85th Transportation Research Board Meeting*, Washington, D.C., 2006.

- [32] American Society of Mechanical Engineers, “Guide for Verification and Validation in Computational Solid Mechanics”, American National Standard V&V 10-2006 (Performance Test Code Committee, PTC-60), ASME, New York, 2006.
- [33] CEN, “*European Standard EN 1317-1 and EN 1317-2: Road Restraint Systems*”, European Standard EN 1317, European Committee of Standardization (CEN), 1998.
- [34] CEN, “*Railway Applications – Crashworthiness Requirements for Railway Vehicles Bodies European Standard prEN 1527*”, (Draft), European Committee of Standardization (CEN), April 2005.
- [35] Federal Aviation Administration, “*Methodology for dynamic seat certification by analysis for use in parts 23, 25, 27 and 29 airplanes and rotorcraft*”, Advisory Circular No. 20-146, Federal Aviation Administration, Washington, D.C., 19 May 2003.
- [36] Ray, M.H., Mongiardini, M., and Atahan, A.O., “Design and Analysis of the Annisquam River Bridge Railing”, *International Journal of Crashworthiness*, Vol. 14, Issue 2, pp. 197-213, 2009.
- [37] Basu, S., and Haghghi, A., “*Numerical Analysis of Roadside Design (NARD) vol. III: Validation Procedure Manual*”, Report No. FHWA-RD-88-213, Federal Highway Administration, Virginia, 1988.
- [38] Ray, M.H., and Hiranmayee, K., "Quantitative Evaluation of the Fidelity of Public-Domain Vehicle Models for Roadside Hardware Research", *unpublished paper*, 1998.
- [39] Tabiei, A., and Wu, J., “Validated crash simulation of the most common guardrail system in the USA”, *International Journal of Crashworthiness*, Vol. 5, Issue 2, 2000, pp. 153 – 168.
- [40] Plaxico, C.A., Ray, M.H., Weir, J.A., Orengo, F., and Tiso, P., “Recommended Guidelines for Curb and Curb–Barrier Installations”, National Cooperative Highway Research Program, Report (NCHRP) Report537, Transportation Research Board, Washington, D.C., 2005.
- [41] Atahan, A.O., and Cansız, Ö.F., “Improvements to G4(RW) Strong-Post Round-Wood W-Beam Guardrail”, *Journal of Transportation Engineering*, American Society of Civil Engineers, ASCE, Vol. 131, Issue 1, 2005, pp 63-73.

- [42] Ray, M.H., "Repeatability of Full-Scale Crash Tests and a Criteria for Validating Finite Element Simulations", *Transportation Research Record*, Vol. 1528, 1996, pp. 155-160.
- [43] Schwer, L.E., "Validation Metrics for Response Time Histories: Perspective and Case Studies", *Engineering. with Computers*, Vol. 23, Issue 4, 2007, pp. 295–309.
- [44] Sean, M.O., and Wekezer, J.W., "Crash Impact Analysis of the G2 Guardrail: a Validation Study", *In Proceedings of the 77th Transportation Research Board Congress*, Washington, D.C., January 1998.
- [45] Oberkampf, L., and Barone, M.F., "Measures of agreement between computation and experiment: validation metrics", *Journal of Computational Physics*, Special issue: Uncertainty quantification in simulation science, Vol. 217, Issue 1, 2006, pp 5–36.
- [46] Russell, D.M., "Error Measures for Comparing Transient Data: Part II: Error Measures Case Study", *Proceedings of the 68th shock and vibration symposium*, 2006, pp. 185–198.
- [47] Cohen, J., Cohen, P., West, S.G., and Aiken, L.S., "*Applied multiple regression/correlation analysis for the behavioral sciences*", Hillsdale, NJ: Lawrence Erlbaum, (3rd ed.), 2003.
- [48] Twisk, D., and Ritmeijer, A., "A Software for Demonstrating Validation of Computer Dummy Models Used in the Evaluation of Aircraft Seating Systems", SAE Paper No. 2007-01-3925, Society of Automotive Engineers, Warrendale, PA, 2007.
- [49] Whang, B., Gilbert, W.E. and Zilliacus, S., "*Two Visually Meaningful Correlation Measures for Comparing Calculated and Measured Response Histories*", Carderock Division, Naval Surface Warfare Center, Bethesda, Maryland, Survivability, Structures and Materials Directorate, Research and Development Report, CARDEROCKDIV-U-SSM-67-93/15, September, 1993.
- [50] Theil, H., "*Economic Forecasts and Policy*", North-Holland Publishing Company, Amsterdam, 1975.
- [51] Geers, T.L. "An Objective Error Measure for the Comparison of Calculated and Measured Transient Response Histories", *The Shock and Vibration Bulletin*, The Shock and Vibration Information Center, Naval Research Laboratory, Washington, D.C., Bulletin 54, Part 2, June 1984, pp. 99-107.

- [52] Russell, D.M., “Error Measures for Comparing Transient Data: Part I: Development of a Comprehensive Error Measure”, *Proceedings of the 68th shock and vibration symposium*, 2006, pp. 175–184
- [53] Shin, Y.S., and Schneider, N. “Ship Shock Trial Simulation of USS Winston S. Churchill (DDG81): Modeling and Simulation Strategy and Surrounding Fluid Volume Effects”, *In Proceedings of the 74th Shock and Vibration Symposium*, San Diego CA, 27-31 October, 2003.
- [54] Jacob, C., Charras, F., Trosseille, X., Hamon, J., Pajon, M., and Lecoz, J.Y., “Mathematical Models Integral Rating”, *International Journal of Crashworthiness*, Vol. 5, Issue 4, 2000 , pp. 417-432.
- [55] Bayarri, M. J., Berger, J. O., Kennedy, M. C., Kottas, A., Paolo, R., and Sacks, J., “Bayesian Validation of a Computer Model for Vehicle Crashworthiness”, *Technical Report 163*, National Institute of Statistical Sciences, July 23, 2005.
- [56] Mahadevan, S., and Rebba, R., “Validation of Reliability Computational Models Using Bayesian Networks”, *Reliability Engineering and System Safety*, Vol. 87, Issue. 5, 2005, pp. 223-232.
- [57] Favarelli, L., “Stochastic Finite Elements for Crash Problems”, *Structural Safety*, Vol. 8, Issue. 1-4, July 1990, pp. 113–130.
- [58] Larsson, A., Pettersson, M., Svensson, H., “CurveAnalyzer v1.0 - User Manual”, Chalmers University, Sweden, 2003.
- [59] Eadie, W.T., Drijard, D., James, F.E., Roos, M., and Sadoulet, B., “Statistical Methods in Experimental Physics”, Amsterdam: North-Holland, 1971, pp. 269–271.
- [60] “*Comparative Shock Analysis (CSA) of Main Propulsion Unit (MPU), Validation and Shock Approval Plan*”, SEAWOLF Program: Contract No. N00024-90-C-2901, 9200/SER: 03/039, September 20, 1994.
- [61] Sprague, M.A., and Geers, T.L., “Spectral elements and field separation for an acoustic fluid subject to cavitation”, *J. Comput. Phys.*, Vol. 162, 2003, pp. 184:149.
- [62] Knowles, C.P., and Gear, C.W., “Revised validation metric”, unpublished manuscript, 16 June 2004 (revised July 2004).

[63] MathWorks, “*MATLAB® – high performance numeric computation and visualization software - User Guide*”, The MathWorks Inc., 3 Apple Hill Drive, Natick, MA, USA, 2009.

[64] SAE J211-1 (R) “*Instrumentation for Impact Test—Part 1—Electronic Instrumentation. SAE International*”, Warrendale PA, USA, Jul 1, 2007.

[65] LS-PrePost – Online Documentation, LSTC

Available online at: <http://www.lstc.com/lsp/index.shtml> (accessed on April 2010).

[66] Road Barrier Upgrade of Standards (Robust) – Deliverable 4.1.1 – Full-Scale Test Results – An Analysis, Report document, ROBUST PROJECT, GRD-2002-70021, 2006.

Available online at: <http://www.ha-research.gov.uk/projects/projectdocuments.php?method=download&ID=100> (accessed on 1 April 2008).

[67] Finite Element Model Archive of National Crash Analysis Center (NCAC)

Available online at: <http://www.ncac.gwu.edu/vml/models.html> (accessed on 8 January 2010).

[68] Road Barrier Upgrade of Standards (Robust) – Geo-Metro Finite Element model (GM\_R3): Improvements of Steering System and Suspensions, Report document, ROBUST PROJECT, Document 05-006, 2005.

Available online at:

[http://www4.vegvesen.no/ROBUSTDOC/Computational\\_mechanics/Report\\_simulation/POMI\\_GM\\_R3\\_Improvements\\_5-006.pdf](http://www4.vegvesen.no/ROBUSTDOC/Computational_mechanics/Report_simulation/POMI_GM_R3_Improvements_5-006.pdf) (accessed on 1 April 2008).

[69] Silvestri, C., Ray, M.H., "Development of a Finite Element Model of the Knee-Thigh-Hip of a 50th Percentile Male including Ligaments and Muscles", *International Journal of Crashworthiness*, Vol. 14, Issue 2, pp: 215-229, 2009.

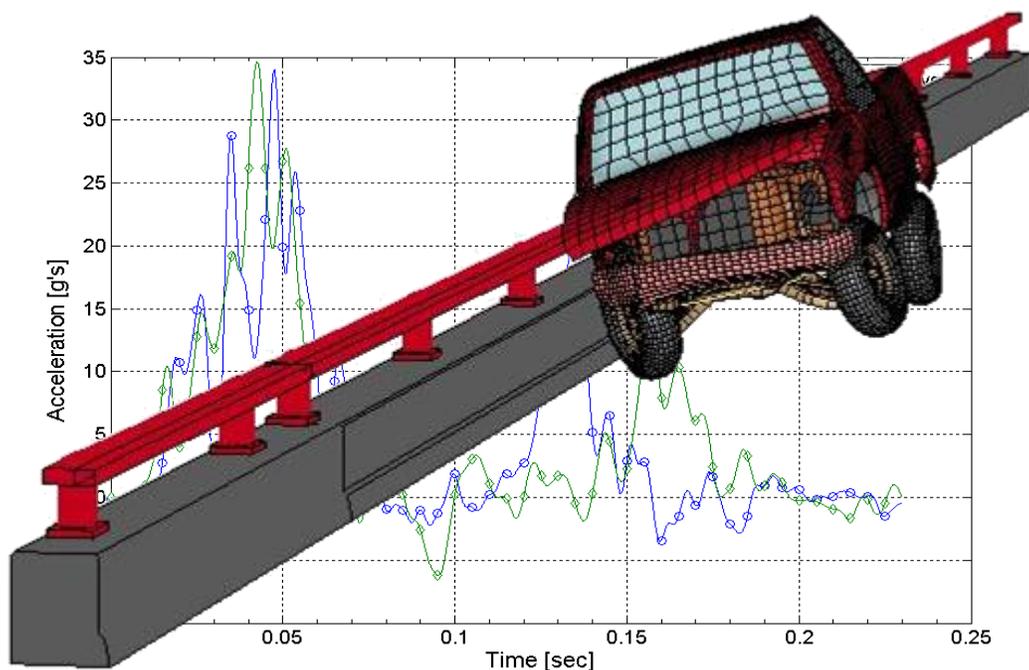
[70] Rupp, J.A., “*Test Reports NB337L-NB448L, 447RR-NB450L, NB450L-NB341L/R*”, Michigan Transportation Research Institute, KTH Injury Investigations, Ann Arbor, MI, 2003.

## **Appendix A**

This appendix contains the user's manual for the computer program developed in this study.

# *Roadside Safety Verification and Validation Program (RSVVP)*

## **User's Manual**



Worcester Polytechnic Institute (WPI)

December 2009 (Rev. 1.4)

Mario Mongiardini

Malcolm H. Ray

## Table of Contents

<b>TABLE OF CONTENTS.....</b>	<b>- 236 -</b>
<b>LIST OF FIGURES.....</b>	<b>- 238 -</b>
<b>LIST OF TABLES .....</b>	<b>- 240 -</b>
<b>A.1. INTRODUCTION TO RSVVP .....</b>	<b>- 241 -</b>
<b>A.1. INSTALLATION.....</b>	<b>- 242 -</b>
A.1.1. SYSTEM REQUIREMENTS.....	- 242 -
A.1.2. INSTALLATION OF THE <i>MATLAB COMPONENT RUNTIME</i> .....	- 242 -
A.1.3. STARTING RSVVP .....	- 243 -
<b>A.2. DATA INPUT AND PREPROCESSING OPTIONS .....</b>	<b>- 244 -</b>
A.2.1. INPUTTING THE CURVES .....	- 244 -
A.2.1.1. <i>Format of input curves</i> .....	- 245 -
A.2.1.2. <i>Copy of the original input curves</i> .....	- 246 -
A.2.2. LOADING A CONFIGURATION FILE .....	- 246 -
A.2.3. INITIAL PREPROCESSING .....	- 251 -
A.2.4. PREPROCESSING .....	- 255 -
A.2.4.1. <i>Filtering</i> .....	- 256 -
A.2.4.2. <i>Shift/Drift controls</i> .....	- 258 -
A.2.4.3. <i>Curve Synchronization (Single-channel mode)</i> .....	- 259 -
A.2.5. MULTICHANNEL OPTIONS.....	- 261 -
<b>A.3. METRICS SELECTION AND TIME INTERVAL .....</b>	<b>- 263 -</b>
A.3.1. METRICS SELECTION .....	- 264 -
A.3.2. TIME INTERVAL.....	- 266 -
A.3.3. COMPRESS IMAGE FILES .....	- 267 -
<b>A.4. METRICS EVALUATION .....</b>	<b>- 268 -</b>
A.4.1. WHOLE TIME WINDOW.....	- 270 -
A.4.2. DEFINITION OF A USER DEFINED TIME WINDOW .....	- 270 -
<b>A.5. SCREEN OUTPUT .....</b>	<b>- 271 -</b>

<b>A.6.</b>	<b>OUTPUT OF RESULTS</b> .....	<b>- 274 -</b>
A.6.1.	TABLE OF RESULTS (EXCEL <sup>®</sup> WORKSHEET).....	- 275 -
A.6.2.	GRAPHS.....	- 277 -
A.6.1.	TIME HISTORIES.....	- 278 -
<b>A.7.</b>	<b>EXAMPLES</b> .....	<b>- 278 -</b>
A.7.1.	SINGLE CHANNEL COMPARISON .....	- 278 -
A.7.1.1.	<i>Preprocessing of the original curves</i> .....	- 279 -
A.7.1.2.	<i>Metric selection and evaluation</i> .....	- 283 -
A.7.1.3.	<i>Results</i> .....	- 287 -
A.7.1.	MULTIPLE CHANNEL COMPARISON .....	- 288 -
A.7.2.	PREPROCESSING .....	- 289 -
A.7.2.1.	<i>Metric selection and evaluation</i> .....	- 292 -
<b>REFERENCES</b> .....		<b>- 298 -</b>

## List of Figures

FIGURE A.1: FORMAT OF THE TEST AND TRUE CURVES. ....	- 246 -
FIGURE A.2: SELECTION OF THE TYPE OF COMPARISON AND RESAMPLING LIMIT. ....	- 248 -
FIGURE A.3: SELECTION OF THE CONFIGURATION FILE. ....	- 248 -
FIGURE A.4: INPUT OF THE TEST AND TRUE CURVES. ....	- 250 -
FIGURE A.5: BUTTONS TO MOVE DIRECTLY TO THE DESIRED CHANNEL. ....	- 251 -
FIGURE A.6: CHECKBOXES FOR THE MANUAL TRIM AND THE TRANSLATION OF THE RAW CURVES. ....	- 252 -
FIGURE A.7: WINDOW FOR THE MANUAL TRIM OF THE INPUT CURVES. ....	- 253 -
FIGURE A.8: SHIFT OF ONE OF THE TWO INPUT CURVES TO THE ORIGIN.....	- 254 -
FIGURE A.9: FILTER OPTIONS' BOX - (A) DROP DOWN MENU AND (B) 'OPTIONAL USER DEFINED CFC' FIELD (IN CASE THE USER DEFINED OPTION HAS BEEN SELECTED). ....	- 257 -
FIGURE A.10: SHIFT AND DRIFT CONTROLS.....	- 259 -
FIGURE A.11: DROP DOWN MENU OF THE 'SYNC OPTIONS' BOX.....	- 260 -
FIGURE A.12: SELECTION OF A NEW STARTING POINT (IN CASE THE USER IS NOT SATISFIED BY THE INITIAL SYNCHRONIZATION OF THE TWO CURVES). ....	- 261 -
FIGURE A.13: SELECTION OF THE METHOD FOR THE COMPUTATION OF THE MULTICHANNEL METRICS.....	- 262 -
FIGURE A.14: SYNCHRONIZATION OF THE CHANNEL/RESULTANT.....	- 263 -
FIGURE A.15: SELECTION OF THE METRIC PROFILES. ....	- 265 -
FIGURE A.16: SELECTION OF THE AVAILABLE COMPARISON METRICS ('USER SELECTED METRICS' PROFILE).....	- 266 -
FIGURE A.17: SELECTION OF THE TYPE OF TIME WINDOW(S).....	- 267 -
FIGURE A.18: OPTION TO COMPRESS/UNCOMPRESS THE IMAGE FILES CREATED BY RSVVP.....	- 268 -
FIGURE A.19: 'EVALUATE METRICS' BUTTON STARTING THE METRIC EVALUATION.....	- 269 -
FIGURE A.20: SELECTION OF THE FOLDER AND NAME FOR THE CONFIGURATION FILE.....	- 270 -
FIGURE A.21: SELECTION OF THE 'USER DEFINED TIME WINDOW' . ....	- 271 -
FIGURE A.22: SCREEN OUTPUT FOR: (A) THE NCHRP 22-24 PROFILE AND (B) THE 'ALL METRICS'/'USER DEFINED' PROFILES. ....	- 272 -
FIGURE A.23: SCREEN OUTPUT FOR: (A) THE NCHRP 22-24 PROFILE AND (B) THE 'ALL METRICS'/'USER DEFINED' PROFILES. ....	- 273 -
FIGURE A.24: SELECTION OF THE FOLDER FOR SAVING THE RESULTS OF THE RSVVP RUN.....	- 275 -
FIGURE A.25: MESSAGE SHOWN DURING THE CREATION OF THE EXCEL FILE. ....	- 275 -
FIGURE A.26: LAYOUT OF THE EXCEL TABLE CONTAINING THE METRICS RESULTS FOR THE VARIOUS TIME INTERVALS. ....	- 276 -
FIGURE A.27: SUMMARY OF PREPROCESSING OPTIONS AND SEPARATE SHEETS FOR EACH INPUT CHANNEL IN THE EXCEL FILE. ....	- 277 -
FIGURE A.28: ORIGINAL ACCELERATION TIME HISTORIES LOADED INTO RSVVP. ....	- 280 -
FIGURE A.29: INPUT CURVES AFTER THE MANUAL TRIMMING OPERATION.....	- 281 -
FIGURE A.30: ORIGINAL AND FILTERED ACCELERATION TIME HISTORIES. ....	- 282 -

FIGURE A.31: FILTERED AND SYNCHRONIZED TIME HISTORIES BASED ON THE (A) LEAST SQUARE AND (B) THE MINIMUM AREA OF RESIDUALS METHOD. .... - 283 -

FIGURE A.32: SELECTION OF THE METRICS PROFILE AND TIME INTERVAL ..... - 284 -

FIGURE A.33: OUTPUT OF RESULTS FOR THE WHOLE TIME INTERVAL..... - 285 -

FIGURE A.34: SETTING OF THE BOUNDARIES FOR THE USER DEFINED TIME INTERVAL. .... - 286 -

FIGURE A.35: OUTPUT OF RESULTS FOR TIME INTERVAL [0.05 SEC , 0.15 SEC].. ..... - 286 -

FIGURE A.36: TIME INTERVAL [0.15 SEC , 0.20 SEC]. ..... - 287 -

FIGURE A.37: OUTPUT OF RESULTS FOR TIME INTERVAL [0.15 SEC , 0.20 SEC]. ..... - 287 -

FIGURE A.38: ORIGINAL AND PREPROCESSED INPUT CHANNELS. .... - 290 -

FIGURE A.39: INPUT CHANNELS BEFORE/AFTER SYNCHRONIZATION. .... - 291 -

FIGURE A.40: SCREEN OUTPUT OF THE RESULTS FOR THE X CHANNEL. .... - 293 -

FIGURE A.41: SCREEN OUTPUT OF THE RESULTS FOR THE Y CHANNEL..... - 293 -

FIGURE A.42: SCREEN OUTPUT OF THE RESULTS FOR THE Z CHANNEL..... - 294 -

FIGURE A.43: SCREEN OUTPUT OF THE RESULTS FOR THE YAW CHANNEL..... - 294 -

FIGURE A.44: SCREEN OUTPUT OF THE RESULTS FOR THE ROLL CHANNEL..... - 295 -

FIGURE A.45: SCREEN OUTPUT OF THE RESULTS FOR THE PITCH CHANNEL. .... - 295 -

FIGURE A.46: SCREEN OUTPUT OF THE RESULTS FOR THE WEIGHTED AVERAGE..... - 296 -

## List of Tables

TABLE A.1: SPECIFICATIONS FOR THE USUAL CFC VALUES.....	- 257 -
TABLE A.2: SUMMARY OF THE METRICS VALUES FOR EACH OF THE TIME INTERVAL CONSIDERED. ....	- 288 -
TABLE A.3: SUMMARY OF THE METRICS VALUES FOR EACH CHANNEL AND THE WEIGHTED AVERAGE. ....	- 297 -
TABLE A.4: DEFINITION OF MPC METRICS.....	- 301 -
TABLE A.5: DEFINITION OF SINGLE-VALUE METRICS. ....	- 302 -
TABLE A.6: ACCEPTANCE CRITERIA SUGGESTED FOR THE NCHRP 22-24 METRICS PROFILE. ....	- 305 -

## A.1. INTRODUCTION TO RSVVP

The Roadside Safety Verification and Validation Program (RSVVP) quantitatively compare one or multiple pairs of curves by computing comparison metrics. Comparison metrics are objective, quantitative mathematical measures of the agreement between two curves. The comparison metrics calculated by RSVVP can be used to validate computer simulation models using data obtained from experimental tests, verify the results of a simulation with another simulation or analytical solution or assess the repeatability of a physical experiment. Although RSVVP has been specifically developed to perform the verification and validation of roadside safety models, it can be used to generally perform a comparison of virtually any pair of curves. All the comparison metrics evaluated by RSVVP are deterministic, meaning they do not specifically address the probabilistic variation of either experiments or calculations (i.e., the calculation results are the same every time given the same input). For a description of each metric calculated by the RSVVP see the Appendix A.

In order to ensure a correct comparison of the curves, RSVVP gives the user the option to perform various preprocessing tasks before the metrics are calculated.

The intuitive and interactive graphical user interface of RSVVP allows the user to effortlessly input the curves to be compared and select any of the existing preprocessing options. Also, a series of automatic warnings alert the user about possible mistakes during the preprocessing phase.

The interpretation of the results obtained using this program is completely the responsibility of the user. The RSVVP program does not presuppose anything about the two curves; it simply compares the data and calculates the metrics. The user must ensure that the curves are appropriate candidates for a comparison and that the data input into the program is appropriate for comparison.

## A.1. INSTALLATION

### A.1.1. System requirements

RSVVP has been written and compiled using Matlab<sup>®</sup>. Running RSVVP requires either the full Matlab<sup>®</sup> (version 2009a or higher) software or the free distributable *MATLAB Component Runtime* (MCR 7.10) software to be installed on the user's system. In either case, the minimum hardware requirements to run RSVVP are:

	32 bit version	64-bit version
<b>CPU</b>	Intel <sup>®</sup> Pentium 4 (and above), Intel Celeron, Intel Xeon, Intel Core, AMD Athlon 64, AMD Opteron, AMD Sempron	Intel <sup>®</sup> Pentium 4 (and above), Intel Celeron, Intel Xeon, Intel Core, AMD64,
<b>RAM</b>	512 MB	1024 MB
<b>Disk space</b>	510 MB (MATLAB <sup>®</sup> only)	510 MB (MATLAB <sup>®</sup> only)

### A.1.2. Installation of the *MATLAB Component Runtime*

The source code of RSVVP has been completely written in Matlab<sup>®</sup> (version R2007b) and then compiled as an executable file for Windows<sup>®</sup> XP/Vista in order to create a standalone program which can run on machines that do not have Matlab installed on them. Although it is a standalone application, if Matlab<sup>®</sup> is not installed RSVVP requires that the *MATLAB Component Runtime* (MCR 7.10) program be installed on the user's computer. MCR provides all the necessary Matlab functional support for the correct execution of the RSVVP software. Before running RSVVP on a machine without Matlab<sup>®</sup> it is first necessary to install the MCR environment. (Note: MCR has to be installed only once).

The latest version of RSVVP and the MCR environment can be downloaded from: [http://civil-ws2.wpi.edu/Documents/Roadsafe/NCHRP22-24/RSVVP/RSVVP\\_1\\_7.zip](http://civil-ws2.wpi.edu/Documents/Roadsafe/NCHRP22-24/RSVVP/RSVVP_1_7.zip)

To install MCR, perform the following steps:

1. Extract the content of the *RSVVP.zip* file in the folder on your PC where you want to install RSVVP (for example: *C:\RSVVP\*).
2. Open the folder where you extracted the files and double-click on the *Installer.bat* file.
3. Follow the instructions of the installation wizard. It may take several minutes to complete the installation. This installs the free Matlab MCR environment that is used in conjunction with RSVVP.
4. Reboot your PC.

At this point RSVVP should be installed on your computer.

### **A.1.3. Starting RSVVP**

After MCR and RSVVP have been installed, simply double-click the *RSVVP.exe* file located in the installation folder previously created (e.g., *C:\RSVVP\*) to start the program.

Once started, a series of graphical user interfaces will guide the user through the preprocessing, the evaluation of the comparison metrics and the saving of the results.

The following sections describe the features and use of the program.

## **A.2. DATA INPUT AND PREPROCESSING OPTIONS**

Each chapter of this manual first gives a brief explanation of the described feature, while the text in the boxed frame indicates how to execute that specific operation.

The first step is the selection of the input curves to be compared and the various preprocessing options. At this stage the user can also decide which comparison metrics will be evaluated by RSVVP.

### **A.2.1. Inputting the curves**

In RSVVP the two curves which have to be compared are called respectively the “true curve” and the “test curve”. The true curve represents the baseline or the reference curve and is assumed to be the correct response, while the test curve represents the model or experiment which has to be verified and/or validated. For example, time histories from a physical crash test would be the “true” curve and an LSDYNA simulation of the same event would be the “test” curve. As the comparison metrics assess the degree of similarity between any pair of curves in general, the input curves may represent various physical entities (e.g., acceleration time histories, force-deflection plots, stress-strain plots, etc.). RSVVP does not presuppose anything about the two curves being compared so the responsibility for ensuring that the units, for example, are compatible rest with the user.

Comparison metrics provide an objective measure of how well two curves match each other and, therefore, can be applied to virtually any pair of curves where a comparison is required. A typical application of the metrics evaluated by RSVVP is the validation of a numerical model by comparing the experimental results with the numerical simulation results. Another application could be to check the repeatability of one experiment by comparing the results obtained from several repetitions of the same experiment. Yet another application is to verify the results of one numerical simulation with the results of another numerical simulation.

Two main types of comparison can be performed in RSVVP:

- Single Channel (i.e., one pair of curves)
- Multiple Channel (i.e., multiple pairs of curves)

In the ‘Single Channel’ option, comparison metrics are based on the comparison of a single pair of input curves while in the ‘Multiple Channel’ option the comparison metrics are evaluated either as a weighted average of the metric values from multiple pairs of curves (i.e., each pair representing a different channel) or by considering the resultant obtained combining the various channels. In either case, the metrics evaluated in the ‘Multiple Channel’ option are intended to give a general assessment of the comparison using all the available channels simultaneously. A typical example of the application of the Multiple Channel option is when the user wants to perform a comparison between experimental and numerical results using all the time history channels that describe the motion of a vehicle (e.g., X, Y Z accelerations and/or roll, pitch and yaw rates) in order to obtain a single comprehensive comparison between a physical test and a numerical simulation. The weighting factors used to combine the values of the comparison metrics from each channel are based on the area under the true curve for that respective channel (see Appendix B for more details).

#### **A.2.1.1. Format of input curves**

The input curve files must be in the ASCII format and can have any extension or no extension at all. The abscissa and ordinate data of the input curves must be tabulated into two columns as shown in Figure A.1. Each line in the input file represents one data point (e.g., time and the acceleration at that time or strain and the corresponding stress). In case one or more files with the input data contain an initial header, RSVVP will automatically detect and skip it. In this case, the user will be warned a header has been detected and RSVVP will ask a confirmation about the number of lines which would be automatically skipped before reading the data series. If the user is not satisfied about the automatic detection of the header, it is possible to manually define how many lines to skip.

Abscissa	Ordinate
0.00000000	0.10000000
0.02000000	0.09900000
0.04000000	0.09800000
0.06000000	0.09700000
0.08000000	0.09600000
0.10000000	0.09500000
0.12000000	0.09400000
0.14000000	0.09300000

.....

**Figure A.1: Format of the test and true curves.**

Although no limitation is imposed or assumed for the units of both the abscissa and ordinate columns, the use of some preprocessing features like the SAE filtering option may only make sense for time history data (i.e., the first column represents time). Note that the actual version of RSVVP assumes that time is in units of seconds. In any case, the user must ensure that the physical meaning and the units of the input curves are consistent, especially when inputting more than one pair of curves in the Multichannel mode.

#### **A.2.1.2. Copy of the original input curves**

A copy of the original input curves is automatically saved into the folder ‘\Input\_curves’ in both the main directory of RSVVP and the ‘Result\_XX’ folder. Any file saved into the ‘\Input\_curves’ folder located in the main directory is deleted at the beginning of each new run of RSVVP.

#### **A.2.2. Loading a configuration file**

The user can also load a configuration file from a previous run of RSVVP. This configuration file contains all the necessary information to retrieve the files containing the original input curves and all the selected options for the preprocessing of the curves and the evaluation of the metrics. This configuration file can be loaded into two different ways:

- Run mode and
- Edit mode.

When the 'run mode' is selected (checkbox 'Run completely' is checked), RSVVP reads the configuration file and automatically evaluates the comparison metrics using the options stored in to the configuration file (e.g. preprocessing, metrics selection time intervals, etc.). This option is a useful tool for providing documentary proof of the values of the comparison metrics obtained during the verification/validation process or to simply enable the user to re-run a previously saved session. During the re-run from a previously saved configuration file, the user can reproduce all the user time intervals of the original run, some of them (but with the constraint to follow the original sequence of the intervals) or also calculate the comparison metrics on new user-defined time intervals. The original configuration file can be updated with the new user defined time intervals at the end of the calculation.

In 'Edit mode' (checkbox 'Edit curves/preprocessing' is checked), RSVVP loads the original input curves and automatically preprocesses them according to the options saved into the configuration file. Once the curves have been preprocessed, the user can go back to any point and modify any of the preprocessing options and even re-load or replace any of the original input curves. This option can be very useful when the analyst wants to perform a very similar evaluation several times and apply only minor changes in order to experiment different sets of pre-processing options.

*At the startup of RSVVP, select between comparing one or more couples of curves and loading the configuration file from a previous run.*

*By default, RSVVP limits the rate at which the curves are resampled to a maximum sampling rate of 10 kHz (see paragraph A.2.4 for more information). In case it is desired to set a higher limit for the resampling of the couple(s) of curves, a new limit can be selected from the drop-down menu 'Resampling rate limit' (Figure A.2).*

*After the resampling limit has been set, click the button corresponding to the desired type of comparison ('Single Channel' or 'Multiple Channel') to input the set(s) of curves and select the preprocessing options.*

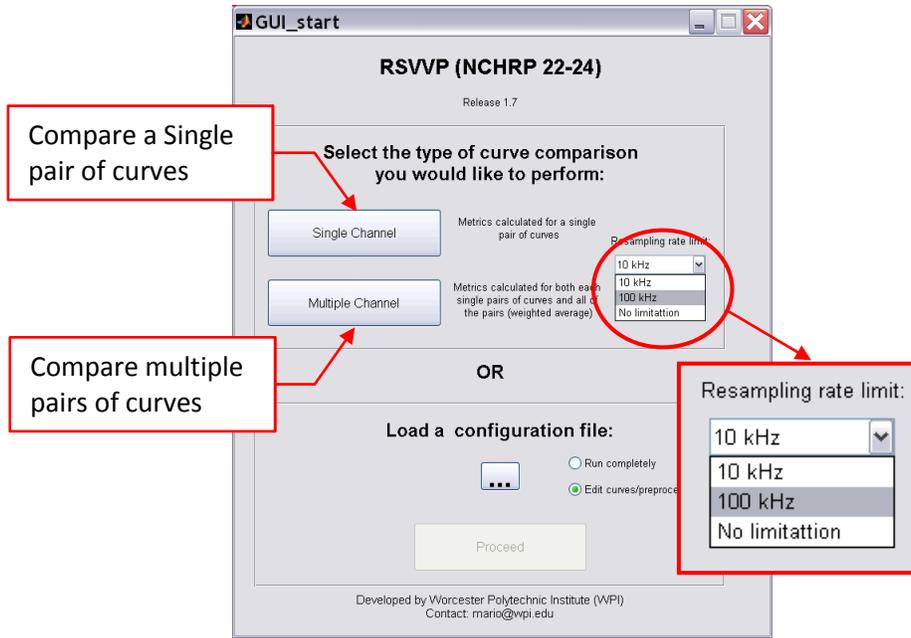


Figure A.2: Selection of the type of comparison and resampling limit.

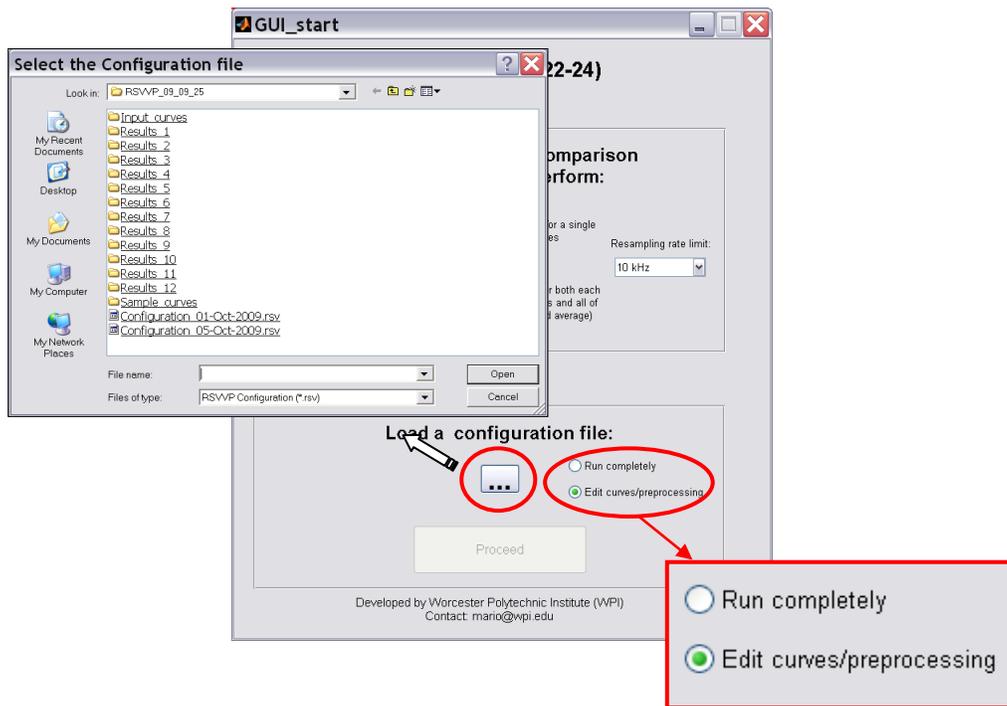


Figure A.3: Selection of the configuration file.

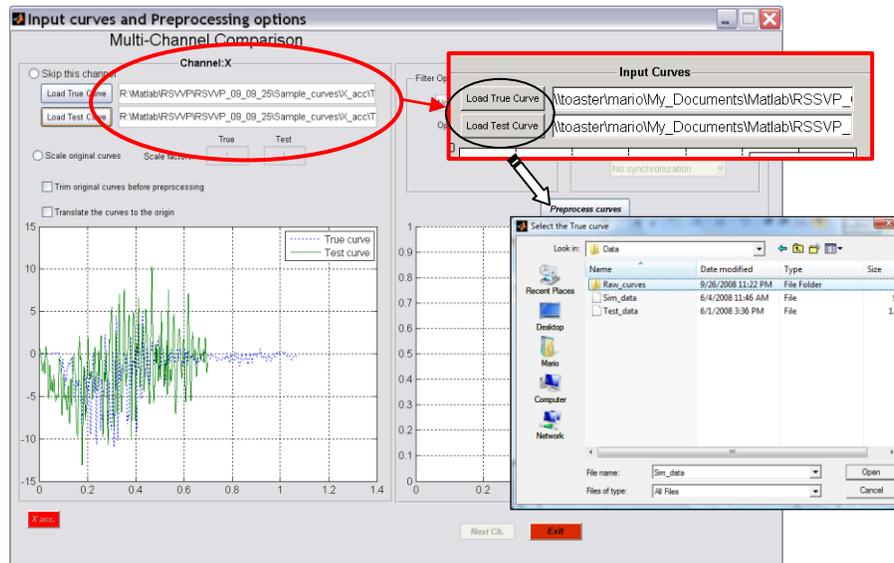
Otherwise, in case a configuration file has to be loaded, click the button with three dots to select the file with the desired configuration. Once the configuration file has been input, the button

*'Proceed' becomes active. Press it to load the files and options from the configuration file (Figure A.3).*

*Note 1: when a configuration file is loaded in 'run mode', any selection made by the user to limit the resampling rate of the input curves is overwritten by the selection loaded from the configuration file. In order to change the re-sampling limit, load the configuration file in 'edit mode'.*

*Note 2: by default, RSVVP loads the configuration file in Edit mode; to switch to Run mode, click the radio button 'Run completely' before pressing the button 'Proceed'.*

To input the test and true curves, click on the respective buttons located at the top of the graphical interface and select the file containing the input data (Figure A.4). Once the curve has been selected, a preview of the raw curves will be shown in the left graph area of the main window.



**Figure A.4: Input of the test and true curves.**

For the multichannel input, clicking on the 'Next Ch.' button located at the bottom of the screen moves on to the input of the next channel (the name of the actual channel appears at the top of the window). Note that, in order to proceed to the next channel, it is necessary to input the actual channel and perform the preprocessing. Alternatively, it is possible to skip any of the six available channels by pressing the radio button 'Skip this channel' located on the top of the window.

It is always possible to go back to the previous input channel by clicking the button 'Prev. Ch.'. Once reached the last channel (i.e., 'Pitch rate'), click the button 'Proceed to metrics evaluation' to proceed to the selection of metrics.

If desired, it is also possible to go directly to a specific channel without the need to sequentially pass through all the channels between the actual and the desired target channel. In the multichannel mode, six buttons are located at the bottom of the GUI (Figure A.5). Each button corresponds to a specific channel and it is possible to select only the channels which have already been input. The button corresponding to the actual channel in use is highlighted in red.

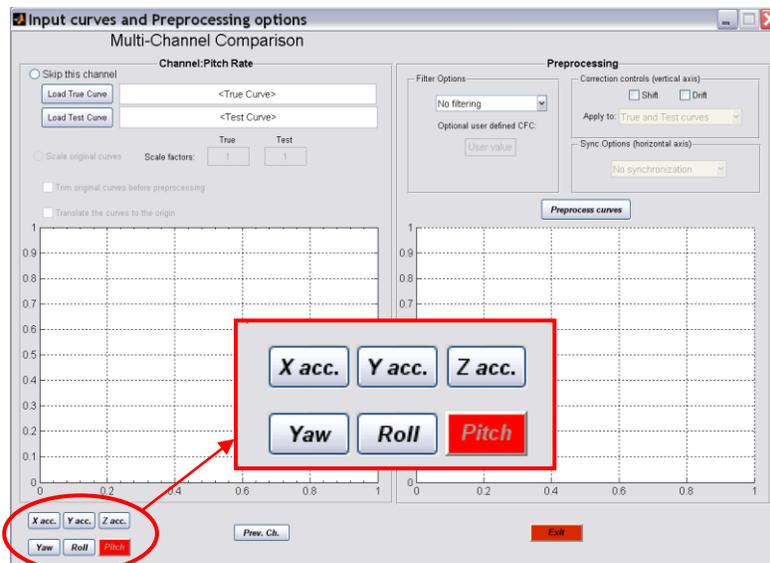


Figure A.5: Buttons to move directly to the desired channel.

### A.2.3. Initial preprocessing

Once both the test and true curves have been input, the user is given the choice to manually perform the following three operations before applying any other preprocessing option:

- Scale original curves
- Trim original curves before preprocessing and/or
- Translate original curves to the origin.

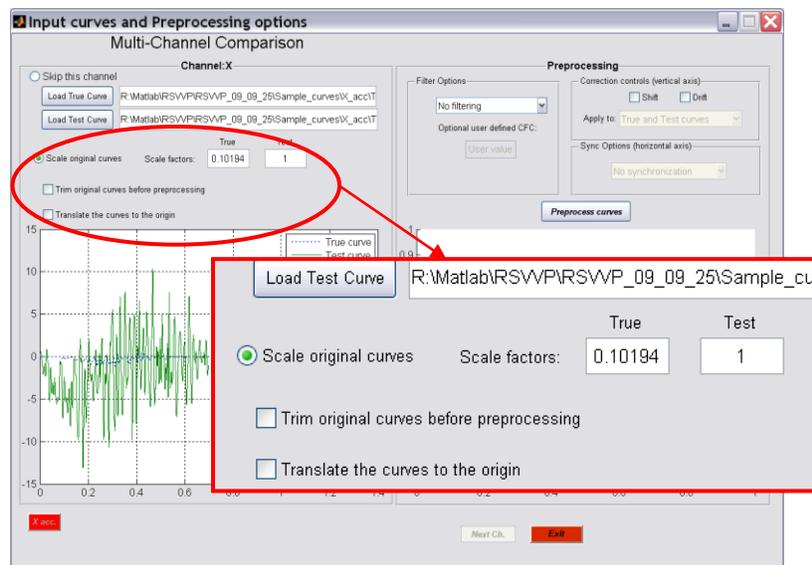
The option to scale the original curves allows scaling the original time histories using user-defined scale factors. The true and test curves can be scaled by different scale factors. This option may be used to invert the sign of time histories or to convert units (e.g., accelerations can be converted from  $m/s^2$  to  $g's$ ).

The option to manually trim the input curves before applying any pre-processing operation will allow the user to trim the initial ‘flat segment’ which typically characterizes the time histories of both experiments and simulations before the impact occurs. Also, this option can be used in case the simulation and experimental curves start diverging at a certain time to allow for a better synchronization of the curves by trimming them immediately before the diverging point. Although it is possible to specify a user defined time interval on which to evaluate the comparison metrics, it is advisable to trim the input curves if they have a ‘null head’ or ‘null tail’

before performing any preprocessing operation because this should improve the results of the synchronization procedure. Either the test or true curve can be trimmed independently from each other in case each curve has been collected with a different initial start time (i.e., the head of the test and true curve have a different length). In case the original input curves are characterized by a high level of noise, it is possible to filter them before defining the upper and lower trim boundaries.

The option to translate the raw curves allows the user to shift the original data points in order to ensure that the beginning of the abscissa vector is coincident with the origin (e.g., if time histories are input, the time vector can be shifted to the origin). This option works for either positive or negative value.

*The radio button to scale the input curves and the checkboxes to activate the option to manual trim and/or translate the curves to the origin can be selected only after both the test and true curves have been input (Figure A.6).*



**Figure A.6: Checkboxes for the manual trim and the translation of the raw curves.**

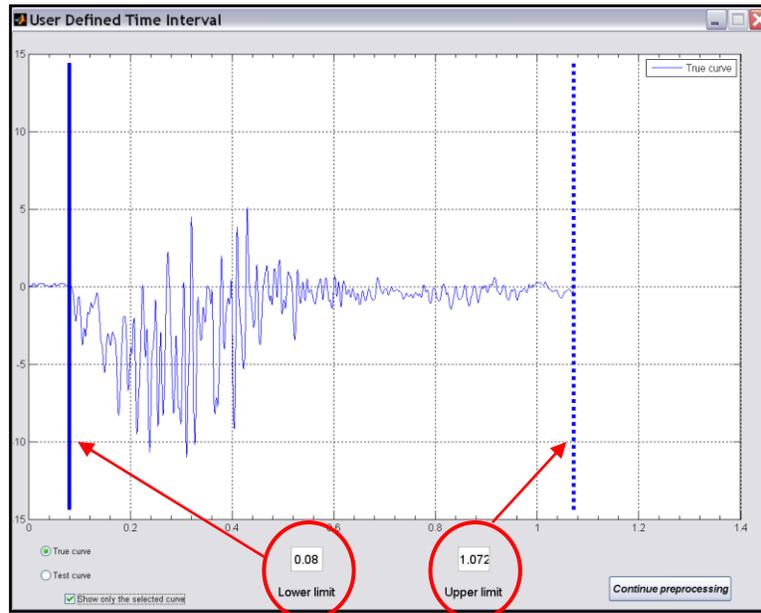
### **Curve scaling**

*To scale either the true or test curve or both of them, check the radio button 'Scale original curves'. Input the scale factor for the true and/or test curves into the respective fields 'True' and 'Test' located beside the radio button. Each time a new scale factor is defined for either the true*

or the test curve (or the scaling option is deselected), the graphs of the original curves are updated to the latest factor defined.

### **Curve trimming**

To trim the original curves, check the box 'Trim original curves before preprocessing'. This action will open the pop-up window shown in Figure A.7.



**Figure A.7: Window for the manual trim of the input curves.**

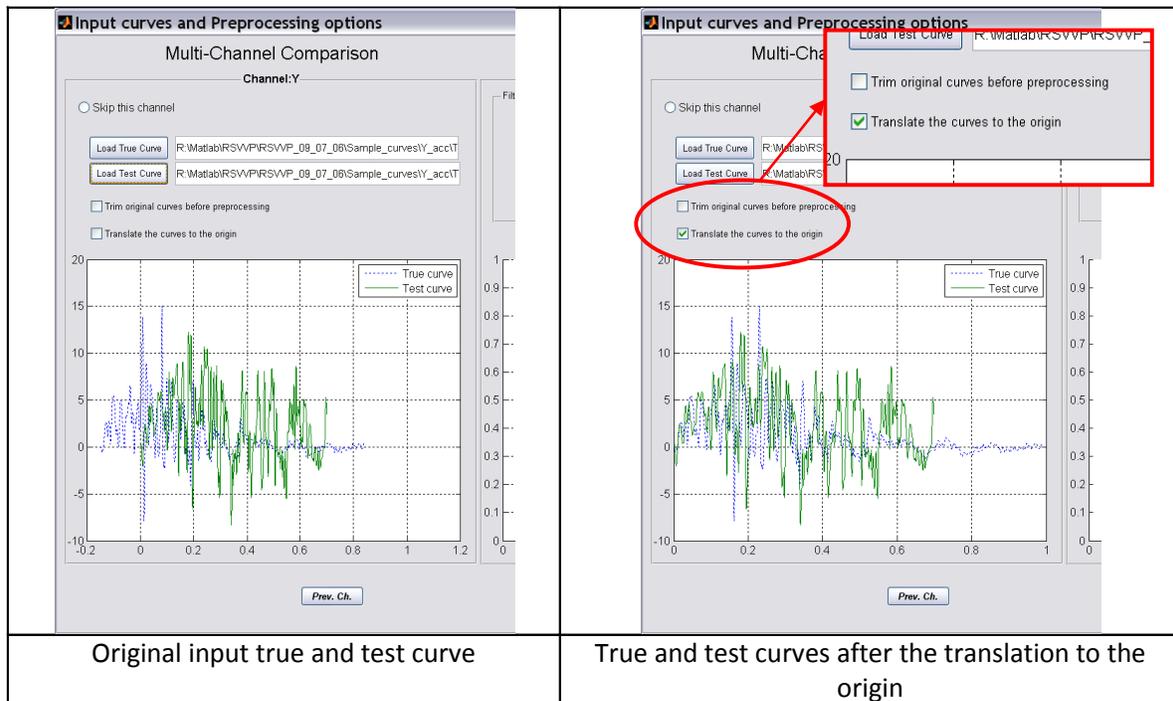
It is possible to filter the original curves before trimming them by selecting the desired CFC value from the drop-down menu located in the 'Filter option' box. In case the desired CFC value is not listed, select the option 'User defined CFC...' from the drop-down menu and define the specific value for the CFC in the field below. For more information about filtering, see paragraph A.2.4.1.

The fields 'Lower limit' and 'Upper limit' show the boundary values for the curve selected using the radio buttons for either the test or true curve. Only one curve at a time can be selected in order to allow for independent trimming of each of the two curves. The curve selection is performed by using the radio buttons for either the true or the test curves which are located at the bottom left of the window. A straight and dotted line respectively indicates the lower and upper limit in the graph area. Both the lines move according to the value input in the user fields (blue and green color are used for the true and test curves, respectively). By default, both the test and true curves are shown in the graph area but it is possible to limit the plot to only the selected curve by checking the option 'Show only the selected curve'. This option can be useful

when the graphs of the two curves cannot be easily distinguished. If the option to manually trim the curves has been selected, the curves are automatically translated to the origin so there is no need to select also this option.

### **Curve translation**

The checkbox to translate the original raw curves is selectable only in the case where manual trimming has not been selected. This option is useful whenever one or both the original input curves are shifted respect to the origin. A typical application is shown in Figure A.8



**Figure A.8: Shift of one of the two input curves to the origin.**

**Note 1:** If the option to scale the original curves is changed or if any of the two scaling factors are changed, RSVVP automatically updates both the graph of the original and preprocessed curves.

**Note 2:** If either one of the checkbox selections or if any of the two input curves are changed all the preprocessing operations applied to the curves are reset.

**Note 3:** NEITHER of these two operations is included in the copy of the original input curves automatically saved by RSVVP.

#### **A.2.4. Preprocessing**

After the true and test curves have been selected and the preview graph shown, RSVVP is ready to perform some basic and necessary pre-processing operations on the original input curves and also some optional preprocessing operations which can be selected by the user on the basis of the appearance of the original test and true curves.

Regardless of whether any optional preprocessing feature has been selected or not, RSVVP still performs a series of basic and necessary preprocessing operations to the original input curves:

- Resampling and
- Trimming.

The user has no control over these necessary steps.

