

A Genetic Algorithm for Fixture Synthesis and Variation

by

Shiping Huang

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

Manufacturing Engineering

by

May 28, 2011

APPROVED:

Yiming (Kevin) Rong, Advisor

Professor of Mechanical Engineering and
Associate Director of Manufacturing and Materials Engineering

Abstract

Concepts in manufacturing such as CIMS (Computer Integrated Manufacturing Systems), JIT (Just In Time), Lean Production, Virtual Manufacturing, and Flexible Fixturing have been proposed to meet the fundamental requirements of manufacturing - decrease the cost and satisfy the needs of customers. Fast fixture generation and fixture reusability are essential in the current manufacturing environment.

The dissertation focuses on the models, methods, and algorithms for fixture synthesis and variation that satisfy the functional requirements specified by on-site industrial engineers. With the reusability of a fixture base combined with variation of other fixture components, fixture configuration can be rapidly adapted and accommodated to the new workpiece. The dissertation presents methods and algorithms for fixture base synthesis, which directly result in fixture reusability. Optimization functions are derived based on engineering requirements due to the mass production nature of automotive parts. Specific optimization algorithms are developed and their complexities, compared to other alternatives, are comprehensively evaluated according to different optimization functions.

The fixture variation and reusability provide an engineering tool to rapidly generate and validate fixtures in production planning stage. It applies scientific reasoning methodology in combination with best knowledge of fixture designs, which heavily relies on designers' manufacturing knowledge and experience. It also provides means to bridge the gap between CAD and CAM integration and therefore reduces the new product and production development cycle time and cost while maintaining the quality of fixtures.

Acknowledgments

I would like to express my gratitude to my advisor, Prof. Yiming Rong, for his support, advice, patience, and encouragement throughout my graduate studies. His technical and editorial advice was essential to the completion of this dissertation and has taught me innumerable lessons and insights on the workings of academic research in general.

My thanks also go to the committee members, Prof. Richard D. Sisson, Prof. Christopher A. Brown, Prof. Mustapha S. Fofana, Prof. Elke Rundensteiner and Prof. Wenzhen Huang, who provided invaluable feedback and suggestions to my previous talks and dissertation drafts that helped to improve the dissertation writing and presentation.

Lastly, but most importantly, I would like to acknowledge my wife Hong's continual support during the past few years. Our son Charlie encouraged me to apply the teacher position at Harrington school upon finishing my PHD degree. Their support and encouragement was in the end what made this dissertation possible.

This dissertation work is partially supported under NSF and Delphi grant.

Contents

1	Introduction	1
1.1	Fixtures in Manufacturing	1
1.1.1	Bill of Process	5
1.1.2	Flexible Fixtures	8
1.2	Industrial practices	9
1.2.1	Part Family	9
1.2.2	Current Practices and Problems	12
1.2.3	What's Needed	14
1.2.4	Motivation	15
1.3	Genetic Algorithms	16
1.4	Objectives	17
1.5	Research Approaches	18
1.6	Dissertation Organization	19
2	Related Work	21
2.1	Automated Fixture Configuration Design	22
2.1.1	General Fixture	22
2.1.2	Modular Fixture	23
2.1.3	Dedicated Fixture	27

2.1.4	Flexible Fixture Systems	29
2.2	Searching Optimization Algorithms	33
2.3	Summary	36
3	A Genetic Algorithm for Fixture Base Synthesis	38
3.1	Fixture Base in Mass Production	39
3.1.1	Fixture Base and its Representation	40
3.1.2	Design Windows of Fixture Base	41
3.1.2.1	Design Windows of Workpiece	41
3.1.2.2	Design Windows of Fixture Base	42
3.2	Engineering Requirements and Optimization Objectives	44
3.3	Fixture base Synthesis Algorithms	47
3.3.1	Genetic Algorithms	47
3.3.2	Point Transformations	49
3.3.3	Minimize Cover Circle	52
3.4	Algorithm Implementation	54
3.4.1	Individual Representation	56
3.4.2	Genetic Operators	57
3.4.2.1	Reproduction	57
3.4.2.2	Crossover	58
3.4.2.3	Mutation	59
3.4.2.4	Constraint Handling	60
3.4.2.5	Elitist Strategy	60
3.4.3	Input Data Format	60
3.4.4	A Case Study	62

4	An Improved Algorithm for Fixture Base Synthesis	66
4.1	Preliminary Fixture Base Synthesis Using Procrustes Analysis	67
4.1.1	Fixture Base Synthesis for Two-Part Family	68
4.1.2	Fixture Base Synthesis for Multi-Part Family	71
4.2	Advanced Fixture Base Synthesis	74
4.2.1	Advanced Synthesis Algorithm	75
4.2.2	Search Algorithm for $Y_{(ct)}$ and $\bar{Y}_{(ct)}$	77
4.3	Engineering Objected Optimization	79
4.3.1	Fixture Base Synthesis: Optimization Objective I	80
4.3.2	Fixture Base Synthesis: Optimization Objective II	81
4.3.3	Fixture Base Synthesis: Optimization Objective III	82
4.4	Algorithm Efficiency	86
5	Structural Analysis and Fixture Modeling	89
5.1	Workpiece Modeling	89
5.1.1	Workpiece Model	90
5.1.2	Workpiece Variation Model	94
5.2	Fixture Modeling	94
5.2.1	Fixture Model	95
5.3	Fixture Variation Model	97
5.4	Standard Components Database	103
5.4.1	Standard Component Database	104
5.4.2	Selection of the Standard Components	105
5.5	Support Modeling	108
5.5.1	Support Model	108

5.5.2	Parametric Dimension Definition	109
6	Implementation	112
6.1	System Integration	112
6.1.1	Conceptual Fixture Design	112
6.1.1.1	Initialization	113
6.1.1.2	Layout Specification	114
6.1.1.3	Conceptual Design	115
6.1.1.4	Feasibility Checking	115
6.1.2	Detailed Fixture Design	118
6.1.2.1	Functional Requirement Analysis	119
6.1.2.2	Approaches and System Interfaces	122
6.2	Variation Fixture Design Process	126
6.3	A Case Study	128
6.3.1	Workpiece and Workpiece Variation	128
6.3.2	Support Model and Its Variation	130
6.3.3	Fixture Variation Results	131
6.4	Summary	131
7	Conclusions	133
7.1	Conclusions	133
7.2	Contributions	134
7.3	Future Work	135

List of Figures

1.1	Examples of Modular Fixtures and Dedicated Fixtures	4
1.2	Bill of Process for a Caliper	6
1.3	A Working Fixture for Calipers	7
1.4	Knuckles and Their Variations	10
1.5	Calipers from Two Part Families	11
2.1	An Intelligent Fixturing System	32
3.1	An Example of Fixture Base and Its Mounting Features	40
3.2	Design Windows for Manufacturing Features of Calipers	42
3.3	Design Windows for Fixture Base	43
3.4	Design Window of Fixture Base Synthesis	45
3.5	A Genetic Algorithm	50
3.6	Minimum Spanning Circle of a Set of Points	53
3.7	The Circle Passing Through Three Points	54
3.8	A Flowchart of Working Principle of a Genetic Algorithm	55
3.9	Binary Representation in GA	56
3.10	Illustration of Crossover Operator	58

3.11	Illustration of Mutation Operator	59
3.12	A Dataset for Fixture base Synthesis	61
3.13	GA Running Process for Fixture base Synthesis	63
3.14	GA Fitness vs Generations	65
4.1	Procedure of Fixture Base Synthesis Method	67
4.2	Coordinate Registration of Configurations	69
4.3	GPA for Multi-part Family	73
4.4	Procedure of the Advanced Configuration Optimization	76
4.5	GA for $Y_{(ct)}$ and $\bar{Y}_{(ct)}$	78
4.6	Search Algorithm for $Y_{(ct)}$ and $\bar{Y}_{(ct)}$	79
4.7	Processing for Engineering Requirement Function III	83
4.8	Enlarged View of Workspace Fixture Unit 6 in Figure 4.7	84
4.9	Fitness Changing of Improved Fixture base Synthesis Algorithm	87
4.10	Times Used of Two Algorithms	88
5.1	Fixture Coordinate Systems	91
5.2	Representation of a Machining Feature	93
5.3	Workpiece Variation Model	95
5.4	Acting Point $\vec{\mathbf{P}}$ and External Normal Vector $\vec{\mathbf{n}}$	97
5.5	Generic Model of a Locator	105
5.6	Standard Components (Locators/Clamps)	106
5.7	Virtual Reference Cubic Box (VRCB)	107
5.8	Dimensions of a Support	110
6.1	Conceptual Fixture Design and Layout Modification	116

6.2	Conceptual Workpiece Layout Feasibility Checking	117
6.3	Cutting Simulation	118
6.4	System Interface for Detailed Fixture Design	123
6.5	Variation Fixture Design Process	127
6.6	Fixturing Features and Machining Features of a Caliper	129
6.7	Datum Frames	131
6.8	Base Level Model of A Support	131
6.9	A Support Used in 540 Fixtures	131
6.10	A Variation of a Support	131

List of Tables

- 1.1 Caliper Families from Production Engineering 11
- 1.2 Current Practices: Fixtures and Machining Features for Calipers . . . 13

- 3.1 GA Parameters 62

- 4.1 Algorithm Complexity 88

- 5.1 Example of a Family Table 104

- 6.1 Manufacturing Features From a Caliper of 540 Family 129
- 6.2 Fixturing Features of Calipers 130

Chapter 1

Introduction

Under global competition, manufacturing enterprises are constantly searching for useful methods to shorten the time span from product conceptualization to marketing. The activities involved in this period include process preparation, fixture design, tool plan, and production line specification. Fixture design, as the bridge of these activities, plays a key role in manufacturing preparation.

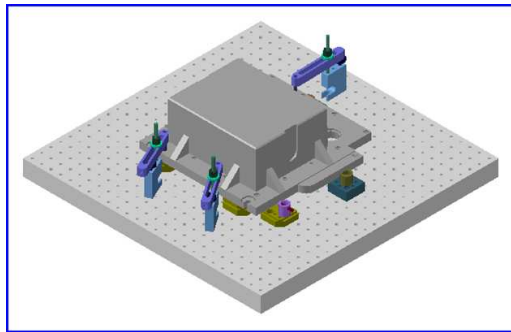
1.1 Fixtures in Manufacturing

Fixtures are defined as a set of systematically structured components functioned to locate and constrain a workpiece in machining, testing, assembling and other manufacturing operations. To ensure that the workpiece is produced according to the specified shape, dimensions, and tolerances, it is essential that it is appropriately located and clamped on the machine tool [Chang, 1998]. The configuration of a machining fixture depends not only on workpiece characteristics, but also on the sequence of machining operations, magnitude and orientation of the expected cutting forces, capabilities of the machining tools, and cost of those operations.

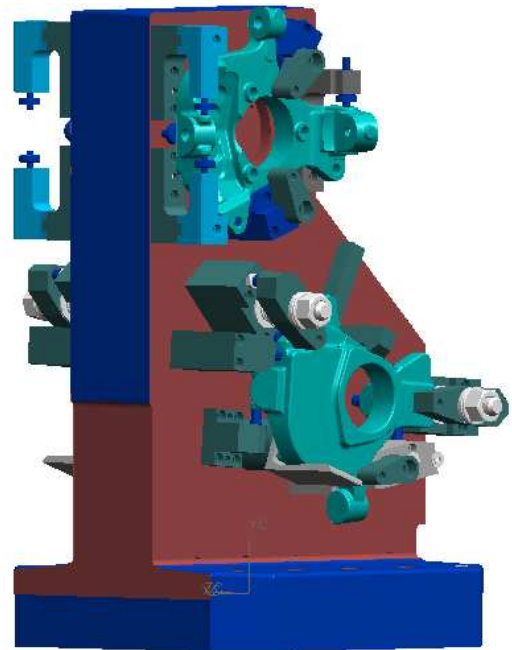
As a workpiece holding device, essentially a fixture must provide functionalities as bellow [Campbell, 1994, Rong, 2005, Rong, 1999]:

- *Locating*: One of the principal purposes of a machining fixture is to locate the workpiece surfaces for performing a machining operation. That means to have all or part of the degrees of freedom of workpiece constrained. This is usually done with respect to a number of factors to be considered such as the reference datum, supporting surfaces, features that are likely to obstruct the tool movement or access direction, etc.
- *Clamping*: A clamp can be defined as a device for providing an invariant location with respect to an external loading system. In other words, the process of clamping induces a locking effect which, through frictional or some other forms of mechanism, provides a stability of location which cannot be changed until and unless external loading is able to overcome the locking effect. Hence, when a cutting force is producing a load or moment on the workpiece, it is necessary that a sufficient clamping force must be exerted to withstand such actions. The creation and retention of locking effect against external loads is the principal objectives of any locking devices.
- *Others*: Beside basic functionalities of locating and clamping, certain additional requirements are desired such as proper clamping sequence to improve accuracy and error proofing to avoid unnecessary mistaken loading. These also include ergonomic and economical issues.

Figure 1.1 shows two examples of workpieces held with a fixture that is placed on the work platform. Figure 1.1a shows workpiece held with a modular fixture, where the fixture base plate and all other fixture components, including locators, clamps, and supports are standard. They can be detached and recomposed to accommodate different workpieces. The displayed fixture has a typical 3-2-1 locating schema. That is, three bottom locating, two side locating and one side locating in two orthogonal sides and three clamping units. Figure 1.1b shows four workpieces (knuckles) loaded on a tombstone simultaneously, where the tombstone and all other components including locators, clamps, and supports are dedicated. They are intentionally designed to hold this workpiece model only. When this workpiece model is discontinued, the fixture is discarded and is not designed to be reconfigured to accommodate to new workpiece models. The dedicated workpiece holding device that has the capability of loading multiple parts can save time in process change over. It also can provide sufficient stiffness and clamping force so as to ensure workpiece stability and locating tolerance when machining as well.



(a) A modular fixture



(b) A dedicated fixture

Figure 1.1: Examples of modular fixtures and dedicated fixtures (Courtesy of Delphi).

1.1.1 Bill of Process

Figure 1.2 is the typical machining process of a caliper under the current manufacturing capabilities. There are two setups to finish the whole machining features. In the setup I, outboard surfaces, mounting holes, piston bores and counterbores are machined. In the setup II two mounting holes and their spotfaces are used to locate the caliper, in which connector/bleeder holes are milled and taped.

If design permits, the all features can be machined in one setup. In this case, the connector/bleeder holes are parallel with piston bore. When machining those holes, the machine table needs to rotate 180 degrees. The typical production line is two combined CNC machines - Toyoda (setup I) and Kitamura (setup II). The two setups also could be finished just in one CNC machine - Mori Seiki (Figure 1.3).

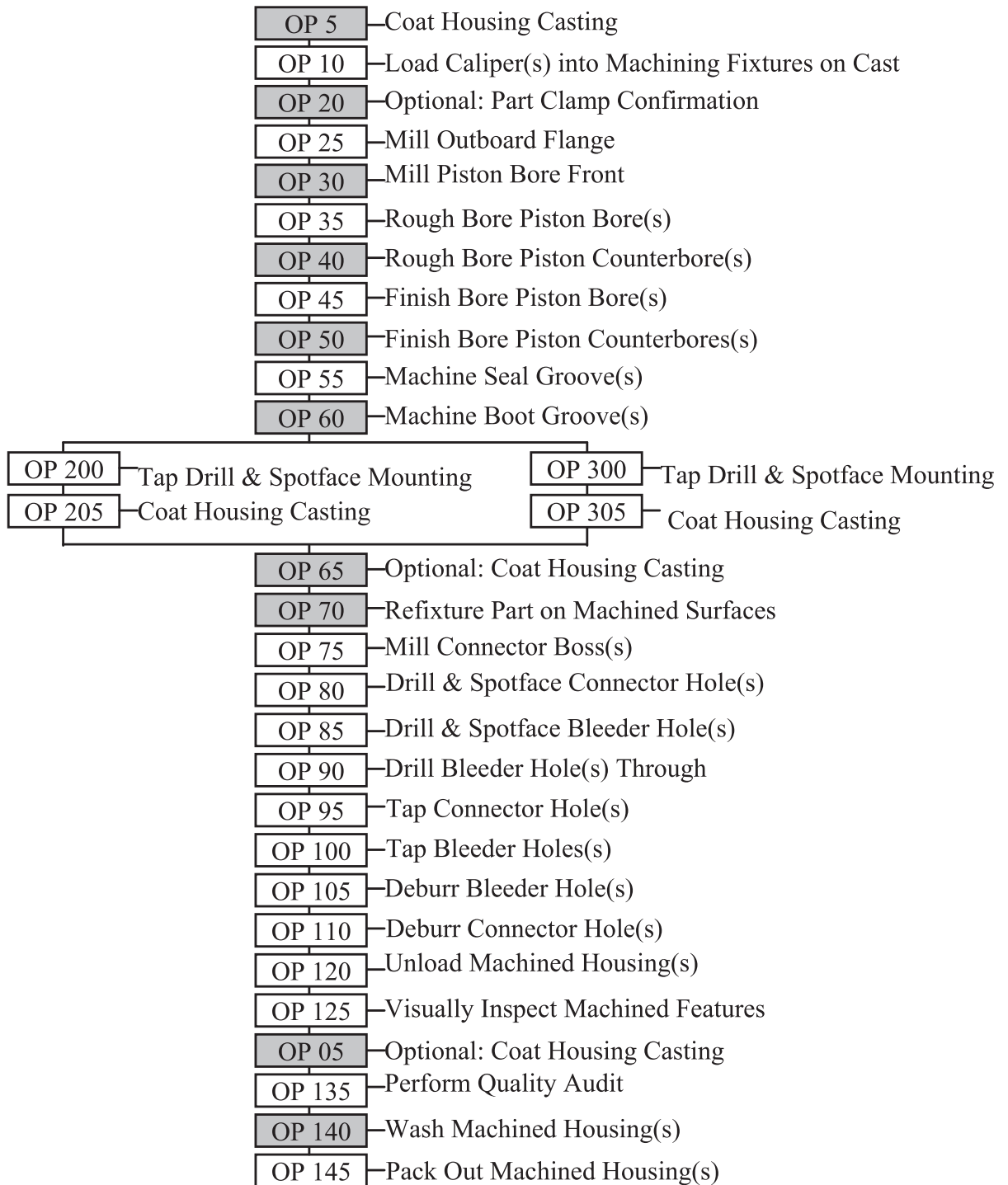


Figure 1.2: Bill of process (BOP) for a caliper (Courtesy of Delphi).

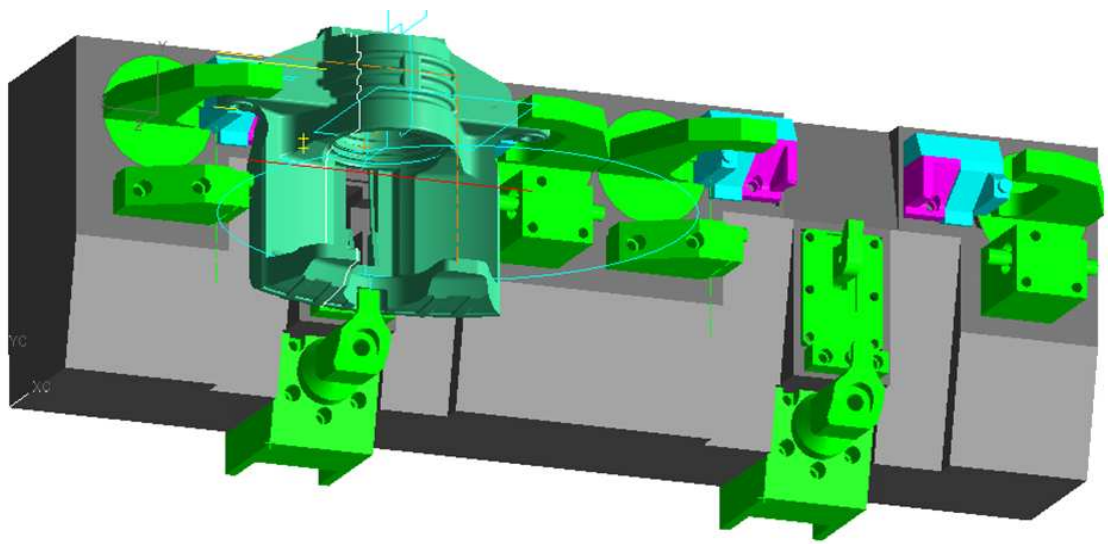


Figure 1.3: A working fixture for calipers (Courtesy of Delphi).

1.1.2 Flexible Fixtures

The cost of designing and fabricating the fixture of an FMS may take 10–20% of the total system cost [Bai, 1995]. To reduce the fixturing cost and lead time, flexible fixturing is a plausible option for such requirements. Flexible fixturing involves using a single device to hold parts or assemblies of different shapes and sizes while they are subjected to a wide variety of external force fields and torque associated with conventional manufacturing operations. There are several different categories of flexible fixtures such as phase-change, modular, adjustable, and programmable fixtures where modular fixtures are the most widely used in industry.

As the trend towards smaller lot sizes and higher accuracy parts continues, many manufacturers are finding that dedicated workholding systems failed to provide the versatility they need. Flexible fixtures, on the other hand, are best suited to small quantities or infrequent production runs. Flexible fixturing systems not only clamp accurately and consistently, but allow multiple machining operations to be performed in one setup, often slashing both production time and cost.

It was reported that flexible fixtures usually combine basic tooling plates and blocks with precision-machined vises. Quickly assembled into different module configurations, they are readily adapted to medium quantities of related families of parts. Combining fixtures on fixturing blocks and cubes in a machining center, these families of parts can also be machined at the same time.

1.2 Industrial practices

1.2.1 Part Family

The concept of part family arose from the need of variation design and is realized through parametric geometry model. Within a family of members, each one is an instantiation of the geometry model with dimensional variations.

In a broad sense, workpiece can be classified into one family if they have the similar solid model, similar fixtures and manufacturing processes while each individual has its own variations in terms of functional features. As in Figure 1.4, a family of knuckles share the similar functional features. They can be potentially machined with similar fixture configurations and manufacturing processes.

For the current caliper production, there could be several classes exist. They can be classified into groups by their physical properties and functional characteristics such as the number of bores, made-from-materials and connector/bleeding hole orientations. As far as the classification of calipers, different group has different classifying strategy. For the production engineering (design group), they comply with the principles of design for manufacturing.

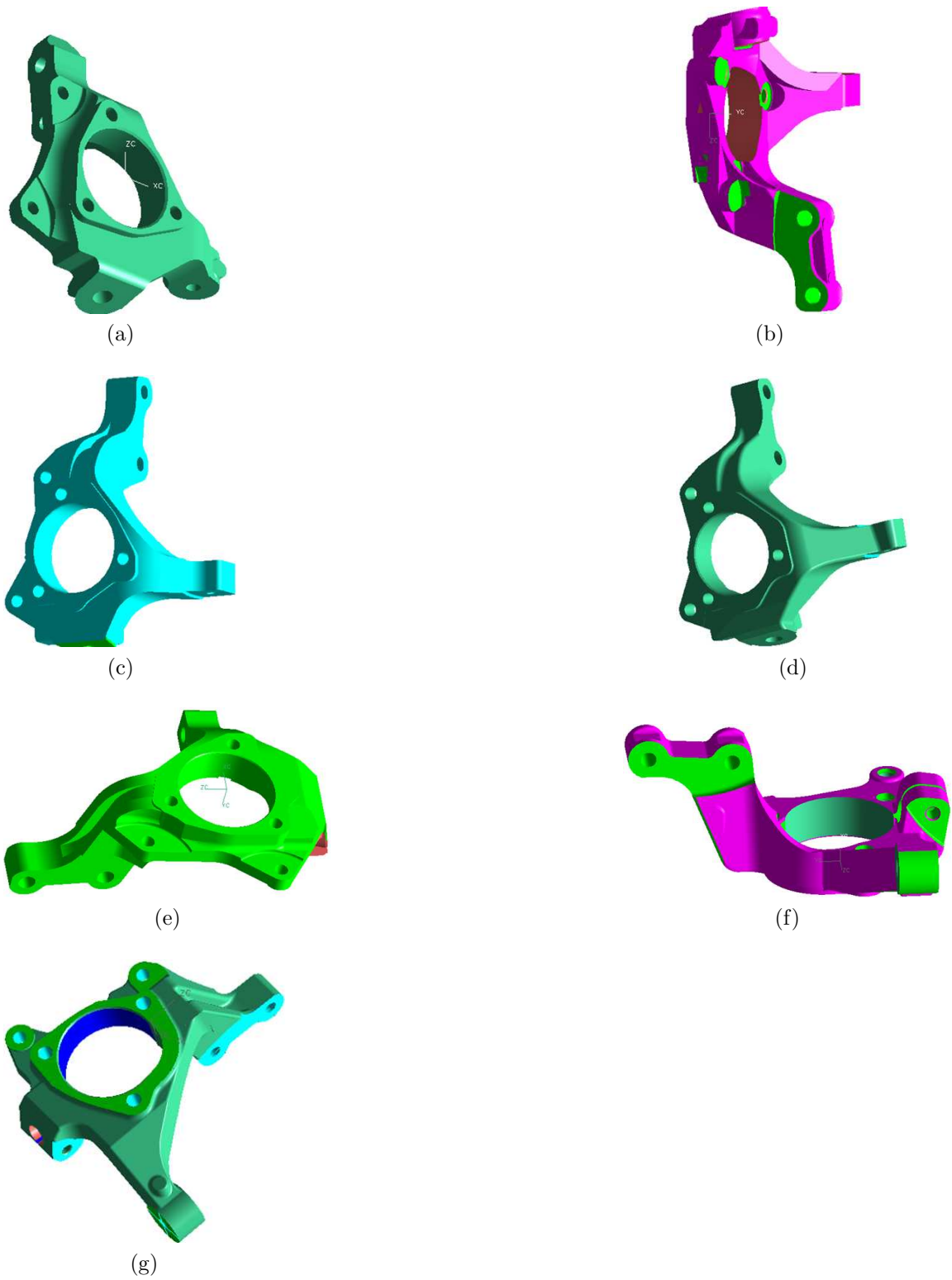


Figure 1.4: Knuckles and their variations (Courtesy of Delphi).

Production engineers classify the current caliper production into several sub-groups. For each sub-group, they define the part family according to the counterbore size. For calipers with single bore that are made from aluminum, they are divided into three families by the bore size (diameter ϕ) (Figure 1.5 and Table 1.1). Production engineers do not take account into the connector/bleeding hole orientations, which are the most important when determining the production lines.

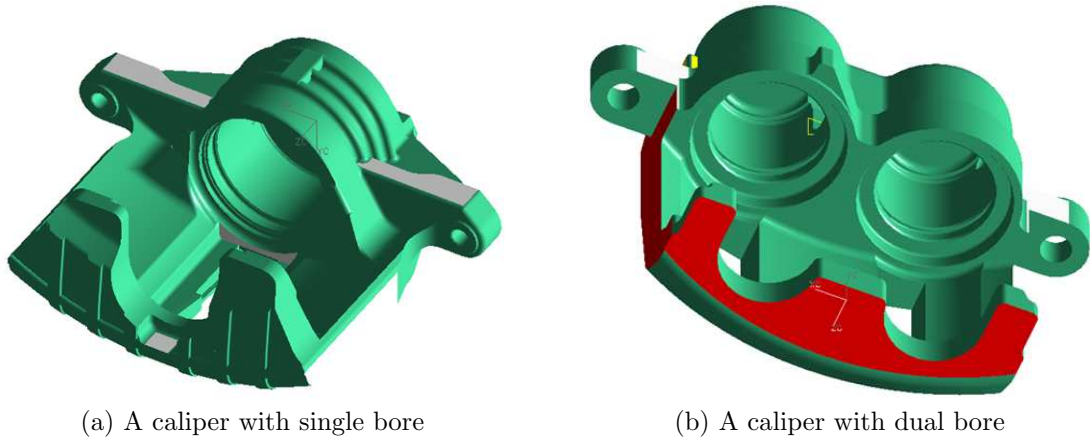


Figure 1.5: Calipers from two part families (Courtesy of Delphi).

