Extended and Unscented Kalman Smoothing for Re-linearization of Nonlinear Problems with Applications

by

Matthew S. Lowe

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
Electrical and Computer Engineering
by

_____

April 30, 2015

APPROVED:

_____
Professor David Cyganski, Major Advisor

_____
Professor Arthur Heinricher

_____
Professor Taskin Padir

Abstract

The Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Ensemble Kalman Filter (EnKF) are commonly implemented practical solutions for solving nonlinear state space estimation problems; all based on the linear state space estimator, the Kalman Filter. Often, the UKF and EnKF are cited as a superior methods to the EKF with respect to error-based performance criteria. The UKF in turn has the advantage over the EnKF of smaller computational complexity.

In practice however the UKF often fails to live up to this expectation, with performance which does not surpass the EKF and estimates which are not as robust as the EnKF. This work explores the geometry of alternative sigma point sets, which form the basis of the UKF, contributing several new sets along with novel methods used to generate them. In particular, completely novel systems of sigma points that preserve higher order statistical moments are found and evaluated. Additionally a new method for scaling and problem specific tuning of sigma point sets is introduced as well as a discussion of why this is necessary, and a new way of thinking about UKF systems in relation to the other two Kalman Filter methods. An Iterated UKF method is also introduced, similar to the smoothing iterates developed previously for the EKF. The performance of all of these methods is demonstrated using problem exemplars with the improvement of the contributed methods highlighted.

Acknowledgements

My family: They have always worked to support and encourage me throughout my life. Most of it, as I imagine, was easy. But even I must admit that their efforts in the last couple must have been heroic. The true foundation of my work has always rested in their stability and love.

My friends and colleagues: They are always a source inspiration and help.

My advisor and committee: My thanks to Professor Cyganski, whose clever thinking keeps both on track and on my toes, I could not have asked for a better advisor. My thanks also to Professors Heinricher and Padir who took the time to enrich this work.

Contents

List of Figures **v**

List of Tables **viii**

1 Introduction **1**

2 The State/Dual Kalman Filter Representation **5**
2.1 Problem Statement 5
2.2 Convex Optimization and the Dual State Solution 7
2.3 Properties of the Symplectic Update Γ 15

3 Smoothing Solutions **17**
3.1 Weather-vane Problem 17
3.2 Reversing the Solution 22
3.3 Numerical Issues 23
3.4 Constraint Based Solution 28

4 Smoothed Solution Convergence in the Blind Tricyclist Problem **35**
4.1 The Blind Tricyclist Problem 36
4.2 Batch Least Squares 44
4.3 Extended Kalman Filter and Smoothers 48
4.3.1 Iterated Smoother 52
4.3.2 Other Smoothers 59

5 Sigma Point / Unscented Methods Geometry **63**
5.1 The Unscented Transform 63
5.2 Orthogonal Geometry of Sigma Point Sets 65
5.3 Lie Algebra of Higher Order Sigma Point Moments 68
5.4 Derived Sigma Point Sets 70
5.4.1 Simplex Set (Simp) 70
5.4.2 One Dimensional Sets 74
5.4.3 Mixed Moment Sets 76
5.5 Results of Using Different Sets 81
5.5.1 Example of Polynomial Functions 81
5.5.2 Fading Channel Example 84

5.5.3 Angle Tracking Example 87

6 Sigma Point / Unscented Smoothing **90**
6.1 Sigma Point Scaling 90
6.1.1 Wrapped Measurement Model 90
6.1.2 Statistical Regions of Approximation and Scaling 100
6.1.3 Wrapping Example with Scaling 106
6.2 Smoothing Equations 106
6.2.1 The Iterative UKF Smoother 109

7 Sigma Point/UKF Application to the Blind Tricyclist Problem **113**
7.1 Scaling in the Blind Tricyclist Problem 113
7.2 New Results in the Blind Tricyclist Problem 121
7.3 Alternate Scenario 124
7.4 Conclusions on the Blind Tricyclist Problem 124

8 Conclusions **127**

A Additional SP Lemma **134**

Bibliography **136**

List of Figures

3.1.1 Phase Plane Plot of Weather-vane Problem . . . 18
3.1.2 Kalman Filter with only Initial Uncertainty . . . 20
3.1.3 Kalman Filter with No Initial Uncertainty . . . 21
3.1.4 Full Kalman Filter Solution . . . 22
3.2.1 Reversed/Smoothed Solution Path . . . 24
3.3.1 Floating Point Smoothed Path . . . 25
3.3.2 Random Error with Perfect Arithmetic . . . 29
3.3.3 Constrained Error with Perfect Arithmetic . . . 30
3.4.1 Constraint Based Smooth . . . 32

4.1.1 Merry-Go-Round State . . . 36
4.1.2 Tricycle State . . . 37
4.1.3 Tricycle Movement . . . 37
4.1.4 Tricycle's Movement . . . 39
4.1.5 Tricycle Bearing Measurement . . . 40
4.1.6 Expected Root Mean Squared Error in 2D Location . . . 42
4.1.7 KF with Perfect Linearizations Example . . . 42
4.1.8 Other Examples of KF with Perfect Linearizations . . . 43
4.1.9 Perfect Linearization KF's Error . . . 44
4.2.1 BLSF (Simple Implementation) . . . 45
4.2.2 BLSF with Relax $\alpha = 0.15$ . . . 46
4.2.3 Other Examples of BLSF with Relax $\alpha = 0.15$ . . . 47
4.3.1 Example EKF Track . . . 49
4.3.2 Other Examples of EKF Performance . . . 50
4.3.3 EKF Error Compare . . . 51
4.3.4 Iterated Smoother (1 Iterations) . . . 53
4.3.5 Iterated Smoother (1 Iterations) Other Examples . . . 54
4.3.6 Iterated Smoother (1 Iterations) Error . . . 55
4.3.7 Iterated Smoother (18-19 Iterations) . . . 56
4.3.8 Iterated Smoother (18-19 Iterations) Other Examples . . . 57
4.3.9 Iterated Smoother (19 Iterations) Error . . . 58
4.3.10 Iterated Perfect KF (19 Iterations) Error . . . 58
4.3.11 BSEKF (19 Iterations) Error . . . 60
4.3.12 Perfect KF (with and without w) Error . . . 60

4.3.13 Estimates of w from the Perfect KF . . . 61
4.3.14 Estimates of v . . . 62

5.5.1 Probability Density Function for χ^2 and $N(1,2)$ . . . 82
5.5.2 Fading Example Median and 75% x_0 Error Plots . . . 85
5.5.3 Fading Example (with abs) Median and 75% x_0 Error Plots . . . 86
5.5.4 Filter Performance on Example Related to Rotation Tracking . . . 88
5.5.5 Filter Performance on Example Related to Rotation Tracking . . . 89

6.1.1 Filter Comparison on Periodic Measurements . . . 92
6.1.2 Periodic Measurement with Filters' Estimated Mean and Covariance . . . 94
6.1.3 50th and 75th Quantile Plots of State Errors for Wrapping Example . . . 95
6.1.4 90th Quantile Plot of State Error for Wrapping Example . . . 95
6.1.5 Performance of Filters with Negative Weights . . . 96
6.1.6 Periodic Measurement with Filters' Estimated Mean and Covariance . . . 98
6.1.7 Quantile Error Plot for $\sigma = 0.5$ . . . 100
6.1.8 Quantile Error Plot for $\sigma = 2$ . . . 101
6.1.9 Quantile Error Plot for $\sigma = 8$ . . . 101
6.1.10 Quantile Error Plot for $\sigma = 8$ as α Scaling Factor (Indicated in parenthetic values on vertical axis) . . . 102
6.1.11 Quantile Error Plot for $\sigma = 8$ as alternative α varies for O5 set . . . 105
6.1.12 Quantile Error Plot for $\sigma = 8$ as alternative α varies for O3/Simp set . . . 105
6.1.13 50th and 75th Quantile Plots of State Errors for Wrapping Example with Scaling . . . 107
6.1.14 90th Quantile Plot of State Error for Wrapping Example with Scaling . . . 107
6.2.1 Quantile Error Plot for $\sigma = 8$ of Iterated EKF . . . 108
6.2.2 Quantile Error Plot for $\sigma = 8$ of Iterated UKF . . . 109
6.2.3 Quantile Error Plot for $\sigma = 8$ of Iterated UKF with Scaling . . . 110

7.0.1 UKF with O5 Example Tracks (Truth in black, UKF in dark green, Smoothed UKF green dashed) . . . 114
7.0.2 UKF O5 Error Compare . . . 115
7.1.1 Simple Scenarios for Parameter Tuning . . . 116
7.1.2 End State Error for Scenario 1 (Point 1 Straight Movement) . . . 117
7.1.3 End State Error for Scenario 2 (Point 1 Curved Movement) . . . 117
7.1.4 End State Error for Scenario 3 (Point 2 Straight Movement) . . . 118
7.1.5 End State Error for Scenario 4 (Point 2 Curved Movement) . . . 118
7.1.6 End State Error for Scenario 5 (Point 3 Straight Movement) . . . 119
7.1.7 End State Error for Scenario 6 (Point 3 Curved Movement) . . . 119
7.1.8 End State Error for Scenario 1 (Point 1 Straight Movement) . . . 119
7.1.9 End State Error for Scenario 5 (Point 3 Straight Movement) . . . 120
7.2.1 UKF with O7 (0.4 scaling) Example Tracks . . . 122
7.2.2 UKF O7 (0.4 Scaling) Error Compare . . . 123
7.2.3 UKF O7 (0.4 Scaling) with Smoothing Error Compare . . . 123
7.3.1 Alternate Scenario EKF Example . . . 124
7.3.2 EKF Error Statistics for Alternate Scenario . . . 125
7.3.3 UKF Error Statistics for Alternate Scenario . . . 125

8.0.1 Iterative UKF Example Realization 1 131
8.0.2 Iterative UKF Example Realization 2 132

List of Tables

3.1 Numerical Comparisons for the Weather-vane Problem 34

5.1 Sigma Point Set Properties 70
5.2 Results for x_0^2 83
5.3 Results for x_0^4 83
5.4 Results for $x_0 x_1$ 84

6.1 Sigma Point Sets' Sampling Distance 106

Chapter 1

Introduction

> *They were two lovely choices. One of them meant giving up every chance of a decent life forever...and the other one scared me out of my mind.*
>
> - Frederik Pohl, *Gateway*

As part of the introduction I would like to provide a brief synopsis of each chapter, both as a way of quickly navigating the core movement of the work and as a way of anticipating some of the its subtleties. Before we can begin this, per-chapter review, we should first elaborate on the topic of the work as so many necessary words would not fit in the title. This work started in my study of iterative techniques for improving the performance of the Extended Kalman Filter (EKF), described in [13, 14, 2], in challenging problems which led me to the results presented in [10], wherein the Blind Tricyclist Problem is introduced and the iterative Backwards Smoothing EKF (BSEKF) [9] is shown as one of the best of the cutting edge methods as compared to the EKF, UKF, and even Partial Filtering. The failure of the Unscented Kalman Filter (UKF) [4] to outperform the EKF, however, was so surprising to me as to warrant a more careful study which has led to my contributions to this method. I will, through this work, present these new methods and, more importantly, a better context to understand them within. All of the techniques shown in this work are, by nature of their Kalman Filter base, linear estimators. This fact is both their strength and weakness. On the one hand they are often sub-optimal for the nonlinear problems that they are intended to solve but on the other hand their efficient computation makes them useful. In this area, nonlinear tracking, the goal is often to find an efficient, simple solution which solves the problem well enough even though not optimally and the EKF has proven itself capable in this regard, making it the first estimator often tried on a new problem. There are, however, problems where the EKF does not quite meet the desired level of performance and in these cases the UKF is thought to be the next natural step in

the complexity/performance trade off. In this work we will be looking at how the UKF can better fill this need. It should be noted that all of the estimators discussed in this work are, at their core, linear estimators meaning that their performance is ultimately going to be limited and, as I shall show, the choices among these several implementations necessarily represent a trade off between accuracy and robustness.

Chapter 2 The State/Dual Kalman Filter Representation Before we can begin this discussion I need to introduce the core Kalman Filter (KF). The Kalman Filter is a well established estimator for the linear state space tracking problem which is both, as a property of the problem, a Maximal Likelihood Estimator (MLE) and a Minimal Mean Squared Error (MMSE) estimator. Its real strength, however, lies in its recursive construction which makes it well suited for real world implementations, where its estimates can easily be corrected as new data is collected without re-calculating all the previous data. This chapter presents its derivation from the point of view of convex optimization. This nonstandard approach will provide a refresher to the topic and show how the filter's recursion is built out of an efficient blocking of the original problem. I feel its great benefit is making a clear distinction between the track of recursive estimates of the system's state and the final estimate of the system's track. It also makes clear that, although the problem is to estimate the entire track of states, the entire problem boils down to finding an optimal estimate of the state for any one time instance. This is something we might expect from the fact that the recursion exists, as it is only estimating the state at one point in time, but in this development this fact is given full form. This form, the Hamiltonian/Symplectic matrix development, is often discussed in its relationship to just the covariance matrix either in terms of steady state response as in [11] or for one step covariance tracking as in [8]. In the development here, however, we explore its practical application as an actual optimal solution propagator.

Chapter 3 Smoothing Solutions Here we are going to make the results of the last chapter concrete by showing the KF in an implementation, breaking apart and showing the separate contributions of the various factors. The main thrust of this chapter will be to provide another recursive estimator which can construct the entire estimated track from an end state KF estimate. This reveals one of the main differences between the theory and practical implementation. Although the original problem can be seen as only needing to estimate the state optimally for one instance in time from which all intermediate state solutions may then be derived, in an implementation this does not work because the sequence of states, necessarily, become uncorrelated across time, causing numerical failure. Again, because of the alternative formulation being presented of the original KF, this leads to some forms which are slightly different from the standard works, and we explore some potentially computational cheaper methods of formulating a

smoother, which can result in more efficient processing.

Chapter 4 Smoothed Solution Convergence in the Blind Tricyclist Problem Now that we have a strong foundation in the linear theory of the KF we need to move on to the nonlinear theory of the EKF and other generalizations including the Iterated Smoother (IS) and Backwards Smoothing Extended Kalman Filter (BSEKF). We will do this by describing the extensions of the theory in parallel with demonstrating its results on a synthetic nonlinear problem exemplar, The Blind Tricyclist Problem introduced in [10]. The advantage of this problem is that, because it is entirely synthetic, it is accurately described in both its noise and function models. We do not need to worry about noise processes which are difficult to describe as Gaussian or propagation/observation functions which are incorrectly, or impossibly, defined. One of the biggest differences in this treatment from others is the demonstration of filter performance in terms of error quantiles instead of simply mean squared error. This difference will be important in contrasting the methods introduced later.

Chapter 5 Sigma Point/Unscented Methods Geometry Given the failure of the standard linearization techniques to achieve theoretically possible results we are going to examine one of the most common alternatives to the simple EKF approach, the Unscented Kalman Filter (UKF) which is based on an estimation technique utilizing sigma points. The treatment of sigma points found here is substantially different from the standard discussion in the literature and focuses on the geometry required to generate new sets of sigma points and its relationship to their higher order moments. Previous work in this area, [15], has demonstrated the ability of 1D sets, defined by a symmetric parameterization, to be constructed to estimate higher order moments. Other works, such as [7] attempt to generate sets which take into account higher order moments, they do not however attempt to match the moments exactly and do not take into account mixed moments. In this work however I remove the restriction to symmetric sets, advance strategies for expanding sets generated to have higher order moments to higher dimensions, provide an accompanying discussion of mixed moments and sets to meet them, and demonstrate their superior performance. Additionally although in a previous work [3] a set which minimized 3rd order moment errors has been discussed, due to a sampling distance issue, the authors later superseded it in [4] with a new set whose points all lie on a sphere. In this work I present a parametrization of sets generated from an optimality condition on 3rd order moments which contains the spherical set described in [4] with a new set whose points all lie on a sphere. In this work I present a parametrization of sets generated from an optimality condition on 3rd order moments which contains the spherical set described in [4].

Chapter 6 Sigma Point/Unscented Smoothing Just as Chapter 3 built on Chapter 2 by confronting difficulties in the original theory, this chapter builds on the previous by looking at the limitations of the last. The primary way I achieve this is by examining the difference between this new method of approximating problems, the UKF, and the more standard EKF using the statistical analysis of the estimators' errors. The results presented here are surprising, demonstrating the trade off I mentioned earlier, and I believe will provide insight into the comparative strengths and weaknesses of the two methods. I will also present a method for achieving results which are in the range between pure UKF and EKF the importance of which is also demonstrated. Additionally I present a method for iterating the UKF, just as was done for the EKF in Chapter 4, giving us a complete continuation of the EKF theory into the UKF. There has been some, limited, exploration in the area of constructing iterative methods for the UKF by [6], which considered only the observation step alone. The methods presented in this work advance this idea by including smoothing, allowing iterations on an entire track, and consider re-sampling the transform after each iteration.

Chapter 7 Sigma Point/Unscented Applications in the Blind Tricyclist Problem This chapter employs The Blind Tricylist problem to demonstrate the gains achieved with the new methods introduced in the previous chapters. First this demonstration re-affirms that the standard UKF fails to outperform the EKF, a result I have motivated in the previous chapter. And secondly, by using a simple tuning strategy available to our new methods I demonstrate their potential superior performance for this same problem. Additionally I introduce and again demonstrate superior performance in an alternative problem scenario, arguing for the general performance increase available to this new filter with the tuning strategies employed.

Chapter 2

The State/Dual Kalman Filter Representation

> *"Yet is common opinion the fool, not I," he said. "He that imagineth after his labours to attain unto lasting joy, as well may he beat water in a mortar. Is there not in the wild benefit of nature instances enow to laugh this folly out of fashion? A fable of great men that arise and conquer the nations: Day goeth up against the tyrant night. How delicate a spirit is she, how like a fawn she footeth it upon the mountains: pale pitiful light matched with the primeval dark. But every sweet hovers in her battalions; and every heavenly influence: coolth of the wayward little winds of morning, flowers awakening, birds a-carol, dews a-sparkle on the fine-drawn webs the tiny spinners hang from fern-frond to thorn, from thorn to wet dainty leaf of the silver birch; the young day laughing in her strength, wild with her own beauty; fire and life and every scent and colour born anew to triumph over chaos and slow darkness and the kinless night."*
>
> -E. R. Eddison, *The Worm Ouroboros*

This chapter will derive the Kalman Filter, a rather common 'filter', from a slightly different perspective than most treatments. The goal here is to provide a slightly different perspective from which it will be possible to gain new insight. The methods discussed result in a set of equations that are often called the Symplectic Representation of the Kalman Filter.

2.1 Problem Statement

We can begin with a time evolving state space system defined in Equation 2.1.1.

$$x_k = F_k x_{k-1} + u_k + L_k w_k \text{ for index } k = 1..\eta \tag{2.1.1}$$

Where

x_k is the *state* of a system with an initial condition determined by the random variable $x_0 \sim N(\hat{x}_0, P_0)$ where both $\hat{x}_0$ and P_0 are known, $P_0 \in \mathbb{R}^{\ell \times \ell}$

F_k is the known *state transition matrix*, $F_k \in \mathbb{R}^{\ell \times \ell}$ and is invertible

u_k is the *forcing function* and is known $u_k \in \mathbb{R}^{\ell}$

w_k is the *process noise* and is distributed as a zero mean normal random variable $w_k \sim N(0, Q_k)$ where $Q_k \in \mathbb{R}^{m_k \times m_k}$ is known, symmetric, and invertible

L_k is the known *process noise transform matrix*, $L_k \in \mathbb{R}^{\ell \times m_k}$

There are other similar representations of this problem and I will briefly try to illuminate why I have chosen this representation. First u_k is sometimes represented as $G_k u_k$, usually when the text in question is exploring the problem from the point of view of controls because it allows a succinct way of representing the limitations of a control input u_k. Here, however, I will be only examining the estimation of such a system and not its control, so this restriction is not useful, combined with the fact that many things not typically considered 'control' inputs are also going to be lumped into this vector. The first of these useful 'lumpings' can be seen in our restriction on w_k to be zero mean. This is no restriction at all when one considers that any mean $\hat{w}_k$ can simply be pulled out and made part of $\tilde{u}_k = u_k + L_k \hat{w}_k$. Additionally the restriction that Q_k be symmetric *invertible* instead of symmetric *positive semi-definite* is eased by the existence of L_k which allows the random variable $L_k w_k$ to be a normal random variable with only a positive semi-definite covariance matrix. In fact we could make one random variable $v \sim N(u_k, L_k Q_k L_k^{\mathrm{T}})$ which absorbs all of these variables. These notations are picked for their impact in the later, nonlinear system, discussions giving room to distinguish effects according to root causes.

Our goal for this problem is to be able to estimate the state of the system at the various times k. To this end we have one additional set of information. We make measurements of the state in the form of Equation 2.1.2.

$$y_k = H_k x_k + J_k v_k \text{ for index } k = 1..\eta \tag{2.1.2}$$

Where

y_k is the *observation*, $y_k \in \mathbb{R}^{b_k}$ and is known

H_k is the *observation transform matrix*, $H_k \in \mathbb{R}^{b_k \times \ell}$ and is known

J_k is the known *observation noise transform*, $J_k \in \mathbb{R}^{b_k \times a_k}$

v_k is the *observation noise* and is distributed as a zero mean normal variable $v_k \sim N(0, R_k)$ where $R_k \in \mathbb{R}^{a_k \times a_k}$ is known, also $J_k R_k J_k^{\mathrm{T}}$ is symmetric and invertible

It is uncommon to include the matrix J which appears wholly unnecessary though, again, its utility will become more apparent later. The restrictions that v_k be zero mean is not actually restrictive, as any mean, $\hat{v}_k$, can simply be removed from the observation, $\tilde{y}_k = y_k - J_k \hat{v}_k$.

Between these two equations, 2.1.1 and 2.1.2 we have a set of constraints, and implicit is that we want to estimate the quantities, x_k. Our goal is to construct the Maximal Likelihood Estimator (MLE)[1] for (x, w, v).

Formally then, our problem is to find values of (x, w, v) which maximize the probability of the observed events, that is to maximize

$$\mathfrak{p}(x_0, w, v) = \frac{e^{\frac{-1}{2}(x_0 - \hat{x}_0)^{\mathrm{T}} P_0^{-1} (x_0 - \hat{x}_0)}}{\sqrt{2\pi}^{\ell} \sqrt{|P_0|}} \cdot \prod_{k=1}^{\eta} \frac{e^{\frac{-1}{2} w_k^{\mathrm{T}} Q_k^{-1} w_k}}{\sqrt{2\pi}^{m_k} \sqrt{|Q_k|}} \cdot \prod_{k=1}^{\eta} \frac{e^{\frac{-1}{2} v_k^{\mathrm{T}} R_k^{-1} v_k}}{\sqrt{2\pi}^{d_k} \sqrt{|R_k|}}, \tag{2.1.3}$$

while also meeting the constraints found in the previous two equations, 2.1.1 and 2.1.2.

2.2 Convex Optimization and the Dual State Solution

In order to solve this problem we will be applying techniques from Convex Optimization[2]. The first step will be to simplify our probability function, shown in Equation 2.1.3, by taking its logarithm, where $\|z\|_{\mathrm{M}}$ is the Mahalanobis Distance of the random variable $z \sim N(\hat{z}, S)$ is $\|z\|_{\mathrm{M}}^2 = (z - \hat{z})^{\mathrm{T}} S^{-1} (z - \hat{z})$.

$$\begin{aligned} \ln(\mathfrak{p}) = &- \ln\left(\sqrt{2\pi}^{\ell} \sqrt{|P_0|}\right) - \frac{1}{2} \|x_0\|_{\mathfrak{M}}^2 + \sum_{k=1}^{\ell} \left(- \ln\left(\sqrt{2\pi}^{m_k} \sqrt{|Q_k|}\right) - \frac{1}{2} \|w_k\|_{\mathfrak{M}}^2\right) \\ &+ \sum_{k=1}^{\ell} \left(- \ln\left(\sqrt{2\pi}^{d_k} \sqrt{|R_k|}\right) - \frac{1}{2} \|v_k\|_{\mathfrak{M}}^2\right) \end{aligned}$$

We can remove the constant terms to form the log likelihood and multiply by -1 to change it into a cost function, which we need to minimize, as shown in Equation 2.2.1.

[1] The systems are all linear with Normal random variable inputs so for the problem as stated the MLE is also a Minimal Mean Squared Error Estimate (MMSE), but as we move through the work we will be sacrificing this equivalence

[2] The forms and results of this section all derive from Section 28 of [12] except I have taken the liberty of applying a transpose.

$$\mathfrak{c}\left(x_0, n, w\right) = -\frac{1}{2}\left(\|x_0\|_{\mathrm{M}}^2 + \sum_{k=1}^{\eta}\|w_k\|_{\mathrm{M}}^2 + \sum_{k=1}^{\eta}\|n_k\|_{\mathrm{M}}^2\right) \tag{2.2.1}$$

The constraints, originally a set of independent equations from the propagation and observation equations 2.1.1, 2.1.2, can be reorganized to form a single, large, $\eta\ell + \prod b_k$ dimensional constraint function[3].

$$\begin{aligned}
&\Theta_{1,1} = -x_1 + F_1 x_0 + u_1 + L_1 w_1 && = 0\\
&\Theta_{1,2} = -x_2 + F_2 x_1 + u_2 + L_2 w_2 && = 0\\
&\vdots && \vdots\\
&\Theta_{1,k} = -x_k + F_k x_{k-1} + u_k + L_k w_k && = 0\\
&\Theta_{1,k+1} = -x_{k+1} + F_{k+1} x_k + u_{k+1} + L_{k+1} w_{k+1} && = 0\\
&\vdots && \vdots\\
\Theta\left(x, w, n\right) = \quad &\Theta_{1,\eta} = -x_\eta + F_\eta x_{\eta-1} + u_\eta + L_\eta w_\eta && = 0\\
&\Theta_{2,1} = -y_1 + H_1 x_1 + J_1 v_1 && = 0\\
&\Theta_{2,k} = -y_2 + H_2 x_2 + J_2 v_2 && = 0\\
&\vdots && \vdots\\
&\Theta_{2,k} = y_k + H_k x_k + J_k v_k && = 0\\
&\vdots && \vdots\\
&\Theta_{2,\eta} = y_\eta + H_\eta x_\eta + J_\eta v_\eta && = 0
\end{aligned} \tag{2.2.2}$$

Using Lagrangian Multipliers we know the solution must have the form,

$$\partial\mathfrak{c}^{\mathrm{T}} + \partial\Theta^{\mathrm{T}} \cdot \nu = 0.$$

where ν is the $\eta\ell + \prod b_k$ dimensional vector of Lagrangian Multipliers. Taking the derivatives, first of Equation 2.2.1, and second of Equation 2.2.2.

[3]This is going to be a linear constraint $A \begin{bmatrix} x \\ w \\ n \end{bmatrix} = 0$ where A is a huge but sparse matrix

$$\partial \mathfrak{c}^{\mathrm{T}} = \begin{bmatrix} P_0^{-\mathrm{T}} (x_0 - \hat{x}_0) \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -Q_1^{-\mathrm{T}} w_1 \\ -Q_2^{-\mathrm{T}} w_2 \\ \vdots \\ -Q_k^{-\mathrm{T}} w_k \\ \vdots \\ -Q_\eta^{-\mathrm{T}} w_\eta \\ -R_1^{-\mathrm{T}} v_1 \\ -R_2^{-\mathrm{T}} v_2 \\ \vdots \\ -R_{k-1}^{\mathrm{T}} v_{k-1} \\ -R_k^{\mathrm{T}} v_k \\ \vdots \\ R_\eta^{-\mathrm{T}} v_\eta \end{bmatrix} \tag{2.2.3}$$

$$\partial\Theta^{\mathrm{T}} = \begin{bmatrix}
F_1^{\mathrm{T}} & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\
-I & F_2^{\mathrm{T}} & 0 & & 0 & & 0 & 0 & H_1^{\mathrm{T}} & 0 & & 0 & 0 & & 0 & 0 \\
0 & -I & F_3 & & 0 & & 0 & 0 & 0 & H_2^{\mathrm{T}} & & 0 & 0 & & 0 & 0 \\
\vdots & & & \ddots & & & & & \vdots & & \ddots & & & \ddots & & \vdots \\
0 & 0 & 0 & & F_k^{\mathrm{T}} & & 0 & 0 & 0 & 0 & & H_{k-1}^{\mathrm{T}} & 0 & & 0 & 0 \\
0 & 0 & 0 & & -I & \ddots & 0 & 0 & 0 & 0 & & 0 & H_k^{\mathrm{T}} & & 0 & 0 \\
\vdots & & & & & & \ddots & & \vdots & & & & & \ddots & & \vdots \\
0 & 0 & 0 & & 0 & & -I & F_\eta^{\mathrm{T}} & 0 & 0 & & 0 & 0 & & H_{\eta-1}^{\mathrm{T}} & 0 \\
0 & 0 & 0 & \cdots & 0 & \cdots & 0 & -I & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & H_\eta^{\mathrm{T}} \\
L_1^{\mathrm{T}} & 0 & 0 & & 0 & & 0 & 0 & 0 & 0 & & 0 & 0 & & 0 & 0 \\
0 & L_2^{\mathrm{T}} & 0 & & 0 & & 0 & 0 & 0 & 0 & & 0 & 0 & & 0 & 0 \\
\vdots & & & \ddots & & & & & \vdots & & & & & & & \vdots \\
0 & 0 & 0 & & L_k^{\mathrm{T}} & & 0 & 0 & 0 & 0 & & 0 & 0 & & 0 & 0 \\
& & & & & \ddots & & & \vdots & & & & & & & \vdots \\
0 & 0 & 0 & & 0 & & 0 & L_\eta^{\mathrm{T}} & 0 & 0 & & 0 & 0 & & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & J_1^{\mathrm{T}} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & & 0 & & 0 & 0 & 0 & J_2^{\mathrm{T}} & & 0 & 0 & & 0 & 0 \\
& & & & & & & & \vdots & & \ddots & & & & & \vdots \\
0 & 0 & 0 & & 0 & & 0 & 0 & 0 & 0 & & J_{k-1}^{\mathrm{T}} & 0 & & 0 & 0 \\
0 & 0 & 0 & & 0 & & 0 & 0 & 0 & 0 & & 0 & J_k^{\mathrm{T}} & & 0 & 0 \\
& & & & & & & & \vdots & & & & & \ddots & & \vdots \\
0 & 0 & 0 & & 0 & & 0 & 0 & 0 & 0 & & 0 & 0 & & J_{\eta-1}^{\mathrm{T}} & 0 \\
0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & J_\eta^{\mathrm{T}}
\end{bmatrix}. \tag{2.2.4}$$

Examining Equations 2.2.3 and 2.2.4 we can see there are natural places to break the equations apart. First we can break the Lagrangian Multiplier, ν, into two subparts. The first, ρ, is aligned with the first, red, portion of $\partial\Theta^{\mathrm{T}}$ that is based on the state transition model, and the second, γ, is aligned with the second portion, based on the observation model. Additionally we can further break these vectors apart along the still preserved matrix dimensions so ρ_k is related to x_k and γ_k to y_k. The exact index ranges are

shown below.

$$\begin{aligned}
\nu_{[1\ldots\ell]} &= \rho_1 \\
\nu_{[\ell+1\ldots2\ell]} &= \rho_2 \\
\nu_{[(k-1)\ell+1\ldots k\ell]} &= \rho_k \\
\nu_{[(\eta-1)\ell+1\ldots\eta\ell]} &= \rho_\eta \\
\nu_{[\eta\ell+1\ldots\eta\ell+d_1]} &= \gamma_1 \\
\nu_{[\eta\ell+1+d_1\ldots\eta\ell+d_1+d_2]} &= \gamma_2 \\
\nu_{\left[\eta\ell+1+\sum_{j=1}^{k-1} d_j\ldots\eta\ell+\sum_{j=1}^{k-1} d_j+d_k\right]} &= \gamma_k \\
\nu_{\left[\eta\ell+1+\sum_{j=1}^{\eta-1} d_j\ldots\eta\ell+\sum_{j=1}^{\eta-1} d_j+d_\eta\right]} &= \gamma_\eta
\end{aligned}$$

Making this, the sub indexed vector ν, substitution into Equation 2.2.3, and noting that all the covariance matrices are symmetric, we get the following system of equations.

$$\begin{aligned}
-P_0^{-1}\left(x_0-\hat{x}_0\right)+F_1^{\mathrm{T}}\rho_1 &= 0 \\
-\rho_1+F_2^{\mathrm{T}}\rho_2+H_1^{\mathrm{T}}\gamma_1 &= 0 \\
-\rho_{k-1}+F_k^{\mathrm{T}}\rho_k+H_k^{\mathrm{T}}\gamma_k &= 0 \\
-\rho_\eta+H_\eta^{\mathrm{T}}\gamma_\eta &= 0 \\
-w_1Q_1^{-1}+L_1^{\mathrm{T}}\rho_1 &= 0 \\
-w_kQ_k^{-1}+L_k^{\mathrm{T}}\rho_k &= 0 \\
-w_\eta Q_\eta^{-1}+L_\eta^{\mathrm{T}}\rho_\eta &= 0 \\
-v_1R_1^{-1}+J_1^{\mathrm{T}}\gamma_1 &= 0 \\
-v_kR_k^{-1}+J_k^{\mathrm{T}}\gamma_k &= 0 \\
-v_\eta R_\eta^{-1}+J_\eta^{\mathrm{T}}\gamma_\eta &= 0
\end{aligned}$$

We can then simplify them into the following equations which include an initial condition, time evolving component, and end state for the dual state ρ.

$$\begin{aligned}
\rho_1 &= F_1^{-\mathrm{T}} P_0^{-1} \left(x_0 - \hat{x}_0\right) \\
\rho_k &= F_k^{-\mathrm{T}} \left(\rho_{k-1} - H_k^{\mathrm{T}} \gamma_k\right) \\
\rho_\eta &= H_\eta^{\mathrm{T}} \gamma_\eta \\
w_k &= Q_k L_k^{\mathrm{T}} \rho_k \\
v_k &= R_k J_k^{\mathrm{T}} \gamma_k
\end{aligned} \tag{2.2.5}$$

Using the constraints from Equations 2.1.1 and 2.1.2, we can combine updates[4].

$$\begin{aligned}
\rho_0 &= P_0^{-1} \left(\hat{x}_0 - x_0\right) && \text{Initial Condition} \\
\rho_k &= F_k^{-\mathrm{T}} \left(\rho_{k-1} - H_{k-1}^{\mathrm{T}} \gamma_{k-1}\right) && \text{Dual State Update} \\
x_k &= F_k x_{k-1} + u_k + L_k Q_k L_k^{\mathrm{T}} \rho_k && \text{System State Update} \\
\gamma_k &= \left(J_k R_k J_k^{\mathrm{T}}\right)^{-1} \left(y_k - H_k x_k\right) && \text{Noise Dual Condition} \\
\rho_\eta &= H_\eta^{\mathrm{T}} \gamma_\eta && \text{End Condition}
\end{aligned} \tag{2.2.6}$$

For simplicity we can introduce the following substitutions,

$$\begin{aligned}
\mathscr{Q}_k &= L_k Q_k L_k^{\mathrm{T}} \\
\mathscr{R}_k &= J_k R_k J_k^{\mathrm{T}} \\
\mathscr{H}_k &= H_k^{\mathrm{T}} \mathscr{R}_k^{-1} H_k.
\end{aligned} \tag{2.2.7}$$

Equation 2.2.6, can be rewritten in the Symplectic Form as

$$\begin{aligned}
\begin{bmatrix} x \\ \rho \end{bmatrix}_k &= \Gamma_k \begin{bmatrix} x \\ \rho \end{bmatrix}_{k-1} + \Upsilon_k \\
\Gamma_k &= \begin{bmatrix} F_k + \mathscr{Q}_k F_k^{-\mathrm{T}} \mathscr{H}_{k-1} & \mathscr{Q}_k F_k^{-\mathrm{T}} \\ F_k^{-\mathrm{T}} \mathscr{H}_{k-1} & F_k^{-\mathrm{T}} \end{bmatrix} \\
\Upsilon_k &= \begin{bmatrix} u_k - \mathscr{Q}_k F_k^{-\mathrm{T}} H_{k-1}^{\mathrm{T}} \mathscr{R}_{k-1}^{-1} y_{k-1} \\ -F_k^{-\mathrm{T}} H_{k-1}^{\mathrm{T}} \mathscr{R}_{k-1}^{-1} y_{k-1} \end{bmatrix}
\end{aligned} \tag{2.2.8}$$

We can define three additional matrices whose time evolving nature is an extension of the previous Equation, 2.2.8[5].

[4]The introduction of the initial dual state of ρ_0 completes the dual nature of the problem without introducing any real difficulties, we simply define $\{z_0, H_0\}$ to be empty.