The couples of curves input in RSVVP are resampled to ensure they use the same sampling rate. In case a limit has been defined for the maximum sampling rate (see paragraph A.2.1), if the sampling rate of one of the two curves is larger than the limit set, it will be automatically limited to the chosen value. This would avoid an excessively long time of calculation of the metrics. In case multiple channels are input, the sampling rate chosen for the first available pair of curves is used also for all the other channels.

In order to compute the comparison metrics, the longer curve has to be trimmed to the same length as the shorter curve. In the multichannel mode, after each pair of curves has been trimmed, the various channels are further trimmed to the length of the shortest channel.

To proceed to the next step (i.e., metrics selection) it is necessary to push the '*Preprocess curves*' button even if no optional preprocessing options have been selected.

Three optional pre-processing operations are available:

- Filtering,
- Shift/drift control and
- Synchronization.

All three of these pre-processing controls are optional and can be selected independently from each other.

Once the desired preprocessing options have been selected, press the ‘*Preprocess curves*’ button located immediately below the Preprocessing box to create a preview of the preprocessed curves.

If the preview is not satisfactory, any of the previous options can be changed and performed again until a satisfactory preview graph is obtained.

If RSVVP is running in Multiple Channel mode, the synchronization option is not active immediately. The synchronization and other options for the pre processing of multiple curves will appear in a second window available only when the Multiple channel mode has been selected.

Following is a description of each of the pre-processing options.

#### **A.2.4.1. Filtering**

RSVVP gives the user the option of filtering the two input curves. This option can be very useful when the original input curves are noisy (e.g., noise created by the transducer during the acquisition process of experimental curves or undesired high-frequency vibrations). In order to obtain a value of the comparison metrics that is as reliable as possible, it is very important to remove noise from both the test and true curves. While noise derives from different sources in physical experiments and numerical simulations, the true and test curves should be filtered using the same filter to ensure that differences in the metric evaluation are not based on the difference in frequency content in the true and test signals.

The filter options in RSVVP are compliant with the SAE J211/1 specification. It is recommended that raw data be used whenever possible in the evaluation to avoid inconsistent processing of the two curves. It is also important that both the test and true curves are filtered in the same way to avoid errors due to different filtering techniques. Although there is no general limitation to the type of units used for the input to RSVVP, the SAE filtering option presumes that the curves are time histories with time values expressed in units of seconds. In a future release of RSVVP, the option to use different units for the time vector of the time histories will be implemented.

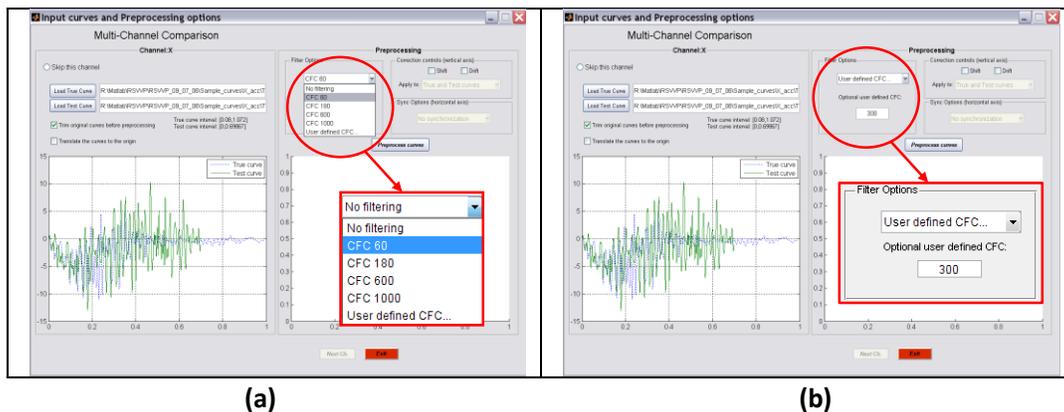
The user can select between the following SAE J211 Channel Frequency Classe (CFC) filters: 60, 180, 600 and 1000. Table A.1 shows the specifications of each CFC value as defined by SAE J211/1.

**Table A.1: Specifications for the usual CFC values.**

CFC value	3 dB limit frequency [Hz]	Stop damping [dB]
60	100	-30
180	300	-30
600	1000	-40
1000	1650	-40

While it is not recommended, if the user wants to use filter specifications different from the standard SAE J211 filters, user defined filters parameters can be specified.

*By default RSVVP does NOT filter the input curves. To apply the filter option, click on the drop-down menu in the 'Filter Options' box (Figure A.9a) and select the desired CFC value. If it is necessary to specify a value for the CFC which is not listed in the menu, select the option 'User defined CFC...' at the end of the list and input the desired CFC parameters in the 'Optional user defined CFC' field located right below (Figure A.9b). (Note: This field is active only if the 'User defined CFC' option is selected from the drop-down menu.)*



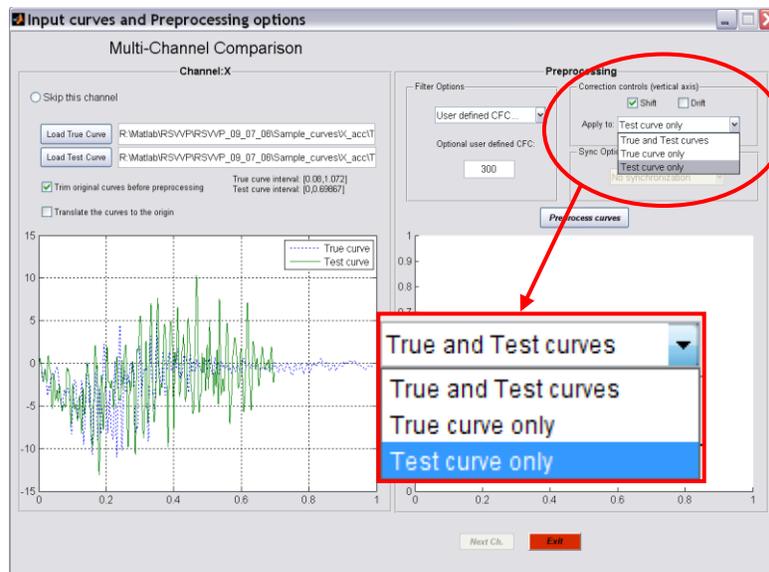
**Figure A.9: Filter Options' box - (a) drop down menu and (b) 'Optional user defined CFC' field (in case the user defined option has been selected).**

*Note: If the original curves have already been filtered during the optional manual trimming process (see paragraph A.2.3), the Filter Options' box will show the filtering option chosen at that time without allowing the user to make any change. If a different filtering option is desired, it is necessary to go back to the manual trimming box to make any change to the previous choice.*

#### **A.2.4.2. Shift/Drift controls**

Another preprocessing option supported by RSVVP is the possibility to correct any initial shift and/or drift in the curves. Experimental data sometimes contain shift and/or drift effects due to the change of temperature immediately before or during the test. The shift effect is an initial vertical shift of the curve due an increase of the temperature after the measurement gauges have been zeroed while the drift effect is a linear drift of experimental curve typical of the increase of the temperature during the test. The shift and drift controls of RSVVP correct the above mentioned effects and, therefore, can be very useful in case one or both the two input curves have been recorded from experimental tests and present either or both these data acquisition problems. As either the initial shift or drift of the test and/or the true curve are caused by an incorrect acquisition of the experimental data, these pre-processing options are important for an accurate evaluation of the comparison metrics. In generally, curves resulting from numerical solution should not need to use these options since shift and drift are features of sensor characteristics in physical tests. The use of the shift and drift options is, therefore, not recommended for curves resulting from computer simulations.

*Both the **shift** and **drift** controls can be activated independently from each other by checking the respective boxes. Once one or both of them have been checked, the user has the choice to apply the selected control/s to the true curve, the test curve or both the true and test curves (Figure A.10). By default these controls are inactive.*



*Figure A.10: Shift and Drift controls.*

#### **A.2.4.3. Curve Synchronization (Single-channel mode)**

RSVVP allows the user to optionally synchronize the two input curves before evaluating the comparison metrics. This option can be very useful if the original test and true curves have not been acquired starting at exactly the same instant (e.g., the test and true curve represent respectively a numerical simulation and an experimental test of the same crash test but the instant at which data collection was started is not the same). The synchronization of the two input curves is very important as any initial shift in the time of acquisition between the test and true curves could seriously affect the final value of the comparison metrics. For example, two identical input curves with an initial phase difference due to a different starting point in the acquisition process would probably lead to poor results of some of the comparison metrics.

Two different synchronization options are available in RSVVP. Both options are based on the minimization of a target function which is respectively: (1) the absolute area between the two curves (i.e., the area of the residuals) and (2) the squared error between the two curves. Although these two methods are similar, they sometimes give slightly different results. Selecting one of these methods will result in the most probable pairing point for the two curves.

Once the original curves have been preprocessed, the user is given the possibility to refine the synchronization further.

By default RSVVP does NOT synchronize the input curves. To apply the synchronization option, click on the drop-down menu in the 'Sync Options' box (Figure A.11) and select one of the two desired synchronization method: (1) Minimum absolute area of residuals or (2) Least Square error. Note that in Multi-channel mode the synchronization option (and other options for the pre processing of multiple curves) appears in a different window which is available only when the Multiple channel mode has been selected. Refer to paragraph A.2.5.

Once the curves have been preprocessed by pushing the 'Preprocess curves' button, a pop-up window will ask the user if the synchronization is satisfactory. If the 'No' button is pushed, another pop-up window with a slider will appear (Figure A.12). Moving the slider changes the initial starting point of the minimization algorithm on which the whole synchronization process is based. In this way, the user can manually adjust the synchronization process of the two curves.

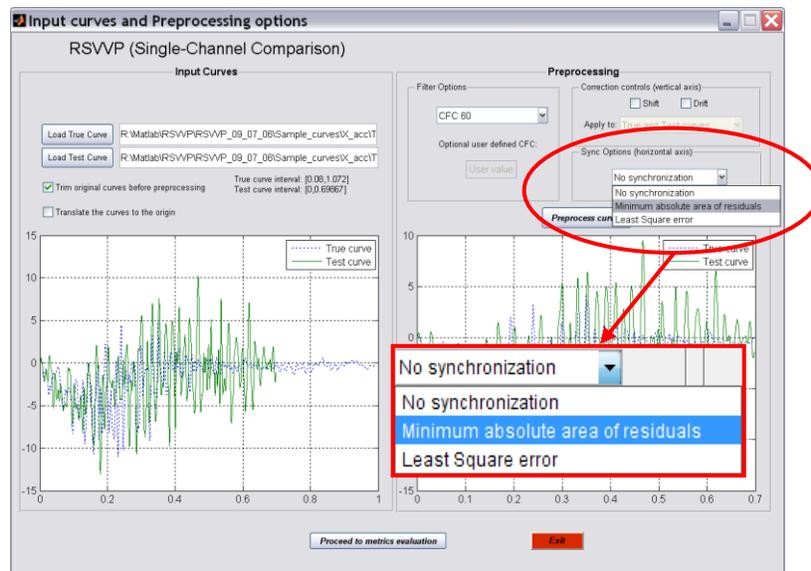
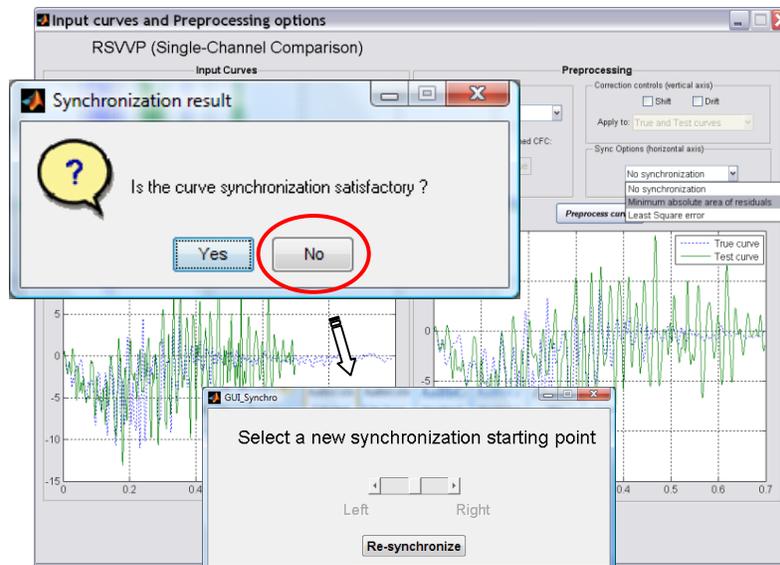


Figure A.11: Drop down menu of the 'Sync Options' box.



**Figure A.12: Selection of a new starting point (in case the user is not satisfied by the initial synchronization of the two curves).**

### A.2.5. Multichannel options

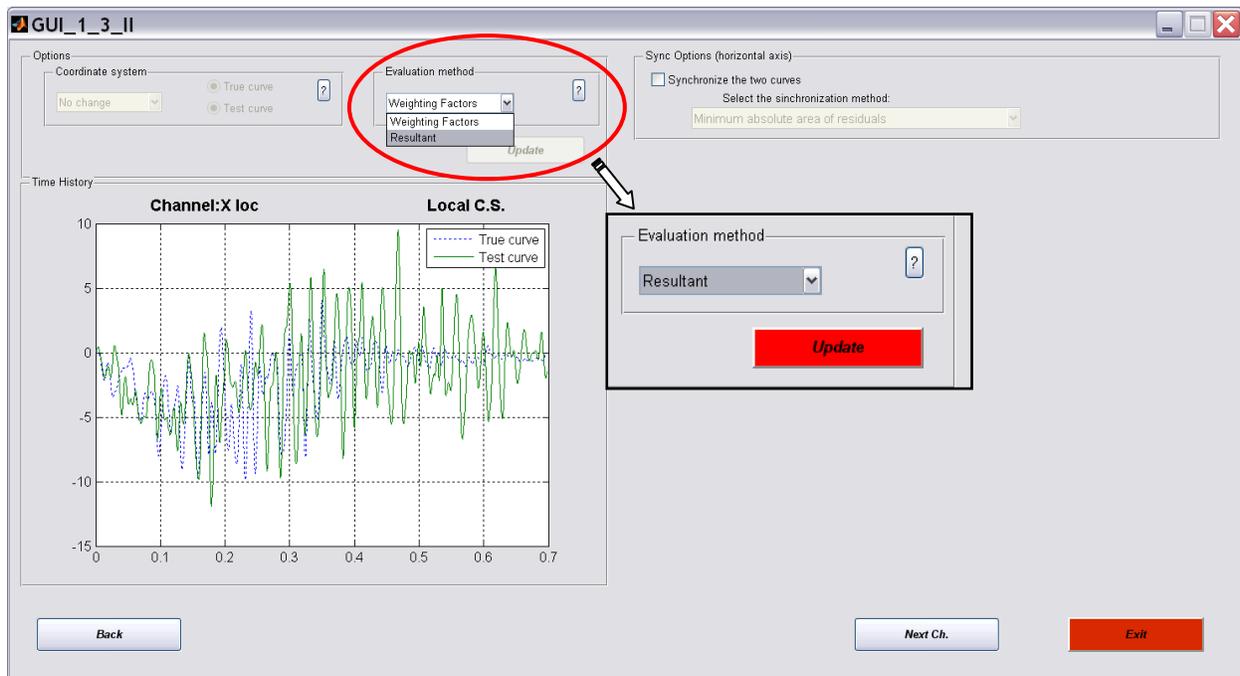
When RSVVP is running in multichannel mode, the user can select between two different methods to evaluate the comparison metrics:

- Weighting Factors and
- Resultant.

Selecting the ‘Weighting Factors’ method, the comparison metrics calculated using the pairs of curves from each channel are combined together using weighting factors, while the ‘Resultant’ option calculates the comparison metrics based on the resultants of respectively the acceleration and rotational rate channels. In either way, the metric values obtained using the ‘Multiple Channel’ case are intended to give a general assessment of the comparison using all the available channels simultaneously.

*When the last of the six input channels has been input press the button ‘Proceed to curves synchro’ to open a second window with options for the preprocessing of the curves and the selection of which method to use. (Note: If the last channel is skipped, RSVVP will automatically open this second GUI.)*

Select which method to use for the evaluation of the multichannel comparison metrics in the dropdown menu labeled 'Evaluation method' located in the 'Option' box (Figure A.13). The default method is set to 'Weighting Factors' and the graph in the 'Time History' box shows the first available channel. To switch to the resultant method, click on the drop down menu and select 'Resultant'. Once the method has been changed, the button 'Update' becomes red. Push this button in order to update to the new selected method. The graph in the 'Time History' box will now show the resultant acceleration.



**Figure A.13: Selection of the method for the computation of the multichannel metrics.**

After the desired evaluation method has been selected, it is possible to synchronize the curves shown in the graph area by clicking the checkbox 'Synchronize the two curves' located in the 'Synch options' box on the left side of the GUI (Figure A.14). The synchronization is optional and it is possible to skip this operation by simply leaving the checkbox unselected. If selected, the synchronization starts automatically. As for the single channel mode, two different synchronization methods are available: (1) minimum area of residuals and (2) least square error. If a different synchronization method is selected in the dropdown menu, RSVVP automatically synchronizes again the two curves accordingly. In case the user is not satisfied by the synchronization, it is possible to change the initial starting point used by the minimization algorithm.

To proceed to the next channel, press the button 'Next Ch.'. In case the resultant method has been selected, this will lead to the resultant of the rotational rate. When the last channel/resultant has been reached, it is possible to go to the next window by pressing the button 'Proceed to metrics selection'.

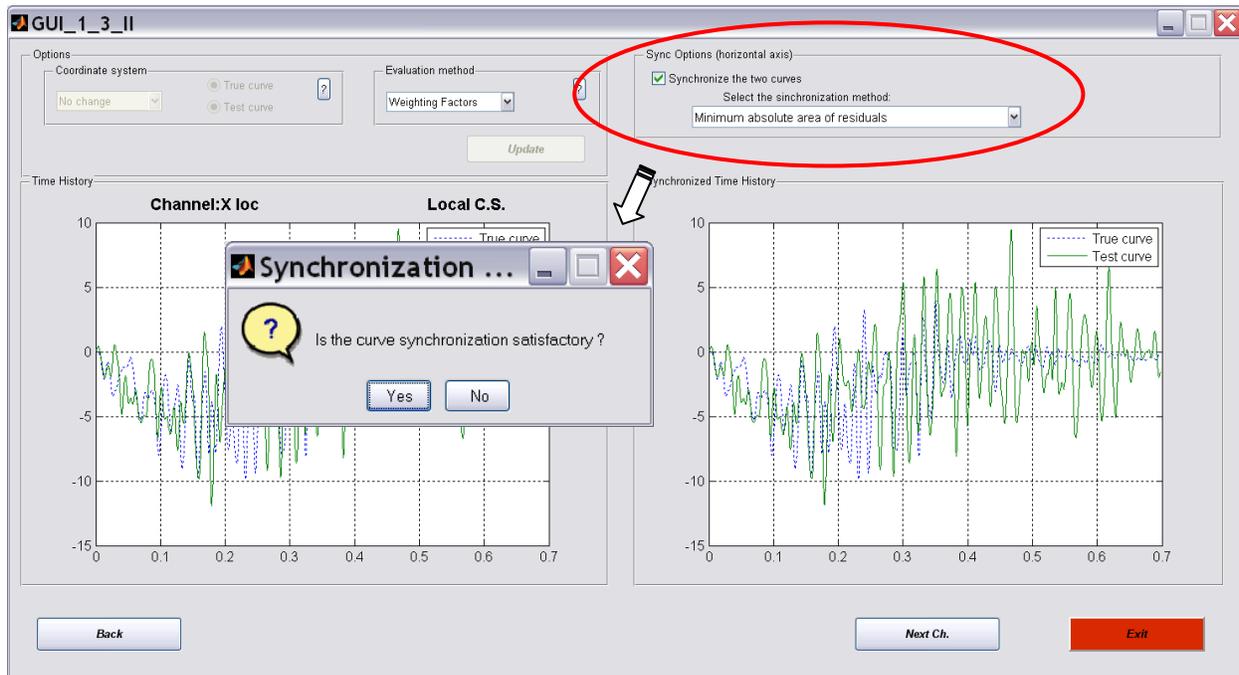


Figure A.14: Synchronization of the channel/resultant.

*Note 1:* Each time the evaluation method is changed, it is necessary to push the 'Update' button to make the change effective. Also, the change of the evaluation method resets any synchronization previously performed on any channel/resultant.

### A.3. METRICS SELECTION AND TIME INTERVAL

Once the test and true curves have been preprocessed, push the 'Proceed to metric evaluation' button to select the desired comparison metrics. In case of multichannel input, this button will appear once the curves for the last channel have been preprocessed.

Three main metric profiles can be selected in RSVVP:

1. NCHRP 22-24,

2. All metrics and
3. User selected metrics.

For each of the three available profiles, the comparison metrics can be evaluated on either the entire time window on which the true and test curves are defined or a smaller user-defined local time window. These features will be described in the following sections.

### **A.3.1. Metrics selection**

The NCHRP 22-24 profile is the default profile and it is strongly suggested that this profile be used when comparing a full-scale crash test to a simulation of a crash test; the other profiles are useful for other types of curve comparisons such as component level comparison of a material stress-strain curve. The NCHRP 22-24 profile evaluates the metrics recommended in the NCHRP 22-24 process for comparison of full-scale crash test results to numerical simulations. The metrics calculated are: (a) the Analysis of Variance (ANOVA) of the signals and (b) the Sprague-Geers MPC metrics. The ANOVA metrics are based on the residuals between the true and test curves while the Sprague and Geers metrics indicate the quality of comparison for the magnitude and phase of the test and true curves. In both cases, the curves are evaluated as-is, that is, in the units of the original curves and in the coordinate system of the original curves.

Although the ANOVA and the Sprague and Geers metrics are the only metrics recommended by the NCHRP 22-24 project, RSVVP gives the user the option of evaluating many other comparison metrics. The second profile ('All metrics') automatically selects all fourteen different comparison metrics that are available in RSVVP while, the third profile ('User selected metrics') allows the user to select the desired comparison metrics from the list of 14 different comparison metrics.

The fourteen different comparison metrics available in RSVVP are divided into three main categories:

- MPC metrics,
- Single-value metrics and
- ANOVA metrics.

A description of each metric is given in the Appendix A

The default metrics profile is the NCHRP 22-24 profile. It is also possible to evaluate all the fourteen available comparison metrics or just some of them by selecting either the option 'All metrics' or 'User selected metrics' from the drop down menu located at the top of the graphical interface (Figure A.15).

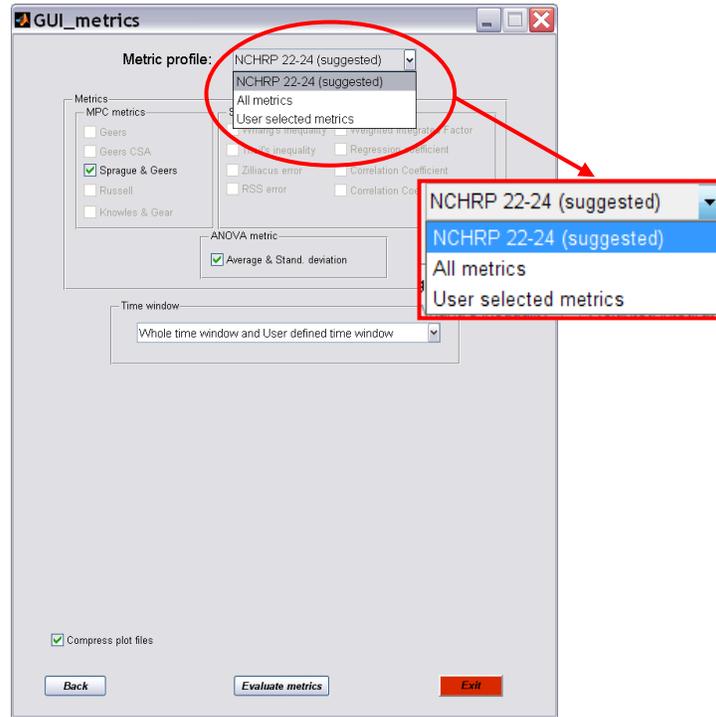


Figure A.15: Selection of the metric profiles.

If the 'User selected metrics' profile has been selected, the checkbox beside each available metric will become active and it will be possible to select which comparison metric to evaluate by checking the corresponding checkbox (Figure A.16).



*Figure A.16: Selection of the available comparison metrics ('User selected metrics' profile).*

### A.3.2. Time interval

In RSVVP, metrics can be evaluated on two different types of time intervals:

- Whole time window and
- User defined time window.

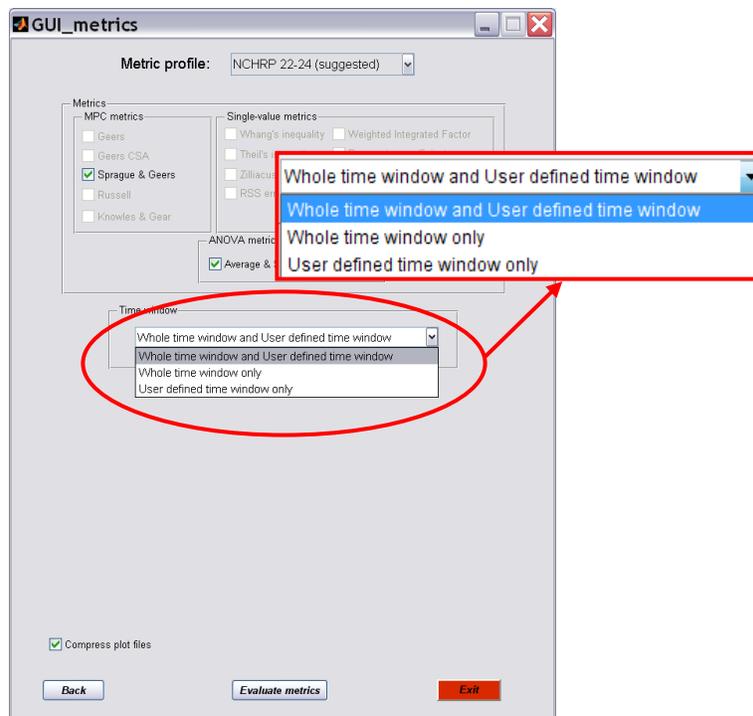
If the “Whole time window” option is selected, the metrics are evaluated considering the whole time interval on which both input curves were defined. Metrics cannot be evaluated on any time point for which just one of the two input curves was defined as only the maximum time interval common to both the two curves can be used in the comparison. If the “User defined time window” option is selected, the metrics are evaluated on one (or more) time intervals arbitrarily defined by the user.

Three different options are available for the time intervals:

- 1) Whole time window and User defined time window,
- 2) Whole time window only and
- 3) User defined time window only.

The user can select either the “Whole Time window” option or the “User defined time window” option or both of them (i.e., “Whole time window and User defined time window” option). In the latter case, RSVVP will proceed to evaluate the comparison metrics first on the “Whole Time interval” and, after showing the results, it will prompt the user to define an arbitrary “User Defined Time interval” on which to evaluate the metrics on a local base.

*By default RSVVP evaluates the selected metrics on both the whole time interval and the user selected time interval. If only the whole time interval or one (or more) user defined time interval(s) is desired, select the corresponding option in the drop down menu located in the ‘Time window’ box, below the list of metrics (Figure A.17).*



*Figure A.17: Selection of the type of time window(s).*

### A.3.3. Compress image files

During the computation of the metric, RSVVP creates several graphs and saves them as bitmap images (.bmp). As the cumulated size of these entire image files can occupy many megabytes, by default RSVVP compress them in a convenient .zip format. In case the user would prefer to have the images immediately available, it is also possible to save them without compression.

By default RSVVP saves the graphs of the metrics time histories, true and test curves, residual plots and other results in a convenient zip format. If an uncompressed format for the image files is desired. Uncheck the box titled 'Compress plot files' (Figure A.18).

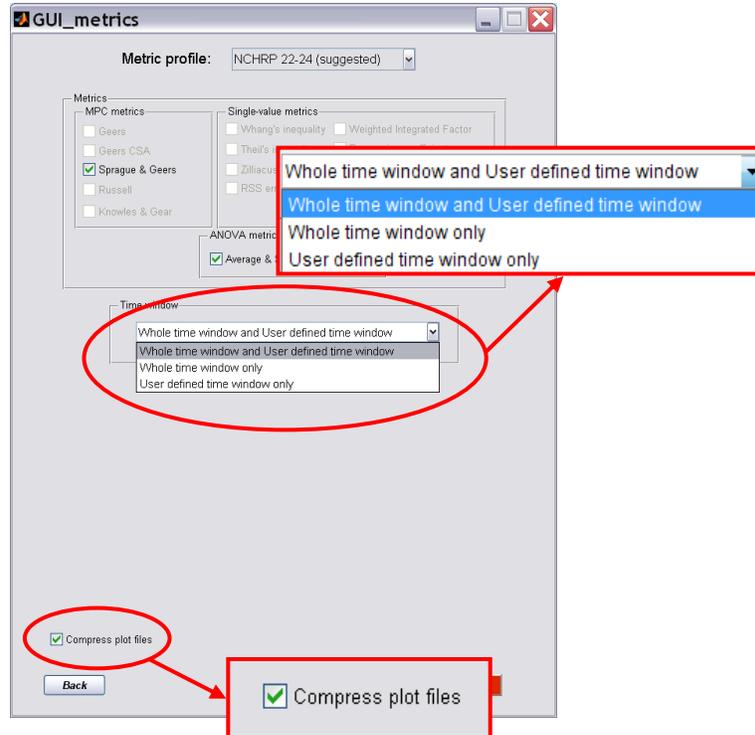


Figure A.18: Option to compress/uncompress the image files created by RSVVP.

#### A.4. METRICS EVALUATION

Once the input curves have been pre-processed, RSVVP proceeds to evaluate the metrics on the time interval selected by the user in the previous step. In multichannel mode, RSVVP first calculates the value of the metrics for each channel (weighting method) or resultant time history (resultant method) and combines together the multiple values for each metric into a single value by calculating a weighted average of the value from each channel in the case of the weighted method or by calculating a simple average in the case of the resultant method.

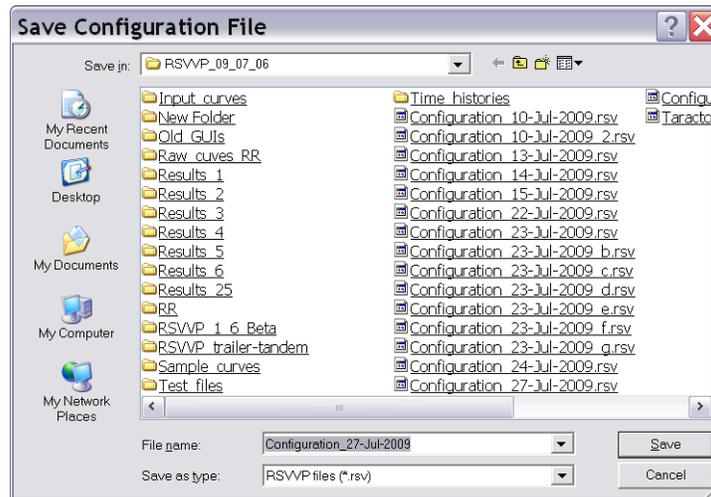
In order to start the evaluation of the metrics, press the 'Evaluate metrics' button located at the bottom of the window (Figure A.19). It is possible to go back to the main graphical interface to

change any of the selected input curves and /or modify any of the preprocessing options by clicking the 'Back' button.



Figure A.19: 'Evaluate metrics' button starting the metric evaluation.

Before the metrics are evaluated, a pop-up window appears asking the user to indicate a location and file name for saving the configuration file which contains all the information necessary to re-run the comparison in the future (Figure A.20). By default, the location of this file is the RSVVP directory and the name of the configuration file is 'Configuration\_Day-Month-Year.rsv', where Day, Month and Year are set according to the actual date.



**Figure A.20: Selection of the folder and name for the configuration file.**

*Note 1: A copy of the configuration file is also saved into the folder containing the results which is created by RSVVP at the end of the run (see chapter A.6 for the details about the result folder).*

*Note 2: The configuration file can be used for two main reasons: (i) allow the user to quickly re-input a set of curves and configurations and modify it or (ii) document a previous run by providing the possibility of fully recalculating the comparison metrics.*

#### **A.4.1. Whole time window**

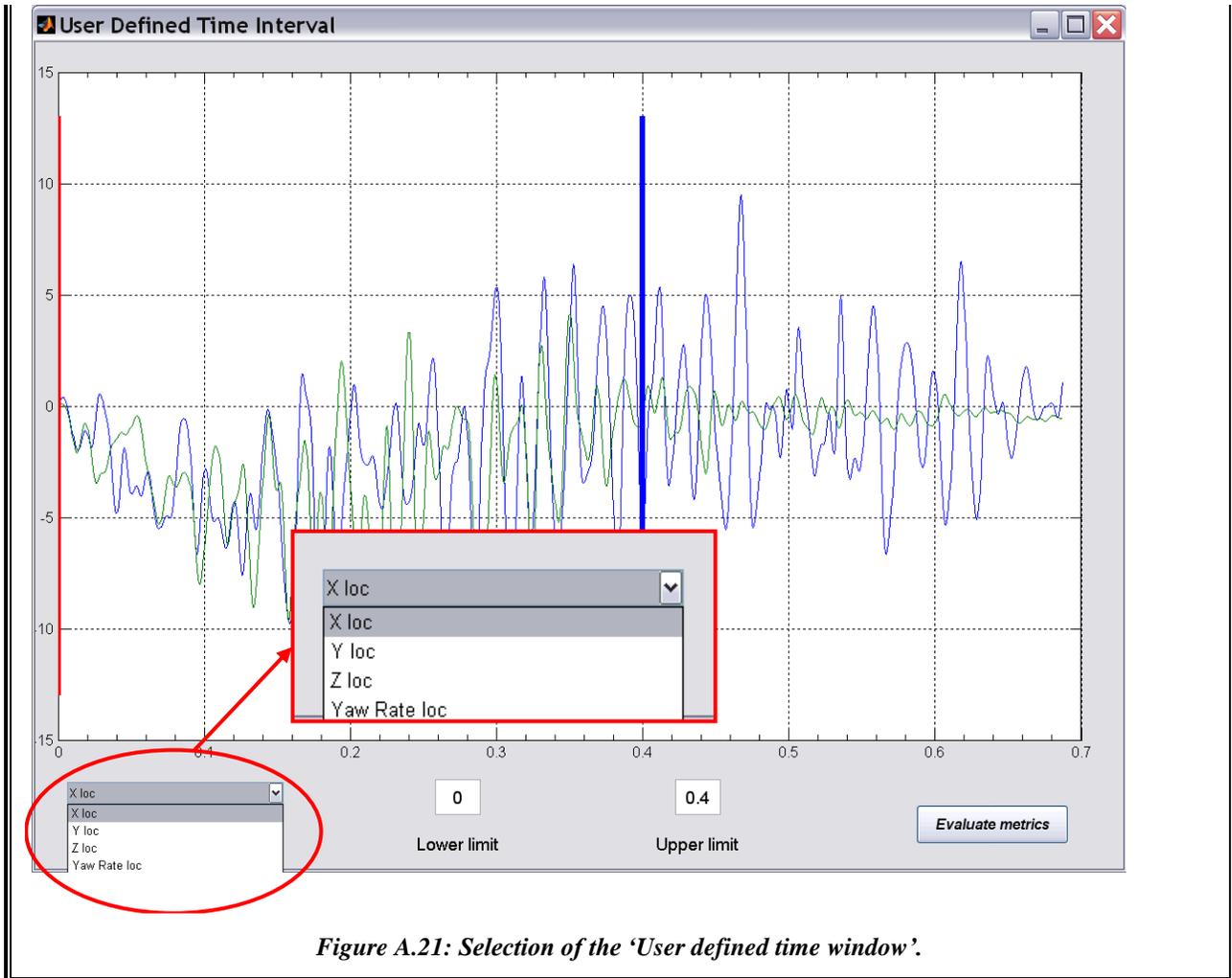
No action is needed to define the time interval for the “Whole time window” option (i.e., options 1 and 2 for the time interval selection) as RSVVP will automatically consider the maximum time interval possible in which both the true and test curves are defined.

#### **A.4.2. Definition of a User defined time window**

If a “User defined time window” has been defined (i.e., options 1 and 3), RSVVP will prompt the user to select the upper and lower boundaries of the local time interval on which the comparison metrics will be evaluated. It is possible to evaluate the metrics on as many user defined time windows as desired; after the results of the user defined time window have been shown, RSVVP will prompt the user for a new *User Defined* time window. The results obtained for each time interval will be saved separately.

*To create a ‘User defined time interval’ it is necessary to specify the lower and upper boundaries of the window. RSVVP shows a window with a graph of the test and true curves and two blank fields at the bottom which are used to define respectively the time value of the lower and upper boundary; fill in the desired values and press the ‘Evaluate metrics’ button to start the evaluation of the metrics on the defined interval. Once values are input into the fields, the upper and lower boundaries are graphically shown as vertical lines in the graph area with the test and true curves. For multichannel input, a drop box menu located at the bottom of the window allows the user to select which channel to show in the plot area.*

*Note 1: The selected upper and lower boundaries do not change when a new channel is plotted as they share the same interval for each channel in the multi-channel option.*



## A.5. SCREEN OUTPUT

For each of the time intervals on which the comparison metrics were evaluated, RSVVP shows various screen outputs to present the results:

- Graph of the test and true curves,
- Graph of the integral of the test and true curves,
- Values of the comparison metrics,
- Graph of residual time history,
- Graph of the residual histogram and
- Graph of the residual cumulative distribution.

Note that comparison metrics are always computed using the curves shown in the graph of the true and test curves. The integrals of the true and test curves are shown only to allow the user to have an easier physical interpretation of the original acceleration curves.

Figure A.22 and Figure A.23 show the typical output screen for the NCHRP 22-24 profile and the other two metric selection profiles respectively (i.e., 'All metrics' or 'User defined' profiles). In case the NCHRP 22-24 profile was selected, only the values of the Sprague and Geers and ANOVA metrics are shown. The word 'Passed' and a green square beside the value of each metric indicate that the NCHRP 22-24 acceptance criterion for that specific metric has been passed while a the word 'Not passed' and a red square indicate that the criterion has not been passed. For the other two general metrics profile all the metrics are shown and, for those which have not been calculated (i.e., metrics not checked by the user in the 'User defined' profile), the word N/A appears. In this case, no criteria are associated to the metrics value and the user has to make his/her personal judgment. Also, only the graph of the true and test curves is shown.

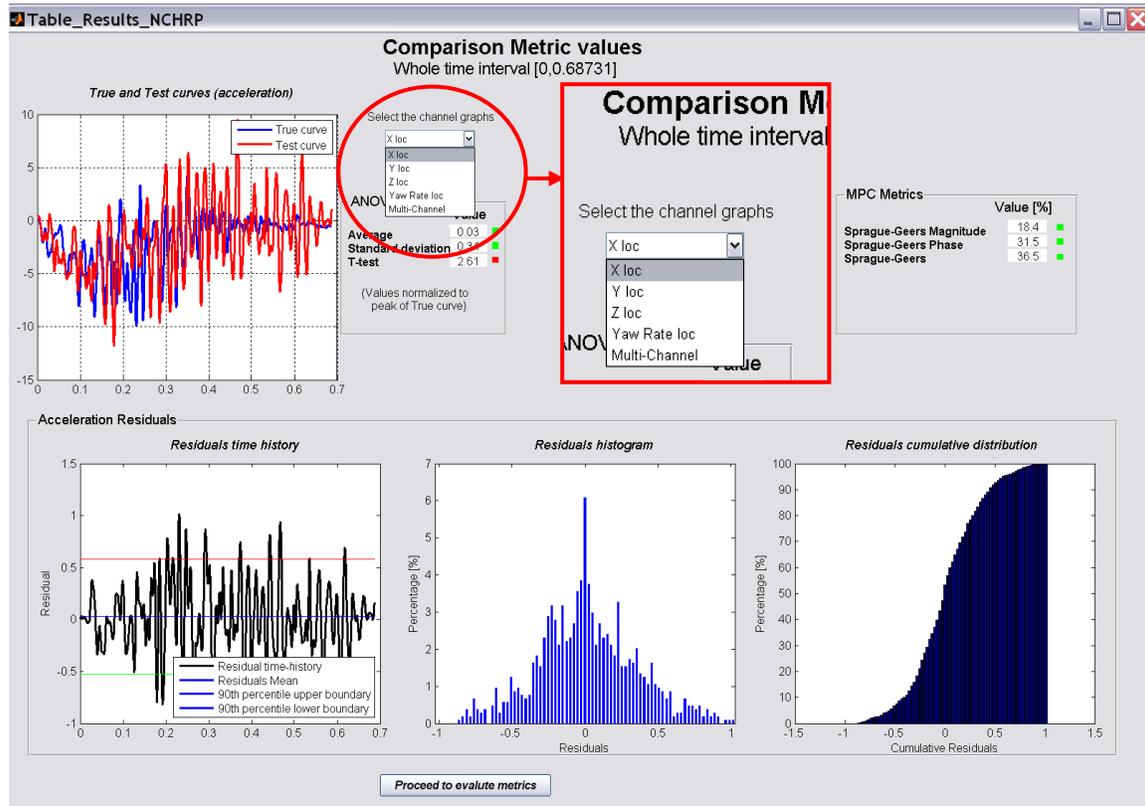
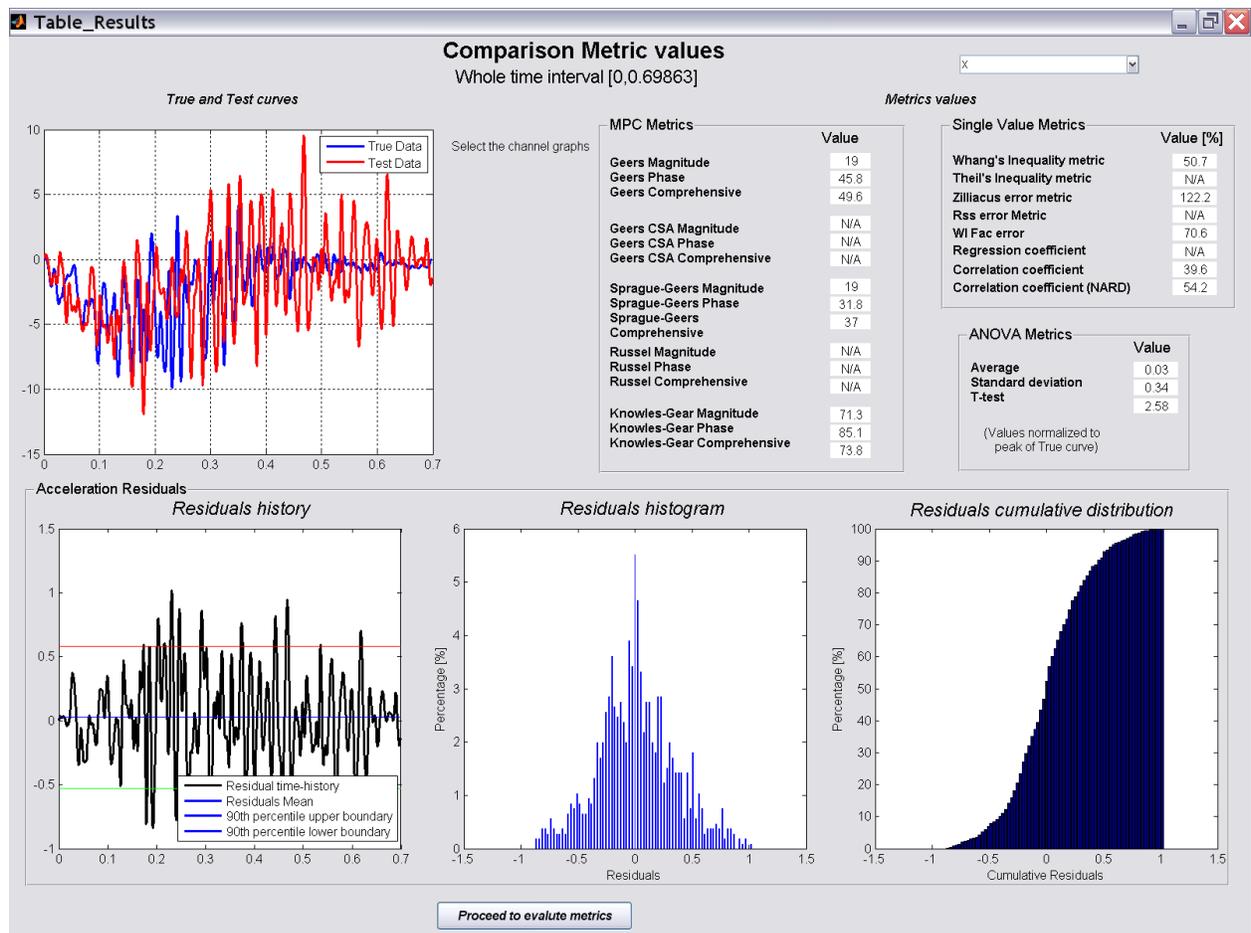


Figure A.22: Screen output for: (a) the NCHRP 22-24 profile and (b) the 'All metrics'/'User defined' profiles.

*For multichannel input, if the weighting factors method has been selected, it is possible to show the either the results calculated for any of the single input channels or the weighted average value by using the drop-down menu located beside the plot area (see Figure A.22). When the Multi-channel choice is selected from the drop-down menu, a histogram graph of the weighting factors used to compute the metric values in the multichannel mode is plotted. This gives an immediate understanding of the weight of each input channel respect to the others in the evaluation of the equivalent multichannel metrics.*

It may be necessary to wait a few seconds before the metric values and the graphs are updated to a new selected channel.



**Figure A.23: Screen output for: (a) the NCHRP 22-24 profile and (b) the 'All metrics'/'User defined' profiles**

If, during the selection of the profile for the time interval it was chosen to calculate the comparison metrics on user-define time segments, the user has the option to: (1) proceed to the

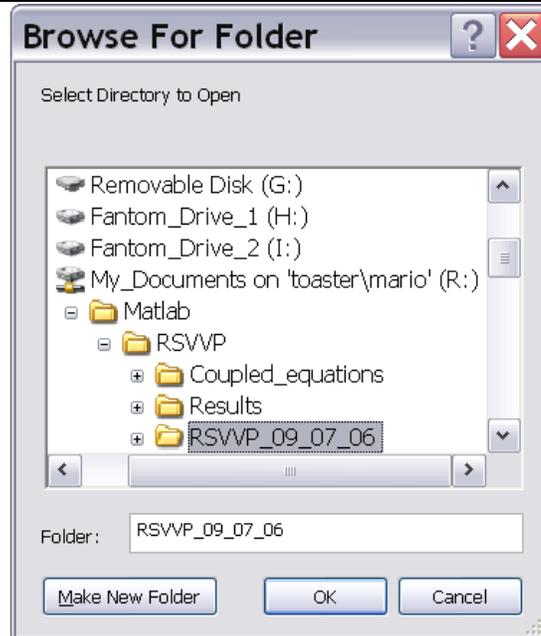
evaluation of a new interval and/or (2) to save the results and quit the program. Push the button corresponding to the action you want to take and wait till the next message will appear. In case both the whole and user defined time interval have been selected, the user will have to evaluate the metrics at least on one user-defined time interval before the option to save the results and quit RSVVP would appear.

## **A.6. OUTPUT OF RESULTS**

During the curve preprocessing and the evaluation of the metrics, RSVVP creates different types of output. All the output data are saved into a folder which can be indicated by the user. If no output folder is selected, RSVVP automatically saves the results in a folder called ‘\Results\_X’, where X is an incremental numbering (i.e., 01, 02, etc). At the beginning of the run, RSVVP checks for the existence of any previous sequence of folders called “\Results\_X”, where X represents the specific number of each Results folder, and creates a new Results folder appending as suffix the first available number.

By either saving the results into a user-defined or an automatically numbered folder, the user can easily keep track of different runs without the need to copy the Results folder to a new location before running RSVVP again. The folder ‘\Results\_X’ is created in the folder where RSVVP was installed (e.g., if the directory where RSVVP was installed is *C:\RSVVP* and there was already a previous folder named Results\_3, the Output folder is *C:\RSVVP\Results\_4\*).