	Aluminum			Cast iron		
Single bore	35 ~ 45	45 ~ 55	55 ~ 65	35 ~ 46	45 ~ 56	55 ~ 65
Dual bore	40 ~ 46	N/A	N/A	40 ~ 46	45 ~ 52	51 ~ 58

Table 1.1: Caliper families from production engineering (Courtesy of Delphi).

1.2.2 Current Practices and Problems

In automobile company, manufacturing facilities, including production line, fixtures, testing equipment, and other accessories are updated as the new models come out. They often rely on third parties to provide manufacturing facilities mentioned above. When the new model is well designed and documented, third parties continue to design new production line and make them ready after through investigations and researches. The nature of this practice determined that more or less the quality of new production line depends on the practice of old models and skills and experience of engineers. In the meantime, when the new production line is setup, the old ones would be put aside and most of time they are never put in use and often are discarded. As in Table 1.2, for caliper families of 540, 541, 543, 544, and 549, each has its production lines and their accessories. For example, for caliper 540 family, the machining processes are either accomplished with two setups on CNC machines Toyoda and Kitamura, or with one or two setups on Mori Seiki. Most of calipers from this family are machined with two setups - Toyoda and Kitamura, each accomplishes one setup and machine its manufacturing features accordingly (Table 1.2).

Caliper families	Fixtures	Locating/clamping description	Machining features
549 Cast iron Single bore	Gillman (F-105505)		Mounting holes, Outboard shoe face
	Moriseiki (F-105529)	shoe face and two mounting holes	Debur holes
	Kitamura (F-105537)	Round + diamond pin	BLDR/CONN holes
543 Cast iron Dual bore CONN/BLDR vertical	Toyoda (F-106952)	3-2-1 locating	Shoe face, mounting holes, Deburr
	Kitamura (F-106974)	Dual bore round + diamond pin	Rotate 90° to machine BLDR/CONN holes
	Mori Seiki		
544 Aluminum, Single bore, BLDR/CONN 30°	Toyoda (F-106890)	3-2-1 locating	Shoe face, Bur holes, Mounting holes
	Kitamura (F-106912)	MTG round + diamond pin	Horizontal position, rotate 60° to machine CONN/BLDR holes
	Mori Seiki	setup I: same as Toyoda setup II: round + diamond pin locating	
541 Aluminum Single bore BLDR/CONN horizontal	Toyoda (F-107521)	Horizontal 3-2-1 locating	Mill drill outboard MTG, Rougher, Finisher, Seal groove, Tap MTG hole
	Kitamura (F-107543)	Horizontal position, round + diamond pin locating	CONN/BLDR mill/drill, Tap BLDR/CONN holes
	Mori Seiki	one setup - similar with Toyoda but with top clamping	
540 Aluminum Single bore	Toyoda (F-107570)	Horizontal 3-2-1 locating	
	Kitamura (F-107593)	Vertical position, round + diamond pin locating	
	Mori Seiki	same as Toyoda	

Table 1.2: Current practices: setups, fixtures and machining features for calipers (Courtesy of Delphi).

In the past decades, automobile industries update models every two to three years. The volume for each model is relative high and the nature of manufacturing is mass production. The exist practice works well for the mass production. But as the market changes, automobile models update more frequently and there is more variations for each model. The exist practice will not be able to suit those changes well. Not only because it can not provide production line and its accessories timely (it used to be between two to three years for each model update. It changes to less than one year in current market), but also automobile manufactures can not afford the cost of production lines that are replaced.

In summary, the problems of exist practices are as below:

- Design a new production line for each new model
- Rely on the third party to provide a new production system for each new model
- Time consuming
- High cost
- Inconsistent solution
- Manufacturing quality control can't be well adapted to new part production

1.2.3 What's Needed

As discussed in previous sections, we conclude the needs as listed below are imperative.

- A methodology to make the best use to existing production line design that include tooling and fixtures. The experience on fixtures design accumulated through years of production has been embedded into exist fixtures and current practice. It's not likely to start from scratch given the complicate design on every aspect of a fixture.
- A single fixture that can be reused to accommodate workpiece variations from a family. With the capability of reconfigurability, a fixture can be rapidly customized to hold new workpiece model rather than redesign and rebuild a different fixture for new workpiece. Manufacturer benefits from such capability of fixtures in terms of cost, lead time, and product quality.
- A computer-aided design tool that is capable of generating new fixture configurations. Given the capability of a fixture, to integrate with other CAD/CAM functionalities a computer-aided tool is indispensable that is able to generate fixture configurations for given new workpiece rapidly.
- A computer-aided software to facilitate tooling, simulation, and verification. Fixture reusability alone can not service well in that the manufacturing is a complicated process. Often there are many other activities involved such as testing and verification.

1.2.4 Motivation

Diversified market requires manufacturers to make frequent product design changes, which demands manufacturing tooling changes accordingly. It is crucial to endow

manufacturing fixtures with capability of reconfiguration and accommodation to new models to decrease the lead time and cost.

In automobile mass production workshop, site engineers have accumulated enormous experience on working fixtures through years of machining a variety of workpiece models. For example, an appropriate locating scheme can ensure machining accuracy, a proper clamping sequence to reduce workpiece deforms, an air hole to avoid incorrect workpiece loading, etc. All of those principles and rules have been well applied in existing fixtures. One of our motivations is to make the use of current best practices and engineering techniques and integrate them into computer-aided tools .

Reusability is one of fundamental principles of flexible manufacturing systems. Reconfigurability or reusability of fixtures become the paramount objective of our research topic in that reuse of fixture design knowledge in existing fixtures can result in significant reduction of incurred costs and total production lead time. This is achieved through workpiece model, fixture model, workpiece variation model, fixture variation model and fixture base synthesis.

Last, we provide a computer-aided engineering package to facilitate manufacturing process simulation, which include fixture design, tooling and machining simulation, tolerance analysis and machining time estimation.

1.3 Genetic Algorithms

The genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimiza-

tion and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

In practice, genetic algorithms have had a widespread impact on optimization problems, such as circuit layout and job-shop scheduling. In the area of fixture design and verification, genetic algorithms have had been employed to solve problems where non-linear optimization is required such as locating schema in terms of optimum locating tolerance, clamping stability and sequence in terms of workpiece deformations, etc.

1.4 Objectives

To make the fixture with capabilities of reusable and reconfigurable so as to make a single fixture configuration can accommodate workpieces from a family becomes the paramount objective of our research. In more details, we state the objective of dissertation as below:

- To investigate methods of variation fixture design for part families in automotive industry.
- To develop models and algorithms to exploit fixture design.
- To explore integrated computer-aided simulation and engineering verification.
- To implement those algorithms and integrate with other functionalities to provide an engineering tool.

1.5 Research Approaches

Fixture bases play a central role in fixtures for mass production. Either in terms of geometry complexity or in times and cost spent on fixture design, fixture bases are much more difficult to design or derive. In the dissertation we follow two approaches as below.

First, rather than redesign a fixture base for each workpiece, a fixture base that can be used to accommodate to a family of parts is composed. This means that fixture base can be reused or reconfigured with rapid adjusting which leads to reduced time and enhanced productivity in fixture design process. To achieve this capability of fixture base, the concept of design windows is presented that describe the adjust range for each mounting feature in the fixture base. A fixture base is composed from fixture bases that have been designed to hold workpieces that come from the same workpiece family.

Two algorithms are presented to synthesize fixture bases. One of them is Genetic Algorithm (GA), which has long been used to process non-linear optimization problems. Based on GA algorithm, We present an improved algorithm for fixture base synthesis. Those algorithms generate a fixture base with design windows.

Specific mechanical design of design windows had been discussed in previous researches [An et al., 1999, An et al., 2000], it is not within the scope of this dissertation and will not be discussed.

The other approach focus on the variation of other fixture components such as supports and locators/clamps. Fixture models, workpiece models, and their variation models are presented to generate variation geometries rapidly.

1.6 Dissertation Organization

The dissertation consists of an introduction, a literature review, a basic fixture base synthesis algorithm, an improved algorithm, parametric models, and finally, a conclusion and suggestions for further work that could be performed to carry the research a further stage forward.

In Chapter 1, the fixture design in manufacturing will be discussed. This includes fixtures, bill of process and the role of fixture design in manufacturing. The current industrial practices and issues encountered in fixture design (most of these issues were identified by the on-site engineers) will also be discussed. The dissertation objectives and approaches will be present in the end of this chapter.

In Chapter 2, a literature review on fixtures, fixture design, and relative technologies will be discussed. For fixture design, the modular fixture design, dedicated fixture design, and flexible fixture design will be discussed orderly. After reviewing the technologies on fixture design, a concise review of the-state-of-the-art in fixture design and problems remained will be discussed.

Chapter 3 presents a genetic algorithm for fixture base synthesis. We present algorithm design and implementation. A case study of this algorithm and problems with this algorithms will be further discussed in detail.

Chapter 4 presents an improved algorithm for the fixture base synthesis and optimization. In this chapter optimization process to satisfy each objective functions will be fully discussed. Algorithm performance study and comparison with basic genetic algorithm present in previous chapter will be present.

Chapter 5 discusses the fixture models that serve as essential rational for the variation fixture design. Those include workpiece model, workpiece variation model,

fixture model, and fixture variation model. A complete variation process based on those models is presented and discussed in detail in this chapter.

Chapter 6 discusses case studies and integration with other functionalities will be presented. The software is a whole virtual manufacturing system aimed at validating the manufacturing process. It eventually gives estimates of labor, cost, and production line requirements. The chapter details aspects ranging from conceptual fixture design to basic validation, and from detailed fixture design to fixture verification. Every aspect of fixture design and performance verification will be discussed.

Finally, in Chapter 7 a concise summary is presented. Also, some suggestions for future work that can be performed to further develop the project are proffered.

Chapter 2

Related Work

In the past decades, fixture design that includes setup planning, fixture design, and verification had been received large amount of attention. Fixture design alone, from fundamental locating principles to automated fixture configuration generation, from essential functionalities such as locating and clamping to more advanced capabilities such as error-proofing and ergonomics, from modular fixtures to dedicated fixtures and even reconfigurable fixtures have been addressed in a variety of extents. In this chapter, We review the state of the art in the research community of fixture design, which include related algorithms and methodologies in general, modular fixture design, and dedicated fixture design. We extend to discuss technologies related to flexible and reconfigurable fixture design. As we utilize genetic algorithms and linear programming techniques to optimize and synthesize fixture base, advantages and applications of those two algorithms are also our interest.

2.1 Automated Fixture Configuration Design

Fixture configuration design should include modular fixture design and dedicated fixture design. In the area of automated fixture configuration design, a large volume of research has contributed to the modular fixture design.

2.1.1 General Fixture

The CAD of jigs and fixtures was most likely first proposed in the 1970s. An automated fixture configuration design can be classified as a rule-based and generative design in nature. A rule-based system uses a rule-base to represent the fixture expert's knowledge, then infers the desired fixture configuration. Such systems include:

One of the earliest contributions to the rule-based expert fixturing system was presented by Makus, et al. [Markus, 1984], who developed an expert system using PROLOG to design modular fixtures. The major contribution of automation is the selection of feasible sets of fixture components and towers, as well as the positioning of the fixture components on the fixture base for the fixture assembly. Rules are created to define the fixture building logic, including the shape constraints, and to evaluate possible collision for simple box-type workpiece.

Pham and Lazaro developed an interactive knowledge-based program to assist fixture designs [Pham and Lazaro, 1990]. It was a rule-based system implemented with a XI-plus expert system shell. Their system is thought to be one of most advanced rule-based systems containing more than 2400 items of rules.

Nee published a paper on the framework for an object/rule-based automated

fixture design system [Nee and Kumar, 1991]. Locating, clamping and supporting planes and points were identified using rule-based as well as mathematical analysis. The final output represented fixture-assembling sequences, and the feasibility for robotic assembly was also discussed. The authors admitted that the proposed framework was only capable of solving relatively simple cases based on modular fixture elements.

An expert fixture design system was developed for an automated manufacturing environment [Kumar et al., 1992]. The system consisted of an intelligent feature recognizer and interface of the models. Generally, the system is able to provide fixturing solutions for prismatic parts with simple machining features. The use of design interface modules for geometry analysis and expert system makes the system versatile and user-friendly. Additionally, they suggest adding vision systems for automated fixture assembly and inspection.

2.1.2 Modular Fixture

Compared with a rule-based system, the generative fixture design system used certain algorithms to reason the configuration of the fixture assembly. It concerns issues such as:

Fixture configuration: developing the types of fixture components required and selecting locating points on the selected elements according to the specified process information.

Fixture assembling: constructing and assembling fixture components. The orientation of each component on the fixture base is determined according to workpiece setup.

Fixture verification: proving the validity of the fixture configuration with considering of some operation systems.

A method was presented to automatically design the configuration of T-slot based modular fixture elements [Whybrew and Ngoi, 1990]. The key feature of the system was the development of a matrix spatial representation technique which permitted the program to search and identify both objects and object intersections. The system was also able to determine the position of the objects during the design process. However, the limitation of the method was that only the blocks whose edges were parallel or perpendicular to each other could be represented. Therefore, the design system could only lay out the fixture elements in such a way that all the edges of fixture elements were parallel or perpendicular to each other.

In computer aided modular fixture design, two research works have received considerable attention. They are Goldberg's algorithm for synthesizing modular fixtures for polygonal workpieces and Rong's automated modular fixture design system [Brost and Goldberg, 1996, Rong and Bai, 1997].

In Brost and Goldberg's work [Brost and Goldberg, 1996], an algorithm was presented for synthesizing planar modular fixtures for polygonal workpiece. The basic assumptions were that a workpiece can be represented with a simple polygon, locators can be represented as circles with identical radii that are less than half the grid spacing of fixture base plate, the fixturing configuration will be three circular locators and a clamp, the base plate is infinite, and all the contacts are frictionless. In addition to polygonal workpiece boundaries a set of geometric access constraints are provided as a list of polygons with clamp descriptions and a quality metric. The output of an algorithm includes the coordinates of three locators, one clamp, and

the translation and rotation of the workpiece relative to the base plate.

The algorithm begin with enumerating the set of possible locator positions. The set of candidate fixtures are then filtered to remove those that cause problems, i.e., collision. The survivors are then scored according to a quality metric and are output in an order that favors such quality metric.

Placement of three circular locator on the base plate are further evaluated while translating and rotating the workpiece relative to the base plate. An algorithm was also presented to find all combinations of the three edges, where two of them may be identical, on the polygon with a satisfaction of hole-alignment conditions with the base plate. For each set of locators and associated contact edges, consistent workpiece configurations or workpiece positions are calculated. All the possible clamp positions are then enumerated based upon the constraint analysis of the constructed force sphere.

The algorithm is claimed as a complete algorithm for planar modular fixture design because it guarantees finding all possible planar fixture design for a specific polygonal workpiece if they do exist. However, the major limitations of the algorithm exist:

- 1) Only polygonal workpiece are considered, i.e., no curved surface are allowed in the workpiece geometry. In reality, many fixture design cases include cylindrical surfaces or circular arcs in $2 - D$ representations.

- 2) Only circular locating pins with uniform radii are considered in the algorithm. In each modular fixture configuration, there are other types of locators that are widely used in fixture designs.

- 3) The algorithm only considers $2 - D$ workpieces. In practice, it can be applied

only for prismatic workpieces with small height. A large volume of workpieces are in 3 – D with complicated manufacturing features.

4) There are other criteria necessary for locating and clamping design in addition to geometric considerations. It may include locating error, accuracy relationship analysis, accessibility checking and other operational conditions which were not considered.

5) Clamp location planning is weak without the consideration of friction forces. This needs to be further improved.

After this research work, other related issues were further investigated and studied. The existence of modular fixtures design solutions for a given fixture configuration model and a workpiece was explored [Zhuang et al., 1994]. Fixture foolproofing for polygonal workpiece was studied [Penev and Requicha, 1994], and partially employed the Brost and Goldberg approach [Brost and Goldberg, 1996]. A framework on automatic design of 3 – D fixtures and assembly pallets was presented, but no detail design methodology, procedure and results were provided [Brost and Peters, 1996].

A geometric analysis for automated fixture planning has been presented, which is an expansion of Goldberg's research on automated fixture configuration design and 2 – D geometric synthesis [Wu et al., 1998b, Wu et al., 1998a]. Cylindrical surfaces, different types of locating components, and 3 – D fixture configurations have been considered in the analysis. A comprehensive automated fixture planning and configuration design system has been developed where analyses of locating accuracy, geometric accessibility, clamp planning, and fixture design stability are all investigated.

An automated modular fixture design system FIX-DES was developed. The system has a representation strategy of modeling modular fixture components and automatically constructing the modular fixture component assembly relationship database (MFEARDB). In this database, the assembly features of modular fixture components are used to describe the components. Mating conditions between components are defined for the possible fixture units. On the other hand, the assembly relationship graph model between fixture components was developed. Based on this model, algorithms were implemented to choose all the suitable fixturing unit candidates and mount the fixture units on the baseplate [Rong and Bai, 1997, Rong et al., 1993, Rong and Bai, 1996].

Other approaches have been proposed to automate the modular fixture design process. A projective spatial occupancy enumeration (PSOE) approach for developing a fixture configuration automatically was presented by Trappy [Trappy et al., 1993]. Alternatively, Chou employed a geometric reasoning method to determine the location and clamping points [Chou and Barash, 1990].

2.1.3 Dedicated Fixture

For the dedicated fixture design, so far it seems that not much research has been done. However, an AFD system was developed by Chou [Chou and Barash, 1990]. The system takes geometry solid data from a CAD system, reasoning with geometry data and other operation data, and generates fixture configuration automatically. Issues related to automatic design, such as completeness and computing efficiency, are also discussed. This system also describes the design process as three stages: conceptual design, preliminary design, and detail design. The detail design is to se-

lect fixture components from a component library for implementing fixturing points determined in the preliminary design. The components are indexed by their space requirements, fixturing function, and application domain.

An automated customized fixture design system was also introduced [Wu, 1996]. Based on the fixture structure analysis, fixture configurations are divided into functional components, fixture base, and supports. A geometry element generator is designed to adaptively generate fixture component types and dimensions according to workpiece geometry and operational information. Locator/clamp selection is automatically conducted with connections to a fixture base. Once fixture units are designed individually for each locating/clamping point, the connection may be modified into combined units based on certain criteria. Interference checking and fixture unit modification is also performed in the system.

A technique of automated dedicated fixture configuration design is studied with predefined fixture component types [An et al., 1999, An et al., 2000]. The design methodology is divided into two stages: basic design and detail design. The basic design activities include 1) selection of functional fixture components such as locators and clamps from a standard fixture component database, 2) generation of customized supports with variable dimensions for different fixture design requirements, and 3) assembly of fixture components into a final configuration on a fixture base. In order to implement the fixture design procedure, models are developed to represent the standard fixture components and customized supports. The assembly relationships among fixture components are established based on a compatibility analysis. The detail design includes fixture unit combination, connection design, interference avoidance modification, and technological-rule-based modification.

2.1.4 Flexible Fixture Systems

As the trend towards smaller lot sizes and higher accuracy parts continues, many manufacturers are finding that dedicated workholding systems just don't provide the versatility they need. Flexible fixture systems, which can be quickly adjusted to adapt to new workpieces, is a good candidate for such variety workpiece models due to its advantage of reusable or reconfigurable nature. The system's quick-change tooling also boosted production by allowing varied configurations to run on existing equipment. An entire system can be quickly assembled and reassembled with reusable clamping units to accommodate changing shapes and sizes of workpieces. Operators swap clamping units between machining centers and quickly "retool" by simply changing vise jaws.

Commercially-available reconfigurable fixtures, used for holding compliant sheet metal, composite and plastic parts during secondary machining operations, are extremely expensive and overly-complicated devices. A computer-controlled, reconfigurable fixturing device (RFD) concept for compliant parts, based on a matrix of individually-stoppable pins lowered by a single rigid platen, has been developed as a simple and low-cost design alternative to commercially-available devices [Walczyk and Longtin, 2000]. Two different approaches to stopping and clamping individual pins have been investigated: a combination electromagnet assist and gas springs compressed with a toggle mechanism, and a pneumatic clamp. Simple mechanical models have been developed for predicting the stopping and clamping performance of both designs including pin positioning accuracy, vertical load-carrying capacity of a pin, and deflection of a pin subjected to lateral loads. An RFD prototype, consisting of a single pin actuated by a servoed platen, has been designed,

built and tested. It has demonstrated the feasibility of this new RFD design.

Reconfigurable Machine Tools (RMTs), assembled from machine modules such as spindles, slides and worktables are designed to be easily reconfigured to accommodate new machining requirements [Gopalakrishnan et al., 2002]. Their goal is to provide exactly the capacity and functionality, exactly when needed. In this paper, we present a novel parallel-actuated work-support module as a part of an RMT to meet the machining requirements of specific features on a family of automotive cylinder heads. A prototype of the proposed module is designed/built and experimental results regarding its performance are presented.

For fixtures to hold stampings where a $N - 2 - 1$ locating schema is usually used, a system that fixture designers can use to synthesize flexible fixture work spaces for a set of different stampings was presented [Lee et al., 1999, Lee, 1995]. In particular, a fixture robot workspace is represented by a circle and a candidate locator region is represented by the vertexes of a polygon. An algorithm scheme is developed to find the optimal arrangement of stampings and workspace sizes and centers, and this algorithm scheme is computationally implemented by employing Grefenstette's program. The workspace synthesis system is tested with lab data and appears to be effective. The system is limited to cylindrical workspaces.

In publications [Kong and Ceglarek, 2006, Kong and Ceglarek, 2003], they presents an approach of fixture workspace synthesis for a family of sheet metal workpiece, which is critical for design of fixtures with reconfigurable tooling elements. The proposed approach applies analytical Procrustes analysis to narrow the search domain for variables and uses pairwise configuration optimization to identify the final solution based on various engineering requirement functions. The approach can

rapidly and accurately find the solutions that minimize the fixture workspace for a family of parts with much less computational complexity than the existing method in the literature. The case studies compare the proposed approach with the existing method. The results demonstrate that, overall, the proposed method can obtain better solutions over the existing method by achieving greater improvement in both algorithm efficiency and solution accuracy.

The latest news on the flexible-fixturing front comes from Lamb Technicon Division of Unova Inc. The company has been working with General Motors, Ford, Chrysler, and Cummins Engine on an Advanced Technology Program funded by the National Institute of Standards and Technology. The Intelligent Fixturing System project aims to create the next generation of flexible fixturing systems for high-volume production of automotive parts [Smith, 1998, Meter et al., 2001a, Meter et al., 2001b].

It was reported that they were working with Pennsylvania State University, the University of Illinois, Georgia Tech, and the University of Michigan to develop sub-systems for our fixturing concept. Once these sub-systems are developed, Lamb will integrate, test, and debug the fixture configuration stations.

In this IFS there are four basic sub-systems in the IFS concept: flexible clampings, a part locator, a micropositioner, and a software support system.

The flexible clamping system, when finished, should be able to clamp a very large family of parts in a single fixture. What will set this system apart is its rigidity. In general, increased flexibility means a decrease in rigidity, which is crucial to the machining operation. The goal is flexibility while maintaining above-standard stiffness.

The part location system is intended to accurately define the part whether it is an as-cast part or one that's already been machined and sense its location. This might be done by touching it or possibly using a vision system, The part can be put into the fixture without hard locators, which are used in traditional fixturing.

Once a part is clamped and the system senses its location, a manipulator will line up the fixture with the machine axis. It was expected that actuators adjusting the entire pallet fixture to the correct location will comprise the manipulating subassembly.

Also a software support system will not only control the IFS, but also contain knowledge of pre-existing process models. With this data the software can determine an optimum part clamping position.



Figure 2.1: An intelligent fixturing system from Lamb Technicon Division of Unova Inc [Smith, 1998].

2.2 Searching Optimization Algorithms

Genetic Algorithm, Simulated Annealing and Linear Programming are three major search optimization algorithms that were used to solve optimization problems in engineering [Goldberg, 1989, Davis, 1991, Russell and Norvig, 2003, Deb, 1997].

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. GAs were introduced as a computational analogy of adaptive systems. They are modelled loosely on the principles of the evolution via natural selection, employing a population of individuals that undergo selection in the presence of variation-inducing operators such as mutation and recombination (crossover). A fitness function is used to evaluate individuals, and reproductive success varies with fitness.

Genetic Algorithm has been widely used in fixture design, which include The optimisation of the locations of active (clamp) and passive (locator/support) elements in the workpiece-fixture system using genetic algorithm (GA) with ANSYS parametric design language (APDL) of finite element analysis [Kumar and Paulraj, 2010]. A real-coded genetic algorithm (RGA) proposed to resolve the optimization problem which should simultaneously infer the suitable mechanisms, satisfy multiple complex constraints, and achieve the cost-minimum requirement [Liu et al., 2008]. A genetic algorithm based approach developed to optimise fixture layout through integrating a finite element code running in batch mode to compute the objective function values for each generation [Kaya, 2006]. A Genetic Algorithm with Learning Automata (GALA) algorithm, which is a population based interconnected learning automata algorithm incorporating genetic operators [Choubey et al., 2005]. and a genetic algorithm (GA)-based optimization method to select automatically the opti-

mal numbers of locators and clamps as well as their optimal positions in sheet-metal assembly fixtures, such that the workpiece deformation due to the gravity effect and resulting variation due to part dimensional variation are simultaneously minimized [Liao, 2000].

Simulated annealing (SA) is a generic probabilistic meta-heuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global optimum of a given function in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more effective than exhaustive enumeration provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

Each step of the SA algorithm replaces the current solution by a random “nearby” solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter T (called the temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly “downhill” as T goes to zero. The allowance for “uphill” moves saves the method from becoming stuck at local optima which are the bane of greedier methods.

In [Lin and Huang, 1997] Simulated Annealing was used to select the required fixture elements and to derive a suitable and economical number of fixture element combinations. The publication [Wang, 2000] focused on the fixture layout problem and discussed an approach based on a technique of optimal pursuit, which allows quickly generation and analyses of feasible fixture layout designs and ultimately determine an overall optimum solution.

Linear programming (LP) is a mathematical method for determining a way to achieve the best outcome (such as maximum profit or lowest cost) in a given mathematical model for some list of requirements represented as linear equations. More formally, linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. Given a polytope and a real-valued affine function defined on this polytope, a linear programming method will find a point on the polytope where this function has the smallest (or largest) value if such point exists, by searching through the polytope vertexes.

Linear programs are problems that can be expressed in canonical form:

$$\begin{aligned} & \textit{maximize} \quad c^T x \\ & \textit{subject to} \quad Ax \leq b \end{aligned}$$

where x represents the vector of variables (to be determined), c and b are vectors of (known) coefficients and A is a (known) matrix of coefficients. The expression to be maximized or minimized is called the objective function ($c^T x$ in this case). The equations $Ax \leq b$ are the constraints which specify a convex polytope over which the objective function is to be optimized. (In this context, two vectors are comparable when every entry in one is less-than or equal-to the corresponding entry in the other. Otherwise, they are incomparable.)

Linear Programming has been used to optimization issues in fixture planning such as clamping stability, minimum clamping force and optimum locating position. In [Zhu and Ding, 2007] it presented an efficient algorithm for grasp synthesis and fixture layout design in discrete domain. Given N candidate contact points on the

surface of a $3 - D$ object, the algorithm determines a minimal subset from the candidate points so that they construct a grasp or a fixture with the form-closure property by solving a single linear program.