[5]Notice that the matrices defined here are based on derivatives not on covariances. The equivalence can be verified by checking their update equations and initial conditions

$$
\begin{aligned}
\Psi_k &= \frac{\partial x_k}{\partial x_0} \\
\Xi_k &= \frac{\partial \rho_k}{\partial x_0} \\
\begin{bmatrix} \Psi \\ \Xi \end{bmatrix}_k &= \Gamma_k \begin{bmatrix} \Psi \\ \Xi \end{bmatrix}_{k-1} \\
C_k &= \frac{\partial x_k}{\partial \rho_k} = \Psi_k \Xi_k^{-1} = P_{k|k-1} \\
\Psi_0 &= I \\
\Xi_0 &= P_0^{-1}
\end{aligned}
\tag{2.2.9}
$$

The dual state gives a condition on optimality, that the state and dual should initialize, propagate, and terminate as in Equation 2.2.6. This is in contrast to the goal of minimizing the cost function, Equation 2.2.1. By itself this only gives us a test to verify that a solution could be optimal[6] and does not actually give us a way to generate a solution. To generate a solution the Kalman Filter uses a recursive solution to the end state[7] of this problem. As with any recursive technique, using the solution of the problem restricted to time $\eta - 1$ we can generate a solution for time η by applying an additional correction. A common notation is that the end state of a solution for time $k-1$ using observations from time up to and including $k-1$ is written as $x_{k-1|k-1}$, this implies that $\rho_{k-1|k-1} = H_{k-1}^{\mathrm{T}} \gamma_{k-1|k-1}$. The recursive algorithm of the Kalman Filter is typically broken into two parts *prediction* and *update*. The first step, called the *prediction step*[8] and shown in Equation 2.2.10, and is used to propagate forward our previous optimal state, $x_{k-1|k-1} \rightarrow x_{k|k-1}$, where $x_{k|k-1}$ is the filter's prediction of the state at time k given all previous information up to time $k-1$.

$$
\begin{aligned}
\rho_{k|k-1} &= F_k^{-\mathrm{T}} \left(\rho_{k-1|k-1} - H_{k-1}^{\mathrm{T}} \gamma_{k-1|k-1} \right) = 0 \\
x_{k|k-1} &= F_k x_{k-1|k-1} + u_k + \mathscr{Q}_k \rho_{k|k-1} \qquad = F_k x_{k-1|k-1} + u_k
\end{aligned}
\tag{2.2.10}
$$

In general this predicted value for the state and its dual will not meet the new end condition, $\rho_k = H_k^{\mathrm{T}} \gamma_k = H_k^{\mathrm{T}} \mathscr{R}_k^{-1} \left(y_k - H_k x_k \right)$, so we will need to *update* it to meet this new information. To do this we introduce a correction to the state, Δx_k, making our new estimate $x_{k|k} = x_{k|k-1} + \Delta x_k$, that will meet this

[6] The uniqueness of the solution can be inferred from the invertability of the system

[7] Often the end state is all you care about, where is the system right now given observations made in the past and then given this estimate how to move forward gaining new information to form a new estimate.

[8] The fact that $x_{k-1|k-1}$ was an optimal solution allowed us to simplify the expression $\left(\rho_{k-1|k-1} - H_{k-1}^{\mathrm{T}} \gamma_{k-1|k-1} \right)$ to 0 which, in turn, simplifies $\rho_{k|k-1}$ to 0. This is not that surprising, given only information from the past we cannot make any inference on the value of w_k so any optimal solution for time k not given an observation at time k must use the a priori most likely value for w_k, 0. Given Eq. 2.2.5, $w_k = Q_k L_k^{\mathrm{T}} \rho_k$, this in turn implies that ρ_k under this condition, $k|k-1$, must be 0.

new end condition. A difficulty arises because this correction changes both the state, $x_{k|k}$, and its dual[9], $\rho_{k|k} = \rho_{k|k-1} + P_k^{-1}\Delta x_k$, meaning we have to plug these changes into both sides of the end condition to solve for Δx_k.

$$\begin{aligned} \rho_{k|k-1} + C_k^{-1}\Delta x_k &= H_k^{\mathrm{T}}\mathscr{R}_k^{-1}\left(y_k - H_k\left(x_{k|k-1} + \Delta x_k\right)\right) \\ \left(C_k^{-1} + \mathscr{H}_k\right)\Delta x_k &= H_k^{\mathrm{T}}\mathscr{R}_k^{-1}\left(y_k - H_k x_{k|k-1}\right) \\ \Delta x_k &= \left(C_k^{-1} + \mathscr{H}_k\right)^{-1} H_k^{\mathrm{T}}\mathscr{R}_k^{-1}\left(y_k - H_k x_{k|k-1}\right) \end{aligned} \tag{2.2.11}$$

This process, Equation 2.2.11, becomes the *update step*, shown in Equation 2.2.12 with its associated Kalman Gain[10], K_k.

$$\begin{aligned} x_{k|k} = x_{k|k-1} + \Delta x_k &= x_{k|k-1} + K_k\left(y_k - H_k x_{k|k-1}\right) \\ K_k = \left(C_k^{-1} + H_k\mathscr{R}_k^{-1}H_k^{\mathrm{T}}\right)^{-1} H_k^{\mathrm{T}}\mathscr{R}_k^{-1} &= P_{k|k-1}H_k^{\mathrm{T}}\left(H_k P_{k|k-1}H_k^{\mathrm{T}} + \mathscr{R}_k\right)^{-1} \end{aligned} \tag{2.2.12}$$

For this to work we need an established initial point for the recursion at $k = 1$ where we need a solution for time $k - 1 = 0$, simply $x_{0|0} = \hat{x}_0$. This is the standard Kalman Filter, which other than some notational changes, is the same as in other works. There are two steps, the prediction step and update step. The *prediction step* which originates from Equation 2.2.8, and simplifies to its more common form, Equation 2.2.10, when given the current state meets the optimal solution's end condition. The *update step* in Equation 2.2.12 corrects the predicted state to match the new end condition in Equation 2.2.6 by using the Kalman Gain which is constructed from the twin notions of a covariance matrix and partial derivatives of Equation 2.2.9.

Although we have defined and used the matrix P_k we have not defined the propagation of this matrix. Equation 2.2.9 can be used to define this, the covariance/derivative update, which, after some manipulation, can be expressed in the following form. This update, although a bit out of order, has the same two parts, a *prediction step,* Equation 2.2.13, and an *update step,* Equation 2.2.14[11].

[9]This change is because we cannot simply change the end state x_k, we need to pick a new initial condition, x_0, that will propigate all the way through to result in the desired end state. We can propigate the changes in initial condition through Ψ_k and Ξ_k so $\Delta x_k = \Psi_k\Delta x_0$ and $\Delta\rho_k = \Xi_k\Delta x_0$, or alternatively $\Delta x_0 = \Psi_k^{-1}\Delta x_k$ and $\Delta\rho_k = \Xi_k\Psi_k^{-1}\Delta x_k$, which is why we need $C_k^{-1} = P_{k|k-1}^{-1} = \Xi_k\Psi_k^{-1}$.

[10]The change in form here is enabled by the matrix inverse identity $\left(A - BD^{-1}C\right)^{-1}BD^{-1} = A^{-1}B\left(D - CA^{-1}B\right)^{-1}$

[11]This is the the Joseph Normal form and performs better numerically than the form directly above it, $\left(I - K_{k-1}H_{k-1}\right)P_{k-1}$

$$
\begin{aligned}
\Psi_k &= \left(F_k + \mathscr{Q}_k F_k^{-\mathrm{T}} \mathscr{H}_{k-1}\right) \Psi_{k-1} + \left(\mathscr{Q}_k F_k^{-\mathrm{T}}\right) \Xi_{k-1} \\
\Xi_k &= \left(F_k^{-\mathrm{T}} \mathscr{H}_{k-1}\right) \Psi_{k-1} + F_k^{-\mathrm{T}} \Xi_{k-1} \\
C_k &= \Psi_k \Xi_k^{-1} \\
&= \left(\left(F_k + \mathscr{Q}_k F_k^{-\mathrm{T}} \mathscr{H}_{k-1}\right) \Psi_{k-1} + \mathscr{Q}_k F_k^{-\mathrm{T}} \Xi_{k-1}\right) \left(F_k^{-\mathrm{T}} \mathscr{H}_{k-1} \Psi_{k-1} + F_k^{-\mathrm{T}} \Xi_{k-1}\right)^{-1} \\
&= \left(F_k \Psi_{k-1} + \mathscr{Q}_k \left(F_k^{-\mathrm{T}} \mathscr{H}_{k-1} \Psi_{k-1} + F_k^{-\mathrm{T}} \Xi_{k-1}\right)\right) \Xi_{k-1}^{-1} \left(F_k^{-\mathrm{T}} \mathscr{H}_{k-1} \Psi_{k-1} \Xi_{k-1}^{-1} + F_k^{-\mathrm{T}}\right)^{-1} \\
&= \left(F_k P C_{k-1} + \mathscr{Q}_k \left(F_k^{-\mathrm{T}} \mathscr{H}_{k-1} C_{k-1} + F_k^{-\mathrm{T}}\right)\right) \left(F_k^{-\mathrm{T}} \mathscr{H}_{k-1} \Psi_{k-1} \Xi_{k-1}^{-1} + F_k^{-\mathrm{T}}\right)^{-1} \\
&= F_k C_{k-1} \left(F_k^{-\mathrm{T}} \mathscr{H}_{k-1} C_{k-1} + F_k^{-\mathrm{T}}\right)^{-1} + \mathscr{Q}_k \\
&= F_k C_{k-1} \left(\mathscr{H}_{k-1} C_{k-1} + I\right)^{-1} F_k^{\mathrm{T}} + \mathscr{Q}_k \\
P_{k|k-1} &= F_k \left(\mathscr{H}_{k-1} + C_{k-1}^{-1}\right)^{-1} F_k^{\mathrm{T}} + \mathscr{Q}_k \\
\left(\mathscr{H}_k + C_k^{-1}\right)^{-1} &= P_{k|k} \\
P_{k|k-1} &= F_k P_{k-1|k-1} F_k^{\mathrm{T}} + \mathscr{Q}_k \qquad (2.2.13) \\
P_{k|k} &= \left(\mathscr{H}_k + C_k^{-1}\right) = C_k - C_k H_k^{\mathrm{T}} \left(\mathscr{R}_k + H_k C_k H_k^{\mathrm{T}}\right)^{-1} H_k C_k \\
&= \left(I - K_k H_k^{\mathrm{T}}\right) P_{k|k-1} \\
P_{k|k} &= \left(I - K_k H_k^{\mathrm{T}}\right) P_k \left(I - K_k H_k^{\mathrm{T}}\right)^{\mathrm{T}} + K_k \mathscr{R}_k K_k^{\mathrm{T}} \qquad (2.2.14)
\end{aligned}
$$

2.3 Properties of the Symplectic Update Γ

Before we continue it will be helpful to discuss some of the properties of the symplectic system constructed in Equation 2.2.8. The first property of interest is that the new state/dual propagation matrix Γ is, as suggested symplectic, which is a matrix defined by the property given in 2.3.1.

$$
\begin{aligned}
M &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \\
\Gamma^{-1} &= M^{\mathrm{T}} \Gamma^{\mathrm{T}} M \qquad (2.3.1)
\end{aligned}
$$

This property gives us an easy way to construct the inverse of Γ which will be useful when we actually need to implement the method. The most immediate consequence of this is that the determinant of Γ is 1. Additionally we have the property given in Equation 2.3.2.

$$
\lambda \text{ is eigenvalue of } \Gamma \implies 1/\lambda \text{ is an eigenvalue of } \Gamma \qquad (2.3.2)
$$

These facts will have implications as we look at how to propagate entire solution paths Chapter 3, keeping in mind that the Kalman Filter only defined a recursive solution to the end state. As more of a curiosity, note that scaling the update noise variance by a scale factor α, $\tilde{Q} = \alpha Q$, does not change any eigenvalues if paired with the same scaling of the observation noise variance, $\tilde{R} = \alpha R$.

$$
\begin{aligned}
\det\left(\Gamma - \lambda I\right) &= \det\left(F^{-\mathrm{T}} - \lambda I\right) \det\left(F + \mathscr{Q}F^{-\mathrm{T}}\mathscr{H} - \lambda I - \mathscr{Q}F^{-\mathrm{T}}\left(F^{-1} - \lambda I\right)F^{-\mathrm{T}}\mathscr{H}\right) \\
\det\left(\tilde{\Gamma} - \lambda I\right) &= \det\left(F^{-\mathrm{T}} - \lambda I\right) \det\left(F + \tilde{\mathscr{Q}}F^{-\mathrm{T}}\tilde{\mathscr{H}} - \lambda I - \tilde{\mathscr{Q}}F^{-\mathrm{T}}\left(F^{-\mathrm{T}} - \lambda I\right)^{-1}F^{-\mathrm{T}}\tilde{\mathscr{H}}\right) \\
&= \det\left(F^{-\mathrm{T}} - \lambda I\right) \det\left(F + \alpha\mathscr{Q}F^{-\mathrm{T}}\frac{1}{\alpha}\mathscr{H} - \lambda I - \alpha\mathscr{Q}F^{-\mathrm{T}}\left(F^{-\mathrm{T}} - \lambda I\right)^{-1}F^{-\mathrm{T}}\frac{1}{\alpha}\mathscr{H}\right) \\
&= \det\left(F^{-\mathrm{T}} - \lambda I\right) \det\left(F + \mathscr{Q}F^{-\mathrm{T}}\mathscr{H} - \lambda I - \mathscr{Q}F^{-\mathrm{T}}\left(F^{-1} - \lambda I\right)F^{-\mathrm{T}}\mathscr{H}\right) \\
\det\left(\tilde{\Gamma} - \lambda I\right) &= \det\left(\Gamma - \lambda I\right)
\end{aligned}
$$

A more practical implication is that the update matrix Γ preserves the identity defined in Equation 2.3.3.

$$\rho_k = C_k^{-1}\left(x_k - x_{k|k-1}\right) \tag{2.3.3}$$

To see this first note the ramification of the definition from Equation 2.2.10, which in this recursive notation makes $\rho_{k|k-1} = 0$. From the derivative in Equation 2.2.9 we have $\rho_k = C_k^{-1}\left(x_k - x_{k|k-1}\right) + \rho_{k|k-1}$ which simplifies to the above Equation 2.3.3. This condition, Equation 2.3.3, will be important later as a way to correct numerical errors introduced machine computation with finite precision arithmetic.

Chapter 3

Smoothing Solutions

Behind our efforts, let there be found our efforts

- Gene Wolfe, *The Book of the New Sun*

Now that we have a recursive solution to the end state of the state space estimation problem defined in the previous chapter it would be helpful to be able to find the total solution path based on all the information, $\begin{bmatrix} x_{1|\eta} & x_{2|\eta} & \dots & x_{\eta|\eta} \end{bmatrix}$, also called the *smoothed* solution. This should be as easy as inverting the state/dual propagation equation 2.2.8 which given the property of symplectic matrices, 2.3.1, should be straightforward. Unfortunately it is a bit more complicated. To demonstrate why, let us consider the following problem.

3.1 Weather-vane Problem

Problem Definition

We are given that a weather-vane's angular motion can be modeled by the following continuous time state-space model[1]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} w, \tag{3.1.1}$$

where x_1 is the weather-vane's angle, x_2 its angular velocity, ζ is the amount of friction in the bearing, ω is the strength of the wind, and w is the direction of the wind. The wind direction, w, is the angle that

[1] This example is a slight modification of the one introduced on page 333 and examined on 348 of [14]. The changes in the noise variances of the problem are made for the benefit of the smoothing examples of the previous chapter.

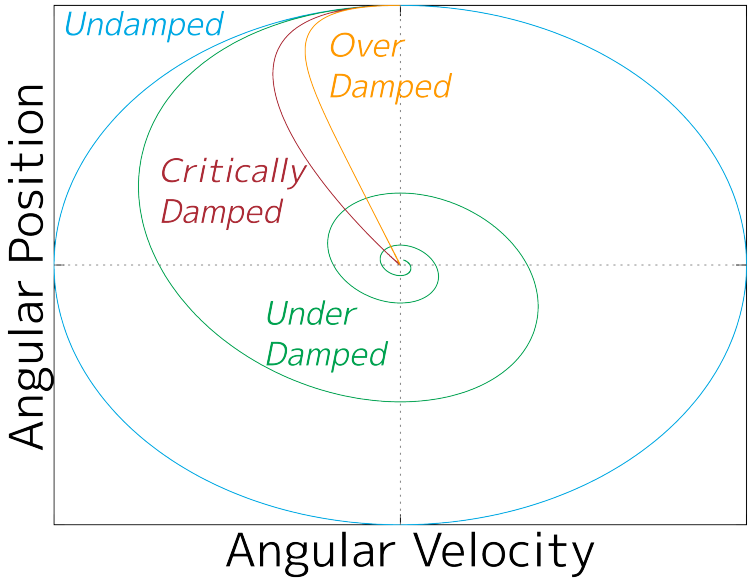

Figure 3.1.1: Phase Plane Plot of Weather-vane Problem

the weathervane will be forced to by the wind and its strength ω determines the velocity at which it will move towards that equilibrium. This is a pretty generic oscillator problem, with forcing and damping but linear in form. We can see the expected phase plane motion for different damping factors towards a fixed equilibrium, $w_k = w$, in Figure 3.1.1.

For our problem we will assume that $\zeta = 0.28$, $\omega = 2\pi$, and that for intervals of time, Δt, of 0.1s the wind direction is constant, the total time of the test will be 3s, $\eta = 30$. To discretize the problem we first solve for the discrete state transition matrix, F, by finding the matrix exponential of the continuous time state matrix[2], which has eigenvalues $\lambda = \{\omega\left(\zeta^2 - 1\right) - \omega\zeta, -\omega\left(\zeta^2 - 1\right) - \omega\zeta\}$.

[2]This exponentiation is based on Putzer's Algorithm.

$$F = \exp\left(\begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \Delta t\right) = r_1(\Delta t)\, S_0 + r_2(\Delta t)\, S_1$$

$$r_1(t) = \exp(\lambda_1 t)$$

$$r_2(t) = \frac{1}{\lambda_1 - \lambda_2}\left(\exp(\lambda_1 t) - \exp(\lambda_2 t)\right) \qquad \text{if } \lambda_1 \neq \lambda_2$$

$$r_2(t) = t\exp(\lambda_1 t) \qquad \text{if } \lambda_1 = \lambda_2$$

$$S_0 = I$$

$$S_1 = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} - \lambda_1 I$$

$$F \approx \begin{bmatrix} 0.829 & 0.079 \\ -3.114 & 0.552 \end{bmatrix}$$

The propagation noise transformation matrix, L, is found through the following equation.

$$L = (F - I)\begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ \omega^2 \end{bmatrix}$$

$$L \approx \begin{bmatrix} 0.171 \\ 3.114 \end{bmatrix}$$

$$w_k \sim N(0, 1)$$

We assumed that the wind direction is constant for $\Delta t = 0.1$ the direction will be Normally distributed, $w_k \sim N(0, 1)$.

At each step we measure both states of the system with varying observation noise.

$$z_k = Hx_k + v_k$$

$$H = I$$

$$v_k \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 0.36 \end{bmatrix}\right)$$

We initialize the system with $\hat{x}_0 \sim N\left(\begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 0.01 \end{bmatrix}\right)$, so $x_0 \sim N\left(\hat{x}_0, \begin{bmatrix} 4 & 0 \\ 0 & 0.01 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$

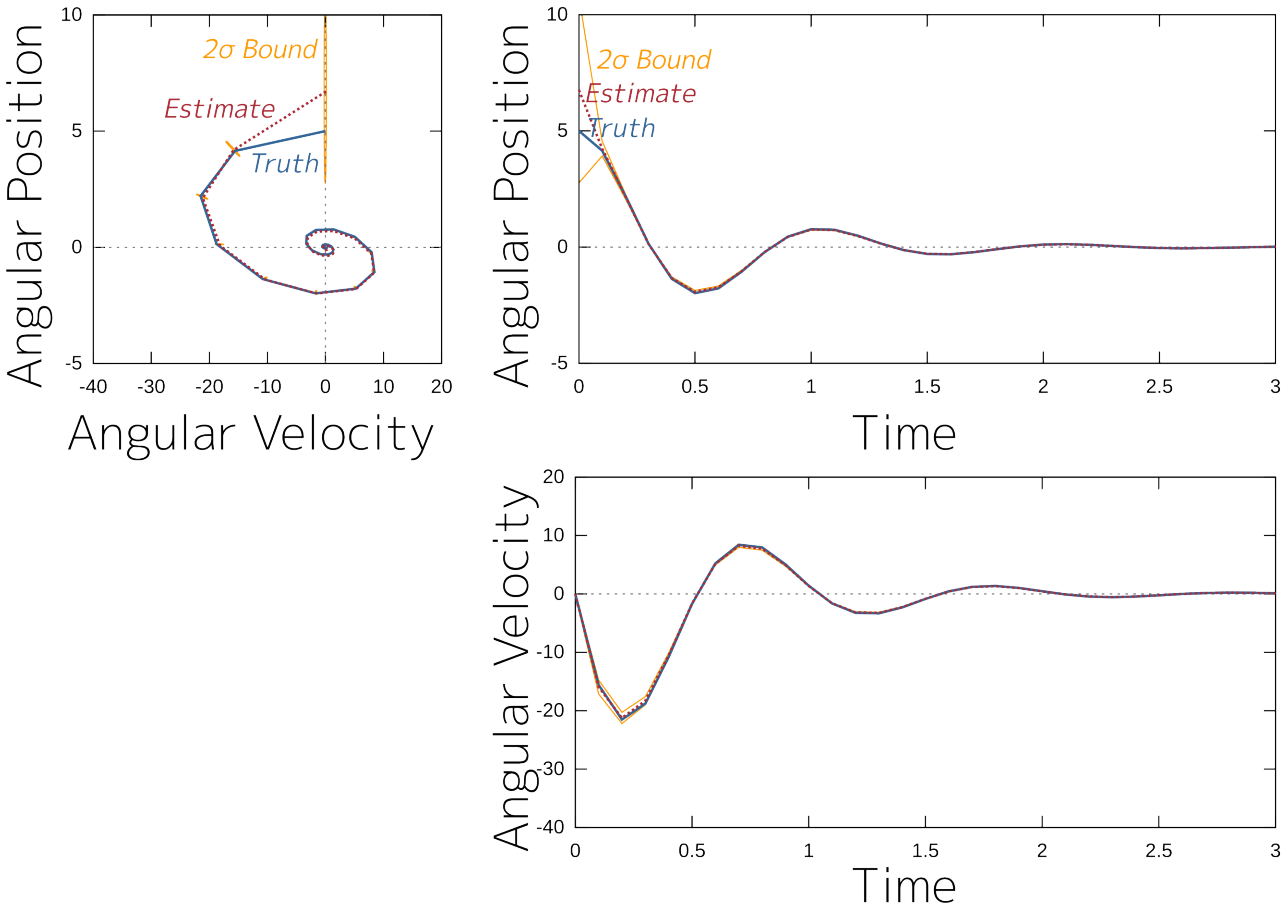

Figure 3.1.2: Kalman Filter with only Initial Uncertainty

Kalman Filter Solution

From this we can implement a full Kalman Filter scenario, but before we do, consider some simplifications first.

Problem 1: Initial Uncertainty Consider first how the the problem would behave if we had initial uncertainty but no propagation noise, that is $w \sim N(0, Q), Q = 0$. Here we would expect that the filter would start with a lot of uncertainty and then eventually, through the observations, solve for the position of the system. This is seen to be the case in Figure 3.1.2.

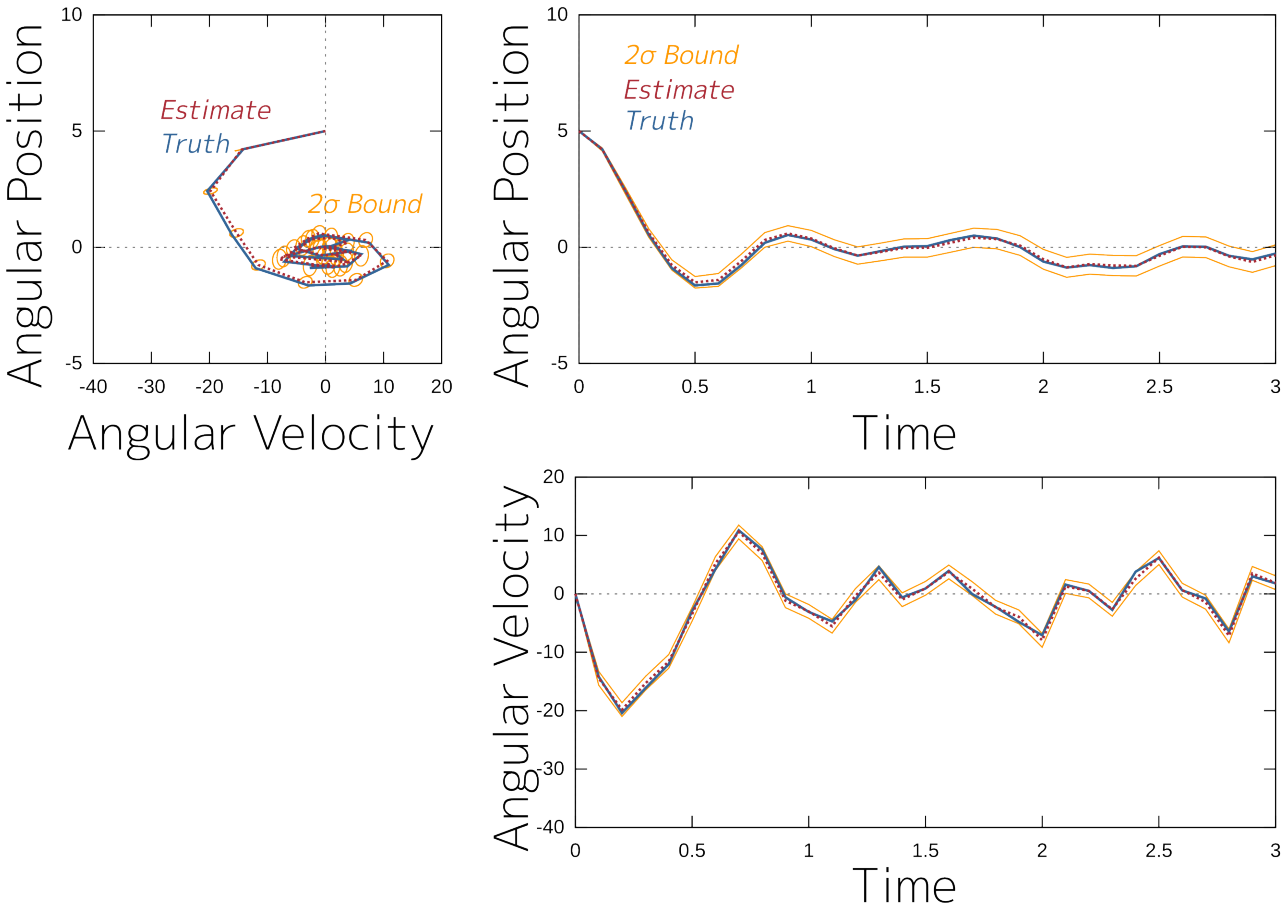

Figure 3.1.3: Kalman Filter with No Initial Uncertainty

Problem 2: Just Propagation Noise

Consider second how the filter would behave with no initial uncertainty but with propagation noise, that is $P_0 = 0$ and $w \sim N(0, Q), Q = 1$. Here the filter starts with perfect knowledge of the location of the weather-vane but then can only track it through observations. The filter starts with perfect knowledge of the location and velocity of the weather vane but as time goes on this knowledge is eroded by the process noise until it reaches a steady state between this loss of information at each step and the gain from the observation. This result can be seen in Figure 3.1.3.

Problem 3: Full Kalman Filter Problem

We can compare these two sub problems against the full Kalman Filter problem with both initial uncertainty and propagation noise. Here we expect to see the combination of the effects of each of the

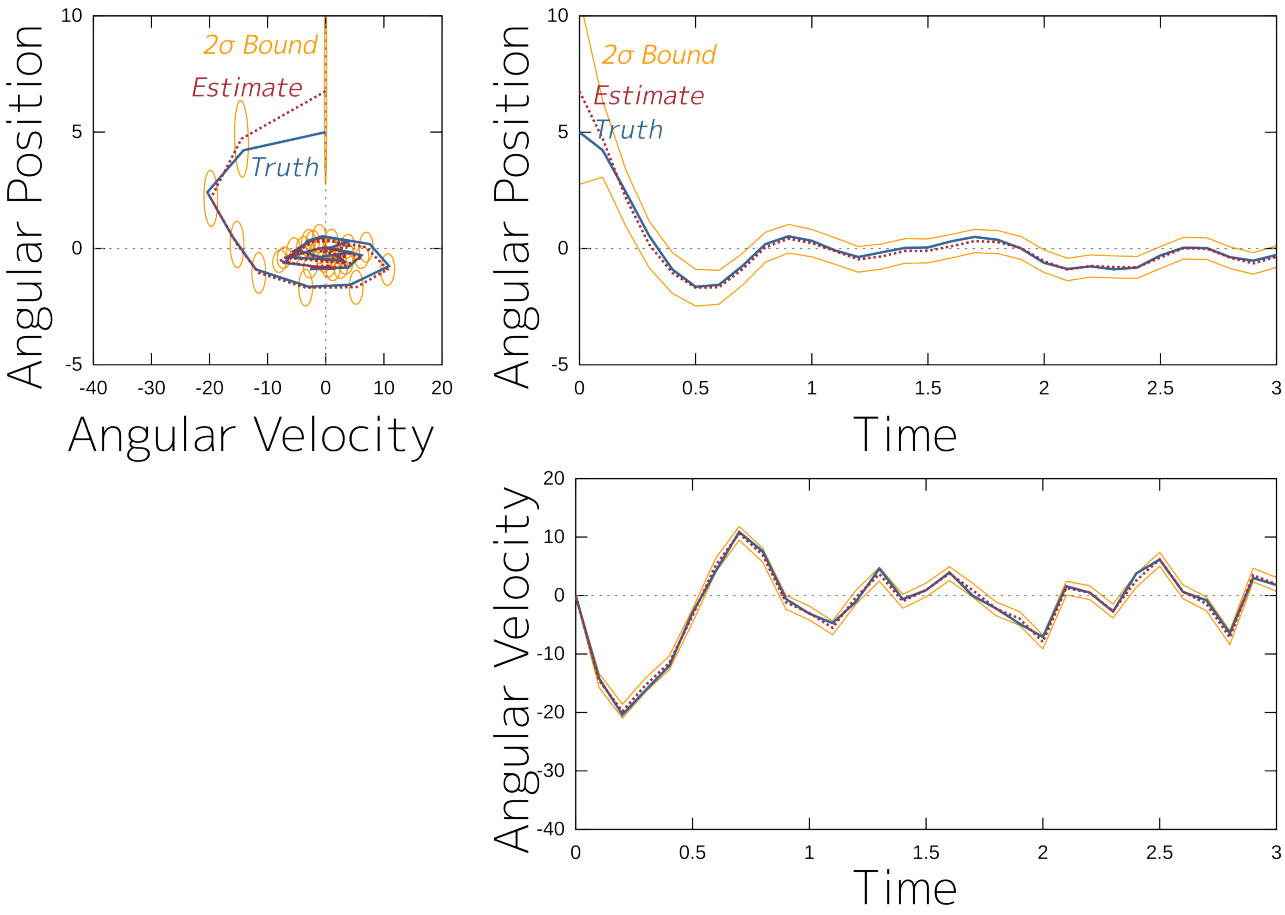

Figure 3.1.4: Full Kalman Filter Solution

previous examples, the solving of the large initial uncertainty and the eventual steady state between the process noise/observations. The full result can be seen in Figure 3.1.4.

3.2 Reversing the Solution

When we look at the solution given in the three previous examples we see only how the estimate of the current state evolves over time. This path, because it is a stitching together of many estimates, lacks physicality, evident in Figure 3.1.2 where the solution makes an impossible movement from the initial point [3]. As outlined before we can reverse the state/dual propagation equation, 2.2.8. That is

[3] It is not physically possible for a weather vane starting in the initial state estimate to propagate forward to the next estimate. Recall that there is no propagation noise, so the weather vane's movements are completely determined. The large jump in position is mostly explained by a change in the estimate of the state due to the change in the information about the state, not the state itself.

$$\begin{bmatrix} x \\ \rho \end{bmatrix}_k = \begin{bmatrix} F_k + \mathscr{Q}_k F_k^{-\mathrm{T}} \mathscr{H}_{k-1} & \mathscr{Q}_k F_k^{-\mathrm{T}} \\ F_k^{-\mathrm{T}} \mathscr{H}_{k-1} & F_k^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} x \\ \rho \end{bmatrix}_{k-1} + \begin{bmatrix} u_k - \mathscr{Q}_k F_k^{-\mathrm{T}} H_{k-1}^{\mathrm{T}} \mathscr{R}_{k-1}^{-1} y_{k-1} \\ -F_k^{-\mathrm{T}} H_{k-1}^{\mathrm{T}} \mathscr{R}_{k-1}^{-1} y_{k-1} \end{bmatrix}$$

$$\begin{bmatrix} x \\ \rho \end{bmatrix}_{k-1} = \Gamma_k^{-1} \left(\begin{bmatrix} x \\ \rho \end{bmatrix}_k - \Upsilon_k \right)$$

becomes

$$\begin{bmatrix} x \\ \rho \end{bmatrix}_{k-1} = \begin{bmatrix} F_k^{-1} & -F_k^{-1} \mathscr{Q} \\ -\mathscr{H}_{k-1} F_k^{-1} & F_k^{\mathrm{T}} + \mathscr{H}_{k-1} F_k^{-1} \mathscr{Q}_k \end{bmatrix} \begin{bmatrix} x \\ \rho \end{bmatrix}_k + \begin{bmatrix} -F_k^{-1} u_k \\ \mathscr{H}_{k-1} F_k^{-1} u_k + H_{k-1}^{\mathrm{T}} \mathscr{R}_{k-1}^{-1} y_{k-1} \end{bmatrix}. \tag{3.2.1}$$

As an aside, there is another way to look at Equation 3.2.1, with a simple flipping of x and ρ, the equation becomes as follows.

$$\begin{bmatrix} \rho \\ x \end{bmatrix}_{k-1} = \begin{bmatrix} F^{\mathrm{T}} + \mathscr{H}_{k-1} F_k^{-1} \mathscr{Q}_k & -\mathscr{H}_{k-1} F_k^{-1} \\ -F_k^{-1} \mathscr{Q}_k & F^{-1} \end{bmatrix} \begin{bmatrix} \rho \\ x \end{bmatrix}_k + \begin{bmatrix} \mathscr{H}_{k-1} F_k^{-1} u_k + H_{k-1}^{\mathrm{T}} \mathscr{R}_{k-1}^{-1} y_{k-1} \\ -F_k^{-1} u_k \end{bmatrix} \tag{3.2.2}$$

Compare this equation, 3.2.2, to the original one, 2.2.8, and we can see many parallels in the structure. Where originally the state moved forward through the action of a state transition matrix F its dual moves backwards with the action of the matrix F^{T}; where there was a forcing function of u there is now $H^{\mathrm{T}} \mathscr{R}^{-1} y$, and $\mathscr{H}$ and $\mathscr{Q}$ have switched places and acquired negative signs.

By estimating the entire problem using rational arithmetic[4] we can reverse the full Kalman Filter solution, seen in Figure 3.1.4, to find the estimated path given all the data, shown in Figure 3.2.1. This process, of finding the complete estimated path, is often referred to as smoothing and usually is expressed in a different form which we will be exploring later, shown in Equation 3.4.1.

3.3 Numerical Issues

The smoothed results obtained for Figure 3.2.1 are only possible with perfect arithmetic, which was possible by using rational arithmetic representations of the problem. This strategy will for significant problems take to much time to compute, and as stated, there are some numerical issues we must deal with. Using floating point computations just for the smoothing gives us the result shown in Figure 3.3.1. Here in a matter of a few iterations the entire path has diverged from the perfect arithmetic solution, a highly undesirable result.

[4] That is instead of solving the original problem, we solve one very similar to it that has a property that we want, mainly that we can express it in rational numbers for exact computation.