*After the button ‘Save results and Exit’ has been pushed in the window which shows the results for the last interval, a window appears asking the user to select the folder where to save all the results (Figure A.24).*



*Figure A.24: Selection of the folder for saving the results of the RSVVP run.*

*In case, it is also possible to create a new directory by clicking the button 'Make New Folder'.*

*If no selection has been made or if the cancel button has been pressed, RSVVP will automatically create a folder named 'Results\_X' in the main folder of RSVVP, where X is an incremental numbering.*

*The process of saving of the results into the selected may take some minutes. During this period, RSVVP shows the message in Figure A.25.*



*Figure A.25: Message shown during the creation of the Excel file.*

### **A.6.1. Table of results (Excel® worksheet)**

*The final values of the comparison metrics are saved in the Excel file Comparison Metrics.xls. In the spreadsheet, the values of the comparison metrics for each time interval considered during the evaluation process are saved into separate columns. The label of each column indicates the*

*boundaries of the specific time interval. Figure A.26 shows a typical layout of the Excel output file.*

	Whole time interval [0,0.5474]	User time interval #1 [0.08005,0.19995]	User time interval #2 [0.12005,0.21995]
<b>MPC Metrics</b>			
Geers Magnitude	Value [%] 7.1	Value [%] 4.7	Value [%] 10.5
Geers Phase	23.9	22.1	21.4
Geers Comprehensive	24.9	22.6	23.8
Geers CSA Magnitude	N/A	N/A	N/A
Geers CSA Phase	N/A	N/A	N/A
Geers CSA Comprehensive	N/A	N/A	N/A
Sprague-Geers Magnitude	N/A	N/A	N/A
Sprague-Geers Phase	N/A	N/A	N/A
Sprague-Geers Comprehensive	N/A	N/A	N/A
Russell Magnitude	5.6	3.8	7.9
Russell Phase	22.5	21.6	21.2
Russell Comprehensive	20.5	19.4	20.1
Knowles-Gear Magnitude	58	101.1	1573.2
Knowles-Gear Phase	1.8	0	0
Knowles-Gear Comprehensive	53	92.3	1436.2
<b>Single Value Metrics</b>			
Whang's inequality metric	Value [%] 38.5	Value [%] 36.5	Value [%] 38.1
Theil's inequality metric	N/A	N/A	N/A
Ziliacus error metric	76.8	76.5	85.9
RSS error metric	N/A	N/A	N/A
WIFac_Error	N/A	N/A	N/A
Regression Coefficient	66.7	49.9	65.2
Correlation Coefficient	N/A	N/A	N/A
Correlation Coefficient(NARD)	76.1	77.9	78.6
<b>ANOVA Metrics</b>			
Average	Value 0.01	Value 0.04	Value 0.05
Std	0.15	0.25	0.16
T-test	7.21	7.39	14.43
T/T_c	2.81	2.88	5.63

**Figure A.26: Layout of the Excel table containing the metrics results for the various time intervals.**

*A summary of the input files and preprocessing options for each channel is written at the end of the Excel file (Figure A.27). If RSVVP is run in multichannel mode using the weighting factors method, the weighting factors and the metrics values calculated for each separate channel are saved into the Excel file as well. In particular, the values of the comparison metrics calculated for each channel are saved into separate sheets of the same Excel file.*

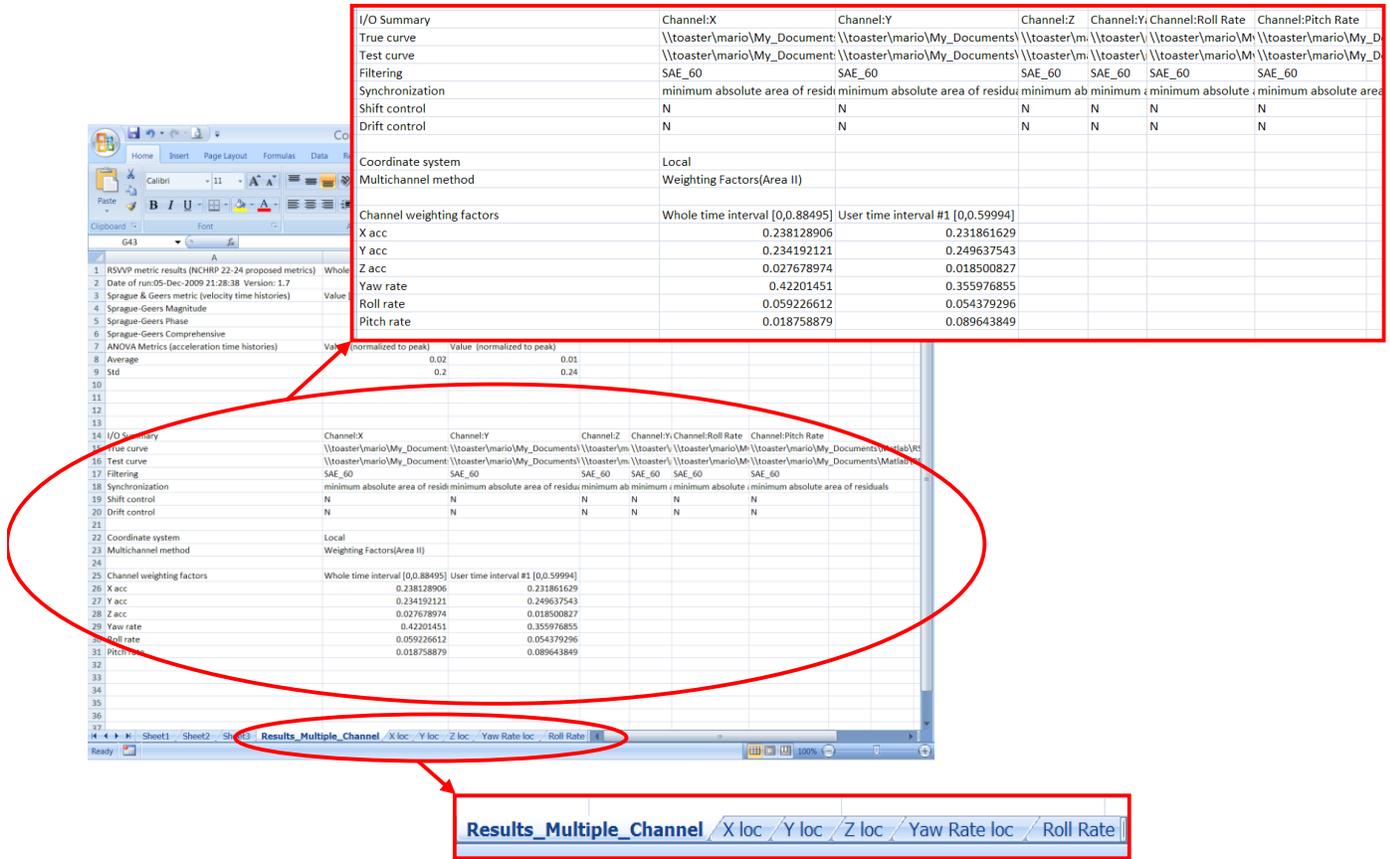


Figure A.27: Summary of preprocessing options and separate sheets for each input channel in the Excel file.

### A.6.2. Graphs

RSVVP creates several graphs during the evaluation of the metrics and saves them as *bitmap* image files. For each time interval considered during the execution of the program, the following graphs are created into the folder */Results/Time-histories/*:

- a) Time histories of the true and test curves,
- b) Time histories of the metrics and
- c) Residuals time histories, histogram and cumulative distribution.

For multichannel input, the time histories of the metrics represent the weighted average of the time histories of the metrics from each channel. Similarly, the residuals time history, histogram and distribution are plotted using the weighted average form the residual histories of each channel.

A specific directory is created for each time interval defined during the run of RSVVP. In order to save disk space, the bitmap files can be compressed in the *.zip* format (see paragraph A.3.3).

Also, the data points of all the time histories are saved in Excel files in order to allow the user to recreate any of the graphs using a spreadsheet editor. A separate file is created for each time interval. The files are named '*Time\_histories\_XXX.xls*', where XXX indicates the specific time interval to which the file is related to. The user can utilize these data to plot the graph of the time histories and histogram of the residuals distribution.

### **A.6.1. Time histories**

The program also saves the data point for the following time histories:

- a) Original input curves
- b) Preprocessed input curves
- c) Metrics

Each of the original and input curves is saved in a separate ASCII file, while the time histories of the metrics are saved in an Excel file format. In the latter case, a separate Excel spreadsheet is used for each of the time interval defined during the run of RSVVP.

## **A.7. EXAMPLES**

This chapter presents two examples of application of RSVVP involving respectively the comparison of a single and multiple pairs of curves. In the first case, RSVVP is used to compare the acceleration time histories obtained from two full-scale crash tests while, in the second example, numerical and experimental results are compared using both the three acceleration and the three rotational rate channels.

### **A.7.1. Single channel comparison**

This example shows how to use RSVVP to compare the acceleration time histories obtained from two full-scale crash tests involving a longitudinal rigid barrier. Both tests were performed using new vehicles (i.e., same model and brand) and the same longitudinal barrier.

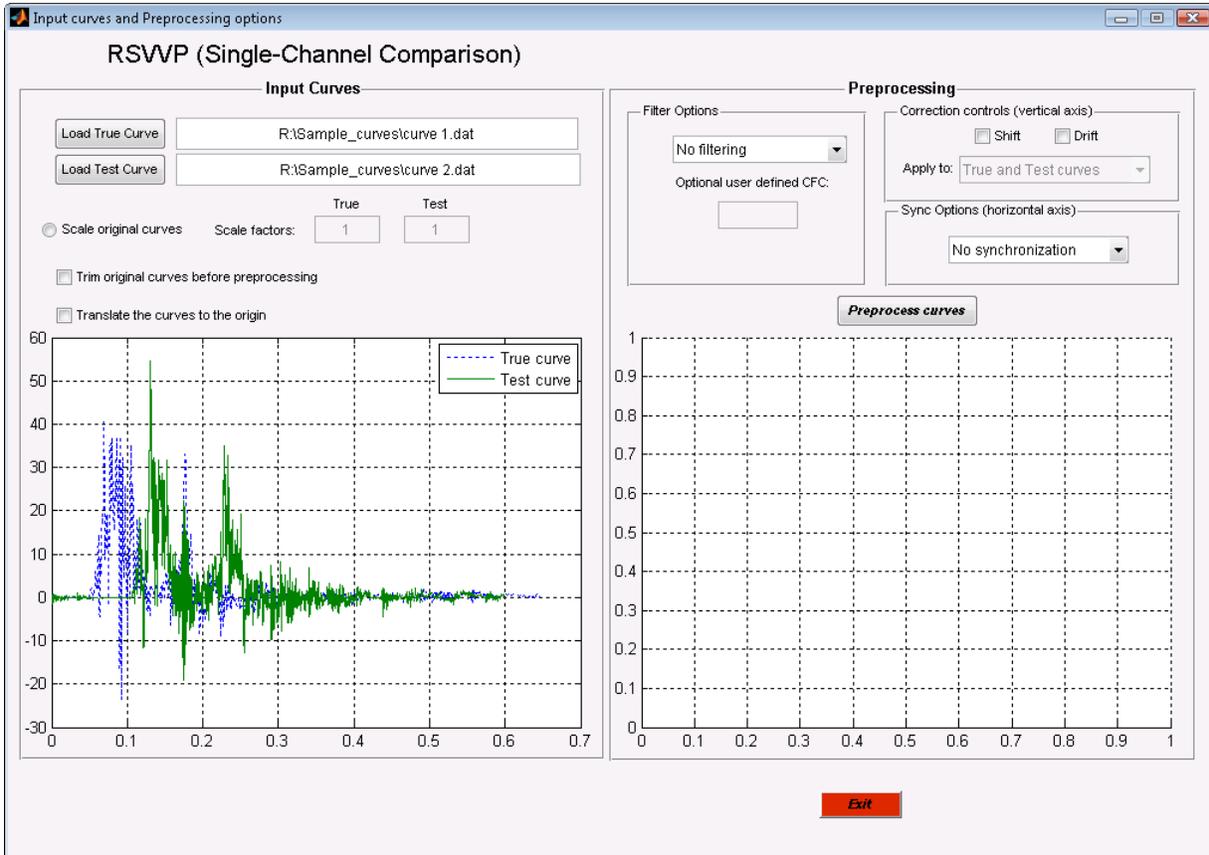
Although conceptually the two crash tests should be identical, in practice, the acceleration curves obtained from each test show some differences. In fact, in such a complex event as a crash test, it is practically impossible to completely control all the parameters involved like the exact angle and velocity of impact, point of impact, behavior of the vehicle's mechanical components, etc.

Before evaluating the comparison metrics, the RSVVP preprocessing options will be used to correctly prepare the data by filtering and synchronizing the original acceleration curves. Next, the two curves will be compared evaluating the comparison metrics of the NCHRP 22-24 profile.

#### **A.7.1.1. Preprocessing of the original curves**

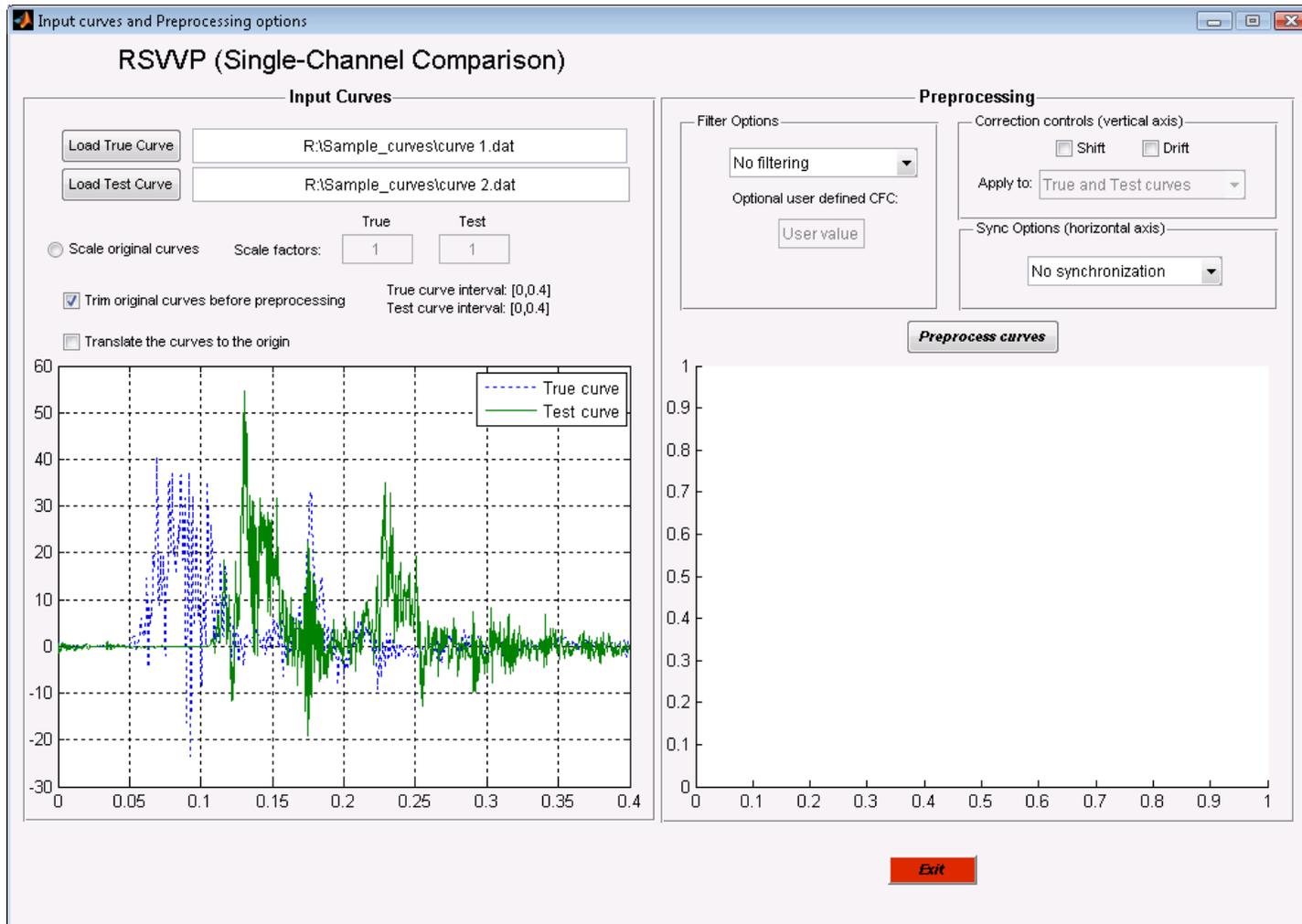
In order to show how each preprocessing option (i.e., filtering and synchronizing) contributes to the improvement of the original input curves, the preprocessing operations are applied incrementally step by step in this example. All the preprocessing operations could have been applied simultaneously however.

*First, the two acceleration time histories (i.e., curve 1 and 2) are loaded into RSVVP considering the curve 1 as the true curve (Figure A.28). As in this case we want to compare only a couple of curves, the 'Single Channel' option has been selected at the beginning of RSVVP.*



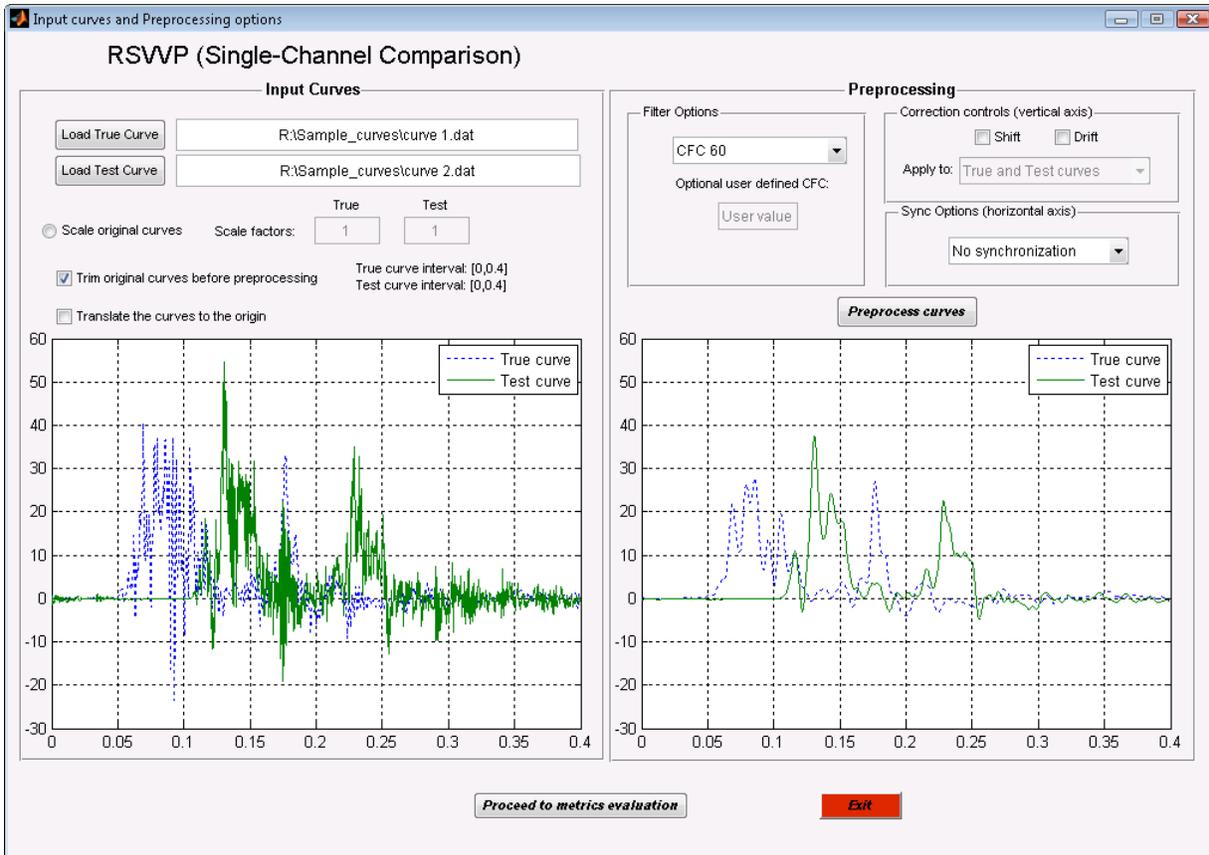
**Figure A.28: Original acceleration time histories loaded into RSVVP.**

As can be seen, the original input curves are characterized by a flat tail after 0.4 sec., they are both manually trimmed using the corresponding option (Figure A.29).



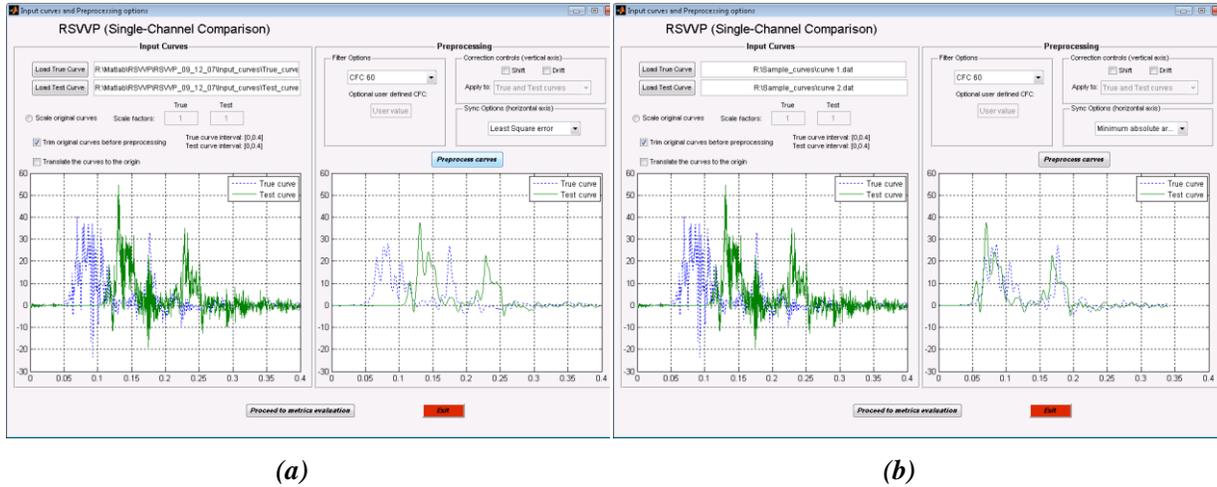
**Figure A.29: Input curves after the manual trimming operation.**

Because the original acceleration time histories are characterized by a certain level of noise and high frequency vibrations, they need to be filtered. In this example a CFC 60 filter is selected (Figure A.30).



**Figure A.30: Original and filtered acceleration time histories.**

From the graph, it can be clearly seen that the two time histories are not synchronized with each other as the initial time at which the accelerations were being recorded was not the same. Initially, a synchronization based on the ‘Least Square Method’ is selected but, as the results are not completely satisfactory, in a second phase the method of the ‘Minimum area of residuals’ is selected. Figure A.31 shows the results obtained using both methods. The synchronization based on the ‘Minimum area of residuals’ gives better visual results in this case.



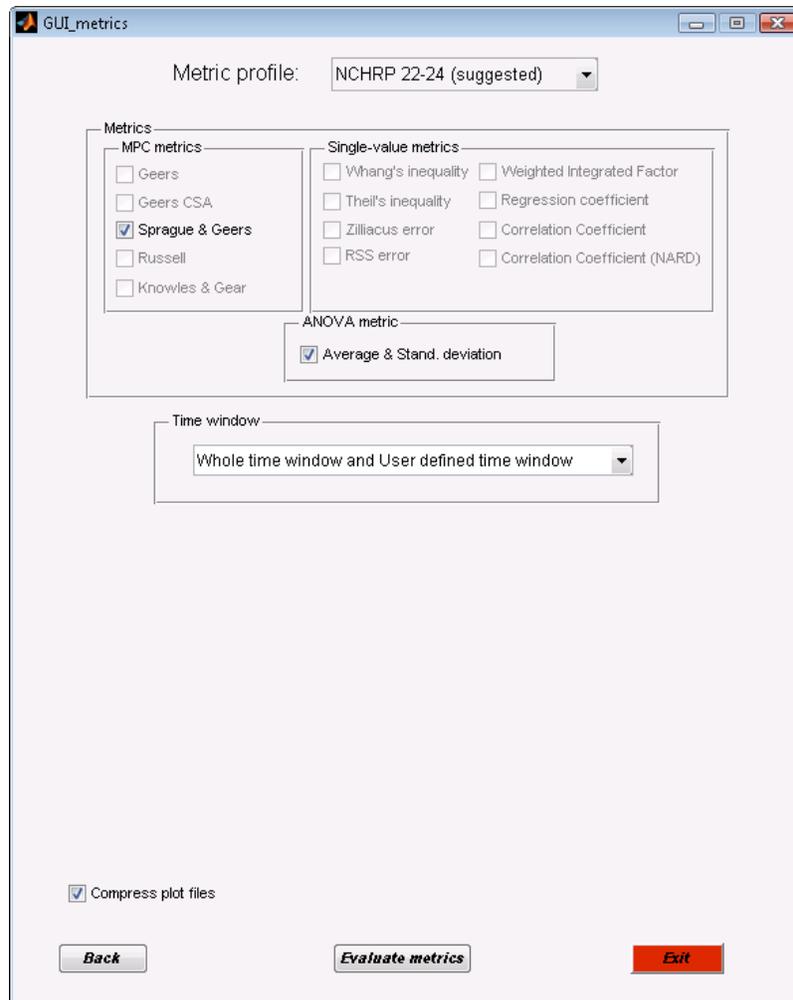
**Figure A.31: Filtered and synchronized time histories based on the (a) Least Square and (b) the Minimum Area of Residuals method.**

RSVVP gives the user a warning message if no filtering and/or synchronization options were selected. After the test and true curves have been preprocessed, we proceed to the next step: the selection of the metrics and time intervals.

#### A.7.1.2. Metric selection and evaluation

In this example, the NCHRP 22-24 metrics profile is selected. This evaluates the ANOVA metrics and the Sprague-Geers MPC metrics. The curves are compared both considering their total length (*Whole time window*) and on some user defined time intervals (*User defined time window*).

The metric evaluation is initiated by pushing the ‘*Evaluate metrics*’ button (Figure A.32).



**Figure A.32: Selection of the metrics profile and time interval.**

During the evaluation of the metrics, various graphs appear and disappear on the computer screen. These graphs are saved as output files by the RSVVP and represent the time histories of metrics and other curves (see chapter 6 for more information about the output files).

When the metrics evaluation is done, the results are shown on a screen like Figure A.33.

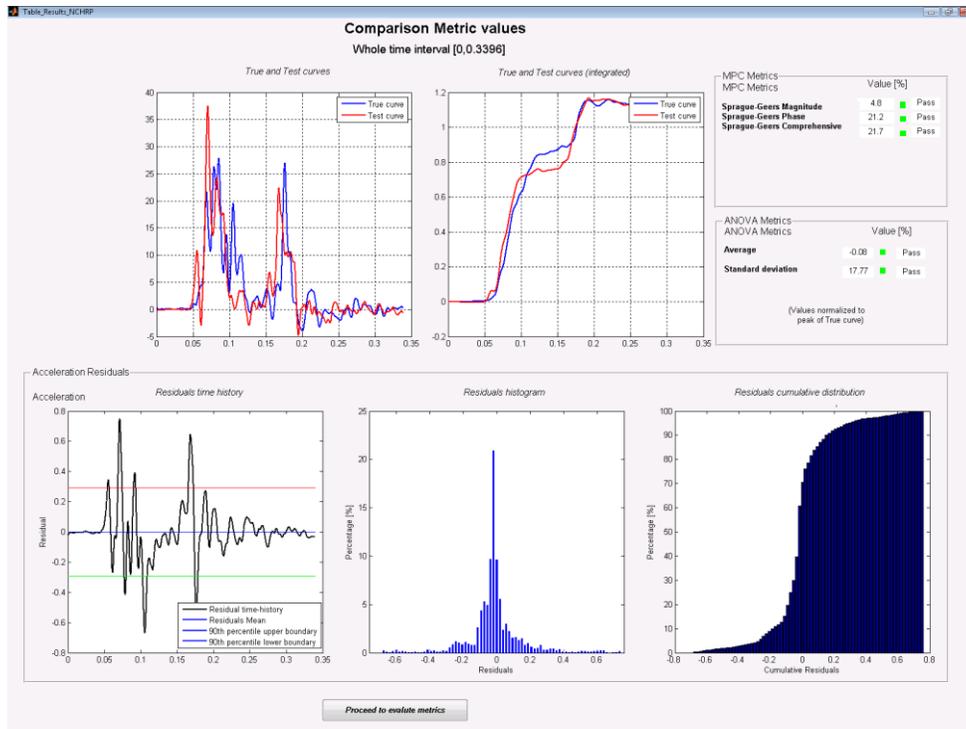
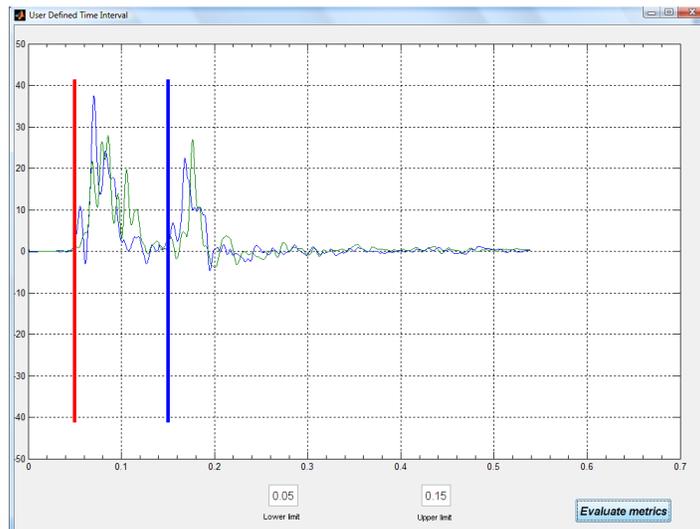


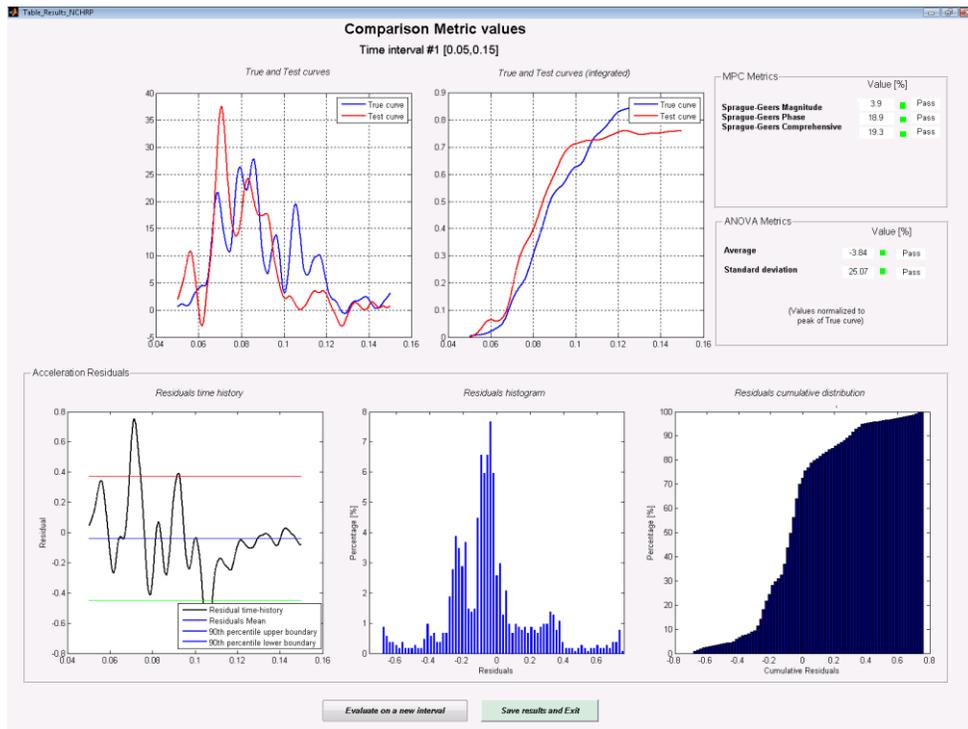
Figure A.33: Output of results for the whole time interval.

By clicking the 'Proceed to evaluate metrics' button, the window shown in Figure A.34 appears. This input screen allows the user to define the upper and lower boundaries for the time interval on which the metrics will be calculated next. The interval selected for this example is: [0.05 sec , 0.15 sec].



**Figure A.34: Setting of the boundaries for the User defined time interval.**

Clicking the ‘*Evaluate metrics*’ button causes RSVVP to evaluate the previously selected metrics but this time using only the user-defined time interval. Figure A.35 shows the metric values obtained for this time interval.



**Figure A.35: Output of results for time interval [0.05 sec , 0.15 sec].**

By clicking the ‘*Evaluate on a new interval*’ button, a new time interval is now defined and the metrics are evaluated again following the same procedure used for the first time interval. In this case, the time interval defined is [0.15 sec , 0.20 sec] (Figure A.36) and gives the results shown in Figure A.37.

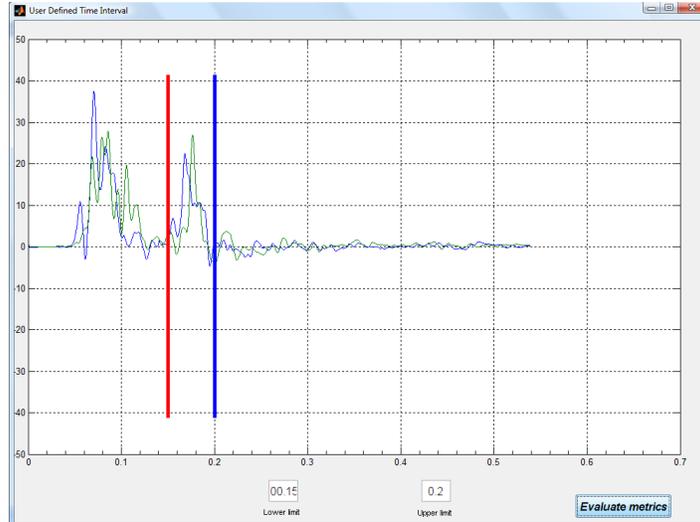


Figure A.36: Time interval [0.15 sec , 0.20 sec].

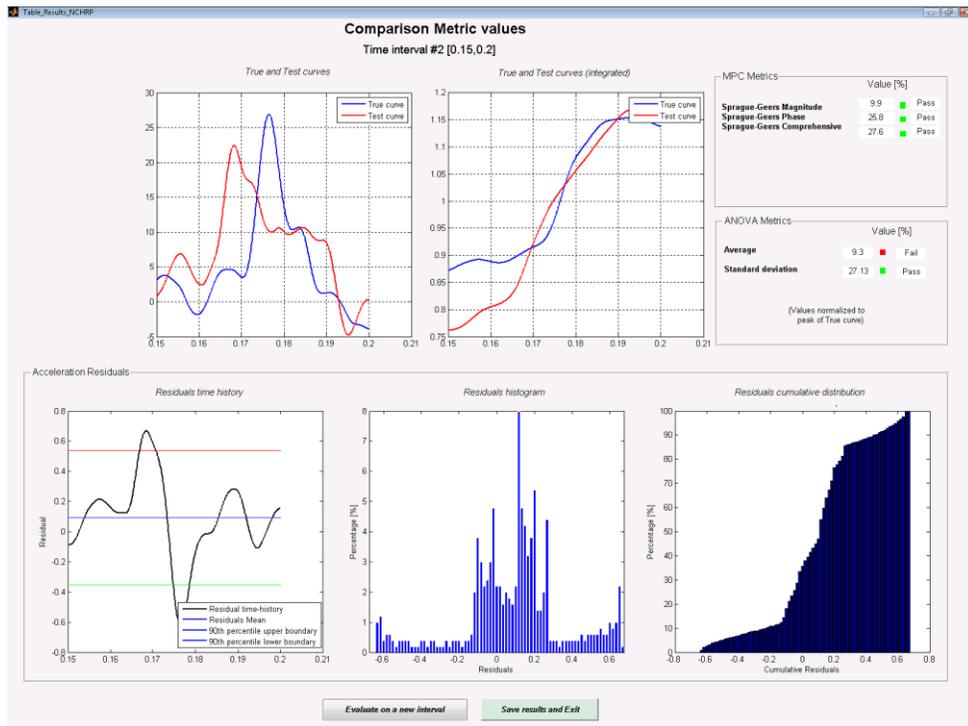


Figure A.37: Output of results for time interval [0.15 sec , 0.20 sec].

### A.7.1.3. Results

Assuming the analysis previously performed on the two time intervals is satisfactorily, it is decided to save all the results and exit RSVVP. After the 'Save results and Exit' button is pressed, RSVVP creates a folder called *\Results\* in the directory where it was installed and a

subfolders for each time interval considered during the metrics analysis. In this example three different subfolders are created:

- Whole\_time\_Interval,
- User\_defined\_interval\_1\_[0.05 , 0.15] and
- User\_defined\_interval\_2\_[0.15005 , 0.19995].

Also, the Excel file *Comparison Metrics.xls* is created, containing a summary of the metrics value for each interval.

Table A.2 summarizes the results obtained comparing the pair of curves on each of the three time intervals (i.e., whole time and two user defined time interval). As can be seen, except for one case, the values of the comparison metrics computed considering the entire time interval ( $[0,0.4]$ ) and each of the two user defined time intervals ( $[0.05,0.15]$  and  $[0.15,0.20]$ ) are within the acceptance criteria suggested by the NCHRP 22 -24 project. Only in the case of the comparison in the time interval  $[0.15,0.2]$  the average of the residuals is larger than the respective proposed criterion, i.e., 9.3 percent against a proposed cut-off value of 5 percent. This can be partially justified by the limited population of the residuals for that small interval.

**Table A.2: Summary of the metrics values for each of the time interval considered.**

	Whole time interval [0,0.3396]	User time interval #1 [0.05,0.15]	User time interval #2 [0.15,0.2]
<b>Sprague &amp; Geers metric</b>	Value [%]	Value [%]	Value [%]
<b>Sprague-Geers Magnitude</b>	4.8	3.9	9.9
<b>Sprague-Geers Phase</b>	21.2	18.9	25.8
<b>Sprague-Geers Comprehensive</b>	21.7	19.3	27.6
<b>ANOVA Metrics</b>	Value [%] (normalized to peak)	Value [%] (normalized to peak)	Value [%] (normalized to peak)
<b>Average</b>	-0.08	-3.84	9.3
<b>Std</b>	17.77	25.07	27.13

### **A.7.1. Multiple channel comparison**

This second example shows an application of RSVVP to perform a validation of a finite element model by comparing the three acceleration and the three rotational rate channels obtained from the numerical simulation with the time histories collected during the corresponding full-scale crash test.

Although the numerical model reproduced in detail the experimental test as for both the characteristics of vehicle and the barrier, the time histories obtained from the numerical simulation are very unlikely to be exactly the same. As a consequence of this scatter in the results between the numerical and experimental tests, it may be difficult to give a subjective assessment of the degree of match of each pair of curves. Moreover, the comparison of multiple channels makes even more difficult the process of determining an overall assessment of the validation level. The quantitative comparison of the time histories achieved using the metrics computed by RSVVP combined with the capability to handle simultaneously multiple channels can greatly help in the process of validating a numerical model.

### **A.7.2. Preprocessing**

As in this case we want to compare multiple channels, the ‘Multiple Channel’ option has been selected at the beginning of RSVVP. Similarly to the previous example, the original time histories are preprocessed before the evaluation of the comparison metrics. As in this case multiple channels are input, the preprocessing is performed separately for each channel. Although in RSVVP it is possible to select the preprocessing options independently for each channel, in this particular case, the same options have been used for all the six input channels. In particular, the following preprocessing operations have been applied:

- Manual trimming (0 – 0.9 sec)
- Filtering (SAE 60)

Figure A.38 shows the input and preprocessed curves for each of the acceleration and rotational rate time histories. Note that, in the multi-channel case, the synchronization is performed in a second step, after all the channels have been input (see Chapter A.2.5 for details). The time histories before and after the synchronization are shown in Figure A.39.

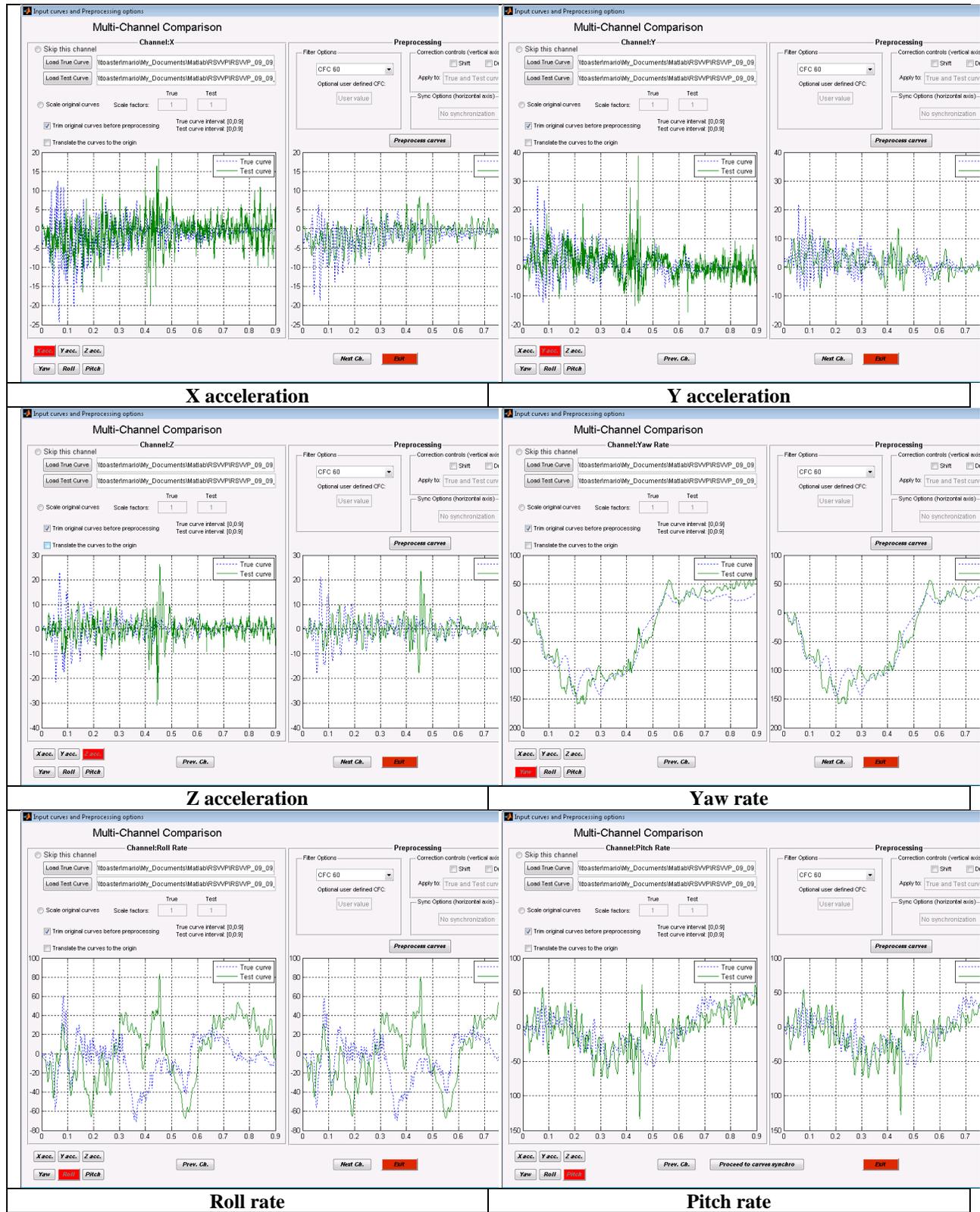


Figure A.38: Original and preprocessed input channels.

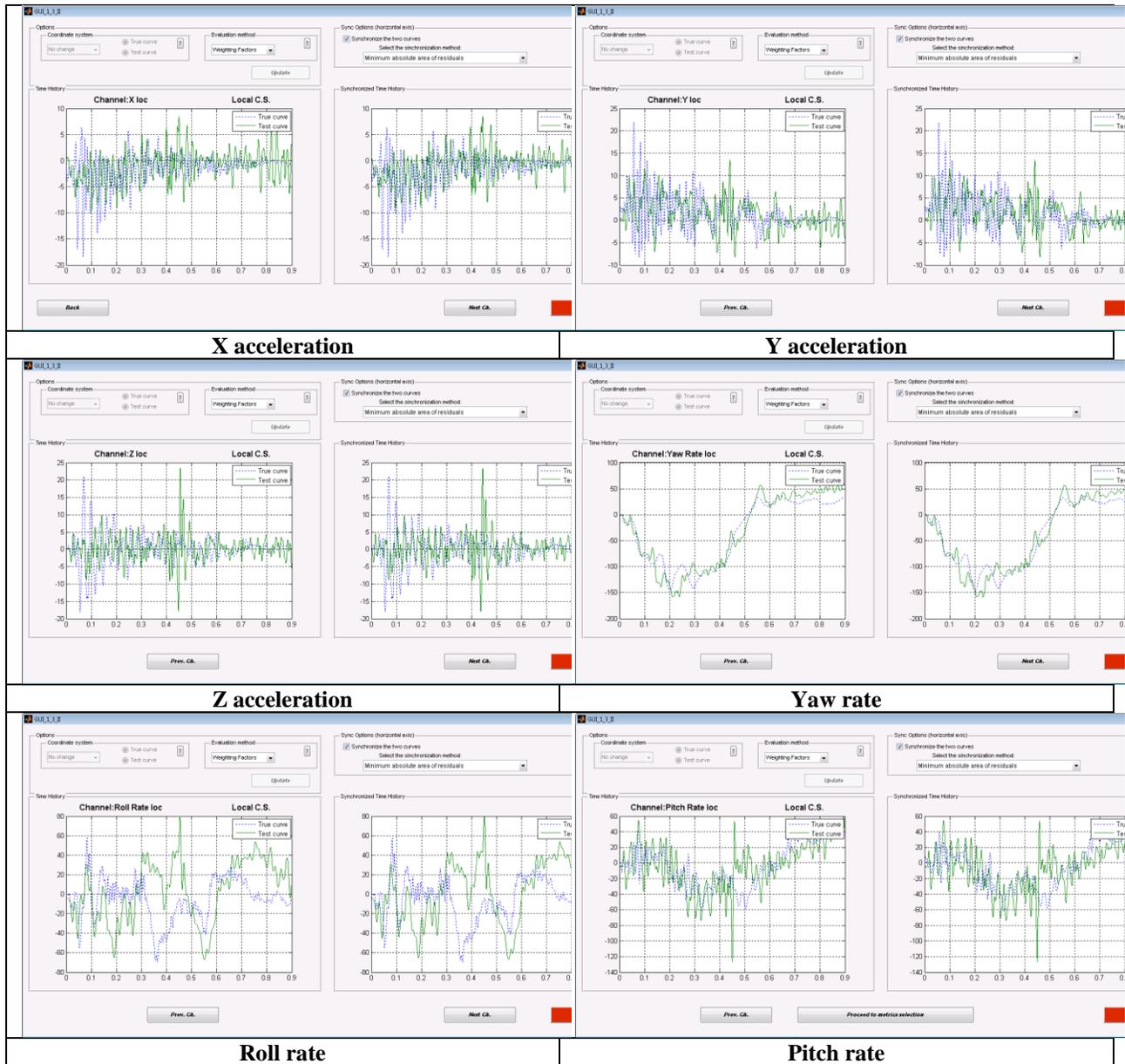


Figure A.39: Input channels before/after synchronization.

### **A.7.2.1. Metric selection and evaluation**

As in the previous example, the NCHRP 22-24 metrics profile is selected (ANOVA metrics and the Sprague&Geers MPC metrics). Because in this case the curves have been already manually trimmed to the time interval representing the actual impact event and we are interested in considering the validation of the model on the entire impact event, the multiple pairs of curves are compared considering their total length after the preprocessing (*Whole time window*).

Because in this case multiple channels have been input, RSVVP will evaluate the value of the metrics for each single channel first and then combine them together by considering a weighted average. The weighting factors are computed based on the area of the true curve of each channel. For more details see section A.2.1 and Appendix B.

After the metrics have been calculated, RSVVP shows the results on the screen. As in this example multiple channels have been input, the user has to select for which channel to display the results. In this particular case, the choice is among any of the six channels (X, Y, Z accelerations and Yaw, Roll, Pitch rates). Also, the weighted average of the results can be shown. Figure A.40 through Figure A.45 illustrate the results obtained for each channel, while Figure A.46 shows the results considering the weighted average.

As usual, RSVVP automatically saves the graphs of the time histories of metrics (see chapter 6 for more information about the output files).

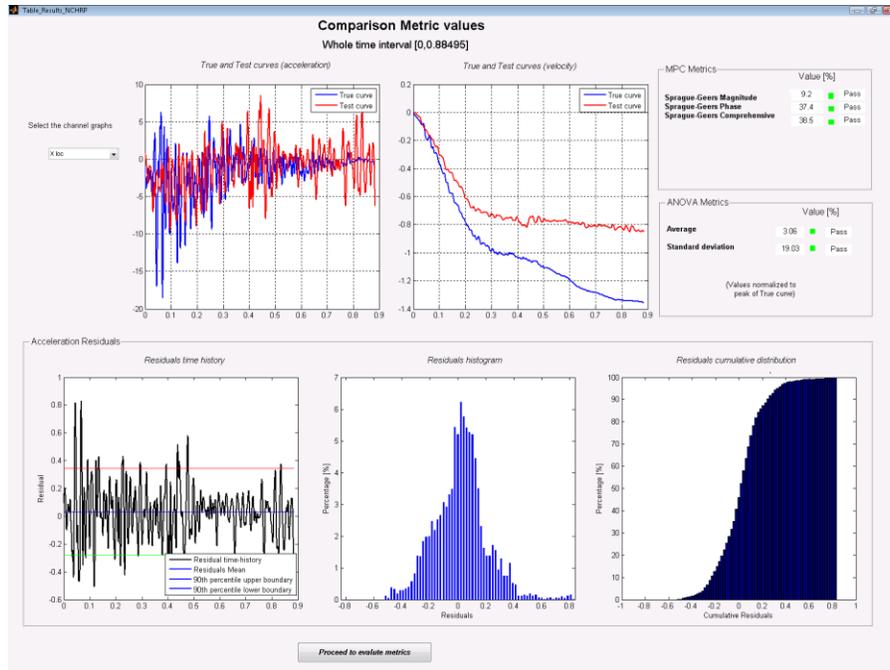


Figure A.40: Screen output of the results for the X channel.

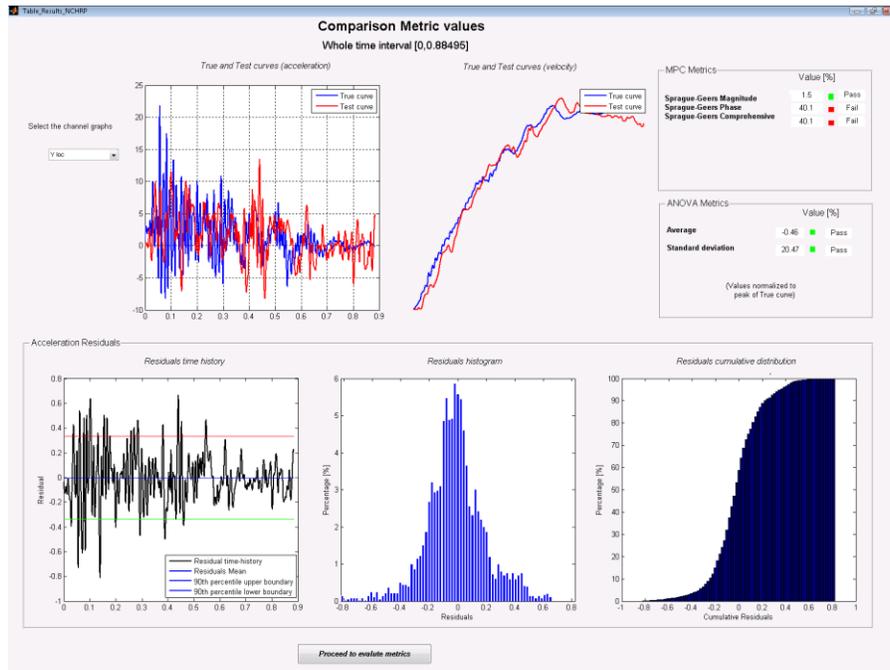


Figure A.41: Screen output of the results for the Y channel.