An algorithm presented a stability test and a new approach to automatically generating the positions of a small set of fixture elements (fixels) that will stabilize an assembly by using linear programming techniques [Wolter and Trinkle, 1994]. The fixture verification system is modelled as a linear optimization problem with respect to minimum clamping forces [Liu and Strong, 2002]. A linear programming method is proposed for stability analysis of the workpiece [Qin and Zhang, 2007]. In [Lin et al., 1999] mixed-integer quadratic programming is used to identify the optimal distribution and position (i.e., topology) of locators in order to minimize the mean compliance of the workpiece.

2.3 Summary

Problem remained:

- Generative dedicated fixture design was based on basic parametric model, which is not suitable for complex fixtures due to geometrical complexity and various engineering requirements.
- Researchers did not make use of the best practice of fixture design knowledge, which is accumulated through years and embedded in precious fixture design of similar parts.
- Lack of CAD tools that treat fixture design, verification and machining simu-

lation simultaneously, and can be used in production environment.

- Operational requirements of fixture design are not considered, such as error proofing, pre-loading, and pre-clamping.

Chapter 3

A Genetic Algorithm for Fixture Base Synthesis

In automotive industry, fixtures conventionally have been dedicated, producing each model in high volume. In recent years, however, the automobile industry has been changing from high volume to small-to-medium volume production per model with an increasing number of models because customer tastes are diversifying. In other words, automobile companies need to cope with fast market changes. As a result, they increasingly rely on reconfigurable production lines that can manufacture a variety of vehicles in small-to-medium volume. Unlike dedicated production lines that can manufacture only one vehicle type, reconfigurable fixtures are essential for such production lines.

According to aforementioned analysis, any variation of a component within a fixturing unit - either the variation of fixture base and support, or the variation of final locating/clamping elements, eventually reflects on the variation of locating/clamping position or orientation. However, it still spends time to manufacture variant components and rebuild the whole fixture configuration even with the help of fixture design tools. Fixture reconfigurability, which enables a family of products

to be produced on a single production line, is becoming paramount importance to cope with increasing diversification of the part models.

With the capability of reconfigurability, fixture configuration can be adjusted to accommodate a given workpiece rather than to rebuild. Another benefit of using reconfigurable fixtures is that automobile companies can reduce cost and time for fixture preparation because it requires high cost and long lead time to rebuild the dedicated fixtures whenever there are new model changes. For example, many automobile companies replace current models every two to five years. Building new dedicated fixtures can cost millions of dollars and can take many months in time.

To address the reconfigurability of dedicated fixture for automobile parts, one of the key issues in reconfigurability is to enable the capability of reconfigurable fixture base. That means to provide the capability of variation in terms of position of mounting features for each fixture unit. When designing a fixture base, if the mounting features for each fixturing unit were designed with adjustable position on fixture base, then it can be readily adjusted to fit the new workpiece models whenever there are such requirements.

3.1 Fixture Base in Mass Production

In manufacturing of automotive parts, often production lines and fixtures are dedicated. Fixture bases are normally dedicated. To cope with reconfigurability or flexibility of fixtures, fixture bases are proved to be the best components that could be designed with such capability. In this section, we discuss the concept of design windows, which include the design windows for workpiece, and such concept applied

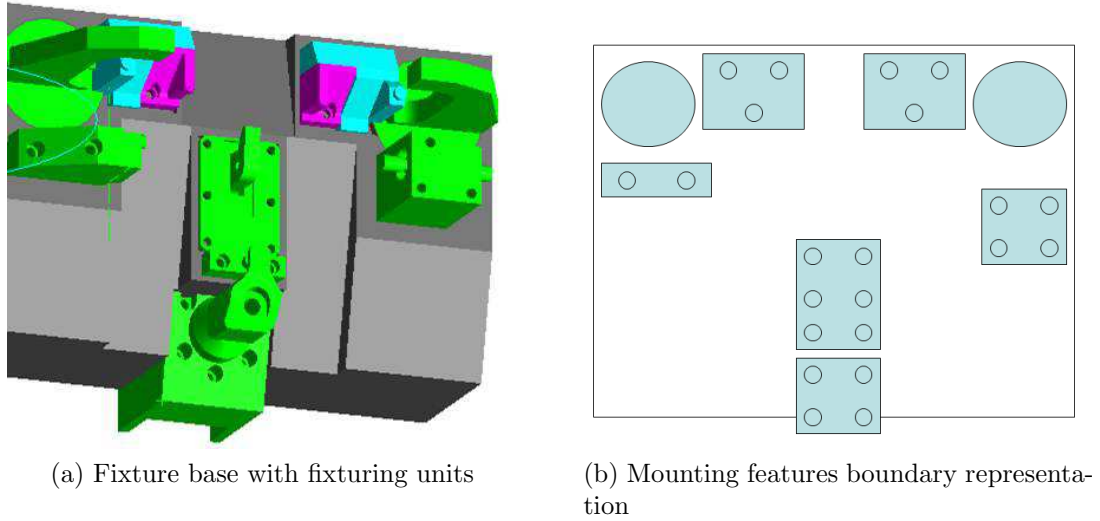


Figure 3.1: An example of fixture base and its mounting features.

on fixtures and fixture bases.

3.1.1 Fixture Base and its Representation

As discussed in previous sections, a fixture base works as a base plate on which all fixturing units can be placed and mounted. For simplicity, in this section we use the central point of a mounting feature to represent its position. For example, the center of a circle, the cross point of two diagonals of a rectangular and the centroid for other boundary shapes.

Figure 3.1 displays an example of fixture base, fixturing units and boundaries of mounting features. Figure 3.1(a) displays a half of fixture base with fixturing units. Figure 3.1(b) shows boundaries of mounting features.

3.1.2 Design Windows of Fixture Base

3.1.2.1 Design Windows of Workpiece

Design for Manufacturing (DFM), has been the practice for manufacturing. DFM describes the process of designing or engineering a product in order to reduce its manufacturing costs. DFM will allow potential problems to be fixed in the design phase which is the least expensive place to address them. The design of the component can have an enormous effects on the cost of manufacturing. Used correctly, DFM can lead to a 25–30% reduction in production cost without capital investment in new facilities [Herrmann and Cooper, 2004].

Based on the DFM, automotive company proposed the concept of Design Windows (DW). For each function/manufacturing feature, they clearly specified the range of its variation. For example, in Figure 3.2 they specified variation ranges for four features. in *3D* model of Figure 3.2a a couple of *3D* solid models from a caliper family are placed coincidentally and green boxes specify variation ranges of four different features. Figure 3.2b is an example of its *2D* dimension specification.

Either workpiece designers or fixture engineers could potentially benefit from a well defined workpiece with design windows. From one side, fixture engineers can provide a window size for each machining feature variation of a workpiece. This could feedback to the product engineer as a guideline to design. In the other side, for a given part, its similarity can be analyzed, and the current fixture whose workpiece is the closest to the part can be searched. The variation design for the given part from the existing fixture is readily derived. In short, the capability for the current existing fixture needs to be evaluated.

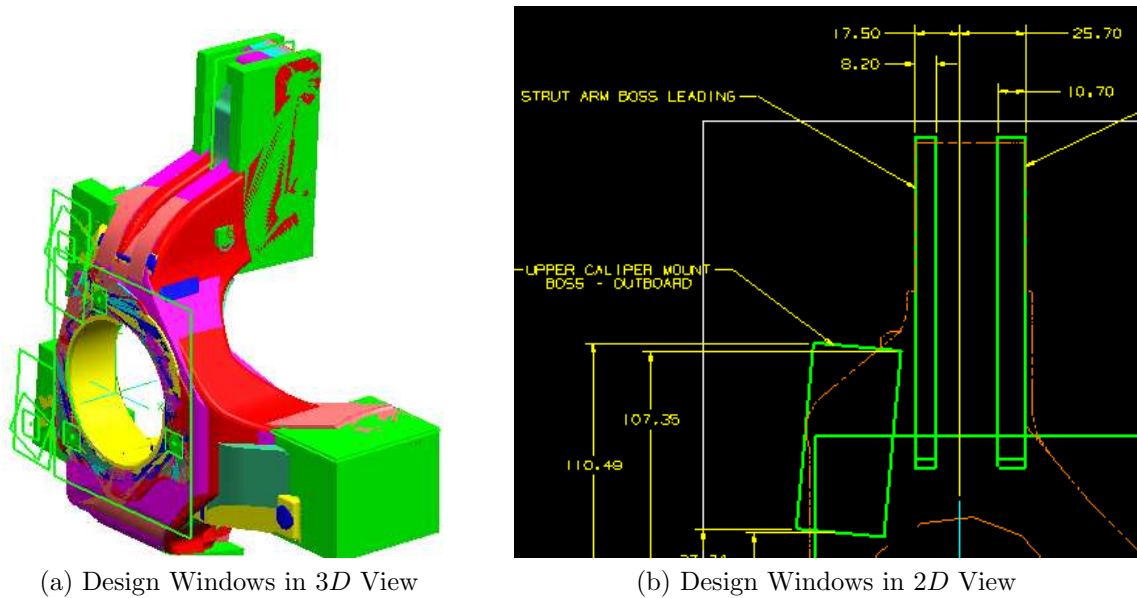


Figure 3.2: Design windows for manufacturing features of calipers.

To accommodate the workpiece from a part family with well defined design windows, fixture base with capability of variations for each mounting feature is required. This directly requires the design windows for fixture base.

3.1.2.2 Design Windows of Fixture Base

Design windows on fixture base reflect the variation of location of mounting features. Boundary centroid \vec{P} of each mounting feature is used to represent location on the fixture base. As the variation of each mounting feature, the centroid of its boundary is changed accordingly. We could use a minimized circle to cover all those points that represent centroids of boundaries and their variations. The diameter (ϕd) is used to represent the size of design window for each mounting feature. Design window for a mounting feature on fixture base can be represented as,

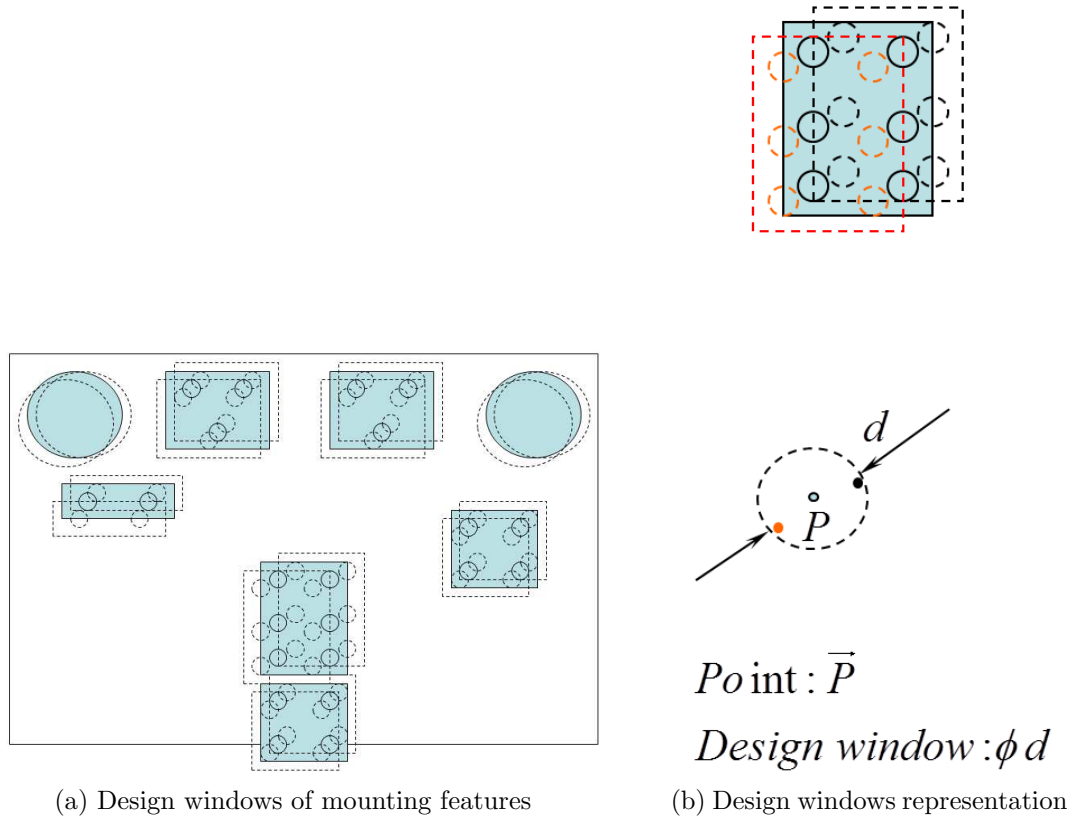


Figure 3.3: Design windows for fixture base.

$$DW = \{\vec{P}, \phi d\} \quad (3.1)$$

Figure 3.3 shows variations of mounting features on fixture base and design window for a mounting feature. Figure 3.3(a) shows boundaries of mounting features and their variations. Figure 3.3(b) The boundary of a single mounting feature and its variations is shown in the upper picture. Design window representation of this mounting feature is shown in the lower picture.

With the definitions of design windows for workpiece and fixture base, we ad-

dress how to synthesize the design windows of fixture base mounting features for M different workpieces from a family of the similar locating and clamping scheme. If a fixture is assumed to have N candidate locator and clamp regions for the N locators and clamps and we position M workpiece together on a flexible fixture base plate, we will have $N \times M$ candidate locator and clamp regions for M workpieces. If we combine M candidate locator and clamp regions into one combined region for each locator or clamp, we will have N combined regions. For these N combined regions, we want to use a flexible fixture that has N fixturing units. In this setting, there are three problems in synthesizing design windows for mounting features on the fixture base: 1) how to represent design windows and candidate locator and clamp regions, 2) how to find a workpiece arrangement that enables us to use the smallest possible mounting features without overlapping between different mounting features, and 3) locations of mounting features to accommodate a new designed workpiece. The first problem is solved in section 3.1.2, and the remaining two problems are solved in following sections.

3.2 Engineering Requirements and Optimization Objectives

Engineering requirement for fixture base is typically different case by case. For example, in a scenario where the space for placing fixture is limited, the ultimate objective of fixture base design is to minimize the size of fixture base. While in other situations, a specific fixture unit is more difficult to implement in terms of capability of variations, to minimize the size of this specific fixture unit would be

objective.

Fixture base configuration is defined as a set of points on a fixture base that is represented by matrix $k \times m$ in Cartesian coordinates, that is, k mounting features each defined in m dimensional space. In the case of fixture base synthesis, $k \times 2$ or $k \times 3$ is considered, which corresponds to $2 - D$ or $3 - D$ part fixturing layouts, respectively.

In a general case of fixture synthesis, it is assumed that a given part family has N parts, and each part has k locating /clamping points. For each fixturing unit, its design window is defined as a circle /sphere F_i ($i = 1, 2, \dots, k$) that encloses the N corresponding locating /clamping points of each part within the part family (Figure 3.4 depicts a $2 - D$ case with $N = 3$ and $k = 5$).

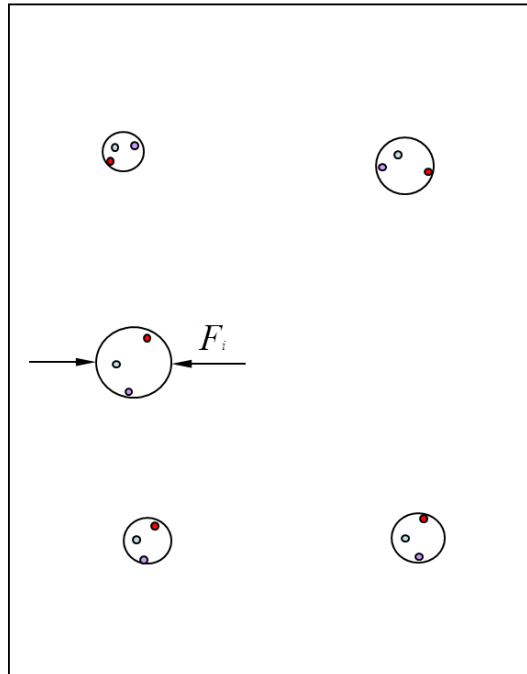


Figure 3.4: Design window of fixture base synthesis.

In general, we summarize those three objective functions base on the current practice.

(1) Minimize the DW of the fixture base with the largest DW within a given fixture. This engineering requirement function allows for application of the fixture base with the relatively smallest reconfigurability, which results in relatively cheaper solution. This requirement can be expressed mathematically as minimizing the largest diameter of F_i ($i = 1, 2, \dots, k$).

$$Min(Max(F_i)) \quad i = 1, 2, \dots, k \quad (3.2)$$

Let $(F_{max})_{min(max)}$ be defined as the largest circle among the k circles obtained based on Equation 3.2. Then, during fixture design, the diameter of the DW for a given reconfigurable fixture unit must equal to or greater than $(F_{max})_{min(max)}$ to ensure that each corresponding locating/clamping pint cluster can be enclosed within the DW of the reconfigurable fixture unit.

(2) Minimize the total DW for all fixture units within a given fixture. Correspondingly, the engineering requirement function can be expressed as:

$$Min(\sum_{i=1}^k F_i) \quad (3.3)$$

which is to minimize the overall DW for all reconfigurable fixture units.

(3) Minimize DW of a specific fixture unit and simultaneously minimize the total DW for all fixture units. The engineering requirement function of this case can be formulated as:

$$Min(Max(F_i)) \quad \text{and} \quad Min(F_s) \quad (3.4)$$

where s indicates a specific fixture unit.

Fixture base synthesis for a family of parts can be accomplished intuitively by the following approach ($2 - D$ case): first, fix one part and then rotate and translate the other $N - 1$ parts to meet one of the engineering requirement functions represented by Equation 3.2-3.4. This approach requires simultaneous optimization of $3 \times (N - 1)$ variables because there are two translations and one rotation for each part in $2 - D$ space. However, as discussed in section 3.3.1, this approach has two disadvantages: (1) the search domain for all of the variables is quite large, and (2) there are a large number of variables that need to be simultaneously optimized, especially when dealing with a large number of parts that are arbitrarily placed in space.

3.3 Fixture base Synthesis Algorithms

3.3.1 Genetic Algorithms

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encoding are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algo-

rithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires: (1) a genetic representation of the solution domain, (2) a fitness function to evaluate the solution domain. A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, inversion and selection operators.

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be “seeded” in areas where optimal solutions are likely to be found.

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming.

The next step is to generate a second generation population of solutions from

those selected through genetic operators: crossover (also called recombination), and /or mutation as displayed in Figure 3.5.

For each new solution to be produced, a pair of “parent” solutions is selected for breeding from the pool selected previously. By producing a “child” solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its “parents”. New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more “biology inspired”, some research suggests more than two “parents” are better to be used to reproduce a good quality chromosome.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

3.3.2 Point Transformations

We show how we can compute the new point coordinates of a candidate locator region after moving a fixture base in two translations and one rotation. For example, we define the points of candidate fixturing unit 1 of fixture base A as P_{A1} , the points of fixturing unit 2 as P_{A2}, \dots , and the points of fixturing unit 6 as P_{A6} . In a similar way, we define $P_{B1}, P_{B2}, \dots, P_{B6}, P_{C1}, P_{C2}, \dots, P_{C6}$ for fixture base B and C. If we move fixture base A by h_A, v_A , and θ_A , we can compute a transformation matrix T_A using Equation 3.5. The values of h_A, v_A , and θ_A are determined by

```

function Genetic_Algorithm(population, Fitness_FN)
    returns an individual
    inputs: population, a set of individuals
             Fitness_FN, fitness function of an individual
    repeat
        new_population  $\leftarrow$  empty set
        loop for i from 1 to size(population) do
            x  $\leftarrow$  Random_selection(population, Fitness_FN)
            y  $\leftarrow$  Random_selection(population, Fitness_FN)
            child  $\leftarrow$  Reproduce(x, y)
            if (small random probability) then child  $\leftarrow$  Mutate(child)
            add child to new_population
        population  $\leftarrow$  new_population
    until some individual is fit enough, or enough time has elapsed
    return the best individual in population according to Fitness_FN

```

```

function Reproduce(x, y) returns an individual
    inputs: x, y, parent individuals
            n  $\leftarrow$  Length(x)
            c  $\leftarrow$  random number from 1 to n
    returns Append(Substring(x, 1, c), Substring(y, c + 1, n))

```

Figure 3.5: A genetic algorithm.

the genetic algorithm. Here, we use homogeneous coordinates because they allow the translations and rotations to be represented by a single matrix. As an example of the homogeneous coordinate representation of a point, Equation 3.6 shows the representation of the i -th point of fixturing unit 1 of fixture base A. A new point (P') of fixture base A after its movement can be computed using Equation 3.7. Similar equations can be derived for fixture base B, whose transformation matrix is composed of h_B , v_B , and θ_B . If we define P'_{A1} , as the new points of candidate fixturing unit 1 of fixture base A, then $P'_{A1} = T_A \times P_{A1}$. The points of the other candidate fixturing unit positions can be computed in a similar way.

$$T_A = \begin{bmatrix} \cos \theta_A & -\sin \theta_A & h_A \cos \theta_A - v_A \sin \theta_A \\ \sin \theta_A & \cos \theta_A & h_A \sin \theta_A + v_A \cos \theta_A \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$P_A = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (3.6)$$

$$P'_{A1} = T_A \times P_{A1} \quad (3.7)$$

After computing the new coordinates of all the points for moved fixture base, we combine all the points of the candidate fixturing unit positions of the same fixturing unit number into one set because these points must be covered by one fixturing unit. For example, the new points (P'_{A1}) from A1, the new points (P'_{B1}) from B1, and the points (P'_{C1}) of C1 are combined into one set of points. Here P_{C1} stays the same because fixturing unit C was designated as stationary. In total, there are six sets of

points for six fixturing units. In general, M fixture bases will have N sets of points for N fixturing units if each fixture base has N fixturing units.

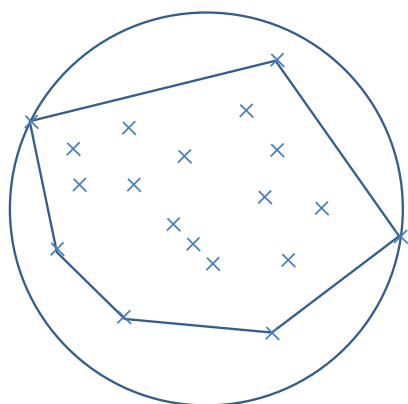
With each set of points we can find the smallest circle that enclose all points of the set. This smallest circle is commonly called the minimum spanning circle. The following section 3.3.3 describes the algorithm in detail.

3.3.3 Minimize Cover Circle

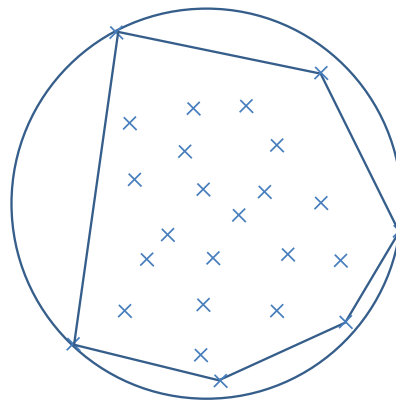
There are many published research results on the minimum spanning circle for a given set of points. These results are thoroughly reviewed by Preparata and Shamos [Preparata and Shamos, 1988]. We briefly review three of them here. First, Rademacher and Toeplitz [Rademacher and Toeplitz, 1957] present an algorithm that runs in $\mathbf{O}(N^4)$. They compute all the circles defined by either two points or three points and choose the smallest circle that encloses all the points. Second, Shamos and Hoey [Shamos, 1975] present an algorithm that runs in $\mathbf{O}(N \log N)$. They construct the farthest point Voronoi diagram in $\mathbf{O}(N \log N)$ and find the two diametrical point circle or the three circumference point circle in $\mathbf{O}(N)$. Third, Megiddo [Megiddo, 1983] presents an algorithm that runs in $\mathbf{O}(N)$. He transforms a quadratic minimization problem (the distance equation between the center and a circumference point is quadratic) into a linear problem and solves the linearized minimization problem employing the simplex method.

In our approach, we take a variation of Rademacher and Toeplitz's algorithm because the number of points in our problem setting is small (tens of points) and it is very easy to program this algorithm. We use three steps in finding the minimum spanning circle. First, we find the convex points [Graham, 1972, O'Rourke, 1995]

for a given set of points because the minimum spanning circle goes through convex points as shown in Figure 3.6. In other words, we can reduce the number of search points by considering convex points only. Second, we find a circle formed by the two points as the circle diameter that are farthest among the convex polygon points. Third, we check whether or not all the convex polygon points are enclosed by this circle. If enclosed, we have found the minimum spanning circle. If not enclosed, we compute all the circles that can be defined by three points out of the convex polygon points, and choose the smallest circle that encloses all the convex polygon points.



(a) Minimum spanning circle defined by two points as diameter



(b) Minimum spanning circle defined by three points

Figure 3.6: Minimum spanning circle of a set of points.

In our approach to finding the minimum spanning circle, it is simple to construct a circle with two points as the circle diameter. However, the construction of a circle that goes through three points requires several preliminary steps. We describe how to construct this circle using Figure 3.7, where A , B , and C are the three points. First, construct a perpendicular line (PO) through the mid-point (P) of line segment AB . Second, construct a perpendicular line (QO) through the mid-point (Q) of line

segment BC . Third, find the intersection point (O) between lines PO and QO . Last, construct a circle that has the center at O and the radius as the distance between point O and point A . Then, the circle also goes through points B and C . This can be easily proved from four right angle triangles OPA , OPB , OQB , and OQC . By the Pythagorean theorem, $OP^2 + PA^2 = OA^2 = OP^2 + PB^2 = OB^2 = OQ^2 + QB^2 = OQ^2 + QC^2 = OC^2$. Therefore, $OA = OB = OC$.

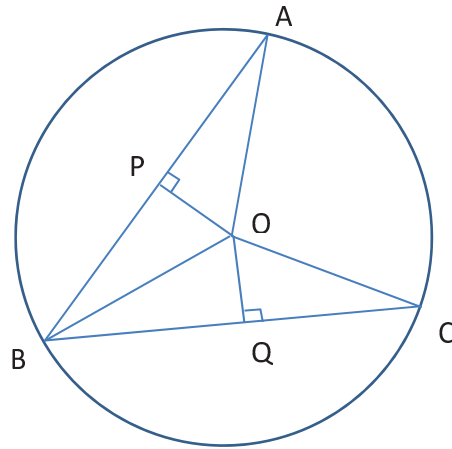


Figure 3.7: A circle passing through three points (not on the same line).

3.4 Algorithm Implementation

Genetic algorithm (GA) is an iterative optimization procedure. Instead of working with a single solution in each iteration, a GA works with a number of solutions (collectively known as a population) in each iteration. A flowchart of the working principle of a simple GA is shown in Figure 3.8. In the absence of any knowledge of the problem domain, a GA begins its search from a random population of solutions. As shown in the figure, a solution in a GA is represented using a string coding of

fixed length. We shall discuss about the details of the coding procedure a little later. But for now notice how a GA processes these strings in iteration. If a termination criterion is not satisfied, three different operators - reproduction, crossover, and mutation - are applied to update the population of strings. One iteration of these three operators is known as a generation in the parlance of GAs. Since the representation of a solution in a GA is similar to a natural chromosome and GA operators are similar to genetic operators, the above procedure is named as genetic algorithm. We now discuss the details of the coding representation of a solution and GA operators in details in the following subsections.