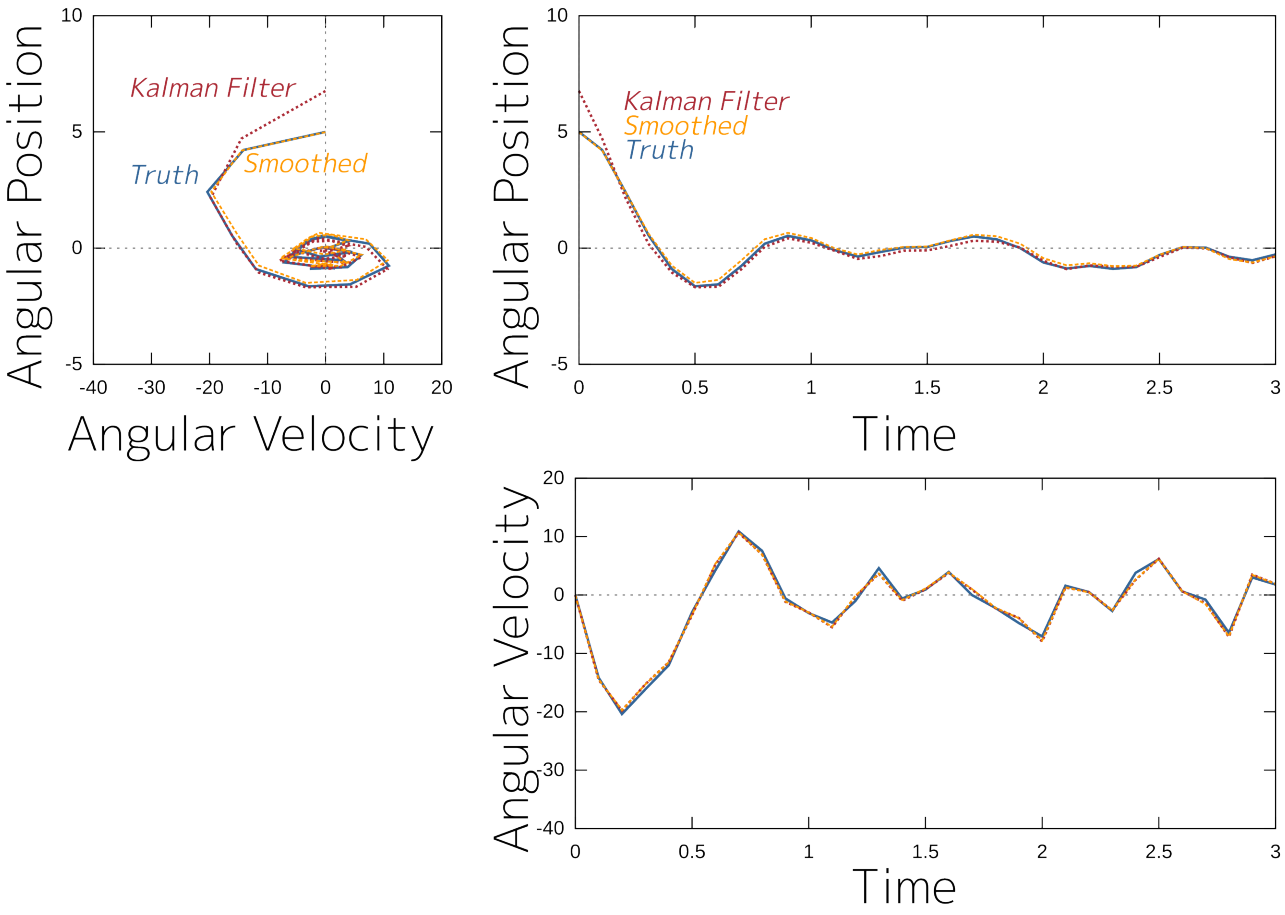

Figure 3.2.1: Reversed/Smoothed Solution Path

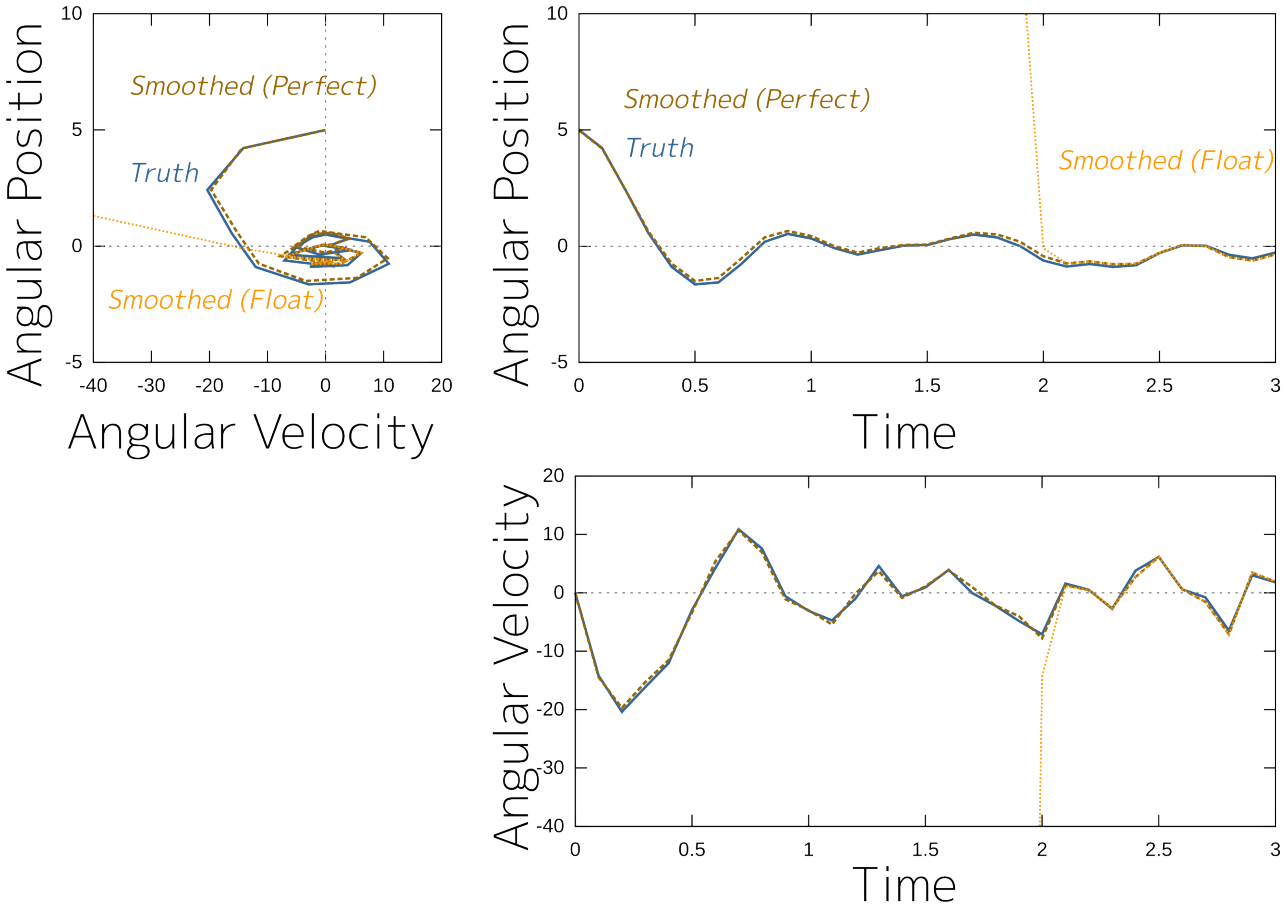

Figure 3.3.1: Floating Point Smoothed Path

Before we explore the problem with the reverse track iterations, first let's explore a numerical issue that we have already avoided. Recall that the propagation equations from 2.2.8, $\begin{bmatrix} x \\ \rho \end{bmatrix}_k = \Gamma_k \begin{bmatrix} x \\ \rho \end{bmatrix}_{k-1} + \Upsilon_k$, work for any given point\dual and if they happen to result in the end condition from Equation 2.2.6, $\rho_\eta = H_\eta^{\mathrm{T}} \mathscr{R}^{-1} \left(y_\eta - H_\eta x_\eta \right)$, then additionally it is the optimal solution. In our notation the initial condition $\begin{bmatrix} x \\ \rho \end{bmatrix}_{0|\eta}$ is the initial condition that will meet this end condition, in fact $x_{0|\eta}$ is the one variable we have been searching for, as all other variables can be found from it as a result of both the original constraints in Equation 2.2.2 and the new equations added with the dual state in Equation 2.2.5. We can ask what would happen if we changed this initial state just a little, from $x_{0|\eta}$ to $\tilde{x}_{0|\eta} = x_{0|\eta} + \epsilon$ and $\tilde{\rho}_{0|\eta} = \rho_{0|\eta} + P_0^{-1}\epsilon$. We can calculate how this would impact the final state, $\tilde{x}_{\eta|\eta}$, with the matrix from Equation 2.2.9, $\Psi_\eta = \frac{\partial x_\eta}{\partial x_0}$. In our test however Ψ has grown very large over the interval, $\Psi_{30} \approx \begin{bmatrix} -7.54 & 480 \\ -131 & 835 \end{bmatrix} \times 10^{45}$, making the problem very sensitive to this kind of error. It is not specific to this problem either. We can show the determinant of this matrix is guaranteed to grow relative to what we would expect from just the state transition matrix[5].

$$
\begin{aligned}
\Psi_k &= \left(F_k + \mathscr{Q}_k F_k^{-\mathrm{T}} \mathscr{H}_{k-1} \right) \Psi_{k-1} + \left(\mathscr{Q}_k F_k^{-\mathrm{T}} \right) \Xi_{k-1} \\
&= \left(F_k + \mathscr{Q}_k F_k^{-\mathrm{T}} \mathscr{H}_{k-1} \right) + \left(\mathscr{Q}_k F_k^{-\mathrm{T}} \right) \Xi_{k-1} \Psi_{k-1}^{-1} \left| \Psi_{k-1} \right| \\
&= F_k \left(I + F_k^{-1} \mathscr{Q}_k F_k^{-\mathrm{T}} \left(\mathscr{H}_{k-1} + C_{k-1}^{-1} \right) \right) \Psi_{k-1} \\
\left| I + F_k^{-1} \mathscr{Q}_k F_k^{-\mathrm{T}} \left(\mathscr{H}_{k-1} + C_{k-1}^{-1} \right) \right| &= \left| P_{k-1|k-1} \right| \left| P_{k-1|k-1}^{-1} + P_{k-1|k-1}^{-1} F_k^{-1} \mathscr{Q}_k F_k^{-\mathrm{T}} P_{k-1|k-1}^{-1} \right| \\
&\geq \left| P_{k-1|k-1} \right| \left(\left| P_{k-1|k-1}^{-1} \right| + \left| P_{k-1|k-1}^{-1} F_k^{-1} \mathscr{Q}_k F_k^{-\mathrm{T}} P_{k-1|k-1}^{-1} \right| \right) \\
&\geq 1 + \left| P_{k-1|k-1} \right| \left| P_{k-1|k-1}^{-1} F_k^{-1} \mathscr{Q}_k F_k^{-\mathrm{T}} P_{k-1|k-1}^{-1} \right| \\
&\geq 1 \\
\frac{|\Psi_k|}{|F_k|} &\geq |\Psi_{k-1}|
\end{aligned}
$$

This fact also means that volumes are going to get larger than we would expect from just the application of the state transition matrix, F. Imagine the volume surrounded by the 1σ radius ellipse at the beginning, this volume ends up bigger when propagated through the state/dual system than if simply

[5]Recall the identity from Equation ,$P_{k|k} = \left(\mathscr{H}_k + C_k^{-1} \right)^{-1}$. The inequality in step 5 is a result of the fact that the determinant is concave for Hermitian non-negative matrices so we can apply Minkowski's Inequality.

propagated through the original system. This is an interesting and perhaps counter intuitive consequence, and we can strengthen this idea, that the state/dual system pushes points farther apart than the original system would. A change to the state, $\tilde{x}_k = x_k + \delta$, would normally propagate to the next state and cause a change $F_k\delta$. Instead, in the state/dual system, this propagates through $\Psi_k\Psi_{k-1}^{-1}$. We will show that this new vector $\Psi_k\Psi_{k-1}^{-1}\delta$ larger than or equal to the expected change, $F_k\delta$, in the same direction[6], that is $\left(\delta^{\mathrm{T}}F_k^{\mathrm{T}}\right)\left(\Psi_k\Psi_{k-1}^{-1}\delta\right) \geq \left(\delta^{\mathrm{T}}F_k^{\mathrm{T}}\right)\left(F_k\delta\right)$ which we will show by demonstrating the matrix $F_k^{\mathrm{T}}\Psi_k\Psi_{k-1}^{-1}$ is positive semidefinite.

$$
\begin{aligned}
\left(\delta^{\mathrm{T}}F_k^{\mathrm{T}}\right)\left(\Psi_k\Psi_{k-1}^{-1}\delta\right) &= \delta^{\mathrm{T}}F_k^{\mathrm{T}}\left(F_k + \mathscr{Q}_k F_k^{-\mathrm{T}}P_{k-1|k-1}^{-1}\right)\delta \\
&= \delta^{\mathrm{T}}F_k^{\mathrm{T}}F_k\delta + \delta^{\mathrm{T}}F_k^{\mathrm{T}}\mathscr{Q}_k F_k^{-\mathrm{T}}P_{k-1|k-1}^{-1}\delta \\
\mathscr{Q}_k &\text{ is Positive Semidefinite} \\
F_k^{\mathrm{T}}\mathscr{Q}_k F_k^{-\mathrm{T}} &\text{ is Positive Semidefinite} \\
P_{k-1|k-1} &= BB^{\mathrm{T}} \text{ Cholesky Decomposition} \\
B^{-1}F_k^{\mathrm{T}}\mathscr{Q}_k F_k^{-\mathrm{T}}B^{-\mathrm{T}} &\text{ is Positive Semidefinite} \\
B\left(B^{-1}F_k^{\mathrm{T}}\mathscr{Q}_k F_k^{-\mathrm{T}}B^{-\mathrm{T}}\right)B^{-1} &\text{ is Positive Semidefinite} \\
&= F_k^{\mathrm{T}}\mathscr{Q}_k F_k^{-\mathrm{T}}P_{k-1|k-1}^{-1}
\end{aligned}
$$

▪

As a special case consider the outcome given $\mathscr{Q} = 0$, i.e. the case where there is no propagation noise, as we did for Figure 3.1.2. Here $\Psi_k = F_k\Psi_{k-1}$ and equality holds. And if we had another time step, $\eta + 1$, there would be another propagation and correction step which would adjust the state at time η, $\begin{bmatrix} x \\ \rho \end{bmatrix}_{\eta|\eta+1} = \begin{bmatrix} x \\ \rho \end{bmatrix}_{\eta|\eta} + \delta$. In our special case, $\mathscr{Q} = 0$, then $\Psi_{k+1} = F_{k+1}\Psi_k$ (by extension $\Psi_\eta = \prod F_k$), this adjustment to the state at time η has to back propagate in its entirety. Any time there is propagation noise, $\mathscr{Q} \neq 0$, the full impact of the change does not have to fully propagate back through Ψ_η^{-1}. We can see this in the case where we have only propagation uncertainty in Figure 3.1.3. When we make a correction to the final state we do not expect that it will have much impact on the initial state of the system. That is we expect that $\Psi_\eta^{-1} \leq \prod F_k^{-1}$ which implies that $\Psi_\eta \geq \prod F_k$.

Contrary to what this would suggest, using typical Kalman Filter equations, we can easily run the filter forward using floating point arithmetic without difficulty. Every time we make a correction to the current state, as in the update step of Equation 2.2.12, we don't bother to update all the reverse states, the correction,Δx_k, is propagated back to the original state through Ψ_k^{-1} but then immediately back to

[6]The fact that$\Psi_k\Psi_{k-1}^{-1}\delta$ projected onto $F\delta$ is greater than or equal to $F\delta$ also implies that $\Psi_k\Psi_{k-1}^{-1}\delta$ is greater than or equal overall. For this note that if A is positive semidefinite then $x^{\mathrm{T}}Ax \geq 0$ and both BAB^{T} and $BAB^{\text{-}1}$ are positive semidefinite

the current state/dual with Ψ_k and Ξ_k, the fact that $\Psi_k\Psi_k^{-1} = I$ and $\Xi_k\Psi_k^{-1} = C_k^{-1} = P_{k|k-1}^{-1}$ ensures that this back and forth between the current state and the initial state does not have to utilize the numerically unstable matricesΨ and Ξ.

From this it should be clear that running the state/dual forward to get an full solution path is not going to be practical[7]. On the other hand this suggests that it should be very easy to run the system backwards, even though it appears, as shown in Figure 3.3.1, not to be the case. All the previous examples assume that the any error meets the condition $\rho_k = P_{k|k-1}^{-1}\left(x_k - x_{k|k-1}\right)$, which, in general, they will not. To examine how an error not meeting those requirements would impact things we need to look at the symplectic state/dual transition matrix, Γ. Recall from Chapter 2 that Γ's eigenvalues come in pairs, $\lambda \implies 1/\lambda$, so, assuming that all the eigenvalues are not 1, we will have some eigenvalues less than 1 and its inverse, Γ^{-1}, will have some greater than 1. Additionally because Γ is constant these eigenvalues will simply multiply together, λ is an eigenvalue of $\Gamma \implies \lambda^n$ is an eigenvalue of $\prod_{k=1}^{n}\Gamma_k$, and any error in the direction of an eigenvector with an eigenvalue greater that 1 will grow exponentially. In our example the largest eigenvalue of Γ^{-1} is ≈ 40. We show the impact of a non-P-constraint error in Figure 3.3.2 where we can see the effect of running the system back, in exact arithmetic, after adding a small error, $\begin{bmatrix}\tilde{x}\\ \rho\end{bmatrix}_\eta = \begin{bmatrix}x\\ \rho\end{bmatrix}_\eta + \epsilon$, generated so that the error introduced to the state, $\tilde{x}_\eta = x_\eta + \epsilon_x$, is scale very small, $\|\epsilon_x\| \approx 1E-14$. If instead we introduce an error which is much larger, $\|\epsilon_x\| = 1$, which meets the constraint, $\rho_\eta + \epsilon_\eta = P_{\eta|\eta-1}^{-1}\left(x_\eta + \epsilon_x\right)$, as shown in Figure 3.3.3 the solution is adjusted but does not explode.

3.4 Constraint Based Solution

At every step we introduce some new numerical error, so to ensure that we still meet the constraint we can simply apply the constraint before going back another step. Recall the typical back iterate from Equation 3.2.1,

$$\begin{bmatrix}x\\ \rho\end{bmatrix}_{k-1} = \begin{bmatrix}F_k^{-1} & -F_k^{-1}\mathscr{Q} \\ -\mathscr{H}_{k-1}F_k^{-1} & F_k^{\mathrm{T}} + \mathscr{H}_{k-1}F_k^{-1}\mathscr{Q}_k\end{bmatrix}\begin{bmatrix}x\\ \rho\end{bmatrix}_k + \begin{bmatrix}-F_k^{-1}u_k \\ \mathscr{H}_{k-1}F_k^{-1}u_k + H_{k-1}^{\mathrm{T}}\mathscr{R}_{k-1}^{-1}y_{k-1}\end{bmatrix}.$$

[7] If F has a small eigenvalue and there is very little propagation noise $\mathscr{Q} \approx 0$ then it might seem easier to run the solution forward. This would be the case for the ONLY INITIAL UNCERTAINTY PROBLEM in Figure 3.1.2 where the damping has almost completely reduced the state to 0. We have not really avoided the problem however because this implies that $\Psi^{-1} \gg 1$, which would make it hard to track how corrections in later states need to be propagated back to the initial state. Additionally ρ propagates through $F^{-\mathrm{T}}$.

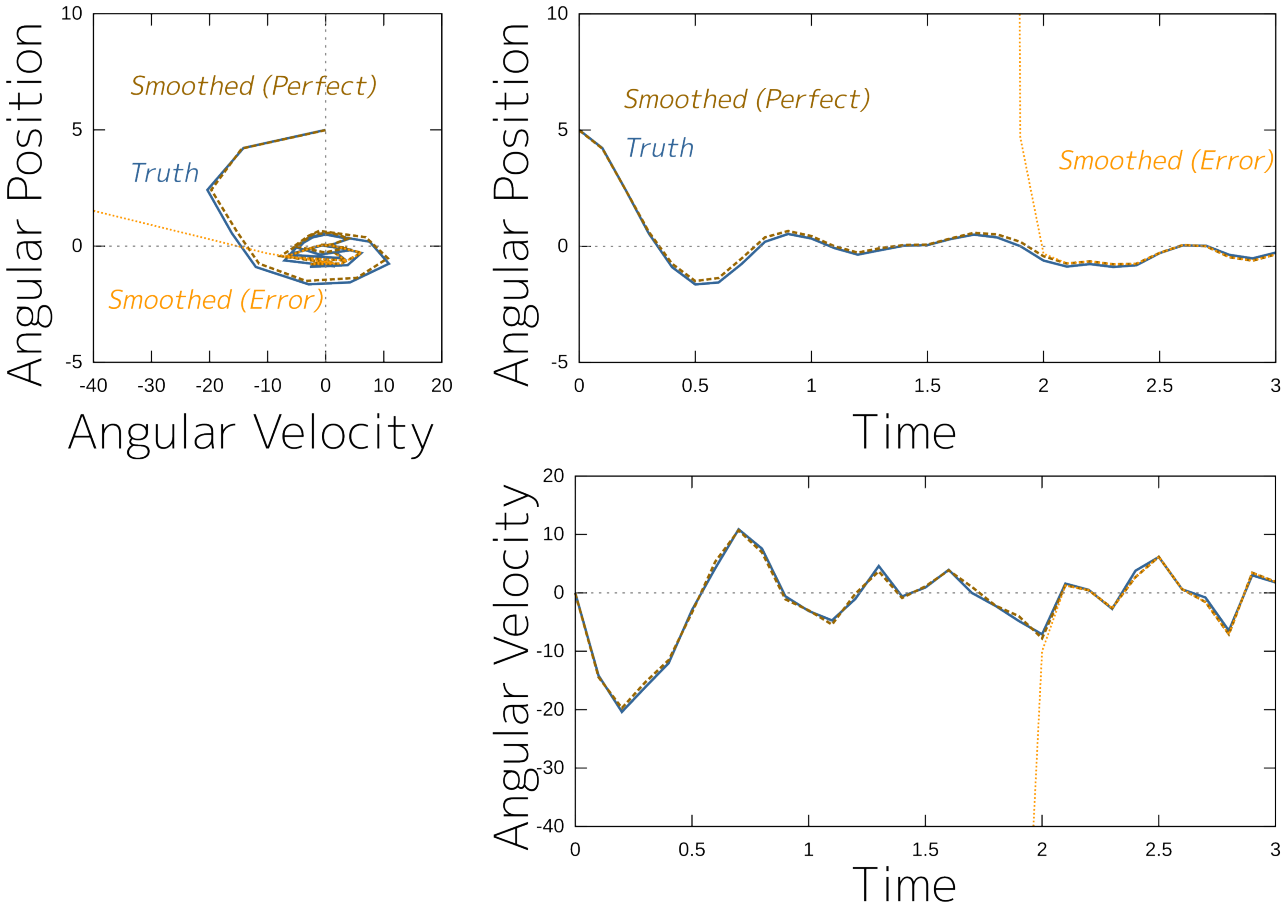

Figure 3.3.2: Random Error with Perfect Arithmetic

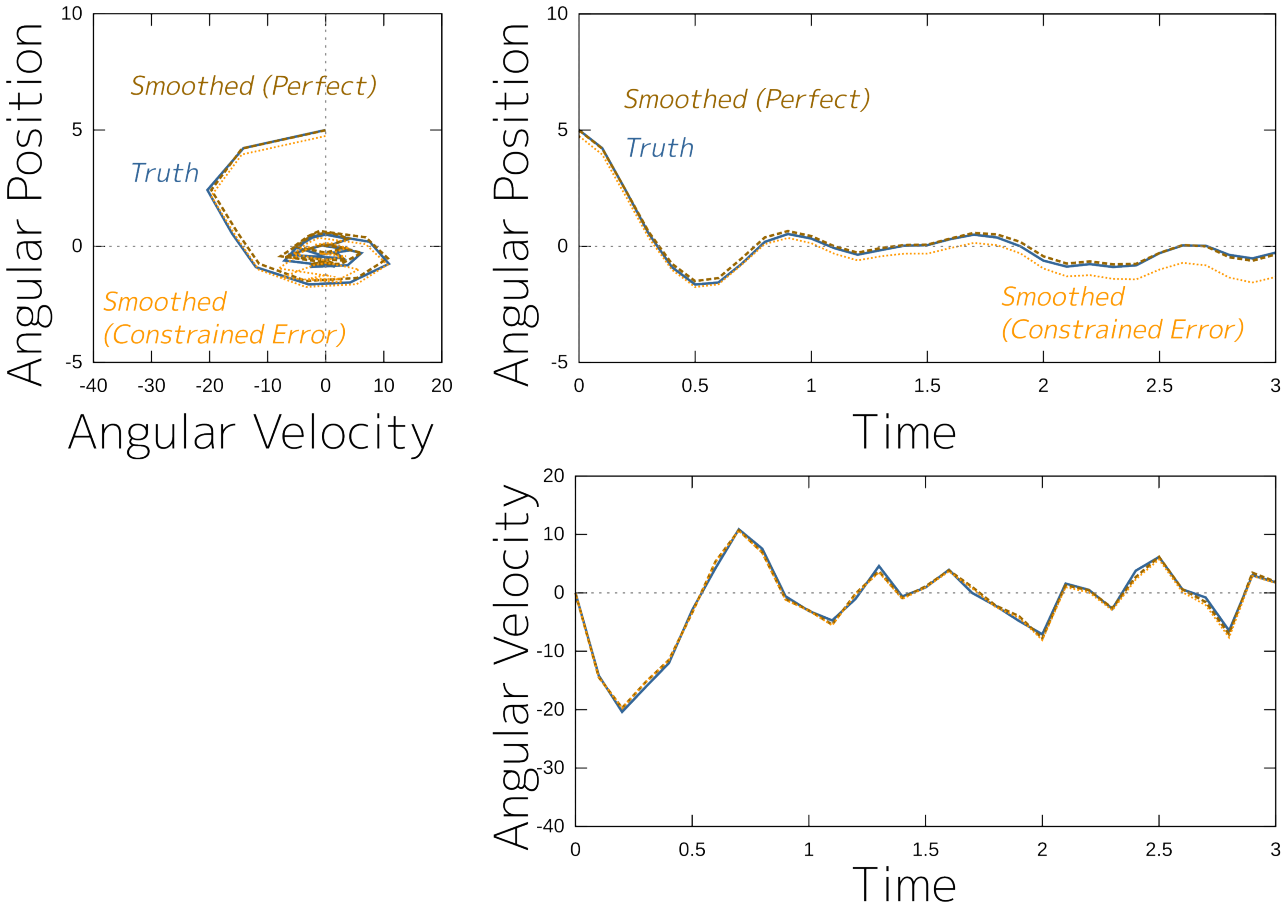

Figure 3.3.3: Constrained Error with Perfect Arithmetic

Given a solution at time k which is potentially distorted by error, $\begin{bmatrix}\tilde{x}\\ \rho\end{bmatrix}_{k|\eta}$, we can simply delete the $\tilde{\rho}$ entry and then re-solve for it using Equation 2.3.3, $\tilde{\rho}_{k|\eta} = P_{k|k}^{-1}\left(\tilde{x}_{k|\eta} - x_{k|k-1}\right)$, this new pair may have error introduced but they should meet the constraint and should behave like Figure 3.3.3 instead of Figure 3.3.2. Substituting this into the back iterate equation, simplifying, and noting that we don't need to solve for $\rho_{k-1|\eta}$ because we are going to throw it away at the next step anyway, we obtain,

$$\begin{aligned}
x_{k-1|\eta} &= \begin{bmatrix}F_k^{-1} & -F_k^{-1}\mathcal{Q}_k\end{bmatrix}\begin{bmatrix}x_{k|\eta}\\ P_{k|k-1}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right)\end{bmatrix} + \begin{bmatrix}-F_k^{-1}u_k\end{bmatrix}\\
&= F_k^{-1}x_{k|\eta} - F_k^{-1}\mathcal{Q}_k P_{k|k-1}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right) - F_k^{-1}u_k\\
&= \left(F_k^{-1} - F_k^{-1}\mathcal{Q}_k P_{k|k-1}^{-1}\right)\left(x_{k|\eta} - x_{k|k-1}\right) + F_k^{-1}x_{k|k-1} - F_k^{-1}u_k\\
&= \left(F_k^{-1} - F_k^{-1}\mathcal{Q}_k P_{k|k-1}^{-1}\right)\left(x_{k|\eta} - x_{k|k-1}\right) + F_k^{-1}\left(x_{k|k-1} - u_k\right)\\
&= \left(F_k^{-1} - F_k^{-1}\mathcal{Q}_k P_{k|k-1}^{-1}\right)\left(x_{k|\eta} - x_{k|k-1}\right) + x_{k-1|k-1}\\
&= \left(F_k^{-1}P_{k|k-1} - F_k^{-1}\mathcal{Q}_k\right)P_{k|k-1}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right) + x_{k-1|k-1}\\
&= \left(F_k^{-1}P_{k|k-1}F_k^{-\mathrm{T}} - F_k^{-1}\mathcal{Q}_k F_{k-1}^{-\mathrm{T}}\right)F_k^{\mathrm{T}}P_{k|k}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right) + x_{k-1|k-1}\\
&= \left(F_k^{-1}\left(P_{k|k-1} - \mathcal{Q}_k\right)F_k^{-\mathrm{T}}\right)F_k^{\mathrm{T}}P_{k|k-1}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right) + x_{k-1|k-1}\\
&= P_{k-1|k-1}F_k^{\mathrm{T}}P_{k|k-1}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right) + x_{k-1|k-1}
\end{aligned}$$

As we might expect from the discussion of Equation 3.2.2, the flipped reverse state/dual propagation, is very similar to the Kalman Update, Equation 2.2.12, and we can express the back iterate with a similar form[8], shown in Equation 3.4.1.

$$\begin{aligned}
x_{k-1|\eta} &= x_{k-1|k-1} + D_k\left(x_{k|\eta} - x_{k|k-1}\right)\\
D_k &= P_{k-1|k-1}F_k^{\mathrm{T}}P_{k|k-1}^{-1}
\end{aligned} \tag{3.4.1}$$

Using this form instead of the one mentioned previously we can achieve near perfect results, as shown in Figure 3.4.1.

When inverting $P_{k|k-1}$ is inconvenient, we do not need to make this correction at every step, only when the error gets bad enough that it needs to be corrected. In this example we can safely use Γ^{-1} for 5 iterations back before the compounding of the largest eigenvalue, $\lambda \approx 40$, gets too large. At this point

[8]This reverse iterate is shown in [2] where it is only used for a single step in the section on the *Interval Smoother*. The oversight that it applies in general can be forgiven.

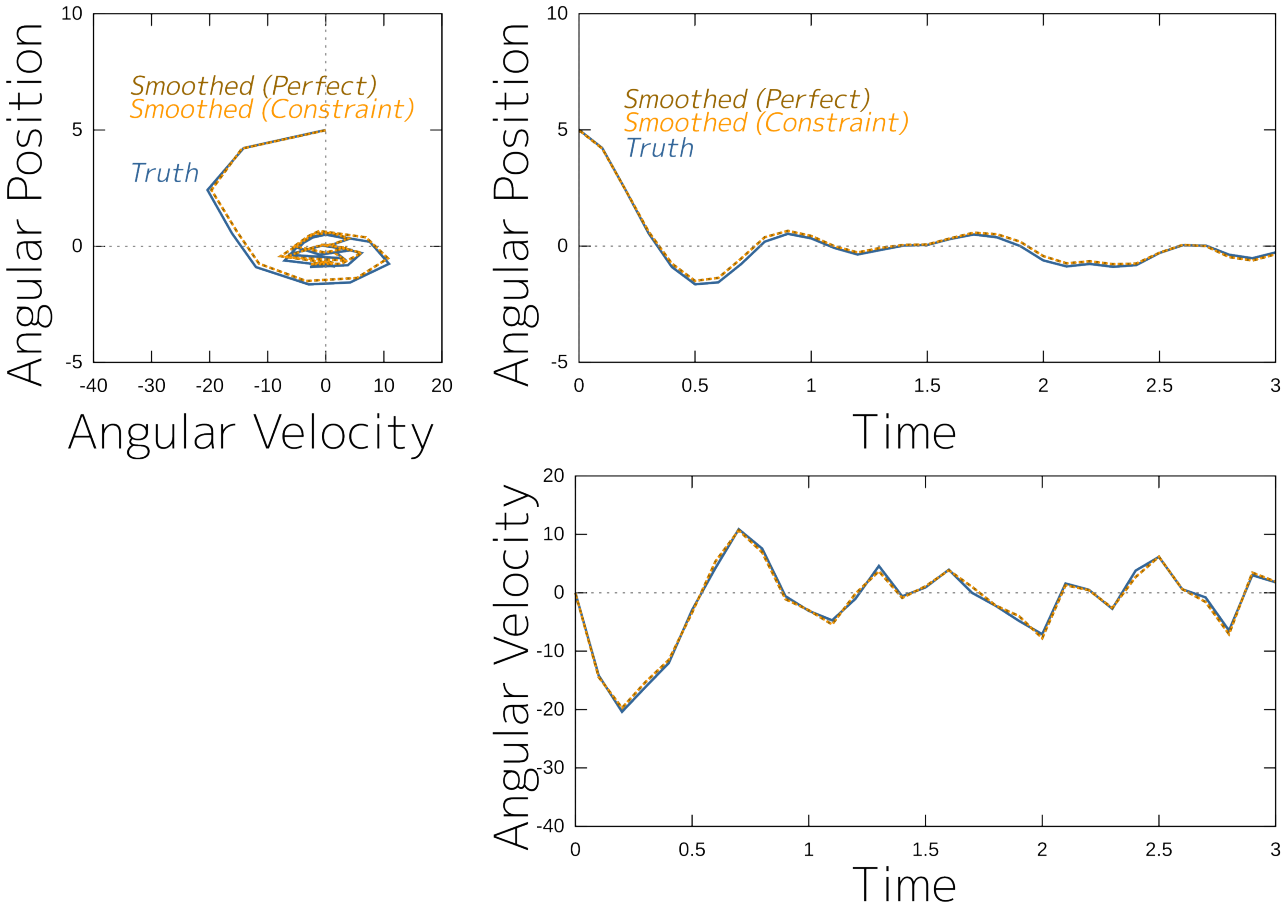

Figure 3.4.1: Constraint Based Smooth

we can apply the constraint as done before, clear the $\rho_{k|\eta}$ state and reset it based on $x_{k|\eta} - x_{k|k-1}$. The advantage is that to form Γ^{-1} we only need F^{-1}, which we could easily find in this problem ahead of time[9], and only every five steps would we need to invert the covariance matrix $P_{k|k-1}$. However, knowing that 5 iterations is an appropriate amount assumes a lot of knowledge about the problem, so instead we could track an error covariance matrix, S_k, through the transform and wait to apply the constraint until a diagonal entry gets larger than a threshold, ϵ, as shown in Equation 3.4.2. This process, with $\epsilon = 1E-20$ and thresh $= 1E-4$ applies the constraint every 4 steps for the present problem.

$$\begin{gathered} S_\eta = \varepsilon I \\ S_{k-1} = \Gamma_k^{-1} S_k \Gamma_k^{-\mathrm{T}} \\ \text{if } \max\left(\operatorname{diag}\left(S_k\right)\right) \geq \text{threshold then } \rho_{k|\eta} = P_{k|k-1}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right) \\ \text{and } S_k = \varepsilon I \end{gathered} \tag{3.4.2}$$

Instead of applying the constraint by reseting the dual state we could treat it as an observation. We know that the state and dual must meet the constraint $P_{k|k-1}^{-1}\left(x_{k|\eta} - x_{k|k-1}\right) = \rho_{k|\eta}$ or alternatively a *state*, the vector/dual combination, $\begin{smallmatrix} x_{k|\eta} \\ \rho_{k|\eta} \end{smallmatrix}$ is observed $P_{k|k-1}^{-1} x_{k|k-1} = B\left(\begin{smallmatrix} x_{k|\eta} \\ \rho_{k|\eta} \end{smallmatrix} + \epsilon\right)$, where ϵ is some nondescript Gaussian noise $\epsilon \sim N\left(0, vI\right)$.

$$\begin{aligned} B_k &= \begin{bmatrix} P_{k|k-1}^{-1} & -I \end{bmatrix} \\ P_{k|k-1}^{-1} x_{k|k-1} &= B_k \left(\begin{bmatrix} x \\ \rho \end{bmatrix}_{k|\eta} + \epsilon \right) \\ \begin{bmatrix} x \\ \rho \end{bmatrix}_{k|\eta} &= \begin{bmatrix} \tilde{x} \\ \rho \end{bmatrix}_{k|\eta} + B_k^\dagger \left(P_{k|k-1}^{-1} x_{k|k-1} - B_k \begin{bmatrix} \tilde{x} \\ \rho \end{bmatrix}_{k|\eta} \right) \\ B_k^\dagger &= B_k^{\mathrm{T}} \left(B_k B_k^{\mathrm{T}}\right)^{-1} \end{aligned} \tag{3.4.3}$$

We can use the error covariance we were back propagating in Equation 3.4.2 to construct an inverse based on the principles of Chapter 2 to balance the strengths of both the state, x, and dual, ρ, instead of simply relying on only the state, when we reset ρ based only on x and the constraint.

$$B_k^\dagger = S_k B_k^{\mathrm{T}} \left(B_k S_k B_k^{\mathrm{T}}\right)^{-1} \tag{3.4.4}$$

This in turn gives a covariance update equation.

[9] In any case where F is created by integrating a continuous time matrix, as in this problem, we could easily substitute $-\Delta t$ in the integration to find F^{-1} with the same effort as finding F.

Strategy	Error in $x_{0\|\eta}$
Reset Dual at Every Step	≈ 0
Reset Dual Every 5 Steps	$1.9E-8$
Reset Dual According to Eq. 3.4.2 ~ 4 steps	$4.8E-11$
Pseudoinverse from Eq. 3.4.3 ~ 4 steps	$2.1E-10$
Observation and Covariance Eq. 3.4.4 and 3.4.5 ~ $3-4$ steps	$4.4E-12$

Table 3.1: Numerical Comparisons for the Weather-vane Problem

$$S_k = \left(I - B_k^{\dagger} B_k\right) S_k + \epsilon I \tag{3.4.5}$$

In cases where computation time is not an issue, sticking to the simpler method of inverting $P_{k|k-1}$ and resetting the dual, as in Equation 3.4.1, at every step is probably the best way to go. In problems where we computational time is critical we have presented strategies to reduce the need to invert the matrix $P_{k|k-1}$. If a lot is known about the problem it's possible that the simple alternative of a fixed reset rate, every 5 updates in this example, may be possible. When not enough is known or we want to automate the process we can use the covariance method described in Equation 3.4.2. With regard to the numerical performance we could any of the alternative strategies presented in Equations 3.4.3, 3.4.4, and 3.4.5 and obtain similar outcomes. The comparison of these methods for this problem are shown in Table 3.1.

Chapter 4

Smoothed Solution Convergence in the Blind Tricyclist Problem

"The complexities of cause and effect defy analysis."

- Douglas Adams, *Dirk Gently's Holistic Detective Agency*

All of the methods discussed in the previous two chapters only apply to linear systems. Many systems of interest fail to meet the linear criteria so from now on we will be exploring strategies to apply what we know about linear theory to nonlinear systems. This is an idea we have, already implicitly put into practice in Chapter 2. The equation for the motion of the weather-vane of the last chapter, 3.1.1, is likely derived from the nonlinear damped pendulum equation,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\omega^2 \sin(x_1 - w) - 2\zeta\omega x_2 \end{bmatrix}.$$

This transformation equation is, perhaps, the most common example of the concept of linearization, where we assume that the system is approximately linear about some point. Here we make the *small angle assumption*, simply assume that $x_1 - w \approx 0$, and take the first two terms of the Taylor polynomial about that point to form the linear equation used in the last chapter. In that example the linearization happens to occur about a critical point, which is also an attractor with any nonzero damping factor making this assumption eventually true[1]. For the following problem this simple idea will have to be expanded.