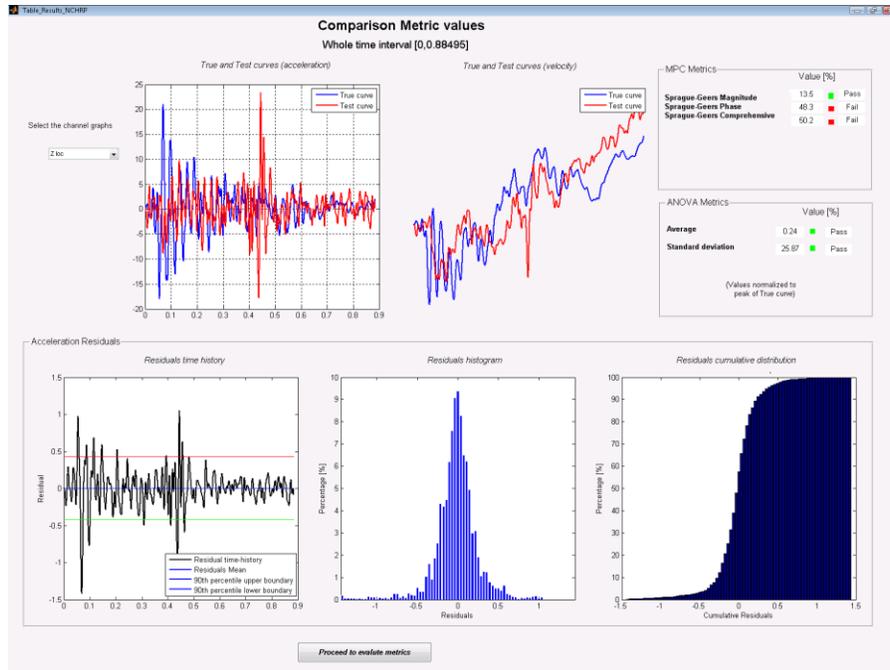


Figure A.42: Screen output of the results for the Z channel.

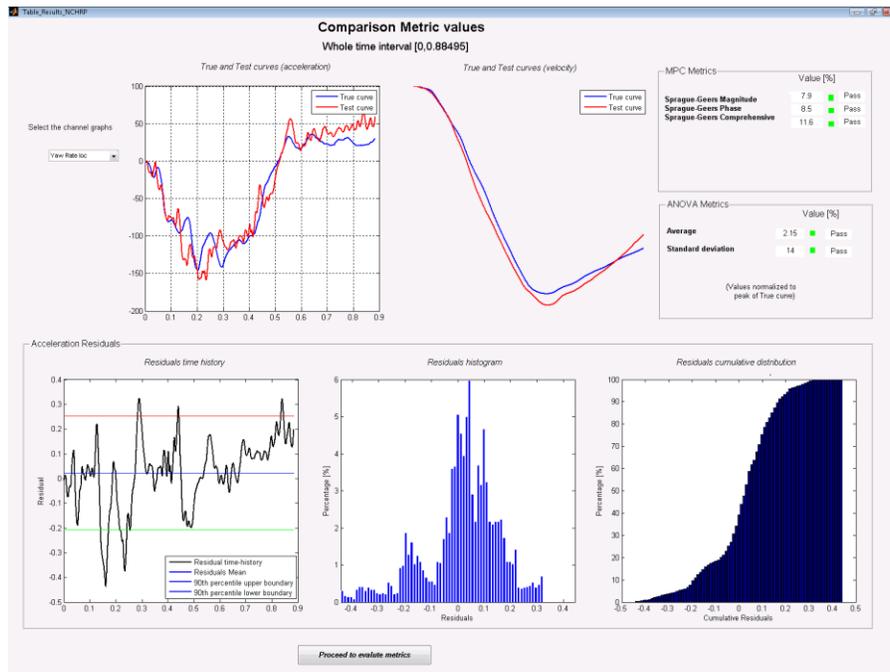


Figure A.43: Screen output of the results for the Yaw channel.

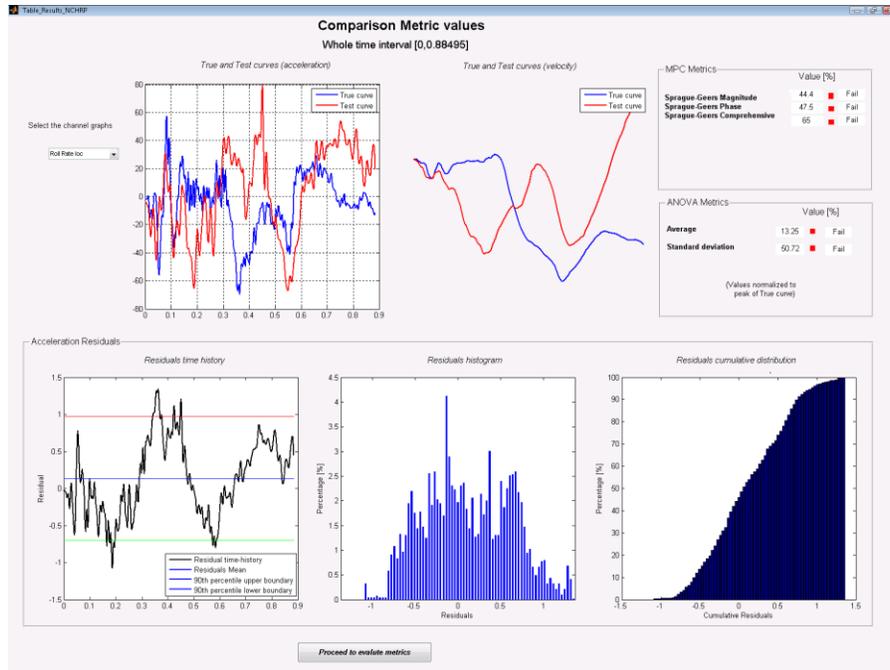


Figure A.44: Screen output of the results for the Roll channel.

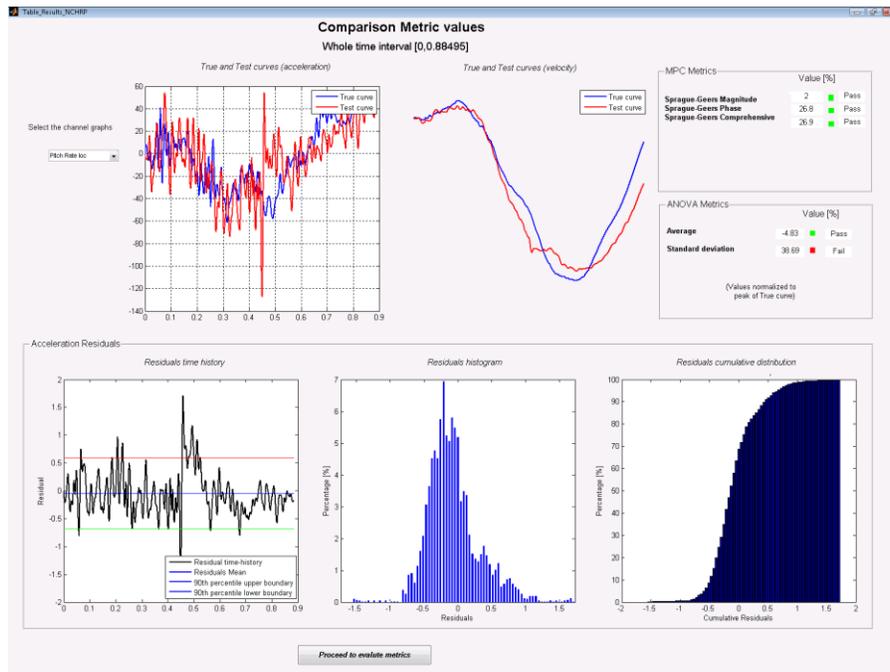
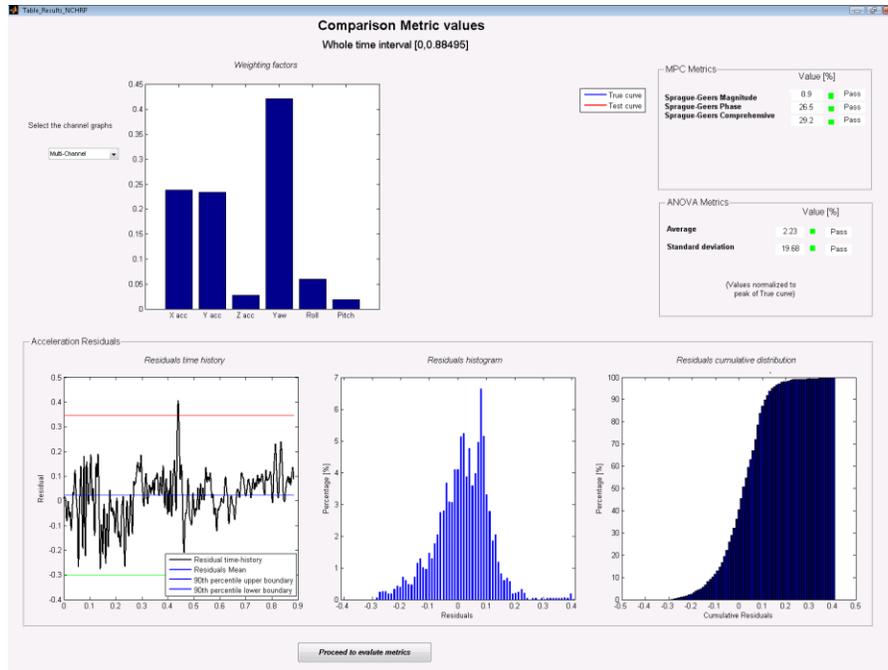


Figure A.45: Screen output of the results for the Pitch channel.



**Figure A.46: Screen output of the results for the weighted average.**

Table A.3 shows a summary of the results obtained from both each single input channel and the weighted average. The values which exceed the acceptance criteria proposed for the verification and validation of numerical models in roadside safety are indicated in red.

As can be seen from Figure A.40 through Figure A.45, except for the X acceleration and the Yaw rate, the metric values computed for each single input channel are not always within the acceptance criteria. In particular, in the case of the Roll channel all the metric values exceed the acceptance criteria. From these comparisons it would be assessed that the numerical model cannot be considered validated as only two out of six channels pass the proposed acceptance criteria. Actually, this analysis would imply that each single channel has the same importance; in fact, this is not what usually happens in most of the cases, where some channels are considered more relevant than others.

As multiple channels have been input, RSVVP automatically assign to each of them a weighting factor based on the area of the true curves (see Chapter A.4 and Appendix A:a)i)(1)(a)(i)Appendix A.II for more details). The results obtained considering the weighted average are shown in the last row of Table A.3. As can be seen, considering each channel

according to its importance, the numerical model now passes the acceptance criteria for both the Sprague&Geers and the ANOVA metrics.

The weighting factors calculated by RSVVP are represented in the histogram plot on the top left corner of Figure A.46.

**Table A.3: Summary of the metrics values for each channel and the weighted average.**

	Sprague-Geers Magnitude	Sprague-Geers Phase	Sprague-Geers Comprehensive	Average	Std
<b>X</b>	9.2	37.4	39	3	19
<b>Y</b>	1.5	40.1	40	0	2
<b>Z</b>	14	48.3	50	0	26
<b>Yaw</b>	7.9	8.5	12	2	14
<b>Roll</b>	44	47.5	65	13	51
<b>Pitch</b>	2	26.8	27	-5	39
<b>Weighted average</b>	8.9	26.5	29	0	2

**REFERENCES**

- [1] M.H. Ray, “Repeatability of Full-Scale Crash Tests and a Criteria for Validating Finite Element Simulations”, *Transportation Research Record*, Vol. 1528, pp. 155-160, (1996).
- [2] W.L. Oberkampf and M.F. Barone, “Measures of Agreement Between Computation and Experiment: Validation Metrics,” *Journal of Computational Physics* Vol. 217, No. 1 (Special issue: Uncertainty quantification in simulation science) pp 5–36, (2006).
- [3] T.L. Geers, “An Objective Error Measure for the Comparison of Calculated and Measured Transient Response Histories”, *The Shock and Vibration Bulletin*, The Shock and Vibration Information Center, Naval Research Laboratory, Washington, D.C., Bulletin 54, Part 2, pp. 99-107, (June 1984).
- [4] *Comparative Shock Analysis (CSA) of Main Propulsion Unit (MPU), Validation and Shock Approval Plan*, SEAWOLF Program: Contract No. N00024-90-C-2901, 9200/SER: 03/039, September 20, 1994.
- [5] M.A. Sprague and T.L. Geers, “Spectral elements and field separation for an acoustic fluid subject to cavitation”, *J Comput. Phys.*, pp. 184:149, Vol. 162, (2003).
- [6] D.M. Russell, “Error Measures for Comparing Transient Data: Part I: Development of a Comprehensive Error Measure”, *Proceedings of the 68th shock and vibration symposium*, pp. 175–184, (2006).
- [7] L.E. Schwer, “Validation Metrics for Response Time Histories: Perspective and Case Studies”, *Engng. with Computers*, Vol. 23, Issue 4, pp. 295–309, (2007).
- [8] C.P. Knowles and C.W. Gear, “Revised validation metric”, unpublished manuscript, 16 June 2004 (revised July 2004).
- [9] J. Cohen, P. Cohen, S.G. West and L.S. Aiken, *Applied multiple regression/correlation analysis for the behavioral sciences*, Hillsdale, NJ: Lawrence Erlbaum, (3rd ed.), 2003.
- [10] S. Basu and A. Haghghi, “*Numerical Analysis of Roadside Design (NARD) vol. III: Validation Procedure Manual*”, Report No. FHWA-RD-88-213, Federal Highway Administration, Virginia, 1988.
- [11] D. Twisk and A. Ritmeijer, “A Software for Demonstrating Validation of Computer Dummy Models Used in the Evaluation of Aircraft Seating Systems”, SAE Paper No. 2007-01-3925, Society of Automotive Engineers, Warrendale, PA, 2007.
- [12] B. Whang, W.E. Gilbert and S. Zilliacus, *Two Visually Meaningful Correlation Measures for Comparing Calculated and Measured Response Histories*, Carderock Division, Naval Surface Warfare Center, Bethesda, Maryland, Survivability, Structures and Materials Directorate, Research and Development Report, CARDEROCKDIV-U-SSM-67-93/15, September, 1993.
- [13] H. Theil, *Economic Forecasts and Policy*, North-Holland Publishing Company, Amsterdam, 1975.
- [14] D.M. Russell, “Error Measures for Comparing Transient Data: Part II: Error Measures Case

Study”, *Proceedings of the 68th shock and vibration symposium*, pp. 185–198, (2006).

## Appendix A.I Comparison metrics evaluated by RSVVP

A brief description of the metrics evaluated by RSVVP is presented in this section. All fourteen metrics available in RSVVP are deterministic shape-comparison metrics. Details about the mathematical formulation of each metric can be found in the cited literature. Conceptually, the metrics evaluated can be classified into three main categories: (i) magnitude-phase-composite (MPC) metrics, (ii) single-value metrics and (iii) analysis of variance (ANOVA) metrics.

### MPC

MPC metrics treat the curve magnitude and phase separately using two different metrics (i.e., M and P, respectively). The M and P metrics are then combined into a single value comprehensive metric, C. The following MPC metrics are included in RSVVP: (a) Geers (original formulation and two variants), (b) Russell and (c) Knowles and Gear. [3-8] Table A.4 shows the analytical definition of each metric. In this and the following sections, the terms  $m_i$  and  $c_i$  refer to the measured and computed quantities respectively with the “i” subscript indicating a specific instant in time.

In all MPC metrics the phase component (P) should be insensitive to magnitude differences but sensitive to differences in phasing or timing between the two time histories. Similarly, the magnitude component (M) should be sensitive to differences in magnitude but relatively insensitive to differences in phase. These characteristics of MPC metrics allow the analyst to identify the aspects of the curves that do not agree. For each component of the MPC metrics, zero indicates that the two curves are identical. Each of the MPC metrics differs slightly in its mathematical formulation. The different variations of the MPC metrics are primarily distinguished in the way the phase metric is computed, how it is scaled with respect to the magnitude metric and how it deals with synchronizing the phase. In particular, the Sprague and Geers metric [5] uses the same phase component as the Russell metric [6]. Also, the magnitude component of the Russell metric is peculiar as it is based on a base-10 logarithm and it is the only MPC metric that is symmetric (i.e., the order of the two curves is irrelevant). The Knowles and Gear metric [7,8] is the most recent variation of MPC-type metrics. Unlike the previously discussed MPC metrics, it is based on a point-to-point comparison. In fact, this metric requires

that the two compared curves are first synchronized in time based on the so called Time of Arrival (TOA), which represents the time at which a curve reaches a certain percentage of the peak value. In RSVVP the percentage of the peak value used to evaluate the TOA was 5%, which is the typical value found in literature. Once the curves have been synchronized using the TOA, it is possible to evaluate the magnitude metric. Also, in order to avoid creating a gap between time histories characterized by a large magnitude and those characterized by a smaller one, the magnitude component  $M$  has to be normalized using the normalization factor  $QS$ .

**Table A.4: Definition of MPC metrics.**

	<i>Magnitude</i>	<i>Phase</i>	<i>Comprehensive</i>
<b>Integral comparison metrics</b>			
<b>Geers</b>	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2}} - 1$	$P_G = 1 - \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_G^2 + P_G^2}$
<b>Geers CSA</b>	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2}} - 1$	$P_{CSA} = 1 - \frac{ \sum c_i m_i }{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\text{sign}\left(\sum c_i m_i\right) \sqrt{M_{CSA}^2 + P_{CSA}^2}$
<b>Sprague &amp; Geers</b>	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2}} - 1$	$P_{SG} = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_{SG}^2 + P_{SG}^2}$
<b>Russell</b>	$M_R = \text{sign}(m) \cdot \text{Log}_{10}(1 +  m )$ where $m = \frac{(\sum c_i^2 - \sum m_i^2)}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$P_R = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{\frac{\pi}{4} (M_R^2 + P_R^2)}$
<b>Point-to-point comparison metrics</b>			
<b>Knowles &amp; Gear</b>	$M_{KG} = \sqrt{\frac{\sum \left(\frac{ m_i }{m_{max}}\right)^p (\tilde{c}_i - m_i)^2}{\sum \left(\frac{ m_i }{m_{max}}\right)^p (m_i)^2}}$ where $\tilde{c} = c(t - \tau)$ (with $\tau = TOA_c - TOA_m$ )	$P_{KG} = \frac{ TOA_c - TOA_m }{TOA_m}$	$\sqrt{\frac{10M_{KG}^2 + 2P_{KG}^2}{12}}$

### Single-value metrics

Single-value metrics give a single numerical value that represents the agreement between the two curves. Seven single-value metrics were considered in this work: (1) correlation coefficient metric, (2) NARD correlation coefficient metric (NARD), (3) Weighted Integrated Factor, (4) Zilliacus error metric, (5) RSS error metric, (6) Theil's inequality metric, (7) Whang's inequality metric and (8) regression coefficient metric. [9-13] The first two metrics are based on integral comparisons while the others are based on a point-to-point comparisons. The definition of each metric is shown in Table A.5.

**Table A.5: Definition of single-value metrics.**

<i>Integral comparison metrics</i>			
<i>Correlation Coefficient</i>	$\frac{n \sum c_i m_i - \sum c_i \sum m_i}{\sqrt{n \sum c_i^2 - (\sum c_i)^2} \sqrt{n \sum m_i^2 - (\sum m_i)^2}}$	<i>Correlation Coefficient (NARD)</i>	$\frac{\sum c_i m_i}{\sqrt{\sum c_i^2} \sqrt{\sum m_i^2}}$
<i>Weighted Integrated Factor</i>		$\sqrt{\frac{\sum \max(m_i^2, c_i^2) \cdot \left(1 - \frac{\max(0, m_i \cdot c_i)}{\max(m_i^2, c_i^2)}\right)^2}{\sum \max(m_i^2, c_i^2)}}$	
<i>Point-to-point comparison metrics</i>			
<i>Zilliacus error</i>	$\frac{\sum  c_i - m_i }{\sum  m_i }$	<i>RSS error</i>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum m_i^2}}$
<i>Theil's inequality</i>	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum c_i^2} + \sqrt{\sum m_i^2}}$	<i>Whang's inequality</i>	$\frac{\sum  c_i - m_i }{\sum  c_i  + \sum  m_i }$
<i>Regression coefficient</i>	$\sqrt{1 - \frac{(n-1) \sum (c_i - m_i)^2}{n \sum (m_i - \bar{m})^2}}$		

### ANOVA metric

ANOVA metrics are based on the assumption that two curves do, in fact, represent the same event such that any differences between the curves must be attributable only to random experimental error. The analysis of variance (i.e., ANOVA) is a standard statistical test that assesses whether the variance between two curves can be attributed to random error.[1,2] When two time histories represent the same physical event, both should be identical such that the mean

residual error,  $\bar{e}$ , and the standard deviation of the residual errors,  $\sigma$ , are both zero. Of course, this is never the case in practical situations (e.g., experimental errors cause small variations between tested responses even in identical tests). Ray proposed a method where the residual error and its standard deviation are normalized with respect to the peak value of the true curve and came to the following acceptance criteria based on six repeated frontal full-scale crash tests [1]:

- The average residual error normalized by the peak response (i.e.,  $\bar{e}^r$ ) should be less than five percent.

$$\bar{e}^r = \frac{\sum (c_i - m_i) / m_{\max}}{n} < 0.05 \cdot m_{\max}$$

- The standard deviation of the normalized residuals (i.e.,  $\sigma^r$ ) should be less than 35 percent.

$$\sigma^r = \sqrt{\frac{\sum (e_i^r - \bar{e}^r)^2}{n-1}} < 0.35 \cdot m_{\max}$$

## Appendix A.II Weighting factors

This appendix presents a description of how the weighting factors used to combine together the metric values from different channels are evaluated.

In the ‘Multichannel’ option, RSVVP combines the value from the different channels into a single weighted metric value. The concept at the basis of the weighting scheme is to calculate a local index for each channel which could be representative of the importance (or weight) of the specific channel respect to the others. Once these indexes have been computed, the weighting factors are calculated by simply dividing the index from each channel by their total sum. In this way, the summation of the obtained weighting factors will be the unity.

For both the accelerations and rotational rates, the areas are evaluated considering the True curve of the respective channels. The areas of the accelerations and rotational rates are considered separately because otherwise the different units may lead to a disproportion in the evaluation of the weighting factors. The weighting factors for each channel are calculated using the following scheme:

- Evaluation of the area of the True curve for each acceleration channel,  $a_i$ , and rotational channel,  $v_i$ .
- Evaluation of the sum of the acceleration areas,  $a_{Sum}$ , and rotational areas,  $v_{Sum}$ .
- Evaluation of the local weight of each acceleration channel,  $lw_i^{(a)} = \frac{a_i}{a_{Sum}}$ , and rotational

$$\text{channel } lw_i^{(v)} = \frac{v_i}{v_{Sum}}$$

- Evaluation of the channel weight factors,

$$w_i^{(a)} = \frac{lw_i^{(a)}}{\sum lw_i^{(a)} + \sum lw_i^{(v)}}$$

$$w_i^{(v)} = \frac{lw_i^{(v)}}{\sum lw_i^{(a)} + \sum lw_i^{(v)}}$$

Once the weighting factors have been evaluated, the time histories of each metric are combined together using a weighted average. Note that the combination of the time histories is performed for each of the metrics selected at the beginning of the run by the user.

## Appendix A.III NCHRP 22-24 acceptance criteria

Table A.6 shows the acceptance criteria proposed for the verification and validation of finite element models in roadside safety using the NCHRP 22-24 metrics profile.

**Table A.6: Acceptance criteria suggested for the NCHRP 22-24 metrics profile.**

<i>Sprague &amp; Geers metrics</i>		<i>ANOVA metrics</i>	
<i>M</i>	$\leq 40$	<i>Mean</i>	$\leq 0.05$
<i>P</i>	$\leq 40$	<i>Standard deviation</i>	$\leq 0.35$
<i>C</i>	$\leq 40$		

Apart from the value of the comparison metrics, it is important that the cumulative distribution and histogram of the residuals have the following typical characteristics of a normal distribution:

- The histogram should have a normal or bell shaped distribution and the
- Cumulative distribution should have an “S” shape

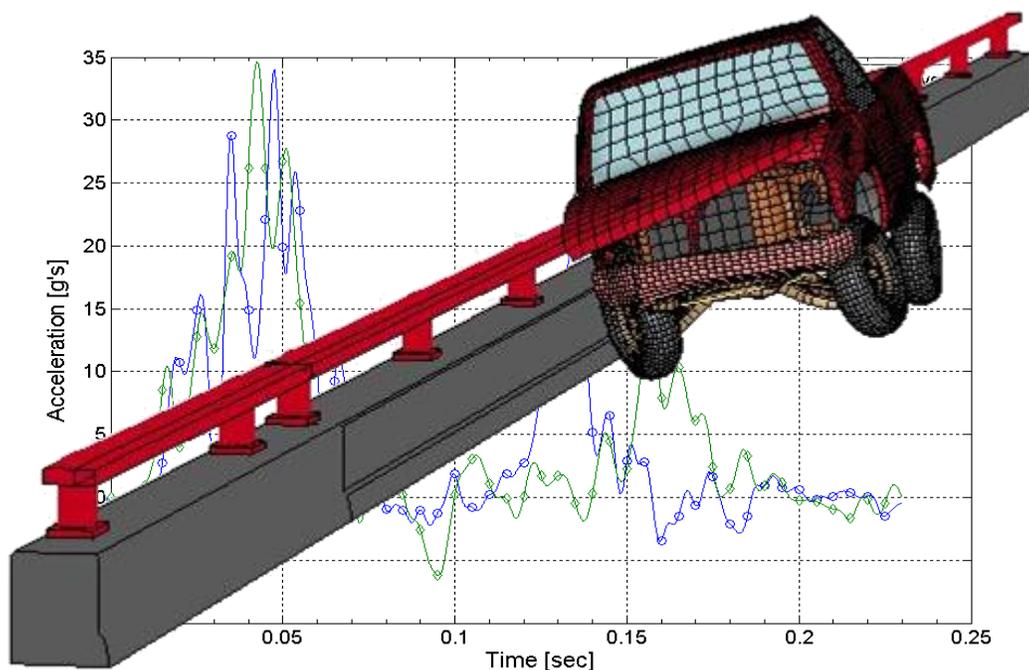
If the histogram and the cumulative distribution do have these shape characteristics, the residuals between the two curves are most likely due to some systematic error which should be identified and corrected.

## **Appendix B**

This appendix contains the programmer's manual for the computer program developed in this study.

# *Roadside Safety Verification and Validation Program (RSVVP)*

## **Programmer's Manual**



Worcester Polytechnic Institute (WPI)

January 2010 (Rev. 1.0)

Mario Mongiardini

Malcolm H. Ray

## Table of Contents

<b>TABLE OF CONTENTS.....</b>	<b>- 308 -</b>
<b>LIST OF FIGURES.....</b>	<b>- 310 -</b>
<b>LIST OF TABLES .....</b>	<b>- 312 -</b>
<b>B.1 PROGRAMMER'S GUIDE.....</b>	<b>- 313 -</b>
<b>B.2 INTRODUCTION.....</b>	<b>- 313 -</b>
<b>B.3 DESCRIPTION OF TASKS.....</b>	<b>- 315 -</b>
B.3.1 INPUT OF DATA .....	- 315 -
B.3.2 PREPROCESSING .....	- 315 -
<i>B.3.2.1 Scaling .....</i>	<i>- 316 -</i>
<i>B.3.2.2 Manual trimming and/or shift of the origin.....</i>	<i>- 316 -</i>
<i>B.3.2.3 Filtering .....</i>	<i>- 316 -</i>
<i>B.3.2.4 Shift/drift correction.....</i>	<i>- 318 -</i>
<i>B.3.2.5 Resampling and Trimming .....</i>	<i>- 319 -</i>
<i>B.3.2.6 Synchronization.....</i>	<i>- 319 -</i>
B.3.3 METRICS EVALUATION .....	- 320 -
B.3.4 POSTPROCESSING .....	- 322 -
B.3.5 OUTPUT OF RESULTS .....	- 323 -
<b>B.4 PROGRAM STRUCTURE .....</b>	<b>- 324 -</b>
B.4.1 ORGANIZATION OF THE CHAPTER .....	- 327 -
<i>B.4.1.1 Notation used for the flow charts .....</i>	<i>- 327 -</i>
B.4.2 GRAPHICAL USER INTERFACES.....	- 328 -
B.4.3 BLOCK A (INITIALIZATION).....	- 331 -
<i>B.4.3.1 Opening (A.1) .....</i>	<i>- 332 -</i>
<i>B.4.3.2 Initialization (A.2).....</i>	<i>- 333 -</i>
B.4.3.2.1 Load_configuration .....	- 334 -
B.4.3.2.2 Load_preprocess_initialization .....	- 334 -
B.4.4 BLOCK B (INPUT AND OPTION SELECTION).....	- 336 -
<i>B.4.4.1 Input/Preprocessing (B.1).....</i>	<i>- 339 -</i>
B.4.4.1.1 Load_Preprocess.....	- 339 -
B.4.4.1.2 Load_curves .....	- 344 -

B.4.4.2	<i>Preprocessing 2 (B.2)</i> .....	- 344 -
B.4.4.2.1	Preprocessing_2.....	- 345 -
B.4.4.3	<i>Metrics selection (B.3)</i> .....	- 348 -
B.4.4.3.1	Metrics_selection.....	- 349 -
B.4.5	BLOCK C (CURVE PREPARATION).....	- 350 -
B.4.5.1	<i>Curves preparation (C.1)</i> .....	- 351 -
1.1.1.	<i>Curves histories (C.2)</i> .....	- 353 -
B.4.5.2	<i>Curves plotting (C.3)</i> .....	- 353 -
B.4.6	BLOCK D (METRICS EVALUATION).....	- 355 -
B.4.6.1	<i>Whole time (D.1)</i> .....	- 356 -
B.4.6.1.1	Weighting_scheme_whole .....	- 357 -
B.4.6.1.2	Whole_time_evaluation .....	- 358 -
B.4.6.1.3	Whole_time_postprocessing .....	- 360 -
B.4.6.1.4	Table_output_whole.....	- 362 -
B.4.6.2	<i>User time (D.2)</i> .....	- 362 -
B.4.6.2.1	Store Results .....	- 365 -
B.4.7	BLOCK E (OUTPUT).....	- 366 -
B.4.7.1	<i>Configuration file (E.1)</i> .....	- 368 -
B.4.7.2	<i>Excel results (E.2)</i> .....	- 368 -
B.4.7.2.1	Excel_results .....	- 369 -
B.4.7.2.2	Excel_results_MPC.....	- 369 -
B.4.7.2.3	Excel_time_histories.....	- 373 -
B.4.7.3	<i>Folder selection (E.3)</i> .....	- 375 -
<b>REFERENCES</b> .....		<b>- 376 -</b>

## List of Figures

FIGURE B.1: REPRESENTATION OF (A) SHIFT AND (B) DRIFT EFFECTS. ....	- 318 -
FIGURE B.2: SKETCH OF THE BEHAVIOR OF THE SHIFT SUBROUTINE FOR A (A) POSITIVE OR (B) NEGATIVE INPUT S. ....	- 320 -
FIGURE B.3: DIAGRAM OF THE FIVE MAIN BLOCKS OF THE RSVVP CODE. ....	- 325 -
FIGURE B.4: DIAGRAM OF THE TASKS PERFORMED BY RSVVP. ....	- 326 -
FIGURE B.5: SYMBOLS USED FOR THE FLOWCHARTS IN THIS MANUAL. ....	- 328 -
FIGURE B.6: STRUCTURE OF A MATLAB GUI. ....	- 329 -
FIGURE B.7: STRUCTURE OF THE VARIABLE <i>HANDLES</i> . ....	- 330 -
FIGURE B.8: SCHEMATIC REPRESENTATION OF THE WORKSPACE OF THE MAIN AND OBJECTIVE FUNCTIONS OF A GUI. ....	- 331 -
FIGURE B.9: DIAGRAM OF BLOCK A. ....	- 332 -
FIGURE B.10: FIELDS OF THE VARIABLE <i>SELECTION</i> . ....	- 332 -
FIGURE B.11: FLOW CHART OF THE ALGORITHM OF SUB-BLOCK A.2 (INITIALIZATION). ....	- 333 -
FIGURE B.12: FLOW CHART OF THE SCRIPTS <i>LOAD_CONFIGURATION</i> (LEFT) AND <i>LOAD_PREPROCESS</i> (RIGHT) (SUB-BLOCK A.2). ....	- 335 -
FIGURE B.13: DIAGRAM OF BLOCK B. ....	- 336 -
FIGURE B.14: SCHEME OF THE LOOP WHICH FORMS BLOCK B. ....	- 337 -
FIGURE B.15: FLOW CHART OF THE MAIN ALGORITHM OF BLOCK B. ....	- 338 -
FIGURE B.16: FLOW CHART OF THE ALGORITHM OF SUB-BLOCK B.1 (INPUT/PREPROCESSING). ....	- 339 -
FIGURE B.17: FLOW CHART OF THE SCRIPT <i>LOAD_PREPROCESS</i> (SUB-BLOCK B.1) – PART A. ....	- 342 -
FIGURE B.18: FLOW CHART OF THE SCRIPT <i>LOAD_PREPROCESS</i> (SUB-BLOCK B.1) – PART B. ....	- 343 -
FIGURE B.19: SKETCH OF THE STRUCTURE OF THE VARIABLE <i>PREPROCESSED</i> . ....	- 344 -
FIGURE B.20: FLOW CHART OF THE SCRIPT <i>LOAD_CURVES</i> (SUB-BLOCK B.1). ....	- 344 -
FIGURE B.21: DIAGRAM OF SUB-BLOCK B.2 (PREPROCESSING2). ....	- 345 -
FIGURE B.22: FLOW CHART OF THE SCRIPT <i>PREPROCESSING_2</i> (SUB-BLOCK B.2). ....	- 347 -
FIGURE B.23: DIAGRAM OF SUB-BLOCK B.3 (METRICS SELECTION). ....	- 349 -
FIGURE B.24: FLOW CHART OF THE SCRIPT <i>METRICS_SELECTION</i> (SUB-BLOCK B.3). ....	- 350 -
FIGURE B.25: DIAGRAM OF BLOCK C. ....	- 351 -
FIGURE B.26: DATA ORGANIZATION OF THE MATRIX VARIABLES <i>TRUE</i> AND <i>TEST</i> . ....	- 352 -
FIGURE B.27: DIAGRAM OF SUB-BLOCK C.1 (CURVES PREPARATION). ....	- 352 -
FIGURE B.28: DIAGRAM OF SUB-BLOCK C.2 (CURVES HISTORIES) (CENTER) AND THE TWO INVOKED SCRIPTS, <i>SAVE_CURVES_ORIGINAL</i> (LEFT) AND <i>SAVE_CURVES_PREPROCESSED</i> (RIGHT). ....	- 353 -
FIGURE B.29: FLOW CHART OF THE SCRIPT <i>WHOLE_PLOT_CURVES</i> (SUB-BLOCK C.3). ....	- 354 -
FIGURE B.30: DIAGRAM OF BLOCK D. ....	- 355 -
FIGURE B.31: MAIN STRUCTURE BLOCK D. ....	- 356 -
FIGURE B.32: DIAGRAM OF SUB-BLOCK D.1 (WHOLE TIME). ....	- 357 -

FIGURE B.33: FLOW CHART OF THE SCRIPT *WEIGHTING\_SCHEME\_WHOLE* (SUB-BLOCK D.1). ..... - 358 -

FIGURE B.34: FLOW CHART OF THE SCRIPTS *WHOLE\_TIME\_EVALUATION* (LEFT) AND *WHOLE\_TIME\_POSTPROCESSING* (RIGHT) (SUB-BLOCK D.1)..... - 359 -

FIGURE B.35: DATA ORGANIZATION OF THE VARIABLES *OUTPUT\_XLS* AND *OUTPUT\_CHANNEL\_XLS*. ..... - 361 -

FIGURE B.36: DATA ORGANIZATION OF THE VARIABLES *OUTPUT\_SINGLE\_HISTORY\_XLS* AND *OUTPUT\_CHANNEL\_HISTORY\_XLS*. ..... - 362 -

FIGURE B.37: DIAGRAM OF SUB-BLOCK D.2 (USER TIME). ..... - 364 -

FIGURE B.38: DATA ORGANIZATION OF THE VARIABLES *OUTPUT\_XLS* AND *OUTPUT\_CHANNEL\_XLS*. ..... - 366 -

FIGURE B.39: DIAGRAM OF BLOCK E. .... - 367 -

FIGURE B.40: MAIN STRUCTURE BLOCK E. .... - 367 -

FIGURE B.41: DIAGRAM OF SUB-BLOCK E.1 (CONFIGURATION FILE)..... - 368 -

FIGURE B.42: DIAGRAM OF SUB-BLOCK E.2 (EXCEL RESULTS). ..... - 369 -

FIGURE B.43: FLOW CHART OF THE SCRIPT *EXCEL\_RESULTS* (SUB-BLOCK E.2). ..... - 371 -

FIGURE B.44: FLOW CHART OF THE SCRIPT *EXCEL\_RESULTS\_MPC* (RECALLED BY THE SCRIPT *EXCEL\_RESULTS*). ..... - 372 -

FIGURE B.45: DATA EXTRACTION FROM THE VARIABLE *OUTPUT\_CHANNEL\_XLS*. ..... - 373 -

FIGURE B.46: FLOW CHART OF THE SCRIPT *EXCEL\_TIME\_HISTORIES* (SUB-BLOCK E.2)..... - 374 -

FIGURE B.47: DIAGRAM OF SUB-BLOCK E.3 (FOLDER SELECTION)..... - 375 -

FIGURE B.48: IDEALIZED TIME HISTORIES USED FOR THE CASE OF (A) 20% MAGNITUDE AND (B) TIME OF ARRIVAL ..... - 379 -

FIGURE B.49: SPRAGUE-GEERS COMPONENT METRIC VS. TIME FOR THE MAGNITUDE-DIFFERENCE TEST ..... - 380 -

FIGURE B.50: SPRAGUE-GEERS COMPONENT METRIC VS. TIME FOR THE PHASE-DIFFERENCE TESTS: (A) +20% AND (B)-20%. ..... - 380 -

FIGURE B.51: EXAMPLE OF A MATLAB STRUCTURE VARIABLE [1]..... - 382 -

FIGURE B.52: EXAMPLE OF A MATLAB STRUCTURE VARIABLE [1]..... - 383 -

FIGURE B.53: ALGORITHM OF THE SAE FILTERING. .... - 385 -

FIGURE B.54: MAIN ALGORITHM OF THE SCRIPT *SHIFT\_DRIFT*. ..... - 386 -

FIGURE B.55: ALGORITHM OF THE USER-DEFINED FUNCTIONS *SHIFT\_VALUE* (LEFT) AND *DRIFT\_VALUE* (RIGHT)..... - 387 -

FIGURE B.56: ALGORITHM OF THE SCRIPT *RESAMPLING\_TRIMMING*..... - 388 -

FIGURE B.57: ALGORITHM OF THE SCRIPT *CURVE\_SYNCHRONIZING*..... - 389 -

FIGURE B.58: ALGORITHM OF THE USER-DEFINED FUNCTIONS *AREA\_RES* (LEFT) AND *SRE* (RIGHT). ..... - 390 -

FIGURE B.59: ALGORITHM OF THE USER-DEFINED FUNCTION *SHIFT*. ..... - 391 -

FIGURE B.60: ALGORITHM OF THE SPRAGUE & GEERS METRIC IMPLEMENTED IN RSVVP. .... - 394 -

FIGURE B.61: ALGORITHM OF THE ANOVA METRICS IMPLEMENTED IN RSVVP. .... - 395 -

## List of Tables

TABLE B. 1: DEFINITION OF MPC METRICS .....	- 321 -
TABLE B. 2: DEFINITION OF SINGLE-VALUE METRICS. ....	- 321 -
TABLE B. 3: DEFINITION OF ANOVA METRICS. ....	- 322 -
TABLE B. 4: VALUE OF THE SPRAGUE & GEERS METRIC COMPONENTS CALCULATED USING THE V&V PROGRAM.....	- 379 -

## **B.1 PROGRAMMER'S GUIDE**

This guide describes the implementation of the Roadside Safety Verification and Validation Program (RSVVP) developed under the NCHRP 22-24 project. The main intent of the guide is to provide the programmer with a comprehensive description of the various parts which compose the RSVVP code and their corresponding algorithms. For this reason, this programmer's manual has the form of a "service" manual. The programmer can refer to this manual to retrieve all the information necessary to locate the section of the code which performs a specific operation and understand the implemented algorithms or, vice versa, given a specific section of the code, the programmer can go back to the task that part of the code performs. In this manner, whatsoever the starting point is, the programmer would always be able to either quickly locate the desired portion of the code or understand what a specific part of the code accomplishes. This information may be useful for future improvements, modification or customization of the original code.

The manual is organized in the following manner. Chapter B.3 gives an initial overview of the different tasks performed by RSVVP along with an explanation of the theory which resides behind them. Both a general and detailed description of the structure of the code and the algorithms used to implement each task is then described in Chapter B.4.

RSVVP has been completely written in Matlab (version R2009 a) [1] and the source code can be either executed directly from the Matlab environment or compiled as an executable application. In the latter case, it is necessary to have the Matlab Compiler Runtime (MCR) component installed on the machine on which it is desired to run the executable. See Appendix B.II for further detail about how to compile and run RSVVP as an executable application.

## **B.2 INTRODUCTION**

RSVVP quantitatively compare one or multiple pairs of curves by computing comparison metrics which are objective, quantitative mathematical measures of the agreement between two curves. The comparison metrics calculated by RSVVP can be used to accomplish one or more of the following operations:

- validate computer simulation models using data obtained from experimental tests
- verify the results of a simulation with another simulation or analytical solution
- assess the repeatability of a physical experiment

Although RSVVP has been specifically developed to perform the verification and validation of roadside safety models, it can be used to generally perform a comparison of virtually any pair of curves. All the comparison metrics evaluated by RSVVP are deterministic, meaning they do not specifically address the probabilistic variation of either experiments or calculations (i.e., the calculation results are the same every time given the same input).

In order to ensure a correct comparison of the curves, RSVVP gives the user the option to perform various preprocessing tasks before the metrics are calculated.

The intuitive and interactive graphical user interface of RSVVP allows the user to effortlessly input the curves to be compared and easily perform any of the existing preprocessing operations. Also, a series of automatic warnings alert the user about possible mistakes during the preprocessing phase.

For programmers interested in modifying or improving the original code, The Matlab<sup>®</sup> source code of RSVVP can be downloaded from:

[http://civil-ws2.wpi.edu/Documents/Roadsafe/NCHRP22-24/RSVVP/Source\\_code.zip](http://civil-ws2.wpi.edu/Documents/Roadsafe/NCHRP22-24/RSVVP/Source_code.zip)

## **B.3 DESCRIPTION OF TASKS**

This chapter gives a description of the operations performed by the program and, when it is possible, the theoretical background behind the tasks is presented, as well.

The actions performed by RSVVP can be logically categorized into six main tasks:

- I. Input of data
- II. Preprocessing
- III. Selection of metrics/time interval
- IV. Metrics evaluation
- V. Postprocessing
- VI. Output of results

Each task may be further divided into various steps or subtasks. A description of the steps performed for each of the above mentioned tasks is given in this chapter.

### **B.3.1 Input of data**

Data are input by loading ASCII files containing the data points of the curves. Each curve must be defined by a distinct file containing two columns; the first column representing the time and the second one the value of the curve at the corresponding time. After the sets of data have been loaded, the program automatically calculates the minimum sampling rate and the maximum time value based on the time vector of each curve. These values are used to perform some of the preprocessing operations.

### **B.3.2 Preprocessing**

The program performs various preprocessing operations. Some of them are necessary and automatically executed, while others are optional and the user can decide whether to accomplish them or not. The following preprocessing tasks are implemented in the code:

- Scaling
- Manual trimming and/or Shift of the origin
- Filtering
- Shift/Drift

- Resampling and Trimming
- Synchronization

The resampling and trimming operations are performed by default as they are necessary to correctly compare any generic couple/s of curves.

In the next sections, a brief description of each preprocessing operation and the theory/method implemented are given. For a description of the algorithms used to implement each preprocessing feature, see Appendix B.IV.

### **B.3.2.1    Scaling**

The original input curves can be scaled by an arbitrary user defined factor. This operation may be useful when the true and test curves have been collected using different units. The scaling of the original curves is performed multiplying the vector containing the data point by the user-defined scale factors. In case the user has not input any of the two optional scale factors, one for each of the two curves of the pair, the default values are automatically set to the unity.

### **B.3.2.2    Manual trimming and/or shift of the origin**

The manual trimming of the original curves is performed after the rescaling operation. After the user has defined the minimum and maximum extreme values for either one or both the two curves, the value of the index corresponding to (or approximating, in case the sampling rate does not allow the exact fit) these values are calculated. The vectors containing the data point for each curve are then trimmed using the index values previously found. After being trimmed, the time vectors are shifted in order to start from the origin, by subtracting the corresponding initial value.

Note that, at this point, the two curves may be still characterized by different sampling periods as the trimming operation is performed independently for each of them.

### **B.3.2.3    Filtering**

Filtering the time histories is the first step of data preprocessing and is a very frequent operation in the preprocessing of data. The filtering operation is performed implementing a digital filter which complies with the specifications of the SAE J211 standard [2], the reference in matter of filtering for the NCHRP Report 350 [3] and EN1317 [4]. A digital four-pole Butterworth low-

pass filter is implemented using an algorithm which uses a double-pass filtering option (forward/backward): data are filtered twice, once forward and once backward using the following difference equation in the time domain:

$$Y(t) = a_0 \cdot X(t) + a_1 \cdot X(t-1) + a_2 \cdot X(t-2) + b_1 \cdot Y(t-1) + b_2 \cdot Y(t-2) \quad (1)$$

Where:  $X(t)$  is the input data sequence and  $Y(t)$  is the filtered output sequence.

The filter coefficients vary with the CFC value and are calculated using the following formulas:

$$a_0 = \frac{\omega_i^2}{1 + \sqrt{2}\omega_i + \omega_i^2} \quad (2)$$

$$a_1 = 2a_0 \quad (3)$$

$$a_2 = a_0 \quad (4)$$

$$b_1 = \frac{-2\omega_i^2 - 1}{1 + \sqrt{2}\omega_i + \omega_i^2} \quad (5)$$

$$b_2 = \frac{-1 + \sqrt{2}\omega_i - \omega_i^2}{1 + \sqrt{2}\omega_i + \omega_i^2} \quad (6)$$

where,

$$\omega_i = 2\pi \cdot CFC \cdot 2.0775 \quad (7)$$

$$\omega_i = \frac{\sin(\omega_i \cdot T/2)}{\cos(\omega_i \cdot T/2)} \quad (8)$$

In order to avoid the typical scatter at both the beginning and the end of the filtered time histories due to the application of the difference equation (1), a head and tail are added to the original data sets consisting respectively of a simple repetition of the first and last data value. Once the modified data sets are filtered, the head and tail are deleted from the final filtered curve. The length of the head and tail is equal to the closest integer approximation of the curve frequency divided by 10.

### B.3.2.4 Shift/drift correction

The shift and drift effects are due to the heating of sensors. In particular, the shift effect is homogeneous vertical translation of the entire experimental curve due to the change of the sensor temperature between the time it has been zeroed and the time the test has been performed (Figure B.1 a). Instead, the drift effect is a linearly increasing translation of the experimental curve due to a change in the temperature of the sensor between the beginning of the test and the end (Figure B.1 b).

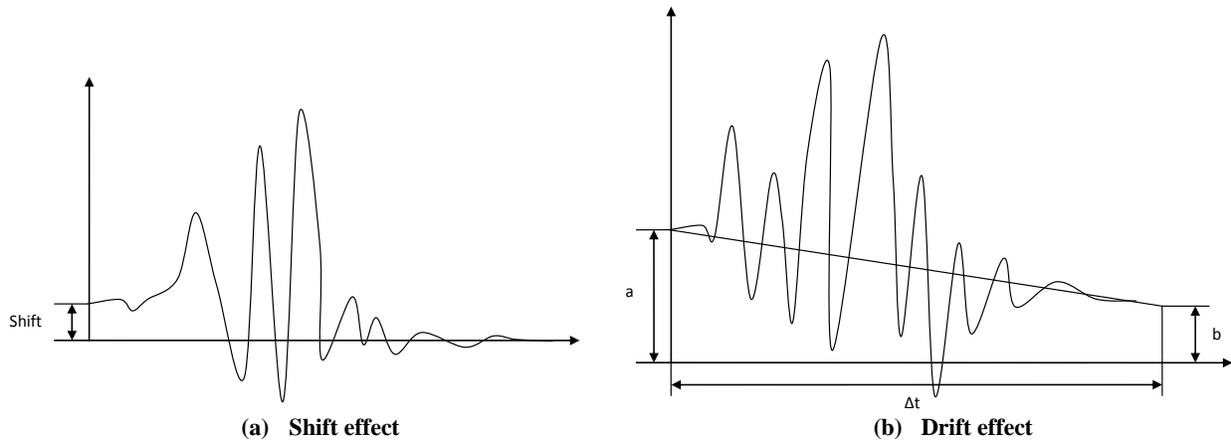


Figure B.1: Representation of (a) Shift and (b) drift effects.

The drifted curve is basically shifted by a value which linearly increases/decreases in the time according to the following simple formula:

$$\Delta = m \cdot t \quad (9)$$

where  $m$  is the slope of the linear drift and can be easily computed as:

$$m = (b - a)/\Delta t \quad (10)$$

with  $a$  and  $b$  representing respectively the initial and final values of the linear drift function and  $\Delta t$  its total length as shown in Figure B.1 b.

The correction of the shift effect can be easily achieved by translating the whole curve by the shift value  $a$ . As for the drift effect, once the value of the slope  $m$  has been calculated from Equation (10), it is possible to correct it by translating each point of the curve by the opposite of the corresponding values obtained from the drift baseline function (Equation (9)). Note that, as the drift baseline is not a constant function, the correction value is different for each sample point of the original curve.

In order to calculate the shift or drift values by which the input curves have to be translated, it is necessary to consider the mean values at the beginning and the end of each curve. In particular, it is important to ensure that the values at the beginning and the end of the vector containing the data points (i.e., head and tail) are sufficiently constant to guarantee that the mean values of these sub-vectors effectively represent the shift/drift values. An initial guess for the point till which the head of the curve can be considered as constant (or from which the tail of the curve is considered constant) is the time at which the curve reaches 5 percent of the peak value (or from which the curve is less than 5 percent of the curve's peak). In order to check that the initially guessed head or tail sub-vectors are constant, their standard deviation must be under a critical value, which was defined to be 0.1. In case the initial sub-vectors do not satisfy the above mentioned criterion, the algorithm iteratively decreases the initial length of the head and tail sub-vectors till the standard deviation of the new reduced sub-vectors is less than the proposed critical value.

#### **B.3.2.5 Resampling and Trimming**

As most of the metrics which quantify the agreement between the test and simulation curves are based on the evaluation of the point-to-point error, in order to correctly calculate the residuals, it is mandatory that the two curves have the same sampling rate. The original curves, however, may have been sampled at different frequencies so it could be necessary to re-sample them in order to compute a point-to-point difference. The program checks if the two sets of data have the same sampling period within a tolerance of 5E-6 sec. If the two curves do not have the same sampling frequency, RSSVP proceeds to re-sample the curve which has the lower sampling rate (i.e., the bigger difference in time between two contiguous data points) at the higher rate of the other curve. The re-sampling is performed by mean of a linear interpolation, assuming that the time vector starts from zero. Also, when the two curves are resampled, the smaller of the end values between the two original time vectors is considered, in order to trim them to the same interval. Note that, because of the new sampling rate the end value of the new time vector may be approximated by defecting the maximum time of the original curve.