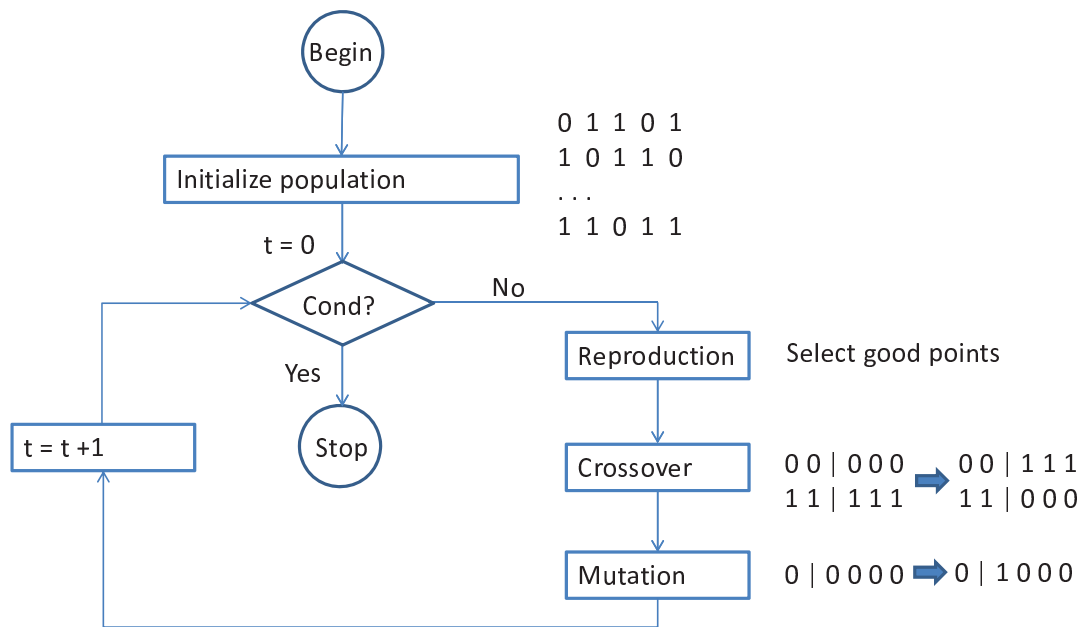


Figure 3.8: A flowchart of working principle of a genetic algorithm.

3.4.1 Individual Representation

The first and most important step in preparing an optimization problem for a GA solution is that of defining a particular coding of the design variables and their arrangement into a string of numerical values to be used as the chromosome by the GA. In most GAs, finite length binary coded strings of ones and zeros are used to describe the parameters for each solution. In a multi-parameter optimization problem, individual parameter coding are usually concatenated into a complete string which is shown in Figure 3.9.

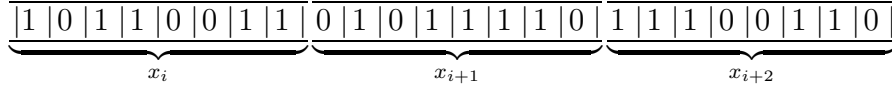


Figure 3.9: Binary representation in GA.

In this dissertation, real representation of binary string is used. The length of the string depends on the required precision. The mapping from a binary string to a real number is completed in two steps:

Step 1: Find code length for x_i ($i = 1, 2, \dots, n$):

$$c = (x_i^{max} - x_i^{min}) \times r \quad (3.8)$$

where r is the required precision ($10^1, 10^2, 10^3, \dots$).

Code length for x_i is as follows:

$$l_{x_i} = n + 1$$

where,

$$2^n \leq c \leq 2^{n+1}$$

Total string length is given by:

$$l = \sum_{i=1}^n l_{x_i}$$

Step 2: Mapping from a binary string to a real number:

$$x_i = x_i^{min} + \frac{x_i^{max} - x_i^{min}}{2^n - 1} \sum_{j=1}^n q_{ij} 2^{j-1} \quad (3.9)$$

where $q_{ij} \in [0, 1]$.

In order to generate the chromosomes, the length of the chromosome is calculated first. Then random numbers in the range of $[0, 1]$ are generated to form the chromosome. Random function is used as a random number generator.

3.4.2 Genetic Operators

Establishing the GA parameters is very crucial in an optimization problem because there are no guidelines. The genetic algorithms contains several operators, e.g. reproduction, crossover, mutation, etc.

3.4.2.1 Reproduction

The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. After assessing the fitness value for each string in the initial population, only a few strings with high fitness value are considered in the reproduction. There are many different types of reproduction operators which are

proportional selection, tournament selection, ranking selection, etc. In this study, tournament selection is selected, since it has better convergence and computational time compared to any other reproduction operator [Deb, 1999]. In tournament selection, two individuals are chosen from the population at random. Then the string which has best fitness value is selected. This procedure is continued until the size of the reproduction population is equal to the size of the population.

3.4.2.2 Crossover

Crossover is the next operation in the genetic algorithm. This operation partially exchanges information between any two selected individuals. Crossover selects genes from parent chromosomes and creates new off-springs. Like reproduction operator, there exist a number of crossover operators in GA. In a single-point crossover operator which is used in this paper, both strings are cut at an arbitrary place and the right-side portion of both strings are swapped among themselves to create two new strings, as illustrated in Figure 3.10.

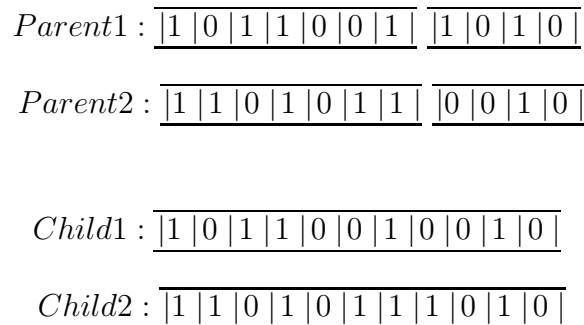


Figure 3.10: Illustration of crossover operator.

In order to carry out the crossover operation, two individuals are selected from

the population at random. Then a random number in the range of $[0, 1]$ is generated. If this random number is less than the probability of crossover then these individuals are subjected to crossover, otherwise they are copied to new population as they are. Also the crossover point is selected at random. Probability of crossover (P_c) is selected generally between 0.6 and 0.9.

3.4.2.3 Mutation

This is the process of randomly modifying the string with small probability. Mutation operator changes 1–0 and vice versa with a small probability of mutation (P_m). The need for mutation is to keep diversity in the population [Deb, 1999]. This is to prevent falling all solutions in population into a local optimum of solved problem. Figure 3.11 illustrates the mutation operation at seventh bit position.

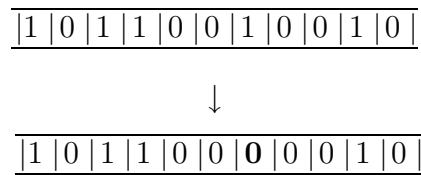


Figure 3.11: Illustration of mutation operator.

In order to determine whether a chromosome is to be subjected to mutation, a random number in the range of $[0, 1]$ is generated. If this random number is less than the probability of mutation, selected chromosome will be mutated. Probability of mutation should be selected very low as a high mutation will destroy fit chromosomes and degenerate the GA into a random walk. P_m should be selected between 0.02 and 0.06 [Deb, 1997].

3.4.2.4 Constraint Handling

In most application of GAs to constrained optimization problems, the penalty function method has been used. In this study a method proposed by Deb [Deb, 2000] is used. Although a penalty term is added to the objective function, this method differs from conventional GA implementations. The method proposes to use a tournament selection operator, where two solutions are compared at a time and the following criteria are always enforced:

- Any feasible solution is preferred to any infeasible solution.
- Among two feasible solutions, the one having better fitness value is preferred.
- Among two infeasible solutions, the one having smaller constraint violation is preferred.

3.4.2.5 Elitist Strategy

In this strategy, some of the best individuals are copied into the next generation without applying any genetic operators. Elitist strategy always clones the best individuals of the current generation into the next generation. This guarantees that the best found design is never lost in future generations.

3.4.3 Input Data Format

Data representing mounting features on fixture base are organized as plain file. As in Figure 3.12, where there are ten fixture bases with six mounting features on each fixture bases, ten fixture bases are sequentially organized. Each fixture base has six mounting features, which are presented by six pair of coordinates in $2D$ space.

- 10 (fixture base) X 6 (fixturing units)

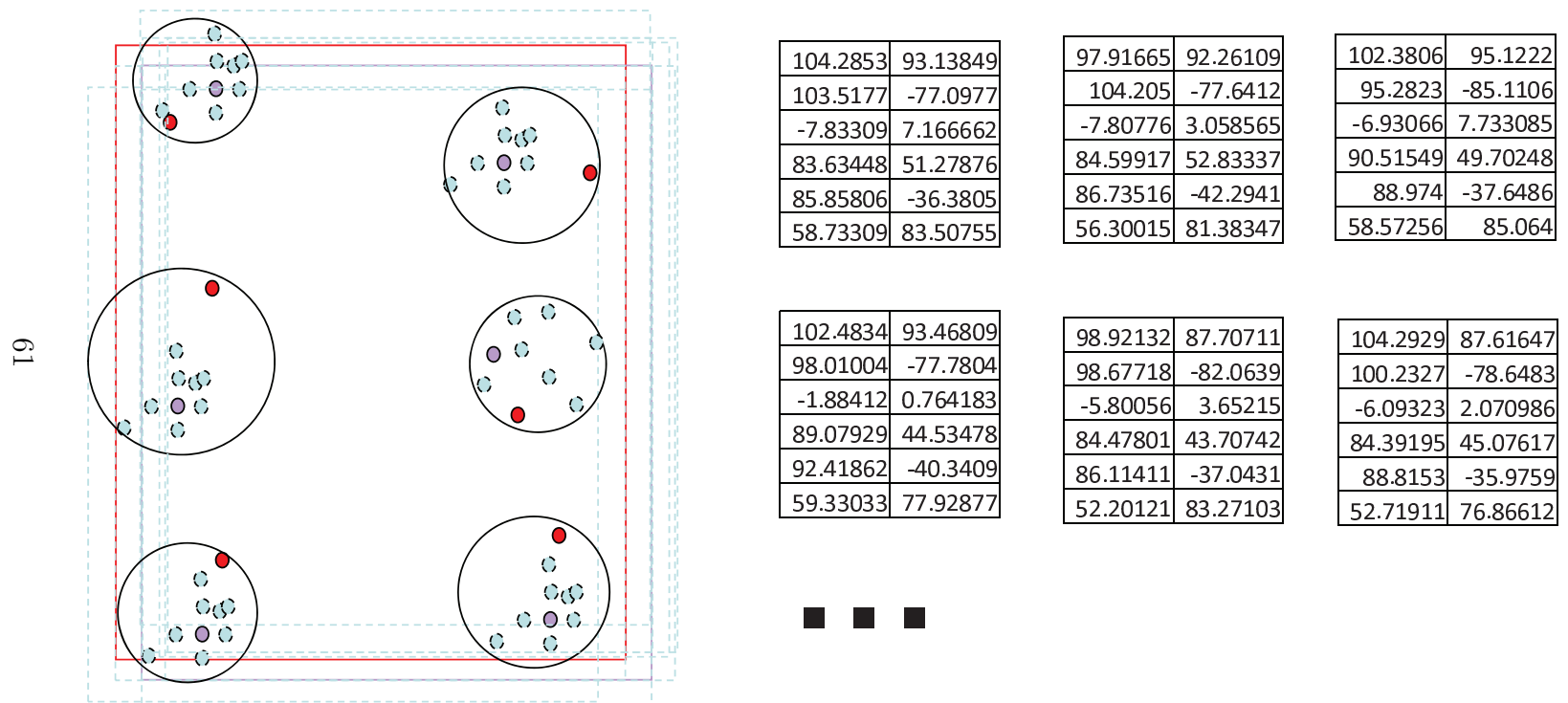


Figure 3.12: A dataset for fixture base synthesis.

Selection	Roulette Wheel Selection
Crossover probability	95%
Mutation probability	15%
Population size	100 – 1000
Elitism	20

Table 3.1: GA parameters used in case study.

3.4.4 A Case Study

With fixture base dataset displayed in Figure 3.12, fixture base optimization process is shown in Figure 3.13. In this case study we choose GA parameters as shown in Table 3.1. We choose engineering requirement function I (Equation 3.2) as the optimization objective function or fitness function in algorithm.

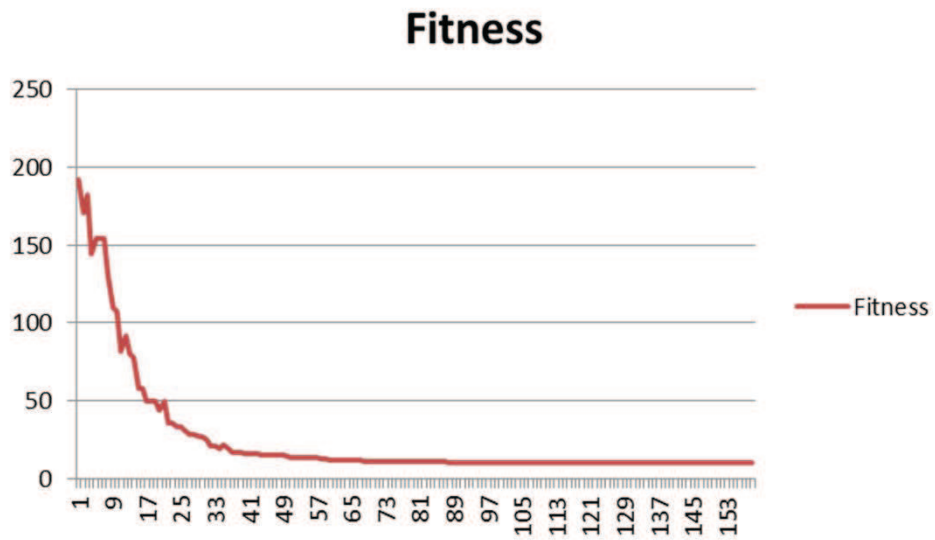
$$Min(Max(F_i)) \quad i = 1, 2, \dots, k$$

GA application for fixture base synthesis shows a good result. As shown in Figure 3.13, it shows five intermediate generations during GA running processes. For each generation, it shows six diameters of corresponding minimized enclosing circle for each mounting feature (red one is the largest among six diameters, which is used as the fitness for guiding algorithm), and ten translation and rotation matrix for each fixture base. Translation matrix is reflected by the first two coordinates in 2D plane, followed by a rotation angle that represents rotation matrix. Fitness values changed from initial setting of 195.5051 to the optimum of 10.2067, which is what expected.

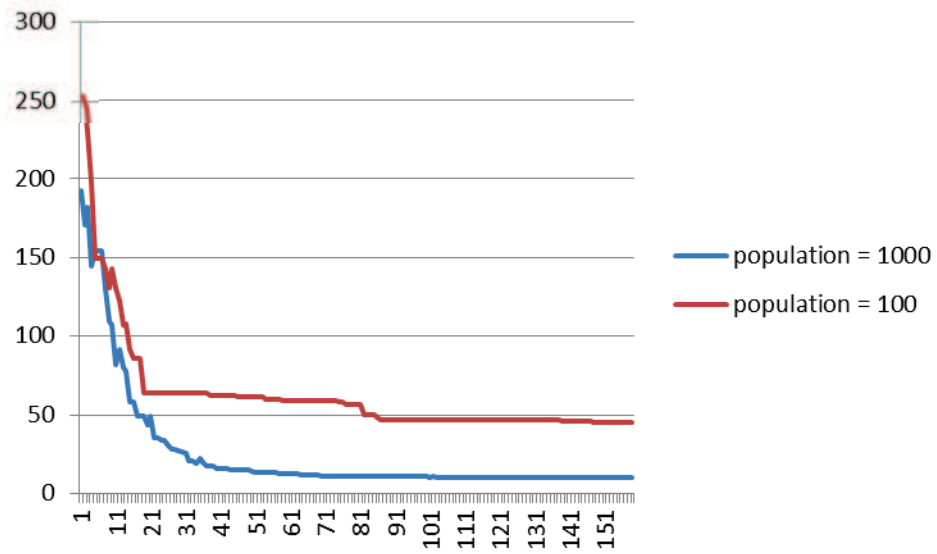
Figure 3.14a shows fitness changing vs generations of algorithm (population size is 1000). It converges quickly in the first fifty generations and then gradually converges with following generations. If fitness is not required to be as precise as 0.01, the generation number of fifty can be treated as the terminating point of the algorithm.

Figure 3.14b shows fitness changing of two running cases where population size are 1000 and 100 respectively. The best fitness can not be achieved when population size is 100. That's because in this case there are thirty optimization variables (10 (fixture bases) \times 3 (transform variables)), an adequate population size is required to produce a appropriate result. According to the established fact, at least population size with equal or greater than 1000 is required to run the algorithm.

However, the running time increase exponentially with the population size. With the computation capability of current popular personal computer (Intel Duo-core CPU with 2.16G Hz and 6 GB PC3 8500 memory), it spends 368 seconds to run first fifty generations when population size is 1000. It proved to be costive with increasing number of population size. This encourage us to develop a more efficient algorithm to synthesize fixture bases, which will be discussed in details in the next chapter.



(a) GA fitness vs generations.



(b) Two GA fitness.

Figure 3.14: GA fitness vs generations.

Chapter 4

An Improved Algorithm for Fixture Base Synthesis

Aimed at the problems existing in the current practice, this chapter presents an approach that synthesizes an optimized fixture base based on the existing workpiece families. The presented methodology is based on (1) analytical Procrustes analysis, to rapidly eliminate unlikely sets of solutions, and (2) advanced optimization of fixture base for a given part family. The optimized objective functions are formulated according to the current practices. The detailed optimization processes are further discussed in this chapter.

To overcome the aforementioned challenges, a fixture base synthesis method for a family of parts is developed by applying Procrustes-based optimization of fixture configurations. Essentially, this method follows two steps shown in Figure 4.1. The first step applies Procrustes analysis to obtain a preliminary fixture base layout that reduces the search domain for the variables. The second step is to use the configuration optimization to obtain the optimized results catering to various objective functions as shown in Equation 3.2 - 3.4.

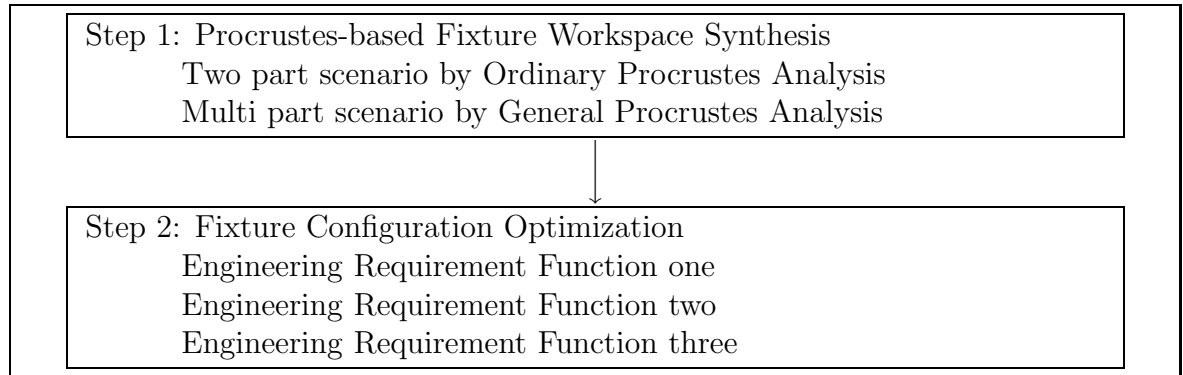


Figure 4.1: Procedure of fixture base synthesis method.

4.1 Preliminary Fixture Base Synthesis Using Procrustes Analysis

The analysis uses isotropic scaling, rigid translation, and rotation transformations to best match one configuration with another wherein a configuration consists of set of multidimensional points. It provides an effective modeling tool for fixture workspace synthesis for a family of parts. Procrustes analysis is an analytical method, therefore, it can produce the output instantly. Because it is based on the least summation of squares, which is not exactly the same as the engineering requirement functions represented by Equation 3.2 - 3.4, the Procrustes analysis is further called preliminary fixture workspace synthesis (Step 1 in Figure 4.1). The term “preliminary” means that it can shorten the overall optimization process, but an optimization method (Step 2 in Figure 4.1) is still needed to obtain the final required fixture configurations. To utilize procrustes analysis, a circle is identified with a minimum diameter to enclose the locating area. Consequently, a point can be employed that is the center of the circle to represent the locating area, along with the diameter of the

circle. The following subsections present the detailed approach.

4.1.1 Fixture Base Synthesis for Two-Part Family

Ordinary Procrustes Analysis (OPA) can be used for fixture base synthesis involving two parts, where a single configuration is fitted to another one by identifying superposition between two configurations [Crosilla, 1999]. The Procrustes analysis for fixture synthesis for a family of two parts is presented as bellows.

(1) Translation Transformations for Coordinate Registration

When two fixture configurations are compared, the first step is to choose a proper coordinate system in which the two configurations are placed. The selected coordinate system should be able to eliminate or significantly reduce the necessary translation transformation and will be further called “coordinate registration”.

Let X be a $k \times m$ matrix representing a configuration with k Tooling Elements (fixture units) in m dimensions each, and $X_{i,j}$ is the (i, j) -th entry of X . The centered coordinate system is used for the coordinate registration, which can be obtained by the following steps. First, to compute the arithmetic average of the coordinate of each configuration:

$$\bar{X}_j = \frac{1}{k} \sum_{i=1}^k X_{i,j}, \quad j = 1, 2, \dots, m \quad (4.1)$$

Then, each configuration is translated to make the point with coordinate $(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_m)$ coincide with the coordinate origin. Figure 4.2 shows an example of 2 – D configurations before and after the coordinate registration.

(2) Rotation Transformation

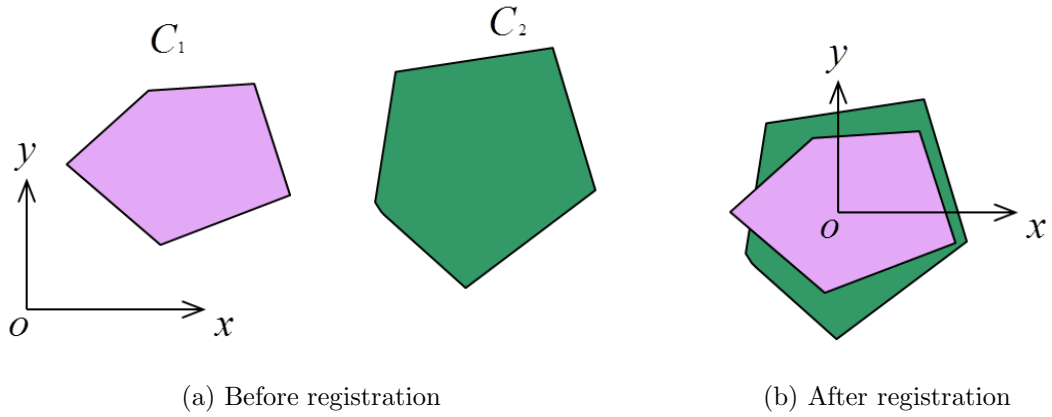


Figure 4.2: Coordinate registration of configurations.

The distances between the corresponding fixture units of the two fixture configurations are minimized by applying similarity transformations, which include translation and rotation of the fixture configurations (the scaling transformation is not used). Thus, after the coordinate registration, some necessary transformations for configurations need to be determined by which the two configurations can reach their best superposition. Assuming C_1 and C_2 are two fixture configurations after the coordinate registration, represented by $k \times m$ matrices, then the following holds:

$$C_1^T L_k = C_2^T L_k = 0$$

where L_k is a $k \times m$ matrix with all elements equal to one. Let R and T be the rotation and translation transformation matrices, respectively. Then, if fitting C_1 to C_2 using R and T transformations, the following is obtained:

$$C_2 = C_1 R + L_k T^T + \epsilon \tag{4.2}$$

where ϵ is the residual error, which indicates the difference between two fixture configurations. By applying the least-squares method, ϵ can be minimized with the objective function as follows (see Figure 3.4 for notation of F_i):

$$Min\left(\sum_{i=1}^k F_i^2\right) \quad (4.3)$$

Based on Equation 4.2, the sum of squares of ϵ , that is, the Euclidean distance between C_1 and C_2 ,

$$\begin{aligned} D^2(C_1, C_2) &= \|\epsilon^T \epsilon\| \\ &= trace(C_2^T C_2 + C_1^T C_1 - C_2^T C_1 R - R^T C_1^T C_2 + kTT^T) \end{aligned} \quad (4.4)$$

Because only the last item (kTT^T) involves T in Equation 4.4, it can be found that:

$$T = 0 \quad (4.5)$$

is the condition for minimizing the sum of squares of the residual error. Therefore, Equation 4.4 becomes:

$$D^2(C_1, C_2) = trace(C_2^T C_2 + C_1^T C_1 - C_2^T C_1 R - R^T C_1^T C_2) \quad (4.6)$$

Consequently, the minimized sum of squares of the residual error is:

$$D_{min}^2(C_1, C_2) = trace(C_2^T C_2 + C_1^T C_1) - 2suptrace(C_2^T C_1 R) \quad (4.7)$$

By using the singular value decomposition, the item $C_2^T C_1$ can be expressed as:

$$C_2^T C_1 = V \Lambda U^T$$

where V and U belong to a special orthogonal group and Λ is the vector of eigenvalues of $C_2^T C_1$. Based on the properties of a special orthogonal group, if

$$R = UV^T \tag{4.8}$$

then the minimized sum of squares of residual error can be achieved and the result is as follows:

$$D_{min}^2(C_1, C_2) = trace(C_2^T C_2 + C_1^T C_1) - 2trace(\Lambda) \tag{4.9}$$

Finally, the solution of the transformations is obtained that can lead to the best superposition based on the least square of the residual error between two configurations, and the translation and rotation transformations are represented by Equation 4.1 and 4.8, respectively.

4.1.2 Fixture Base Synthesis for Multi-Part Family

General Procrustes Analysis (GPA) is applied for fixture workspace synthesis involving more than two parts. Assume there are $N \geq 2$ fixture configurations represented by X_1, X_2, \dots, X_N , where each configuration is represented by a $k \times m$ matrix, k is the number of fixture units, and m represents dimension. The GPA also uses similarity transformations to translate and rotate the configurations to minimize the total

sum of squares of the distance of each pair of configurations $D(X_1, X_2, \dots, X_N)$, which can be expressed as follows:

$$D(X_1, X_2, \dots, X_N) = \frac{1}{N} \sum_{i=1}^N \sum_{j=i+1}^N \|(X_i R_i + L_k T_i) - (X_j R_j + L_k T_j)\|^2 \quad (4.10)$$

where R_i and T_i represent rotation and translation transformations and i represents the i -th fixture configurations. Equation 4.10 indicates that the approach is composed of all of the combinations of OPA for each pair of configurations. Based on this relationship, an efficient algorithm can be developed (Figure 4.3), the procedure for which is as follows:

- (1) Translation transformations for coordinate registration

This is done using translation to center all of the configurations. After this step, for all X_i , $i = 1, 2, \dots, N$, the following holds,

$$X_i^T L_k = 0$$

where L_k is a $k \times m$ matrix with all elements equal to one.

- (2) Rotation transformation

For each configuration X_i , $i = 1, 2, \dots, N$, the average is computed for other $N - 1$ configurations, which is represented by:

$$\bar{X}_{(i)} = \frac{1}{N-1} \sum_{j=1, j \neq i}^N X_j$$

Consequently, for the two resulting configurations, X_i and $\bar{X}_{(i)}$, the OPA can be

```

function Syn_MultiPart_Family( $X_1, X_2, \dots, X_N$ )
returns fixture base configurations with minimum sum square
input:  $X_1, X_2, \dots, X_N$ , a set of fixture configurations
    Translation_registration( $X_1, X_2, \dots, X_N$ )
repeat
    loop for  $i$  from 1 to  $N$  do
         $\bar{X}_i \leftarrow \frac{1}{N-1} \sum_{j=1, j \neq i}^N X_j$ 
         $(T_i, R_i) \leftarrow \text{OPA}(X_i, \bar{X}_{(i)})$ 
         $D(X_1, X_2, \dots, X_N) \leftarrow \frac{1}{N} \sum_{i=1}^N \sum_{j=i+1}^N \|(X_i R_i + L_k T_i) - (X_j R_j + L_k T_j)\|^2$ 
    end loop
until  $D(X_1, X_2, \dots, X_N)$  can not be further reduced.
return fixture base configuration  $(X_1, X_2, \dots, X_N)$  with minimum sum square

```

```

function Translation_registration( $X_1, X_2, \dots, X_N$ )
returns  $X_1, X_2, \dots, X_N$ , a set of translated fixture configurations
inputs:  $X_1, X_2, \dots, X_N$ , a set of fixture configurations
    loop for  $n$  from 1 to  $N$  do
        loop for  $j$  from 1 to  $m$  do
             $\bar{X}_j \leftarrow \frac{1}{k} \sum_{i=1}^k X_{i,j}$ 
        end loop
         $T_n \leftarrow (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_m)$ 
         $X_n \leftarrow T_n X_n$ 
    end loop
return  $X_1, X_2, \dots, X_N$ , a set of translated fixture configurations

```

Figure 4.3: GPA for multi-part family.

applied to obtain the necessary translation T_i and rotation R_i for X_i so as to achieve the best superposition with $\overline{X}_{(i)}$. Calculate $D(X_1, X_2, \dots, X_N)$ using Equation 4.10.