[1]For more discussion on this concept I recommend [16], Chapter 2 Section 9 which describes the Simple Pendulum of which the weather-vane is an example.

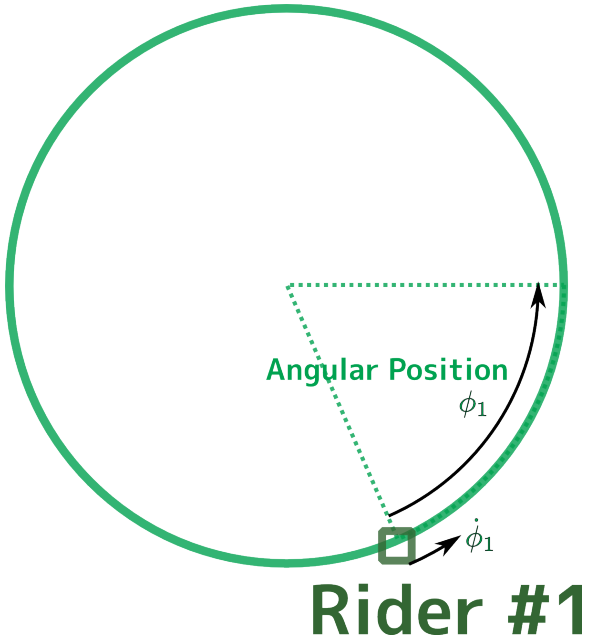

Figure 4.1.1: Merry-Go-Round State

4.1 The Blind Tricyclist Problem

To demonstrate various strategies of handling nonlinear problems we will be using the the following exemplar from [10]. In this problem we are tasked with tracking the xy-position of a tricyclist in an amusement park and as a nuisance we will also need to track the position of two people riding merry-go-rounds and other states of the tricycle. With absolute certainty we know the center of each merry-go-round and their unique radii. The merry-go-rounds rotate at a constant speed which is not known. The location of each merry-go-round rider will be tracked as two values, their angular position on the merry-go-round and its angular velocity, the speed of the merry-go-round, as shown in Figure 4.1.1. When needed their 2D location can be constructed from this and the knowns. The location of the tricyclist is tracked as (x, y) and the angle the tricycle is facing, as shown in Figure 4.1.2. The state vector for the system is, $s = \begin{bmatrix} x & y & \theta & \phi_1 & \phi_2 & \dot{\phi}_1 & \dot{\phi}_2 \end{bmatrix}$, where (x, y) is the location of the tricycle, θ is the facing angle of the tricycle.

The propagation of the tricycle position is a source of the nonlinearity. The tricyclist turns the front wheel to some angle, γ, relative to the direction the tricycle is facing and pedals forward with some velocity, v. Without noise the movement mechanic is shown in Figure 4.1.3 and described as follows, where $\text{cinc}(x) = \frac{\cos(x)-1}{x}$ and b is the wheel base length,

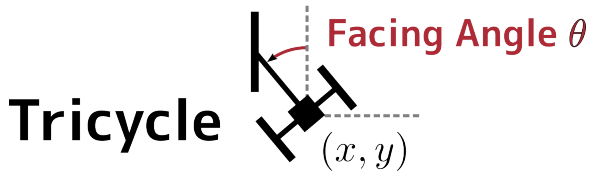

Figure 4.1.2: Tricycle State

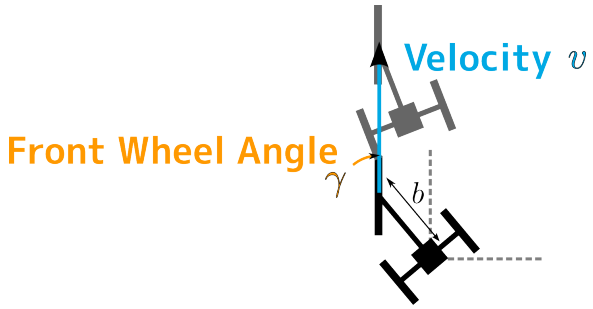

Figure 4.1.3: Tricycle Movement

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{k+1} = \begin{bmatrix} x_k + v\Delta t \left(\sin(\theta_k) \operatorname{cinc}\left(\frac{v\Delta t \tan(\gamma)}{b}\right) + \cos(\theta_k) \operatorname{sinc}\left(\frac{v\Delta t \tan(\gamma)}{b}\right)\right) \\ y_k + v\Delta t \left(\sin(\theta_k) \operatorname{sinc}\left(\frac{v\Delta t \tan(\gamma)}{b}\right) - \cos(\theta_k) \operatorname{cinc}\left(\frac{v\Delta t \tan(\gamma)}{b}\right)\right) \\ \theta_k + \frac{v\Delta t \tan(\gamma)}{b} \end{bmatrix}.$$

In addition to these two known known components of the forcing function, ν and γ, there are 5 noise variables, two noise states for the two controls, $\tilde{v} = v + w_1$ and $\tilde{\gamma} = \gamma + w_2$, and another 3 for the tricycle's state, $\tilde{x} = x + \Delta t w_3$, $\tilde{y} = y + \Delta t w_4$, and $\tilde{\theta} = \theta + \Delta t w_5$. This gives us our total propagation function shown in Equation 4.1.1. The final movement of the tricycle with a specific instance of propagation noise is shown in Figure 4.1.5. The propagation noise and true initial condition will be fixed, thus this path will be constant for all future simulations.

$$
\begin{aligned}
\begin{bmatrix} x \\ y \\ \theta \\ \phi_1 \\ \phi_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix}_{k+1} &= f(s_k, w_k) = \begin{bmatrix} x_k + \tilde{v}_k \Delta t \left(\sin(\theta_k)\,\mathrm{cinc}(a_k) + \cos(\theta_k)\,\mathrm{sinc}(a_k)\right) + \Delta t w_{k,3} \\ y_k + \tilde{v}_k \Delta t \left(\sin(\theta_k)\,\mathrm{sinc}(a_k) - \cos(\theta_k)\,\mathrm{cinc}(a_k)\right) + \Delta t w_{k,4} \\ \theta_k + a_k + \Delta t w_{k,5} \\ \phi_1 + \Delta t \dot{\phi}_1 \\ \phi_2 + \Delta t \dot{\phi}_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \\
a_k &= \frac{\tilde{v}_k \Delta t \tan(\tilde{\gamma}_k)}{b} \\
\tilde{v}_k &= v_k + w_{k,1} \\
\tilde{\gamma}_k &= \gamma_k + w_{k,2} \\
\mathrm{sinc}(x) &= \frac{\sin(x)}{x} \\
\mathrm{cinc}(x) &= \frac{1 - \cos(x)}{x}
\end{aligned}
\tag{4.1.1}
$$

The state of the system is observed by the tricycle rider who makes relative bearing measurements to the merry-go-round riders, the angle between the direction the tricycle is facing and one of the two riders of the merry-go-round, as shown in Figure 4.1.5 and in Equation4.1.2. Additionally this measurement structure is periodic with the measurement made to each merry-go-round rider being made every 6 time increments, with the two measurements staggered, one measurement every 3 steps, alternating. Summarizing the observation protocol,

$$
\begin{aligned}
y &= h_i(s_k) + v_{i,k} \\
&= \arctan\left(m_{i,y} + r_i \sin(\phi_{i,k}) - y_k - d\sin(\theta_k),\, m_{i,x} + r_i \cos(\phi_{i,k}) - x_k - d\cos(\theta_k)\right) - \theta_k + v_i \\
m_1 &= \begin{bmatrix} 0 & -15 \end{bmatrix} \text{ is the center of merry-go-round 1} \\
m_2 &= \begin{bmatrix} 2 & 15 \end{bmatrix} \text{ is the center of merry-go-round 2} \\
r_1 &= 7.5 \text{ is the radius of merry-go-round 1} \\
r_2 &= 6.5 \text{ is the radius of merry-go-round 2} \\
&\text{Periodic Measurments } \begin{bmatrix} - & 1 & - & - & 2 & - \end{bmatrix} \text{ where } - \text{indicates no measurement, and 1 or 2 represents respecti}
\end{aligned}
\tag{4.1.2}
$$

All the mechanics have been defined so far leaving only statistical quantities which we have taken from the *large initial uncertainty* case in [10] and shown in Equation 4.1.3. It is important to note that for all

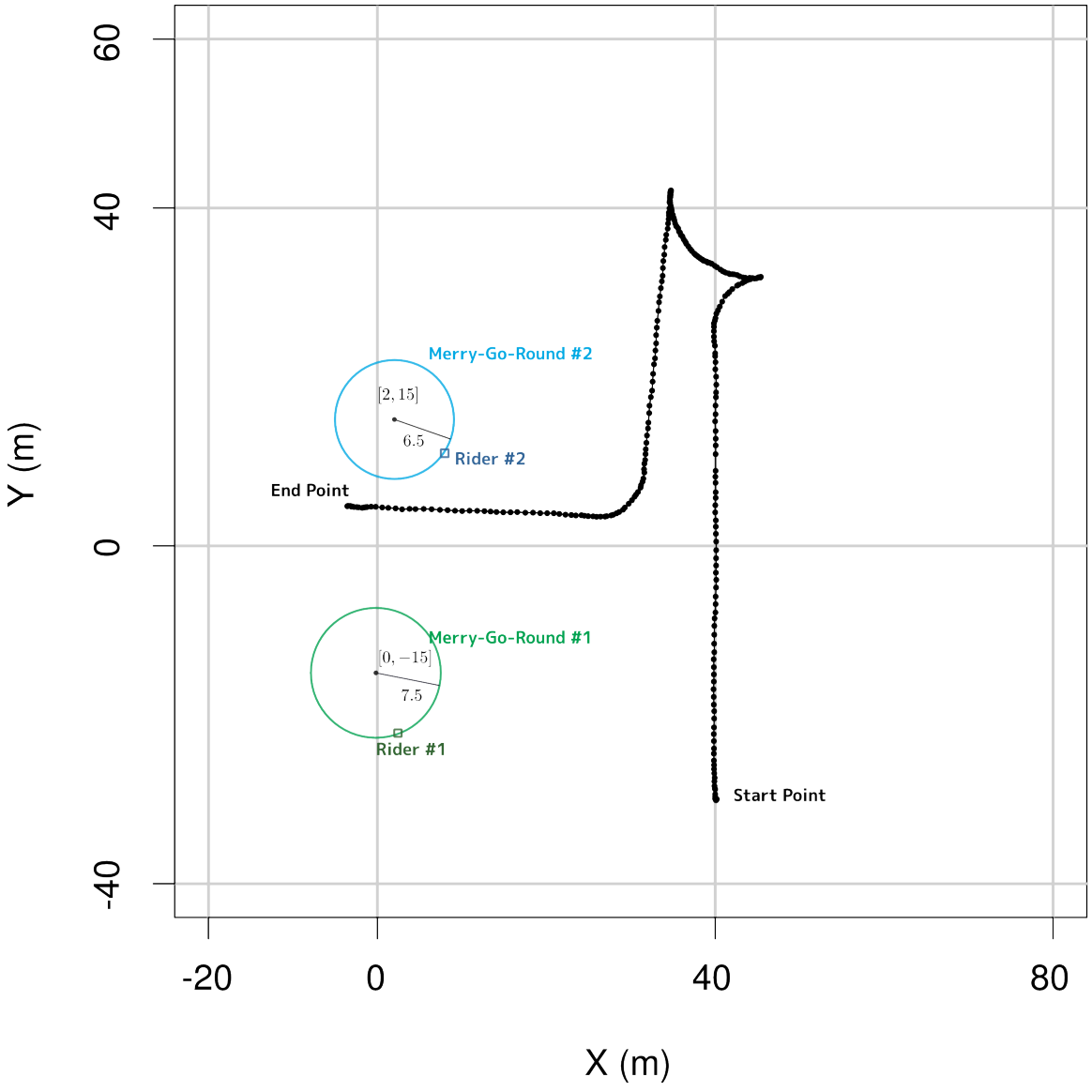

Figure 4.1.4: Tricycle's Movement

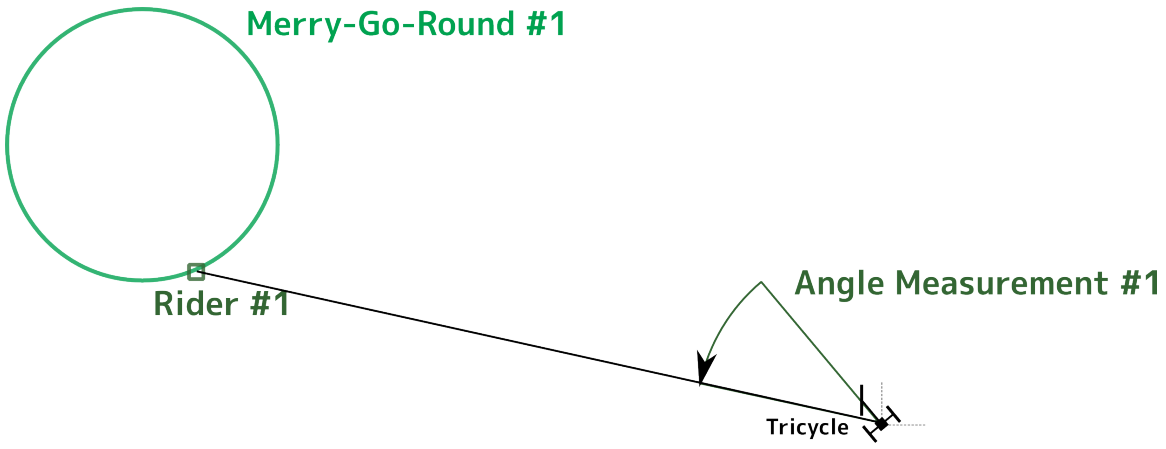

Figure 4.1.5: Tricycle Bearing Measurement

discussions only $\hat{s}_0$ and v will be generated randomly for each run, as opposed to w, which was generated once, and s_0, which was fixed. This means the actual path of the tricycle is fixed and we can compare how different realizations of the observations of this path, $\hat{s}_0$ and v, compare.

$$\begin{aligned} s_0 &= \begin{bmatrix} 40.197 & -30.097 & \frac{\pi}{2} & 5.1191 & 5.6913 & 0.1257 & -0.0898 \end{bmatrix} \\ \hat{s}_0 &\sim N\left(s_0, \text{Diag}\begin{bmatrix} (18.75)^2 & (18.75)^2 & \left(\frac{5}{8}\pi\right)^2 & \left(\frac{5}{6}\pi\right)^2 & \left(\frac{5}{6}\pi\right)^2 & (1.857\text{e-}2)^2 & (1.857\text{e-}2)^2 \end{bmatrix}\right) \\ w &\sim N\left(0, \text{Diag}\begin{bmatrix} (0.238)^2 & (1.9363\text{e-}3)^2 & (7.94\text{e-}2)^2 & (7.94\text{e-}2)^2 & (1.701\text{e-}3)^2 \end{bmatrix}\right) \\ v &\sim N\left(0, \text{Diag}\begin{bmatrix} (1.745\text{e-}2)^2 & (1.164\text{e-}2)^2 \end{bmatrix}\right) \end{aligned} \tag{4.1.3}$$

Given the truth data we can linearize all the functions, f and h, in the problem about the truth to create an *estimate* of the problem[2].

[2]We have a solution, in the form of the KF, for linear problems but have not explored nonlinear ones. Instead of attempting to solve the nonlinear one as state we will solve a linear problem which approximates the nonlinear one, namely some linearization of the original. We know our solution is optimal for this approximate problem and we hope/assume that the two problems are similar enough that the solutions are approximately the same.

$$
\begin{aligned}
s_k &= f_k\left(s_{k-1}, w_k\right) \\
&\approx F_k s_{k-1} + L_k w_k + u_k \\
F_k &= \left.\frac{\partial f_k(s,w)}{\partial s}\right|_{s=s_{k-1|\text{truth}}, w=w_{k|\text{truth}}} \\
L_k &= \left.\frac{\partial f_k(s,w)}{\partial w}\right|_{s=s_{k-1|\text{truth}}, w=w_{k|\text{truth}}} \\
u_k &= f\left(s_{k-1|\text{truth}}, w_{k|\text{truth}}\right) - F_k s_{k-1|\text{truth}} - L_k w_{k|\text{truth}}
\end{aligned}
\tag{4.1.4}
$$

$$
\begin{aligned}
z_k &= h_k\left(s_k, v_k\right) \\
y_k &\approx H_k s_k + J_k v_k \\
H_k &= \left.\frac{\partial h_k(s,v)}{\partial s}\right|_{s=s_{k-1|\text{truth}}, v=v_{k|\text{truth}}} \\
V_k &= \left.\frac{\partial h_k(s,v)}{\partial v}\right|_{s=s_{k-1|\text{truth}}, v=v_{k|\text{truth}}} \\
y_k &= z_k - \left(h\left(s_{k|\text{truth}}, v_{k|\text{truth}}\right) - H_k s_{k|\text{truth}} - J_k v_{k|\text{truth}}\right)
\end{aligned}
\tag{4.1.5}
$$

Because this new problem is linear and now meets the form suggested in Chapter 2 we can use a simple Kalman Filter solution derived there. Our scenario has only a linear dependence on v and all the other variables are fixed, so one linearization will be valid for all our realizations of the problem. Additionally the covariance terms are also independent of our specific realization, meaning we can calculate them once we have the linearizations, and from this we can bound performance on Root Mean Squared Error (RMSE) for this *estimated* problem[3]. Because we originally were interested in tracking only the tricycist's 2D position, x, y, the RMSE 2D location error, xy error, is shown in Figure 4.1.6.

Creating a realization of the problem, that is generating the values v and $\hat{s}_0$, we can see how the Kalman Filter performs in Figure 4.1.7. Looking at this and 8 other realizations in Figure 4.1.8 we can begin to form a picture of how the filter performs in general.

Instead of just looking at individual cases we can leverage the ability to create many realizations of this set up and compare their errors statistically. We generate 100 runs to generate robust statistics and we plot 4 important metrics, the MSE of the filter as a function of time because this is directly comparable to the covariance generated statistic, the two quantiles 50% (median) and 75% because they give us an idea how the majority of filters are performing, and the max error which gives us an idea how bad the filter could be. For the Kalman Filter (given perfect linearizations) this is shown in Figure 4.1.9. Just as we

[3] Again this, linear, problem is not the one we started with so its bounds on performance are not necessarily the same. Just as before we hope/assume that it well approximates the original.

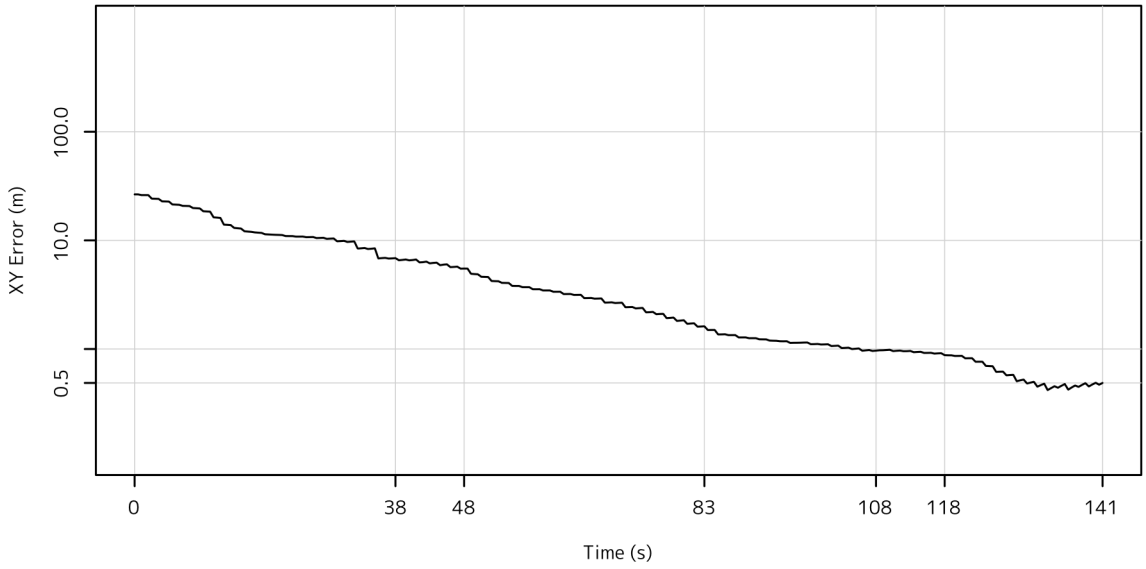

Figure 4.1.6: Expected Root Mean Squared Error in 2D Location

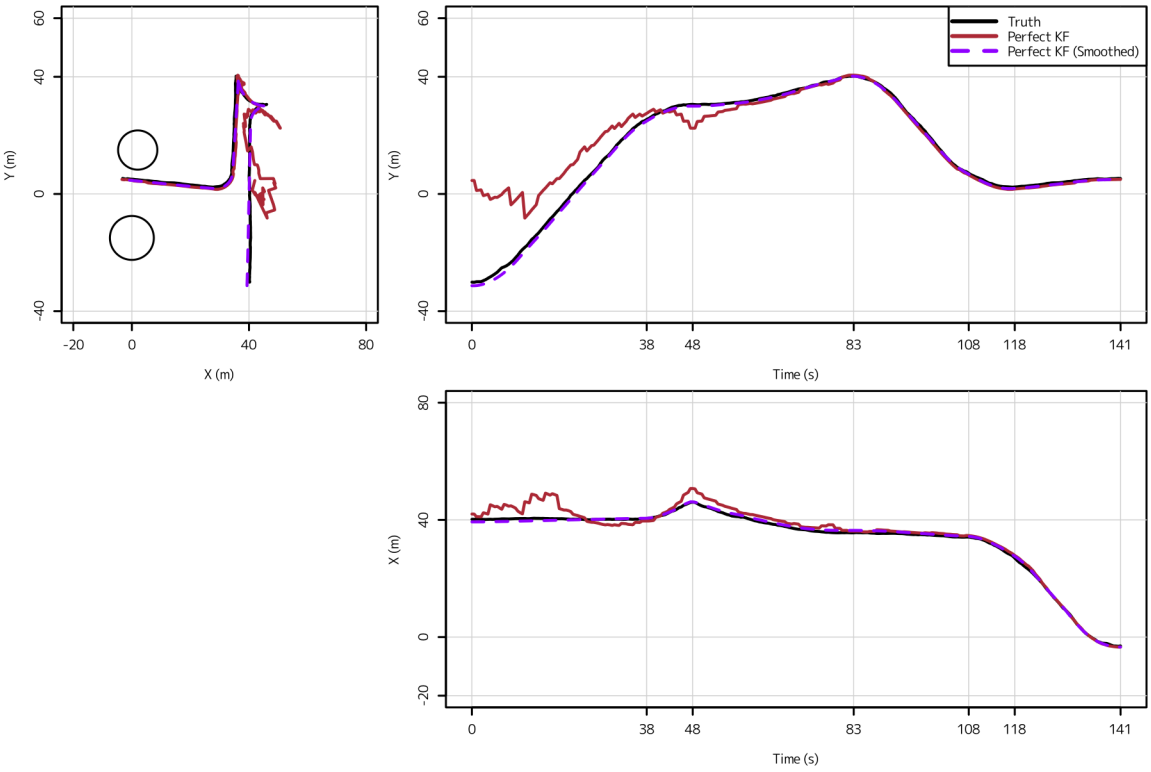

Figure 4.1.7: KF with Perfect Linearizations Example

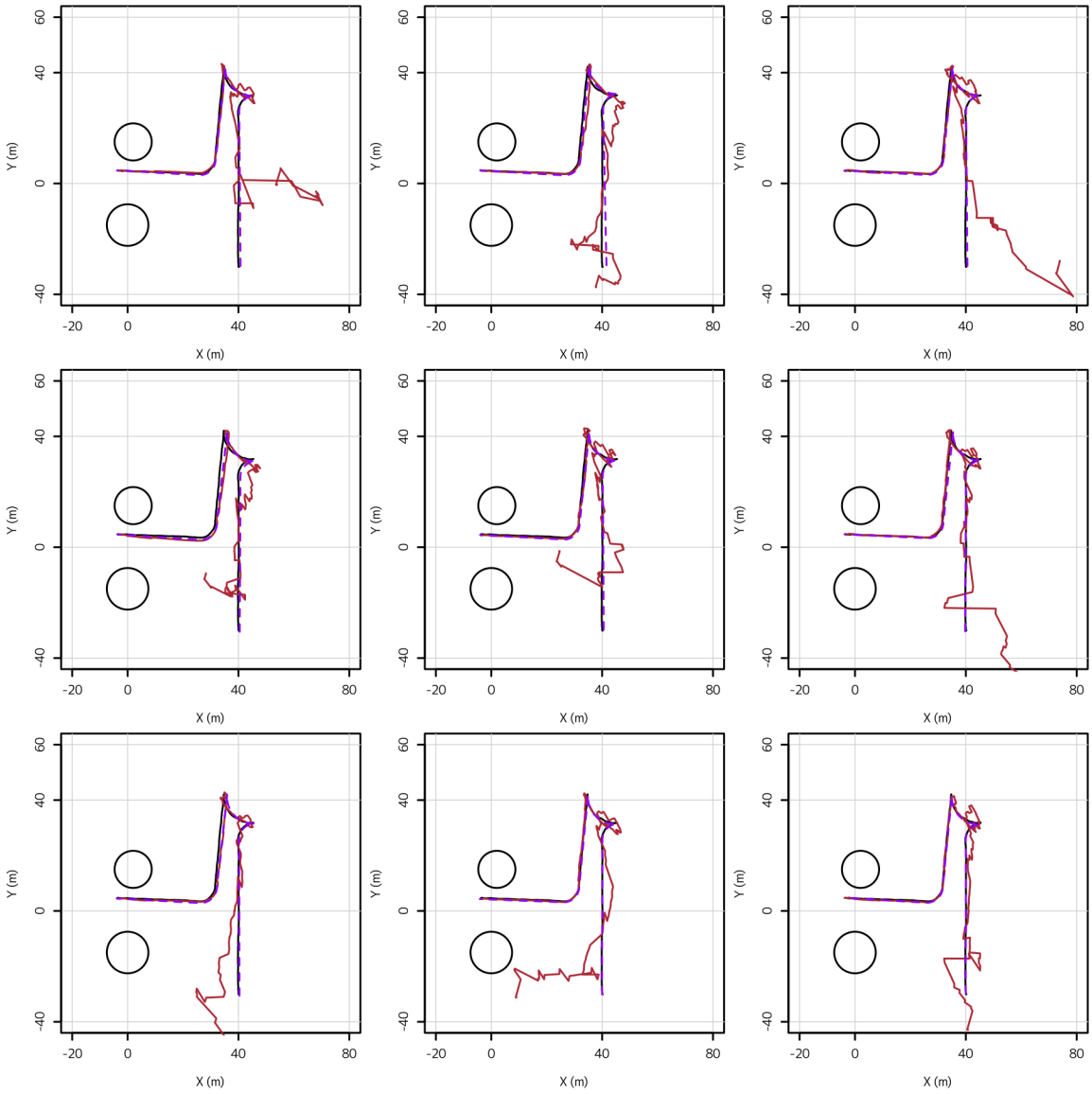

Figure 4.1.8: Other Examples of KF with Perfect Linearizations

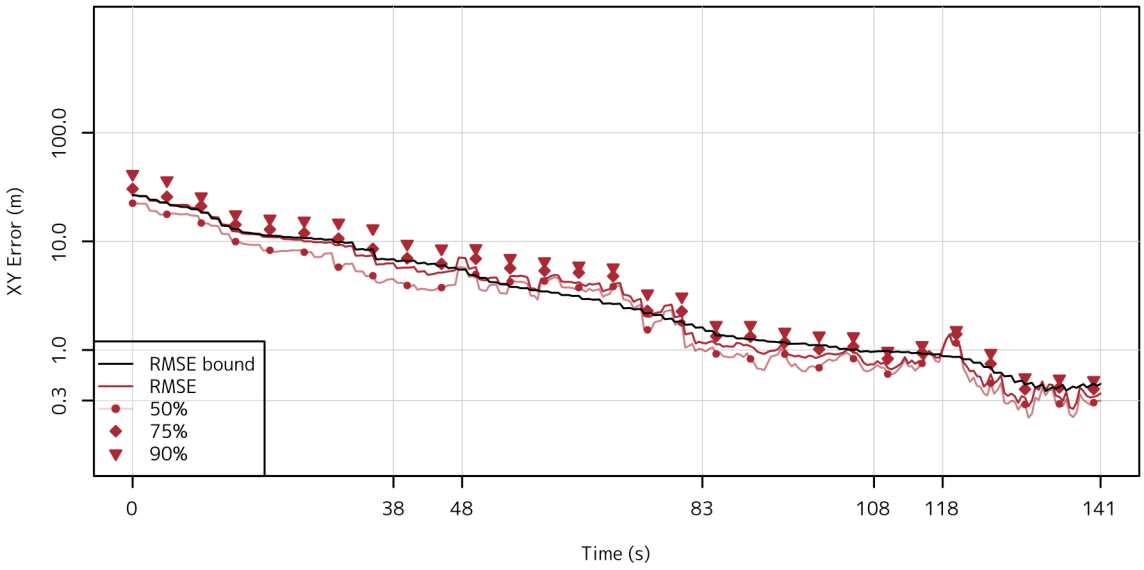

Figure 4.1.9: Perfect Linearization KF's Error

did in Chapter 3, we can smooth the KF's solution to form a complete, smoothed, estimate of the tricycles path[4], shown in the previous Figures 4.1.7 and 4.1.8 with its error performance shown in Figure 4.1.9.

4.2 Batch Least Squares

Before we work with the various more complicated filters let's start with a simple one that makes some rather extreme simplifications to the problem. We will start by assuming that there is no initial information, P_0, or propagation noise, w, which simplifies the problem to on of finding an initial state, s_0. The minimization problem reduces to the form,

$$\text{minimize} \left\| \begin{bmatrix} h\left(f\left(s_0\right)\right) \\ h\left(f\left(f\left(f\left(s_0\right)\right)\right)\right) \\ h\left(f\left(f\left(f\left(f\left(f\left(f\left(s_0\right)\right)\right)\right)\right)\right)\right) \\ \vdots \end{bmatrix} - y \right\|.$$

We can attempt to solve this problem using Gauss-Newton iterations[5] by linearizing the entire prob-

[4]Recall this path, the smoothed estimate, is an *estimate of a track* given all the data whereas the Kalman Filter's estimate is a *track of estimates*, a string of estimates each one an end point to the an estimated track given all the data up to that point in time.

[5]Notice that once linearized the problem is a simplified version of that suggested in Chapter 2, where $Q = 0$, $P_0 \to \infty$, and $R = I$, and each Gauss-Newton iteration is, in essence, a Kalman Filter run followed by a smoothing run back to find the entire solution.

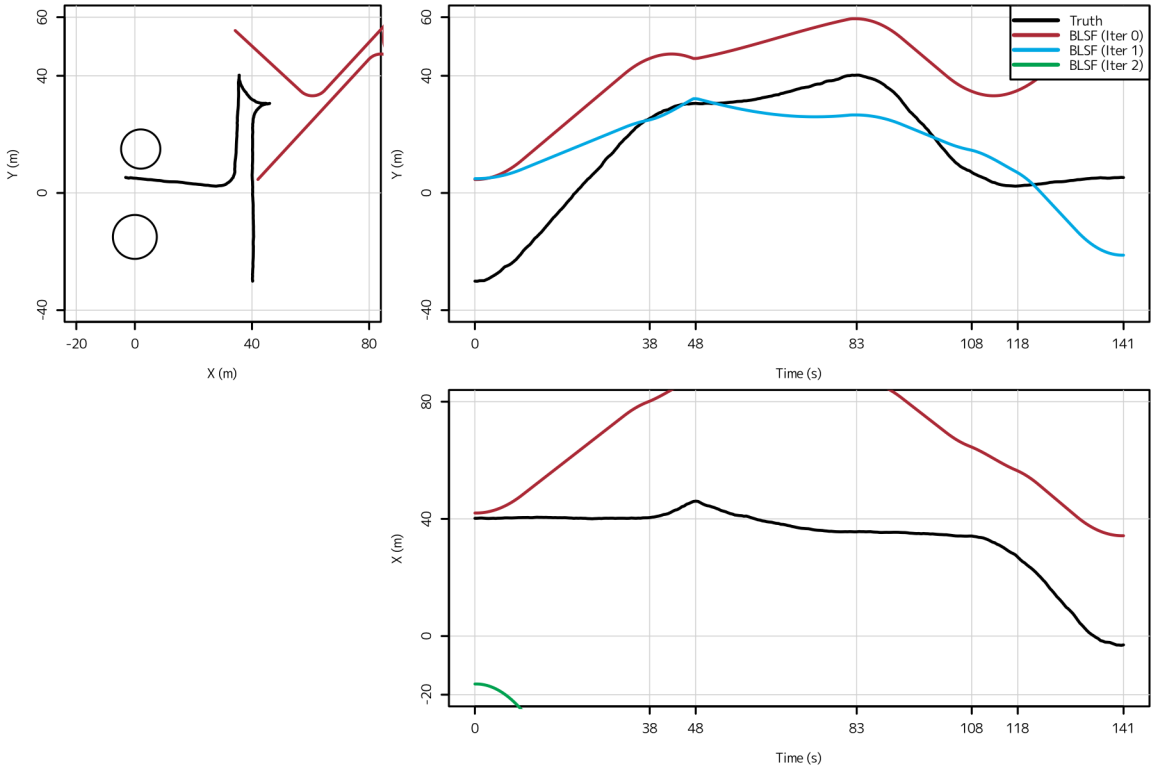

Figure 4.2.1: BLSF (Simple Implementation)

lem about expectations, as the Blind Tricyclist paper, [10], recommends. For a simple implementation of this solution, as suggested, within two iterations the solution for most realizations have left the bounds area completely. An example is shown in Figure 4.2.1 in which at step 2 the initial $[x, y]$ value was estimated to be $[-16.35, 3405]$. This is not that surprising given that there is a lot of error in the initial point which throws off the entire original track, the ITER 0 result.

The fact that a basic implementation of Gauss-Newton iterations did not perform well on an estimate of the actual problem is not that surprising. The most immediate fix would be to relax the iteration to be of the form $s_{0\{i+1\}} = \alpha H^{\dagger}_{\{i\}} y + (1 - \alpha)\, s_{0\{i\}}$. While this greatly helps, the method still does not perform anywhere near as well as what we might expect getting caught in a local minima as shown in Figure 4.2.2. This case is not unique in and 9 others are shown in Figure 4.2.3.