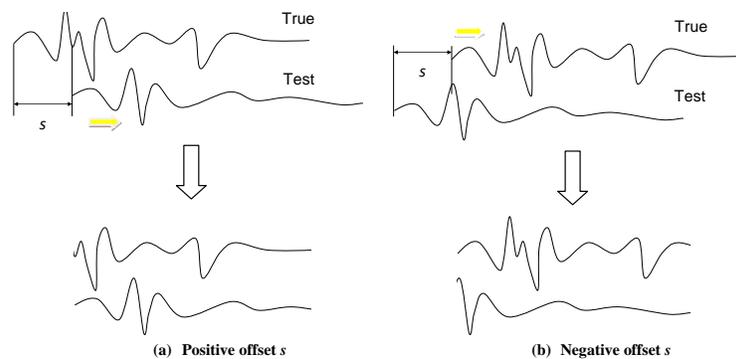
#### **B.3.2.6 Synchronization**

Usually the experimental and numerical time histories do not start at the same time and, hence, the two curves are shifted by a certain value in the time direction. In this case, the curves should

be shifted back or forth so that the impact time in each of them is synchronized. As the comparison metrics are mostly based on the evaluation of the residual error, it is necessary to delete or at least reduce as much as possible any time shift between the test and simulation curves; otherwise, even if the two curves are exactly the same, this gap in time would negatively affect the final metric result. The last preprocessing operation performed by the RSSVP before the metric evaluation is synchronizing the test and simulation curves.

Two different methods of synchronizing have been implemented in RSVVP, based on (1) the minimum area between the curves or (2) the least square error of the residuals. The main idea in both the cases is to find the shift value which minimizes the target function.

In order to implement these two methods of synchronization, a specific function shifts along the time direction either one of the two curves by a value  $s$ , with a positive value of  $s$  meaning a forward shift for the test curve and a negative value being equivalent to a backward shift for the simulation curve (Figure B.2). RSVVP identifies the shift value which minimizes either the absolute area of residuals (method 1) or the sum of squared residuals (method 2). The shift value corresponding to the minimum error is the most probable matching point between the curves. In case the result is not satisfactory, the user can repeat the synchronization procedure using a different initial shift value at the beginning of the minimization algorithm or using the other minimization method.



**Figure B.2: Sketch of the behavior of the shift subroutine for a (a) positive or (b) negative input  $s$ .**

### B.3.3 Metrics evaluation

Various comparison metrics have been implemented into the code. The mathematical formulation of the metrics is shown in Table B. 1 through Table B. 3, where the measured and

computed data points are indicated respectively as  $m_i$  and  $c_i$ . For more details about the comparison metrics implemented in RSVVP refer to the Appendix A in the User's Manual.

For a description of the algorithms used to implement each metric, see Appendix B.V.

**Table B. 1: Definition of MPC metrics.**

	<i>Magnitude</i>	<i>Phase</i>	<i>Comprehensive</i>
<b>Integral comparison metrics</b>			
<b>Geers</b> [5]	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2}} - 1$	$P_G = 1 - \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_G^2 + P_G^2}$
<b>Geers CSA</b> [6]	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2}} - 1$	$P_{CSA} = 1 - \frac{ \sum c_i m_i }{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\text{sign}(\sum c_i m_i) \sqrt{M_{CSA}^2 + P_{CSA}^2}$
<b>Sprague&amp;Geers</b> [7,8]	$M_G = \sqrt{\frac{\sum c_i^2}{\sum m_i^2}} - 1$	$P_{SG} = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{M_{SG}^2 + P_{SG}^2}$
<b>Russell</b> [9]	$M_R = \text{sign}(m) \cdot \text{Log}_{10}(1 +  m )$ where $\frac{(\sum c_i^2 - \sum m_i^2)}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$P_R = \frac{1}{\pi} \cos^{-1} \frac{\sum c_i m_i}{\sqrt{\sum c_i^2 \sum m_i^2}}$	$\sqrt{\frac{\pi}{4} (M_R^2 + P_R^2)}$
<b>Point-to-point comparison metrics</b>			
<b>Knowles&amp;Gear</b> [8]	$M_{KG} = \sqrt{\frac{\sum \left(\frac{ m_i }{m_{max}}\right)^p (\tilde{c}_i - m_i)^2}{\sum \left(\frac{ m_i }{m_{max}}\right)^p (m_i)^2}}$ where $\tilde{c} = c(t - \tau)$ (with $\tau = TOA_c - TOA_m$ )	$P_{KG} = \frac{ TOA_c - TOA_m }{TOA_m}$	$\sqrt{\frac{10M_{KG}^2 + 2P_{KG}^2}{12}}$

**Table B. 2: Definition of single-value metrics.**

<b>Integral comparison metrics</b>			
<b>Correlation Coefficient</b> [10]	$\frac{n \sum c_i m_i - \sum c_i \sum m_i}{\sqrt{n \sum c_i^2 - (\sum c_i)^2} \sqrt{n \sum m_i^2 - (\sum m_i)^2}}$	<b>Correlation Coefficient (NARD)</b> [11]	$h \frac{\sum c_i m_i}{\sqrt{\sum c_i^2} \sqrt{\sum m_i^2}}$

<i>Weighted Integrated Factor</i> [12]		$\sqrt{\frac{\sum \max(m_i^2, c_i^2) \cdot \left(1 - \frac{\max(0, m_i \cdot c_i)}{\max(m_i^2, c_i^2)}\right)^2}{\sum \max(m_i^2, c_i^2)}}$	
<b>Point-to-point comparison metrics</b>			
<i>Zilliacus error</i> [13]	$\frac{\sum  c_i - m_i }{\sum  m_i }$	<i>RMS error</i> [13]	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum m_i^2}}$
<i>Theil's inequality</i> [14]	$\frac{\sqrt{\sum (c_i - m_i)^2}}{\sqrt{\sum c_i^2} + \sqrt{\sum m_i^2}}$	<i>Whang's inequality</i> [13]	$\frac{\sum  c_i - m_i }{\sum  c_i  + \sum  m_i }$
<i>Regression coefficient</i> [10]		$\sqrt{1 - \frac{(n-1) \sum (c_i - m_i)^2}{n \sum (m_i - \bar{m})^2}}$	

Table B. 3: Definition of ANOVA metrics.

<b>Average residual error*</b> [15,16]	$e^{-r} = \frac{\sum (c_i - m_i) / m_{\max}}{n}$
<b>Standard deviation of the residuals*</b> [15, 16]	$\sigma^r = \sqrt{\frac{\sum (e^{-r})^2}{n-1}}$
(*) normalized to the peak of the measured values	

Although all the metrics by definition give a scalar value (i.e., a simple number), they are implemented in order to be evaluated on various time intervals characterized by an increasing size. The smaller time interval on which the input curves are compared is ten percent of the total time and, at each step, it is increased in size till it becomes the total interval. In this way, it is possible to reconstruct a time history of the metrics indicating how the values evolve comparing the curves on increasing portions of their total time interval.

### B.3.4 Postprocessing

The postprocessing of data consists of the following operations:

- Compute the weighted average of the metrics

- Plot the time history of the metrics
- Prepare the variable to output results in Excel files.

The program can evaluate metrics considering either a single couple of curves or multiple pairs simultaneously. In the latter case, the results obtained from each pair of curves (channels) are combined together during the postprocessing of data by computing a weighted average. The weighting factors are automatically calculated by the user based on the area of the true curve for each channel. Following is a description of the procedure implemented to calculate the weighting factors:

- Evaluation of the area of the True curve for each acceleration channel,  $a_i$ , and rotational channel,  $v_i$ .
- Evaluation of the sum of the acceleration areas,  $a_{Sum}$ , and rotational areas,  $v_{Sum}$ .
- Evaluation of the local weight of each acceleration channel,  $lw_i^{(a)} = \frac{a_i}{a_{Sum}}$ , and rotational

$$\text{channel } lw_i^{(v)} = \frac{v_i}{v_{Sum}}$$

- Evaluation of the channel weight factors,

$$w_i^{(a)} = \frac{lw_i^{(a)}}{\sum lw_i^{(a)} + \sum lw_i^{(v)}}$$

$$w_i^{(v)} = \frac{lw_i^{(v)}}{\sum lw_i^{(a)} + \sum lw_i^{(v)}}$$

Apart from calculating the weighted values of the metrics, the postprocessing operation also consists in plotting the time histories of the metrics and preparing the variables with a complex data structure which contain the time histories and results for the output in Excel files.

### B.3.5 Output of results

Results are output in various formats:

- ASCII files
- Excel files
- Graphs (bmp. pictures)

In particular, the output of the results in Excel format requires to store the results in variables characterized by particular data structures which will be discussed in detail in the next chapter of this manual (section B.4.6).

## **B.4 PROGRAM STRUCTURE**

This chapter gives a description of how the code is implemented. Initially, a general view of the organization of the code is given while, in the second part of the chapter, a detailed description of each segment of the code is presented.

The code is logically divided into five major blocks and each of them performs one or more of the tasks described in Chapter B.3. The code has a modular structure: all the five blocks which compose the code are contained into a principal script (*Main.m*) which is started at the execution of the program. On their turn, each of the blocks invokes one or more specific secondary scripts to manage the various particular tasks. Both the principal and secondary scripts may recall either specific functions from various Matlab libraries (toolboxes) or user-defined functions programmed *ad hoc* to perform the desired specific operations. Each operation may be performed by one or more scripts and, in some cases, a secondary script may also recall one or more subscripts.

Figure B.3 shows respectively the diagram of the five blocks in which the code of RSVVP has been logically divided. Each block may achieve more than one of the six tasks performed by the program which are described in Chapter B.3. The diagram in Figure B.4 illustrates which tasks are performed by each block. This diagram may be useful to understand in which block(s) of the code a specific task has been implemented.

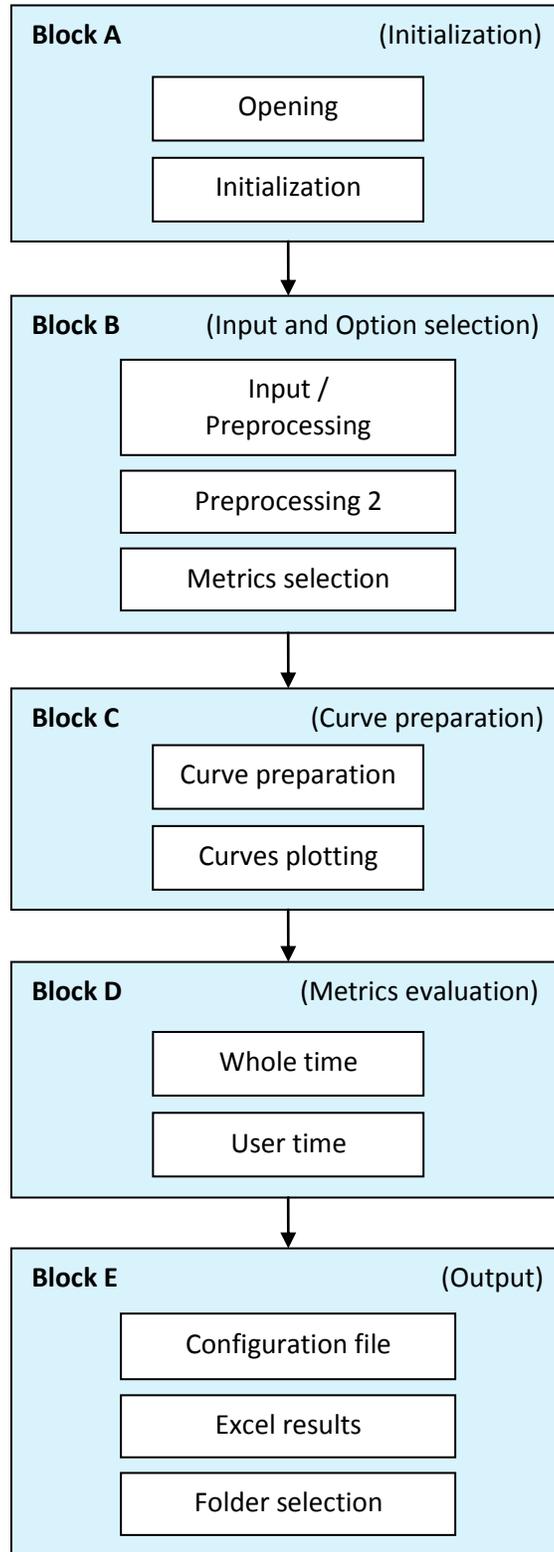
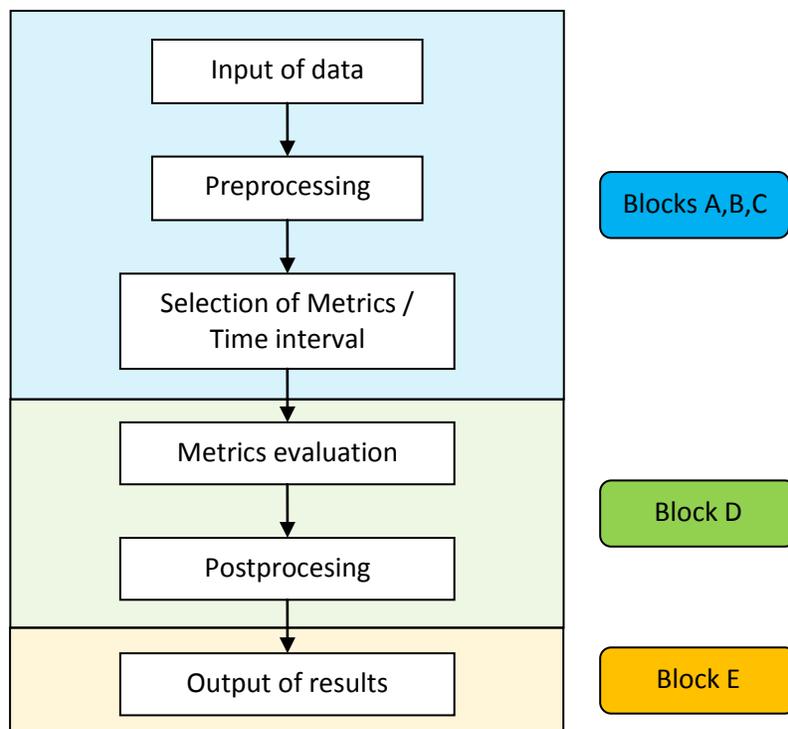


Figure B.3: Diagram of the five main blocks of the RSVVP code.

The information in the manual is organized in order to allow the programmer to easily find either the part(s) of the code where a specific task is performed or vice versa, given a specific section of the code, to back up to the task/s that the code performs. In this manner, whatsoever the starting point is, the programmer would always be able to quickly locate the desired portion of the code or understand what a specific part of the code accomplishes.

As can be seen from the diagram in Figure B.3, the code is composed of the following five blocks:

- I. Initialization
- II. Input & Options selection
- III. Curve preparation
- IV. Metrics evaluation
- V. Output



**Figure B.4: Diagram of the tasks performed by RSVVP.**

## **B.4.1 Organization of the chapter**

Every block is divided into two or three sub-blocks, each of them performing specific operations. In most of the blocks the interaction with the user is accomplished through graphical interfaces.

Section B.4.2 describes the general concepts behind the implementation of the graphical interfaces in Matlab while, in section B.4.3 and following, the user is guided step by step through the algorithms and data structures used to implement the different sub-blocks composing each of the main blocks in which the code is subdivided. Due to various reasons which will be explained later in this manual, in general, it was not possible to implement each of the tasks described in Chapter B.3 in a specific corresponding block (i.e., a one to one correspondence between the tasks and the blocks).

Because of the complexity of the code, the algorithms implemented in each block are described at different levels of detail, starting from a general overview and going more into details at each further level of the flowcharts. In particular, each block is described using flowcharts at three different levels:

- 1. Block level**

Delineates the main frame of the block and the relations between the various sub-blocks.

- 2. Sub-block level**

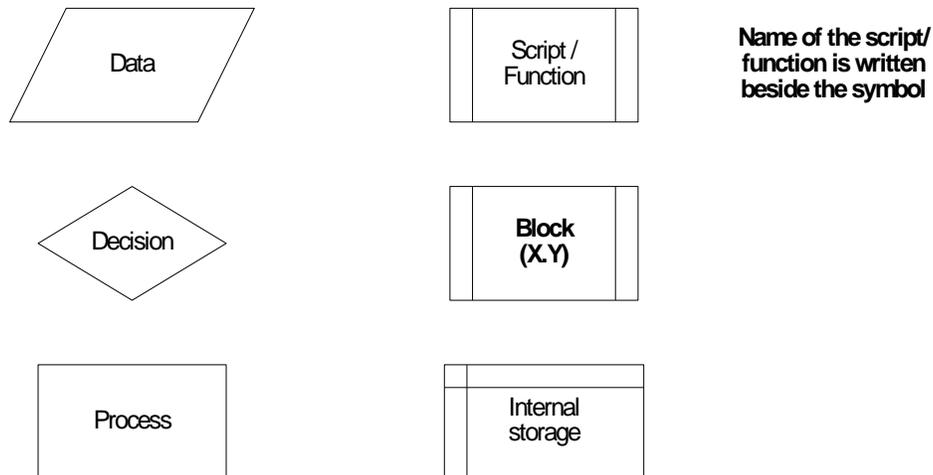
Describes the implementation of each sub-block.

- 3. Script level**

Provides a detailed description of the specific scripts invoked by a sub-block.

### **B.4.1.1 Notation used for the flow charts**

The flowchart diagrams presented in the next sections of this manual have been drawn using a set of standard symbols. Figure B.5 displays the symbols used in the flowcharts and their related meaning.



**Figure B.5: Symbols used for the flowcharts in this manual.**

Note that the filename containing a specific script/function which performs the operation indicated by each 'predefined' shape is shown beside the shape in bold characters.

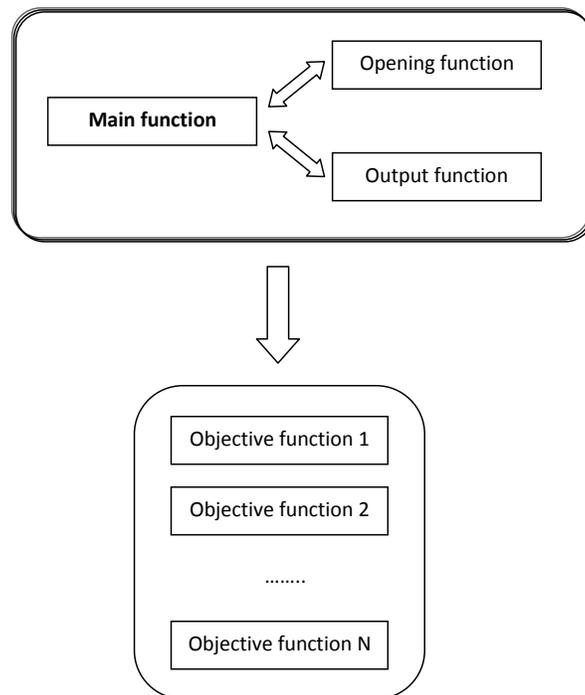
### **B.4.2 Graphical User Interfaces**

The interaction between the program and the user is achieved using various Graphical User Interfaces (GUI's). A graphical interface in Matlab is regarded as a function, which means it is possible to define both input and output variables. The possibility to input variables allows a GUI to load information about the configuration saved during the previous instance or a default configuration in case the GUI is opened for the first time. In fact, some of the GUIs are implemented into iterative loops and, after being invoked for the first time, they may be opened again. In this case, the new instance of a GUI is given as input a variable which contains all the information about the configuration previously saved. In order to organize all the configuration information in a single variable which is easy to pass through the GUI functions, the Matlab structure type has been adopted. A structure variable is composed of multiple fields which can store an array of any available Matlab data type. The fields of a structure variable can be heterogeneous, thus allowing to store in the same variable different type of information. For more information see Appendix B.III.

In Matlab, each GUI is composed of two main components: (1) a figure which includes all the various graphical objects (checkbox, radio button, drop-down menu, plot area, etc.) and (2) an M-file which recalls the figure and manages the various components. The development and

modification of the figure can be achieved only in Matlab by using the command “*guide*” (GUI Developer), which opens a graphical interface for the creation/modification of the GUI figure. The M-file of each GUI is a Matlab script containing a set of functions and is basically composed of the following main parts (Figure B.6):

- Main function
- Opening function
- Output functions
- Object functions



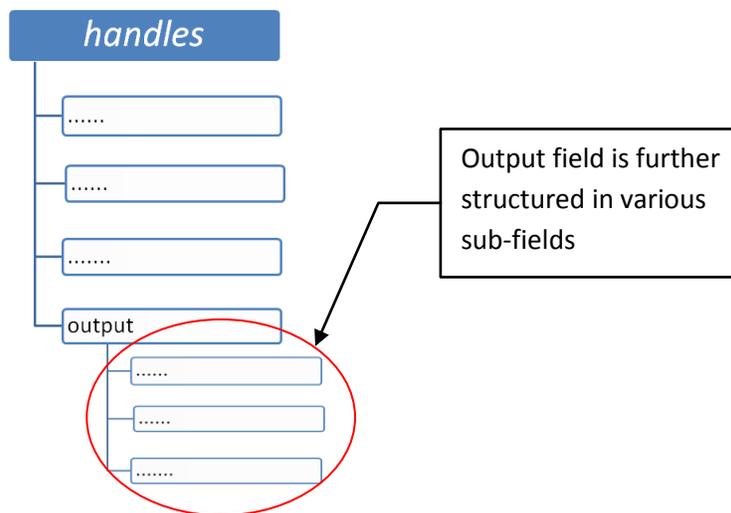
**Figure B.6: Structure of a Matlab GUI.**

The **Main function** has the same name of the GUI and is the function which is recalled in the code to start the GUI itself. This function invokes in sequence the **Opening function** and the **Output function** and is then set into “wait” mode in order to make the GUI wait for user response before returning to the code where it has been invoked. The Opening function performs all the necessary operations to initialize the GUI immediately before it appears on the screen. In particular, it may receive the structure variable passed to the Main function which contains the configuration saved during a previous instance of that GUI. In this case, the fields of the input structure variable are used to properly configure the GUI.

The Output function, instead, manages the data passed back by the GUI function when it is closed.

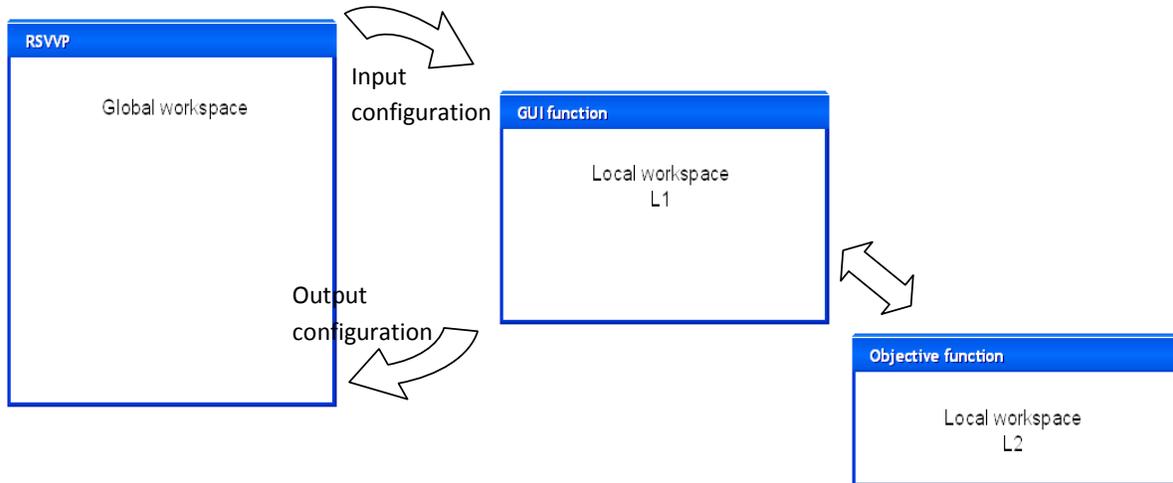
Once the GUI has appeared, the user can select the different objects. To each object of the GUI corresponds a specific **Object function**, which implements the operations to be performed for that particular object. During the period in which the Main function is in wait mode, each time the user interacts with an object of the GUI, the corresponding function is invoked from the local workspace of the Main function. In general, each Object function receives as input the three variables “*hObject*”, “*eventdata*”, “*handles*” from the workspace of the Main function and may return the variable “*handles*” to the same local workspace. In most cases, except some rare exceptions, the exchange of information between an Object function and the Main function is achieved using the field ‘output’ of the structure variable “*handles*” (i.e., *handles.output*).

The field “*handles.output*” is usually further structured into various subfields (Figure B.7) according to specific GUI and is the one which, eventually, is passed back by the Main function to the principal code of the program.



**Figure B.7: Structure of the variable *handles*.**

In case the GUI is iteratively recalled, the values of this structure variable are repeatedly passed back and forth from the main code to the GUI and vice versa (Figure B.8).



**Figure B.8:** Schematic representation of the workspace of the Main and Objective functions of a GUI.

Note that, when the GUI main function closes and returns to the main invoking code, the related figure is not automatically closed by Matlab. Thus, in order to close the figure, it is necessary to add into the main code the command “*close all*” immediately after the command which recalled the GUI.

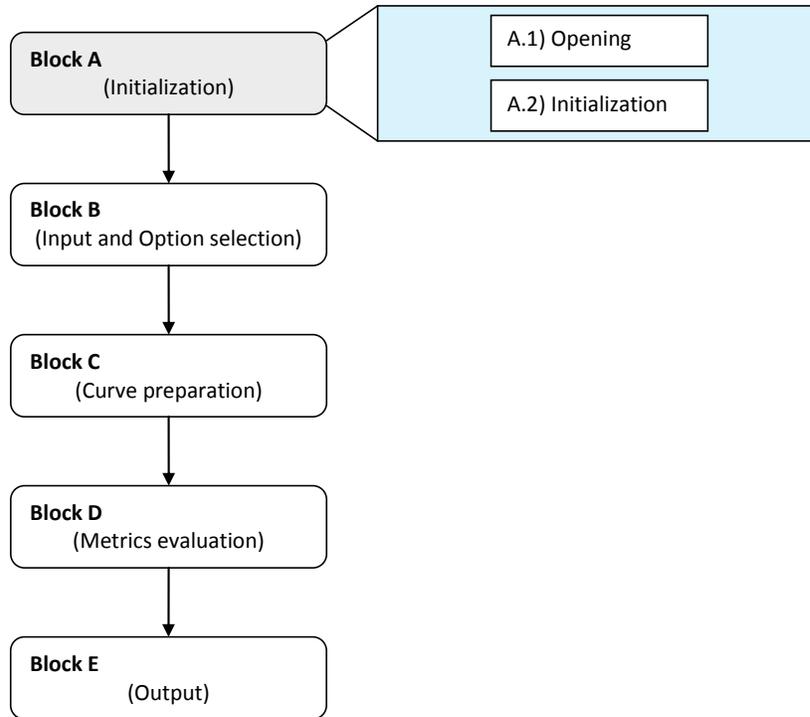
### B.4.3 Block A (Initialization)

Block A is the first of the blocks in which the program code is subdivided; it performs the initial operations necessary to start the program. The functions performed by this block are: defining the type of comparison, selecting between some basic choices and performing the initialization of the major variables used during the execution of RSVVP. Moreover, in case a configuration file is input, this block performs the following tasks: (i) input of data and (ii) preprocessing.

As shown in Figure B.9, Block A is divided into two sub-blocks:

- Opening and
- Initialization.

A detailed description of each sub-block is given in the following sections.



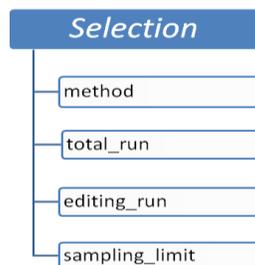
**Figure B.9: Diagram of Block A.**

#### B.4.3.1 Opening (A.1)

The sub-block *Opening* contains the first interactive Graphical User Interface (GUI) of the code. This GUI, which is called *GUI\_Start*, manages the user choice for the following options:

- Single / Multiple channel
- Resampling rate limit
- Configuration file

When the GUI function *GUI\_Start* closes, the options selected by the user are passed to the variable *Selection* which is used afterwards in the code.



**Figure B.10: Fields of the variable *Selection*.**

Figure B.10 shows the fields of the structured variable *Selection* which are used to store the options.

### B.4.3.2 Initialization (A.2)

The sub-block *Initialization* performs an initialization of most of the variables used by RSVVP or, in case a configuration file has been loaded, it reads the variables containing information about the preprocessing options from the file and performs the necessary input/preprocessing operations. Figure B.11 shows the main structure of the sub-block *Initialization*. As can be seen from the flow chart, different operations are performed according to whether a configuration file has been input or not.

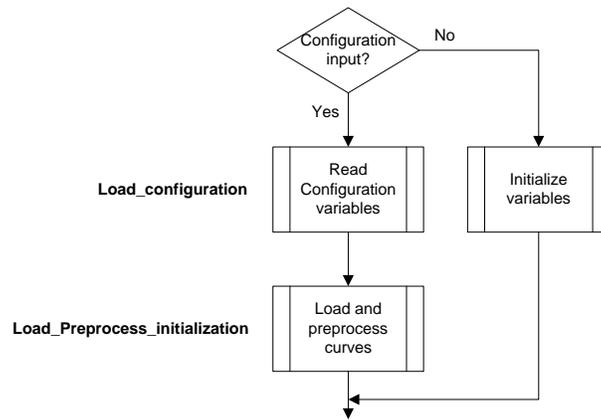


Figure B.11: Flow chart of the algorithm of sub-block A.2 (Initialization).

In the case no configuration file is input, the sub-block A.2 initializes most of the variables used during the execution of the code. Note that, although in Matlab<sup>®</sup> there is no need to statically allocate memory for the definition of variables, the way RSVVP has been implemented requires that an initial default value is assigned to the variables which control the input/preprocessing options. Following is a list of the variables initialized:

- *Reply* (*v*)
- *Reply\_2* (*v*)
- *Procedure*
- *Channel\_list*
- *Time\_interval* (*v*)
- *Time\_interval\_total\_run* (*v*)

In the case of multiple channels, most of the option values for each input channel are stored into vectors instead of scalar variables (The variables which become vectors are indicated with “(v)” in the previous list). In this case, each element of the vector represents the default option for the corresponding channel.

In case a configuration file has been loaded, the input curves are read from the original data files and then preprocessed according to the options obtained from the configuration file.

Figure B.12 describes in detail the algorithms used to implement respectively the two scripts *Load configuration* and *Load preprocess* recalled by the main algorithm of Block A.2.

#### *B.4.3.2.1 Load\_configuration*

This script implements the operations necessary to load the variables contained in the configuration file. As can be seen in the flowchart of the algorithm, if the user has selected to load the configuration in edit mode, a copy of the input resampling rate is saved at the beginning in order to be overwritten to the value of the corresponding variable once it has been loaded from the configuration file.

The algorithm checks if the file name of the configuration file is correct and, in case, load the variables; otherwise, a warning message is displayed and the program quits.

In order to avoid a crash of RSVVP in case no channel has been manually trimmed, it is also necessary to delete the corresponding flag variable *Manual\_trim\_config*.

#### *B.4.3.2.2 Load\_preprocess\_initialization*

This script manages the load of the data from the input curves and the preprocessing operations according to the information obtained from the configuration file. The two branches clearly visible in the flowchart diagram of this script indicate the two main sections in which it can be divided. These two sections of the algorithm run in series, one after the other, and each of them contains a loop (both indicated by a red rectangle).

The first branch is a loop which cycles over the total number of channels (i.e., one for the case of a single channel or six for the case of multiple channels). Each iteration of the loop loads the input curves and performs the preprocessing operations according to the information read from

the configuration file. For more details about the scripts or part of the code which perform the loading and the specific preprocessing operations, refer to the next section (Block B). Before concluding each iteration, the preprocessed curves are saved in the matrix variable *Preprocessed* (see section B.4.4.1 for more details about the structure of this variable) and the variable *Ch\_num* is incremented.

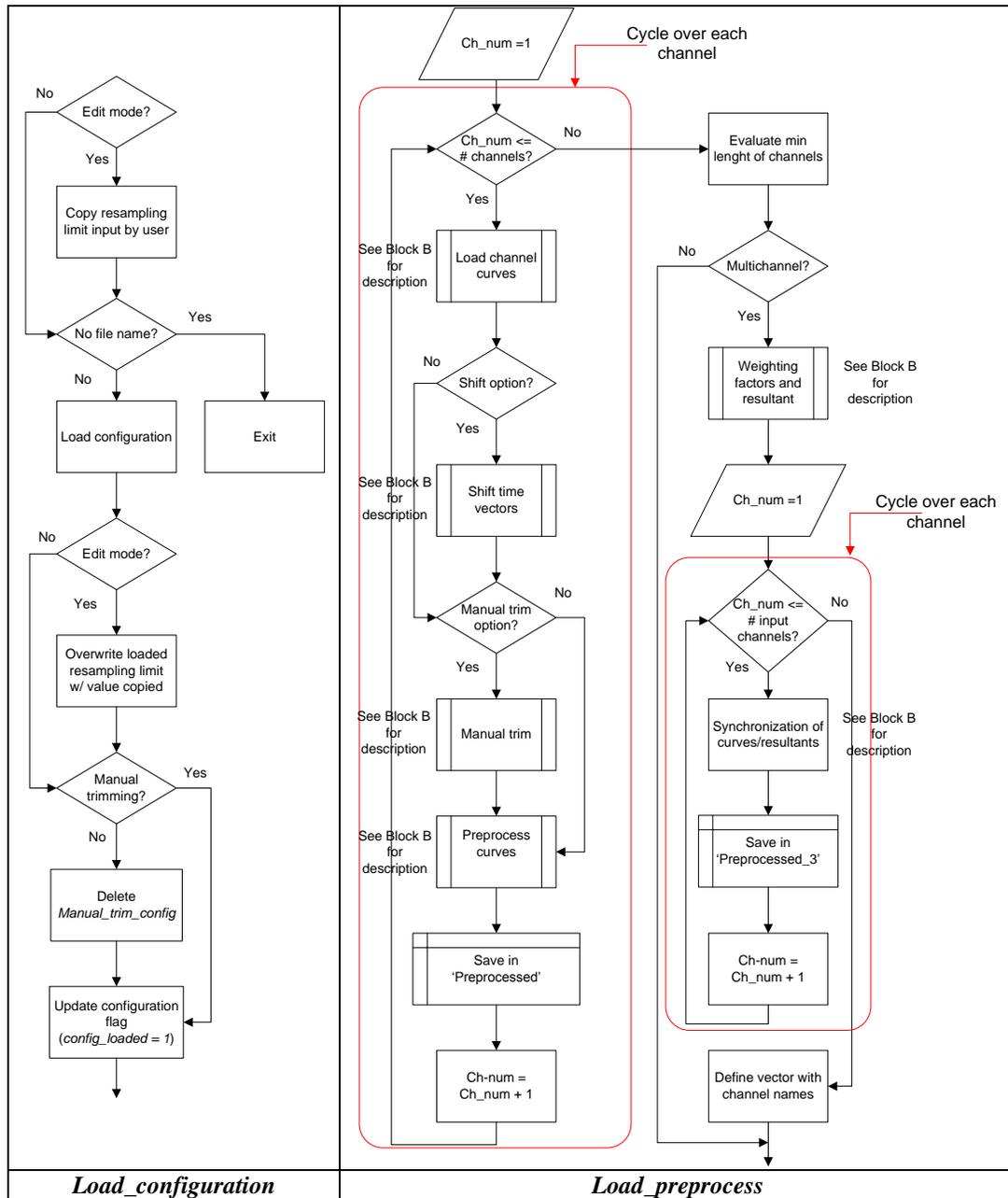


Figure B.12: Flow chart of the scripts *Load configuration* (left) and *Load preprocess* (right) (sub-block A.2).

After the cycle has concluded and all the channels have been input and preprocessed, the minimum length between all the couple of channels is computed. In case of multiple channels, first the weighting factors or the resultants are calculated and, then, the program cycles over the channels/resultants to perform the synchronization of the curves in case it is requested by the configuration file. Eventually, a vector with the names of the specific input channels (or the resultants) is created to be used during during the postprocessing operations for the output of results.

#### B.4.4 Block B (Input and Option selection)

Block B implements most of the interaction activity with the user. In fact, this block manages three different graphical interfaces which respectively perform three different tasks: (i) input of data, (ii) preprocessing and (iii) selection of metrics/time intervals.

The diagram of Block B is shown in Figure B.13; the block is divided into three sub-blocks:

- Input/Preprocessing (GUI\_1\_3),
- Preprocessing 2 (GUI\_1\_3\_II)
- Metrics selection (GUI\_metrics)

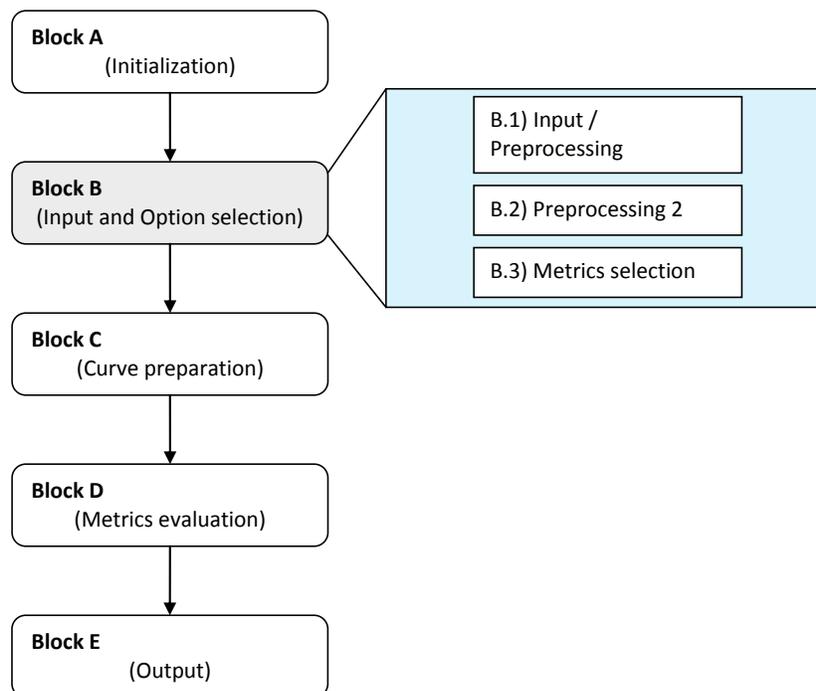
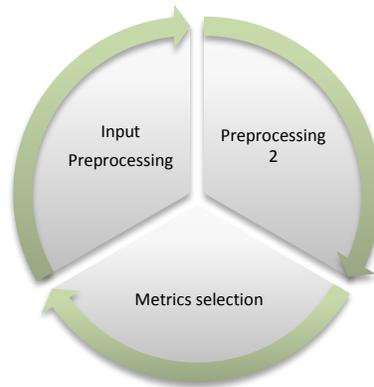


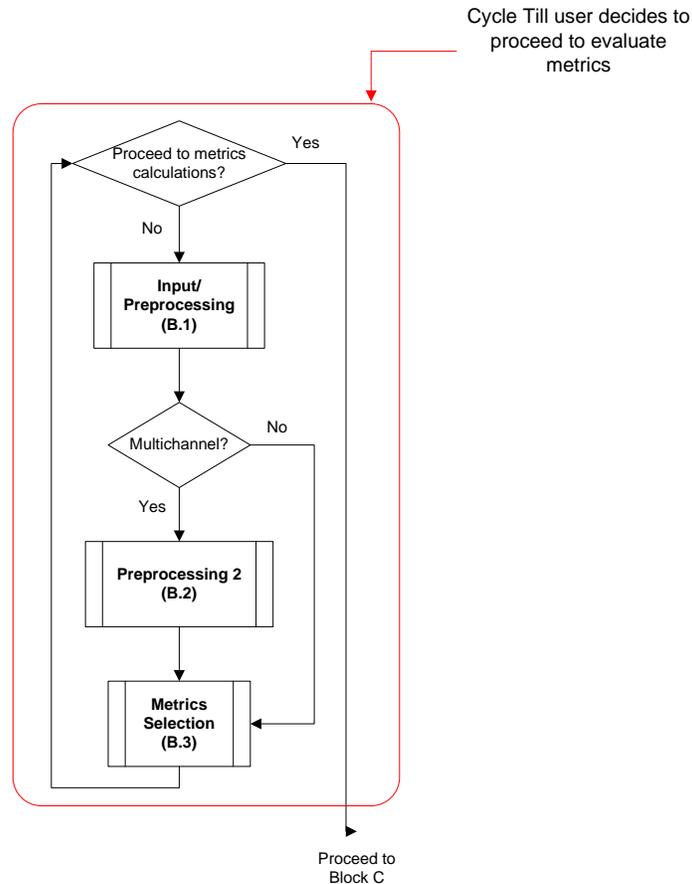
Figure B.13: Diagram of Block B.

The main characteristic of Block B is that the three sub-blocks are implemented in sequence into a loop which terminates only when the user decides to proceed to the evaluation of metrics (Figure B.14).



**Figure B.14: Scheme of the loop which forms Block B.**

Figure B.15 shows the main algorithm of Block B. The implementation of the three sub-blocks into a loop allows the possibility to go back to the previous sub-block to apply any further change to the options selected in the previous instance. In fact, by skipping directly to the next iteration of the loop, it is possible to go back to the first sub-block (i.e., B.1)). In case the user is executing operations implemented in the sub-block B.3 (i.e., the last of the three blocks), it is also possible to skip directly to the second sub-block, B.2, during the new iteration of the loop by defining a flag variable and a conditional statement at the beginning of the first block (see section B.4.4.1 for the details). In this case, given a particular value of the flag variable, the conditional statement for the execution of the first block would skip it and start the new iteration of the loop directly from sub-block B.2.



**Figure B.15: Flow chart of the main algorithm of Block B.**

Any time they open, the graphical interfaces of each of the three sub-blocks load the options and the various data input by the user during the previous iteration of the main loop. This information is read from variables whose structure and content will be specifically discussed ahead in this section. Using such a structure, whenever the user goes back to the previous sub-block, the graphical interface which is reopened shows the same information of the previous iteration.

Note: in the case a configuration file has been loaded, the input/preprocessing operations performed by Block B are skipped as they have already been performed in Block A. See section B.4.4.1 for more details.

A detailed description of each sub-block is given in the following sections.

### B.4.4.1 Input/Preprocessing (B.1)

This sub-block is the first of Block B and implements the GUI which handles the input of the curves and their preprocessing (GUI\_1\_3).

The flow chart in Figure B.16 shows the general implementation of this sub-block. As can be seen from the flow chart, the core of sub-block B.1 is the script *Load\_preprocess*, which is invoked only under the condition that both no configuration file has been input ( $config\_loaded = 0$ ) and that GUI\_1\_3 is set to be open ( $open\_GUI\_1\_3 = 1$ ). The former condition avoids that curves are input and preprocessed when a configuration file has been loaded, while the latter condition is used to skip the execution of this block in order to go directly to the next sub-block during the new iteration of the main loop defined by the cycle in Figure B.15. In either the case the script *Load\_Preprocess* has been invoked or not, before proceeding to the sub-block B.2, a copy of the original input curves is saved.

If the *Exit* button has been pressed during the execution of GUI\_1\_3, the algorithm terminates the program.

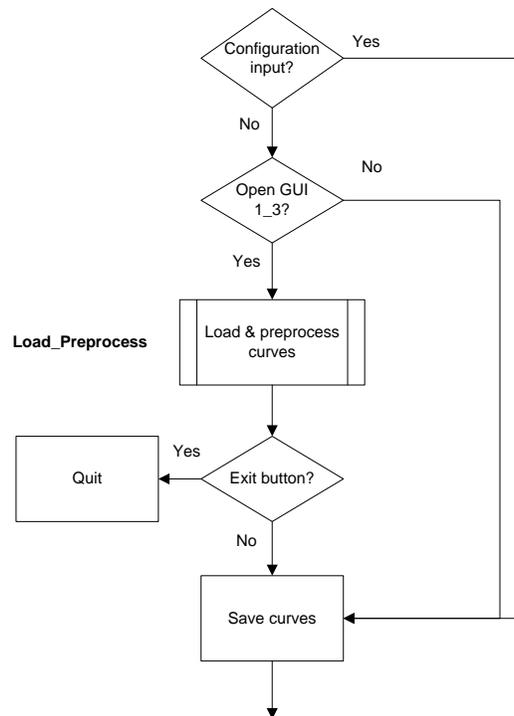


Figure B.16: Flow chart of the algorithm of sub-block B.1 (Input/Preprocessing).

#### B.4.4.1.1 *Load\_Preprocess*

As previously mentioned, this script is the core of Block B.1 which manages the input of the channel/s and the corresponding preprocessing. The algorithm of the script is shown in Figure B.17 and Figure B.18.

The script is embedded into a loop which stops only when, after the last channel has been input and preprocessed, the user decides to proceed to the selection of metrics. Using a loop it is possible to move back and forth through any of the available channels by incrementing/decrementing the variable *ch\_num* which is used to define the specific channel. This allows the user to go back to a previously input channel and make any modification also after this channel has already being input.

The flag variable which controls the loop is *Reply.flag* and the condition to keep cycling is either the value 0 or 4. Initially, before starting the cycle, this flag variable is assigned the null value; while the value 4 means that the preprocessing options of the specific channel have been reset and a new iteration has to be performed on that channel. When the user pushes the button to proceed to the selection of metrics in the corresponding GUI (*GUI\_1\_3*), *Reply.flag* becomes unity and the cycle terminates. The variable *ch\_num* specifies the channel which is considered at each iteration of the loop. The value of this variable is by default equal to 1 in case of a single channel or is assigned by the variable *Reply.channel\_id* (written by the *GUI\_1\_3*) in case of multiple channels.

At the beginning of each iteration of the loop, *GUI\_1\_3* opens. The first time *GUI\_1\_3* is opened for each channel indicated by the variable *ch\_num*, the default values created during the initialization are recalled. When the GUI closes, the algorithm proceeds to load the curves from the selected files and perform the preprocessing according to the selection made by the user. At the beginning of the next iteration, the GUI is opened again and shows the preprocessed curves. In case the GUI has already been opened for a specific channel before, it reloads any previous option or data and, if the user has modified/reset any of the preprocessing options, when the GUI closes the algorithm proceeds to reload the curves and modify the preprocessing according to the new options. In case the last channel has been skipped, the algorithm automatically ends the loop.

Apart from the scaling operation, which is performed by the script *Load\_curves*, and the manual trimming of the curves, which is implemented in the script *Manual\_trim\_shift*, all the other preprocessing operations are invoked by the script *Preprocessing*.

The scaling option is implemented by simply multiplying the vector containing the data points by the scaling factor defined by the user for that specific channel and curve.

Data vectors are manually trimmed by limiting the original vectors within the indexes which are closer to either the lower or upper boundary values, respectively; while, the shift of the time vector is obtained by subtracting the initial time value from the time vectors of the curves.

The other preprocessing operations are performed by recalling the script *Preprocessing* which, on its turn, recalls one or more of the following scripts according to the specific options selected for each channel:

- Filtering
- Shift\_drift
- Resampling\_trimming
- Curve\_synchronizing

The algorithm of these preprocessing operations are described in Appendix B.IV.

The preprocessed curves are saved into the variable *Preprocessed* which is a matrix of cells. As shown in Figure B.19, the  $i^{\text{th}}$  row contains respectively the time, true data and test data vectors of the corresponding channel  $i$ . Note that the possibility to store vectors for each of the element of the matrix *Preprocessed* is achieved by first converting the vector into a cell. In case of a single channel comparison, this matrix reduces to a horizontal vector.

After the loop ends, the pairs of curves from the various input channels are trimmed to the length of the shortest and the algorithm proceeds to the next block.

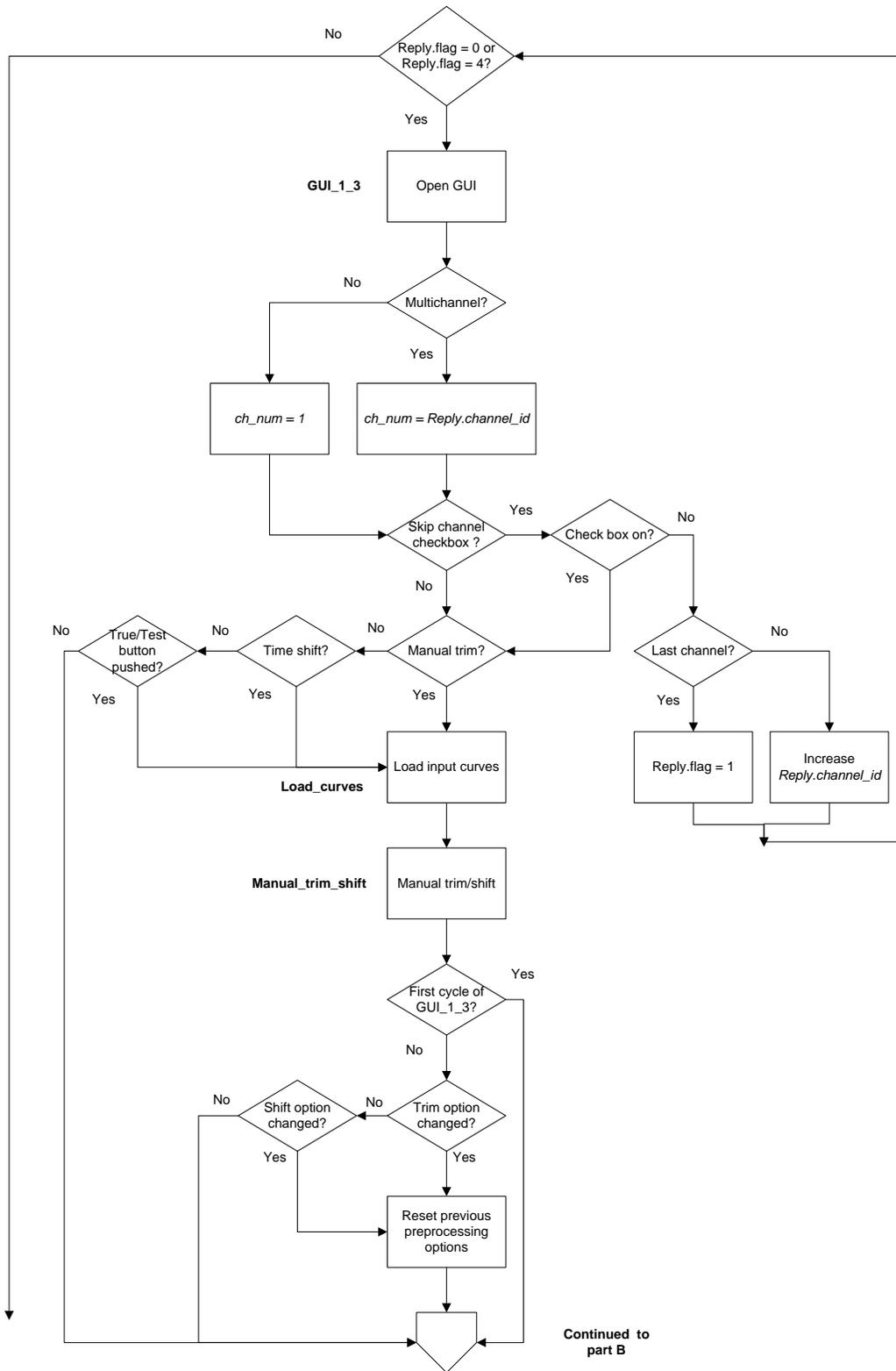


Figure B.17: Flow chart of the script *Load\_Preprocess* (sub-block B.1) – part A.