(3) Repeat step (2)

From Equation 4.5, it can be seen that there are no more translations needed after the coordinate registration. Therefore, step (2) can be repeated for all $i = 1, 2, \dots, N$, and the resulting rotation transformations will make the N configurations iteratively closer and closer, so the value of $D(X_1, X_2, \dots, X_N)$ will be decreased. The procedure will stop when $D(X_1, X_2, \dots, X_N)$ cannot be reduced any further.

By using the GPA algorithm, the necessary translation (from Step (1)) and rotation (from Step (2)) for each configuration can be obtained, which leads to the best match for all configurations in terms of the least sum of squares of the coordinate differences between every pair of the configurations.

4.2 Advanced Fixture Base Synthesis

In previous section presents a preliminary fixture workspace synthesis using analytical Procrustes analysis based on the objective function expressed as Equation 4.3, to reduce search domain of the variables that need to be optimized. In this section, an advanced configuration optimization method is developed to obtain the final fixture workspace layout required by the objective functions expressed by Equation 3.2-3.4. The flow of configuration transitions and the corresponding analysis is described in Figure 4.1.

4.2.1 Advanced Synthesis Algorithm

The challenge in conducting such advanced optimization for the final solution of fixture workspace synthesis is how to conduct it for a number of parts equal to any $N \geq 2$. If one part is simply fixed and the other $N - 1$ parts are rotated and translated, then the simultaneous optimizing of $3 * (N - 1)$ variables is required for $2 - D$ cases. The complexity involved is excruciatingly high. To solve this problem, an advanced configurations optimization method is used, which simultaneously deals with only three variables at any time. The basic procedure is illustrated in Figure 4.9. For N parts, assume that $Y_i, i = 1, 2, \dots, N$, represents their fixture layout configurations obtained from the conducted Procrustes-based fixture workspace synthesis analysis. For each configuration, Y_{ct} ($ct = 1, 2, \dots, N$), the average configuration, $\bar{Y}_{(ct)}$ is computed based on all configurations except Y_{ct} .

$$\bar{Y}_{(ct)} = \frac{1}{N - 1} \sum_{j=1, j \neq ct}^N Y_j$$

Then search optimization for the pair of configurations Y_{ct} and $\bar{Y}_{(ct)}$ is conducted based on one of the objective functions represented by Equation 3.2-3.4. During the optimization, the configuration of $\bar{Y}_{(ct)}$ is fixed, and rotation and translation are made to Y_{ct} . For example, for $2 - D$ cases in the XY plane, the search variables are T_x, T_y (translations in X and Y directions) and R (rotation in XY plane). The final result of the search optimization identifies rotation and translation transformations needed to transform Y_{ct} , to satisfy one of the objective functions. This search optimization process is repeated for every Y_{ct} ($ct = 1, 2, \dots, N$) until the obtained value of the objective function cannot be reduced any further. Then, the best match

```

function Advanced_Optimization( $Y_1, Y_2, \dots, Y_N$ )
returns  $Y_1, Y_2, \dots, Y_N$ : a set of optimized fixture configurations
input:  $Y_1, Y_2, \dots, Y_N$ : a set of fixture configurations

begin
     $ct \leftarrow 1$ 
    repeat
         $\bar{Y}_{(ct)} \leftarrow \frac{1}{N-1} \sum_{j=1, j \neq ct}^N Y_j$ 
        Search_Optimization( $Y_{(ct)}, \bar{Y}_{(ct)}$ )
         $ct \leftarrow ct + 1$ 
    until the value of objective function can't be reduced further
end
return  $Y_1, Y_2, \dots, Y_N$ , a set of optimized fixture configurations

```

Figure 4.4: Procedure of the advanced configuration optimization.

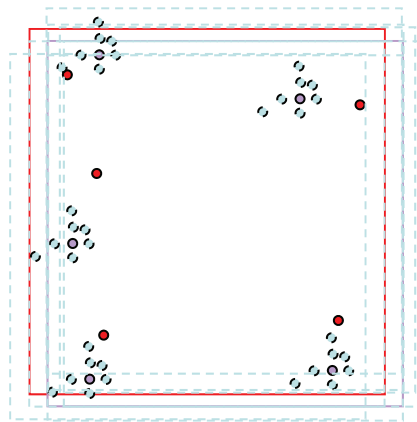
for the N configurations is determined, which meets the corresponding objective function.

The algorithm conducts all computations by using simultaneously only three variables. Compared to genetic algorithm that uses simultaneously $3 * (N - 1)$ variables, where N is the number of parts. The results in the complexity of $O(K^{3*(N-1)})$, if the number of search steps for each variable is K . The method proposed in this dissertation deals simultaneously with only three variables, which results in the complexity of $O(K^3)$. Therefore, the complexity of optimization is drastically reduced by using the proposed algorithm.

The configuration optimization algorithm can be conducted with various objective functions as required by specific industrial needs. As discussed in section 3.2, the dissertation explores three objective functions, described in sections 4.3.1, 4.3.2, and 4.3.3, respectively.

4.2.2 Search Algorithm for $Y_{(ct)}$ and $\bar{Y}_{(ct)}$

For a given for $Y_{(ct)}$ and $\bar{Y}_{(ct)}$, genetic algorithm serves as a perfect algorithm in that there are only three variables in searching space. Figure 4.6 shows genetic algorithms that applied to $Y_{(ct)}$ and $\bar{Y}_{(ct)}$. Figure 4.5 demonstrate population initialization and cross over of genetic algorithm.



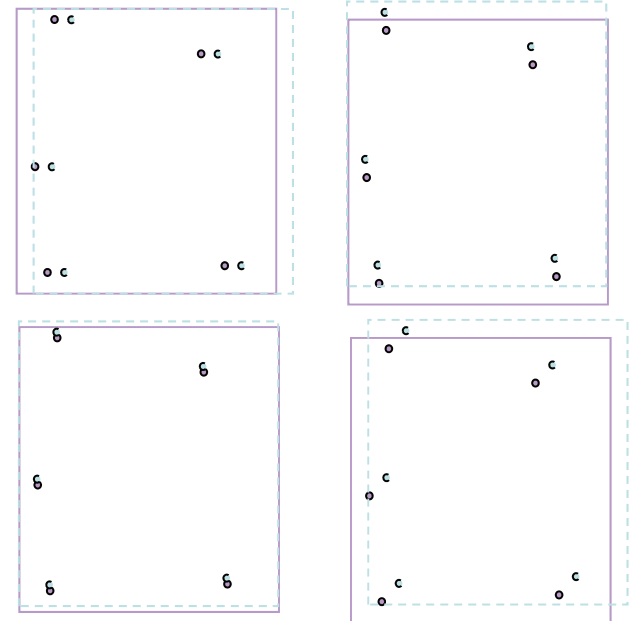
Search population

Target configuration

Initial configuration

Search population

(a) Search population



(b) Cross over

Figure 4.5: GA for $Y_{(ct)}$ and $\bar{Y}_{(ct)}$.

```

function Search_Optimization( $Y_1, Y_2$ )
returns:  $Y_1, Y_2$ , two optimization fixture configurations
input:  $Y_1, Y_2$ , two fixture configurations

begin
    initialize search population  $Y_{11}, Y_{12}, \dots, Y_{1n}$ 
    repeat
        reproduction
        op_crossover
        op_mutation
        evaluate fitness functions (objective functions)
    until no further optimization required
end
return  $Y_1, Y_2$ , two optimization fixture configurations

```

Figure 4.6: Search algorithm for $Y_{(ct)}$ and $\bar{Y}_{(ct)}$.

4.3 Engineering Objected Optimization

As we proposed engineering requiring objective functions in Equation 3.2 - 3.4, subsections discuss optimization processes in terms of corresponding objective functions.

4.3.1 Fixture Base Synthesis: Optimization Objective I

The first objective function for the fixture workspace synthesis is to minimize the DW of the fixturing unit with the largest DW within a given fixture. Let $(F_i)_{min(max)}$ ($i = 1, 2, \dots, k$) represent the diameters of the circles obtained based on objective function I (Equation 3.2). Assume that the obtained solution by applying Procrustes analysis for all F_i (Figure 3.4) based on Equation 4.3 is represented as:

$$(F_i)_{LS} \quad (i = 1, 2, \dots, k) \quad (4.11)$$

The diameter of the maximum circle obtained based on Equation 4.3 is represented as $(F_{max})_{LS}$. From the definition of the least sum of squares, the following then must hold:

$$\left(\sum_{i=1}^k (F_i)_{LS}^2 \right) \leq \left(\sum_{i=1}^k (F_i)_{min(max)}^2 \right) \quad (4.12)$$

Assuming $(F_{max})_{min(max)}$ is the maximum amid $(F_i)_{min(max)}$, $i = 1, 2, \dots, k$, the following can be obtained:

$$\left(\sum_{i=1}^k (F_i)_{LS}^2 \right) \leq k \left((F_{max})_{min(max)}^2 \right) \quad (4.13)$$

It is known that Equation 3.2 is the objective function for the fixture workspace synthesis that minimizes the largest circle among the ones that enclose the corresponding locating point cluster of different configurations. Therefore, the following inequality holds:

$$(F_{max})_{min(max)} \leq (F_{max})_{LS} \quad (4.14)$$

Based on Equation 4.13 and 4.14, the following can be obtained:

$$\sqrt{\frac{\left(\sum_{i=1}^k (F_i)_{LS}^2\right)}{k}} \leq (F_{max})_{min(max)} \leq (F_{max})_{LS} \quad (4.15)$$

Equation 4.15 actually provides the boundary condition for further pairwise configuration optimization, and it can be utilized to determine the range of variables that need to be optimized. Afterward, a simple exhaustive search optimization method can be applied to rapidly find the desired solution.

4.3.2 Fixture Base Synthesis: Optimization Objective II

Here is minimized the total DW for all mounting features within a given fixture. The objective function II is formulated as Equation 3.3.

The notation in Equation 3.3 is the same as in Equation 3.2, and the expected configuration can still be described by Figure 3.4. Assume $(F_i)_{sum}$, $i = 1, 2, \dots, k$, are the diameters of the circles obtained using Equation 3.3. Then,

$$\sum_{i=1}^k (F_i)_{sum} \leq \sum_{i=1}^k (F_i)_{LS} \quad (4.16)$$

where $(F_i)_{LS}$ is obtained using the Procrustes analysis. If the diameter of minimum circle among $(F_i)_{sum}$, $i = 1, 2, \dots, k$, is notated as $(F_{min})_{sum}$, the following inequality must hold:

$$(F_{min})_{sum} \leq \frac{\sum_{i=1}^k (F_i)_{LS}}{k} \quad (4.17)$$

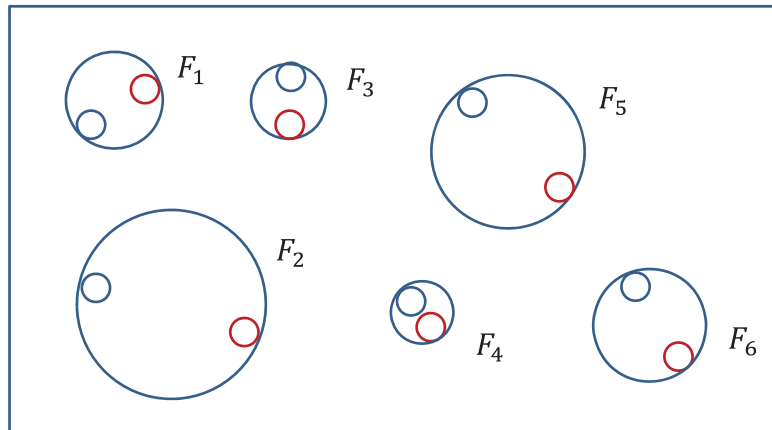
Equation 4.17 can be utilized to narrow the range of the variables that need to be optimized during the further optimization based on objective function II.

4.3.3 Fixture Base Synthesis: Optimization Objective III

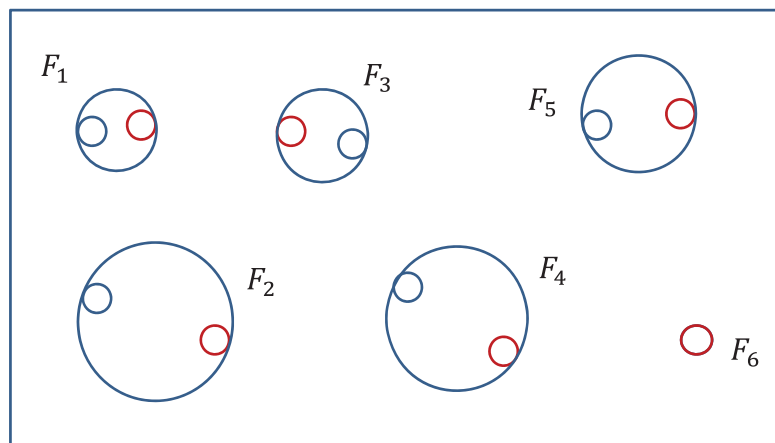
This objective function is to (i) minimize the DW of one selected fixturing unit and (ii) simultaneously to minimize the total DW for all fixturing units. This scenario is much more specific than others because it has requirements for both a workspace of a selected fixturing unit and an overall fixture workspace. The objective function for this case is expressed as Equation 3.4. This is a two-step optimization. The first step is to minimize the workspace of a specified fixturing unit. Then, the second step will minimize the overall fixture workspace, meanwhile maintaining the specified fixture unit workspace unchanged.

In section 3.1.2, a point called is used to represent a fixturing unit, and this point is the center of the minimum circle that encloses the corresponding fixturing unit. The procedure for satisfying objective function III is explained by using a simple example with the two fixture configurations shown in Figure 4.7, Assume that fixture unit 6 was selected to be minimized. Thus, after the Procrustes analysis (Figure 4.7a), both fixture unit 6's for fixture base A and B are coincided by using translation (Figure 4.7b). This can ensure that the size of workspace for fixture unit 6 is minimized, and it is actually equal to the diameter of the larger circle among the two work spaces of fixture unit 6 for fixture base A and B that is shown in Figure

4.7b. See Figure 4.8 for details.



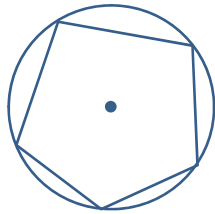
(a) Configurations after procrustes analysis



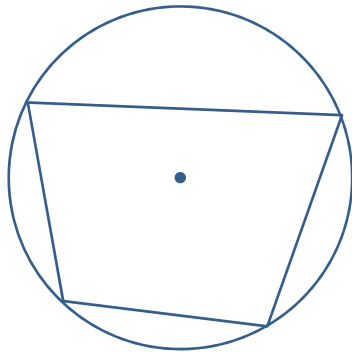
(b) Minimizing F_6 by coinciding fixturing unit 6 for A and B

Figure 4.7: Processing for engineering requirement function III.

In Figure 4.8, which is, in fact, an enlarged view of the workspace of fixture unit 6 in Figure 4.7a, it can be seen that the two configurations A and B (Figure 4.8a) become one point after they are coincided (Figure 4.8b), and consequently, the aforementioned larger workspace of the two fixture units for fixture configurations A and B determines the required minimum workspace of fixture unit 6.

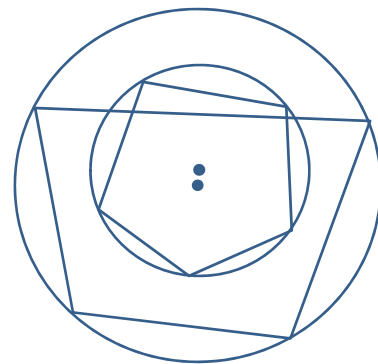


FU 6 of fixture A

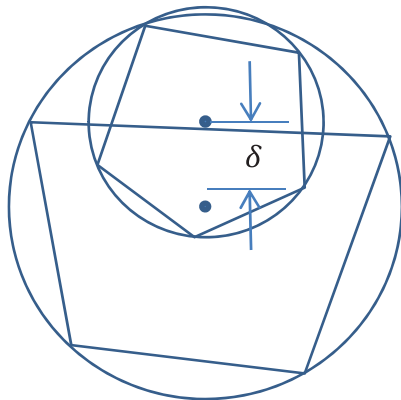


FU 6 of fixture B

(a) Two configurations A and B of fixturing unit 6



(b) Translation to coincide fixturing unit 6 for both configuration A and B



(c) Adjustment

Figure 4.8: Enlarged view of workspace fixture unit 6 in Figure 4.7b.

Thus, after the specified workspace of fixture unit 6 is minimized, the overall workspace also needs to be minimized. This can be done by rotating one of the configurations around the center of the coincided fixture unit 6 to identify the overall minimum fixture workspace. In this way, the selected fixture unit workspace (fixture unit 6 in Figure 4.7) still remains minimized. Moreover, it can be seen that to remain the minimized size of the specified fixture unit workspace it is not necessary to keep the two fixture unit 6's for configuration A and B completely coincided. For example, as long as the distance "δ" (Figure 4.8c) between the two fixture unit 6 for configurations A and B is small enough so that Circle C_1 is always within Circle C_2 , the resulting workspace, F'_6 , is still minimized (F'_6 is determined by the size of Circle C_2). Therefore, this provides an extra capability (translations) to minimize the overall fixture workspace in addition to the aforementioned rotating operation.

In general, for k locator point clusters (in Figure 4.7, $k = 6$) circles are represented as F'_i , $i = 1, 2, \dots, k$, based on the process described in Figure 4.7b. The selected fixture unit workspace to be minimized is noted as F'_p and marked as D_{min} . It is assumed that F''_i , $i = 1, 2, \dots, k$, represents the final solution, which meets both requirements of minimizing the DW of one selected fixturing unit and simultaneously minimizing the total DW for all tooling elements. During optimization of the overall fixture workspace, the selected workspace of fixture unit 6 (Circle C_2) that has been minimized will not change, that is,

$$F''_p = F'_p = D_{min} = D_{circle2} \quad (4.18)$$

Based on the optimization requirement, it is known that:

$$\sum_{i=1}^k F_i'' \leq \sum_{i=1}^k F_i' \quad (4.19)$$

It is assumed that the smallest circle among F_i'' , excluding F_p'' is F_{2nd_min}'' . Then, based on Equation 4.19, the following relationship holds:

$$F_{2nd_min}'' \leq \frac{\left(\sum_{i=1}^k F_i' - D_{min} \right)}{k} \quad (4.20)$$

Equation 4.20 can be used to reduce the ranges of the variables that need to be optimized during the further optimization based on objective function III. The presented procedure can be summarized in the following way by using the example in Figure 8. After applying Procrustes analysis, the coinciding operation applied to the selected fixture unit of configurations A and B illustrated in Figure 4.8b is conducted. Then, the minimized workspace will remain unchanged as long as Circle C_1 is within Circle C_2 , as shown in Figure 4.8c. Under this constraint and based on Equation 3.4, the pairwise optimization approach is applied and the final solution can be obtained, which satisfies both the requirements of (i) minimizing the DW of one selected tooling element and (ii) simultaneously minimizing the total DW for all tooling elements.

4.4 Algorithm Efficiency

With the same dataset as displayed in Figure 3.12 and the same objective optimization function, It proved that the fitness converges rapidly compared to the GA algorithm discussed in previous chapter. As shown in Figure 4.9, fitness converge to

10.28 (the minimum acquired from GA algorithm) after about 26 iterations. Notice that the initial fitness value is about 11.8, which is evaluated after preliminary analysis.

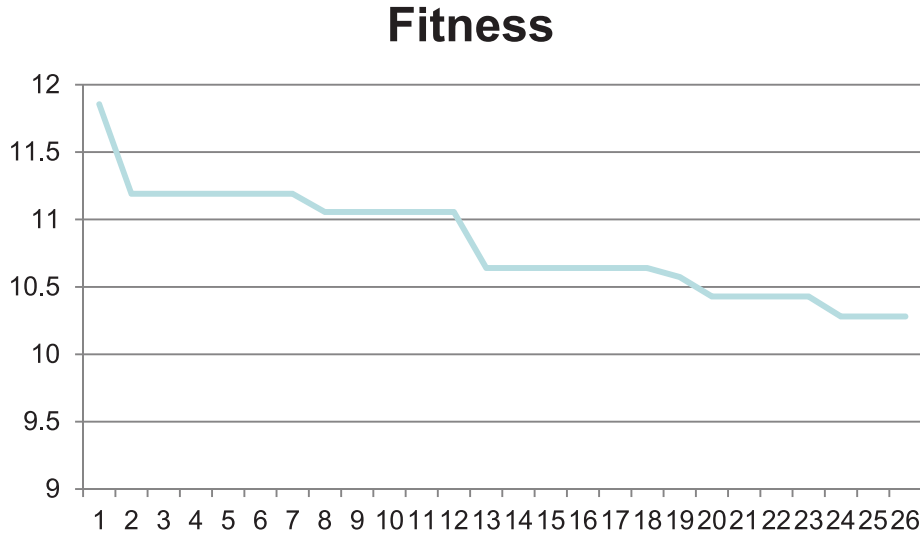


Figure 4.9: Fitness changing of improved fixture base synthesis algorithm.

In terms of computation efficiency, the improved algorithm has much better performance than GA. Figure 4.10 displayed times used (in milliseconds) when fitness reaches to 10.28. GA uses about 368×10^3 milliseconds while improved algorithm uses about 19×10^3 milliseconds. The improved algorithm demonstrated an exponentially decreased time over the genetic algorithm.

Algorithm complexity analysis supports our witness of algorithm performances. Genetic algorithm require adequate populations in nature to get appropriate results. Often population size is increased exponentially with encoding string length, which is linear to the number of variables and encoding resolution. In other words,

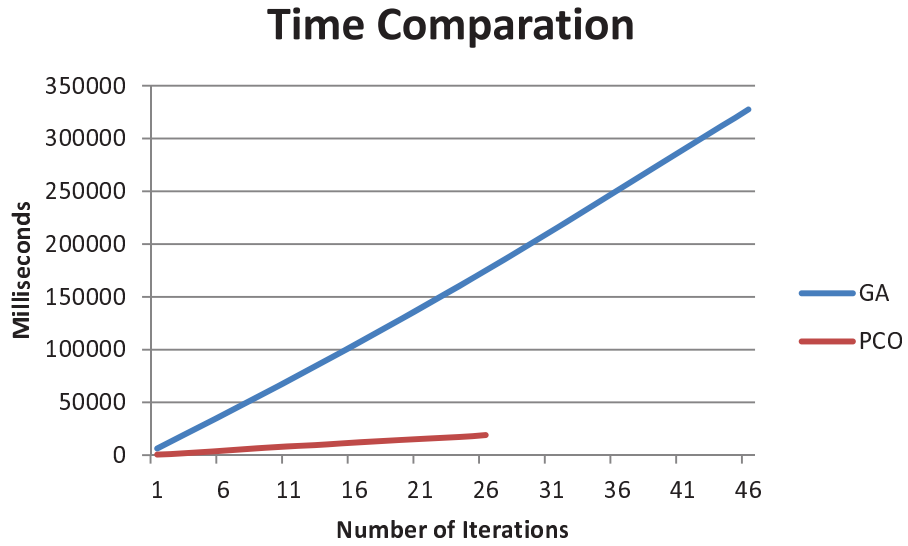


Figure 4.10: Times used of two algorithms.

	GA	Improved
Variables	$3 \times (N - 1)$	3
Complexity	$O(K^{3 \times (N-1)})$	$O(K^3)$

Table 4.1: Algorithm complexity

computation time can be exponentially increased with the number of optimization variables. Table 4.1 lists computation complexity of two algorithm where K can be treated as a constant that depends on fitness convergence requirements.

Chapter 5

Structural Analysis and Fixture Modeling

As for a family of work parts, their typical fixture configuration and fixture assembly had been well defined and implemented. Together with machining processes and tools form the current best practices. The fixture configuration and fixture assembly of a given family member are expected to be derived from the predefined fixture configuration and fixture assembly.

In this chapter we discuss the fixture models that serve as essential rational for the variation fixture design. Those include workpiece model, workpiece variation model, fixture model, and fixture variation model. A complete variation process based on those models is presented and discussed in detail in this chapter.

5.1 Workpiece Modeling

For a better derivation of the fixture design from a given fixture prototype, all solids concerned with the design process must be modeled, including the workpiece, workpiece variation, fixture, fixture variation and component. In this chapter, these

models will be discussed separately.

5.1.1 Workpiece Model

Generally, the workpiece is a part or assembly whose fixturing equipment is needed for machining to proceed. It can be evaluated by any kind of available solid modeling packages such as Unigraphics.

To fit the fixture design requirements, two kinds of features are emphasized. One of these features that will be fabricated is machining features such as surfaces, slots, holes, etc. Another kind of features is those which the fixture activity will be acted, which usually are planar or cylindrical surfaces. They are called fixturing features.

Each fixturing activity, either locating or clamping, is conducted on some surfaces of a workpiece. For modeling simplicities it can be abstractly treated as acting on a point of surface. An acting point and an acting surface for each fixturing activity are used to represent a fixturing feature.

In the point domain, a fixture is described as follows. Let us denote the n permissible surface locations by the equation $r_k = \{x, y, z\}_k^T$ on the workpiece by Γ , which are assumed to be practically feasible for fixturing. A fixture is represented as a collection of two distinct sets

$$\Gamma = \{\mathcal{L}, \mathcal{C}\} \quad (5.1)$$

where

$$\mathcal{L} = \{\vec{\mathbf{r}}_i, \vec{\mathbf{n}}_i\} \quad i = 1, 2, \dots, n; \quad n = 6 \quad (5.2)$$

$$\mathcal{C} = \{\vec{\mathbf{r}}_j, \vec{\mathbf{n}}_j\} \quad j = 1, 2, \dots, c; \quad c \geq 1 \quad (5.3)$$

$$\forall i, j, \vec{\mathbf{r}}_{i,j} \in \Gamma, \vec{\mathbf{n}}_{i,j} \in \Gamma$$

$\{\mathcal{L}\}$ is a locator set and $\{\mathcal{C}\}$ a clamp set. Locators are essential for providing unique and accurate location (both position and orientation) of the workpiece with respect to a fixture reference frame. They also provide support for force-closure after clamping. For unique locating, exactly 6 locators must be used in 3 – D (or three locators in 2 – D). It should be noted that locators are passive elements. A clamp is represented as a force applied on the workpiece to provide a complete restraint of it against any external forces. At least one clamp is required.