The results in [10] contradict this, reporting a RMSE, in situations very similar to ours, of 4.38. The Least Squares problem has many alternative algorithms and I can only assume a very complicated one was used, an idea that is backed up by the mean compute time reported in [10] of 197.94s, an astronomical number. I do not claim to have applied all possible least squares solution methods, just that the simplest

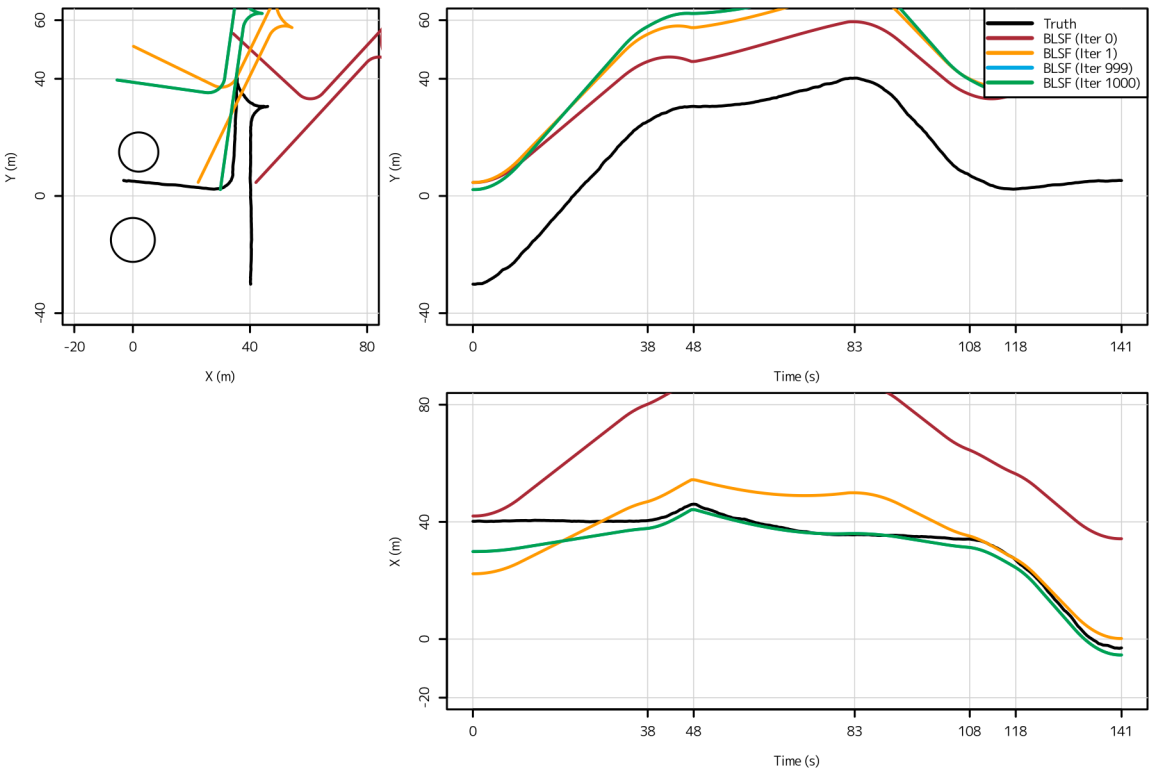

Figure 4.2.2: BLSF with Relax $\alpha = 0.15$

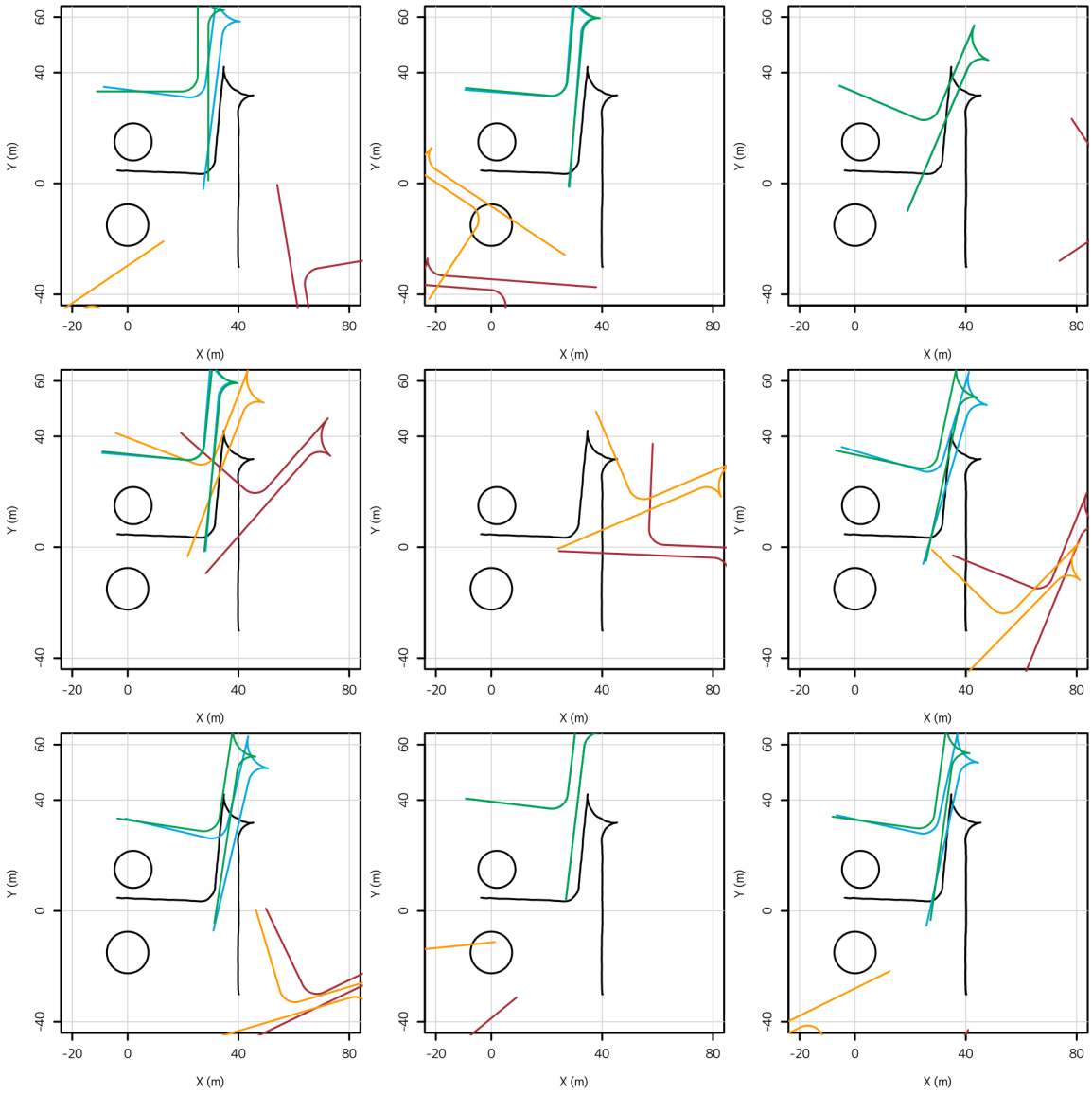

Figure 4.2.3: Other Examples of BLSF with Relax $\alpha = 0.15$

approach are unsuccessful.

4.3 Extended Kalman Filter and Smoothers

As we saw in the previous problem we can linearize the nonlinear problem about some point and solve the resulting estimation/smoothing problem using linear theory. In this way the Extended Kalman Filter (EKF) is an extension of the Kalman Filter to solve, approximately, the nonlinear problem. The Kalman Filter is a much more general framework than the BLSF we saw before, allowing a much more complete problem description, including process noise and initial estimate covariance. The second major advantage is that because the Kalman Filter is solved sequentially we can wait to linearize parts of the problem until we get to them, and at that time we can use the current estimate, $x_{k|k}$, instead of the one generated from the initial condition, $x_{k|0}$, as we did in the previous example. This process is described by Equations 4.3.1 and 4.3.2.

$$\begin{aligned}
x_k &= f_k\left(x_{k-1}, w_k\right) \\
&\approx F_k x_{k-1} + L_k w_k + u_k \\
F_k &= \left.\frac{\partial f_k(x,w)}{\partial x}\right|_{x=x_{k-1|k-1}, w=w_{k|k-1}} \\
L_k &= \left.\frac{\partial f_k(x,w)}{\partial w}\right|_{x=x_{k-1|k-1}, w=w_{k|k-1}} \\
u_k &= f\left(x_{k-1|k-1}, w_{k|k-1}\right) - F_k x_{k-1|k-1} - L_k w_{k|k-1}
\end{aligned} \tag{4.3.1}$$

$$\begin{aligned}
z_k &= h_k\left(x_k, v_k\right) \\
y_k &\approx H_k x_k + J_k v_k \\
H_k &= \left.\frac{\partial h_k(x,v)}{\partial x}\right|_{x=x_{k-1|k-1}, v=v_{k|k-1}} \\
V_k &= \left.\frac{\partial h_k(x,v)}{\partial v}\right|_{x=x_{k-1|k-1}, v=v_{k|k-1}} \\
y_k &= z_k - \left(h\left(x_{k|k-1}, n_{k|k-1}\right) - H_k x_{k|k-1} - J_k v_{k|k-1}\right)
\end{aligned} \tag{4.3.2}$$

The advantage of this is that as time goes on, and we hopefully get better and better estimates of the state, as we will introduce less and less linearization error. This behavior can be seen in Figure 4.3.1 where the EKF's track converges towards the truth, this is also reflected in other examples as show in Figure 4.3.2 and in general as shown in Figure 4.3.3. These plots are shown with the accompanying smoothed solution

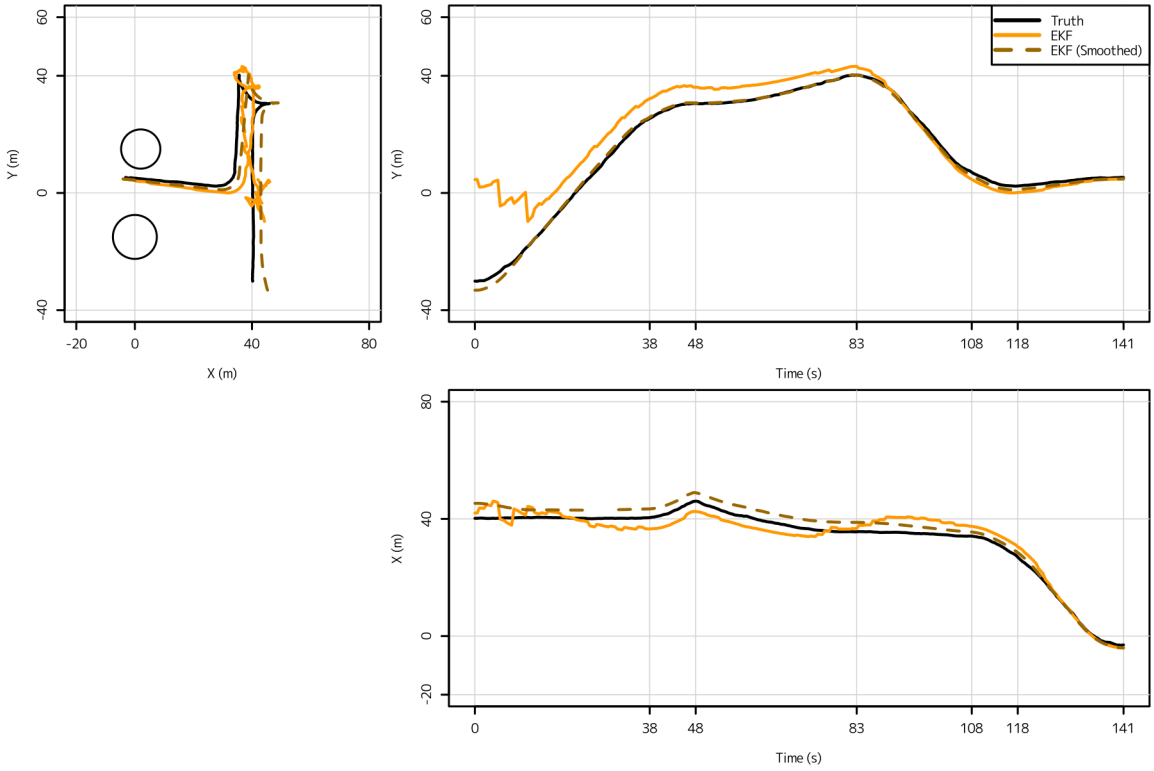

Figure 4.3.1: Example EKF Track

generated from Equation 3.4.1 on page 31, $x_{k-1|\eta} = x_{k-1|k-1} + P_{k-1|k-1} F_k^{\mathrm{T}} P_{k|k-1}^{-1} \left(x_{k|\eta} - x_{k|k-1} \right)$, using the EKF linearizations.

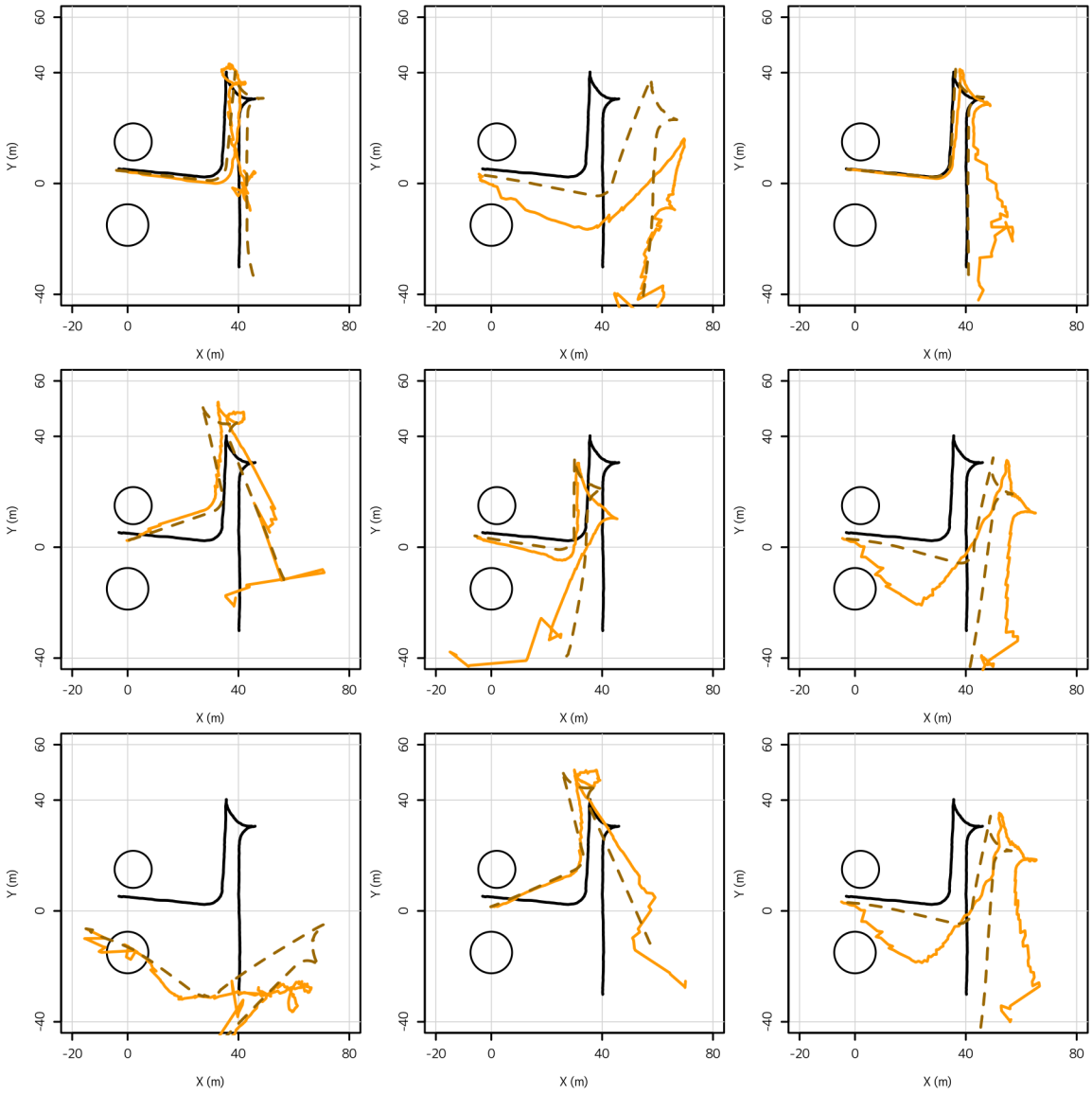

Figure 4.3.2: Other Examples of EKF Performance

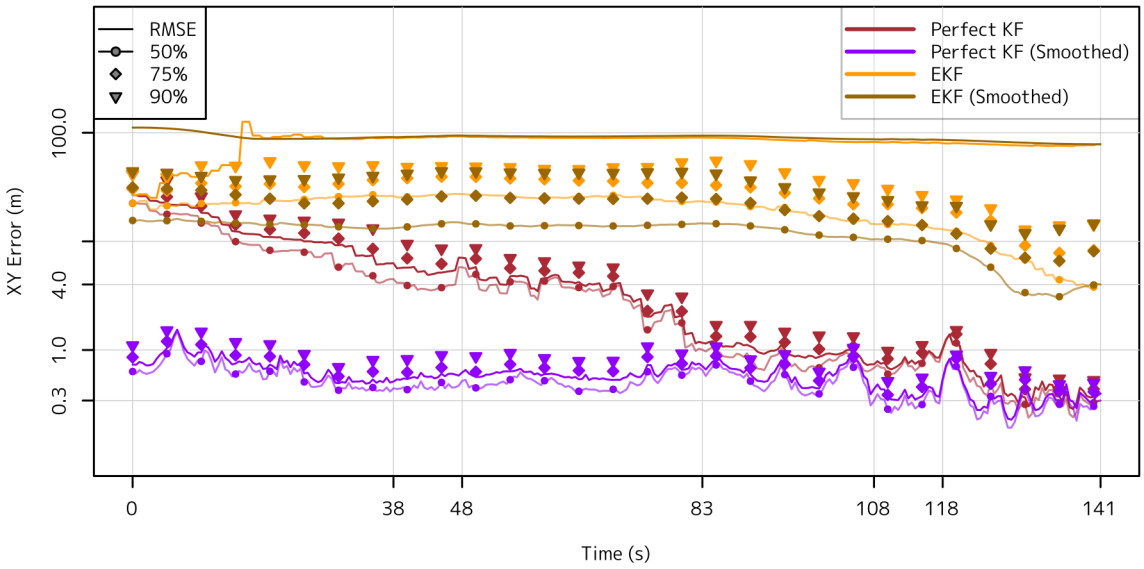

Figure 4.3.3: EKF Error Compare

4.3.1 Iterated Smoother

Concept

Recall from Equations 4.3.1 and 4.3.2 that when we ran the EKF we used the filter's current estimate of the state to linearize the problem. As we have seen these estimates have error and from the degradation in performance from the Perfect Linearization Kalman Filter to the EKF, shown in Figure 4.3.3, we can assume that these errors in linearization cause additional errors in the result. Notice that additionally the final smoothed path has less error than the original EKF run. The idea of the Iterated Smoother is to use this new, smoothed, result to relinearize the problem and hopefully get a still better result. To keep things straight we'll call the EKF's smoothed path $x_{k|\eta\{0\}}$, that is the state at time k given the data for all time, η, iteration $\{0\}$. Equations 4.3.3 and 4.3.4 describe the new linearization equations.

$$\begin{aligned}
x_k &= f_k\left(x_{k-1}, w_k\right) \\
&\approx F_{k\{1\}} x_{k-1} + L_{k\{1\}} w_k + u_{k\{1\}} \\
F_{k\{1\}} &= \left.\frac{\partial f_k(x,w)}{\partial x}\right|_{x=x_{k-1|\eta\{0\}}, w=w_{k|k-1}} \\
L_{k\{1\}} &= \left.\frac{\partial f_k(x,w)}{\partial w}\right|_{x=x_{k-1|\eta\{0\}}, w=w_{k|k-1}} \\
u_{k\{1\}} &= f\left(x_{k-1|\eta\{0\}}, w_{k|k-1}\right) - F_k x_{k-1|\eta\{0\}} - L_k w_{k|k-1}
\end{aligned} \tag{4.3.3}$$

$$\begin{aligned}
z_k &= h_k\left(x_k, v_k\right) \\
y_{k\{1\}} &\approx H_{k\{1\}} x_k + J_{k\{1\}} v_k \\
H_{k\{1\}} &= \left.\frac{\partial h_k(x,v)}{\partial x}\right|_{x=x_{k-1|\eta\{0\}}, v=v_{k|k-1}} \\
V_{k\{1\}} &= \left.\frac{\partial h_k(x,v)}{\partial v}\right|_{x=x_{k-1|\eta\{0\}}, v=v_{k|k-1}} \\
y_{k\{1\}} &= z_k - \left(h\left(x_{k|\eta\{0\}}, n_{k|k-1}\right) - H_k x_{k|\eta\{0\}} - J_k v_{k|k-1}\right)
\end{aligned} \tag{4.3.4}$$

Using these new linearizations we can run another Kalman Filter on the problem and hopefully do better than we did before. The new smoothed path may be a bit better but its hard to tell. Figure 4.3.5 gives a little more insight, some cases here really seem to be doing better which is confirmed in the error plot[6] of Figure 4.3.6, where although the end of the run is largely unchanged the majority of the run is doing better.

[6]Only the smoothed path is compared for error because it would be misleading to talk about the error in the estimate at time k given linearizations based on information for all time, η, but only using data up to time k.

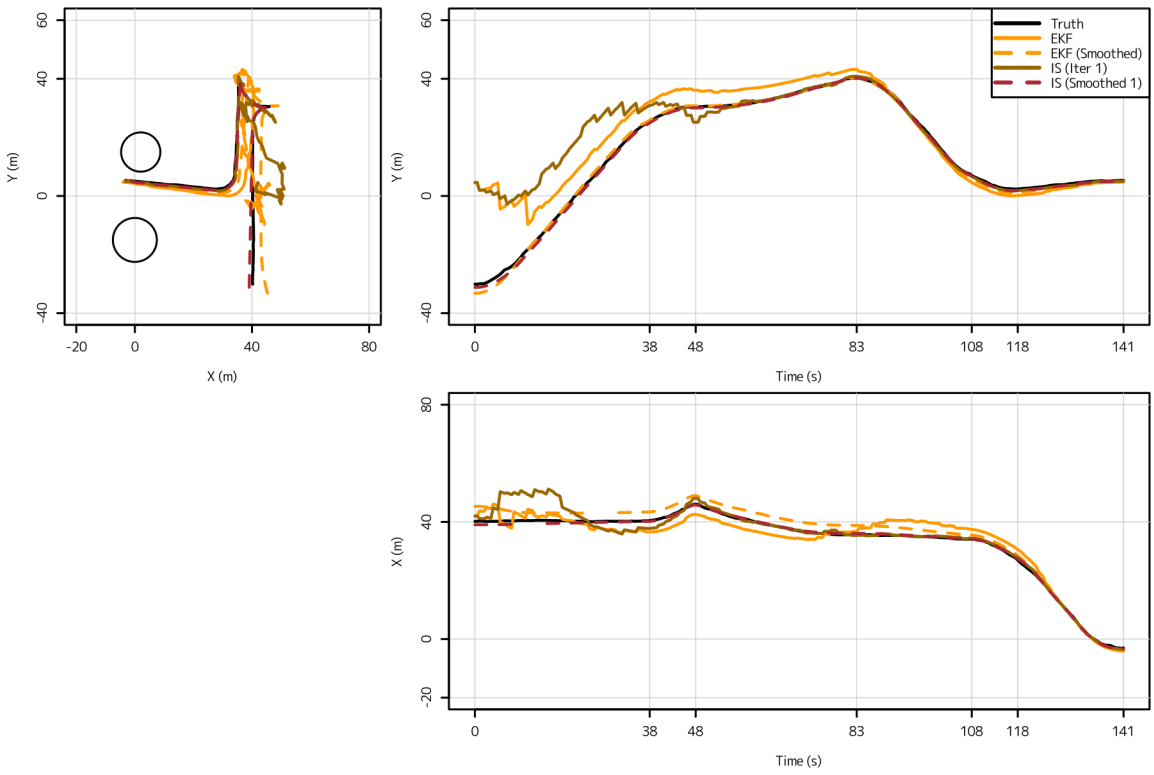

Figure 4.3.4: Iterated Smoother (1 Iterations)

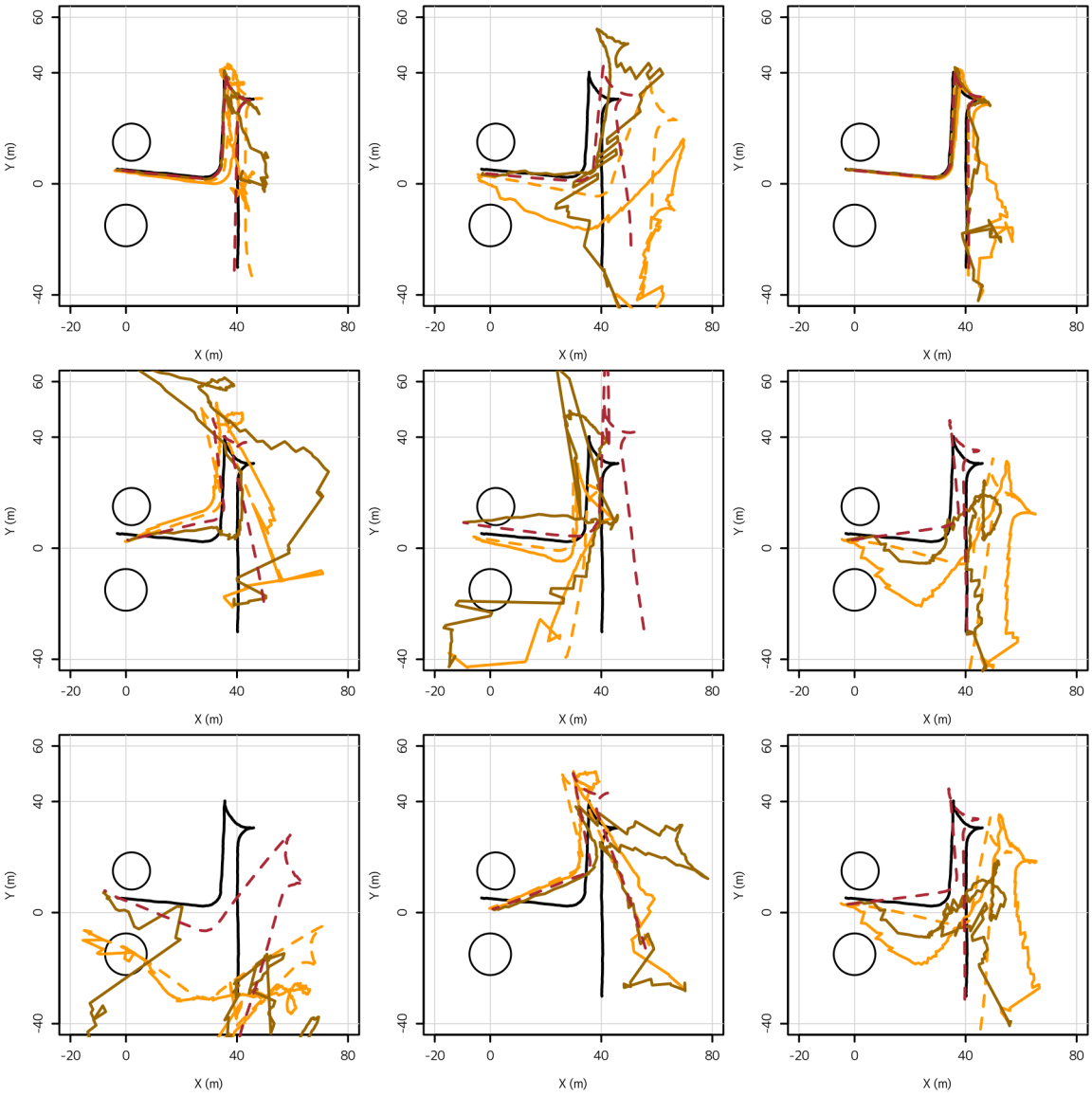

Figure 4.3.5: Iterated Smoother (1 Iterations) Other Examples

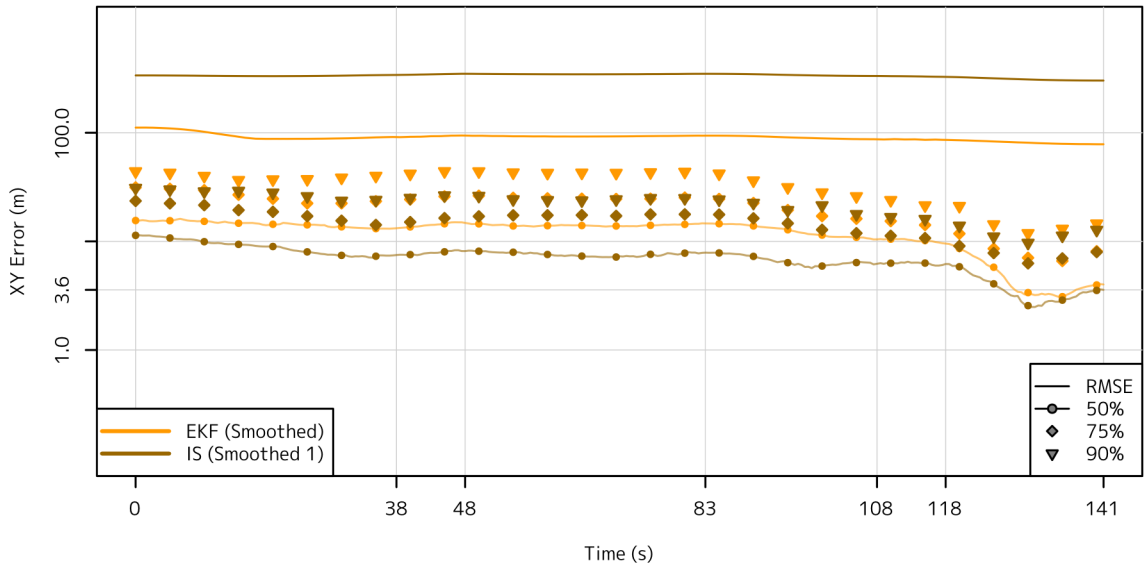

Figure 4.3.6: Iterated Smoother (1 Iterations) Error

Additional Iterations

This process can be repeated, this time the IS smoothed result for iteration i, labeled $x_{k|\eta\{i\}}$, is used to linearize the problem for the next iteration. The new linearization equations are given in 4.3.5 and 4.3.6. The result of this iterate is shown in Figure 4.3.7 where we can see the smoothed path for iterations 18 and 19 (20 total iterations counting the EKF as iteration 0). The improvement in this example is impressive, appearing to reach near Perfect Kalman Filter of Figure 4.1.7 on page 42. This result however does not appear in general as we can see in many of the examples in Figure 4.3.8 on page 57 and in the error plot, Figure 4.3.9 on page 58.

$$
\begin{aligned}
x_k &= f_k\left(x_{k-1}, w_k\right) \\
&\approx F_{k\{i+1\}} x_{k-1} + L_{k\{i+1\}} w_k + u_{k\{i+1\}} \\
F_{k\{i+1\}} &= \left.\frac{\partial f_k(x,w)}{\partial x}\right|_{x=x_{k-1|\eta\{i\}}, w=w_{k|k-1}} \\
L_{k\{i+1\}} &= \left.\frac{\partial f_k(x,w)}{\partial w}\right|_{x=x_{k-1|\eta\{i\}}, w=w_{k|k-1}} \\
u_{k\{i+1\}} &= f\left(x_{k-1|\eta\{i\}}, w_{k|k-1}\right) - F_k x_{k-1|\eta\{i\}} - L_k w_{k|k-1}
\end{aligned}
\tag{4.3.5}
$$

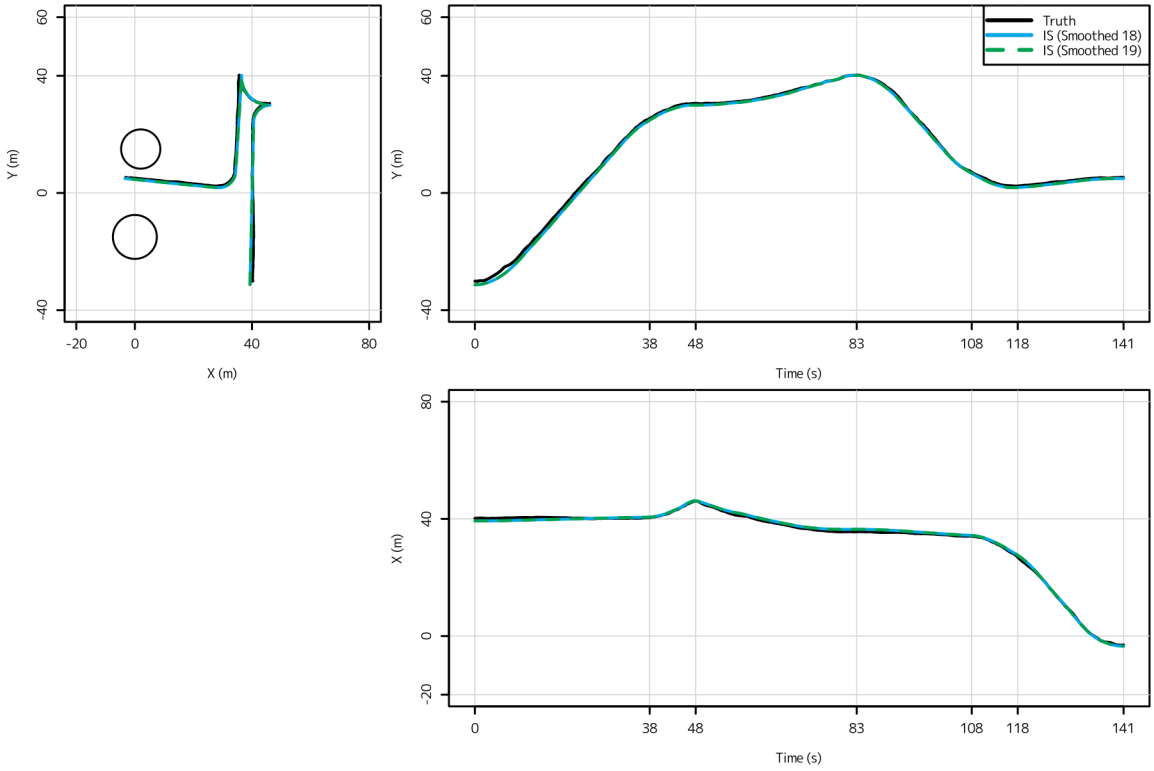

Figure 4.3.7: Iterated Smoother (18-19 Iterations)

$$
\begin{aligned}
z_k &= h_k\left(x_k, v_k\right) \\
y_{k\{i+1\}} &\approx H_{k\{i+1\}} x_k + J_{k\{i+1\}} v_k \\
H_{k\{i+1\}} &= \left.\frac{\partial h_k(x,v)}{\partial x}\right|_{x=x_{k-1|\eta\{i\}}, v=v_{k|k-1}} \\
V_{k\{i+1\}} &= \left.\frac{\partial h_k(x,v)}{\partial v}\right|_{x=x_{k-1|\eta\{i\}}, v=v_{k|k-1}} \\
y_{k\{i+1\}} &= z_k - \left(h\left(x_{k|\eta\{i\}}, n_{k|k-1}\right) - H_k x_{k|\eta\{i\}} - J_k v_{k|k-1}\right)
\end{aligned}
\tag{4.3.6}
$$

The immediate question is if this method is simply limited to some poor estimates just as the BLSF was. To test this we can start with the perfect linearizations as iteration $\{0\}$ and then continue iterating, with each iteration potentially introducing additional linearization error. This process will settle to some local minima which, other than being seeded by the truth, no longer utilizes the perfect linearizations. We can see in Figure 4.3.10 indications that there is a minima close to the perfect filter. The poor performance of the IS is simply the result of getting caught in a different, false, local minima not an inherent limitation.

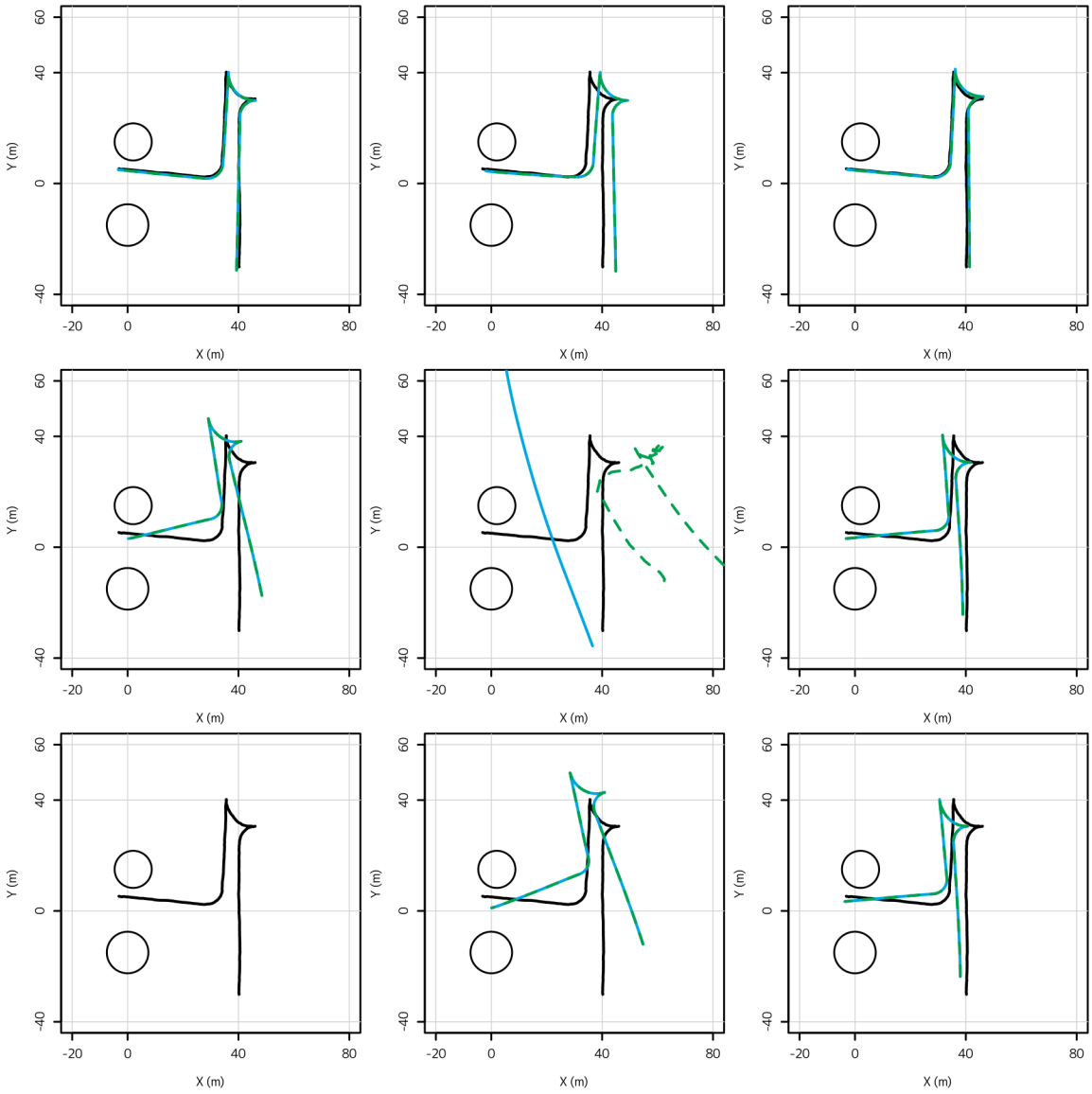

Figure 4.3.8: Iterated Smoother (18-19 Iterations) Other Examples

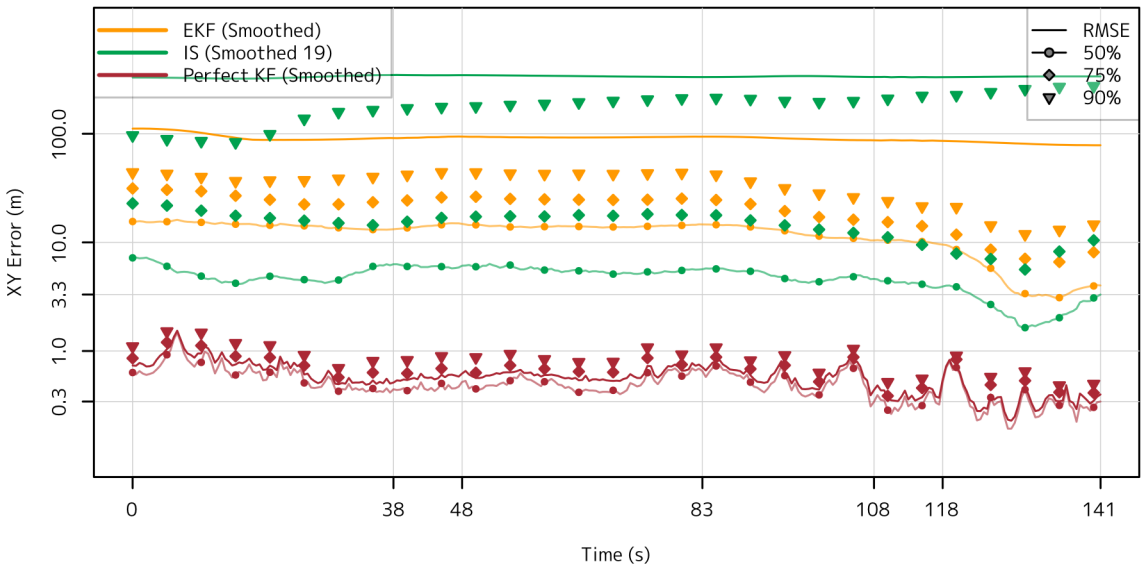

Figure 4.3.9: Iterated Smoother (19 Iterations) Error

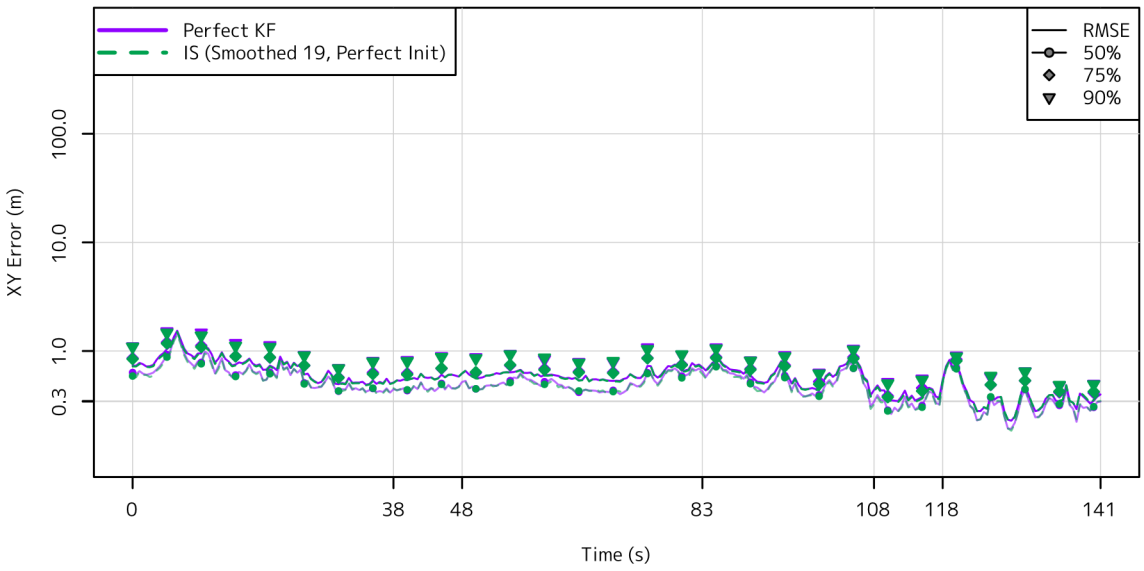

Figure 4.3.10: Iterated Perfect KF (19 Iterations) Error

4.3.2 Other Smoothers

BSEKF

If you notice in our definition of the Iterated Smoother we only used the smoothed paths estimate for $x_{k|\eta\{i\}}$ to relinearize the problem but there are two other quantities we could use, w and v. There is non-linear dependence of v in this problem so we will start by analyzing what would happen if we included estimates of w in our relinearization of the problem. This iterate is referred to as the Backward-Smoothing Extended Kalman Filter (BSEKF)[9]. To find the values of w we can use the Dual State information from Chapter 2, Equations 2.2.5 on page 12 and 2.3.3 on page 16, which simplifies into Equation 4.3.7.

$$w_{k|\eta\{i\}} = Q_k L_{k\{i\}}^{\mathrm{T}} P_{k|k-1\{i\}}^{-1} \left(x_{k|\eta\{i\}} - x_{k|k-1\{i\}} \right) \tag{4.3.7}$$

Just as the Iterated Smoother allowed us to use all of our information to correct linearizations based on the state this should allow us to correct linearizations based on w and improve performance. There is no improvement in performance for this example as shown in Figure 4.3.11, where the BSEKF performs slightly worse in most statistics. The actual cause of this failure to improve performance is bipartite. The first issue is that linearization of w has almost no impact on the performance of the filter which can be seen in Figure 4.3.12 where we run a KF linearized with and without truth values for w. This indicates that in this problem this strategy cannot have a very large impact.