Continued  
from part A

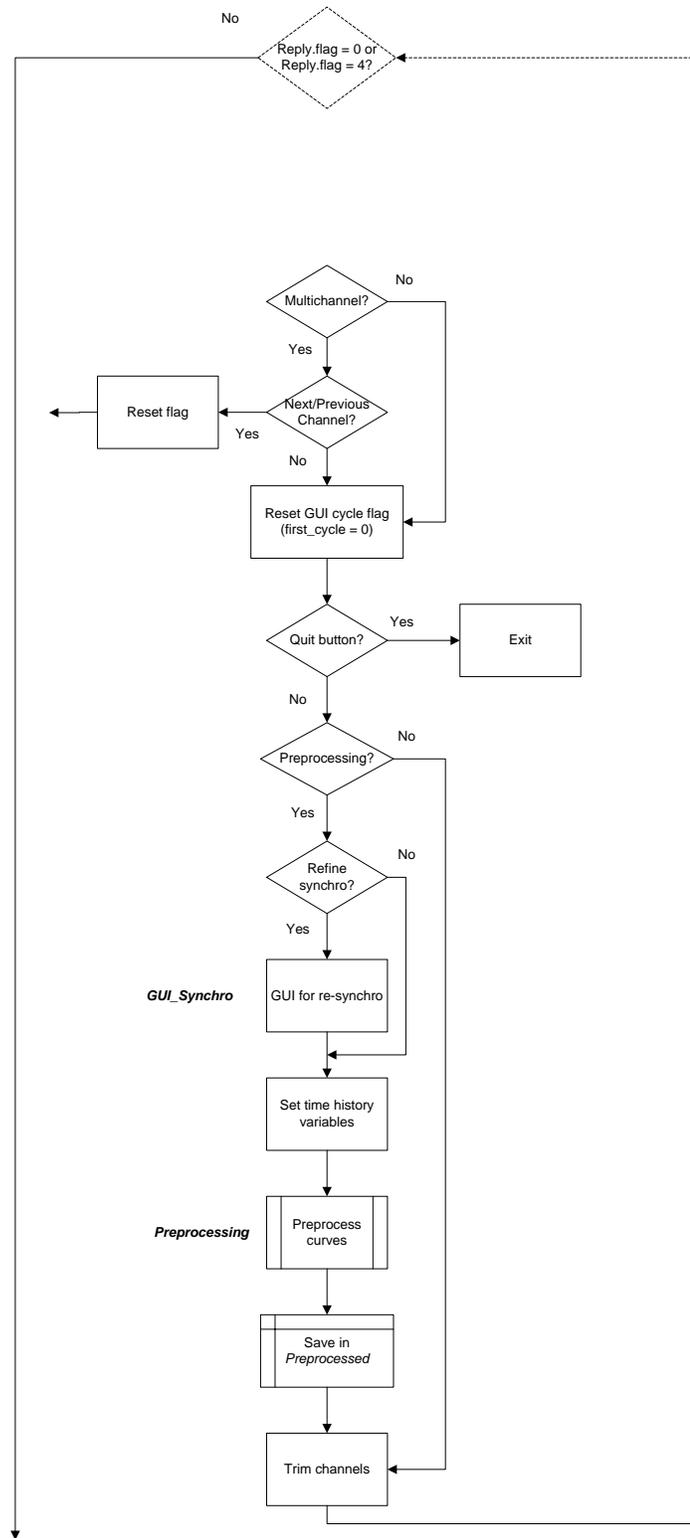


Figure B.18: Flow chart of the script *Load\_Preprocess* (sub-block B.1) – part B.

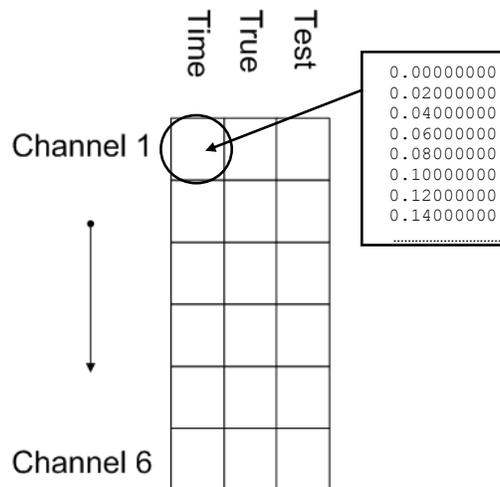


Figure B.19: Sketch of the structure of the variable *Preprocessed*.

#### B.4.4.1.2 *Load\_curves*

This script manages the load of the input curves from the ASCII files provided by the user. In particular, it skips any of the initial rows containing non-numeric characters and scales the data point by the user defined factors. The algorithm of this script is shown in Figure B.20. The same algorithm is repeated of both the test and true curves.

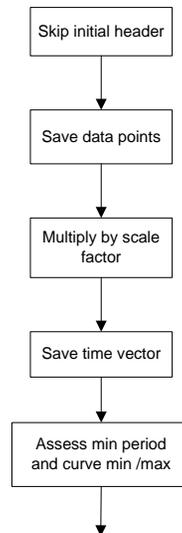


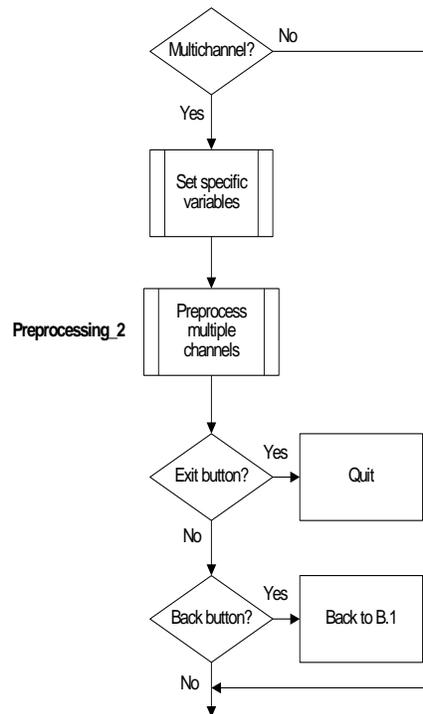
Figure B.20: Flow chart of the script *Load\_curves* (sub-block B.1).

#### B.4.4.2 Preprocessing 2 (B.2)

This sub-block implements the synchronization of the curves in case multiple channels are input. In this case, the synchronization is performed after the single channels have been input and

partially preprocessed in order to allow the user the option to compute the resultants from the acceleration and rotational rate channels. In fact, if the resultant option is selected, the true and test curves from the input channels are combined together in order to first obtain the corresponding resultant curves which are then synchronized.

The flow chart in Figure B.21 shows the general implementation of this sub-block. It is recalled only in the case multiple channels are input (*comparison\_type* = 2) and is completely skipped in the case of a single channel. At the beginning, the sub-block B.2 initializes some variables and then proceeds to the core script *Preprocess\_2* which manages *GUI\_1\_3\_II*. In case the Exit or the Back button has been pressed in *GUI\_1\_3\_II*, the sub-block B.2 respectively quits the program or goes back to *GUI\_1\_3* by forcing a new iteration of the loop described by the flow chart in Figure B.15.



**Figure B.21: Diagram of sub-block B.2 (Preprocessing2).**

#### B.4.4.2.1 *Preprocessing\_2*

This script is the core of sub-block B.2. It manages the synchronization of the multiple channels and the selection of the method to use for computing the equivalent metrics (weighting factors or resultant). Similarly to the previous sub-block, also in this case the script is mostly contained

into a loop in order to implement the possibility to move back and forth through any of the available channels by incrementing/decrementing the variable *ch\_num* which is used to define the specific channel.

The algorithm of the script is shown in Figure B.22. Before entering the loop, the variable *Preprocessed\_2* is initialized as a copy of the variable *Preprocessed*. In this way, the curves originally preprocessed by the previous sub-block are always available in case the user decides to reset the preprocessing performed by this sub-block.

At the beginning of each iteration of the loop, the GUI called *GUI\_1\_3\_II* is opened. Similarly to *GUI\_1\_3*, also the first time that *GUI\_1\_3\_II* is opened for each channel the default values created during the initialization are recalled and, as soon as the GUI closes, the algorithm performs the preprocessing operations. When the GUI is opened again at the beginning of the next iteration, both the initial and preprocessed curves are shown.

If the synchronization option has been selected in the GUI (indicated by the variable *Reply\_2.synchro*) the specific channel is synchronized. Also, in case the user decided to re-synchronize the curves, a new GUI is invoked (*GUI\_Synchro*) to define a new starting point for the synchronization procedure. Note that in this case, after the re-synchronization, the algorithm will go to the next iteration of the loop where *GUI\_1\_3\_II* will be invoked to show the new synchronized curves.

The default option for the computation of multichannel (weighting factors/resultant) is set to weighting factors. In case this option is changed (identified by the variable *Reply\_2.update*) in the GUI, the algorithm proceeds to re-initialize the variable *Preprocessed\_2* accordingly by recalling the script *Initialization\_2*.

In case the iteration of the loop is the first for that channel (*Reply\_2.first\_iteration = 1*), the variable *Preprocessed\_3* is initialized. If the next or previous buttons have been pushed in the *GUI\_1\_3\_II*, a new iteration of the loop is forced and *GUI\_1\_3\_II* is invoked showing the next or previous channels.

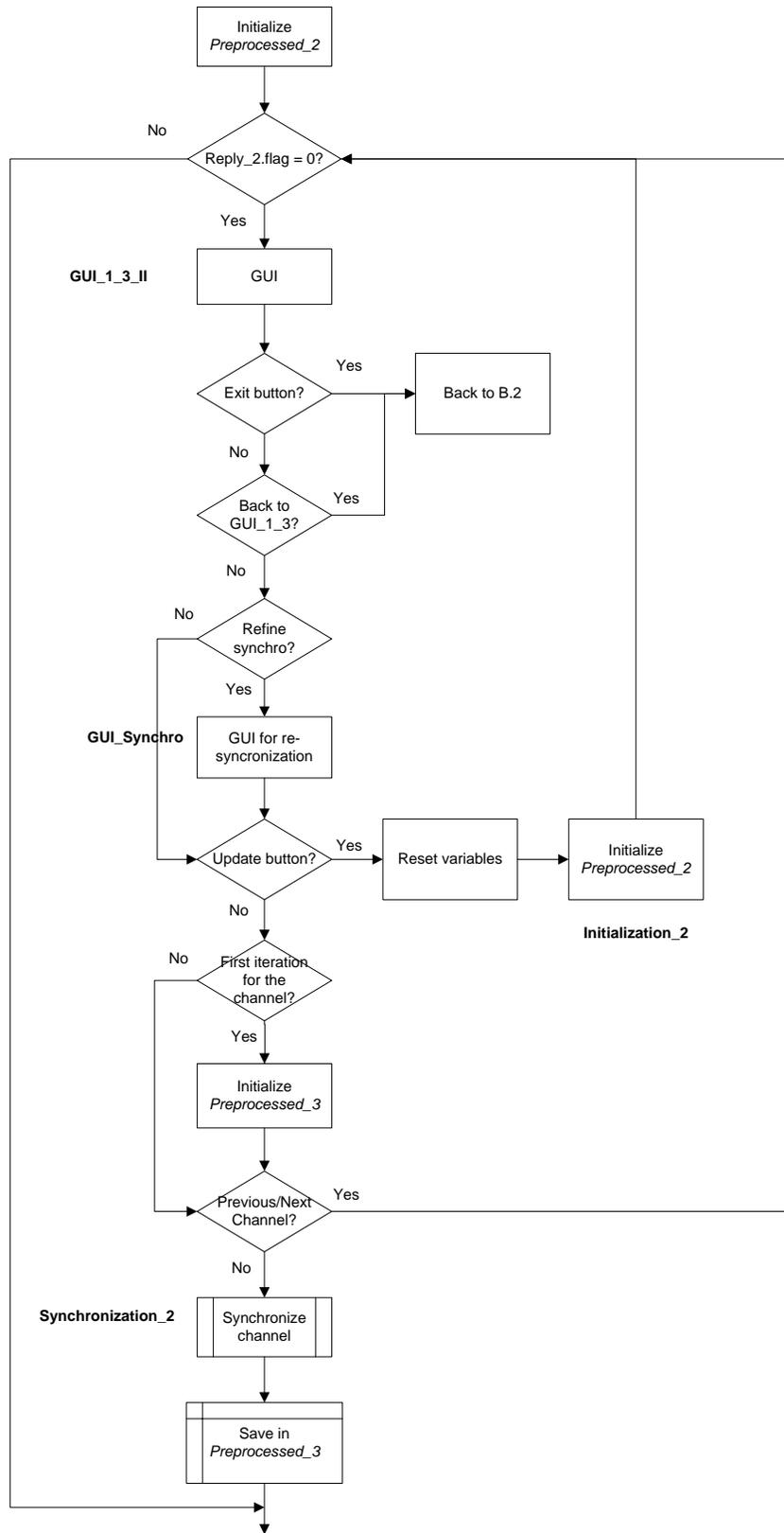


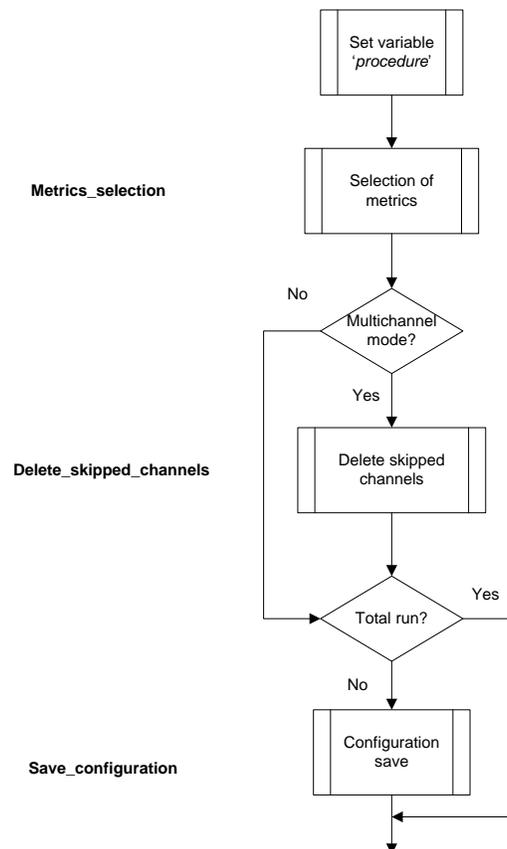
Figure B.22: Flow chart of the script *Preprocessing\_2* (sub-block B.2).

### B.4.4.3 Metrics selection (B.3)

Metrics selection is the last of the sub-blocks of Block B. It implements the GUI for the selection of the metrics and time intervals on which compare the input curves (whole time and/or user defined intervals).

The flow chart in Figure B.23 shows the general implementation of this sub-block. The first action performed is set the value of the variable *procedure* according to the type of multichannel comparison (i.e., weighting factors or resultant). This variable is then used in subsequent parts of the program code.

In case multiple channels are input, a script checks if any channel has been skipped, and in case, null vectors are added to the corresponding row in the variable *Preprocessed\_3*. Also, the algorithm creates a configuration file relative to the RSVVP run. Note that, if a previous configuration file has been loaded without editing (total-run mode), the creation of the configuration file is skipped.



**Figure B.23: Diagram of sub-block B.3 (Metrics selection).**

#### *B.4.4.3.1 Metrics\_selection*

This script is the core of sub-block B.3 and manages the GUI for the selection of the metrics and the type of time intervals on which to compare the curves (whole and/or user-defined time intervals).

The algorithm of the script is shown in Figure B.24. The script manages the GUI for the selection of metrics (*GUI\_metrics*). In case a configuration file is loaded in 'run mode', the GUI is not opened as the information is taken directly from the configuration file and RSVVP automatically re-runs the comparison. Also in this case, once the GUI has collected the information entered by the user, it is closed and the algorithm proceeds according to the selected option/s.

If the back button has been pressed in the GUI, the algorithm set the variables *open\_GUI\_1\_3* and *config\_loaded* equal to zero and then a new iteration of the main loop of Block B is forced. The values assigned to these variables skip the opening of *GUI\_1\_3* at the next iteration of the main loop.

In case the user has selected the metrics and time intervals, the algorithm sets the variable *Metrics\_list* according to the metrics selected in the GUI. Note that, in this case, the variable *Metrics.flag* is set to the unity in order to quit the main loop of Block B and proceed to Block C.

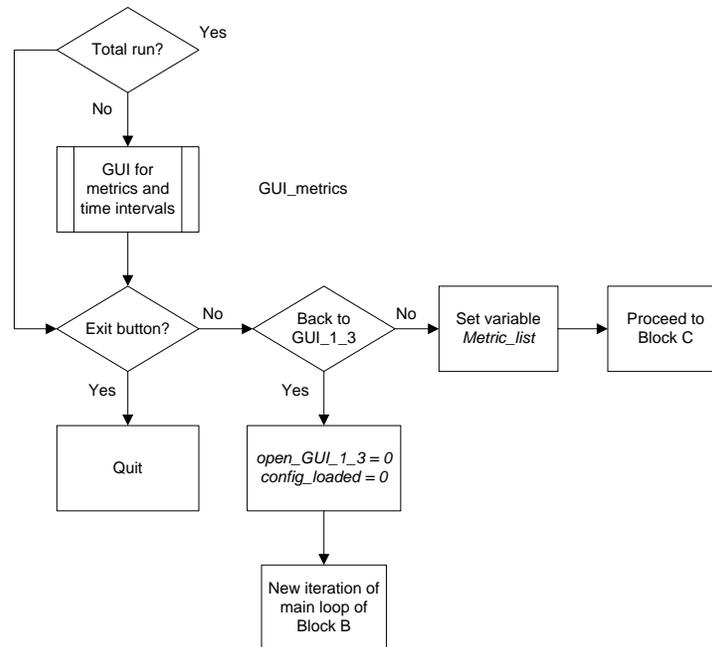


Figure B.24: Flow chart of the script *Metrics\_selection* (sub-block B.3).

### B.4.5 Block C (Curve preparation)

Block C finalizes the preprocessing activities and creates the plots of the input curves after the preprocessing. Figure B.25 shows the diagram of this block, which is composed of three sub-blocks:

- Curve preparation,
- Curves histories and
- Curves plotting.

A detailed description of each sub-block is given in the following sections.

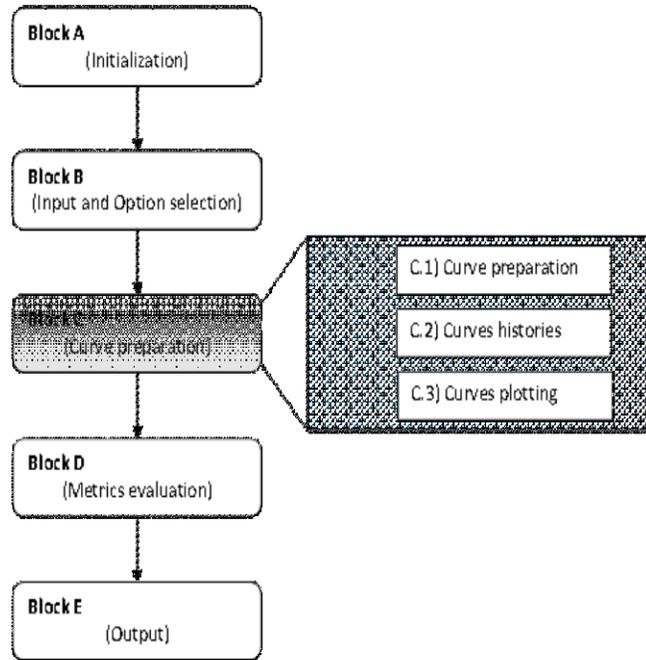


Figure B.25: Diagram of Block C.

#### B.4.5.1 Curves preparation (C.1)

The sub-block *Curves preparation* refines the preprocessing by trimming the couples of curves from the various channels to the length of the shortest channel. This operation is performed only in multi-channel mode (*comparison\_type* = 2).

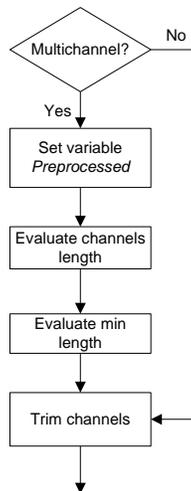
Data points of the input channels are initially stored in the variable *Preprocessed\_3*, which is a matrix of cell (for details about the structure of this variable, refer to section B.4.5). The vectors of the true and test curves of each channel are extracted from the respective cells structures, trimmed and stored in the matrix variable *True* or *Test* correspondingly. As the length of the vectors is the same for all the channels after the trimming, it is possible to store the data in a simple matrix structure without the need to use a cell configuration, like required before the trimming operation.

The organization of the matrices *True* and *Test* is represented in Figure B.26. For both the matrices, the vectors of the data points for the channel  $i$  is saved in the  $i^{\text{th}}$  column.

Channel 6																				
Channel 5																				
Channel 4																				
Channel 3																				
Channel 2																				
Channel 1																				

**Figure B.26: Data organization of the matrix variables *True* and *Test*.**

The flow chart in Figure B.27 shows the general implementation of this sub-block.



**Figure B.27: Diagram of sub-block C.1 (Curves preparation).**

### 1.1.1. Curves histories (C.2)

This sub-block saves the time histories of both the original and preprocessed input curves. The original and preprocessed time histories are managed respectively by the script *Save\_curves\_original* and *Save\_curves\_preprocessed*. The main implementation of this sub-block and the two mentioned scripts are shown in Figure B.28.

As usual, in case of multiple channels, in the algorithm of both the scripts a loop cycles over the total number of input channels.

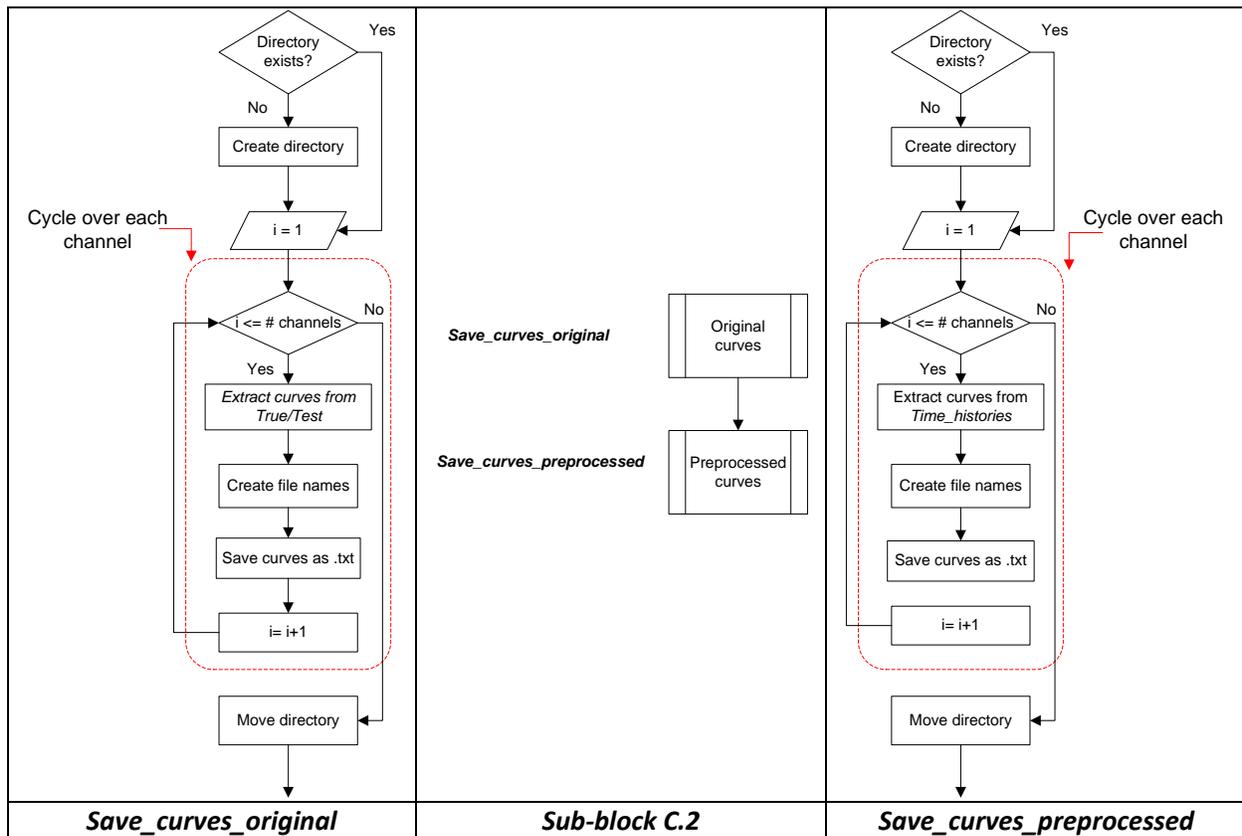


Figure B.28: Diagram of sub-block C.2 (Curves histories) (center) and the two invoked scripts, *Save\_curves\_original* (left) and *Save\_curves\_preprocessed* (right).

### B.4.5.2 Curves plotting (C.3)

The sub-block *Curves plotting* performs two main operations: (i) plot the preprocessed curves after they have been finalized by the previous sub-block (*Curve preparation*) and (ii) evaluate the area of the true curve for each channel.

Before plotting the curves, the code creates the destination folder where the corresponding bitmap files are saved. In case the NCHRP 22-24 profile has been chosen, also the integrals of the original input curves are plotted. In both the cases, a conditional statement based on the value of the variable *zip\_flag* handles the possibility to compress the bitmap files.

The evaluation of the area of each channel is implemented at the end of the block and the corresponding values are saved in the vector variable *Channel\_area*. This vector is used in the following part of the code (Block D) to evaluate the weighting factors for the whole time interval. Note that, for each user-defined time interval, the values of the channel areas are evaluated again in a following part of the code located in Block D (see sub-chapter B.4.6.2 for details).

The flow chart in Figure B.29 shows the general implementation of this sub-block.

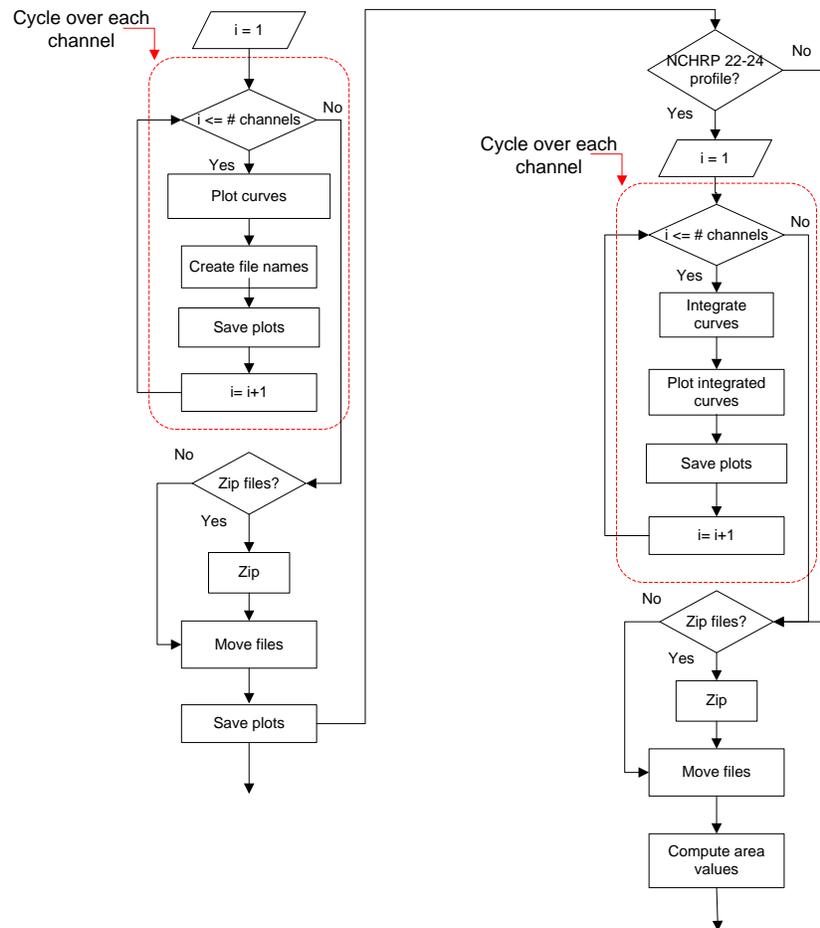


Figure B.29: Flow chart of the script *Whole\_plot\_curves* (sub-block C.3).

### B.4.6 Block D (Metrics evaluation)

Block D implements the computation of the comparison metrics and the corresponding postprocessing operations necessary to evaluate the weighted average of the metrics obtained from the single channels. As can be seen in Figure B.30, Block D is composed of two sub-blocks which handles the evaluation and postprocessing of metrics respectively on the whole time interval and the user defined time interval/s.

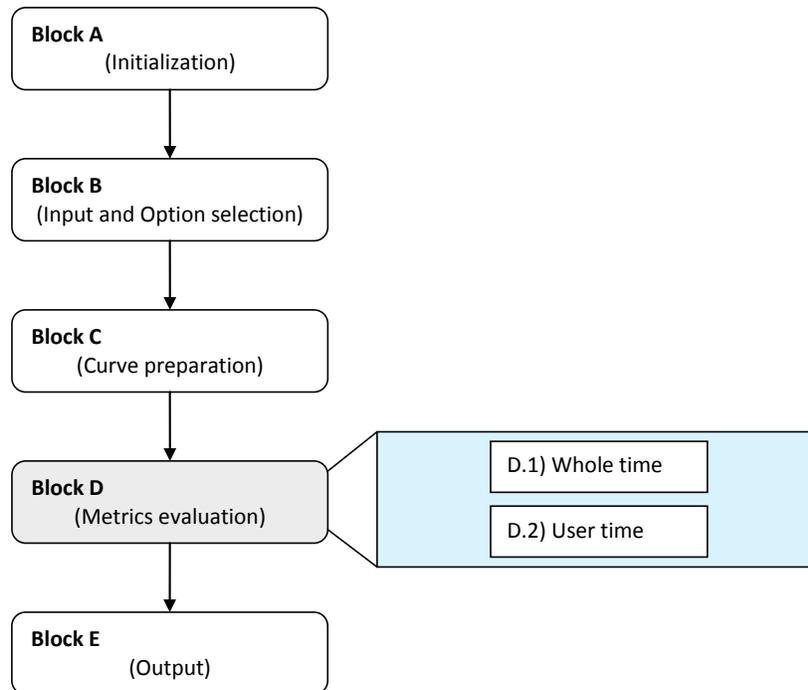


Figure B.30: Diagram of Block D.

The main structure of Block D is shown in Figure B.31. The sub-blocks Whole time (D.1) and User time (D.2) are invoked respectively in the case the curves are compared on the whole time or a user time interval. In particular, in case the whole time option is not selected, a void vector/matrix is created for the corresponding output in the Excel file.

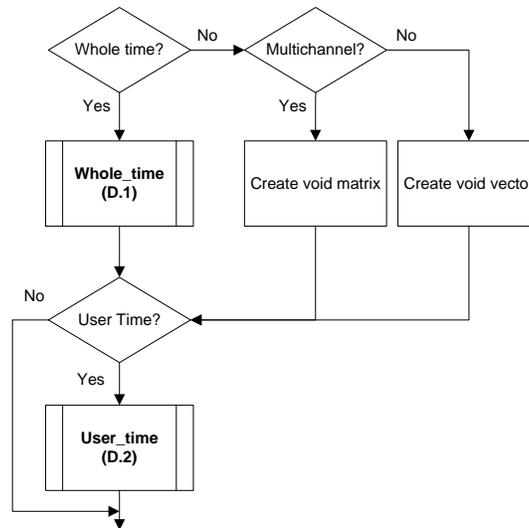


Figure B.31: Main structure Block D.

#### B.4.6.1 Whole time (D.1)

The sub-block *Whole time* calculates the metrics in the full time interval on which the curves are defined. The flow chart in Figure B.32 shows the general implementation of this sub-block.

If RSVVP is running in single channel mode, the weight assigned to the channel is the unity. In multichannel mode, at the beginning of its execution, the sub-block computes the weighting factors according to the specific method selected by the user. In case the resultants for the accelerations and rotational rates have been computed, each of them is assigned a weight equal to 50 percent of the total; otherwise, the weighting factors for each channel are computed based on the area of the true curves. In the latter case, the script *Weighting\_scheme\_whole* is invoked.

The calculation of the metrics is performed by invoking the script *Metrics\_evaluation* which is cycled over each of the input channels (the loop in the flowchart is indicated by the red rectangle). After the metrics have been computed for each channel, the sub-block D.1 also performs two main postprocessing operations:

- Computation of the weighted average (multiple channels)
- Creation of variables for the Excel output

The postprocessing is executed by the script *Whole\_time\_postprocessing*.

Eventually, the sub-block D.1 displays the results (metrics values and various graphs) through a GUI which is managed by the script *Table\_output\_whole*.

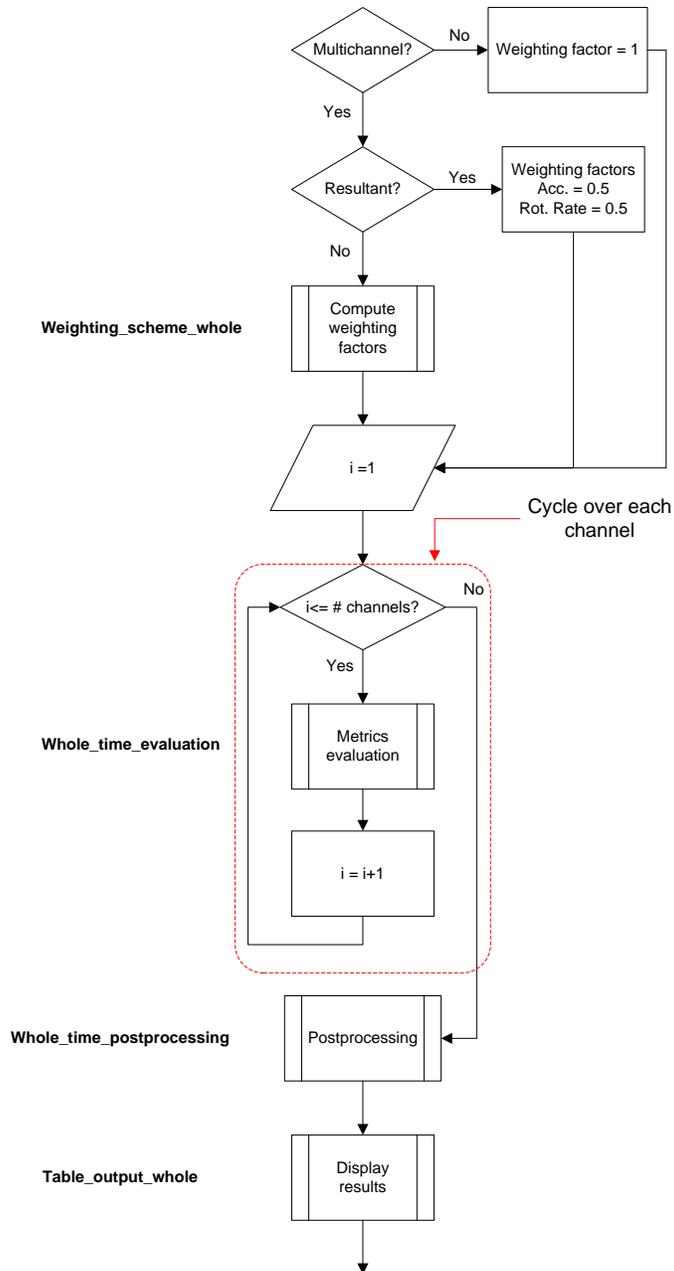
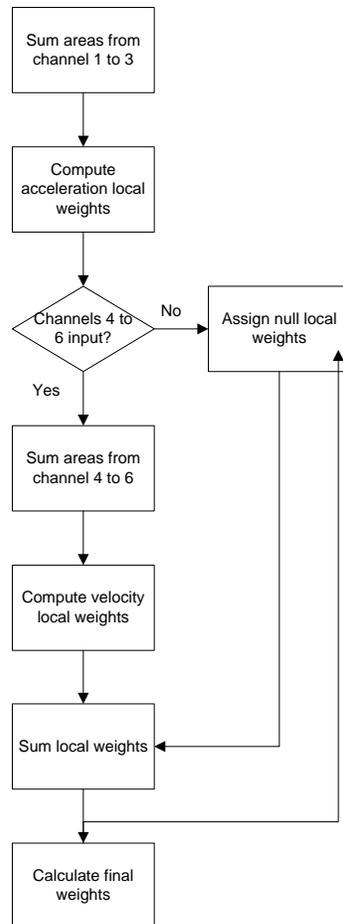


Figure B.32: Diagram of sub-block D.1 (Whole time).

B.4.6.1.1 *Weighting\_scheme\_whole*

This script calculates the weighting factors in case of multiple channels. The steps followed to compute the weighting factors are shown in the flowchart in Figure B.34.

The method is based on the computation of the area of the true curve for each channel and the areas of the accelerations and rotational rates are considered separately because, otherwise, the different units may lead to a disproportion in the evaluation of the weighting factors.



**Figure B.33: Flow chart of the script *Weighting\_scheme\_whole* (sub-block D.1).**

#### *B.4.6.1.2 Whole\_time\_evaluation*

This script manages the computation of the metrics according to the selection made by the user in the corresponding GUI. The variable used to store the metric flags is *Metrics*. The algorithm of this script is composed of a series of conditional blocks, one for each of the metrics implemented in the code. The generic flowchart of these blocks is shown in Figure B.34 (left).

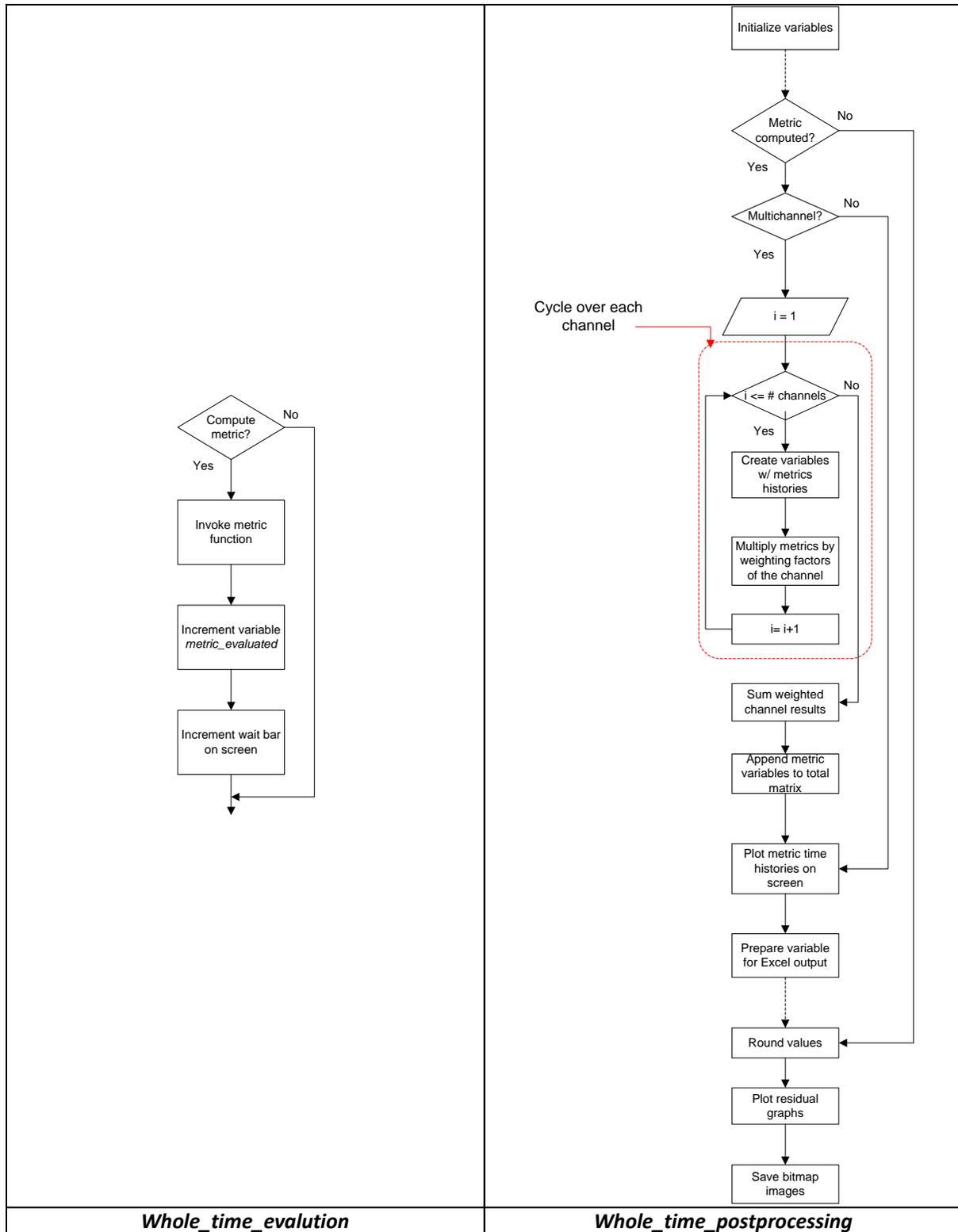


Figure B.34: Flow chart of the scripts *Whole\_time\_evaluation* (left) and *Whole\_time\_postprocessing* (right) (sub-block D.1).

If it has been selected, a metric is evaluated by invoking the corresponding function. Metric functions are programmed in Matlab and, generally, they receive as input the preprocessed curves corresponding to a specific channel and give as output one or more vectors containing the time history/ies of the metric. Once the metric values have been computed, a counter variable (*metric\_evaluated*) is incremented which then is used to add a bar to an on-screen waiting bar used to inform the user about how many metrics have been calculated/remain.

#### *B.4.6.1.3 Whole\_time\_postprocessing*

This script is the core of sub-block D.1. It manages the following four operations:

- calculation of the weighted average of metrics (multiple channels)
- creation of the variables containing the metrics values and time histories (for the output both on screen and using Excel files)
- round-off of the metrics values
- residuals plotting (time histories, histogram and cumulative distribution)

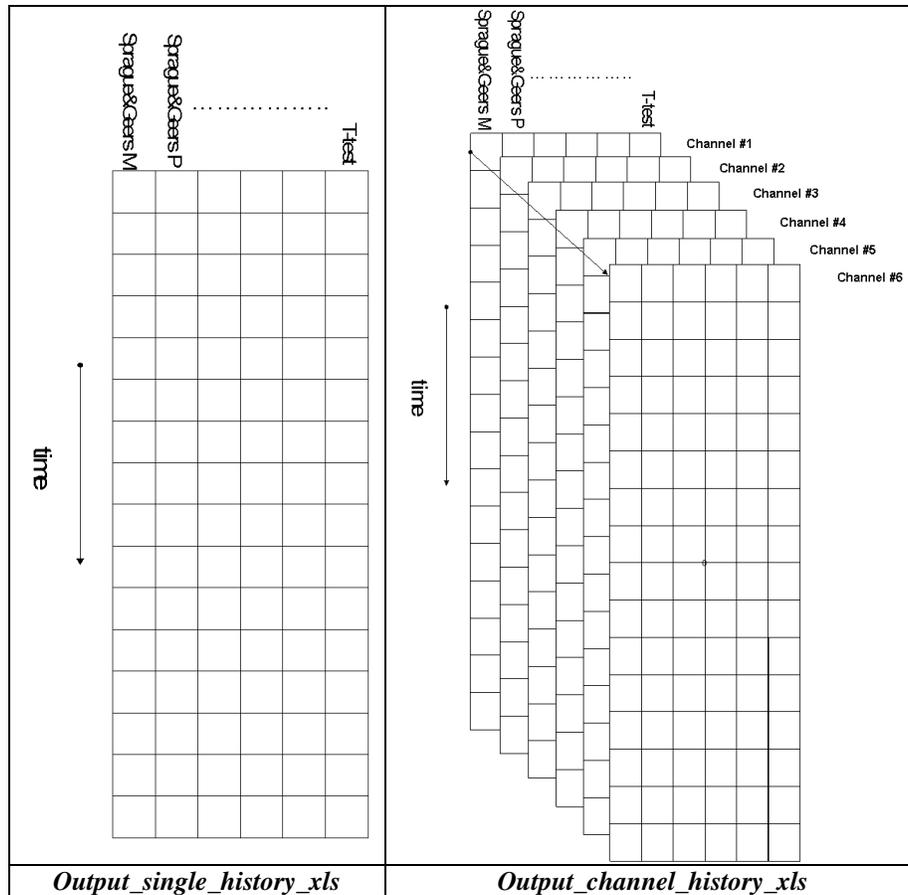
The flowchart of the whole script is shown in Figure B.34 (right). Similarly to the previous script, also the algorithm of this script is mainly composed of a series of conditional blocks, one for each of the metrics implemented in the code.

The generic structure of each of these conditional blocks is delimited in the dotted rectangle in the flowchart. In the whole flowchart, each block is related to a specific metric and is performed only if that metric has been computed. In case of multiple channels, a loop is defined which cycles over each of the input channels. For each iteration of the loop, a local variable is created which contains the metric values for that specific channel. Also, the values for each channel are then multiplied by the corresponding weighting factors previously computed by the script *Weighting\_scheme\_whole*. These weighted values are then summed up immediately after the loop ends in order to obtain a weighted average.

Once variables with metrics values and time histories have been created for a specific metric, they are appended to a corresponding global variable which cumulates the results for the various metrics computed during the run.

After the script has postprocessed all the metrics, it proceeds to round the values and plot the graphs of the residuals, histogram and cumulative distribution. Note that, during the rounding of





**Figure B.36: Data organization of the variables *Output\_single\_history\_xls* and *Output\_channel\_history\_xls*.**

*B.4.6.1.4 Table\_output\_whole*

This script is the last recalled by sub-block D.1 and contains commands to create a summary table with graphics and the values of the metrics. The summarizing table is a GUI managed by the function *Table\_Results\_NCHRP* or *Table\_Results* according to which type of metric profile has been selected at the beginning of the run.

**B.4.6.2 User time (D.2)**

The sub-block *User time* calculates the metrics on the time interval/s defined by the user during the execution of RSVVP. The flow chart in Figure B.37 shows the general implementation of this sub-block. The main characteristic of the algorithm for this sub-block is that it is implemented in a loop (indicated by the bigger red rectangle in the flow chart) in order to allow to cycle over an arbitrary number of user-defined time intervals.

Because the scripts used in the following sub-blocks have a structure similar to the corresponding scripts used in the preceding sub-block, D.1, the reader can refer to the description already given in the previous section. The only script which is described next in this section is *Store\_results*, as it is peculiar of sub-block D.2.

At the beginning of each iteration of the loop, the script *User\_time\_interval* is invoked which manages a GUI for the definition of time intervals and the curves are plotted on that interval by the script *User\_plot\_curves*. After the time interval has been defined, the weighting factors and the comparison metrics are calculated using two algorithms (respectively *Weighting\_scheme\_user* and *User\_time\_evaluation*) similar to those used to compute the metrics for each channel in sub-block D.1. Similarly to sub-block D.1, also for the user-defined time interval, after the metrics have been computed for each channel, they are postprocessed by the script *User\_time\_postprocessing*. The variables containing the final values of the metrics have the same structure of those earlier described

Also, the various results for each user-defined time interval (i.e., each iteration of the loop) are appended to the corresponding variables which are used in Block E to write the results in Excel format. In fact, as multiple user time intervals may be defined, at the end of each iteration the matrices containing the results for the specific time interval considered are stored before being rewritten for the next time interval (see the description of script *Store Results* for details).

Similarly to sub-block D.1, before concluding the iteration, the results for the specific time interval are displayed on the screen through a GUI which is managed by the script *Table\_output\_user*. In case the user decides to define a new time interval, the variable *Time\_interval* which is used to count the number of user-defined time intervals is incremented and a new iteration of the cycle starts.

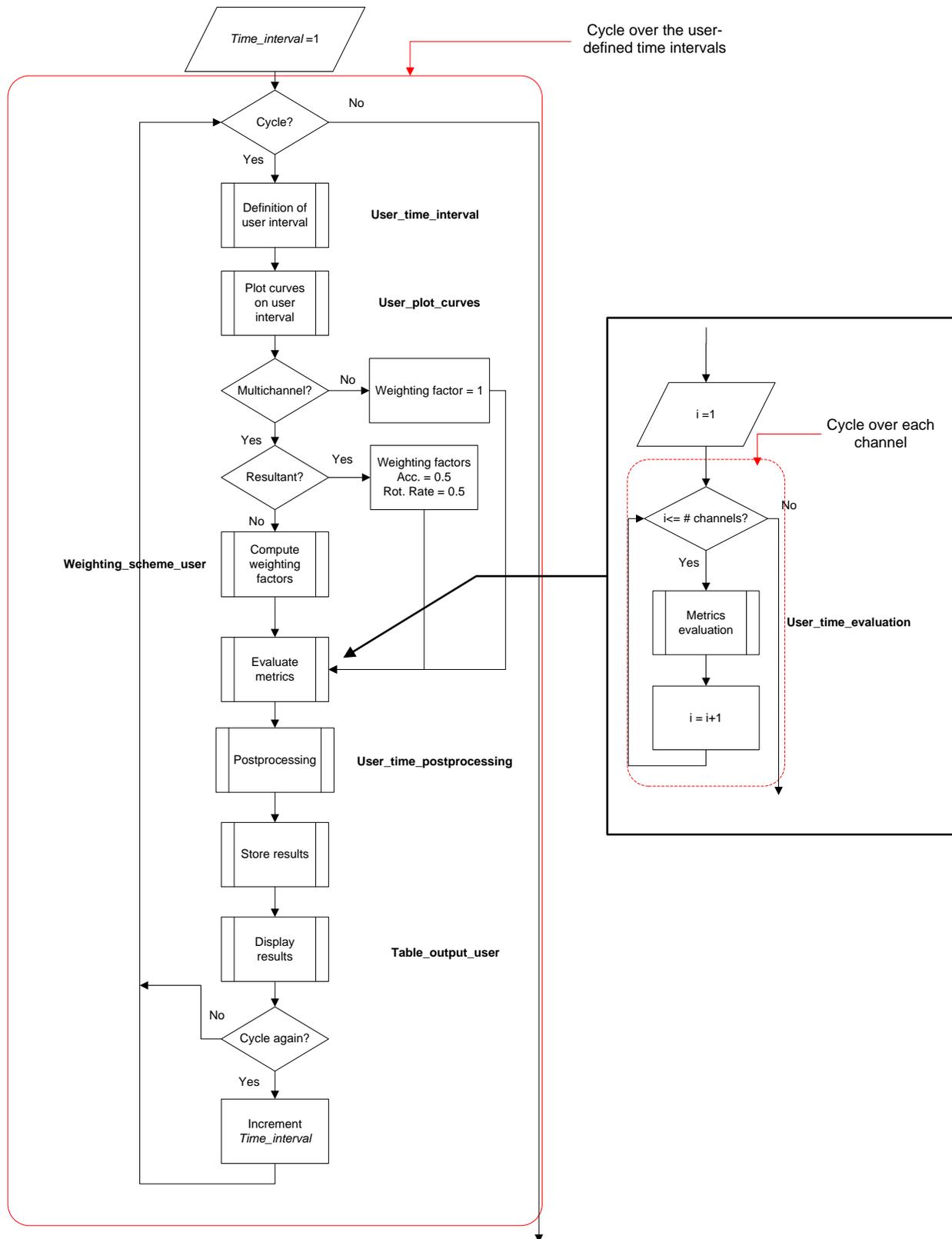


Figure B.37: Diagram of sub-block D.2 (User time).