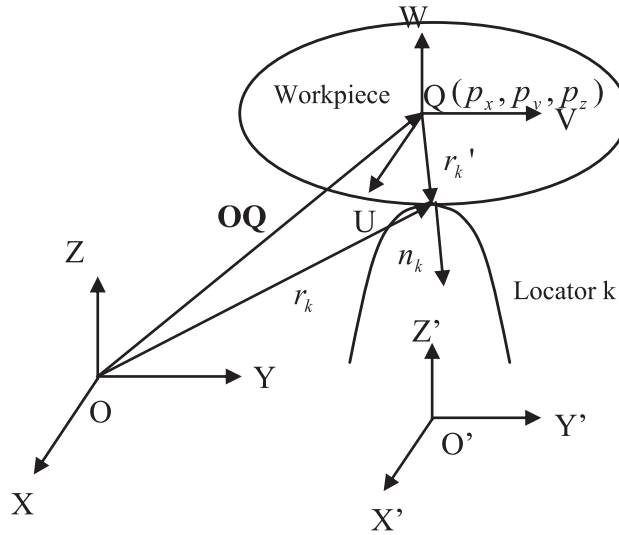


Figure 5.1: Fixture coordinate systems.

Figure 5.1 shows a workpiece, a locator, and three coordinate systems - the global coordinate system $OXYZ$, locator local coordinate system $O'X'Y'Z'$, and

the workpiece local coordinate system $QUVW$. A transformation matrix could be used to describe the relationship between $\vec{\mathbf{r}}_k$ and $\vec{\mathbf{r}}'_k$.

$$\vec{\mathbf{r}}_k = T\vec{\mathbf{r}}'_k \quad (5.4)$$

where T is the transformation matrix between $QUVW$ and $OXYZ$.

Transformation matrix T determines the position and orientation of two coordinate systems uniquely. It could be acquired through a displacement and a series of rotations of the coordinate system.

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

where, p_x , p_y and p_z are the coordinates of the origin of $QUVW$ in the $OXYZ$, n_x , n_y , n_z , o_x , o_y , o_z , a_x , a_y , a_z are the cosine of the unit vector along the U , V and W axes of $QUVW$ in x , y and z direction of $OXYZ$.

In the fixture design environment, a workpiece model can be expressed as:

$$WP = \{CAD\ 3 - D\ data, MF_SET, FIX_SET\} \quad (5.6)$$

where the MF_SET is a set of manufacturing features and the FIX_SET is a set of fixturing features in the workpiece.

For machining features, the geometrical information of itself and cutting tools that were used to machine it could be used to represent them in fixture design. Geo-

metrical information can be extracted from 3-D CAD models. Tooling information can be acquired from the Bill of Processes (BOP) (Figure 5.2).

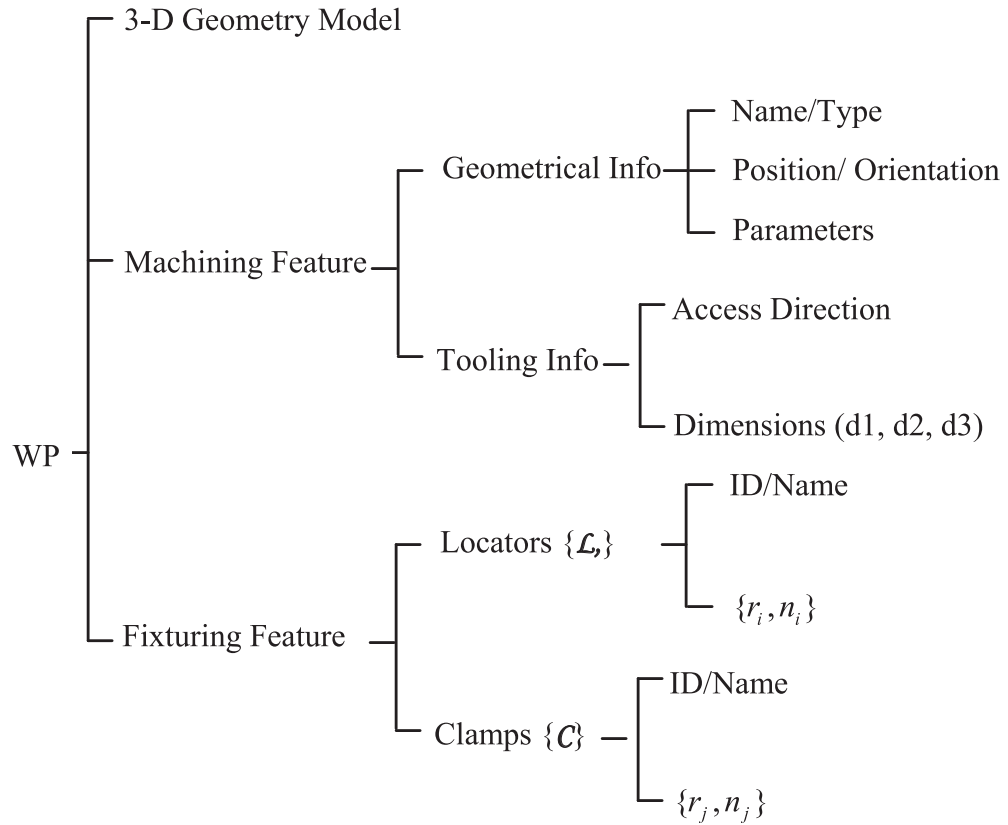


Figure 5.2: Representation of a machining feature.

Fixturing feature set could be thought as a set of locating features and clamping features, which could be described as:

$$FIX_SET = \{\mathcal{L}, \mathcal{C}\} \quad (5.7)$$

5.1.2 Workpiece Variation Model

The variation in the workpiece is mainly concerned with changes in dimension and minor topological variations, such as the change of normal directions of surfaces. For a giving acting surface, we have:

$$\vec{\mathbf{r}}'_k = \vec{\mathbf{r}}_k + \Delta\vec{\mathbf{r}}_k \quad (5.8)$$

$$\vec{\mathbf{n}}'_k = \vec{\mathbf{n}}_k + \Delta\vec{\mathbf{n}}_k \quad (5.9)$$

In these equations, $\vec{\mathbf{n}}'_k$ can be acquired directly from the solid model of the workpiece, and $\Delta\vec{\mathbf{r}}_k$ can be solved by the proportional calculation according to the acting point location in the acting surface.

Figure 5.3 shows the variation for a fixturing feature in the workpiece.

In this section a generic workpiece model and workpiece variation model have been discussed. This ground fundamental information for the ongoing fixture variation design process.

5.2 Fixture Modeling

To derive the fixture of a given part from the exist fixture of the typical part which belongs to the same class, the exist fixture configuration, which includes the fixture assembly, fixture components and their relationship, must be represented.

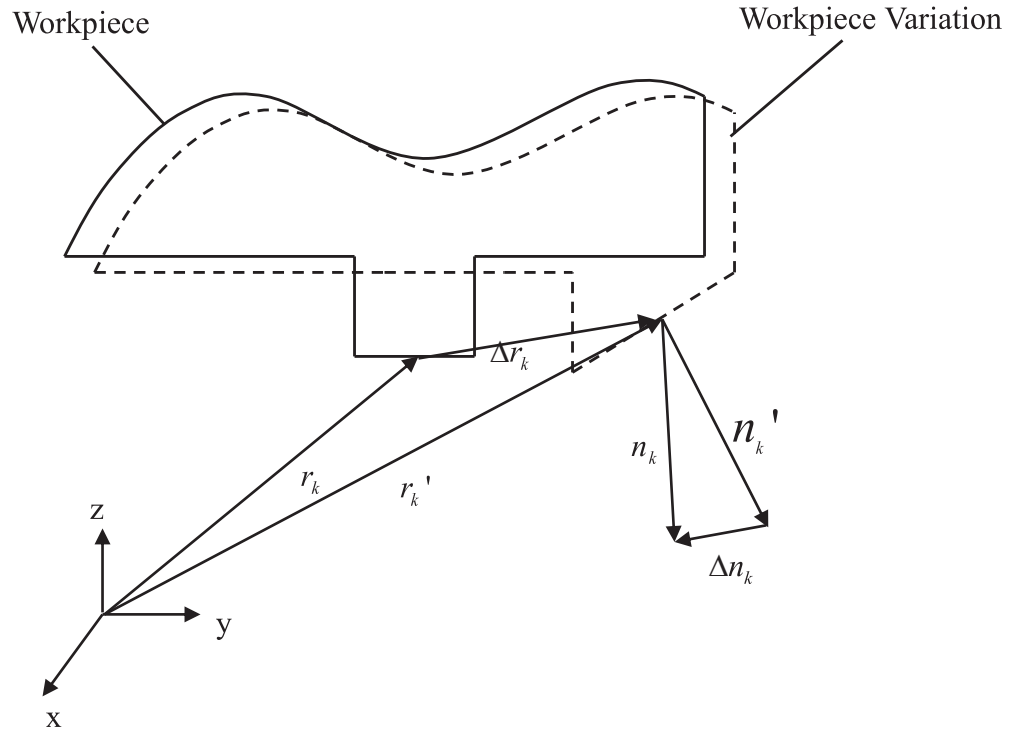


Figure 5.3: Workpiece variation model.

5.2.1 Fixture Model

A fixture consists of several sub-assemblies. Each sub-assembly performs one or more fixturing functions. These sub-assemblies in a fixture are considered fixture units. In a fixture unit, all fixture components (elements) are connected with one another directly where only one element is mounted directly on the fixture base and one or more elements in the subset are contacted directly with the workpiece serving as the locator, clamp, or support.

Let \mathcal{F} denote the fixture, we have

$$\mathcal{F} = \{\mathcal{U}_i \mid i \in n_u\} \quad (5.10)$$

$$\mathcal{U}_i = \{\mathcal{E}_{ij} \mid j \in n_{ei}\} \quad (5.11)$$

then

$$\mathcal{F} = \{\{\mathcal{E}_{ij} \mid j \in n_{ei}\} \mid i \in n_u\} \quad (5.12)$$

where, \mathcal{U}_i denotes a fixture unit in a fixture. n_u is the number of units in a fixture \mathcal{F} . n_{ei} is the number of elements in unit \mathcal{U}_i .

Figure 5.4 is an example of a fixture unit. For this unit, support 0 is directly mounted on the fixture base; support n is connected with a clamp and the clamp is acted on the workpiece. In the 3 – D space, we have,

$$\begin{aligned} \mathcal{FB}^{\mathcal{G}} &= \{r_0, \theta_0\} & r_0, \theta_0 &\in \mathfrak{R}^3 \\ \mathcal{S}_0^{\mathcal{L}} &= \{r_{0,1}, \theta_{0,1}\} & r_{0,1}, \theta_{0,1} &\in \mathfrak{R}^3 \\ \mathcal{S}_1^{\mathcal{L}} &= \{r_{1,2}, \theta_{1,2}\} & r_{1,2}, \theta_{1,2} &\in \mathfrak{R}^3 \\ && \vdots & \\ \mathcal{S}_n^{\mathcal{L}} &= \{r_{n,n+1}, \theta_{n,n+1}\} & r_{n,n+1}, \theta_{n,n+1} &\in \mathfrak{R}^3 \end{aligned}$$

where, \mathcal{G} represents in the global coordinate system. \mathcal{L} represents in the local coordinate system. $r_0, r_{0,1}, \dots, r_{n,n+1}$ are the offset vectors of the origins of the

local coordinate systems. $\theta_0, \theta_{0,1}, \dots, \theta_{n,n+1}$ are the orientation angles for the local coordinate systems.

Finally, the unit contact point can be represented in the global coordinate system as:

$$\{r_k, n_k\}^T = \{r_0, \theta_0\} \{r_{0,1}, \theta_{0,1}\} \cdots \{r_{n,n+1}, \theta_{n,n+1}\}^T \quad (5.13)$$

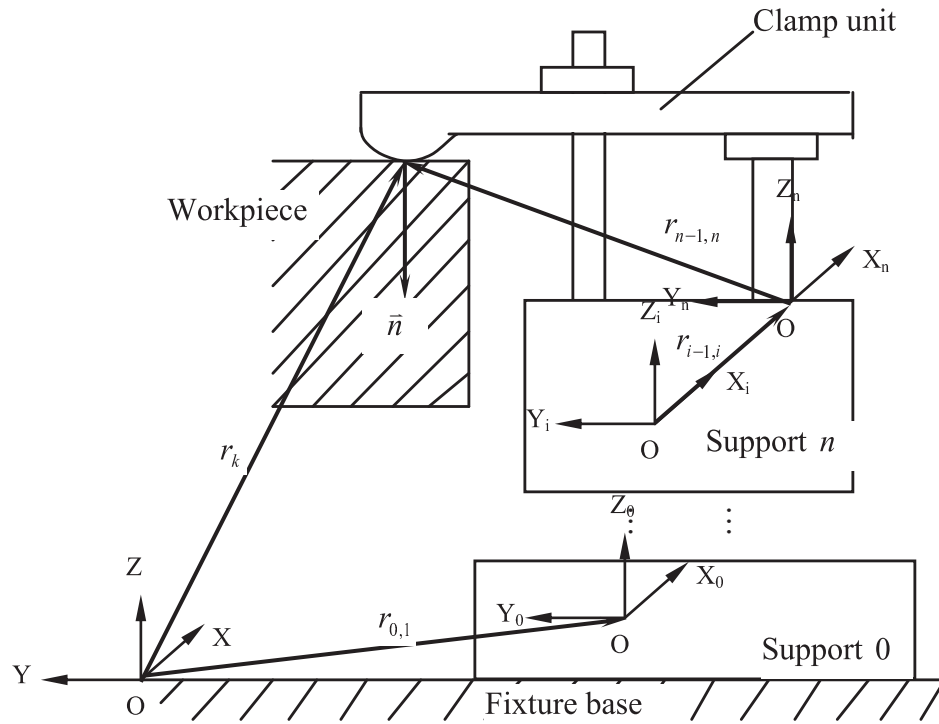


Figure 5.4: Acting point \vec{P} and external normal vector \vec{n} .

5.3 Fixture Variation Model

Basically, the position and orientation of the final function parts are decided by the dimension and orientation of all components. For each fixturing unit, we have

$$T_i = T_{i-1}T_{i-1,i}T_{am} \quad (5.14)$$

or,

$$T_i = T_{0,1}T_{1,2} \cdots T_{i-1,i}T_{am} \quad (5.15)$$

where,

$$T_{i-1,i} = \begin{bmatrix} n_{xi} & o_{xi} & a_{xi} & p_{xi} \\ n_{yi} & o_{yi} & a_{yi} & p_{yi} \\ n_{zi} & o_{zi} & a_{zi} & p_{zi} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.16)$$

$T_i(i = 1, 2, \dots, n)$ is the transformation matrix of the acting point in the global coordinate system. $T_{i-1,i}(i = 1, 2, \dots, n)$ are the relative transformation matrices between two adjacent components; and T_{am} is the transformation matrix of the acting marker which is attached to the functional component. The origin point is the acting point, the z direction is the acting direction, and direction of the other two axes can be arbitrary.

The assembly tree is searched bottom-up from the fixture representation. For each contacting component (locator or clamping component), we can determine its intermediate (link) components (supports), the constraints between them, and their corresponding transformation matrices. Therefore, for each functional unit, we have components $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_n\}$ and their relative transformation matrix $\{T_{0,1}, T_{1,2}, \dots, T_{n-1,n}\}$.

For each functional point and face, in terms of the contribution in the z -axis and the change in normal direction of the components, we have,

$$\vec{\mathbf{r}} = \vec{\mathbf{r}}_0 + \vec{\mathbf{r}}_{0,1} + \cdots + \vec{\mathbf{r}}_{n-1,n}$$

$$\vec{\mathbf{n}} = \vec{\mathbf{n}}_0 + \vec{\mathbf{n}}_{0,1} + \cdots + \vec{\mathbf{n}}_{n-1,n}$$

where,

$$\vec{\mathbf{r}}_0 = T_0 \vec{\mathbf{r}}'_0$$

$$\vec{\mathbf{r}}_{0,1} = T_0 T_{0,1} \vec{\mathbf{r}}'_{0,1}$$

⋮

$$\vec{\mathbf{r}}_{n-1,n} = T_0 T_{0,1} \cdots T_{n-1,n} \vec{\mathbf{r}}'_{n-1,n}$$

$$\vec{\mathbf{n}}_0 = T_0 \vec{\mathbf{n}}'_0$$

$$\vec{\mathbf{n}}_{0,1} = T_0 T_{0,1} \vec{\mathbf{n}}'_{0,1}$$

⋮

$$\vec{\mathbf{n}}_{n-1,n} = T_0 T_{0,1} \cdots T_{n-1,n} \vec{\mathbf{n}}'_{n-1,n}$$

We can conclude,

$$\Delta \vec{\mathbf{r}} = \Delta \vec{\mathbf{r}}_0 + \Delta \vec{\mathbf{r}}_{0,1} + \cdots + \Delta \vec{\mathbf{r}}_{n-1,n} \quad (5.17)$$

$$\Delta \vec{\mathbf{n}} = \Delta \vec{\mathbf{n}}_0 + \Delta \vec{\mathbf{n}}_{0,1} + \cdots + \Delta \vec{\mathbf{n}}_{n-1,n} \quad (5.18)$$

where,

$$\Delta \vec{\mathbf{r}}_0 = \Delta T_0 \vec{\mathbf{r}}'_0 + T_0 \Delta \vec{\mathbf{r}}'_0$$

$$\Delta \vec{\mathbf{r}}_{0,1} = \Delta T_0 T_{0,1} \vec{\mathbf{r}}'_{0,1} + T_0 \Delta T_{0,1} \vec{\mathbf{r}}'_{0,1} + T_0 T_{0,1} \Delta \vec{\mathbf{r}}'_{0,1}$$

⋮

$$\begin{aligned} \Delta \vec{\mathbf{r}}_{n-1,n} = & \Delta T_0 T_{0,1} \cdots T_{n-1,n} \vec{\mathbf{r}}'_{n-1,n} + T_0 \Delta T_{0,1} \cdots T_{n-1,n} \vec{\mathbf{r}}'_{n-1,n} + \cdots \\ & + T_0 T_{0,1} \cdots \Delta T_{n-1,n} \vec{\mathbf{r}}'_{n-1,n} + T_0 T_{0,1} \cdots T_{n-1,n} \Delta \vec{\mathbf{r}}'_{n-1,n} \end{aligned}$$

$$\Delta \vec{\mathbf{n}}_0 = \Delta T_0 \vec{\mathbf{n}}'_0 + T_0 \Delta \vec{\mathbf{n}}'_0$$

$$\Delta \vec{\mathbf{n}}_{0,1} = \Delta T_0 T_{0,1} \vec{\mathbf{n}}'_{0,1} + T_0 \Delta T_{0,1} \vec{\mathbf{n}}'_{0,1} + T_0 T_{0,1} \Delta \vec{\mathbf{n}}'_{0,1}$$

⋮

$$\begin{aligned} \Delta \vec{\mathbf{n}}_{n-1,n} = & \Delta T_0 T_{0,1} \cdots T_{n-1,n} \vec{\mathbf{n}}'_{n-1,n} + T_0 \Delta T_{0,1} \cdots T_{n-1,n} \vec{\mathbf{n}}'_{n-1,n} + \cdots \\ & + T_0 T_{0,1} \cdots \Delta T_{n-1,n} \vec{\mathbf{n}}'_{n-1,n} + T_0 T_{0,1} \cdots T_{n-1,n} \Delta \vec{\mathbf{n}}'_{n-1,n} \end{aligned}$$

Δ is the difference of variation. $\vec{\mathbf{r}}$ is the acting vector. $\vec{\mathbf{n}}$ stands for the normal direction of the functional face in global coordinate system; $\vec{\mathbf{r}}_0, \vec{\mathbf{r}}_{0,1}, \cdots, \vec{\mathbf{r}}_{n-1,n}$ stand for the point on the contact surface in global coordinate system; $\vec{\mathbf{n}}_0, \vec{\mathbf{n}}_{0,1}, \cdots, \vec{\mathbf{n}}_{n-1,n}$ are the differences of the normal direction of the supporting surface and the supported surface in global coordinate system.

In order to solve the equation, a rule base was used to decide the modification preference between the components. When the modification components have been determined, all the $\Delta \vec{\mathbf{r}}_{0,1}, \Delta \vec{\mathbf{r}}_{1,2}, \cdots, \Delta \vec{\mathbf{r}}_{n-1,n}$ and $\Delta \vec{\mathbf{n}}_{0,1}, \Delta \vec{\mathbf{n}}_{1,2}, \cdots, \Delta \vec{\mathbf{n}}_{n-1,n}$ are available. Therefore we have

$$\Delta \vec{\mathbf{r}}'_{i-1,i} = T_i^{-1} \Delta \vec{\mathbf{r}}_{i-1,i} \quad (i = 1, 2, \cdots, n) \quad (5.19)$$

$$\Delta \vec{\mathbf{n}}'_{i-1,i} = T_i^{-1} \Delta \vec{\mathbf{n}}_{i-1,i} \quad (i = 1, 2, \cdots, n) \quad (5.20)$$

where, $\vec{\mathbf{n}}'_{i-1,i}$ is its counterpart of $\vec{\mathbf{n}}_{i-1,i}$ in the local coordinate systems, and T_i is the transformation matrix of the local coordinate system i relative to the global

coordinate system. More specifically, we have,

$$\begin{bmatrix} \Delta \vec{r}'_x \\ \Delta \vec{r}'_y \\ \Delta \vec{r}'_z \\ 1 \end{bmatrix} = T \begin{bmatrix} \Delta \vec{r}_x \\ \Delta \vec{r}_y \\ \Delta \vec{r}_z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \Delta \vec{n}'_x \\ \Delta \vec{n}'_y \\ \Delta \vec{n}'_z \\ 1 \end{bmatrix} = T \begin{bmatrix} \Delta \vec{n}_x \\ \Delta \vec{n}_y \\ \Delta \vec{n}_z \\ 1 \end{bmatrix}$$

where,

$$T = \begin{bmatrix} n_{xi} & o_{xi} & a_{xi} & p_{xi} \\ n_{yi} & o_{yi} & a_{yi} & p_{yi} \\ n_{zi} & o_{zi} & a_{zi} & p_{zi} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To determine the orientation of fixture component i in the global coordinate system $OXYZ$, the direction cosines of the first two axes of the local coordinate system are calculated as,

$$\begin{bmatrix} n_{xi} \\ n_{yi} \\ n_{zi} \\ 1 \end{bmatrix} = T \begin{bmatrix} n'_{xi} \\ n'_{yi} \\ n'_{zi} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} o_{xi} \\ o_{yi} \\ o_{zi} \\ 1 \end{bmatrix} = T \begin{bmatrix} o'_{xi} \\ o'_{yi} \\ o'_{zi} \\ 1 \end{bmatrix}$$

where,

$$T = \begin{bmatrix} n_{xi-1} & o_{xi-1} & a_{xi-1} & p_{xi-1} \\ n_{yi-1} & o_{yi-1} & a_{yi-1} & p_{yi-1} \\ n_{zi-1} & o_{zi-1} & a_{zi-1} & p_{zi-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.4 Standard Components Database

The thumb rule of fixture design is to select the standard components as much as possible. It not only can shorten the design and fabricating time of a fixture, so as to decrease the labor and cost, but it can also contribute to the standardization of whole fixture assembly. In the design of dedicated fixtures discussed in this

Instance	ϕdia_a	thk_b	oal_c	ϕdia_d
4_41475	9.7	4.78	12.7	6.375
4_45065	10	6	12	6.025
4_45066	10	8	14	6.025
4_45067	13	6	14	8.025
4_45068	13	6	16	8.025
4_45455	13	7	20	10
4_45060	19	10	25	12.025
4_45061	19	12	28	12.025

Table 5.1: Example of a family table

dissertation, a group of standard components (locators/clamps), extensively used in industry, have been modeled, and a database including their generic models and instance has been constructed.

5.4.1 Standard Component Database

For a certain detailed type of locator/clamp, there may exist many component instances with different standardized dimensions. In fact, the standardized component library is mostly composed of families of components (also called "table-driven" components). A family of components is a collection of similar components varying in size or with slightly different detailed features. Every family has a generic base model that all instances of the family resemble. Table-driven families provide a very simple and compact way of creating and sorting large numbers of standardized components and facilitate interchangeability of components in an assembly. Figure 5.5 is a generic model of a locator and its dimension; Table 5.1 shows all instances of this locator and its specific dimensions. Figure 5.6 is an example of some locators and clamps listed in the database.

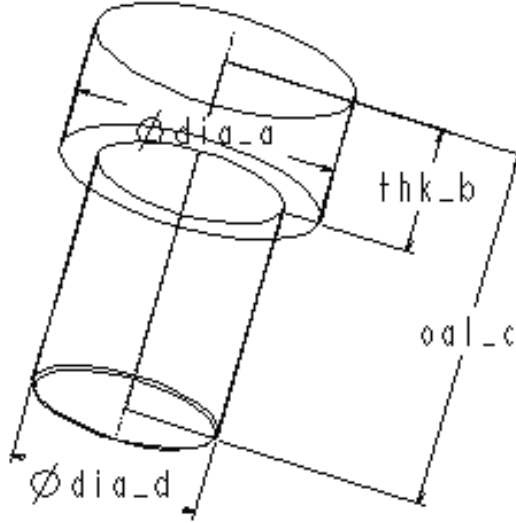
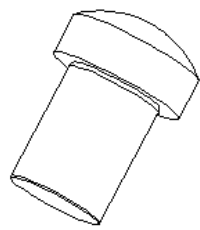


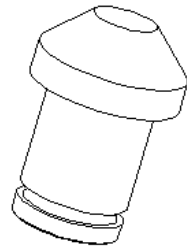
Figure 5.5: Generic model of a locator.

5.4.2 Selection of the Standard Components

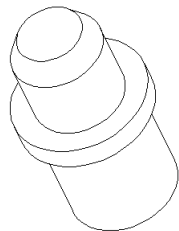
The first step of fixture design discussed in this dissertation is to select the proper standard components, which includes standard locators and clamps. The appropriate instance of the generic standard component to suit the given workpiece, which is the same type in the existing fixture, must be determined. Obviously, to maintain sufficient stiffness and strength so as to assure the location precision and clamping force is the basic principle in selecting the standard components. The algorithm of instance selection should be concerned more with choosing the primary design dimension, because it is the key factor that affects how the standardized component suits the workpiece. To easily define the rule for choosing the primary design dimension of the component, the concept of the virtual reference cubic box (VRCB) is introduced. During the design, the edge length d_{EP} of the VRCB is set equal to the length of the external bounding box of the workpiece of the machining orientation,



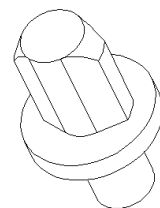
(a) Radia rest but-
ton



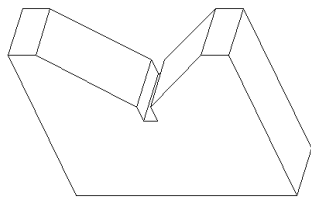
(b) Flat rest button



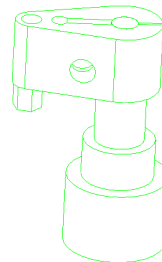
(c) Round pin



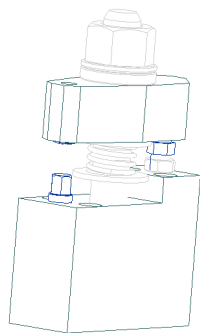
(d) Diamond pin



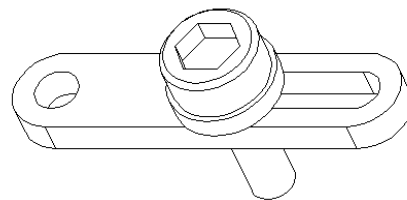
(e) V block



(f) Hydraulic clamp

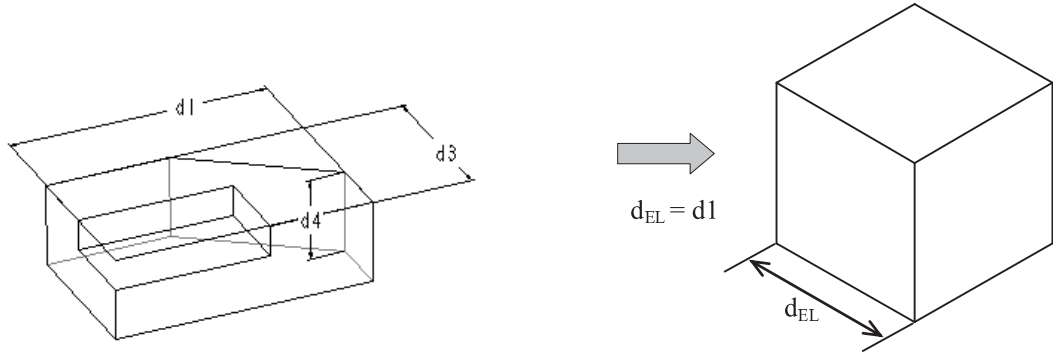


(g) T-shaped clamp



(h) Hook clamp

Figure 5.6: Standard components (locators/clamps).