Additionally and more fundamentally the estimates of w are almost always going to be useless. The only reason we can get a good estimate of x is because it's highly correlated over time, allowing us to leverage large periods of data to solve for an almost static quantity[7]. The process noise, w, on the other hand has little correlation across time. To know what w was at every step k we would need to have measured the state of the system accurately at *every*[8] time increment k. The inability to estimate w can be seen in Figure 4.3.13 where we form the estimates of $w_{k|\eta}$ using the Perfectly Linearized Kalman Filter (a best case scenario).

[7] The extreme version of this can be seen in the Batch Least Squares method where, ignoring w, we leverage all the data to actually solve a static quantity

[8] To illustrate this conundrum consider the following thought experiment. The system has one state which is simply a random walk $x_k = x_{k-1} + w_k$. Assume that $P_0 = 0$ and we only measure the state at time $k = 4$ perfectly, $v_4 = 0$. For simplicity assume that $x_4 = x_0 + a$. What we know from this is that $\sum_1^4 w_k = a$, but nothing else. We can identify what the most likely values are, $w_k = 1/a$, which might be better for linearization reasons but in all likelihood the linearizations made at expected value are going to be pretty similar. The expected value linearizations would be wrong if the values for w were something more like $[0.3a, 1.2a, -0.7a, 0.2a]$ which is a sequence that we will never guess.

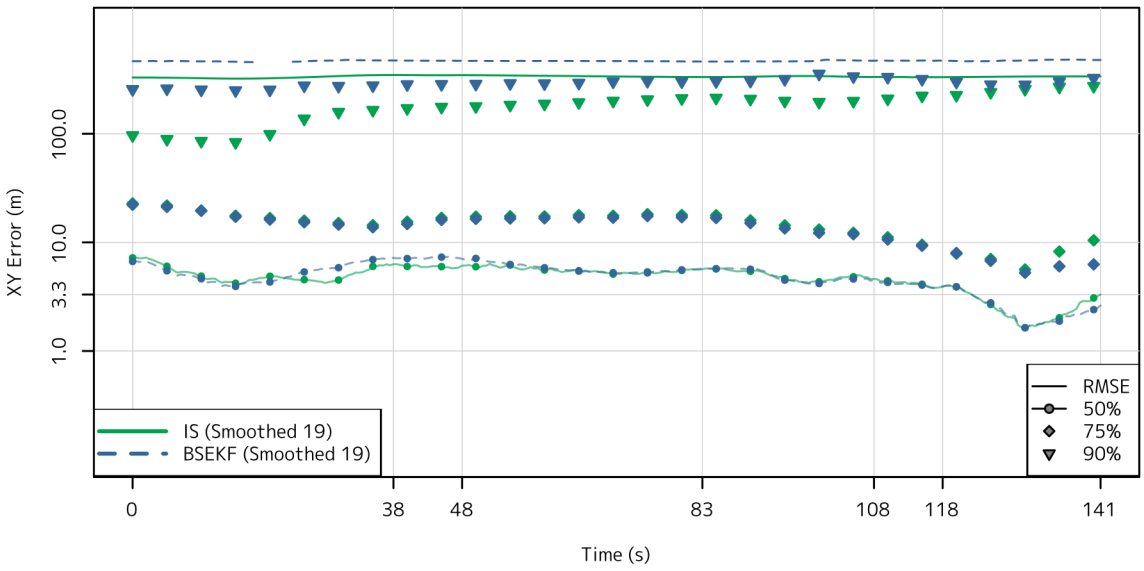

Figure 4.3.11: BSEKF (19 Iterations) Error

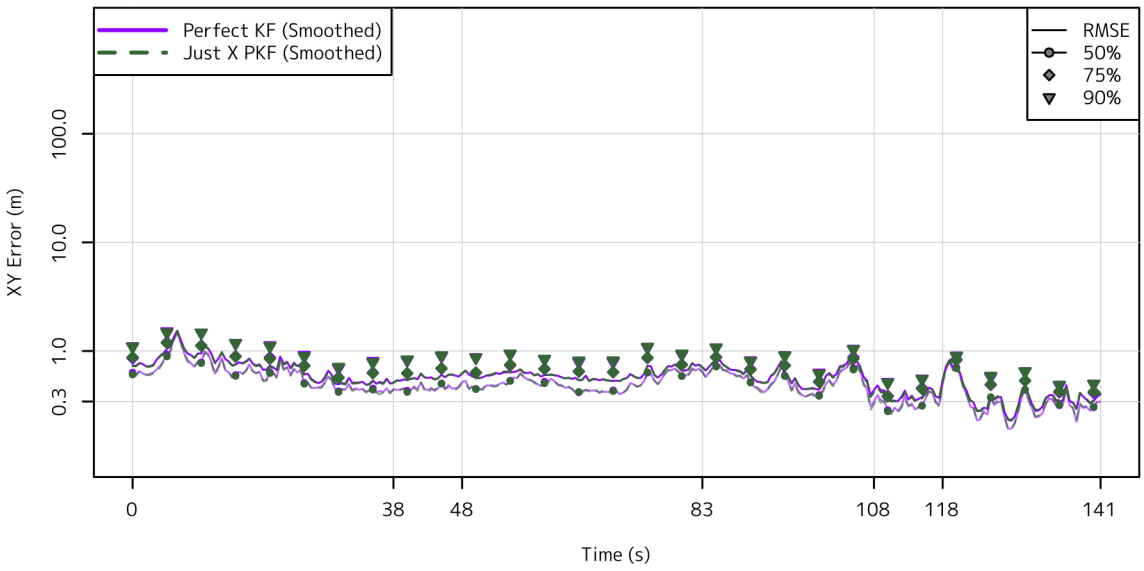

Figure 4.3.12: Perfect KF (with and without w) Error

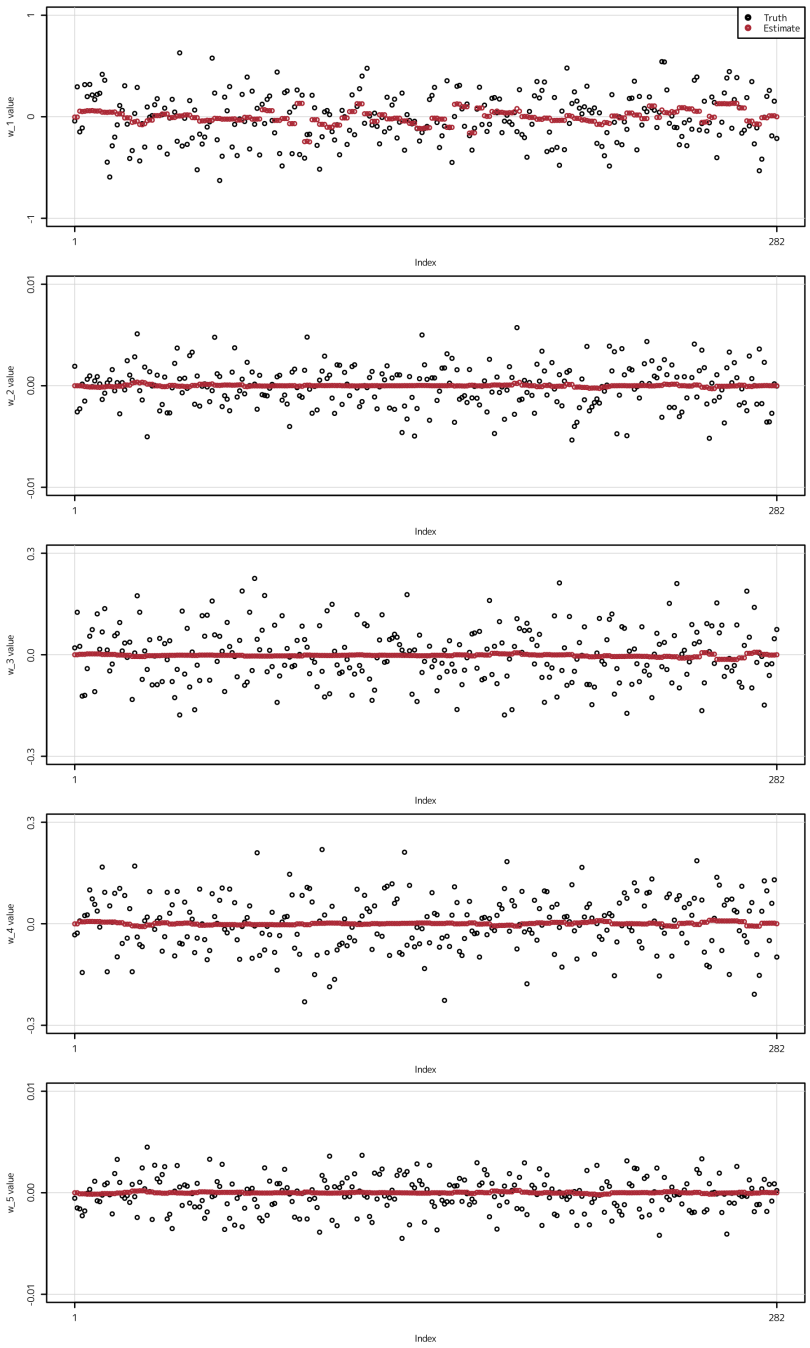

Figure 4.3.13: Estimates of w from the Perfect KF

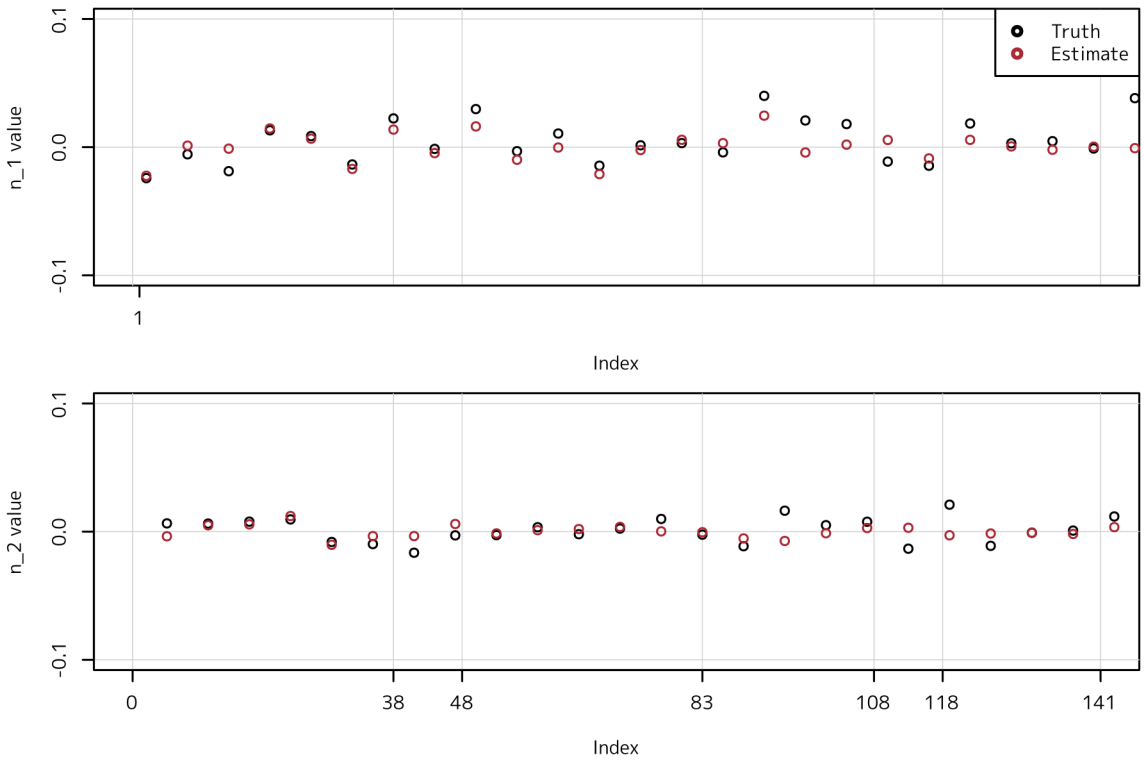

Figure 4.3.14: Estimates of v

Relinearizing v

The values of the observation noise, v, present another random variable we can relinearize with respect to. This iterate, show in Equation 4.3.8, is formed from the two Equations 2.2.6 and 2.2.5 on page 12. Unlike our estimate of w we have some hope of estimating v because its dimensionality is approximately the same as the observation. We can see our ability to estimate v in Figure 4.3.14, and extrapolate that if it had a nonlinear impact on the observation we could correct some of it using our estimate.

$$v_{k|\eta\{i\}} = R_k J_{k\{i\}}^{\mathrm{T}} \left(J_{k\{i\}} R_k J_{k\{i\}}^{\mathrm{T}} \right)^{-1} \left(y_{k\{i\}} - H_{k\{i\}} x_{k|\eta\{i\}} \right) \tag{4.3.8}$$

The EKF provides a first step to solving this problem but its results are less than extraordinary. Applying iterations based on estimates of the state and various noise sources, while theoretically gives us the capability of achieving near best case results, the actual outcome still falls short in the majority of our scenarios. To move forward we will have to look at some alternative methods.

Chapter 5

Sigma Point / Unscented Methods Geometry

"Wonderful", the Flatline said,"I never did like to do anything simple when I could do it ass-backwards."

- William Gibson, *Neuromancer*

As we have seen in the previous chapter the EKF often does not perform as well as needed in some situations such as with the case of the blind tricyclist. A commonly implemented alternative, which can have superior performance, in nonlinear problems is the Unscented Kalman Filter (UKF). In the EKF we attempted to approximate the transformations as linear about linearization points, an approach which we have already shown to have limitations. Instead of approximating the transform by linearizing about a point the Unscented Transform approximates the covariances in the problem through strategic sampling. The following description of the method is taken, with a change in notation where appropriate, from [4].

5.1 The Unscented Transform

The goal of the unscented/σ-point transform is to estimate the result, z, of a nonlinear transformation, f, of a Gaussian random variable, $x \in \mathbb{R}^n \sim N(\mu, P)$. To do this it estimate the random variable x as a list of points, $\mathscr{X}$ referred to as σ-points, and a weight vector, w, which can reconstruct into the original random variable's statistics using the process described as follows.

$$
\begin{aligned}
x \sim N(\mu_x, P_{xx}) & & x \in \mathbb{R}^m & \quad \text{The input RV} \\
z = f(x) & & y \in \mathbb{R}^n & \quad \text{The transformed RV } (f \text{is nonlinear}) \\
x \to \left(\mathscr{X} = \left[\mathscr{X}_1, \mathscr{X}_2, \mathscr{X}_3, \ldots, \mathscr{X}_p\right], w\right) & & \mathscr{X}_i \in \mathbb{R}^m, w \in \mathbb{R}^p & \quad \text{The } \sigma\text{-pointsof the RV } x \\
\mu_x \approx \sum_{i=1}^{p} w_i \mathscr{X}_i & & \mu_x \in \mathbb{R}^n & \quad \text{The mean of } x \\
P_{xx} \approx \sum_{i=1}^{p} w_i \left(\mathscr{X}_i - \mu_x\right)\left(\mathscr{X}_i - \mu_x\right)^{\mathrm{T}} & & P_{xx} \in \mathbb{R}^{n \times n} & \quad \text{The covariance of } x
\end{aligned}
\tag{5.1.1}
$$

The σ-point transform assumes that by propagating the representative points through the nonlinear transform we can use the same process to construct the resulting random variable's statistics.

$$
\begin{aligned}
z = f(x) & & z \in \mathbb{R}^n & \quad \text{The transformed RV } (f \text{is nonlinear}) \\
\mathscr{Z}_i = f(\mathscr{X}_i) & & \mathscr{Z}_i \in \mathbb{R}^n & \quad \text{The transformed } \sigma\text{-points} \\
\mu_z \approx \sum_{i=1}^{p} w_i \mathscr{Z}_i & & \mu_z \in \mathbb{R}^n & \quad \text{The approximate mean of } z \\
P_{zz} \approx \sum_{i=1}^{p} w_i \left(\mathscr{Z}_i - \mu_z\right)\left(\mathscr{Z}_i - \mu_z\right)^{\mathrm{T}} & & P_{zz} \in \mathbb{R}^{n \times n} & \quad \text{The approximate covariance of } z \\
P_{xz} \approx \sum_{i=1}^{p} w_i \left(\mathscr{X}_i - \mu_x\right)\left(\mathscr{Z}_i - \mu_z\right) & & P_{xz} \in \mathbb{R}^{m \times n} & \quad \text{The approximate cross covariance} \\
\mathfrak{S}(f, x) = (\mu_z, P_{zz}, P_{xz}) & & & \quad \text{The unscented transform}
\end{aligned}
\tag{5.1.2}
$$

The advantage of this method is that it actively samples the nonlinear function with points chosen to represent the original random variable. The σ-points are a middle ground between the brute force method of simply statistically sampling the nonlinear function as employed in the Ensemble Kalman Filter and the EKF's assumption of linearity. It is often pointed out that the method does not require the first derivative, i.e. the Jacobian, of the function[1]. To aid in generality we will construct a standard set of σ-points for $N(0, I)$ and include the affine transform which maps a standard normal to $N(\mu, P)$ as part of the function f. If $x \sim N(0, I)$ then the constraints arise from Equation 5.1.1 given by

[1]The Jacobian is often vilified to be some difficult to compute notational nightmare. In the current age of computer algebra systems where these systems can both, efficiently and without error, compute derivatives and generate code for them this should not be a factor. Additionally it is true that the σ-point method is uncaring of the function's possible lack of first derivative, but in my opinion that does not mean the implementor should be. Poorly behaved functions need to be handled with great care and as the σ-points are generated deterministically they lack the statistical convergence theorems that aid the brute force methods of PF and Ensemble methods.

$$\begin{aligned} \sum_{i=1}^{p} w_i &= 1 \\ \sum_{i=1}^{p} w_i \mathscr{X}_i &= 0 \\ \sum_{i=1}^{p} w_i \mathscr{X}_i \mathscr{X}_i^{\mathrm{T}} &= I. \end{aligned} \tag{5.1.3}$$

5.2 Orthogonal Geometry of Sigma Point Sets

The conditions as stated in Equation 5.1.3 are helpful when thinking about how the σ-points can be combined to form a sample mean and covariance but as we move forward and start discussing higher order moments and examining the geometry of the points themselves I have found it conducive for visualization to move to a second form. Consider instead the simplification of introducing the vectors[2] of i^{th} components of the original σ-points, $\mathscr{Y}_i, i = 1 \dots n$, which we refer to as the σ^{T}-point.

$$\mathscr{Y}_i = \begin{bmatrix} \mathscr{X}_{1,i} & \mathscr{X}_{2,i} & \mathscr{X}_{3,i} & \cdots & \mathscr{X}_{p,i} \end{bmatrix}$$

Lemma. *The σ^{T}-points lie in the hyperplane normal to the weight vector.*

Proof. This should be clear by rewriting the condition $\sum_{i=1}^{p} w_i \mathscr{X}_i = 0$ as $w \cdot \mathscr{Y}_k = 0$ □

Lemma. *The σ^{T}-points and weight vector form a basis for an n-dimensional space.*

Proof. The covariance constraint can be rewritten as $w \cdot \left(\mathscr{Y}_i \star \mathscr{Y}_j\right) = \delta_{i,j}$ where $\star$ is element-wise multiplication, the Hadamard product. By contradiction assume that $\mathscr{Y}$, the set of σ^{T}-points, does not span a n-dimensional space then there would exist at least one vector of it, $\mathscr{Y}_q$, which could be written as the sum of the others.

[2] That is that $\mathscr{Y}_i$ is the ith component of all $\mathscr{X}_k$. If there are p σ-points, $\mathscr{X}_k$, and each is an element of $\mathbb{R}^n$ then there are n $\mathscr{Y}$-vectors with each $\mathscr{Y}_i$ is an element of $\mathbb{R}^p$, $\mathscr{Y} = \mathscr{X}^{\mathrm{T}}$. The advantage here is than w is also an element of $\mathbb{R}^p$, meaning $\mathscr{Y}$ and w are in the same space and we can use geometry when working with the σ^{T}-point and the weight vector w.

$$
\begin{aligned}
\mathscr{Y}_q &= \sum_{i \neq q} a_i \mathscr{Y}_i \\
w \cdot \left(\mathscr{Y}_q \star \mathscr{Y}_r\right) &= w \cdot \left(\left(\sum_{i \neq q} a_i \mathscr{Y}_i\right) \star \mathscr{Y}_r\right) \\
&= a_r w \cdot \left(\mathscr{Y}_r \star \mathscr{Y}_r\right) + \sum_{i \neq r, i \neq q} a_i w \cdot \left(\mathscr{Y}_i \star \mathscr{Y}_j\right) \\
&= a_r = 0 \quad \forall r \neq q \\
\mathscr{Z}_q &= 0 \\
w \cdot \left(\mathscr{Y}_q \star \mathscr{Y}_q\right) &= 0 \neq 1 \, \text{contradiction}
\end{aligned}
$$

□

These previous two statements have both exploited the geometric advantage of the σ^{T}-points over the σ-points which results in the conditions in Equation 5.1.3 being rewritten as follows in Equation 5.2.1. Between the two statements we have a minimum dimension of $\mathscr{Y}$, i.e. given $\mathscr{Y}$ spans an n-D space and w is orthogonal to it, all these σ^{T}-point vectors must have dimension $p \geq n+1$.

$$
\begin{aligned}
1 &= \sum_{i=1}^{p} w_i \\
0 &= w \cdot \mathscr{Y}_i \\
\delta_{i,j} &= w \cdot \left(\mathscr{Y}_i \star \mathscr{Y}_j\right) \qquad \text{where } \star \text{is elementwise multiplication}
\end{aligned}
\tag{5.2.1}
$$

Lemma. *For every set of σ-points which is minimal for a dimension, $p = n+1$, $\{w, \mathscr{Y}_1, \mathscr{Y}_2, \ldots, \mathscr{Y}_n\}$, there is a related, orthogonal, prototype matrix $\mathcal{S} = \begin{bmatrix} \sqrt[\surd]{w} \\ \sqrt[\surd]{w} \star \mathscr{Y}_1 \\ \vdots \\ \sqrt[\surd]{w} \star \mathscr{Y}_p \end{bmatrix} \in O\left(p, \mathbb{C}\right)$, where $\sqrt[\surd]{w}$ is the element-wise square-root of w.*

Proof. The key here is to recognize the orthogonality constraint as an alternative construction of our original set in Equation 5.2.1.

$$
\mathcal{S}\mathcal{S}^{\mathrm{T}} = \begin{bmatrix}
\sqrt[\surd]{w} \cdot \sqrt[\surd]{w} & \sqrt[\surd]{w} \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) & \cdots & \sqrt[\surd]{w} \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) & \cdots & \sqrt[\surd]{w} \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right) \\
\left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) \cdot \sqrt[\surd]{w} & \left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) & & \left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) & & \left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right) \\
\vdots & & & & & \\
\left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) \cdot \sqrt[\surd]{w} & \left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) & & \left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) & & \left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right) \\
\vdots & & & & & \\
\left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right) \cdot \sqrt[\surd]{w} & \left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_1\right) & & \left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_k\right) & & \left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right) \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_p\right)
\end{bmatrix}
$$

The elements of this matrix can be simplified greatly, recalling the properties, $a \cdot (b \star c) = b \cdot (a \star c)$ and $(a \star b) \cdot (c \star d) = (a \star d) \cdot (c \star b)$. Restating the constraint on the sum of weights $1 = \sum_{i=0}^{p} w_i = \sum_{i=0}^{p} \sqrt[\surd]{w}_i \sqrt[\surd]{w}_i = \sqrt[\surd]{w} \cdot \sqrt[\surd]{w}$. Restating the constraint on the mean $0 = w \cdot \mathscr{Y}_i = \left(\sqrt[\surd]{w} \star \sqrt[\surd]{w}\right) \cdot \mathscr{Y}_i = \sqrt[\surd]{w} \cdot \left(\sqrt[\surd]{w} \star \mathscr{Y}_i\right)$. Restating the constraint on covariance $\delta_{i,j} = w \cdot (\mathscr{Y}_i \star \mathscr{Y}_j) = \left(\sqrt[\surd]{w} \star \sqrt[\surd]{w}\right) \cdot (\mathscr{Y}_i \star \mathscr{Y}_j) = \left(\sqrt[\surd]{w} \star \mathscr{Y}_i\right)\left(\sqrt[\surd]{w} \star \mathscr{Y}_j\right)$, so $SS^{\mathrm{T}} = I$. □

Theorem 1. *For every set of σ^{T}-points, $\{w, \mathscr{Y}_1, \mathscr{Y}_2, \ldots, \mathscr{Y}_n\}$, there is a related, orthogonal prototype matrix given in Equation 5.2.2, where $\sqrt[\surd]{w}$ is the element-wise square-root of w and $\mathscr{Y}_k$ for $k > n$ are unused values. We call the values $\{w, \mathscr{Y}_1, \mathscr{Y}_2, \ldots, \mathscr{Y}_p\}$ the complete set of σ-points$^{\mathrm{T}}$ as opposed to the original, incomplete, set, $\{w, \mathscr{Y}_1, \mathscr{Y}_2, \ldots, \mathscr{Y}_n\}$.*

Proof. We have already shown that $\{w, \mathscr{Y}_1, \mathscr{Y}_2, \ldots, \mathscr{Y}_n\}$ span an $(n+1)$ dimensional space, by all the same logic that demonstrated that we can show that $\left\{\sqrt[\surd]{w}, \sqrt[\surd]{w} \cdot \mathscr{Y}_1, \sqrt[\surd]{w} \cdot \mathscr{Y}_2, \ldots, \sqrt[\surd]{w} \cdot \mathscr{Y}_n\right\}$ also spans a $(n+1)$ dimensional space and we need only expand this basis, which is already orthonormal to a full set to form the matrix $\mathcal{S}$. However, only the σ-points$^{\mathrm{T}}$ which are going to be used need to be real which frees the others to be complex, also allowing negative weights, in the form of purely imaginary elements of $\sqrt[\surd]{w}$. Not all elements $\mathcal{S} \in O(p, \mathbb{C})$ have σ-point equivalents however as $\{w, \mathscr{Y}_1, \mathscr{Y}_2, \ldots, \mathscr{Y}_n\}$ must all be real. □

This gives us a way of parameterizing the space of possible p σ-points for an n-dimensional state space. We can start with the parameterization of the space $O(p, \mathbb{C})$, which we can define as the product of $(p-1)(p-2)/2$ Givens rotation matrices, and then transform the orthogonal matrix into a σ-point set via the inverse of Equation 5.2.2. The previous definition does not guarantee that all orthogonal matrices have related σ-point sets and so the parameterization will yield impossible sets when, for example, there is an element of $\sqrt[\surd]{w}$ which is zero or geometrically unsuitable sets, when the used (the first n), vectors of $\mathscr{Y}$ have elements which are not real.

$$\mathcal{S} = \begin{bmatrix} \sqrt[\surd]{w} \\ \sqrt[\surd]{w} \star \mathscr{Y}_1 \\ \vdots \\ \sqrt[\surd]{w} \star \mathscr{Y}_n \\ \vdots \\ \sqrt[\surd]{w} \star \mathscr{Y}_p \end{bmatrix} \in O(p, \mathbb{C}) \tag{5.2.2}$$

5.3 Lie Algebra of Higher Order Sigma Point Moments

The conditions of 5.1.3 guarantee that our original, untransformed, estimates of the first and second moments are correct but in nonlinear problems it is be helpful to correctly estimate other higher order moments, for an example of the benefits of considering higher order moments see Section 5.5. We can define the higher order moments of a set of σ-points the same way that we define the lower order ones, in terms of their weighted sample sums.

$$E\left(x_{q_1}x_{q_2}...x_{q_k}\right) \approx \sum_{i=1}^{p} w_i \mathscr{X}_{i,q_1}\mathscr{X}_{i,q_2}\dots\mathscr{X}_{i,q_k} = w\cdot\left(\mathscr{Y}_{q_1}\star\mathscr{Y}_{q_2}\star\cdots\star\mathscr{Y}_{q_k}\right) \tag{5.3.1}$$

By our earlier definitions we know that the first and second order moments are correct so let us begin by considering the 3rd order moments of the σ-points. The equation for these moments is $E_{j,k,m} = w\cdot\left(\mathscr{Y}_j\star\mathscr{Y}_k\star\mathscr{Y}_m\right)$. Just as we were able to split the matrix of second order moments, the covariance, into an orthogonal matrix we can factor matrices of 3rd order moments by the same orthogonal matrix. The process is as follows,

$$\begin{aligned} w\cdot\left(\mathscr{Y}_j\star\mathscr{Y}_k\star\mathscr{Y}_m\right) &= \left(\overset{\surd}{w}\star\overset{\surd}{w}\right)\cdot\left(\mathscr{Y}_j\star\mathscr{Y}_k\star\mathscr{Y}_m\right) \\ &= \left(\overset{\surd}{w}\star\mathscr{Y}_j\right)\cdot\left(\mathscr{Y}_k\star\overset{\surd}{w}\star\mathscr{Y}_m\right) \\ &= \left(\overset{\surd}{w}\star\mathscr{Y}_j\right)\cdot D_{\mathscr{Y}_k}\cdot\left(\overset{\surd}{w}\star\mathscr{Y}_m\right) \end{aligned}$$

D_x is the matrix with diagonal elements from x

Which gives us the definition in Equation 5.3.2.

$$A_k = \mathcal{S}D_{\mathscr{Y}_k}\mathcal{S}^{\mathrm{T}} \tag{5.3.2}$$

Lemma. *The third order moments with a given index j, $E_{m,j,n}$, are all elements of the matrix A_j*

Proof. The element (m,n), zero indexed, of the matrix A_j, $A_{j,m,n}$ is

$$\mathcal{S}_m D_{\mathscr{Y}_j}\mathcal{S}_n^{\mathrm{T}} = \begin{cases} \left(\overset{\surd}{w}\right)\cdot\left(\mathscr{Y}_j\star\overset{\surd}{w}\right) = 0 & \text{if } m=0, n=0 \\ \overset{\surd}{w}\cdot\left(\mathscr{Y}_j\star\left(\overset{\surd}{w}\star\mathscr{Y}_n\right)\right) = \delta_{j,n} & \text{if } m=0, n>0 \\ \left(\overset{\surd}{w}\star\mathscr{Y}_m\right)\cdot\left(\mathscr{Y}_j\star\overset{\surd}{w}\right) = \delta_{j,m} & \text{if } n=0, m>0 \\ \left(\overset{\surd}{w}\star\mathscr{Y}_m\right)\cdot\left(\mathscr{Y}_j\star\left(\overset{\surd}{w}\star\mathscr{Y}_n\right)\right) = E_{m,j,n} & \text{if } m>0, n>0 \end{cases}$$

$$A_j = \begin{bmatrix} 0 & 0 & 0 & & 1 & & 0 \\ 0 & E_{1,j,1} & E_{1,j,2} & \cdots & E_{1,j,j} & \cdots & E_{1,j,p-1} \\ 0 & E_{2,j,1} & E_{2,j,2} & & E_{2,j,j} & & E_{2,j,p-1} \\ & \vdots & & & \vdots & & \vdots \\ 1 & E_{j,j,1} & E_{j,j,2} & & E_{j,j,j} & \cdots & E_{j,j,p-1} \\ & \vdots & & & \vdots & & \vdots \\ 0 & E_{p-1,j,1} & E_{p-1,j,2} & & E_{p-1,j,j} & & E_{p-1,j,p-1} \end{bmatrix}$$

□

Remark. The matrices $\{A_j\}$ are simultaneously diagonalizable by definition and therefore pairwise commute, $A_j A_k = A_k A_j$.

Additionally the values from the mean can be found in the first element of these matrices $(A_k)_{0,0} = u_k$, the covariance matrix can be found in the first row/column, $P_{j,k} = (A_k)_{0,j} = (A_k)_{j,0}$.

Theorem 2. *Just as the third order moments with a given index j, $E_{m,j,n}$, are the elements of A_j, the fourth order moments of a given pair (j,k), are the elements of $A_j A_k$.*

Proof. $A_j A_k = \mathcal{S} D_{\mathscr{Y}_j} \mathcal{S}^{\mathrm{T}} \mathcal{S} D_{\mathscr{Y}_k} \mathcal{S} = \mathcal{S} D_{\mathscr{Y}_j \star \mathscr{Y}_k} \mathcal{S}^{\mathrm{T}}$ and as before we can select an element (m,n), zero indexed of this product,

$$\left(A_j A_k\right)_{m,n} = \mathcal{S}_m D_{\mathscr{Y}_j \star \mathscr{Y}_k} \mathcal{S}_n^{\mathrm{T}} = \begin{cases} \left(\overset{\vee}{w}\right) \cdot \left(\left(\mathscr{Y}_j \star \mathscr{Y}_k\right) \star \left(\overset{\vee}{w}\right)\right) = E_{j,k} = \delta_{j,k} & \text{if } m = 0, n = 0 \\ \overset{\vee}{w} \cdot \left(\left(\mathscr{Y}_j \star \mathscr{Y}_k\right) \star \left(\overset{\vee}{w} \star \mathscr{Y}_n\right)\right) = E_{j,k,n} & \text{if } m = 0, n > 0 \\ \left(\overset{\vee}{w} \star \mathscr{Y}_m\right) \cdot \left(\left(\mathscr{Y}_j \star \mathscr{Y}_k\right) \star \overset{\vee}{w}\right) = E_{m,j,k} & \text{if } n = 0, m > 0 \\ \left(\overset{\vee}{w} \star \mathscr{Y}_m\right) \cdot \left(\left(\mathscr{Y}_j \star \mathscr{Y}_k\right) \star \left(\overset{\vee}{w} \star \mathscr{Y}_n\right)\right) = E_{m,j,k,n} & \text{if } m > 0, n > 0 \end{cases}$$

□

Remark. It should be clear from this trend, that n^{th} order moments can be found in products of $(n-1)$ elements of $\{A_j\}$, should be clear by extension

As before elements of two moments down can be found in the $0,0$ elements of all products, and elements of moments one level down are the first row/column.