#### B.4.6.2.1 Store Results

This script manages the storage of the results obtained for each iteration of the main loop of sub-block D.2 (i.e., for each user-defined time interval on which the comparison metrics are computed). In fact, because the various variable in which the script *User\_time\_postprocessing* saves the results are rewritten during each iteration, this script appends the results obtained for every specific time interval to the corresponding global variables.

Eventually, the following global variables are created which will be used by the following block (Block E:).

- *Time\_history\_channel\_xls\_user*
- *Time\_history\_single\_xls\_user*
- *Output\_xls*
- *Output\_channel\_xls*

The diagrams in Figure B.38 show the data organization of the variables *Output\_xls* and *Output\_channel\_xls*.

The column vector containing the final metrics values for the specific time interval (*Output\_xls\_user*) is appended to the end of the matrix *Output\_xls*, which has been previously created by sub-block D.1 and already contains the final results for the whole time interval. Note that, in case, the whole time interval has not been considered, the results will be appended to an originally null matrix. In case of multiple channels, the final metrics values for the actual time interval have been previously saved in the matrix *Output\_channel\_xls*. In this case, in a manner similar to that used to append the single/weighted results, the matrix containing the results for the current time interval is appended to the third dimension of the 3D matrix *Output\_channel\_xls*.

As for the matrices containing the time histories of the metrics for the single/weighted channel (*Output\_single\_history*) and each of the input channels (*Output\_channel\_history*), they are transformed in a single cell and appended to respectively the vector *Time\_history\_channel\_xls\_user* and *Time\_history\_single\_xls\_user*.

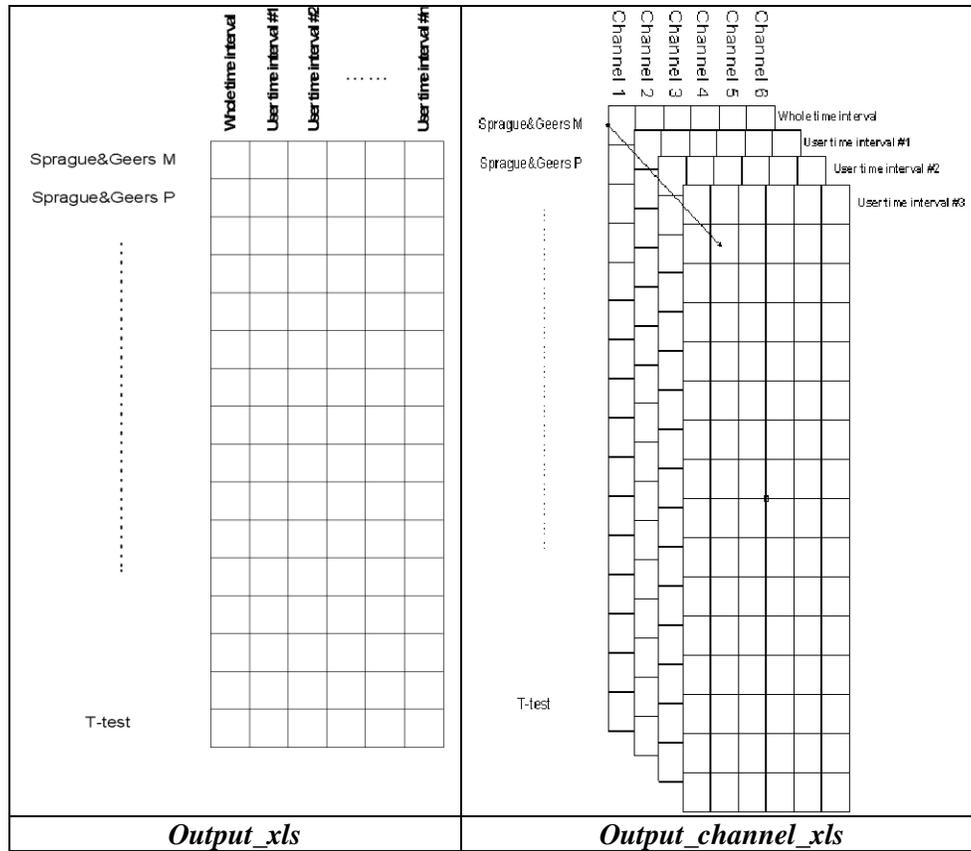
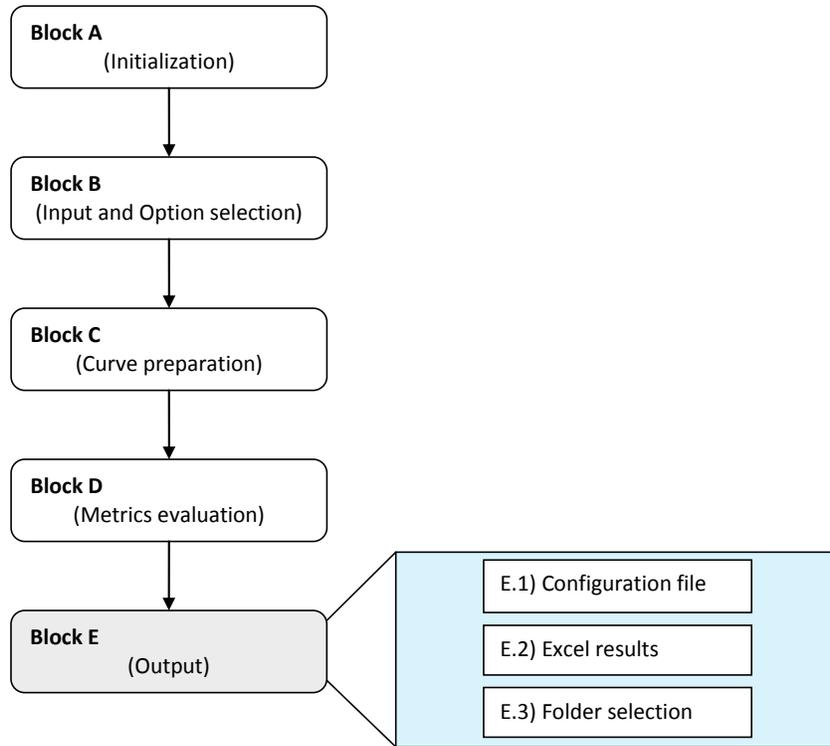


Figure B.38: Data organization of the variables *Output\_xls* and *Output\_channel\_xls*.

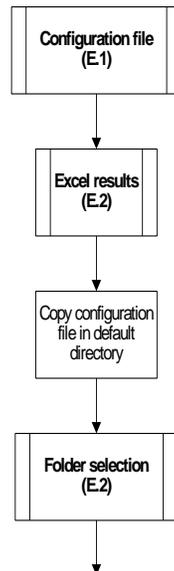
### B.4.7 Block E (Output)

Block E is the last block implemented in the program code. It handles the various results output by RSVVP. As can be seen in Figure B.39, this block is composed of three sub-blocks: Configuration file, Excel results and Folder selection.



**Figure B.39: Diagram of Block E.**

The main structure of Block E is shown in Figure B.40.



**Figure B.40: Main structure Block E.**

### B.4.7.1 Configuration file (E.1)

The sub-block *Configuration file* manages the possibility to update the configuration file with the information about any time interval defined by the user during the execution of RSVVP.

The flow chart in Figure B.37 shows the general implementation of this sub-block. The option to update the configuration file is given for any case the user has defined any time interval during the execution of the program. The algorithm skips the update only in two cases: (1) a previous configuration file has been loaded in total run mode and no new user intervals have been defined or (2) in case the user does not want to update the previous configuration file.

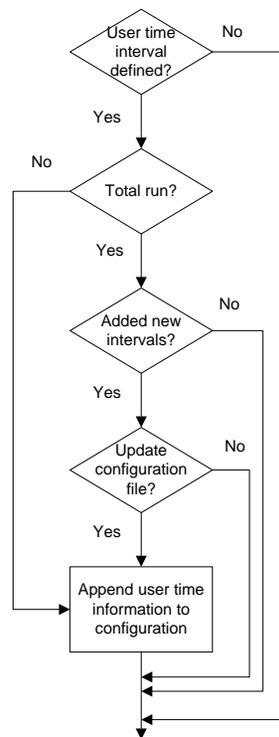
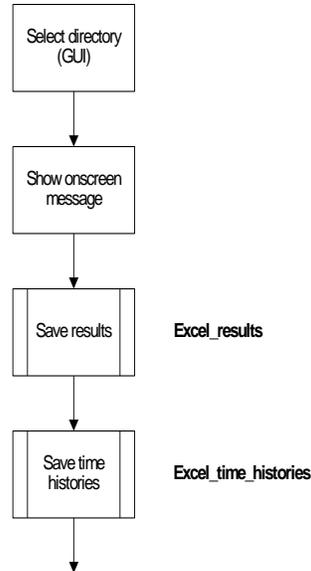


Figure B.41: Diagram of sub-block E.1 (Configuration file).

### B.4.7.2 Excel results (E.2)

This sub-block is the core of the Block E as it creates the Excel files containing respectively the metrics values and time histories.



**Figure B.42: Diagram of sub-block E.2 (Excel results).**

As can be seen from the algorithm shown in Figure B.42, after the GUI for the selection of the destination folder where to save all the results, the code performs the scripts *Excel\_results* and *Excel\_time\_histories*.

#### *B.4.7.2.1 Excel\_results*

This script manages the creation of the Excel file containing the metrics values for the various time intervals considered during the run of RSVVP. The flowchart of this script is shown in Figure B.43. In case of multiple channels, the algorithm also cycles over the total number of input channels and creates specific sheets containing the results for each of them. The writing of the results for the MPC, Single-value and ANOVA metrics is managed by separate scripts, respectively *Excel\_results\_MPC*, *Excel\_results\_Single* and *Excel\_results\_ANOVA*. Note that, in case the NCHRP 22-24 profile has been selected, the single metrics are not computed and, hence, the corresponding script is skipped as they do not appear in the Excel file.

#### *B.4.7.2.2 Excel\_results\_MPC*

Figure B.44 shows the flowchart of the script *Excel\_results\_MPC*. The first step performed by this script is to create the headers for each of the time interval considered during the run of the program and write them in the main sheet of the Excel file. In case of multiple channels, a new sheet is written for each channel and the same headers are written also for any of the channel sheets. This operation is implemented in a loop which cycles over the number of input channels.

As this script is focused on managing and writing only the values for the MPC metrics, after the headers have been created, the script proceeds to extracting the metrics values from the variable *Output\_xls* which has been created during the postprocessing in Block D (for details about the structure of this variable refer to section B.4.6.1). The values corresponding to the MPC metrics are extracted by considering only a certain range of rows of the matrix *Output\_xls*. Note that, according to the metric profile chosen, the number of computed metrics varies and, hence, also the corresponding number of rows extracted. In case of multiple channels, a similar extraction procedure is applied on the 3-D matrix *Output\_channel\_xls* in order to extract only the desired metrics values. Also, in this particular case, the results for each channel are extracted by cycling over their total number. See Figure B.45 for a graphical visualization of the extraction procedure.

The other two scripts which manage the writing of the results for the single-value and the ANOVA metrics (*Excel\_results\_Single* and *Excel\_results\_ANOVA*) are implemented in a similar manner and, hence, the corresponding flowcharts are not shown for the sake of conciseness.

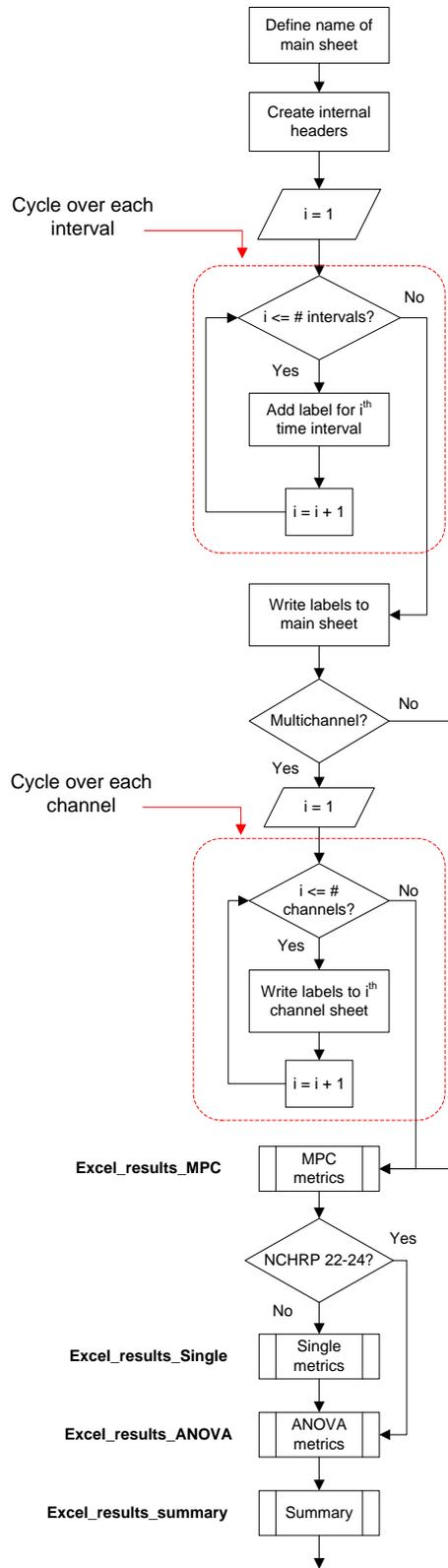


Figure B.43: Flow chart of the script *Excel\_results* (sub-block E.2).

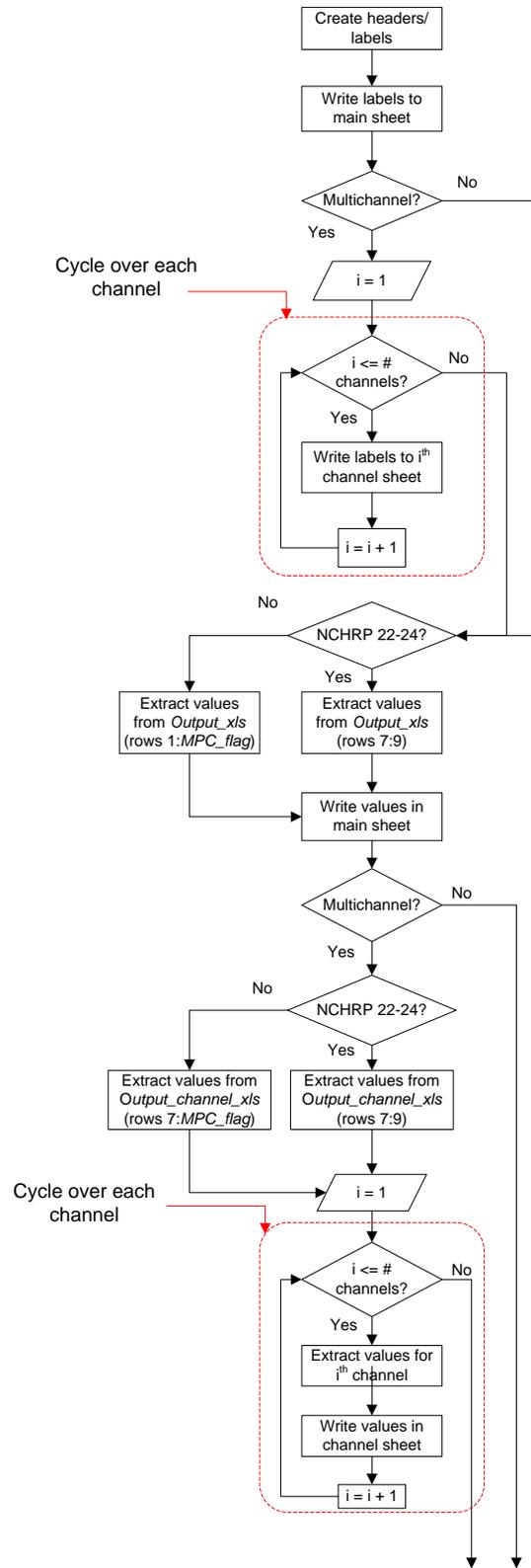


Figure B.44: Flow chart of the script *Excel\_results\_MPC* (recalled by the script *Excel\_results*).

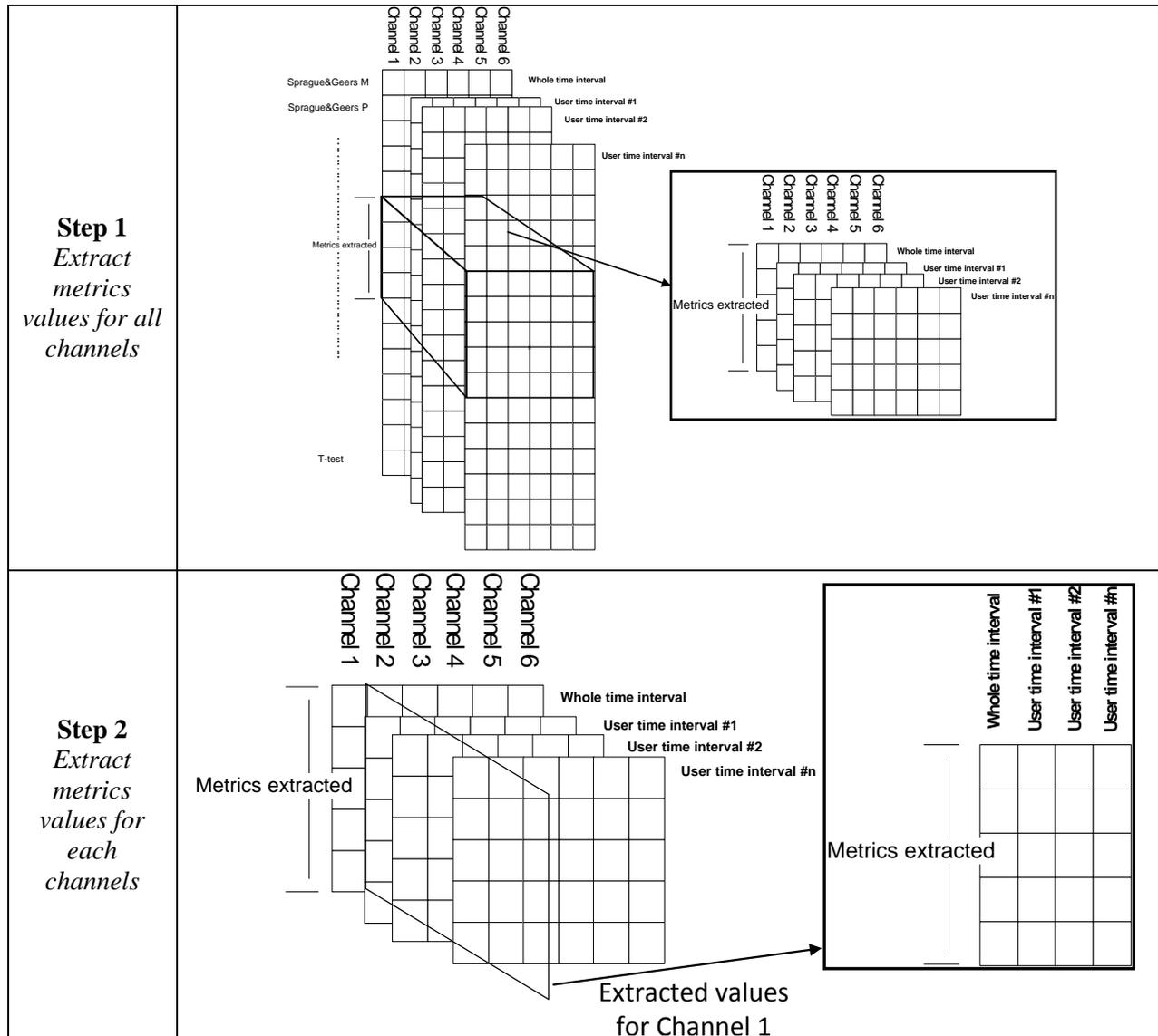


Figure B.45: Data extraction from the variable *Output\_channel.xls*.

B.4.7.2.3 *Excel\_time\_histories*

This script manages the creation of Excel files containing the time histories for the metrics computed by RSVVP. Figure B.46 shows the algorithm of the script. The first step performed is the selection of the name to be given to the sheet containing accordingly the metrics time histories for either the single channel or the weighted mean from the multiple channels. The algorithm then creates an Excel file containing the results for the comparison between the curves on the whole time interval and/or separate Excel files for each of the user-defined time intervals, depending on which selection was made during the run of the program. In the latter case, a loop cycles over the number of time intervals defined by the user and creates an Excel file during each

iteration. Also, if multiple channels were input, in either the case the comparison was performed on the whole time interval or user defined time interval/s the algorithm cycles over each of the input channels in order to save them in separate sheets of the same excel file.

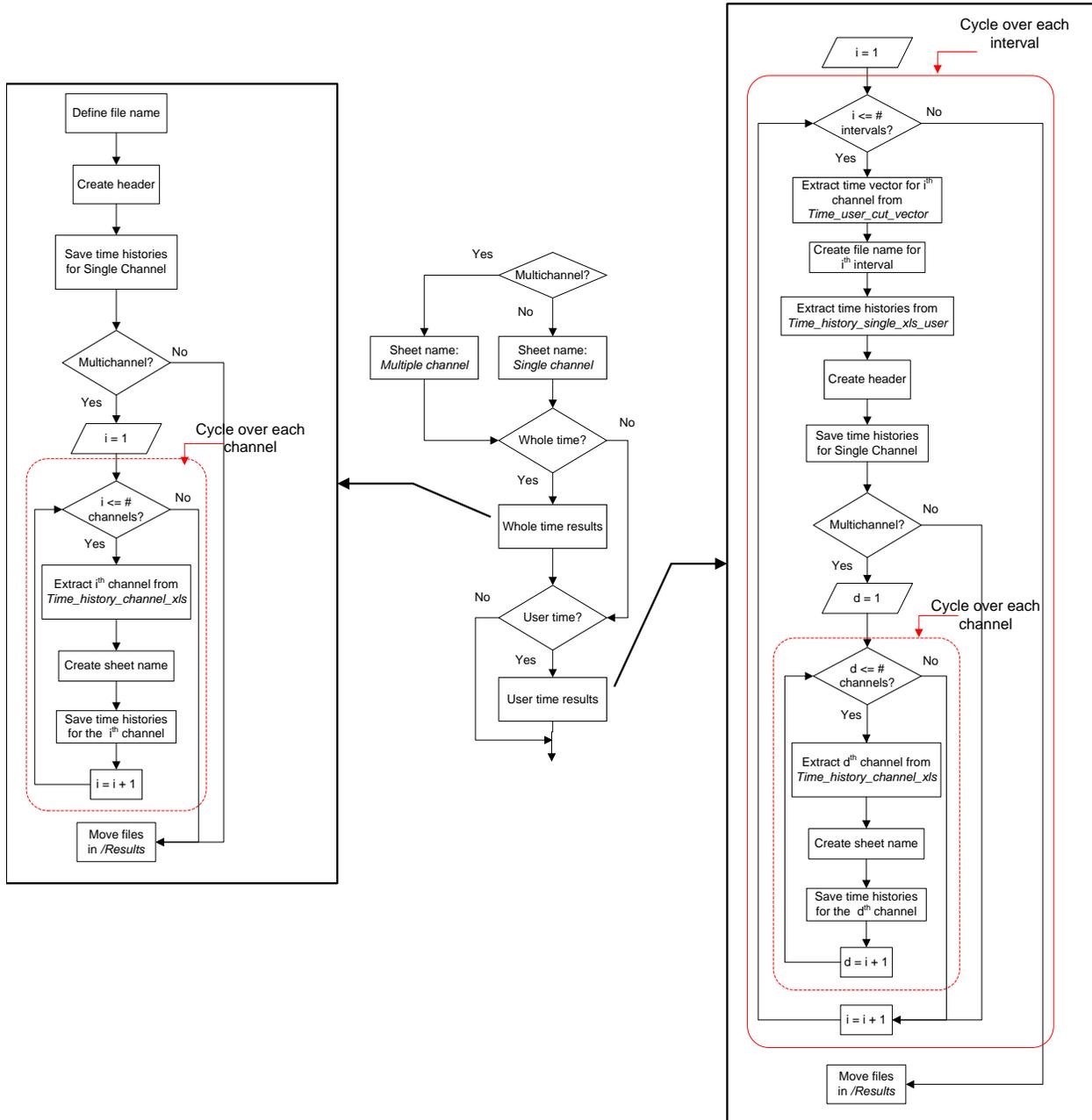
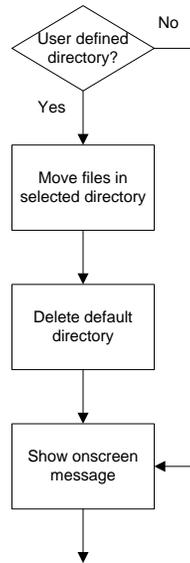


Figure B.46: Flow chart of the script *Excel\_time\_histories* (sub-block E.2).

### B.4.7.3 Folder selection (E.3)

This sub-block (Figure B.47) checks if the user defined a directory where to save the output files and, in case, moves all the previously created files from the default directory */Results\_XX* to that folder. In any case, at the end, a message is shown on the screen indicating to the user that the results have been saved in the selected directory.



**Figure B.47: Diagram of sub-block E.3 (Folder selection).**

**REFERENCES**

- [1] MathWorks, “*MATLAB® User Guide – high performance numeric computation and visualization Software*”, The MathWorks Inc., 3 Apple Hill Drive, Natick, MA, USA, 2008.
- [2] Society of Automotive Engineers, “SAE J211-1 (R) Instrumentation for Impact Test—Part 1—Electronic Instrumentation. SAE International”, Warrendale PA, USA, Jul 1, 2007.
- [3] H.E. Ross, D.L. Sicking, R.A. Zimmer, and J.D. Michie, “*Recommended Procedures for the Safety Performance Evaluation of Highway Features*”, National Cooperative Highway Research Program (NCHRP) Report 350, Transportation Research Board, Washington, D.C., 1993.
- [4] European Committee of Standardization, “European Standard EN 1317-1 and EN 1317-2: Road Restraint Systems”, CEN (1998).
- [5] T.L. Geers, “An Objective Error Measure for the Comparison of Calculated and Measured Transient Response Histories”, *The Shock and Vibration Bulletin*, The Shock and Vibration Information Center, Naval Research Laboratory, Washington, D.C., Bulletin 54, Part 2, pp. 99-107, (June 1984).
- [6] *Comparative Shock Analysis (CSA) of Main Propulsion Unit (MPU), Validation and Shock Approval Plan*, SEAWOLF Program: Contract No. N00024-90-C-2901, 9200/SER: 03/039, September 20, 1994.
- [7] M.A. Sprague and T.L. Geers, “Spectral elements and field separation for an acoustic fluid subject to cavitation”, *J Comput. Phys.*, pp. 184:149, Vol. 162, (2003).
- [8] L.E. Schwer, “Validation Metrics for Response Time Histories: Perspective and Case Studies”, *Engng. with Computers*, Vol. 23, Issue 4, pp. 295–309, (2007).
- [9] D.M. Russell, “Error Measures for Comparing Transient Data: Part I: Development of a Comprehensive Error Measure”, *Proceedings of the 68th shock and vibration symposium*, pp. 175–184, (2006).
- [10] J. Cohen, P. Cohen, S.G. West and L.S. Aiken, *Applied multiple regression/correlation analysis for the behavioral sciences*, Hillsdale, NJ: Lawrence Erlbaum, (3rd ed.), 2003.
- [11] S. Basu and A. Haghghi, “*Numerical Analysis of Roadside Design (NARD) vol. III: Validation Procedure Manual*”, Report No. FHWA-RD-88-213, Federal Highway Administration, Virginia, 1988.
- [12] D. Twisk and A. Ritmeijer, “A Software for Demonstrating Validation of Computer Dummy Models Used in the Evaluation of Aircraft Seating Systems”, SAE Paper No. 2007-01-3925, Society of Automotive Engineers, Warrendale, PA, 2007.
- [13] B. Whang, W.E. Gilbert and S. Zilliagus, *Two Visually Meaningful Correlation Measures for Comparing Calculated and Measured Response Histories*, Carderock Division, Naval Surface Warfare Center, Bethesda, Maryland, Survivability, Structures and Materials Directorate, Research and Development Report, CARDEROCKDIV-U-SSM-67-93/15, September, 1993.
- [14] H. Theil, *Economic Forecasts and Policy*, North-Holland Publishing Company, Amsterdam, 1975.

- [15] M.H. Ray, “Repeatability of Full-Scale Crash Tests and a Criteria for Validating Finite Element Simulations”, *Transportation Research Record*, Vol. 1528, pp. 155-160, 1996.
- [16] W.L. Oberkampf and M.F. Barone, “Measures of Agreement Between Computation and Experiment: Validation Metrics,” *Journal of Computational Physics* Vol. 217, No. 1 (Special issue: Uncertainty quantification in simulation science) pp 5–36, 2006.

## Appendix B.I Code verification

The implementation of the following main features of RSVVP has been verified:

- Sprague&Geers metrics
- Knowles&Gear metrics

In order to verify the correct implementation of the Sprague & Geers metric, a comparison of ideal analytical wave forms differing only in magnitude or phase was performed and the results were compared with the outcomes obtained by Schwer [8] using the same benchmark curves.

The baseline analytical wave form was given by the following decayed sinusoidal curve:

$$m(t) = e^{-(t-\tau)} \sin 2\pi(t - \tau) \quad (\text{A1})$$

where the parameter  $\tau$  was used to create a phase in time or “time of arrival”.

Following Schwer’s work, two different tests were performed, considering respectively as the test function: (a) a wave form with the same time of arrival of (A1), but an amplitude 20% greater than the original wave form (*magnitude-error test*) and (b) a wave form with the same amplitude of (A1) but a time of arrival such that the phase was about  $\pm 20\%$  respect to the original wave form (*phase- error test*). The analytical forms used for the magnitude-error test were:

$$\begin{cases} m(t) = e^{-(t-0.14)} \sin 2\pi(t - 0.14) \\ c(t) = 1.2 \cdot e^{-(t-0.14)} \sin 2\pi(t - 0.14) \end{cases} \quad (\text{A2})$$

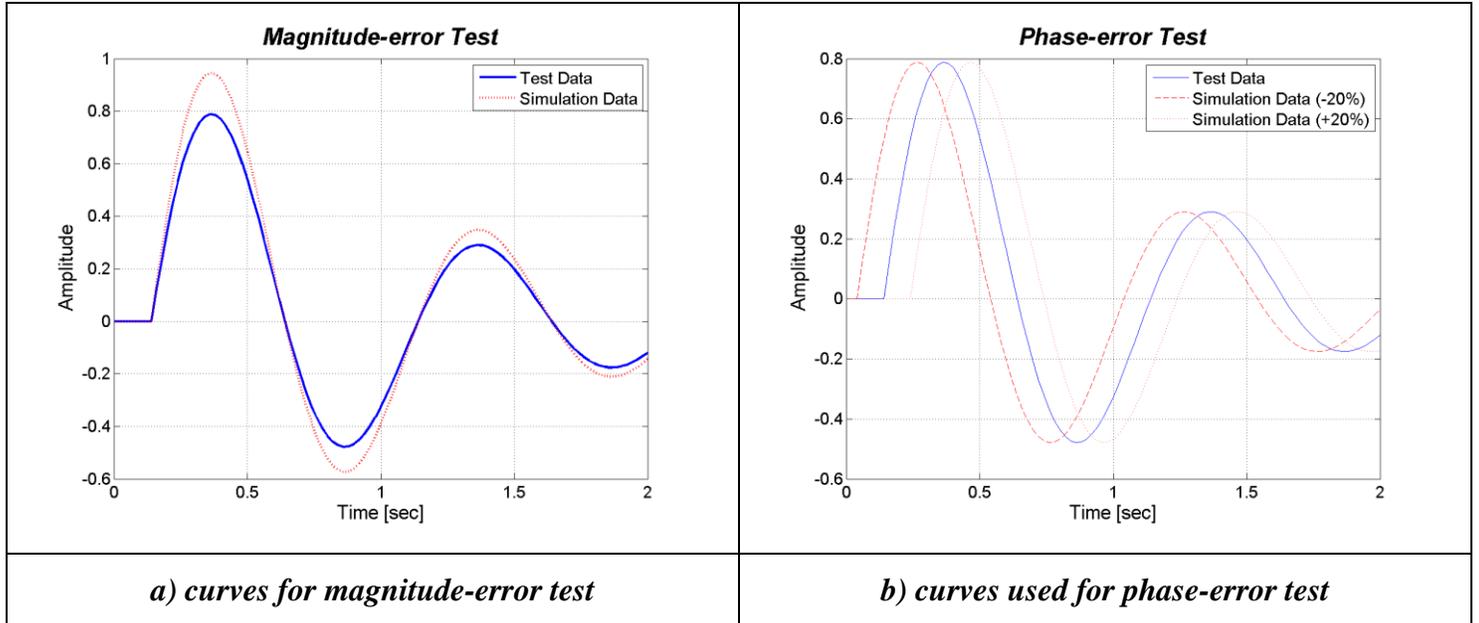
while the analytical wave forms used for the phase-error test were defined as:

$$\begin{cases} m(t) = e^{-(t-0.14)} \sin 2\pi(t - 0.14) \\ c(t) = e^{-(t-0.04)} \sin 2\pi(t - 0.04) \end{cases} \quad (\text{A3})$$

and,

$$\begin{cases} m(t) = e^{-(t-0.14)} \sin 2\pi(t - 0.14) \\ c(t) = e^{-(t-0.24)} \sin 2\pi(t - 0.24) \end{cases} \quad (\text{A4})$$

In both cases, the sampling period was  $\Delta t = 0.02 \text{sec}$  and  $0 \text{sec} \leq t \leq 2 \text{sec}$ . Figure B.48 shows the graphs of the analytical curves used respectively for the magnitude error and the phase error tests.



**Figure B.48: Idealized time histories used for the case of (a) 20% magnitude and (b) time of arrival.**

As in this case, the phase shift between the baseline and the test curves was a wanted characteristic, the metrics was applied without synchronizing the two time histories in the preprocessing phase. The results obtained using the Sprague & Geers metric implemented in RSVVP for the difference in magnitude and the difference in time of arrival are shown respectively in Figure B.49 and Figure B.50. Table B. 4 shows the value of the metric component obtained considering a time interval equal to the total length of the time histories. These values match completely with those obtained by Schwer, thus confirming the correct implementation of this metric.

**Table B. 4: Value of the Sprague & Geers metric components calculated using the V&V program.**

<i>Metric component</i>	<i>20 % Magnitude difference</i>	<i>Phase difference</i>	
		<i>+20%</i>	<i>-20%</i>
Magnitude	0.2	$\approx 0$	$\approx 0$
Phase	0	0.195	0.195
Combined	0.2	0.195	0.195

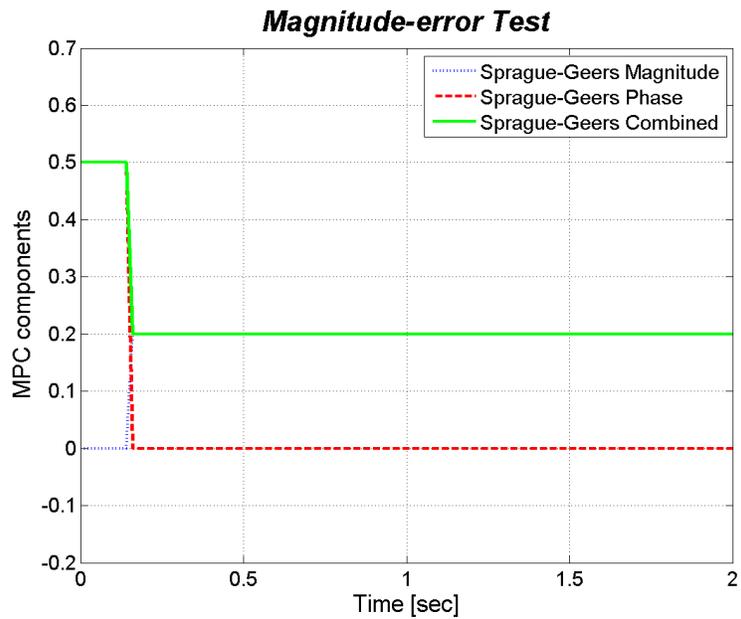


Figure B.49: Sprague-Geers component metric vs. time for the magnitude-difference test.

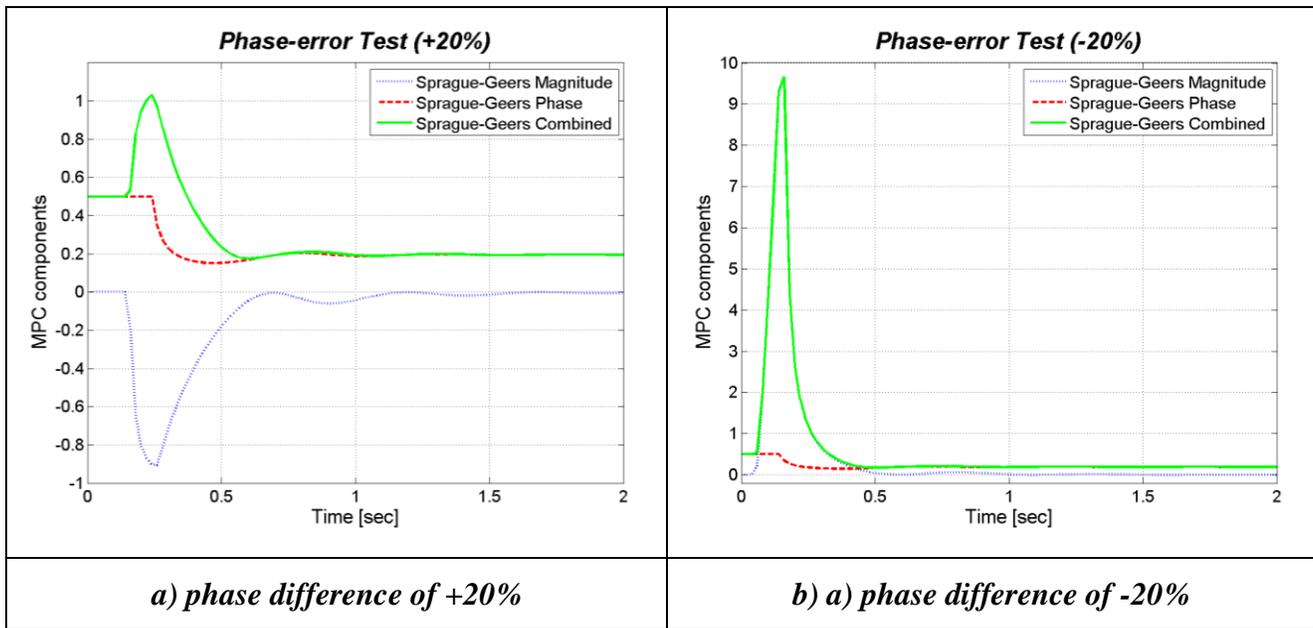


Figure B.50: Sprague-Geers component metric vs. time for the phase-difference tests: (a) +20% and (b)-20%.

## Appendix B.II Compiling RSVVP as a standalone application

The Matlab code of RSVVP can be compiled as a standalone executable. This allows users who do not have Matlab installed on their machine to be able to run RSVVP.

The standalone application of RSVVP is compiled using Matlab Compiler, a Matlab toolbox which requires a specific license. In order to create an executable version of RSVVP, set the current directory of Matlab to the folder where the RSVVP files are located and launch the following command:

```
mcc -m -v RSVVP
```

Note 1: In order to compile the standalone application, it is first necessary to indicate to Matlab which compiler to use by launching the following command:

```
mbuild -setup
```

The 32-bit Windows version of Matlab has a built-in C compiler called 'Lcc-win32'. In case other third-part C/C++ compilers are installed on the machine, it is also possible to select any of them.

Note 2: the previous versions of Matlab Compiler allowed to create a standalone application only from functions and it was not possible to directly compile scripts. As the principal file of RSVVP (*Main.m*) is a script, in order to compile the first versions of the program, the code was originally invoked from a function called RSVVP. While this trick allowed to compile RSVVP, the side effect was that it required to save to the Matlab *base* space some few local variables created in specific functions of the code. This was necessary in order to permit a sharing of these specific local variables among different functions. Although the latest versions of Matlab Compiler can now directly compile a script, making the conversion of these local variables into global variables no more necessary, these conversions may still appear in some part of the code.

### Appendix B.III Type of variables used in the code

The main typologies of variables available in Matlab and used in the implementation of the code of RSVVP are:

- Matrices and arrays (floating-point/integer data, characters and strings)
- Structures
- Cell arrays

**Matrices and arrays** are used to store both numbers and text characters. These data type includes also scalar numbers or single characters which are considered to be a 1x1 matrix. Numbers can be either stored as floating-point or integer data.

Structures and cell arrays provide a way to store dissimilar types of data in the same array. A Matlab **structure** is a data type that provides the means to store selected data together in a single entity. A structure consists mainly of data containers, called fields, and each of these fields stores an array of some Matlab data type. Figure B.51 shows an example of structure variable, *s*, which has three fields: *a*, *b*, and *c*.

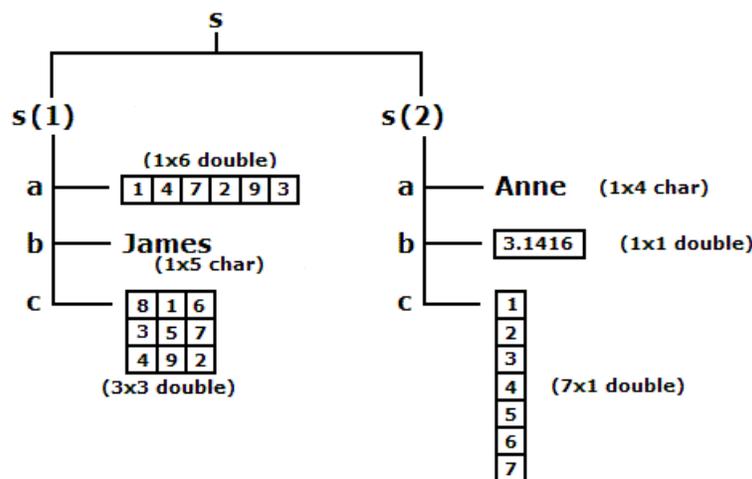
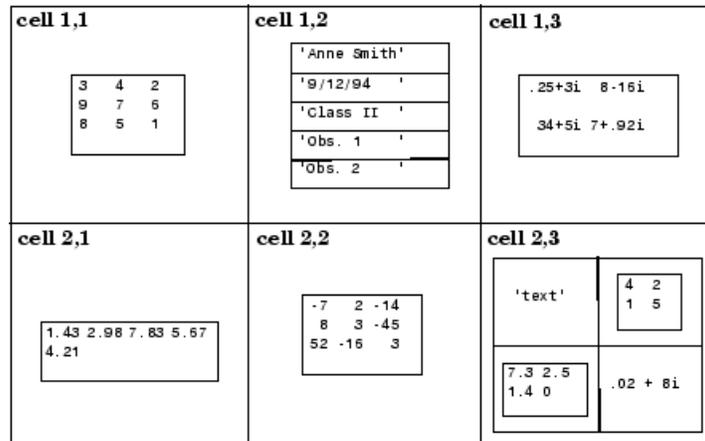


Figure B.51: Example of a Matlab structure variable [1].

A Matlab **cell array** is a collection of containers called cells in which different types of data can be stored. As an example, Figure B.52 shows a 2-by-3 cell array in which the cells in row one hold an array of unsigned integers, an array of strings, and an array of complex numbers, while

row two holds three other types of arrays, the last being a second cell array nested in the outer one.



**Figure B.52: Example of a Matlab structure variable [1].**

Structure-type variables are used to store in an organized manner the several options selected by the user in the various graphical interfaces while cell-type variables are used to conveniently store in a single variable all the data vectors (or matrices) which are characterized by different dimensions and would have otherwise required singular specific array (or matrix) variables.

## Appendix B.IV      Preprocessing algorithms

This appendix describes the general algorithms used to perform the following preprocessing operations:

- Filtering
- Shift/drift
- Resampling & trimming
- Synchronization

### Filtering

The filter process is implemented in the function *sae\_filter*, whose algorithm is shown in Figure B.53. The function receives as input the following three variables: (i) *CFC*, (ii) *T* and (iii) *X*. *CFC* and *T* are scalar variables containing respectively the value of the filter class and the sampling period of the input curves, while *X* is a vector containing the data points of the curve. The algorithm assumes that the sampling period is constant and, hence, the time vector is not necessary.

Before filtering the data a head and a tail vectors are created by repeating respectively the first and last value of the vector containing the data points. These vectors are then appended at the beginning and end of the vector containing the original data, correspondingly. The modified vector is then filtered by applying equation (1) a first time frontward and a second time backward in order to have a two-pass filter. The filter coefficients are calculated using the formulas indicated in equations (2) through (6) described in section B.3.2.3 of this manual.

Once the data are filtered, the added head and tail are trimmed and the filtered vector *Y* is given as the function output.

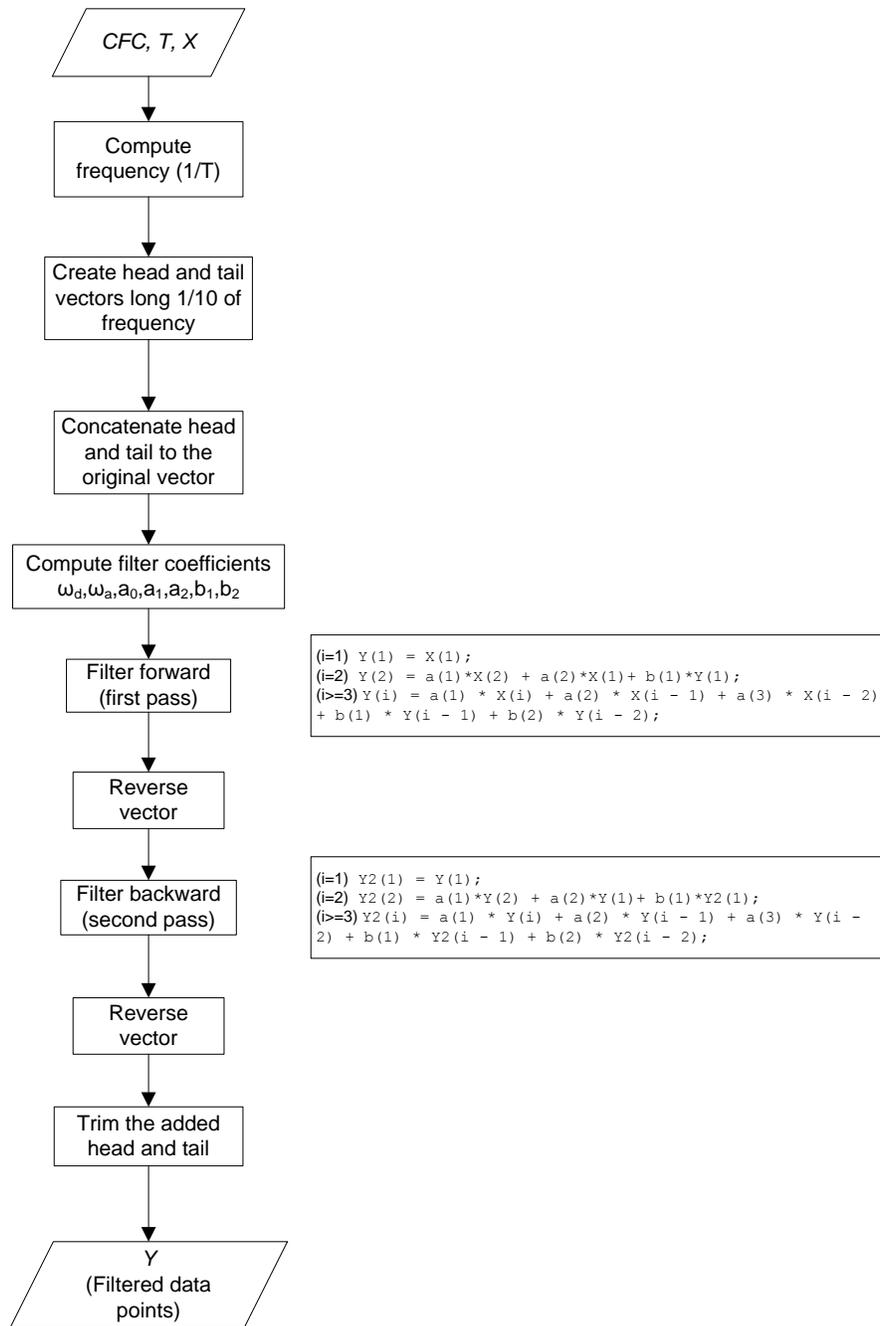
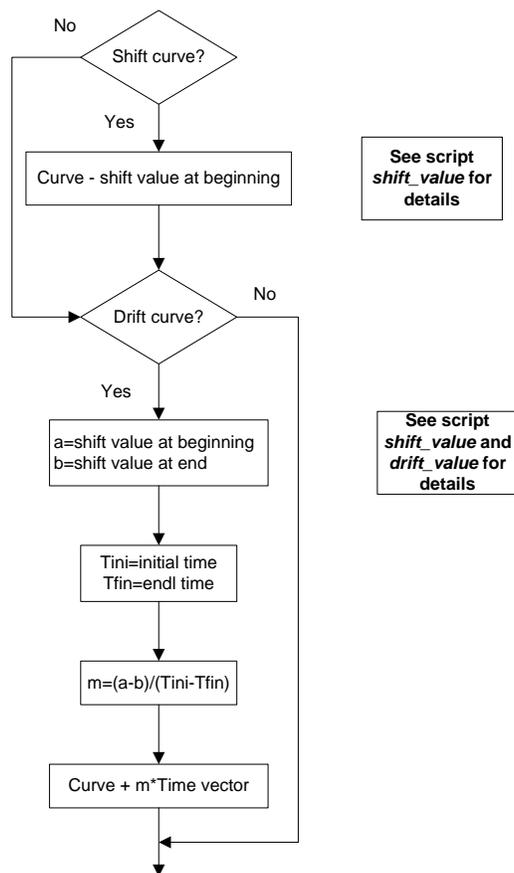


Figure B.53: Algorithm of the SAE filtering.