(a) Workpiece example with external dimension displayed

(b) Virtual reference cubic box

Figure 5.7: Virtual reference cubic box (VRCB)

as the example shown in Figure 5.7.

For the workpiece and the specified fixturing surface/point, it is assumed that the component is to locate/clamp the corresponding VRCB at the ideal position - locate at the bottom surface of VRCB (bottom locating), locate at the middle height of the side surface of VRCB (side locating), clamp at the middle height of the side surface (side clamping), and clamp at the top surface (top clamping). Under such an assumption, the estimate of the primary design dimension of the component can be obtained by the expression,

$$\tilde{d}_{PD} = dim_ratio \times d_{EL} \quad (5.21)$$

where \tilde{d}_{PD} is the estimate of the primary design dimension of the standard component, and d_{EL} is the edge length of the VRCB.

After \tilde{d}_{PD} has been determined, the instance whose d_{PD} value is closer to \tilde{d}_{PD} is selected as the locator/clamp for the current workpiece.

5.5 Support Modeling

To carry out the geometrical and topological modification of a fixture component (support), the full understanding and analysis of the functional feature of the support is required, such as the locating holes and surfaces and screw holes, etc. A relationship between the total component dimensional variation (the height of component) and geometrical information of the component feature are constructed to drive the modification of these features.

5.5.1 Support Model

By the thorough study and analysis of the component functional features, we define the functional feature type of the component as follows:

- Supported surface, which is usually supported by the fixture base boss or the other components; it can be the datum face of the local frame;
- Supporting surface, which is usually used to support the locating part or the other components. The distance between the supported surface and the supporting surface can be employed to infer the other geometrical data of the component;
- Locating holes, which are associated with the supported surface or the supporting surface;
- Screw holes, associated features of the supported surface, or the supporting surface;
- Support, which is used to enforce the component; and

- Other features, which greatly contribute to the function of component.

5.5.2 Parametric Dimension Definition

The relationship between the geometries includes the topological and dimensional information, which can be easily retrieved from the solid modeler. When modifying the component, the topological information need not be changed; rather, the dimensional modification of the components is mainly need to be implemented. Here, we focus on the expression and representation of the relationship between the geometries.

As to the modification of the fixture components, different types of dimensions are defined, as shown in Figure 5.8. One is the size dimension, such as the diameter of a hole ($\phi d6, \phi d16$), the height of cylinder, and so on. Another is the locating dimension, which is used to specify the location of some features, such as the locating hole and screw hole ($d4, d5$), which are associated with the supported surface or supporting surface.

In all, those feature dimensions can be regarded as a dimension set D , which is composed of d_1, d_2, \dots, d_n , can be expressed as

$$D = f(d_1, d_2, \dots, d_n)(i = 1, 2, \dots, n) \quad (5.22)$$

The element of this set, $d_i(i = 1, 2, \dots, n)$, is defined as the function of \vec{n} , h and A ,

$$D_i = f(\vec{n}, h, A) \quad (5.23)$$

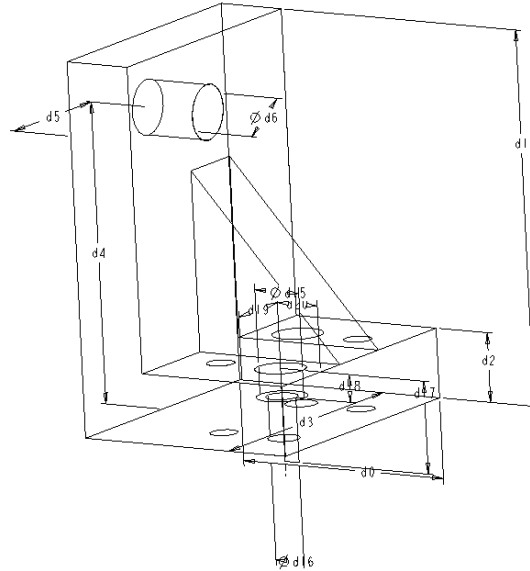


Figure 5.8: Dimensions of a support.

where \vec{n} , h , and A are the normal direction, height of the component and the accessible area of the supporting surface in the local coordinate system.

With \vec{n} , h and A being altered, the $d_i (i = 1, 2, \dots, n)$ can be changed continuously or discretely. Therefore, we define the functions as follows.

- *Continuous function*: the relationship between dimensions d_i and \vec{n} , h , and A can be represented by an expression clearly, such that these dimensions should be changed with \vec{n} , h and A ;
- *Discrete function*: the expression between dimensions d_i and \vec{n} , h , and A are represented in a discrete way; and
- *Constant value*: the value of the variable is set to a constant.

Through the boundary list of the supporting surfaces, the adjacent faces are

determined. A solid modeler, Unigraphics, is employed to change the direction of the supporting surface. All modification is limited to adjacent faces.

After modifying the supporting surface, the height of the component should be modified and the fixture boss moved along the x and y axes.

In order to suit the requirements for the dedicated fixture design, the fixture models and design method have been discussed. The fixture derivation process from the existing one can be divided into two levels. In the assembly level, mating conditions have been employed to represent the fixture assembly, and by the satisfaction of these constraints, the relationship between the components has been maintained, and the primary design dimension for each component can be acquired. In the component level, we use a parametric model to describe the internal relationship among dimensions of features. From primary dimension the component (support) can be generated by this parametric model.

Chapter 6

Implementation

In this chapter we discuss integration with other functionalities of the software package. The software is a whole virtual manufacturing system aimed at validating the manufacturing process. It eventually gives estimates of labor, cost, and production line requirements. The chapter details aspects ranging from conceptual fixture design to basic validation, and from detailed fixture design to fixture verification. A case study with Delphi fixture variation will be presented in the end of this chapter.

6.1 System Integration

6.1.1 Conceptual Fixture Design

Conceptual fixture design is composed with five components - Initialization, layout specification, conceptual design, tombstone size change and feasibility checking.

Initialization - as a component of the whole system, conceptual fixture design need acquire the setup and machining information of the specified workpieces. These information include setup and load information like locating datum and machining information like predefined tool paths for each machining feature. All this infor-

mation is the outputs of precede process and the input of the conceptual fixture design.

Workpiece Layout - specify the layout of the workpiece on the fixture base, how many faces of fixture base should be used to load workpiece, how many workpiece should be load for each face and which load should be loaded for each position.

Conceptual Fixture Design - after specified the workpiece layout on the fixture base, a fixture configuration could be generated by predefined workpiece position and orientation. In this stage, you can also modify the position and orientation for each workpiece part.

Change Fixture base Size - the dimensions could be changed according to user's preference.

Feasibility Checking - after the fixture configuration has been generated, all the fixturing issues should be checked. This include if the selected machine on which the fixture will be mounted has the adequate space (volume) to accomodate the fixture. The second concerns is about the machine moving range capability - if the machine table and head have the long moving range capability enough to access all the machining features. The last concerns is to check the orientation of the machining feature - to determine if the machine head has the same orientation as the machining feature so as to check if this machining feature can be accessed or not under current fixture configuration.

6.1.1.1 Initialization

Before conceptual fixture design, all the information for the workpiece machining should be prepared either in the production design stage or in the manufacturing

customization stage. This information is essential for the conceptual fixture design and verification. It includes,

- Locating datum of workpiece for the workpiece position and orientation determination. All the workpieces should have locating datum - primary locating, secondary locating and tertiary locating datum.
- Machining feature list. Strut arm mill, steering arm, lower ball joint and upper ball joint, to name a few.
- Predefined tool path. For conceptual fixture design and workpiece machining, the entire tool path for each machining feature should be predefined.
- Selected machine. Only specified the machine which will be used to machine these workpieces, the machine envelop and machine table and head moving range could be checked to determine if the selected machine has the space and moving range capability for the current fixture settings and loads.

6.1.1.2 Layout Specification

Different layout could result in different fixture configuration. This is implemented through a pre-defined parametric model. For example, a tombstone has four faces. Users can specify layout settings for each face.

Tombstone has four faces which could be used to load workpieces. They are 0° face, 90° face, 180° face and 270° face. users can specify layout settings for each face. System provide four load modes for each tombstone face. They are: mode 1, load one part in one face; mode 2, load two parts in one face; mode 3, load three parts in one face, and mode 4, load four parts in one face. In addition, for each load

position, there are eight options to load. They are, Left part load A, Left part load B, Left part load C, Left part load D, Right part load A, Right part load B, and Right part load C and Right part load D.

6.1.1.3 Conceptual Design

In this step, the fixture configuration is generated according to the position and orientation requirement.

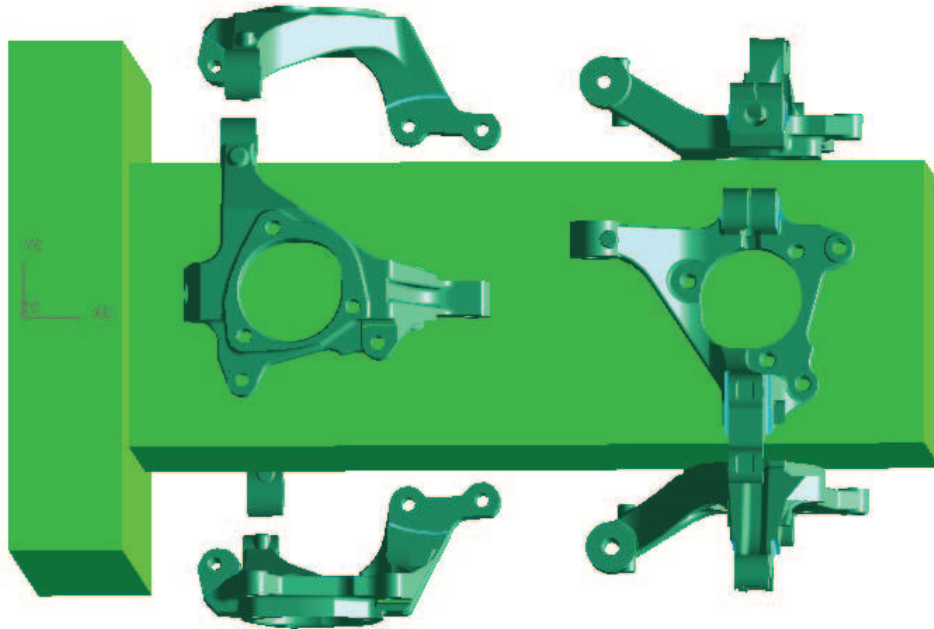
In this step, user can also change the orientation and/or position for each workpiece. User need select a workpiece, and then type in the number of offset he expected to move, or, the angle he expected to rotate a knuckle around the bore. Figure 6.1 shows the initial workpiece layout and modified layout after the lower parts have been rotated around the bore 90° degrees.

The dimensions of the tombstone can be modified as users preferences. An user interface is designed for this purpose. User can input the values of various dimensions, the tombstone dimension is changed according to user's input.

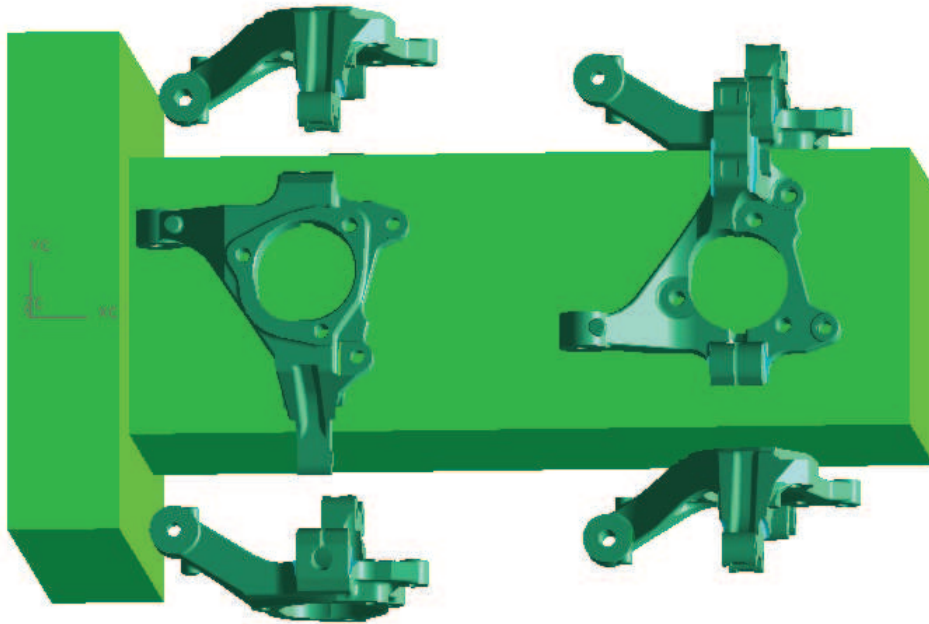
6.1.1.4 Feasibility Checking

The designed workpiece layout is required to be checked and verified as to the machine capabilities and machining feature accessibility. These include,

- *Machine envelop checking*: Whether the selected machine has adequate space /envelop to accommodate the specified and designed fixture is to be checked. Especially for the fixture design of knuckles, machine space/envelop is a key issue when the multiple knuckles have been put on the tombstone. Here the volume the tombstone and knuckles combined was used to compare with the



(a) Initial knuckles layout on the tombstone



(b) Knuckles layout after the lower parts have been rotated around the bore 90° degrees

Figure 6.1: Conceptual fixture design and layout modification.

machine envelope to check if the selected machine has this capability, as displayed in Figure 6.2.

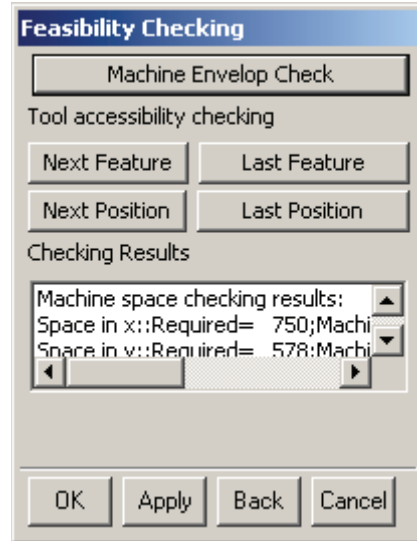


Figure 6.2: Conceptual workpiece layout feasibility checking.

- *Machine moving range checking*: It is essential that the machine has the adequate head/table moving range capabilities to access the machining features on the multiple workpieces. The whole predefined tool paths were traversed to search the maximum moving range requirement for current fixture configuration. And then these numbers are compared with the machine moving ranges.
- *Machining feature accessibility checking*: Whether or not the machine head can access the machining feature in current fixture configuration is another issue for the fixture design. All the access directions for all the machining features were traversed and are compared with the machine head access direction; the checking results were shown on the result window.

- *Cutting simulation:* During machining feature accessibility checking, the cutting simulation are accomplished through loading the cutters in the predefined tool path positions and orientation. This step-by-step simulation gives the fixture designer a virtual view and good sense on the machining and interference among fixtures, workpieces and cutters (Figure 6.3).

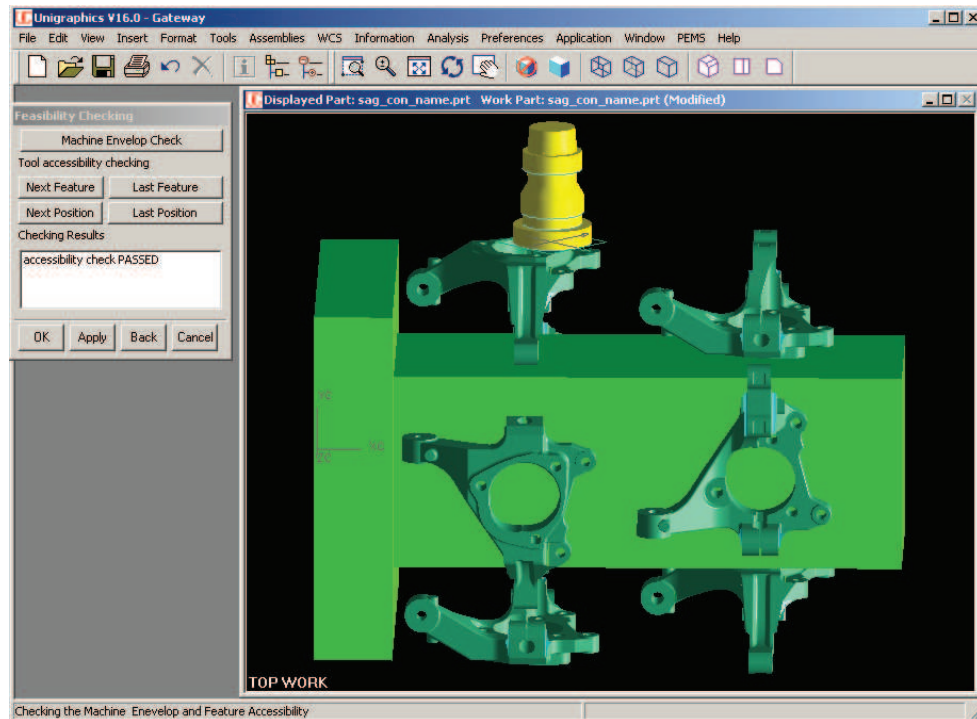


Figure 6.3: Machining simulation.

6.1.2 Detailed Fixture Design

Detailed fixture design, as a module of engineering package, functions to design details for the fixtures of knuckle, caliper, master cylinder and ABS, checks and verifies the validity of the designed details during the machining process, and modifies

the fixture automatically or manually. It provides a tool to the process engineer to use on the fixture design. It also service as the essential part to evaluate and validate the manufacturing systems for knuckles, calipers, master cylinders and ABS. Systematically we have the following specification for the detailed fixture design.

- The objects of detailed fixture design are knuckles, calipers, master cylinders and ABS.
- It provides a tool for detailed fixture design so that the process engineer can use and design the fixture details for the part to be machined.
- It provides the functionalities of checking and verifying to validate the feasibility of designed fixture details.
- The process engineer can modify the designed fixture details on his preference.

6.1.2.1 Functional Requirement Analysis

The functional requirement for the detailed fixture design could be classified into the functions for fixturing specification, design and verification.

Functions for fixturing specification include:

- Fixturing information (locating/clamping surfaces, points and orientations) could be either specified interactively by CAD package such as Unigraphics, or read from 3-D model. In the later situation, the fixturing information (locating/clamping surfaces, points and orientations) must be modeled into CAD 3-D model.

- The specified fixturing mode information could be saved so that it could be accessible for the fixture design of new product.

For detailed fixture design, three modular functions should be included - fixture components database construction and management, fixturing unit generation and placement, and fixturing unit modification.

Fixture components database construction and management is the essential part in the detailed fixture design. Only the fixture component fixturing information such as supporting surface/point and supported surface/point has been stored in the system, the detailed fixture design can call these components and their fixturing information, and then generate the fixturing unit and place them in the desired position and orientation. We have these functions for fixture component database construction and management as below.

- Fixture components (locators, clamps and supports) should be classified into different categories so that they could be called and positioned in the appropriate location and orientation automatically.
- For standard components, a family table is constructed for them. This could make the member parts search-able.
- Fixture component database is expandable and has interfaces that permit the user to add, delete and edit fixture component.
- Fixture component database has some robustness so that the user can add the new components at any time during design process. For example, when the user modified the fixturing unit including fixturing components, he can add the modified parts into database if he wants.

- If possible, the fixture components database should have interface to input the knowledge of process engineer and fixture designer. This make the knowledge of process engineer and fixture designer can be shared with other people.

The fixturing unit generation and placement is the central part of the detailed fixture design. It calls the fixture components database, reasons the most suitable candidates to the current locating/clamping requirement and then select one and place it in the designated position and orientation.

- The system reasons the type and subtype that are the most suitable to the current locating/clamping requirements, and then select the default candidate. If user wants, a graphical interface is provided so that the user can specify.
- The system generates the most suitable family member for standard components. The user can change through a graphical window that should list all candidates.
- For supports, the system generates the dimensions to reach the desired fixturing height. The user can modify the support model through interactive functions provided by CAD package if he wants.

Fixturing unit modification includes functions as below.

- Function to permit the user to modify the system generated fixture design. This include that the user can add, delete and replace the fixturing unit.
- The user can change the fixture component type and subtype, family member, and dimensions.

- The user can edit the mounting features between fixturing unit and fixture base. After the mounting feature in the fixturing unit has been modified, the corresponding mounting features in the fixture base should reflect the modifications.
- The location/orientation of fixture unit can also be changed. The locating/orientation of mounting features in the fixture base can be changed according to the location/orientation of fixturing unit.

At last, fixture verification includes interference checking. The system can check the interference between workpiece and fixture base, cutting tool and workpiece/fixture base during the operation. When an interference is found, a warning message would be issued.

6.1.2.2 Approaches and System Interfaces

The system is composed with five straightforward modules - fixturing information specification, fixture unit candidate selection and dimension customization, fixture unit position and orientation, verification, and modification. They will be discussed module by module. The detailed fixture design menu is as Figure 6.4.

Fixturing Information Specification

User development functions provided by CAD packages such as UFUNC are called to let user specify the fixturing surfaces, fixturing points and orientation references (points or surfaces). If fixturing information had been modeled into CAD 3-D model, it can be retrieved. Consequently, the specified fixturing information can be stored for the coming call.

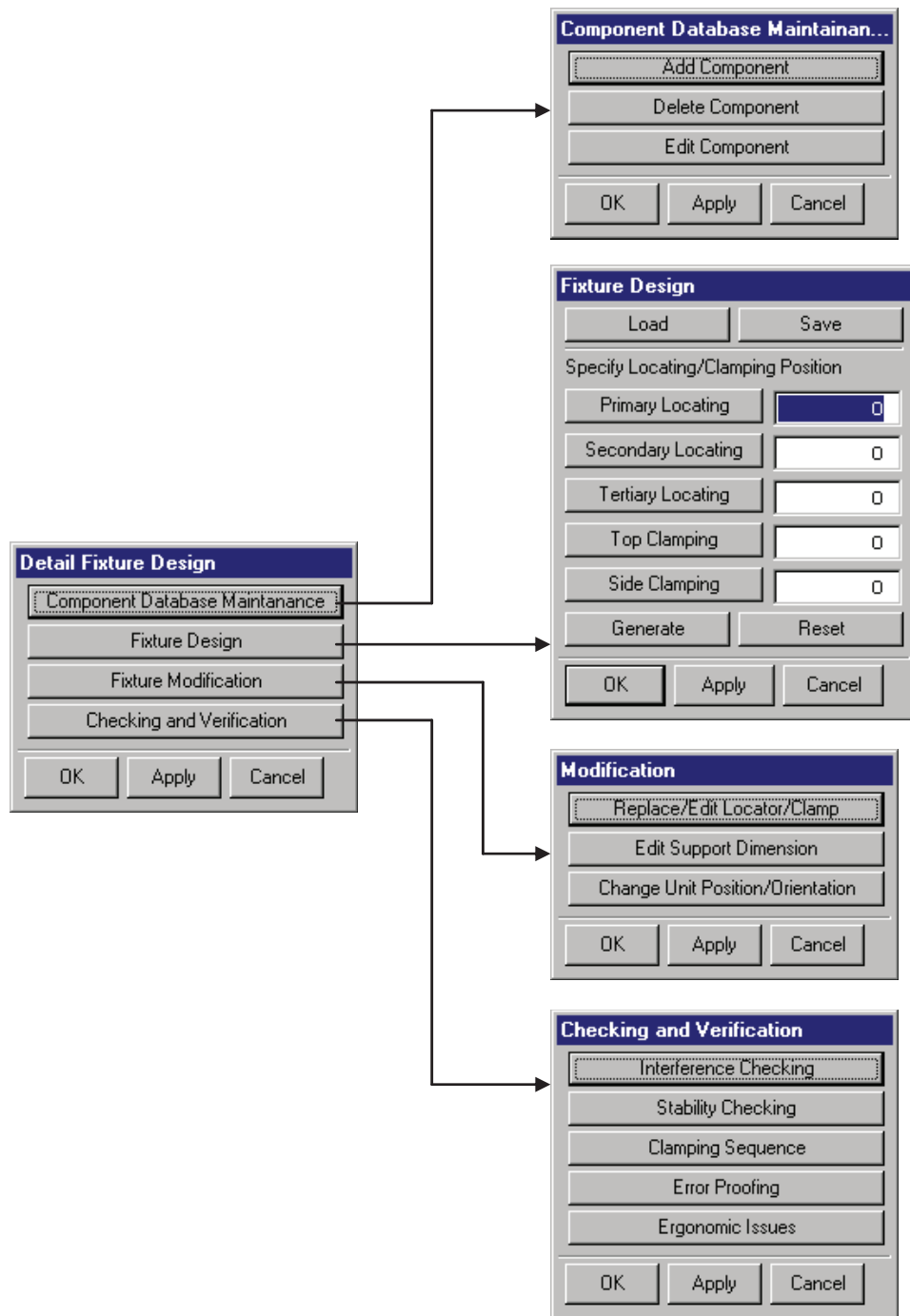


Figure 6.4: System interface for detailed fixture design.

Fixture Component Database Construction

In general, the fixture components can be classified into locators, clamps and supports. Locators and clamps can be categorized into different types and subtypes. For example, clamps can have types of mechanic and hydraulic. For each type, they can have subtypes of skinny clamps, round clamps and other types. Skinny clamps are used when a straddle mill has been specified for the knuckle steering arm. For the given type and subtype, a family table is used to model all possible dimensions.

Fixturing Unit Determination

Fixturing unit is composed with the locator/clamp and the support. The hydraulic cylinder is considered as support. The fixturing unit determination includes the decision of the type and size of the locator/clamp and the support.

The types and subtypes of the locator/clamp are decided by the locating/clamping functions. They are usually associated with the specified locators/clamps. For example, in the CNC machine, the hydraulic T-shape clamps are used to clamp the workpiece. If the straddle mill is used to machine the strut arm, the skinny detail clamp will be used to clamp the strut arm. Of course, the user can change the type and subtype of the locator/clamp as his/her preference.

The product of the workpiece design dimension (usually the diagonal length of bounding box) and `pdm_ratio` is referenced to compare with the primary dimension of the locator/clamp, the family member whose primary dimension is mostly close to the product is selected as the candidate locator/clamp. User can also modify the selected candidate if he wants.

The type of support can be decided by the support-locator/clamp relationship. Usually for a specified locator/clamp, we have limited number of supports that can

be used to support them. The relationship of locator/clamp with the support not only describes the capability of the given locator/clamp can mount on the specified support, it also preserves the associative features between them, such as mounting holes.

The next step is to decide the geometrical size of the support. As the same method to decide the size of locator/clamp, the product of the workpiece reference dimension and `pdm_ratio` is used as the reference to the driven dimension of the support. If the support has a hydraulic cylinder, take it as the standard component - a family member is selected.

The fixturing height also needs to be taken considered in the process of support dimension determination. Both for the general support and for the hydraulic cylinder, a spacer is used to reach the desired fixturing height.

The fixturing unit mounting process is sequentially behaved. First the locator/clamp is mounted on the workpiece based on the defined constraints in the database. Then the support is mounted on the locator/clamp according to the defined mounting relationship.