Theorem 3. *The set $\{I, A_1, A_2, \ldots, A_{p-1}\}$ forms a basis for the Lie algebra of matrices which are simultaneously diagonalizable by $\mathcal{S}$. Additionally if an element B is an element of this algebra then its first row, B_0, comprises the coefficients of the basis, $B = B_{0,0} I + B_{0,1} A_1 + \cdots + B_{0,p-1} A_{p-1}$.*

Set Name	Moments	Number of Points (general)	4D	10D
Simp	Min 2norm error of 3rd order	$d+1$	5	11
O5	Order 5 for any 1 Dimension	$2d+1$	9	21
O7	Order 7 for any 1 Dimension	$4d+1$	17	41
O5f	Order 5 Full Match	$4\binom{d}{2}+4d+1$	41	221

Table 5.1: Sigma Point Set Properties

Proof. Let B be diagonalizable by $\mathcal{S}$, $B = \mathcal{S}D_b\mathcal{S}^{\mathrm{T}}$. Then $b \in \mathbb{C}^p$ and $\left\{w, \mathscr{Y}_1, \ldots, \mathscr{Y}_{p-1}\right\}$ is a basis of $\mathbb{C}^p$, so $\left\{\mathcal{S}D_w\mathcal{S}^{\mathrm{T}}, \mathcal{S}D_{\mathscr{Y}_1}\mathcal{S}^{\mathrm{T}}, \ldots, \mathcal{S}D_{\mathscr{Y}_{p-1}}\mathcal{S}^{\mathrm{T}}\right\}$ is a basis and we just substitute the vector of all ones, 1, in for w, as it contains an element in the direction of w, $1 \cdot w = 1$, and the rest of the basis $\mathscr{Y}$ does not. $D_1 = I$, $\mathcal{S}I\mathcal{S}^{\mathrm{T}} = I$ so the new basis becomes $\left\{I, A_1, \ldots, A_{p-1}\right\}$. The second part is clear from the first rows of the basis which each have only one, non-overlapping, nonzero element. □

5.4 Derived Sigma Point Sets

We can use these these geometric results to find qualified σ-point sets which have different properties. The two important properties of σ-point sets we will be examining are the number of points needed (which determines the numerical complexity of using the set) and the higher order moments (See next section for benefits). Table 5.1 lists these properties for the sets of interest we will construct.

5.4.1 Simplex Set (Simp)

Consider first the question of finding a "good" minimal set of σ-points for a given dimension, d. To define the quality of a minimal set I suggest that we first try and minimize the error in the third order moments. For a standard Gaussian all the third order moments, $E_{i,j,k}$, should be zero. For a set of σ-points all these values, the third order moments $E_{i,j,k}$, can be found in the matrices A_k, i.e. $E_{i,j,k} \in A_k \forall i, j$. Using a minimal set of points it will not be possible to meet these, 3rd order, constraints, $E_{i,j,k} = 0$, so instead consider minimizing the error in these matrices via the sum Frobenious norm, $\sum \|A_k\|_{\mathrm{F}}$. This norm will count certain entries[3] multiple times and include an offset but algebraically has many advantages. To minimize the error we might consider the derivative in error function in terms of changes in prototype matrix $\mathcal{S}$. The prototype matrix, $\mathcal{S}$, is orthogonal and any transformation of it can be expressed as another rotation matrix R. We will define the result of this transformation with the acute diacritical mark, ́, for all

[3]Both $E_{1,1,2}$ and $E_{2,1,1}$ appear in A_1 but are the same value and they again appear in A_2 as $E_{1,2,1}$ leading to it being triple counted, where as $E_{1,1,1}$ only appears once in A_1

other dependent quantities (e.g. $\acute{\mathscr{Y}}$ is the transformed σ^{T}-point).

$$\acute{\mathcal{S}} = R\mathcal{S}$$

The full impact of this transformation on the 3rd order moment matrices, $\acute{A}_k = \acute{\mathcal{S}} D_{\acute{\mathscr{Y}}_k} \acute{\mathcal{S}}^{\mathrm{T}}$, is difficult to find because we need to find the new element $\acute{\mathscr{Y}}_k$. Ignoring this one difficulty the rest of the transform takes an easily recognized similarity transform structure, $\acute{A}_k = R\mathcal{S} D_{\acute{\mathscr{Y}}_k} S^{\mathrm{T}} R^{\mathrm{T}}$ which gives us the following form.

$$\hat{A}_k = R A_k R^{\mathrm{T}}$$

This new matrix is formed directly from the previous one and because it is diagonalized by $R\mathcal{S} = \acute{\mathcal{S}}$ it is a member of the Lie algebra discussed in Theorem 3. By the other properties of that Theorem we know that it can be written as a linear combination of the matrices $\left\{ I, \acute{A}_1, \ldots, \acute{A}_p \right\}$.

$$\begin{aligned} \hat{A}_k &= R A_k R^{\mathrm{T}} \\ &= I M_{k,0} + \acute{A}_1 M_{k,1} + \cdots + \acute{A}_p M_{k,p} \\ M_{k,j} &= A_{k,(0,j)} \end{aligned}$$

The next step is recognizing that the form $\hat{A}_k = R A_k R^{\mathrm{T}}$ can be written using a Kronecker product $\mathscr{V}\left(\hat{A}_k\right) = R \otimes R \mathscr{V}\left(A_k\right)$ where $\otimes$ is the Kronecker product and $\mathscr{V}(\cdot)$ is the row vectorization operation, taking a matrix and flattening it into a vector by stacking its rows.

$$\begin{aligned} \mathscr{V}\left(\hat{A}_k\right) &= (R \otimes R)\, \mathscr{V}\left(A_k\right) \\ &= \mathscr{V}(I)\, M_{k,0} + \mathscr{V}\left(\acute{A}_1\right) M_{k,1} + \cdots + \mathscr{V}\left(\acute{A}_p\right) M_{k,p} \end{aligned}$$

It is apparent from the right side of this equation that the matrix $\mathcal{A} = \begin{bmatrix} \mathscr{V}(I) & \mathscr{V}\left(A_1\right) & \ldots & \mathscr{V}\left(A_p\right) \end{bmatrix}^{\mathrm{T}}$ and its transformed version, $\acute{\mathcal{A}}$, are going to be helpful. This matrix has another form which is obvious when examining the actual elements.

$$
\begin{aligned}
\mathcal{A} &= \begin{bmatrix} \mathcal{V}(I) & \mathcal{V}(A_1) & \mathcal{V}(A_2) & \dots & \mathcal{V}(A_p) \end{bmatrix}^{\mathrm{T}} \\
&= \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & & 0 \\
0 & 0 & 1 & & 0 \\
\vdots & & & & \vdots \\
0 & 0 & 0 & \dots & 1 \\
0 & 1 & 0 & \cdots & 0 \\
1 & E_{1,1,1} & E_{1,1,2} & & E_{1,1,p} \\
0 & E_{1,2,1} & E_{1,2,2} & & E_{1,2,p} \\
\vdots & & & & \vdots \\
0 & E_{1,p,1} & E_{1,p,2} & \dots & E_{1,p,p} \\
\vdots & & & & \vdots
\end{bmatrix} \\
&= \begin{bmatrix} I \\ A_1 \\ A_2 \\ \vdots \\ A_p \end{bmatrix}
\end{aligned}
$$

Additionally the matrix M can be written as a projection of the related matrix $\hat{\mathcal{A}}$.

$$
\begin{aligned}
M &= \begin{bmatrix} I & 0 & 0 & \dots & 0 \end{bmatrix} \hat{\mathcal{A}} \\
&= \begin{bmatrix} I & 0 & 0 & \dots & 0 \end{bmatrix} (R \otimes R)\, \mathcal{A} \\
&= (R_1 \otimes R)\, \mathcal{A}
\end{aligned}
$$

This gives us the following form,

$$
(R \otimes R)\, \mathcal{A} = \acute{\mathcal{A}}\, (R_1 \otimes R)\, \mathcal{A},
$$

which greatly simplifies if we restrict R to the subset of matrices of the form $\left[\begin{smallmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{smallmatrix}\right] \in O(p-1, \mathbb{C})$.

Lemma. *The 2-norm error of the 3rd order matrices, A_k, is only dependent on the weight vector $w \in S^{p-1}$*

Proof. As we have shown transforms of the form $\left[\begin{smallmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{smallmatrix}\right]$ change the 3rd order error matrices through the following form

$$
\begin{aligned}
\acute{\mathcal{A}} &= \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \mathcal{A} \\
\left(\begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \right) \mathcal{A} &= \acute{\mathcal{A}} \left(\begin{bmatrix} 1 & 0 & \dots \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \right) \mathcal{A} \\
&= \acute{\mathcal{A}} \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \\
\acute{\mathcal{A}} &= \left(\begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \right) \mathcal{A} \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix}^{\mathrm{T}} \\
\mathscr{V}\left(\acute{\mathcal{A}}\right) &= \left(\begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix} \right) \mathscr{V}\left(\mathcal{A}\right) \\
&= \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{bmatrix}^{\otimes 3} \mathscr{V}\left(\mathcal{A}\right)
\end{aligned}
\tag{5.4.1}
$$

The matrix $\left[\begin{smallmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{smallmatrix}\right]$ is orthogonal so its 3rd Kronecker power $\left[\begin{smallmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{smallmatrix}\right]^{\otimes 3}$ is as well. Orthogonal matrices preserve 2-norms on vector spaces so that $\left\| \mathscr{V}\left(\acute{\mathcal{A}}\right) \right\|_2 = \left\| \mathscr{V}\left(\mathcal{A}\right) \right\|_2$. The decomposition of a general prototype matrix, $\mathcal{S}$, into the two parts $\mathcal{S} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{smallmatrix}\right] \mathcal{W}$, where we define $\mathcal{W}$ in Equation 5.4.2 using Givens angles, $\phi_1 \dots \phi_k$, makes the 2-norm only dependent on $\mathcal{W}$.

$$
\mathcal{W}\left(\phi_1, \phi_2, \dots, \phi_{p-1}\right) = \begin{bmatrix}
\prod_{k=1}^{p-1} \cos\left(\phi_k\right) & \sin\left(\phi_{p-1}\right) \prod_{k=1}^{p-2} \cos\left(\phi_k\right) & \cdots & \sin\left(\phi_3\right) \cos\left(\phi_1\right) \cos\left(\phi_2\right) & \sin\left(\phi_2\right) \cos\left(\phi_1\right) & \sin\left(\phi_1\right) \\
-\sin\left(\phi_{p-1}\right) & \cos\left(\phi_{p-1}\right) & \cdots & 0 & 0 & 0 \\
-\cos\left(\phi_{p-1}\right) \sin\left(\phi_{p-2}\right) & -\sin\left(\phi_{p-1}\right) \sin\left(\phi_{p-2}\right) & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots \\
-\left(\prod_{k=3}^{p-1} \cos\left(\phi_k\right)\right) \sin\left(\phi_2\right) & -\sin\left(\phi_{p-1}\right) \left(\prod_{k=3}^{p-2} \cos\left(\phi_k\right)\right) \sin\left(\phi_2\right) & \cdots & -\sin\left(\phi_3\right) \sin\left(\phi_2\right) & \cos\left(\phi_2\right) & 0 \\
-\left(\prod_{k=2}^{p-1} \cos\left(\phi_k\right)\right) \sin\left(\phi_1\right) & -\sin\left(\phi_{p-1}\right) \left(\prod_{k=2}^{p-2} \cos\left(\phi_k\right)\right) \sin\left(\phi_1\right) & \cdots & -\sin\left(\phi_3\right) \cos\left(\phi_2\right) \sin\left(\phi_1\right) & -\sin\left(\phi_2\right) \sin\left(\phi_1\right) & \cos\left(\phi_1\right)
\end{bmatrix}
\tag{5.4.2}
$$

□

The next step is to solve the optimization problem of minimizing the 2-norm error in terms of the parameters of $\mathcal{W}$. Minimizing the sum of the Frobenius norms of the matrices A_k is the same as minimizing the squared sum of their eigenvalues, which for these[4] A_k are simply the σ-point$^{\mathrm{T}}$, $\mathscr{Y}_k$.

Conjecture. *The minimal set of σ-points which minimizes the Frobenius norm error of 3rd order moments are those which arise from equal weights.*

Proof. We have shown this to be the case analytically for small n, $n \leq 6$. □

[4]If the values of A_k were allowed to be complex this may not be the case as the decomposition $SD_{\mathscr{Y}}S^{\mathrm{T}}$ is no longer the eigenvalue decomposition but because we have the minimal number of σ-points we can guarantee that they all must be real.

5.4.2 One Dimensional Sets

The next types of sets to consider are those which meet some higher order moments exactly by using more σ-points than the minimal set, $p > d + 1$. In general it is very difficult to find such sets because of the number of variables which need to be considered. A typical example might be to find a set of points which meets all 3rd order constraints for a state space of 5 variables using 10 σ-points. The parameterization based on the Orthogonal space representation would have 28 angles to search and thus becomes very difficult very fast. Instead let us consider a different strategy. Imagine we had a set of σ-points, $\{w, \mathscr{Y}_1\}$, which met some set of conditions for a 1D system. We can scale it up to two dimensions by creating a stacked version,

$$\begin{aligned} \acute{w} &= \begin{bmatrix} -1 & w & w \end{bmatrix} \\ \acute{\mathscr{Y}}_1 &= \begin{bmatrix} 0 & \mathscr{Y}_1 & 0 \end{bmatrix} \\ \acute{\mathscr{Y}}_2 &= \begin{bmatrix} 0 & 0 & \mathscr{Y}_1 \end{bmatrix} \end{aligned} .$$

This combination still meets all the original conditions (Equation 5.1.3), the addition of a 0 point with a negative weight maintains the $\sum \acute{w} = 1$ without needing to scale w and because $\acute{\mathscr{Y}}_1$ and $\acute{\mathscr{Y}}_2$ are mutually exclusive all products $\acute{\mathscr{Y}}_1 \star \acute{\mathscr{Y}}_2 = 0$ maintaining cross covariance terms, otherwise the vector $\mathscr{Y}_1$ and its weight are unchanged, so all moments involving only one index are the same, $\acute{E}_1 = E_1 = \acute{E}_2$, $\acute{E}_{1,1} = E_{1,1} = \acute{E}_{2,2}$, et cetera. So this set will preserve all the same 1D moments and have 0 for all cross moments (which is not correct, for Gaussians $E_{1,1,2,2} = E\left(X_1 X_1 X_2 X_2\right) = 1$). Using this strategy we can construct a set with these 1D properties for an arbitrary dimensional state space. The naming convection for the sets will be determined by the depth of 1D moments they satisfy so the following Order 3 (O3) set meets all 1D moments up to and including the 3rd, Order 7 (O7) goes up to and including 7th order moments, these sets will all have 0 mixed moments, which is correct only for the odd orders.

Order 3 (O3)

The first problem we will look at is to find how many moments can be met with only 2 points. To solve this problem I suggest that we look at the matrix A_1 because it contains all the information needed to compute all the 1D moments of the set, $E_{1,1,1} \in A_1$, $E_{1,1,1,1} \in A_1 A_1$ et cetera.

$$A_1 = \begin{bmatrix} 0 & 1 \\ 1 & E_{1,1,1} \end{bmatrix}$$

If we set $E_{1,1,1} = 0$, the correct 3rd order moment we have the matrix $\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$. With no more free variables we are done and can diagonalize this matrix to find the prototype which gives us the σ^{T}-point set $w = \left[\begin{smallmatrix} 1/2 & 1/2 \end{smallmatrix}\right], \mathscr{Y}_1 = \left[\begin{smallmatrix} 1 & -1 \end{smallmatrix}\right]$. This form when stacked to obtain structures supporting $d > 1$, beyond a 1 dimensional system, will need $p = 2d + 1$ points. This set fits the more typical σ-points setup[4] with the parameter $\lambda = 1 - d$.

Order 5 (O5)

We can increase the number of σ-points to 3 and check again to see how many values we can match, again by looking at powers of A_1.

$$
\begin{aligned}
A_1 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & E_{1,1,1} & E_{1,1,2} \\ 0 & E_{1,1,2} & E_{1,2,2} \end{bmatrix} \\
E_{1,1,1} &= 0 \\
A_1 A_1 &= \begin{bmatrix} 1 & 0 & E_{1,1,2} \\ 0 & E_{1,1,2}^2 + 1 & E_{1,1,2} E_{1,2,2} \\ 0 & E_{1,1,2} E_{1,2,2} & E_{1,1,2}^2 + E_{1,2,2}^2 \end{bmatrix} \\
E_{1,1,1,1} &= 3 = E_{1,1,2}^2 + 1 \\
E_{1,1,2} &= \sqrt{2} \\
A_1^3 &= \begin{bmatrix} 0 & 3 & \sqrt{2} E_{1,2,2} \\ 3 & 2E_{1,2,2} & \sqrt{2} E_{1,2,2}^2 + 3\sqrt{2} \\ \sqrt{2} E_{1,2,2} & \sqrt{2} E_{1,2,2}^2 & E_{1,2,2}^3 + 4E_{1,2,2} \end{bmatrix} \\
E_{1,1,1,1,1} &= 0 = 2E_{1,2,2} \\
E_{1,2,2} &= 0 \\
A_1 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & \sqrt{2} \\ 0 & \sqrt{2} & 0 \end{bmatrix}
\end{aligned}
$$

Diagonalizing gives us the 5rd Order 1D σ^{T}-point set defined in Equation 5.4.3.

$$
\begin{aligned}
w &= \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \\
\mathscr{Y}_1 &= \begin{bmatrix} 0 & \sqrt{3} & -\sqrt{3} \end{bmatrix}
\end{aligned}
\tag{5.4.3}
$$

Normally we would expect this set to require $3d+1$ points, but because contains a 0 vector these along with the weight balancing 0 vector can be combined, giving us a set using $2d+1$ points which is popular in the literature[4] as the set with the parameter $\lambda = 3-d$

Order 7 (O7)

Increasing the number of points again, we can find a set of points which will meet up to 7th order constraints. The sequence of equations are generated the same way as before except they cannot be solved independently. The software tool Axiom[1] was used to solve the resulting system of polynomials and the tool Macaulay2[5] to show that 8th order constraints are impossible with 4 σ-points. The resulting 7th Order 1D σ^{T}-points are shown in Equation 5.4.4 and require $4d+1$ σ-points for the d dimensional case.

$$\begin{aligned} w &= \begin{bmatrix} \frac{1}{4\left(3+\sqrt{6}\right)} & \frac{1}{4\left(3+\sqrt{6}\right)} & \frac{1}{4\left(3-\sqrt{6}\right)} & \frac{1}{4\left(3-\sqrt{6}\right)} \end{bmatrix} \\ \mathscr{Y}_1 &= \begin{bmatrix} \sqrt{3+\sqrt{6}} & -\sqrt{3+\sqrt{6}} & \sqrt{3-\sqrt{6}} & -\sqrt{3-\sqrt{6}} \end{bmatrix} \end{aligned} \tag{5.4.4}$$

Order 9 (O9)

We can continue the process once more to find a set which meets up to 8th order constraints. This set, like the 5th order set, includes a 0 vector and so requires less points than we might expect, requiring $4d+1$ instead of $5d+1$. The 1D set itself is presented in [15] without the geometric results presented here.

$$\begin{aligned} w &= \begin{bmatrix} \frac{8}{15} & \frac{3}{20\left(7+2\sqrt{2}\sqrt{5}\right)} & \frac{3}{20\left(7+2\sqrt{2}\sqrt{5}\right)} & \frac{3}{20\left(7-2\sqrt{2}\sqrt{5}\right)} & \frac{3}{20\left(7-2\sqrt{2}\sqrt{5}\right)} \end{bmatrix} \\ \mathscr{Y}_1 &= \begin{bmatrix} 0 & \sqrt{5+\sqrt{5}\sqrt{2}} & -\sqrt{5+\sqrt{5}\sqrt{2}} & \sqrt{5-\sqrt{5}\sqrt{2}} & -\sqrt{5-\sqrt{5}\sqrt{2}} \end{bmatrix} \end{aligned} \tag{5.4.5}$$

5.4.3 Mixed Moment Sets

Sets which meet all moments, including the mixed moments, will be marked with the letter f, for full set. Because only even orders have nonzero mixed moments we will focus this section on assigning 4th order mixed moments, because the 1D sets already have correct, 0, 3rd and 5th order mixed moments[5] because there is no overlap in stacked points all the products $\mathscr{Y}_j \star \mathscr{Y}_k = 0, j \neq k$. For a set which meets all 4th order moments we will need to modify our strategy. Just as before we will consider a strategy of extending a lower dimensional set to higher dimension by stacking. This time, however, the base system will need two points to meet the moment constraint, $E_{1,1,2,2} = 1$. The basic idea will be to take this basic

[5]Recall that for a Gaussian random variable $X \sim N(0, I)$ that the 4th order moment $E(X_1X_1X_2X_2) = 1 = E_{1,1,2,2}$. Also recall that these elements can be found in the product of 3rd order matrices A_k, $E_{j,k,\ell,m} \in A_\ell A_m$.

set of σ-points$^{\mathrm{T}}$ and create a new higher dimensional system out of them by creating a new 'stack' for each possible pairing. Given the two dimensional set of points, $\mathscr{Y}_1, \mathscr{Y}_2$, we can create a 3D system which will have moments, $E_{1,1,2,2}, E_{1,1,3,3}, E_{2,2,3,3}$, by stacking the points as follows.

$$\begin{bmatrix} \mathscr{Y}_1 & \mathscr{Y}_1 & 0 \\ \mathscr{Y}_2 & 0 & \mathscr{Y}_1 \\ 0 & \mathscr{Y}_2 & \mathscr{Y}_1 \end{bmatrix}$$

Method 1 (O4f)

There is an additional complexity however, given a set of σ^{T}-points $w, \mathscr{Y}_1, \mathscr{Y}_2$ which have the moments $E_{i,j,k} = 0, E_{1,1,1,1} = 3 = E_{2,2,2,2}, E_{1,1,2,2} = 1,$ and $E_{i,j,k,\ell} = 0$ otherwise. We could patch these points together as follows to form a combined set.

$$\begin{bmatrix} -2 & w & w & w \\ 0 & \mathscr{Y}_1 & \mathscr{Y}_1 & 0 \\ 0 & \mathscr{Y}_2 & 0 & \mathscr{Y}_1 \\ 0 & 0 & \mathscr{Y}_2 & \mathscr{Y}_2 \end{bmatrix}$$

Although his new set will have the correct moments, $E_{1,1,2,2} = E_{1,1,3,3} = E_{2,2,3,3} = 1$, because the system contains two copies of $\mathscr{Y}_1$ in its new $\acute{\mathscr{Y}}_1$, the 1D contributions will be double counted which will void the original covariance constraint,

$$\acute{E}_{1,1} = \acute{w}\left(\acute{\mathscr{Y}}_1 \star \acute{\mathscr{Y}}_1\right) = \acute{w}\begin{pmatrix} \mathscr{Y}_1 \\ \mathscr{Y}_1 \\ 0 \end{pmatrix} \star \begin{pmatrix} \mathscr{Y}_1 \\ \mathscr{Y}_1 \\ 0 \end{pmatrix} = 2E_{1,1}.$$

We can fix this by scaling the weight vector w by the number of duplicates (which is also the dimension of the final space $d-1$), $\acute{w} = {}^{w}/_{(d-1)}$, giving us the stacked system,

$$\begin{bmatrix} -1/2 & w/2 & w/2 & w/2 \\ 0 & \mathscr{Y}_1 & \mathscr{Y}_1 & 0 \\ 0 & \mathscr{Y}_2 & 0 & \mathscr{Y}_1 \\ 0 & 0 & \mathscr{Y}_2 & \mathscr{Y}_2 \end{bmatrix}$$

This has the unfortunate side effect of scaling all moments back by ${}^{1}/_{(d-1)}$ so our new system will have the cross term moment $\acute{E}_{1,1,2,2} = {}^{1}/_{(d-1)}E_{1,1,2,2}$ meaning that for every system dimension d we will need a unique set of σ-points$^{\mathrm{T}}$which meet the conditions $E_{i,j,k} = 0, E_{1,1,1,1} = 3 = E_{2,2,2,2}, E_{1,1,2,2} = (d-1)$, and $E_{i,j,k,\ell} = 0$ otherwise. I do not know a *good* way of creating this set of points, using the geometry of the moment matrices, A_k. It can be shown that we will need at least 6 σ-points to meet these

conditions, and with 6 points it is not possible to match 5th order moments. Using the Givens rotations parameterization and trimming the angles which do not impact the first 2 σ^{T}-points gives us 12 parameters to search. Using simple optimization with random starting points I've been able to find sets for up to dimension 11, noting that after dimension 4 (where $d-1=3$ so $E_{1,1,1,1}=E_{1,1,2,2}$) we need to allow for a negative weight. In general however the difficulty of finding a valid set using this strategy seems to increases with dimension. The number of points required is also large, $6\binom{d}{2}+1$ so for a 10 dimensional system we will need 271 points, making it somewhat unwieldy.

Method 2 (O5f)

The previous method meets all of the moment constraints in one set and then stacks it simply to get a finished set. Instead I suggest that we find the smallest set which will give us a desired cross moments $E_{i,j,k,\ell}, i \neq \ell$ and then fix any 1D moments that are incorrect with 1D sets. This should lead to fewer points for larger dimensional problems, if we need p_1 points for the pairwise product the final system will need $p_1\binom{d}{2}$ points and if we need an additional p_2 points to correct the 1D moments the final system will only need an additional $p_2 d$ points as compared to a system which requires p_1+p_3 points with no 1D corrections, requiring $p_1\binom{d}{2}+p_3\binom{d}{2}$ total points. The other benefits of this method are happy side effects but include a closed form set of points which will also meet some 5th order moments. To be considered as a possible cross term set we must have a set of points which have zero cross term moments where we need $E_{1,1,2}=E_{1,2,2}=E_{1,1,1,2}=E_{1,2,2,2}=0$ because none of our 1D 'corrections' will allow us to correct these errors. Additionally the set should have a 1D symmetry so we do not need to keep track of 1D errors separately, $E_{1,1,1}=E_{2,2,2}, E_{1,1,1,1}=E_{2,2,2,2}$. Under these conditions it is not possible to create a set using only 3 points. Starting with only these constraints we use Macaulay2[5] to find a Grobner Basis for the system of polynomials from moment parameterization, A_k.

$$\left\{\begin{array}{ccccc}
E_{1,1,1}-E_{2,2,2} & E_{1,2,2} & E_{1,3,3}-E_{2,3,3} & E_{1,1,2} & E_{2,3,3}E_{1,1,3} \\
E_{1,2,3}E_{1,1,3} & E_{2,3,3}^2-E_{2,3,3}E_{1,1,1} & E_{1,2,3}E_{2,3,3}-E_{1,2,3}E_{1,1,1} & E_{2,2,3}E_{2,3,3} & E_{1,2,3}^2-E_{2,2,3}E_{1,1,3}-1 \\
E_{2,2,3}E_{1,2,3} & E_{2,2,3}^2-E_{1,1,3}^2 & E_{1,1,3}^2+E_{2,2,3} & E_{2,2,3}E_{1,1,3}^2+E_{1,1,3} & E_{2,2,3}E_{1,1,1}E_{1,1,3}-E_{2,3,3}+E_{1,1,1}
\end{array}\right\}$$

Within this basis, relevant entries highlighted in red, we find that $E_{1,1,2}=E_{1,2,2}=0$, giving us an element $(A_1A_1)_{[2,2]}=E_{1,1,2,2}=E_{1,2,3}$ which we want to be nonzero which, in turn implies[6] $E_{1,1,3}=0$. Simplifying the ideal with this new identity gives us $E_{1,2,3}=1$

[6]This was relatively easy to identify but in general I believe the correct method for this is to saturate the ideal with $E_{1,2,3}$

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & E_{1,1,1} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & E_{1,3,3} \end{bmatrix}$$

$$A_1 A_1 A_1 = \begin{bmatrix} E_{1,1,1} & E_{1,1,1}^2 + 1 & 0 & 0 \\ E_{1,1,1}^2 & \left(E_{1,1,1}^2 + 1\right) E_{1,1,1} + E_{1,1,1} & 0 & 0 \\ 0 & 0 & E_{1,3,3} & E_{1,3,3}^2 + 1 \\ 0 & 0 & E_{1,3,3}^2 + 1 & \left(E_{1,3,3}^2 + 1\right) E_{1,3,3} + E_{1,3,3} \end{bmatrix}$$

We can be greedy here and meet our constraints and both 3rd and 5th order moments by setting $E_{1,1,1} = E_{1,1,3} = 0$. Plugging in all these values and decomposing A_1 will give us a prototype matrix for the actual σ-points which can be used to generate these sets.

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathcal{S}^{\mathrm{T}}$$

At first blush it would appear that this set of σ-points is not usable because it has weights of 0 but we can mix eigenvectors with the same eigenvalue together with the mixing matrix[7] T.

[7] Any mix can be used but we desire the symmetry given by the chosen T

$$T = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\mathcal{S} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

$$w = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right]$$

$$\mathscr{Y}_1 = [1, 1, -1, -1]$$

$$\mathscr{Y}_2 = [1, -1, 1, -1]$$

This set does not have any flexibility in the $E_{1,1,2,2}$ moment so we will need to correct the stacking issue, that the 1D moments are adding for each stack, in a different way. To fix this we introduce a 1D set which mirrors this set in 1D moments and invert the weight vector. The inversion of the weight vector makes this mirror set undo all 1D contributions, including covariance $E_{1,1}$. This leaves the combined set with a clean set $E_{1,1,2,2} = 1, E_{i,j} = E_{i,j,k} = E_{i,j,k,\ell} = 0$. We happen to have already found a canceling set, which has $E_{1,1,1} = 0, E_{1,1,1,1} = 1, E_{1,1,1,1,1} = 0$, when we found the 1D set for 2 points, the Order 3 set.

$$w = \left[\frac{1}{2}, \frac{1}{2}\right]$$

$$\mathscr{Y}_1 = [1, -1]$$

Lastly we need to reintroduce all the correct 1D moments, which is easily done by stacking in a copy of the Order 5 set.

$$w = \left[\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right]$$

$$\mathscr{Y}_1 = \left[0, \sqrt{3}, -\sqrt{3}\right]$$

All in all the composite set is shown in Equation 5.4.6 with each of the different components, 0 point correction term, mixed moment (black), 1D cancellation (red), and 1D correction (blue) subsets horizontally concatenated, requiring $4\binom{d}{2} + 4d + 1$ points.

$$\left[\begin{array}{c|cccc|c|cccc|c|cccc|c|cc|cc|cc|cc|cc}
1-\binom{d}{2}+d\,(d-1)+d2/3-d & 1/4 & 1/4 & 1/4 & 1/4 & \cdots & 1/4 & 1/4 & 1/4 & 1/4 & \cdots & 1/4 & 1/4 & 1/4 & 1/4 & \cdots & -d-1/2 & d-1/2 & -d-1/2 & d-1/2 & 1/6 & 1/6 & \cdots & 1/6 & 1/6 \\
0 & 1 & 1 & -1 & -1 & & 1 & 1 & -1 & -1 & & 0 & 0 & 0 & 0 & & 1 & -1 & 0 & 0 & \sqrt{3} & -\sqrt{3} & & 0 & 0 \\
0 & 1 & -1 & 1 & -1 & & 0 & 0 & 0 & 0 & & 1 & 1 & -1 & -1 & & 0 & 0 & 0 & 0 & 0 & 0 & & 0 & 0 \\
\vdots & \vdots & & & \vdots & & \vdots & & & \vdots & & \vdots & & & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & & 1 & -1 & 1 & -1 & & 1 & -1 & 1 & -1 & & 0 & 0 & 1 & -1 & 0 & 0 & & \sqrt{3} & -\sqrt{3}
\end{array}\right] \tag{5.4.6}$$

5.5 Results of Using Different Sets

The implicit assumption of this method is that given a Gaussian random variable $x \sim N$ the transformed random variable, $y = f(x)$, will be well approximated with the Gaussian density inferred from the transformed σ-points, or *well enough* for our purposes. The fact is that when a Gaussian random variable is passed through a nonlinear function it is almost never the case that the resulting random variable is Gaussian. If we consider the random variable $y = x^2$, $x \sim N(0,1)$,the result $y \sim \chi^2$ is not Gaussian. For the EKF of the previous chapters we assume that the transform is approximately linear about the expected value and use this to predict the the resulting Gaussian. Given the benefit of a small offset, $\bar{x} = 1/10$, so the covariance doesn't collapse the linearized model predicts the result will be $y = x^2 \approx \bar{x}x \sim N\left(0, (1/10)^2\right)$. While it is true that the result is non Gaussian we can approximate it as the Gaussian which shares the first 2 moments, $N(1,2)$. The actual probability density functions for these random variables can see in Figure 5.5.1. We will examine ability of the different generated σ-points to correctly estimate the resulting first and second order moments of densities generated by different polynomial functions and the performance benefit of these estimates in a pair of exemplars.

5.5.1 Example of Polynomial Functions

We will start by restricting the nonlinear functions to polynomials because the result's true mean and covariance are easily analytically calculated. To demonstrate the different sets and their varying capabilities to model polynomial functions we can set up a base random variable,

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \sim N\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right).$$

We transform this random variable by a one dimensional polynomial function $f(x)$ and compute the covariance of $\begin{bmatrix} x_0 \\ x_1 \\ f(x_0,x_1) \end{bmatrix}$, which demonstrates both the set's ability to estimate both the variable $y \sim f(x)$ and the correlation between this random variable and the the original states (which would be important for observation functions). It should be clear that the order of the polynomial determines the order of

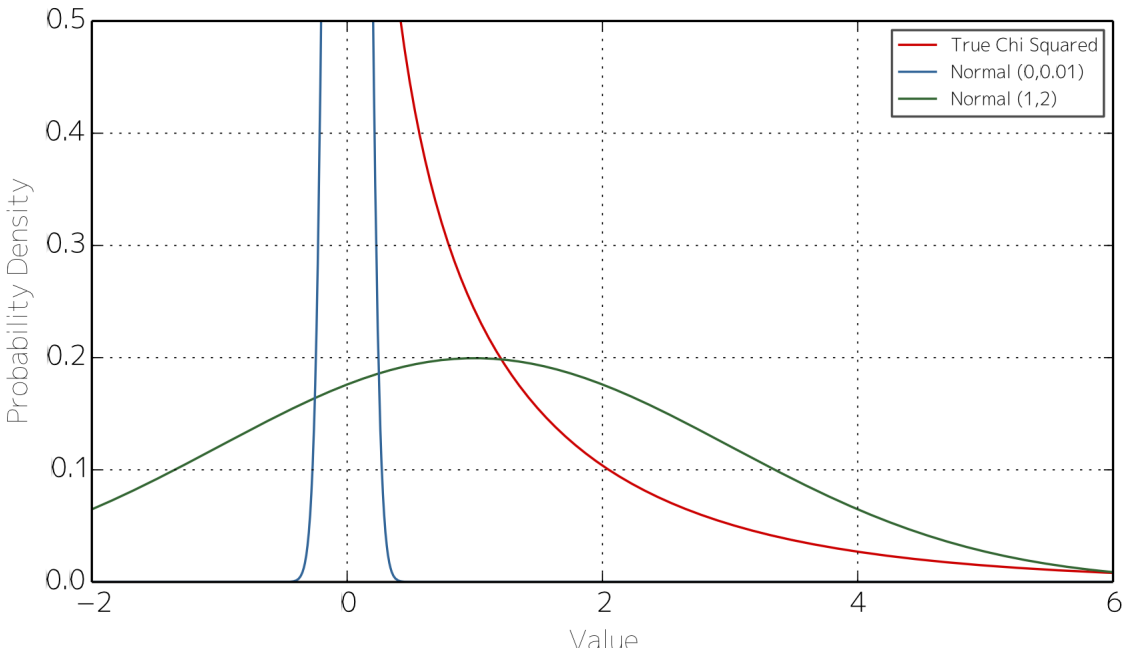

Figure 5.5.1: Probability Density Function for χ^2 and $N(1,2)$

the set needed to estimate the mean and its square for the covariance[8]. For example, $f(x) = x_0^2$ is a second order polynomial so to correctly estimate the mean, $E(f(x)) = E\left(x_0^2\right) = E_{0,0}$, we will need to match moments of the second order (covariance) and to estimate covariance, $E\left(\left(f(x) - E(f(x))\right)^2\right) = E\left(\left(x_0^2 - E_{0,0}\right)^2\right) = E_{0,0,0,0} - E_{0,0}$, we will need to match up to 4th order moments. We can see the result of the different estimation techniques summarized in Table 5.2 where we can see all the sets correctly estimate the mean but only the higher order sets correctly estimate the covariance. Likewise Table 5.3 demonstrates increase in difficulty of the function x_0^4 which will require 4rd order moments for mean and 8th for covariance estimates. Finally we have an example with the function $f(x) = x_0 x_1$, here the mean again only requires the 2nd order moments but the covariance will require a non 1D 4th order moment, specifically the cross moment $E_{1,1,2,2}$ which only the full 5th order set, O5f, meets provides the necessary capabilities as shown in Table 5.4 where only this set correctly estimates the covariance.