## Shift/drift

The shift and drift corrections are implemented in the script *Shift\_drift*. The main steps followed to perform this preprocessing task are shown in the algorithm in Figure B.54. The steps described in the algorithm are performed on either the true or test curves or both of them, according to the user selection.



**Figure B.54: Main algorithm of the script *Shift\_drift*.**

The shift values at the beginning and the end of the curve are computed respectively by the user-defined functions *shit\_value* and *drift\_value* (Figure B.55). The former function evaluates the vertical shift at the beginning of the curve, while the *drift* function at the end of the curve. The algorithms of both the two functions are very similar; the shift values are computed as the mean of respectively the initial and final portion of the data vector till/from 5 percent of the peak value. Also, if the standard deviation of the sub-vector is greater than 0.1, the algorithm iteratively decreases the size till the standard deviation drops within this value.

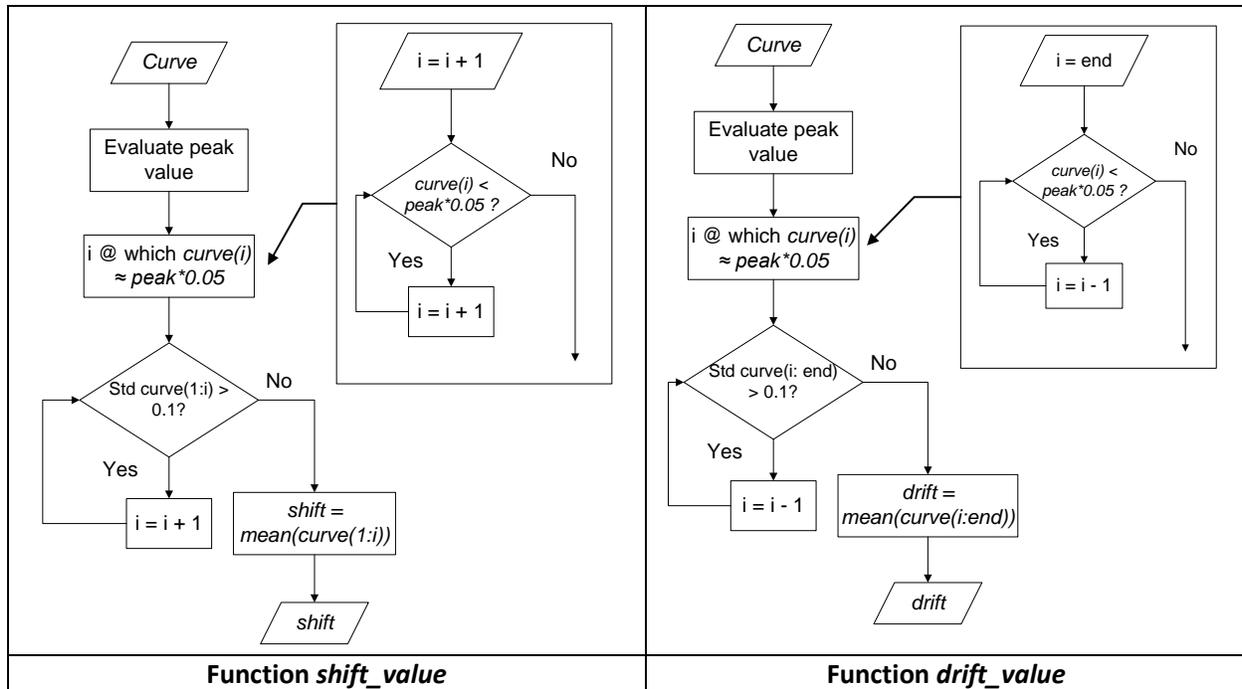


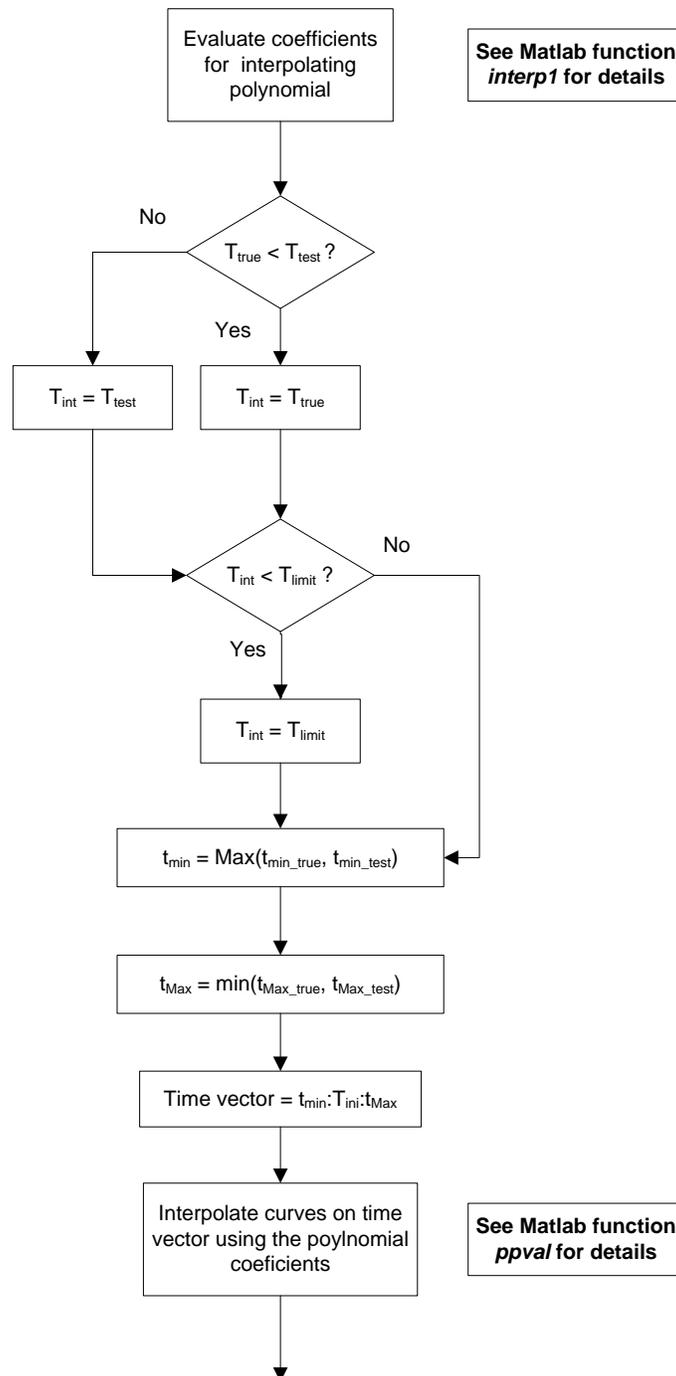
Figure B.55: Algorithm of the user-defined functions *shift\_value* (left) and *drift\_value* (right).

## Resampling & trimming

The resampling and trimming operations are implemented simultaneously in the script *Resampling\_trimming* whose algorithm is shown in Figure B.56. In fact, the curves are resampled on a trimmed time interval which considers respectively the maximum between the initial time values of the two curves and the minimum of the final time values. The Resampling is performed using a linear polynomial whose coefficients are evaluated using the predefined Matlab function *interp1*.

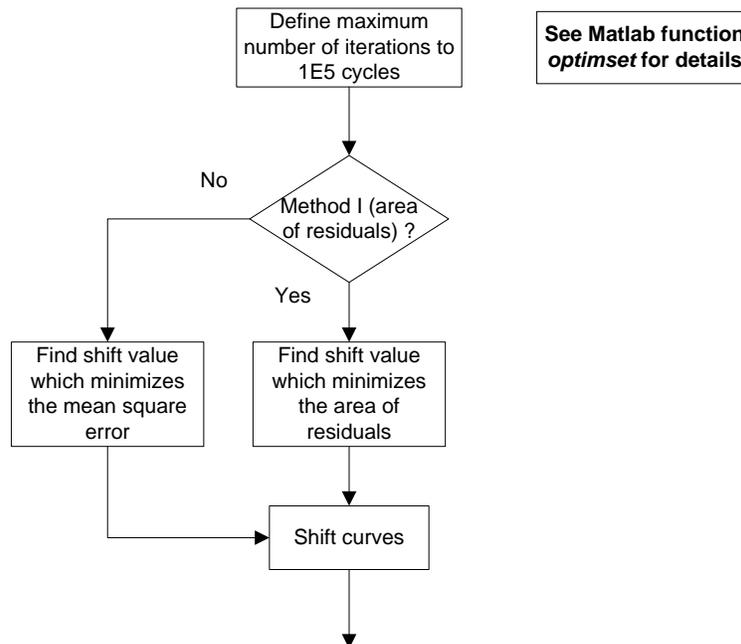
The sampling period (i.e., the minimum time between the collection of two consecutive data points) is defined as the minimum between that of the two curves and, in any case, it cannot be smaller than a minimum limit value (defined by the user at the beginning of the calculation). The minimum sampling period is determined by considering the first input channel; the same period is then used to resample also any of the other channels.

Eventually, both the true and test curves are interpolated on the trimmed time interval using their respective polynomial coefficients through the Matlab function *ppval*. In case of multiple channels, the same procedure is repeated for each of them.

Figure B.56: Algorithm of the script *Resampling\_trimming*.

### Synchronization

The automatic synchronization of the curves is implemented in the script *Curve\_synchronizing* whose algorithm is shown in Figure B.57. This script calculates the shift value which minimizes a target function, which can be either the area between the two curves (area of the residuals method) or the square error (lest square error method). The two target functions are recalled by user-defined Matlab functions, respectively *area\_res* and *rse*. In both the two functions, the couple of curves can be shifted respect to each other by an arbitrary value  $s$ . The minimization process is performed using a Matlab function (*fminsearch*) which iteratively calculates the values of the selected target function (area of residual or square-root error) varying the value of the shift variable in order to find the optimal solution. Eventually, the algorithm shifts the two input curves by the optimal value obtained from the minimization process.



**Figure B.57: Algorithm of the script *Curve\_synchronizing*.**

Both the user-defined functions *are-res* and *sre* require as input only the value by which to shift the two curves before calculating respectively the area between them or the square-root error. The algorithms of these two functions are shown in Figure B.58. For either of these two functions, the shifting of the two curves is performed by invoking the user-defined function *shift* whose algorithm is shown in Figure B.59.

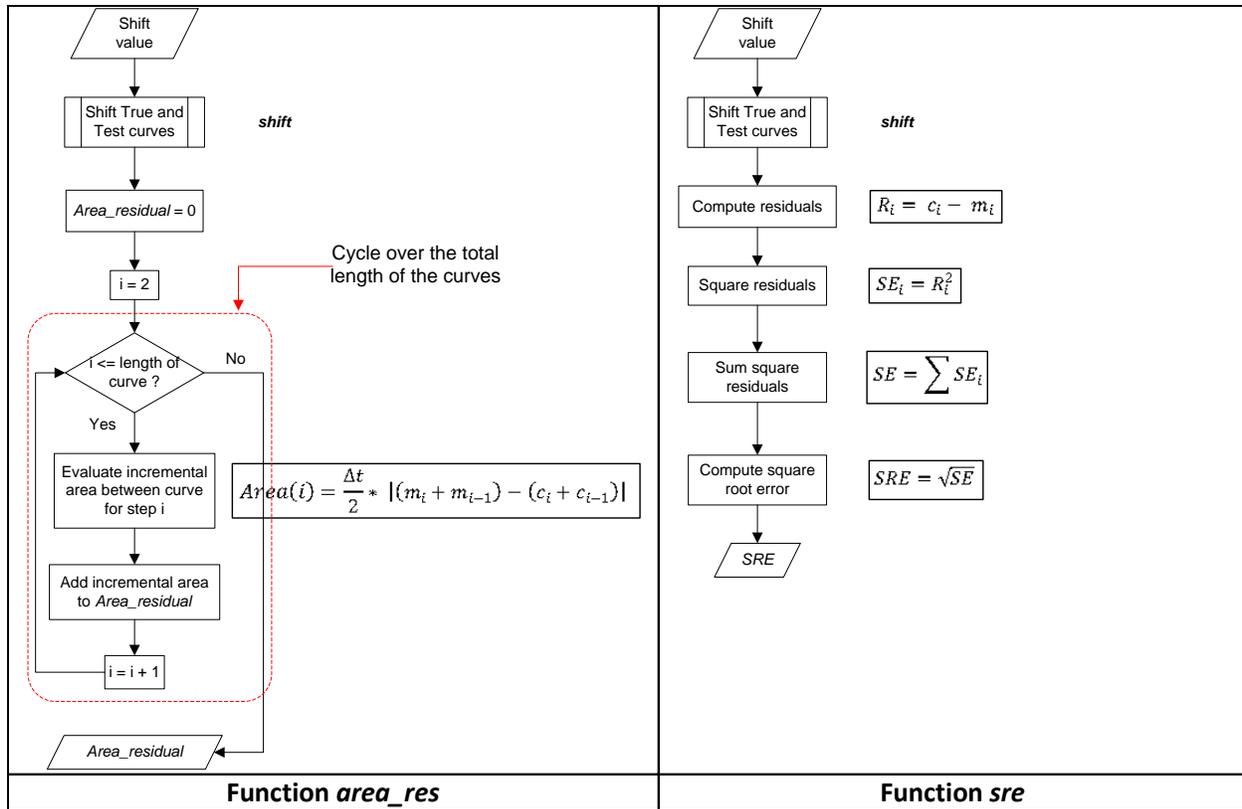


Figure B.58: Algorithm of the user-defined functions *area\_res* (left) and *sre* (right).

The function *shift* has only one input, the value by which to shift the two curves respect to each other. In fact, the data points of the two curves are read directly from the global space of Matlab. This allows the shift function to be recalled also within the local space *area\_res* and *sre*, where the data points of the input curves would not otherwise be available. Based on the sign of the input shift value, the algorithm of the function *shift* creates a shifted time vector for either the test or the true curve and a vector trimmed at the end for the other curve. In particular, a positive shift value corresponds to a time vector for the true curve which starts at the shift value and a time vector for the test curve which is trimmed at the end by the same shift value, in order to have two vectors with the same length. Vice versa happens in case of a negative shift value. Once the appropriate time vectors have been defined, the shifted curves are obtained by computing the corresponding interpolating polynomials on these time points. Eventually, a common time vector is used for both the interpolated true and test curves, which starts at time zero and is trimmed at the end by a value equal to the shift.

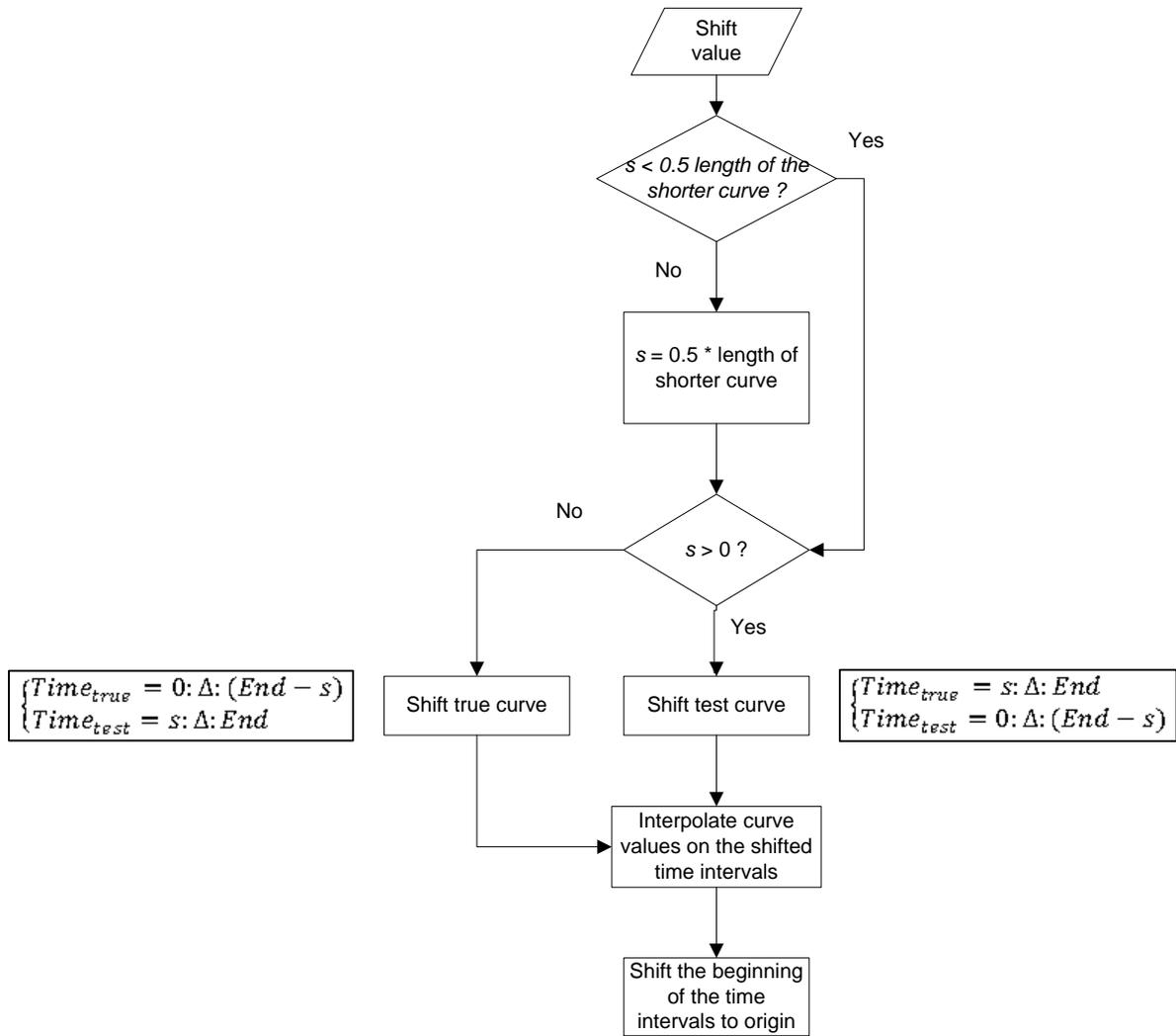


Figure B.59: Algorithm of the user-defined function *shift*.

## Appendix B.V Metrics algorithms

This appendix gives a detailed description of the algorithms used to implement the various comparison metrics available in RSVVP. Note that each metric implemented in RSVVP is repeatedly evaluated considering time intervals increasing in size in order to track the behavior of the metrics when the two curves are compared on a limited portion of the time domain. The starting interval has a dimension equal to 10 percent of the total time interval and the final time interval is the entire time domain on which the true and test curves are compared. In the following sections, the measured and the computed curves are indicated respectively as  $m$  and  $c$ .

### Sprague&Geers

The analytical definition of the magnitude and phase component of the Sprague&Geers metric is defined by the following formulas:

$$M_{SG} = \sqrt{I_{cc}/I_{mm}} - 1 \quad (E5)$$

$$P_{SG} = \frac{1}{\pi} \cos^{-1}(I_{mc}/\sqrt{I_{mm}I_{cc}}) \quad (E6)$$

where  $I_{mm}$ ,  $I_{cc}$  and  $I_{mc}$  are respectively defined as:

$$I_{mm} = (t_2 - t_1)^{-1} \int_{t_1}^{t_2} m^2(t) dt \quad (E7)$$

$$I_{cc} = (t_2 - t_1)^{-1} \int_{t_1}^{t_2} c^2(t) dt \quad (E8)$$

$$I_{mc} = (t_2 - t_1)^{-1} \int_{t_1}^{t_2} m(t) \cdot c(t) dt \quad (E9)$$

The comprehensive error is defined as the vector whose two components are  $M_{SG}$  and  $P_{SG}$ :

$$C_{SG} = \sqrt{M_{SG}^2 + P_{SG}^2} \quad (E10)$$

Assuming that the two curves are characterized by a uniform and common sampling rate, the integrals defined by equations E7 through E9 reduces to the following summation:

$$I_{mm} = \sum m^2 \quad (E11)$$

$$I_{cc} = \sum c^2 \quad (\text{E12})$$

$$I_{mc} = \sum mc \quad (\text{E13})$$

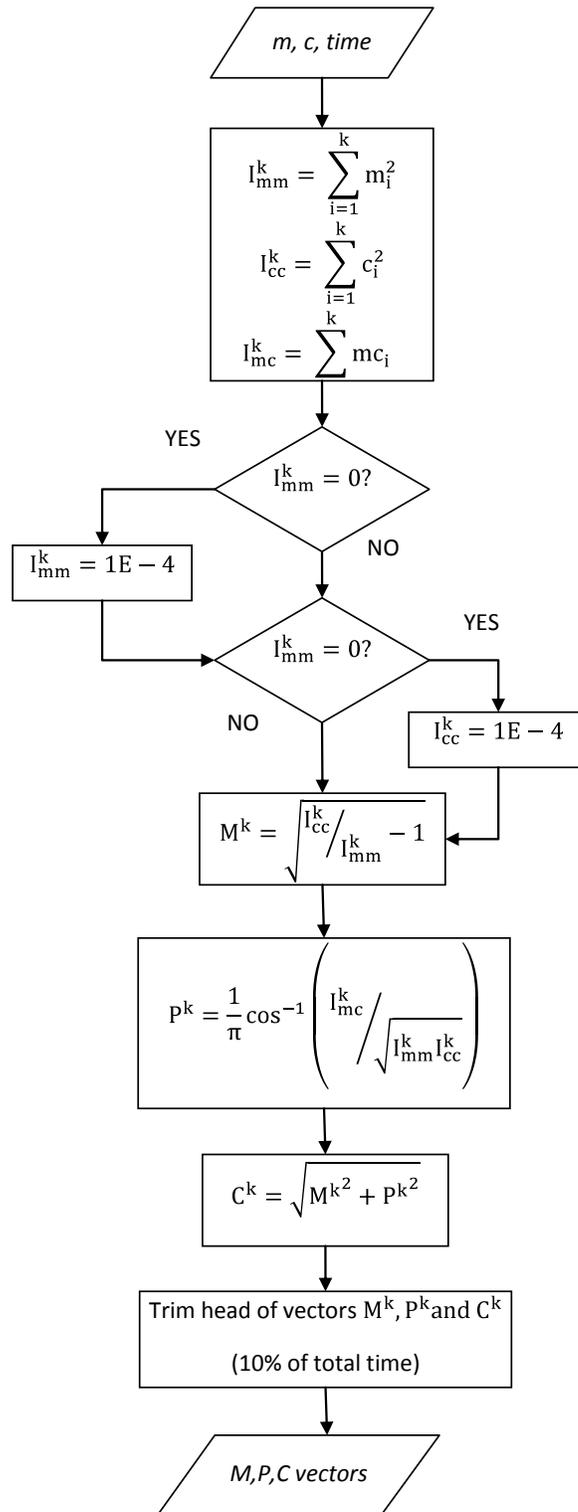


Figure B.60: Algorithm of the Sprague & Geers metric implemented in RSVVP.

## ANOVA

The Analysis of Variance metrics are based on the residuals between the measured and the computed curves. In particular, the residuals are normalized to the peak value of the measured curve. The algorithm of the ANOVA metrics (mean, standard deviation, and t-test of the residuals) is shown in Figure B.61.

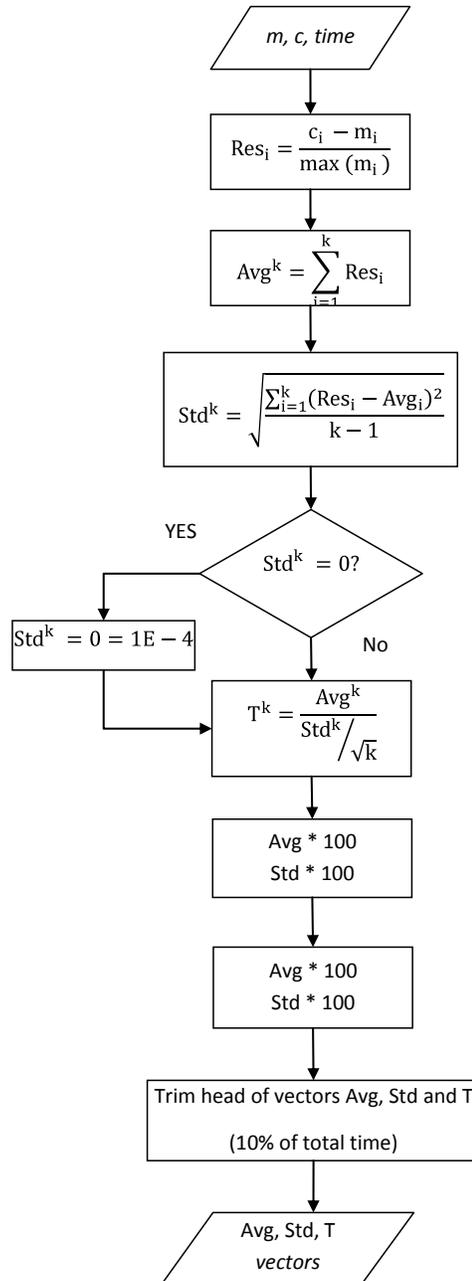


Figure B.61: Algorithm of the ANOVA metrics implemented in RSVVP.

## Appendix C

This appendix shows the values of the comparison metrics computed considering the velocity time histories for each curve of any of the two sets of tests and also their combination.

Two different baseline curves were considered:

- (i) Average response
- (ii) First test of each set/combination.

Table C.1 through Table C.3 show the results considering the average responses for Set 1, Set 2 and their combination respectively; while the corresponding results obtained considering the latter baseline curve are shown in Table C.4 through Table C.6.

**Table C.1. Values of the comparison metrics computed considering the velocity time histories for Set#1 (baseline curve: average response)**

	Lab#1 (Set 1)	Lab#2 (Set 1)	Lab#3 (Set 1)	Lab#4 (Set 1)	Lab#5 (Set 1)	Mean	STD	Acceptable value	
	Value [%]	Value [%]	Value [%]	min [%]	max [%]				
<b>MPC Metrics</b>									
<i>Geers Magnitude</i>	-1.1	-0.6	4	3.3	-5	2.8	1.89	-0.3	5.9
<i>Geers Phase</i>	0.2	0.1	0.1	0.1	0	0.1	0.07	0.0	0.2
<i>Geers Comprehensive</i>	1.1	0.6	4	3.3	5	2.8	1.89	-0.3	5.9
<i>Geers CSA Magnitude</i>	-1.1	-0.6	4	3.3	-5	2.8	1.89	-0.3	5.9
<i>Geers CSA Phase</i>	0.2	0.1	0.1	0.1	0	0.1	0.07	0.0	0.2
<i>Geers CSA Comprehensive</i>	1.1	0.6	4	3.3	5	2.8	1.89	-0.3	5.9
<i>Sprague-Geers Magnitude</i>	-1.1	-0.6	4	3.3	-5	2.8	1.89	-0.3	5.9
<i>Sprague-Geers Phase</i>	2.1	1.5	1.6	1.2	0.9	1.5	0.45	0.7	2.2
<i>Sprague-Geers Comprehensive</i>	2.4	1.6	4.3	3.5	5.1	3.4	1.41	1.0	5.7
<i>Russell Magnitude</i>	-0.9	-0.5	3.3	2.8	-4.3	2.4	1.61	-0.3	5.0
<i>Russell Phase</i>	2.1	1.5	1.6	1.2	0.9	1.5	0.45	0.7	2.2
<i>Russell Comprehensive</i>	2.1	1.4	3.2	2.6	3.9	2.6	0.97	1.0	4.2
<i>Knowles-Gear Magnitude</i>	6.4	4.4	7.1	4.9	5.3	5.6	1.11	3.8	7.5
<i>Knowles-Gear Phase</i>	24.2	3.3	15.2	8.7	12.2	12.7	7.80	-0.2	25.7
<i>Knowles-Gear Comprehensive</i>	11.5	4.2	8.9	5.7	6.9	7.4	2.85	2.7	12.2
<b>Single Value Metrics</b>	Value [%]	Value [%]	Value [%]	min [%]	max [%]				
<i>Whang's inequality metric</i>	2.9	1.7	2.5	2.2	2.8	2.4	0.49	1.6	3.2
<i>Theil's inequality metric</i>	3.4	2.3	3.2	2.5	3	2.9	0.47	2.1	3.7
<i>Ziliacus error metric</i>	5.8	3.4	5	4.5	5.5	4.8	0.94	3.3	6.4
<i>RSS error metric</i>	6.7	4.6	6.5	5	5.8	5.7	0.91	4.2	7.2
<i>WIFac_Error</i>	6.6	4.5	6.2	4.8	5.8	5.6	0.90	4.1	7.1
<i>Regression Coefficient</i>	98.7	99.4	98.8	99.3	99	99.0	0.30	98.5	99.5
<i>Correlation Coefficient</i>	98.9	99.4	99.4	99.7	99.8	99.4	0.35	98.9	100.0
<i>Correlation Coefficient(NARD)</i>	99.8	99.9	99.9	99.9	100	99.9	0.07	99.8	100.0
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value	Value	min	max
<i>Average</i>	0.00	0.00	-0.02	-0.02	0.04	0.02	0.02	-0.01	0.04
<i>Std</i>	0.05	0.04	0.04	0.03	0.02	0.04	0.01	0.02	0.05
<i>T-test</i>	-0.96	5.29	-34.8	-35.5	116				

**Table C.2. Values of the comparison metrics computed considering the velocity time histories for Set#2 (baseline curve: average response)**

	Lab#1 (Set 2)	Lab#2 (Set 2)	Lab#3 (Set 2)	Lab#4 (Set 2)	Lab#5 (Set 2)	Mean	STD	Acceptable value	
	Value [%]	Value [%]	Value [%]	min [%]	max [%]				
<b>MPC Metrics</b>									
<i>Geers Magnitude</i>	-4.2	-0.4	-2	-0.4	7.8	3.0	3.12	-2.2	8.1
<i>Geers Phase</i>	0.1	0.1	0.2	0.2	0.1	0.1	0.05	0.0	0.2
<i>Geers Comprehensive</i>	4.2	0.4	2.1	0.4	7.8	3.0	3.11	-2.2	8.2
<i>Geers CSA Magnitude</i>	-4.2	-0.4	-2	-0.4	7.8	3.0	3.12	-2.2	8.1
<i>Geers CSA Phase</i>	0.1	0.1	0.2	0.2	0.1	0.1	0.05	0.0	0.2
<i>Geers CSA Comprehensive</i>	4.2	0.4	2.1	0.4	7.8	3.0	3.11	-2.2	8.2
<i>Sprague-Geers Magnitude</i>	-4.2	-0.4	-2	-0.4	7.8	3.0	3.12	-2.2	8.1
<i>Sprague-Geers Phase</i>	1	1.5	2.1	2.2	1.7	1.7	0.48	0.9	2.5
<i>Sprague-Geers Comprehensive</i>	4.3	1.5	3	2.3	7.9	3.8	2.51	-0.4	8.0
<i>Russell Magnitude</i>	-3.6	-0.3	-1.8	-0.3	6.1	2.4	2.47	-1.7	6.5
<i>Russell Phase</i>	1	1.5	2.1	2.2	1.7	1.7	0.48	0.9	2.5
<i>Russell Comprehensive</i>	3.3	1.4	2.5	2	5.6	3.0	1.63	0.3	5.7
<i>Knowles-Gear Magnitude</i>	5.1	4.4	6.7	5.2	8.5	6.0	1.64	3.3	8.7
<i>Knowles-Gear Phase</i>	0.5	23.1	57.2	25.5	14.9	24.2	20.86	-10.4	58.9
<i>Knowles-Gear Comprehensive</i>	4.6	10.2	24.2	11.5	9.9	12.1	7.27	0.0	24.1
<b>Single Value Metrics</b>									
<i>Whang's inequality metric</i>	2.6	2.3	2.9	2.9	4.6	3.1	0.90	1.6	4.5
<i>Theil's inequality metric</i>	2.7	2.3	3.5	3.5	4.6	3.3	0.88	1.9	4.8
<i>Ziliacus error metric</i>	5	4.5	5.7	5.8	9.5	6.1	1.97	2.8	9.4
<i>RSS error metric metric</i>	5.3	4.7	6.9	7	9.6	6.7	1.90	3.5	9.9
<i>WIFac_Error</i>	5.3	4.6	6.9	6.9	8.9	6.5	1.67	3.8	9.3
<i>Regression Coefficient</i>	99.3	99.4	98.7	98.7	97.5	98.7	0.76	97.5	100.0
<i>Correlation Coefficient</i>	99.7	99.7	99.2	99.1	99.3	99.4	0.28	98.9	99.9
<i>Correlation Coefficient(NARD)</i>	99.9	99.9	99.8	99.8	99.9	99.9	0.05	99.8	100.0
<b>ANOVA Metrics</b>									
<i>Average</i>	0.03	0.01	0.03	-0.01	-0.06	0.03	0.02	-0.01	0.06
<i>Std</i>	0.03	0.03	0.04	0.05	0.04	0.04	0.01	0.02	0.05
<i>T-test</i>	80.2	27.44	39.34	-15	-93.49				

Table C.3. Values of the comparison metrics computed considering the velocity time histories for combination of Set#1 and Set#2 (baseline curve: average response)

	Lab#1 (Set 1)	Lab#1 (Set 2)	Lab#2 (Set 1)	Lab#2 (Set 2)	Lab#3 (Set 1)	Lab#3 (Set 2)	Lab#4 (Set 1)	Lab#4 (Set 2)	Lab#5 (Set 1)	Lab#5 (Set 2)	Mean	STD	Acceptable value	
	Value [%]	Value [%]	Value [%]	min [%]	max [%]									
<b>MPC Metrics</b>														
<i>Geers Magnitude</i>	-3.2	-2.1	-2.7	1.8	1.8	0.1	1.2	1.8	-7	10.1	3.2	3.04	-1.9	8.2
<i>Geers Phase</i>	0.4	0.1	0.2	0.2	0.1	0.3	0	0.2	0	0.1	0.2	0.13	0.0	0.4
<i>Geers Comprehensive</i>	3.2	2.1	2.7	1.8	1.8	0.4	1.2	1.8	7	10.1	3.2	3.00	-1.8	8.2
<i>Geers CSA Magnitude</i>	-3.2	-2.1	-2.7	1.8	1.8	0.1	1.2	1.8	-7	10.1	3.2	3.04	-1.9	8.2
<i>Geers CSA Phase</i>	0.4	0.1	0.2	0.2	0.1	0.3	0	0.2	0	0.1	0.2	0.13	0.0	0.4
<i>Geers CSA Comprehensive</i>	3.2	2.1	2.7	1.8	1.8	0.4	1.2	1.8	7	10.1	3.2	3.00	-1.8	8.2
<i>Sprague-Geers Magnitude</i>	-3.2	-2.1	-2.7	1.8	1.8	0.1	1.2	1.8	-7	10.1	3.2	3.04	-1.9	8.2
<i>Sprague-Geers Phase</i>	2.8	1.6	2	2.1	1.3	2.6	0.8	1.7	0.8	1.5	1.7	0.67	0.6	2.8
<i>Sprague-Geers Comprehensive</i>	4.2	2.6	3.3	2.8	2.2	2.6	1.4	2.5	7.1	10.2	3.9	2.71	-0.6	8.4
<i>Russell Magnitude</i>	-2.7	-1.8	-2.3	1.5	1.5	0.1	1	1.5	-5.9	7.6	2.6	2.33	-1.3	6.5
<i>Russell Phase</i>	2.8	1.6	2	2.1	1.3	2.6	0.8	1.7	0.8	1.5	1.7	0.67	0.6	2.8
<i>Russell Comprehensive</i>	3.4	2.1	2.7	2.3	1.8	2.3	1.1	2	5.3	6.9	3.0	1.78	0.0	5.9
<i>Knowles-Gear Magnitude</i>	9.1	4.7	6.6	7.1	5	8.1	2.5	3.1	7.3	10.8	6.4	2.62	2.1	10.8
<i>Knowles-Gear Phase</i>	27.9	8	1.8	32.3	9.6	69	3.4	19.9	6.7	8.5	18.7	20.46	-15.2	52.7
<i>Knowles-Gear Comprehensive</i>	14.1	5.4	6.1	14.7	6	29.1	2.7	8.6	7.2	10.4	10.4	7.58	-2.2	23.0
<b>Single Value Metrics</b>														
<i>Whang's inequality metric</i>	4.1	2.2	2.7	3.5	1.5	3.2	1.2	2	3.7	5	2.9	1.21	0.9	4.9
<i>Theil's inequality metric</i>	4.7	2.7	3.4	3.4	2.3	4.1	1.4	2.9	3.9	5.3	3.4	1.15	1.5	5.3
<i>Ziliacus error metric</i>	8.1	4.4	5.4	7	3.1	6.4	2.4	4	7.1	10.5	5.8	2.47	1.7	9.9
<i>RSS error metric</i>	9.2	5.3	6.7	7	4.6	8.3	2.8	5.8	7.5	11.2	6.8	2.40	2.9	10.8
<i>WIFac_Error</i>	9	5.2	6.6	6.7	4.5	8.1	2.8	5.7	7.5	10.1	6.6	2.18	3.0	10.2
<i>Regression Coefficient</i>	97.6	99.2	98.7	98.6	99.4	98.1	99.8	99.1	98.4	96.5	98.5	0.96	96.9	100.1
<i>Correlation Coefficient</i>	98.1	99.4	99	99.4	99.5	98.9	99.8	99.4	99.8	99.4	99.3	0.50	98.4	100.1
<i>Correlation Coefficient(NARD)</i>	99.6	99.9	99.8	99.8	99.9	99.7	100	99.8	100	99.9	99.8	0.13	99.6	100.0
<b>ANOVA Metrics</b>														
<i>Average</i>	0.01	0.02	0.01	0.00	-0.01	0.02	-0.01	-0.02	0.05	-0.07	0.02	0.02	-0.01	0.06
<i>Std</i>	0.07	0.03	0.05	0.05	0.03	0.06	0.02	0.04	0.03	0.05	0.04	0.02	0.02	0.07
<i>T-test</i>	10.08	39.93	18.85	3.43	-22.82	17.84	-20.51	-40.57	109.6	-105.2				

**Table C.4. Values of the comparison metrics computed considering the velocity time histories for Set#1 (baseline curve: Lab#1)**

	Lab#2 (Set 1)	Lab#3 (Set 1)	Lab#4 (Set 1)	Lab#5 (Set 1)	Mean	STD	Acceptable value	
	Value [%]	Value [%]	Value [%]	Value [%]	Value [%]	Value [%]	min [%]	max [%]
<b>MPC Metrics</b>								
<i>Geers Magnitude</i>	0.5	5.1	4.5	-4	3.5	2.07	0.1	7.0
<i>Geers Phase</i>	0.2	0.6	0.5	0.4	0.4	0.17	0.1	0.7
<i>Geers Comprehensive</i>	0.5	5.1	4.5	4	3.5	2.07	0.1	7.0
<i>Geers CSA Magnitude</i>	0.5	5.1	4.5	-4	3.5	2.07	0.1	7.0
<i>Geers CSA Phase</i>	0.2	0.6	0.5	0.4	0.4	0.17	0.1	0.7
<i>Geers CSA Comprehensive</i>	0.5	5.1	4.5	4	3.5	2.07	0.1	7.0
<i>Sprague-Geers Magnitude</i>	0.5	5.1	4.5	-4	3.5	2.07	0.1	7.0
<i>Sprague-Geers Phase</i>	2	3.5	3.1	2.8	2.9	0.64	1.8	3.9
<i>Sprague-Geers Comprehensive</i>	2	6.2	5.4	4.9	4.6	1.83	1.6	7.7
<i>Russell Magnitude</i>	0.4	4.1	3.6	-3.4	2.9	1.68	0.1	5.7
<i>Russell Phase</i>	2	3.5	3.1	2.8	2.9	0.64	1.8	3.9
<i>Russell Comprehensive</i>	1.8	4.8	4.2	3.9	3.7	1.30	1.5	5.8
<i>Knowles-Gear Magnitude</i>	6.2	12.3	10.2	8.5	9.3	2.59	5.0	13.6
<i>Knowles-Gear Phase</i>	36.2	52	43.4	48	44.9	6.78	33.6	56.2
<i>Knowles-Gear Comprehensive</i>	15.8	24	20	21.1	20.2	3.40	14.6	25.9
<b>Single Value Metrics</b>								
<i>Whang's inequality metric</i>	2.7	5.1	4.7	4.4	4.2	1.06	2.5	6.0
<i>Theil's inequality metric</i>	3.1	6	5.4	4.8	4.8	1.25	2.8	6.9
<i>Zilliacus error metric</i>	5.3	10.3	9.4	8.5	8.4	2.18	4.8	12.0
<i>RSS error metric metric</i>	6.2	12.3	11	9.5	9.8	2.63	5.4	14.1
<i>WIFac_Error</i>	6.1	11.5	10.3	9.3	9.3	2.32	5.5	13.1
<i>Regression Coefficient</i>	98.7	94.9	96	97	96.7	1.61	94.0	99.3
<i>Correlation Coefficient</i>	99	96.9	97.7	98.1	97.9	0.87	96.5	99.4
<i>Correlation Coefficient(NARD)</i>	99.8	99.4	99.5	99.6	99.6	0.17	99.3	99.9
<b>ANOVA Metrics</b>								
<i>Average</i>	0.00	-0.02	-0.02	0.04	0.02	0.02	-0.01	0.05
<i>Std</i>	0.05	0.09	0.08	0.06	0.07	0.02	0.04	0.10
<i>T-test</i>	4.99	-16.4	-14.2	44				

**Table C.5. Values of the comparison metrics computed considering the velocity time histories for Set#2 (baseline curve: Lab#1)**

	Lab#2 (Set 2)	Lab#3 (Set 2)	Lab#4 (Set 2)	Lab#5 (Set 2)	Mean	STD	Acceptable value	
	Value [%]	Value [%]	Value [%]	Value [%]	Value [%]	Value [%]	min [%]	max [%]
<b>MPC Metrics</b>								
<i>Geers Magnitude</i>	4	2.2	4	12.5	5.7	4.63	-2.0	13.4
<i>Geers Phase</i>	0.1	0.2	0.4	0.3	0.3	0.13	0.0	0.5
<i>Geers Comprehensive</i>	4	2.3	4	12.5	5.7	4.60	-1.9	13.3
<i>Geers CSA Magnitude</i>	4	2.2	4	12.5	5.7	4.63	-2.0	13.4
<i>Geers CSA Phase</i>	0.1	0.2	0.4	0.3	0.3	0.13	0.0	0.5
<i>Geers CSA Comprehensive</i>	4	2.3	4	12.5	5.7	4.60	-1.9	13.3
<i>Sprague-Geers Magnitude</i>	4	2.2	4	12.5	5.7	4.63	-2.0	13.4
<i>Sprague-Geers Phase</i>	1.5	2.2	2.7	2.5	2.2	0.53	1.4	3.1
<i>Sprague-Geers Comprehensive</i>	4.3	3.1	4.8	12.7	6.2	4.38	-1.0	13.5
<i>Russell Magnitude</i>	3.3	1.9	3.3	9.2	4.4	3.25	-1.0	9.8
<i>Russell Phase</i>	1.5	2.2	2.7	2.5	2.2	0.53	1.4	3.1
<i>Russell Comprehensive</i>	3.2	2.6	3.8	8.5	4.5	2.69	0.1	9.0
<i>Knowles-Gear Magnitude</i>	6.8	9	7	13.7	9.1	3.21	3.8	14.4
<i>Knowles-Gear Phase</i>	22.5	56.5	25.8	15.3	30.0	18.19	-0.2	60.2
<i>Knowles-Gear Comprehensive</i>	11.1	24.5	12.3	14	15.5	6.13	5.3	25.7
<b>Single Value Metrics</b>	Value [%]	Value [%]	Value [%]	Value [%]	Value [%]	Value [%]	min [%]	max [%]
<i>Whang's inequality metric</i>	2.9	3.1	3.7	7	4.2	1.91	1.0	7.4
<i>Theil's inequality metric</i>	3.1	3.6	4.7	7.1	4.6	1.78	1.7	7.6
<i>Zilliacus error metric</i>	5.8	6.2	7.7	15	8.7	4.30	1.5	15.8
<i>RSS error metric metric</i>	6.4	7.3	9.5	15.1	9.6	3.91	3.1	16.1
<i>WIFac_Error</i>	6.1	7.1	9.1	13.3	8.9	3.19	3.6	14.2
<i>Regression Coefficient</i>	98.9	98.6	97.6	93.9	97.3	2.30	93.4	101.1
<i>Correlation Coefficient</i>	99.6	99.1	98.7	98.4	99.0	0.52	98.1	99.8
<i>Correlation Coefficient(NARD)</i>	99.9	99.8	99.6	99.7	99.8	0.13	99.5	100.0
<b>ANOVA Metrics</b>	Value	Value	Value	Value	Value	Value	min	max
<i>Average</i>	-0.02	0.00	-0.04	-0.09	0.04	0.04	-0.03	0.10
<i>Std</i>	0.04	0.05	0.05	0.06	0.05	0.01	0.04	0.06
<i>T-test</i>	-27.2	-5.36	-51.7	-96				

Table C.6. Values of the comparison metrics computed considering the velocity time histories for combination of Set#1 and Set#2 (baseline curve: Lab#1 of Set 1)

	Lab#1 (Set 2)	Lab#2 (Set 1)	Lab#2 (Set 2)	Lab#3 (Set 1)	Lab#3 (Set 2)	Lab#4 (Set 1)	Lab#4 (Set 2)	Lab#5 (Set 1)	Lab#5 (Set 2)	Mean	STD	Acceptable value	
	Value [%]	Value [%]	Value [%]	min [%]	max [%]								
<b>MPC Metrics</b>													
<i>Geers Magnitude</i>	1	0.5	5.1	5.1	3.3	4.5	5.1	-4	13.7	4.7	3.80	-1.6	11.0
<i>Geers Phase</i>	0.8	0.2	1.1	0.6	1.2	0.5	0.3	0.4	0.4	0.6	0.35	0.0	1.2
<i>Geers Comprehensive</i>	1.3	0.5	5.2	5.1	3.5	4.5	5.1	4	13.7	4.8	3.75	-1.5	11.0
<i>Geers CSA Magnitude</i>	1	0.5	5.1	5.1	3.3	4.5	5.1	-4	13.7	4.7	3.80	-1.6	11.0
<i>Geers CSA Phase</i>	0.8	0.2	1.1	0.6	1.2	0.5	0.3	0.4	0.4	0.6	0.35	0.0	1.2
<i>Geers CSA Comprehensive</i>	1.3	0.5	5.2	5.1	3.5	4.5	5.1	4	13.7	4.8	3.75	-1.5	11.0
<i>Sprague-Geers Magnitude</i>	1	0.5	5.1	5.1	3.3	4.5	5.1	-4	13.7	4.7	3.80	-1.6	11.0
<i>Sprague-Geers Phase</i>	4	2	4.7	3.5	5	3.1	2.4	2.8	2.8	3.4	1.02	1.7	5.1
<i>Sprague-Geers Comprehensive</i>	4.2	2	7	6.2	6	5.4	5.6	4.9	13.9	6.1	3.25	0.7	11.5
<i>Russell Magnitude</i>	0.9	0.4	4.1	4.1	2.7	3.6	4.1	-3.4	9.9	3.7	2.71	-0.8	8.2
<i>Russell Phase</i>	4	2	4.7	3.5	5	3.1	2.4	2.8	2.8	3.4	1.02	1.7	5.1
<i>Russell Comprehensive</i>	3.7	1.8	5.6	4.8	5.1	4.2	4.2	3.9	9.1	4.7	1.96	1.4	8.0
<i>Knowles-Gear Magnitude</i>	11.8	6.2	16.1	12.3	15.3	10.2	9.1	8.5	17.7	11.9	3.84	5.5	18.3
<i>Knowles-Gear Phase</i>	49.8	36.2	83.5	52	134.4	43.4	11.1	48	26.9	53.9	36.05	-5.9	113.8
<i>Knowles-Gear Comprehensive</i>	23	15.8	37.1	24	56.6	20	9.5	21.1	19.6	25.2	13.89	2.1	48.2
<b>Single Value Metrics</b>	Value [%]	Value [%]	Value [%]	min [%]	max [%]								
<i>Whang's inequality metric</i>	5.4	2.7	7.2	5.1	6.5	4.7	3.7	4.4	7	5.2	1.52	2.7	7.7
<i>Theil's inequality metric</i>	6.4	3.1	7.8	6	8.1	5.4	4.5	4.8	7.8	6.0	1.71	3.1	8.8
<i>Zilliacus error metric</i>	10.6	5.3	14.6	10.3	12.9	9.4	7.5	8.5	14.9	10.4	3.22	5.1	15.8
<i>RSS error metric</i>	12.8	6.2	16.1	12.3	16.4	11	9.2	9.5	16.6	12.2	3.65	6.2	18.3
<i>WIFac_Error</i>	12.2	6.1	14.8	11.5	15.4	10.3	8.7	9.3	14.5	11.4	3.14	6.2	16.6
<i>Regression Coefficient</i>	94.5	98.7	91.1	94.9	90.7	96	97.2	97	90.5	94.5	3.07	89.4	99.6
<i>Correlation Coefficient</i>	96.2	99	95.7	96.9	95.2	97.7	98.2	98.1	97.9	97.2	1.28	95.1	99.3
<i>Correlation Coefficient(NARD)</i>	99.2	99.8	98.9	99.4	98.8	99.5	99.7	99.6	99.6	99.4	0.35	98.8	100.0
<b>ANOVA Metrics</b>	Value	Value	Value	min	max								
<i>Average</i>	0.01	0.00	-0.01	-0.02	0.01	-0.02	-0.03	0.04	-0.08	0.02	0.02	-0.02	0.06
<i>Std</i>	0.10	0.05	0.12	0.09	0.12	0.08	0.06	0.06	0.09	0.09	0.03	0.04	0.13
<i>T-test</i>	7.29	4.99	-4.37	-16.38	3.25	-14.17	0.024	44	-60.89				