After the support has been mounted, the mounting geometry (holes, bosses) will be copied into fixture base and the corresponding mounting geometry on fixture base will be created, and the mounting features on fixture base will be associated with the mounting features of the support. By this associative relationship, the mounting features will be automatically modified to reflect the fixturing unit modification.

Fixturing Unit Modification

User can modify each component of a designed fixture configuration through standard functions provided by CAD packages. Then the regeneration of a fixture

configuration is behaved to assure correct locating/clamping positions and orientations.

Verification

Finally verification such as interference checking can be carried out. As in the same process behaved in the phase of conceptual design (Section 6.1.1.4), machine moving range check, machining accessibility verification and machining simulation can be proceeded to assure final feasibility of the designed fixture.

6.2 Variation Fixture Design Process

Figure 6.5 shows the variation fixture design process. Five major steps are involved in this process.

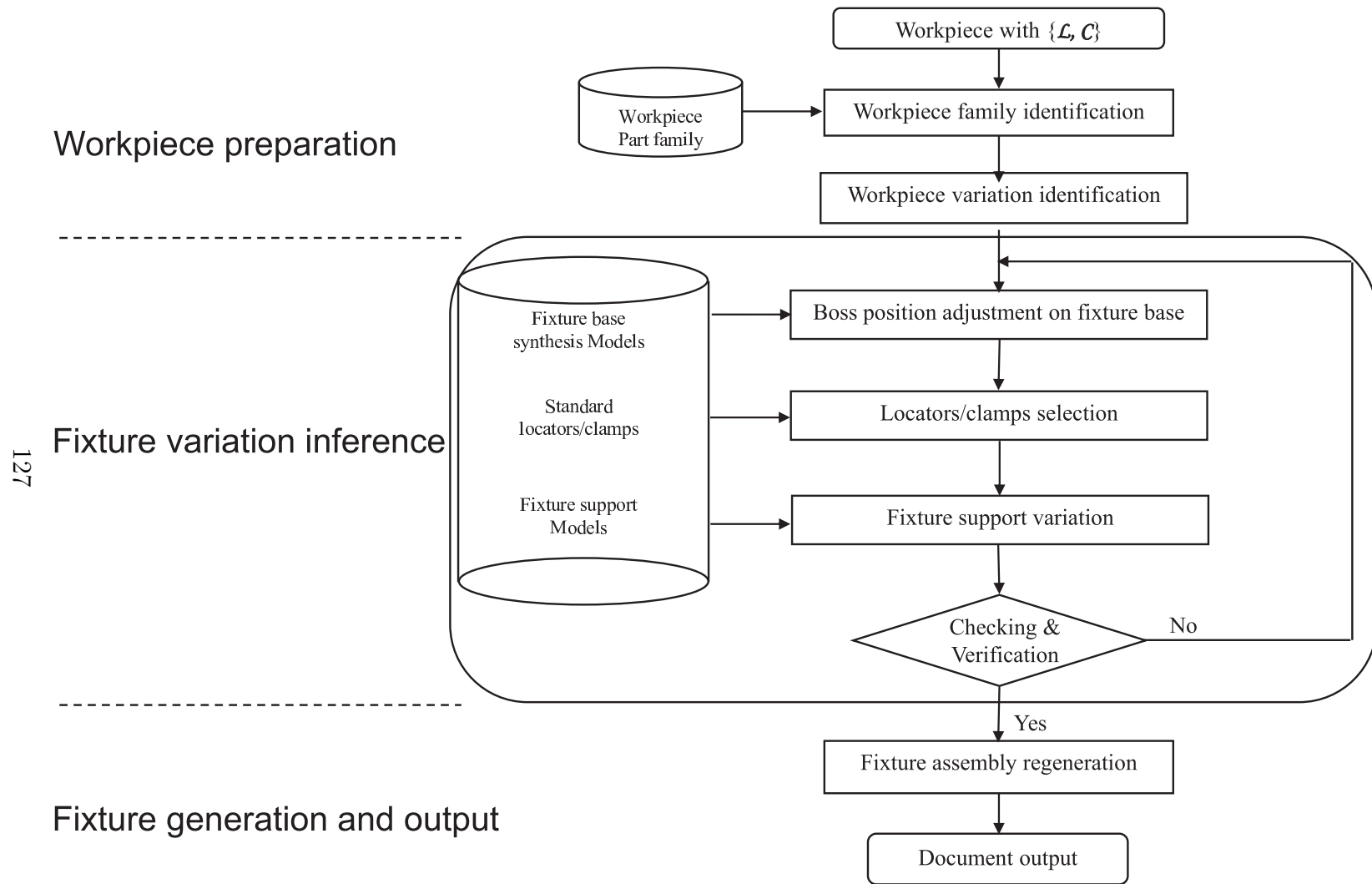


Figure 6.5: Variation fixture design process.

A new workpiece with locating and clamping information $\{\mathcal{L}, \mathcal{C}\}$ will be loaded first. The workpiece part family general models will be searched and the closest one will be selected as the candidate general model with which the new workpiece will be compared.

The delta values in terms of fixturing information between the new workpiece model and the candidate models will be calculated.

Then these variation values will be mapped onto the fixture components, which include the standard components such as locators and clamps, supports and fixture base. This variation process is addressed in the previous section.

After that the component itself will be regenerated by its parametric model, which has already been discussed in the previous sections.

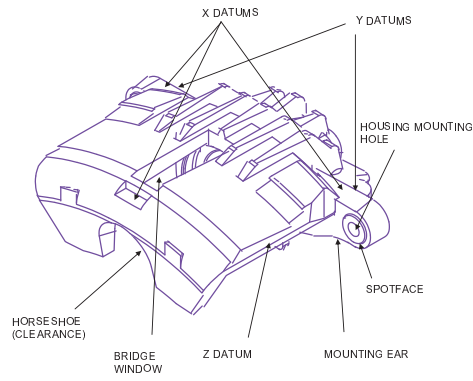
The last step is the fixture regeneration and the documentation.

6.3 A Case Study

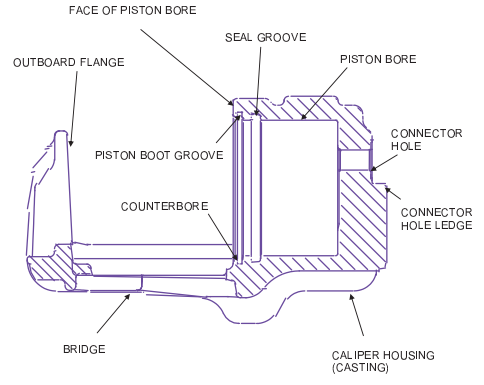
The models and design algorithms discussed in this chapter are applied on a workpiece. The object is to derive the fixture for a part family member from the exist fixture models for this part family. The workpiece and its variations, fixture and its variations will be discussed.

6.3.1 Workpiece and Workpiece Variation

Figure 6.6a and 6.6b are fixturing features and machining features, respectively, of a caliper. Table 6.1 lists the machining features. Table 6.2 lists the fixturing features for calipers from different families.



(a) Fixturing features (locating datum) of a caliper



(b) Machining features of a caliper

Figure 6.6: Fixturing features and machining features of a caliper (Courtesy of Delphi).

MFG FEATURES	Position/Orientation	Dimension	Tool #
BORE	(61, 0, 17)	60.50 × 46.40	OP25T
OUTBOARD FLANGE	(0, 0, 0)	NONE	OP30T
MTG HOLE	(86.5, 92, 12)	8.30 × 16.80	OP35T
MTG HOLE	(86.5, -92, 12)	8.30 × 16.80	OP55T
CONN HOLE	(138.15, 28, 19.5)	4.75 × 25	OP60T
BLDG HOLE	(132.15, 0, 30.5)	8.917 × 12.25	OP80T

Table 6.1: Manufacturing features from a caliper of 540 family (Courtesy of Delphi).

Fix Feature	540	544	543	777
LOC_X1	(95,87,0)	(86.5,77,0)	(92.25,84.5,0)	(94.2,86.5,0)
LOC_X2	(95,-87,0)	(86.5,-77,0)	(92.2,-84.5,0)	(94.2,-86.5,0)
LOC_X3	(-11,0,24)	(-11.5,0,-10)	(6.5,0,-23.52)	(2.2,0,-24.56)
LOC_Y1	(83,43,-17)	(79.5,43,-6)	(-13.5,62,21.48)	(122.2,56.84,-1)
LOC_Y2	(83,-43,17)	(79.5,-43,-6)	(-13.5,-62,21.48)	(122.2,-56.84,-1)
LOC_Z1	(51,76,-2)	(48,71.3,-4)	(86.5,0,-41)	(70,72.5,-2)
CL_EAR1	(95,87,27)	(86.5,77,24)	(92.25,84.5,26)	(94.2,86.5,26)
CL_EAR2	(95,-87,27)	(86.5,-77,24)	(92.2,-84.5,26)	(94.2,-86.5,26)
CL_WIN	(69,0,-19)	(57,0,-17.5)	NONE	NONE
CL_TOP	(-71,0,-13.6)	NONE	(-16,0,25.52)	(2.2,0,49.55)
CL_Z	(51,-76,-2)	(48,-71.3,-4)	NONE	(70,-72.5,-2)
CL_SIDE	NONE	(-11.5,59,24)	(124.9,35,0)	(-16,56.84,-1.06)
CL_SIDE	NONE	(-11.5,-59,24)	(124.9,-35,0)	(-16,-56.84,-1.06)

Table 6.2: Fixturing features of calipers (Courtesy of Delphi).

6.3.2 Support Model and Its Variation

The datum frames could be used to represent these coordinate systems and markers (Figure 6.7). The offset of three marker datum from coordinate system datum (Dim_X, Dim_Y, and Dim_Z) solely determine the position of the marker $O'X'Y'Z'$. These three dimensions are in the conceptual level. They could be used as the driven dimensions (primary dimensions).

In the base level, every detail feature could be modeled relative to these datum frames. An example is a support used in the 540 caliper fixtures as shown in Figure 6.9. It has two fixturing elements - locating X1 and locating Y1. Figure 6.8 shows the modeling process of Loc_Y1.

After $\Delta r' = (\Delta x', \Delta y', \Delta z')^T$ has been calculated, it could be mapped into the primary dimensions (Dim_X, Dim_Y, Dim_Z). A variation of the support shown in Figure 6.9 is shown in Figure 6.10.

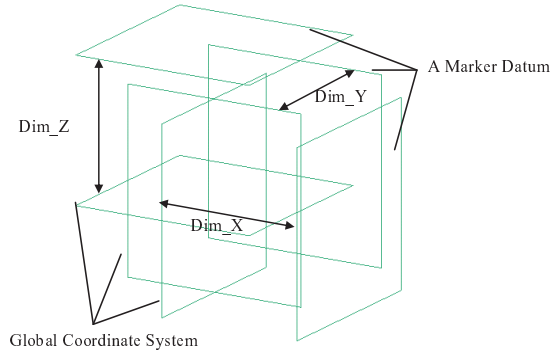


Figure 6.7: Datum frames.

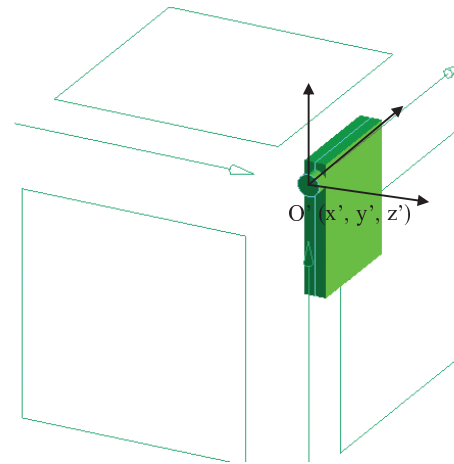


Figure 6.8: Base level model of A support.

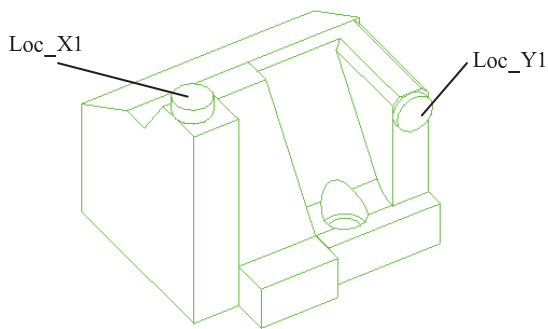


Figure 6.9: A support used in 540 fixtures.

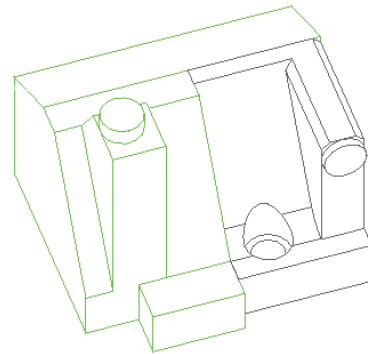


Figure 6.10: A variation of a support.

6.3.3 Fixture Variation Results

The details of fixture component variation could be found from Figures 6.9 and 6.10.

6.4 Summary

In this chapter a special attention has been focused on the variation fixture design. The models, variation processes, data structures, and algorithms have been discussed

in depth. From workpiece model and workpiece variation model to fixture assembly model and fixture variation model, every aspect for the variation fixture design has been addressed. A case study that demonstrates the support variation has been presented.

Chapter 7

Conclusions

7.1 Conclusions

Around reusability and variation for fixtures in mass production, the dissertation introduced the concept of design windows to tackle fixture reusability - make each fixturing unit on the fixture base the capabilities to rapidly adjust to a given workpiece. Then based on the fact that (1) all new workpiece models are mostly like to be within a design window, and (2) past practices proved to have been improved and should be reused, two fixture base synthesis algorithms were presented and studied with engineering cases. Fixture variation was also discussed to cope with rapid fixture generations.

In summary, we conclude,

- The implementation of design window is an efficient way to cope with the fixture reusability in mass production. By adjusting the position of each fixturing unit, it not only can be rapidly accommodated to the given workpiece so as to save time and cost, it also improve performance in terms of manufacturing tool consistence and quality control.

- As a widely used non-linear optimization algorithm, genetic algorithm proved to be a good fit for fixture base synthesis where often there are a large number of optimization variables and very complex engineering optimization functions. However it proved to be exponential in terms of computation complexity.
- An improved pair-wise optimization algorithm is a good enhancement over the GA algorithm. It borrows algorithms in other domain and significantly decrease number of optimization variables so as to decrease computation complexity exponentially and in the mean time it offers good performance in terms of fitness convergence and time cost.
- Through workpiece and fixture models, support variation is a fast way to generate new fixture components for a given workpiece.

7.2 Contributions

This dissertation has five major contributions:

- Proposed the concept of design window to cope with fixture reusability in mass production.
- Implemented genetic algorithm for fixture base synthesis.
- Proposed an enhanced algorithm to decrease computation complexity.
- Proposed fixture models to implement fixture variation.
- Implemented and integrated into an engineering tool to tackle with different aspects of fixtures in mass production.

7.3 Future Work

Fixture engineering is a complicated practice. Beside fixture design, a variety of aspects need to be considered. For example,

- *Stability*: adequate clamping force is required while it did not cause workpiece deformation and stresses.
- *Locating errors*: locating errors are the major source of machining errors and inaccuracies.
- *Error proofing*: in mass production, it is important to prevent incorrect workpiece loading.
- *Clamping sequence*: proper clamping sequence is essential to have appropriate clamping while not causing stresses within workpiece.
- *Chip shedding*: bad chip shedding can scratch machined surface so as to affect machining quality.
- *Ergonomic issues*: manufacturing is a human involved process, issues such as load and unload ability can affect how difficult an operator can operate on it.

Bibliography

- [An et al., 1999] An, Z.; Huang, S.; Rong, Y.; and Jayaram, S. (1999). “Development of Automated Fixture Design Systems with Predefined Fixture Component Types: Part 1, Basic Design”. *International Journal of Flexible Automation and Integrated Manufacturing*, 7(1), pp. 15–25.
- [An et al., 2000] An, Z.; Huang, S.; Rong, Y.; and Jayaram, S. (2000). “Development of Automated Fixture Design Systems with Predefined Fixture Component Types: Part 2, Detail Design”. *International Journal of Advanced Manufacturing Technology*, 16(2), pp. 115–126.
- [Bai, 1995] Bai, Y. (1995). “Computer Aided Modular Fixture Configuration Design System”. Master’s thesis, University of Illinois at Carbondale.
- [Brost and Goldberg, 1996] Brost, R. C. and Goldberg, K. Y. (1996). “A complete algorithm for designing planar fixtures using modular components”. *IEEE Transactions Journal of Robots Automation*, 12(1), pp. 31–46.
- [Brost and Peters, 1996] Brost, R. C. and Peters, R. R. (1996). “Automated design of 3 – D fixtures and assembly pallets”. In *Proceedings of IEEE, International Conference on Robotics and Automation, 1996*, pages 110–118.
- [Campbell, 1994] Campbell, P. (1994). *Basic Fixture Design*. Industrial Press Inc.
- [Chang, 1998] Chang, T. C. (1998). *Computer-Aided Manufacturing*. Prentice-Hall, Inc.
- [Chou and Barash, 1990] Chou, Y. C. and Barash, M. M. (1990). “Automated configuration of machining fixtures: object constraint reasoning”. *Manufacturing International*.
- [Choubey et al., 2005] Choubey, A. M.; Prakash, T. S.; and Tiwari, M. K. (2005). “Solving a fixture configuration design problem using genetic algorithm with learning automata approach”. *International Journal of Production Research*, 43(22), pp. 4721–4743.

- [Crosilla, 1999] Crosilla, F. (1999). “Procrustes analysis and geodetic science”. Technical report, Dept. of Geodesy and Geoinformatics, Univ. of Stuttgart, Stuttgart, Germany.
- [Davis, 1991] Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.
- [Deb, 1997] Deb, K. (1997). “Genetic Algorithm in Search and Optimization: The Technique and Applications”. In *Proceedings of International Workshop on Soft Computing and Intelligent Systems, Calcutta, India*, pages 58–87.
- [Deb, 1999] Deb, K. (1999). “An introduction to genetic algorithms”. *Sadhana Journal*, 24(4), pp. 293–315.
- [Deb, 2000] Deb, K. (2000). “An efficient constraint handling method for genetic algorithms”. *Computer Methods in Applied Mechanics and Engineering*, 186(6), pp. 311–338.
- [Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- [Gopalakrishnan et al., 2002] Gopalakrishnan, V.; Fedewa, D.; Mehrabi, M. G.; Kota, S.; and Orlandea, N. (2002). “Parallel Structures and Their Applications in Reconfigurable Machining Systems”. *Journal of Manufacturing Science and Engineering*, 124(2), pp. 483–486.
- [Graham, 1972] Graham, R. (1972). “An Efficient Algorithm for Determining the Convex Hull of a Finite Point Set”. *Info. Proc. Letters*, 1, pp. 132–133.
- [Herrmann and Cooper, 2004] Herrmann, J. W. and Cooper, J. t. (2004). “New Directions in Design for Manufacturing”. In *Proceedings of DETC04 ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference September 28–October 2, 2004, Salt Lake City, Utah USA*, pages 1–9.
- [Kaya, 2006] Kaya, N. (2006). “Machining fixture locating and clamping position optimization using genetic algorithms”. *Computers in Industry*, 57(6), pp. 112–120.
- [Kong and Ceglarek, 2003] Kong, Z. and Ceglarek, D. (2003). “Rapid Deployment of Reconfigurable Assembly Fixtures Using Workspace Synthesis and Visibility Analysis”. *Annals of the CIRP*, 52(1), pp. 13–16.

- [Kong and Ceglarek, 2006] Kong, Z. and Ceglarek, D. (2006). “Fixture Workspace Synthesis for Reconfigurable Assembly Using Procrustes-Based Pairwise Configuration Optimization”. *Journal of Manufacturing Systems*, 25(1), pp. 25–38.
- [Kumar et al., 1992] Kumar, A. S.; Nee, A. Y.; and Prombanpong, S. (1992). “Expert fixture-design system for an automated manufacturing environment”. *Computer-Aided Design*, 24(6), pp. 316–326.
- [Kumar and Paulraj, 2010] Kumar, K. and Paulraj, S. G. (2010). “Genetic algorithm based deformation control and clamping force optimisation of workpiece fixture system”. *International Journal of Production Research*, 48(6), pp. 1210–1218.
- [Lee, 1995] Lee, J. (1995). *Set-based Design Systems for Stampings and Flexible Fixture Workspaces*. PhD thesis, The University of Michigan.
- [Lee et al., 1999] Lee, J.; Hu, S. J.; and Ward, A. C. (1999). “Workspace Synthesis for Flexible Fixturing of Stampings”. *Transactions of ASME, Journal of Manufacturing Science and Engineering*, 121(8), pp. 478–484.
- [Liao, 2000] Liao, Y. G. (2000). “A genetic algorithm-based fixture locating positions and clamping schemes optimization”. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 217(8), pp. 1075–1083.
- [Lin et al., 1999] Lin, C.-J.; Michelena, N.; and Saigal, R. (1999). “Topological fixture synthesis using semidenite programming”. In *the Third World Congress of Structural and Multidisciplinary Optimization (WCSMO-3), May 17-21 1999*, pages 221–226.
- [Lin and Huang, 1997] Lin, Z.-C. and Huang, J.-C. (1997). “The application of the simulated annealing method and heuristic algorithms in the selection of fixture elements”. *The International Journal of Advanced Manufacturing Technology*, 13(3), pp. 191–205.
- [Liu and Strong, 2002] Liu, J. and Strong, D. R. (2002). “Machining fixture verification for linear fixture systems”. *International Journal of Production Research*, 40(14), pp. 3441–3459.
- [Liu et al., 2008] Liu, T.; Chen, C.-H.; and Chou, J.-H. (2008). “Intelligent design of the automobile fixtures with genetic algorithms”. *International Journal of Innovative Computing, Information and Control*, 4(10), pp. 2533–2543.

- [Markus, 1984] Markus, A. (1984). “Fixture design using PROLOG: an expert system”. *Robotics and CIMS*, 1(2), pp. 167–172.
- [Megiddo, 1983] Megiddo, N. (1983). “Linear-Time Algorithms for Linear Programming in R^3 and Related Problems”. *SIAM J. Computation*, 12, pp. 759–776.
- [Meter et al., 2001a] Meter, E. C.; De, S.; and Philip, S. (2001a). “Part location algorithms for an intelligent fixturing system Part 1: system description and algorithm development”. *Journal of Manufacturing Systems*, 20(2), pp. 124–134.
- [Meter et al., 2001b] Meter, E. C.; De, S.; and Philip, S. (2001b). “Part location algorithms for an Intelligent Fixturing System Part 2: algorithm testing and evaluation”. *Journal of Manufacturing Systems*, 20(2), pp. 135–148.
- [Nee and Kumar, 1991] Nee, Y. C. and Kumar, A. S. (1991). “A framework for an object/rule-based automated fixture design system”. *Annals of the CIRP*, 24(6), pp. 147–151.
- [O’Rourke, 1995] O’Rourke, J. (1995). *Computational Geometry in C*. Cambridge University Press.
- [Penev and Requicha, 1994] Penev, K. and Requicha, A. (1994). “Fixture foolproofing for polygonal parts”. In *Proceedings of IEEE, International Symposium on Assembly and Task Planning, Pittsburgh, PA May, 1995*, pages 12–19.
- [Pham and Lazaro, 1990] Pham, D. T. and Lazaro, A. (1990). “AutoFix C an expert CAD system for jigs and fixtures”. *International Journal of Machine Tools and Manufacture*, 30(3), pp. 403–411.
- [Preparata and Shamos, 1988] Preparata, F. P. and Shamos, M. I. (1988). *Computational Geometry*. Springer-Verlag.
- [Qin and Zhang, 2007] Qin, G. and Zhang, W. (2007). “Modeling and analysis of workpiece stability based on the linear programming method”. *The International Journal of Advanced Manufacturing Technology*, 32(1), pp. 78–91.
- [Rademacher and Toeplitz, 1957] Rademacher, H. and Toeplitz, O. (1957). *The Enjoyment of Mathematics: Selections from Mathematics for the Amateur*. Princeton University Press.
- [Rong, 1999] Rong, Y. (1999). *Computer-Aided Fixture Design*. Marcel Dekker, Inc.
- [Rong, 2005] Rong, Y. (2005). *Advanced Computer-Aided Fixture Design*. Elsevier Academic Press.

- [Rong and Bai, 1996] Rong, Y. and Bai, Y. (1996). “Machining accuracy analysis for computer aided fixture design verification”. *Transactions of ASME, Journal of Manufacturing Science and Engineering*, 118(3), pp. 289–300.
- [Rong et al., 1993] Rong, Y.; Zhu, J.; and Li, S. (1993). “Fixturing feature analysis for computer-aided fixture design”. In *Proceedings of ASME, International Mechanical Engineering Congress and Exposition, PED Vol. 64, 28 November – 3 December, 1993. New Orleans, LA*, pages 267–271.
- [Rong and Bai, 1997] Rong, Y. M. and Bai, Y. (1997). “Automated generation of fixture configuration design”. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 119(5), pp. 208–220.
- [Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence A Modern Approach, 2nd Edition*. Prentice Hall.
- [Shamos, 1975] Shamos, M. I. (1975). “Geometric Complexity”. In *Conference Record of Seventh Annual ACM Symposium on Theory of Computation, 5-7 May 1975, Albuquerque, New Mexico, USA*, pages 224–233.
- [Smith, 1998] Smith, P. L. (1998). “American Machinist - A flexible fixturing future”.
- [Trappy et al., 1993] Trappy, J. C.; Su, C. S.; and Huang, S. H. (1993). “Methodology for location and orientation of modular fixtures”. In *ASME WAM, New Orleans, LA*, pages 333–342.
- [Walczyk and Longtin, 2000] Walczyk, D. F. and Longtin, R. S. (2000). “Fixturing of compliant parts using a matrix of reconfigurable pins”. *Journal of manufacturing science and engineering*, 122(4), pp. 766–772.
- [Wang, 2000] Wang, M. Y. (2000). “Optimum 3D Fixture Layout Design”. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation June 28 – July 2, 2000. Hefei, P. R. China*, pages 79–84.
- [Whybrew and Ngoi, 1990] Whybrew, K. and Ngoi, K. A. (1990). “Computer aided design of modular fixture assembly”. *International Journal of Advanced Manufacturing Technology*, 7(3), pp. 267–276.
- [Wolter and Trinkle, 1994] Wolter, J. D. and Trinkle, J. C. (1994). “Automatic selection of fixture points for frictionless assemblies”. In *Robotics and Automation, 1994. Proceedings, 1994 IEEE International Conference on. 08 May 1994 - 13 May 1994, San Diego, CA , USA*, pages 528–534.

- [Wu et al., 1998a] Wu, Y.; Rong, Y.; Ma, W.; and LeClair, S. R. (1998a). “Automated modular fixture planning: Accuracy, clamping, and accessibility analyses”. *Robotics and Computer-Integrated Manufacturing*, 14(1), pp. 17–26.
- [Wu et al., 1998b] Wu, Y.; Rong, Y.; Ma, W.; and LeClair, S. R. (1998b). “Automated modular fixture planning: Geometric analysis”. *Robotics and Computer-Integrated Manufacturing*, 14(1), pp. 1–15.
- [Wu, 1996] Wu, Y. F. (1996). “Geometry generation for computer aided dedicated fixture design”. Master’s thesis, University of Illinois at Carbondale.
- [Zhu and Ding, 2007] Zhu, X. and Ding, H. (2007). “An Efficient Algorithm for Grasp Synthesis and Fixture Layout Design in Discrete Domain”. *Robotics, IEEE Transactions on*, 23(1), pp. 157–163.
- [Zhuang et al., 1994] Zhuang, Y.; Goldberg, K. Y.; and Wong, Y. (1994). “On the existence of modular fixtures”. In *Proceedings of IEEE, International Symposium on Assembly and Task Planning, May, 1994*, pages 12–19.