[8]Given the polynomial $\mathcal{P} = \sum_k a_k x_0^k$ its expected value, $E(\mathcal{P})$, is the given by $\sum_k a_k E_{0^{\oplus k}}$, where $0^{\oplus k}$ is the k dimensional index with all zeros, $0^{\oplus 4} = (0,0,0,0)$ so is dependent on k^{th} order moments. The variance $E\left(\left(\mathcal{P} - E(\mathcal{P})\right)^2\right)$ can be expanded and has a highest order term $E\left(a_k x_0^k a_k x_0^k\right) = E_{0^{\oplus 2k}}$. This becomes more complicated when $\mathcal{P}$ is a multivariate polynomial but the same properties hold.

Function	$[x_0, x_0^2]$	
Method	Mean	Covariance
Truth	$[1, 2]$	$\begin{pmatrix} 1 & 2 \\ 2 & 6 \end{pmatrix}$
10000 Sample Point	$[0.989, 2.00]$	$\begin{pmatrix} 1.02 & 2.01 \\ 2.01 & 6.03 \end{pmatrix}$
Linear (EKF)	$[1.0, 1.0]$	$\begin{pmatrix} 1.0 & 4.0 \\ 4.0 & 16.0 \end{pmatrix}$
UKF (Sim)	$[1.0, 2.0]$	$\begin{pmatrix} 1.0 & 2.0 \\ 2.0 & 4.5 \end{pmatrix}$
UKF (O5)	$[1.0, 2.0]$	$\begin{pmatrix} 1.0 & 2.0 \\ 2.0 & 6.0 \end{pmatrix}$
UKF (O9)	$[1.0, 2.0]$	$\begin{pmatrix} 1.0 & 2.0 \\ 2.0 & 6.0 \end{pmatrix}$
UKF (O5f)	$[1.0, 2.0]$	$\begin{pmatrix} 1.0 & 2.0 \\ 2.0 & 6.0 \end{pmatrix}$

Table 5.2: Results for x_0^2

Function	$[x_0, x_0^4]$	
Method	Mean	Covariance
Truth	$[1, 10]$	$\begin{pmatrix} 1 & 16 \\ 16 & 664 \end{pmatrix}$
10000 Sample Point	$[0.989, 10.0]$	$\begin{pmatrix} 1.02 & 15.9 \\ 15.9 & 628.0 \end{pmatrix}$
Linear (EKF)	$[1.0, 1.0]$	$\begin{pmatrix} 1.0 & 32.0 \\ 32.0 & 1020.0 \end{pmatrix}$
UKF (Simp)	$[1.0, 8.5]$	$\begin{pmatrix} 1.0 & 10.0 \\ 10.0 & 128.0 \end{pmatrix}$
UKF (O5)	$[1.0, 10.0]$	$\begin{pmatrix} 1.0 & 16.0 \\ 16.0 & 418.0 \end{pmatrix}$
UKF (O9)	$[1.0, 10.0]$	$\begin{pmatrix} 1.0 & 16.0 \\ 16.0 & 664.0 \end{pmatrix}$
UKF (O5f)	$[1.0, 10.0]$	$\begin{pmatrix} 1.0 & 15.1 \\ 15.1 & 418.0 \end{pmatrix}$

Table 5.3: Results for x_0^4

Function	$[x_0, x_1, x_0x_1]$	
Method	Mean	Covariance
Truth	$[1, 1, 1]$	$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 3 \end{pmatrix}$
10000 Sample Point	$[0.989, 0.986, 0.971]$	$\begin{pmatrix} 1.02 & -0.004 & 1.02 \\ -0.004 & 0.993 & 0.992 \\ 1.02 & 0.992 & 3.07 \end{pmatrix}$
Linear (EKF)	$[1.0, 1.0, 1.0]$	$\begin{pmatrix} 1.0 & 0.0 & 2.0 \\ 0.0 & 1.0 & 2.0 \\ 2.0 & 2.0 & 8.0 \end{pmatrix}$
UKF (Simp)	$[1.0, 1.0, 1.0]$	$\begin{pmatrix} 1.0 & 0.0 & 1.71 \\ 0.0 & 1.0 & 1.0 \\ 1.71 & 1.0 & 3.91 \end{pmatrix}$
UKF (O5)	$[1.0, 1.0, 1.0]$	$\begin{pmatrix} 1.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 2.0 \end{pmatrix}$
UKF (O9)	$[1.0, 1.0, 1.0]$	$\begin{pmatrix} 1.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 2.0 \end{pmatrix}$
UKF (O5f)	$[1.0, 1.0, 1.0]$	$\begin{pmatrix} 1.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 3.0 \end{pmatrix}$

Table 5.4: Results for x_0x_1

5.5.2 Fading Channel Example

To see the impact this might have on a realistic example consider the following problem based on Rayleigh fading. We have a signal level of x_0 which we observe through three separate channels with responses, x_1, x_2, x_3 and noise v_1, v_2, v_3.

$$h(x, v) = \begin{bmatrix} x_1x_0 + v_0 \\ x_2x_0 + v_1 \\ x_3x_0 + v_2 \end{bmatrix}$$

The signal level is static and the channel gains fade to zero and but are driven by process noise $v_{1,}v_2, v_3$.

$$f(x, w) = \begin{bmatrix} x_0 \\ \frac{x_1}{2} + w_0 \\ \frac{x_2}{2} + w_1 \\ \frac{x_3}{2} + w_2 \end{bmatrix}$$

The random variables are defined as

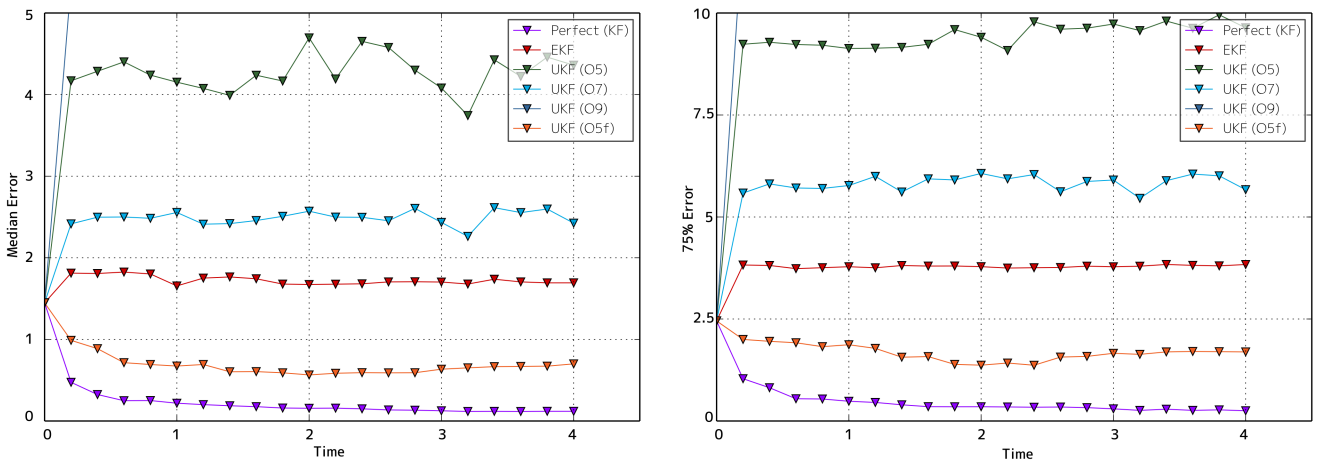

Figure 5.5.2: Fading Example Median and 75% x_0 Error Plots

$$x_{[0]} \sim N\left(\begin{bmatrix} 0 \\ 0.0964 \\ 0.131 \\ -0.171 \end{bmatrix}, \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0.04 & 0 & 0 \\ 0 & 0 & 0.04 & 0 \\ 0 & 0 & 0 & 0.04 \end{bmatrix}\right)$$

$$w \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.04 \end{bmatrix}\right)$$

$$v \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.0025 & 0 & 0 \\ 0 & 0.0025 & 0 \\ 0 & 0 & 0.0025 \end{bmatrix}\right)$$

This problem setup has a linear update step with a nonlinear observation which has a multivariate term. We have already shown the advantage of higher order sets in correctly estimating the moments after transforms but ultimately we are concerned with filter state estimation performance, not their theoretic correctness. To this end, we can examine the different filters median and 75% performance on this problem in Figure 5.5.2. Aligning with our intuition, because the observation function includes as multivariate polynomial the O5f set performs better than the other 1D sets. Ultimately it still makes an incorrect Gaussian assumption which is why a perfectly linearized model still out performs it. The UKF (Simp) filter diverges to failure in some runs causing its exclusion from the graph.

For Rayleigh fading the sign of the channel factor x_1 should be stripped as negative channel responses

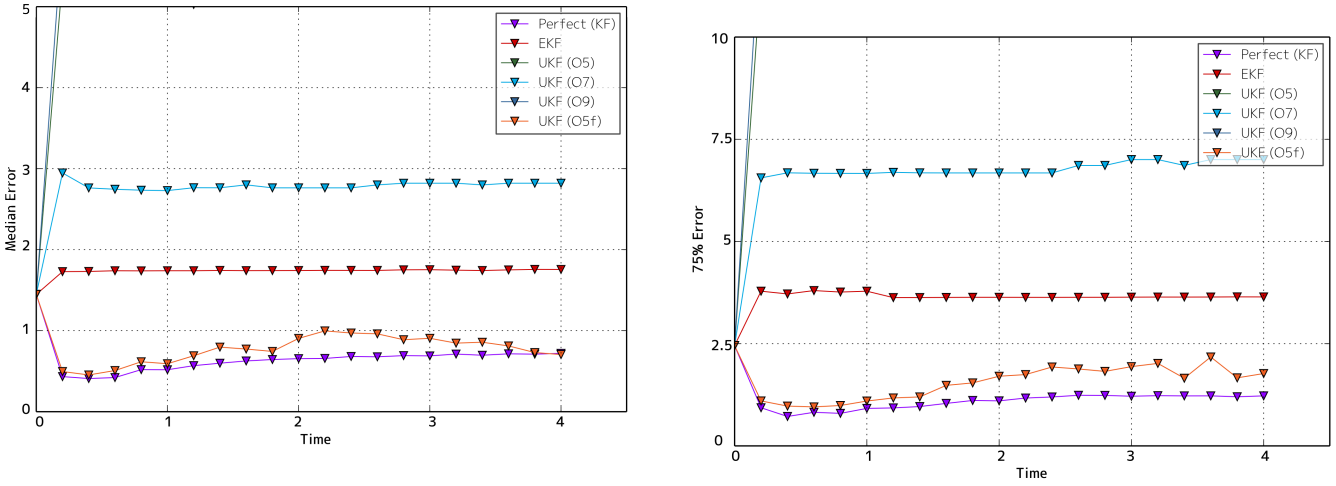

Figure 5.5.3: Fading Example (with abs) Median and 75% x_0 Error Plots

don't have physical meaning in the classical model.

$$h(x,v) = \begin{bmatrix} |x_1|\, x_0 + v_0 \\ |x_2|\, x_0 + v_1 \\ |x_3|\, x_0 + v_2 \end{bmatrix}$$

This new observation function is no longer a simple polynomial so it is no longer easy to predict the effectiveness of the different σ-point sets but the error plots in Figure 5.5.3 tell a very similar story, with the main difference being that the perfectly linearized KF performs markedly worse.

5.5.3 Angle Tracking Example

Another example originates from tracking angles. Consider tracking an angel, θ, through the pair of trigonometric functions $\cos(\theta), \sin(\theta)$.

$$
\begin{aligned}
x &= \begin{bmatrix} \cos(\theta[k]) \\ \sin(\theta[k]) \\ \dot{\theta}[k] \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \\
x[k+1] &= \begin{bmatrix} \cos\left(\theta[k] + \Delta t \dot{\theta}\right) \\ \sin\left(\theta[k] + \Delta t \dot{\theta}\right) \\ \dot{\theta}[k] + w[k] \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta[k]) \cos\left(\Delta t \dot{\theta}[k]\right) - \sin(\theta[k]) \sin\left(\Delta t \dot{\theta}[k]\right) \\ \cos(\theta[k]) \sin\left(\Delta t \dot{\theta}\right) + \sin(\theta) \cos\left(\Delta t \dot{\theta}\right) \\ \dot{\theta}[k] + w[k] \end{bmatrix} = \begin{bmatrix} x_0 \cos(\Delta t x_2) - x_1 \sin(\Delta t x_2) \\ x_0 \sin(\Delta t x_2) + x_1 \sin(\Delta t x_2) \\ x_2 \end{bmatrix}
\end{aligned}
$$

This setup is a bit contrived for a single angle but when tracking a multidimensional rotation something like this becomes necessary. The system is observed by a simple linear observation on the x_1 state, $\sin(\theta)$, and has the following initial conditions.

$$
\begin{aligned}
x[0] &\sim N\left(\begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \end{bmatrix}, \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 \\ 0.0 & 0.0 & 0.3 \end{bmatrix} \right) \\
w &\sim N(0, 0.16) \\
y[k] &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x[k] + v[k] \\
v &\sim N(0, 0.25)
\end{aligned}
$$

Although this problem is set up like a rotation tracking problem we will, at first, be tacking it based purely on the nonlinear setup shown, i.e. we do not require that the states x_0 and x_1 maintain the trigonometric identity $\cos^2(\theta) + \sin^2(\theta) = 1$. The performance of the different σ-point sets is shown in Figure 5.5.4, showing 3 plots with the 50% 75% and 90% errors. We can see there is an improvement in 75% and 90% performance for the O5f set but otherwise all the σ-point sets perform similarly.

Additionally we can construct the transform which takes into account a normalization step, in order to maintain the trigonometric identity. The propagation function becomes,

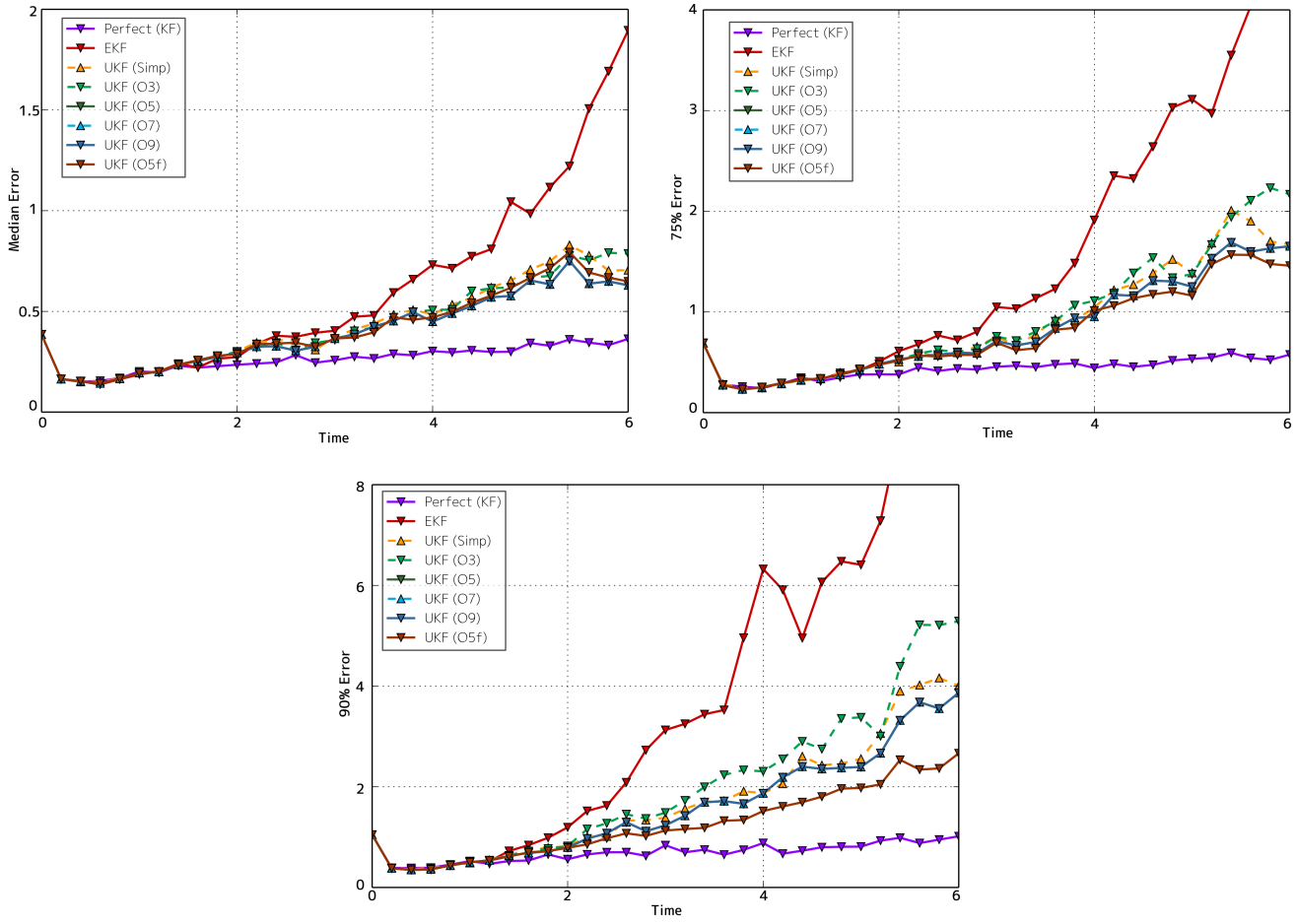

Figure 5.5.4: Filter Performance on Example Related to Rotation Tracking

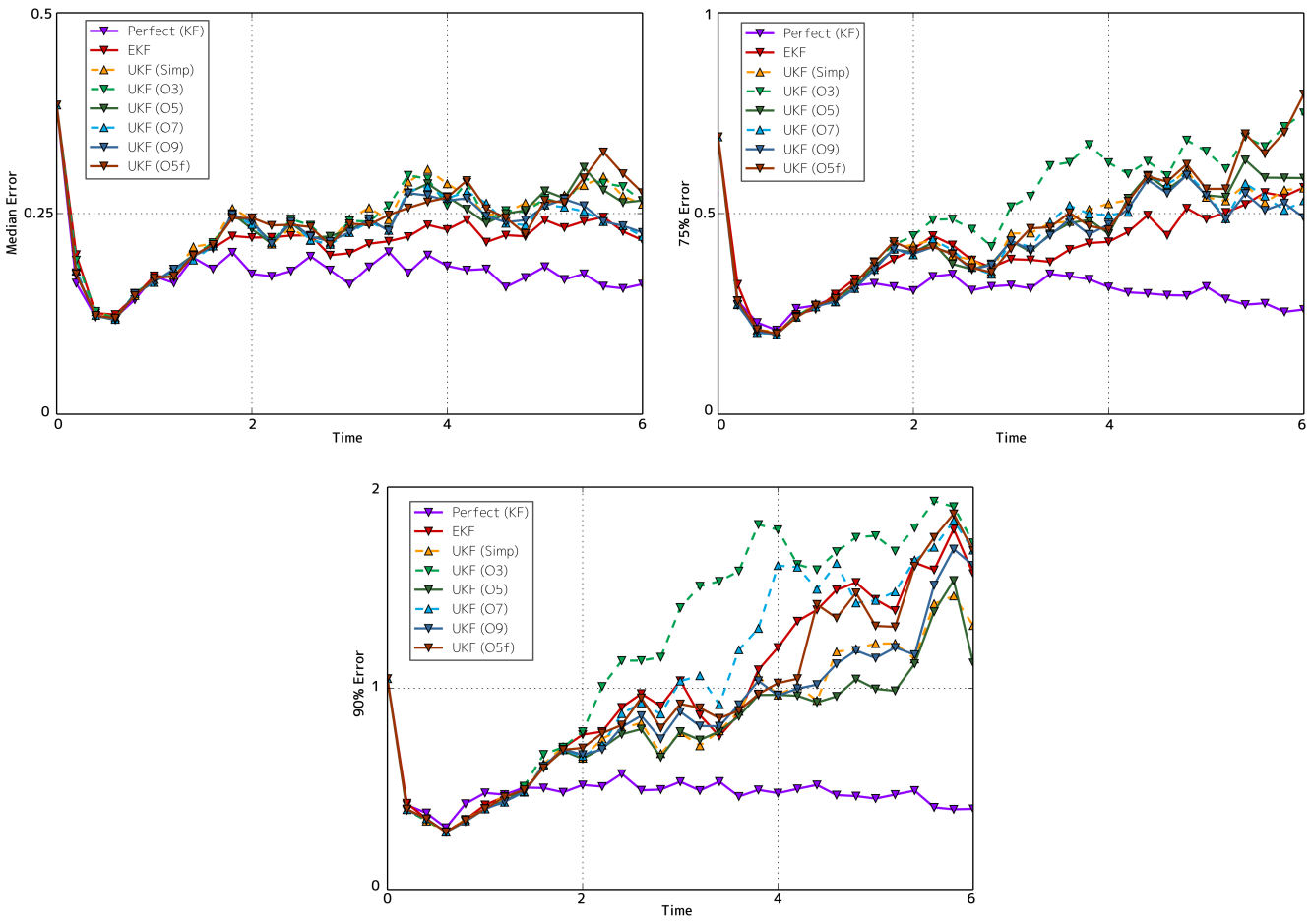

Figure 5.5.5: Filter Performance on Example Related to Rotation Tracking

$$x\left[k+1\right] = \begin{bmatrix} \frac{x_0 \cos(\Delta t x_2) - x_1 \sin(\Delta t x_2)}{\sqrt{x_0^2 + x_1^2}} \\ \frac{x_0 \sin(\Delta t x_2) + x_1 \sin(\Delta t x_2)}{\sqrt{x_0^2 + x_1^2}} \\ x_2 \end{bmatrix}.$$

The results for this setup can be seen in Figure 5.5.5, where we can see that the most of the sets are struggling to out perform the simple EKF's linear model. This is counterintuitive given that we have constructed this method with the intent of improving performance in nonlinear cases. This is an effect that we will be looking into in the next section.

Chapter 6

Sigma Point / Unscented Smoothing

> *"...it is easy not to believe in monsters, considerably more difficult to escape their dread and loathsome clutches."*
>
> - Stanisław Lem, *The Cyberiad*

As well as σ-point methods perform with respect to estimating the densities that result from a Gaussian process passing through a polynomial transforms, not all problems are as simple. In this chapter we will be exploring problems which, although they seem simple, will challenge the UKF. We will then explore the cause of the problems and some ways of adapting the Unscented/σ-point transforms to improve outcomes.

6.1 Sigma Point Scaling

6.1.1 Wrapped Measurement Model

Let us start by reexamining the weather-vane problem from Chapter 3 where we have an object oscillating back to a forced, randomly moving equilibrium. We introduce one key difference in the observation function. Instead of the simply measuring the rotation of the weather-vane we will only observe the angle it makes relative to zero. This subtle difference makes all values $z + 2\pi$ result in the same observation. The system is described in Equation 6.1.1, where the propagation step is linear and the observation function h_o is the re-centering of the measurement function, h, about the measurement, z. The re-centering pushes the discontinuities as far away from the measurement as possible.

$$
\begin{aligned}
x &= \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \\
&\sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} (2\pi)^2 & 0 \\ 0 & \left(\frac{\pi}{2}\right)^2 \end{bmatrix} \right) \\
F &= \begin{bmatrix} 0.809 & 0.0935 \\ -3.69 & 0.808 \end{bmatrix} \\
L &= \begin{bmatrix} 0.191 \\ 3.69 \end{bmatrix} \\
Q &= \begin{bmatrix} 0.0001 \end{bmatrix} \\
h(x, v) &= (\theta + v \mod 2\pi) - \pi \\
h_o(x, v, z) &= (\theta + v - z \mod 2\pi) - \pi + z \\
R &= \begin{bmatrix} \left(\frac{\pi}{4}\right)^2 \end{bmatrix}
\end{aligned}
\tag{6.1.1}
$$

We can examine an instance of this problem with ten steps of propagation and a measurement. Just as in Chapter 4 we include a perfectly linearized filter as a baseline. We can see the results of the EKF and various UKFs in Figure 6.1.1, noting that the EKF achieves an identical result to the perfectly linearized KF, an indication that it is performing nearly optimally, and the UKFs do not, which is not a good sign.

To explore why the UKF filters seem to be having trouble we only need to look at the first measurement made. Shown in Figure 6.1.2 is a view into the mechanics of the different filters. The observed value $h_o(\theta, v_0, z_0)$ is plotted against the true state value θ. The measurement is shown as a purple dot, at $\theta = -1.77$ and measured value $z = 0.93$, and acts as the center of the modular calculation of the observation function. Given our estimate of the state and measurement noise we generate many random samples and observe them all through h_0 to estimate the transform's ensemble statistics. These points are shown in gray and their ensemble mean is a gray 'x' and covariance is shown as a dark gray ellipse. From these, random samples, we can see that both the assumption of normality is breached and the actual measurement is not expected to correlate well with the actual value of θ. The EKF, which linearizes about the assumption that both θ and v are zero, on the other hand predicts a very strong correlation, indicated in the tighter red ellipse. We have been extremely lucky[1] and this assumed covariance is correct allowing

[1] I have chosen the random seed maliciously to create the effect. There is no way for the actual filter to know that the measurement was in this particular alias cell, the true state θ could have just as easily have been $1.77 + 2\pi$ and the EKF's correction would have been in the completely wrong direction. In other cases, as we shall see later, this sort of thing can cause to the EKF to work very well sometimes and poorly at others.

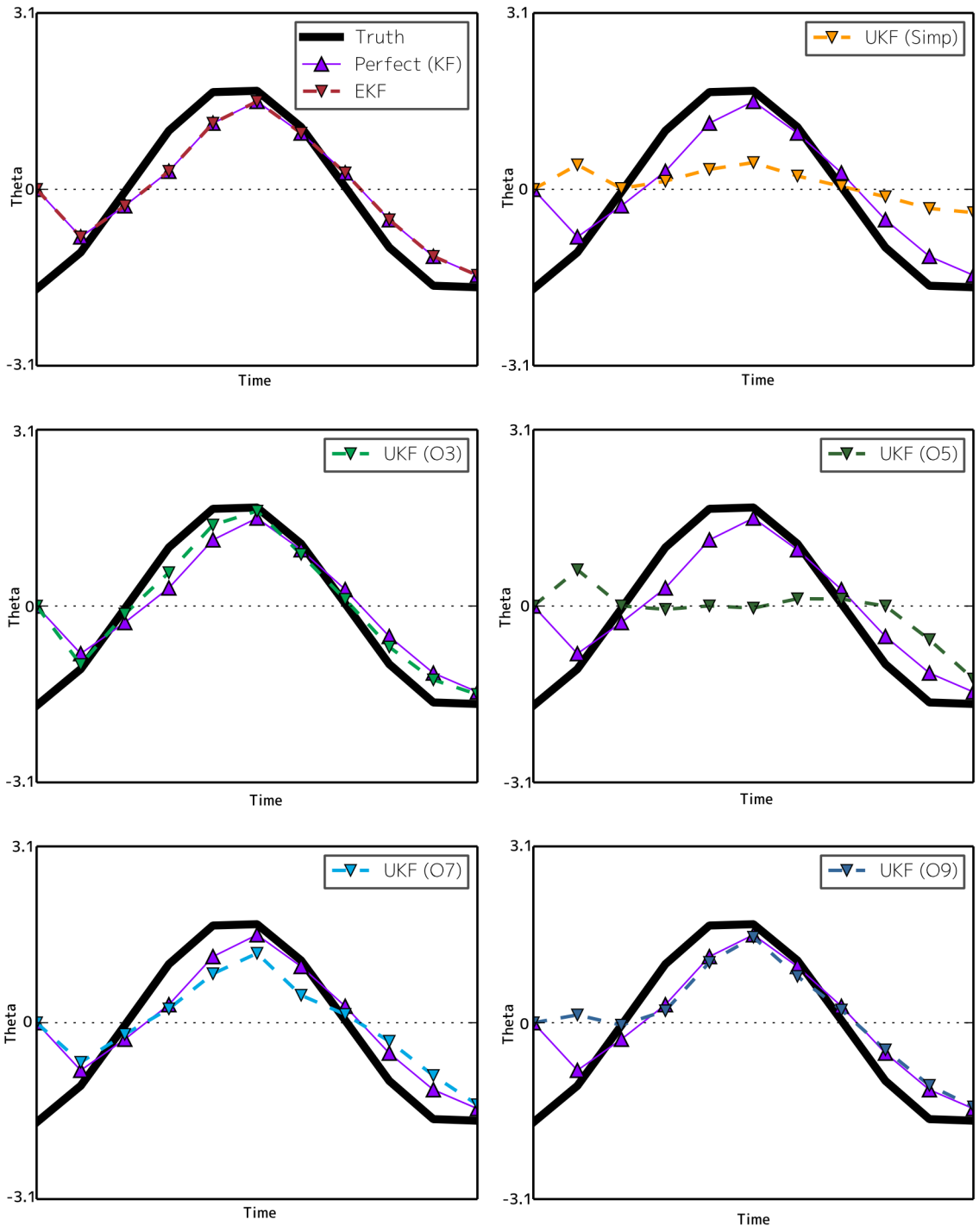

Figure 6.1.1: Filter Comparison on Periodic Measurements

the EKF to extract the maximum amount of information from the measurement as possible. For each of the UKFs we can see the various evaluated σ-points and the resulting mean and covariance. For these filters the wrapping causes somewhat less apt estimates.

There are two things happening which cause the UKF to appear, in this case, to be performing poorly. The first is that the nonlinearity introduced by the discontinuities is not well approximated by a polynomial of small enough order for any of the σ-points sets to accurately model. We can see this in the difference between the mean and covariance estimates from the ensemble and the σ-point estimates. The second *issue* is that the actual covariance between the measurement and state value is very small, as can be seen in the ensemble covariance. The EKF on the other hand is predicting a strong covariance and seems to be do, paradoxically, very well. The EKF's assumption, that there is no wrapping function, leads to near perfect results when the truth value is within the same aliasing cell, but disastrous when it is not. This is the inherent trouble with this problem, at its core there is no linear update which will correctly take into account both the probable 1 to 1 correlation of the measurement, when it lies within the same wrapping cell, and the possible reverse if it does not.

To examine each filter's statistical performance in this scenario we can examine their errors in estimating the θ state across 200 different runs, as shown in Figure 6.1.3 where both the 50th and 75th quantile are plotted. For most cases we might expect that the wrapping discontinuity will not come into effect, which is shown in the 50th Quantile plot, where the EKF does very well. Even on the 75 Quantile plot the EKF seems to be doing the best of the non perfect filters. The Ensemble filter, because it is always taking into account the possibility that the state is in one of the wrapped cells seems to be paralyzed by this for the majority of cases, making almost no corrections based on the measurements, a strategy which does not seem to be paying off until we look at the 90th Quantile plot in Figure 6.1.4. The σ-point filters land somewhere in the middle of the two extremes, trading off some of the 90% robustness of the Ensemble for improved performance in the majority of scenarios. in many applications we can accept the potential for error as long as the filter performs well most of the time. The next question is whether we can adapt the σ-point method to perform more like the EKF when that is desired.

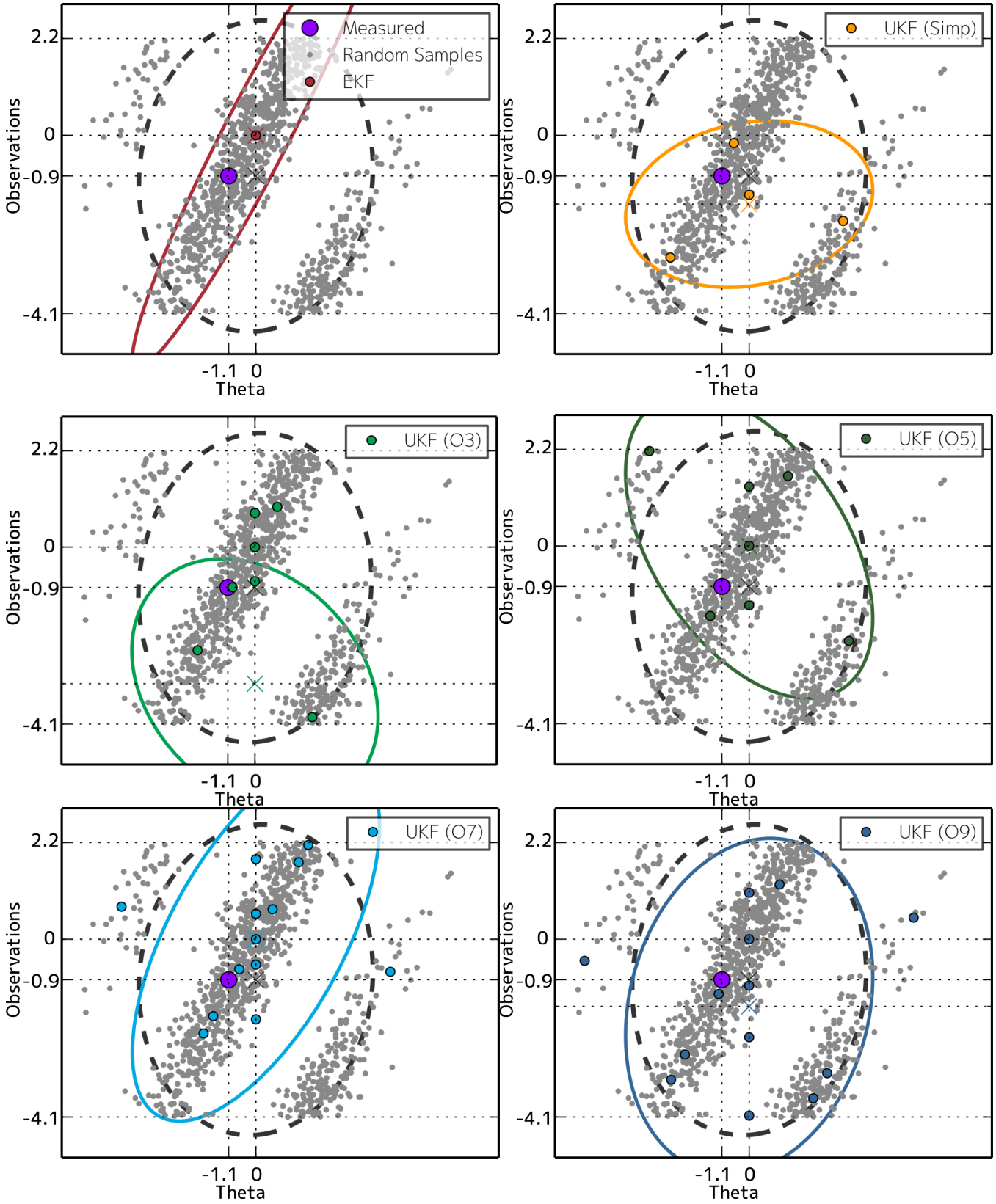

Figure 6.1.2: Periodic Measurement with Filters' Estimated Mean and Covariance

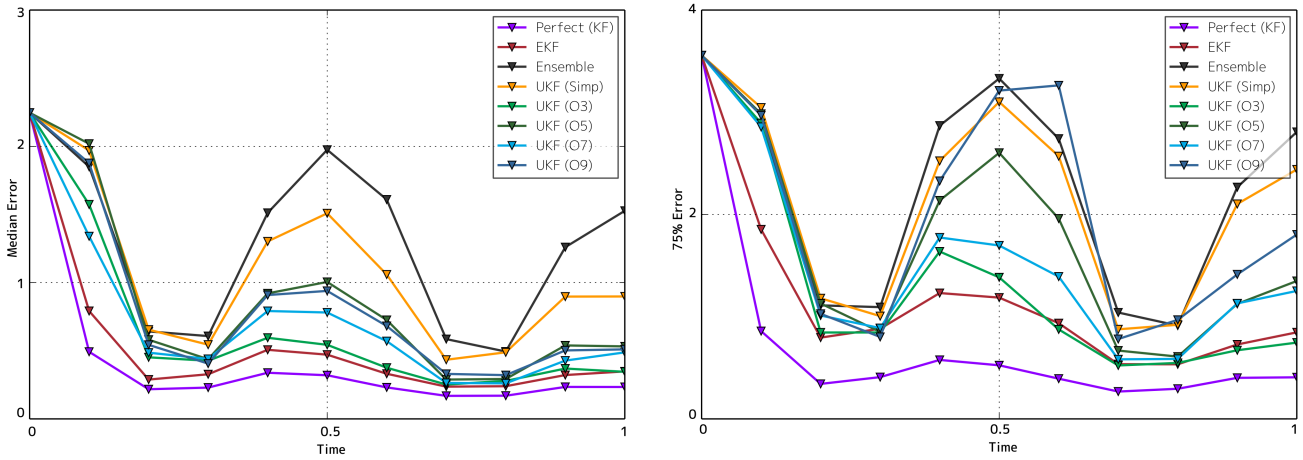

Figure 6.1.3: 50th and 75th Quantile Plots of State Errors for Wrapping Example

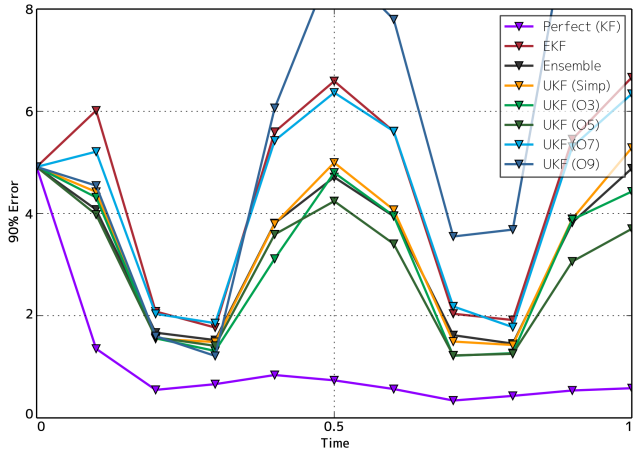

Figure 6.1.4: 90th Quantile Plot of State Error for Wrapping Example

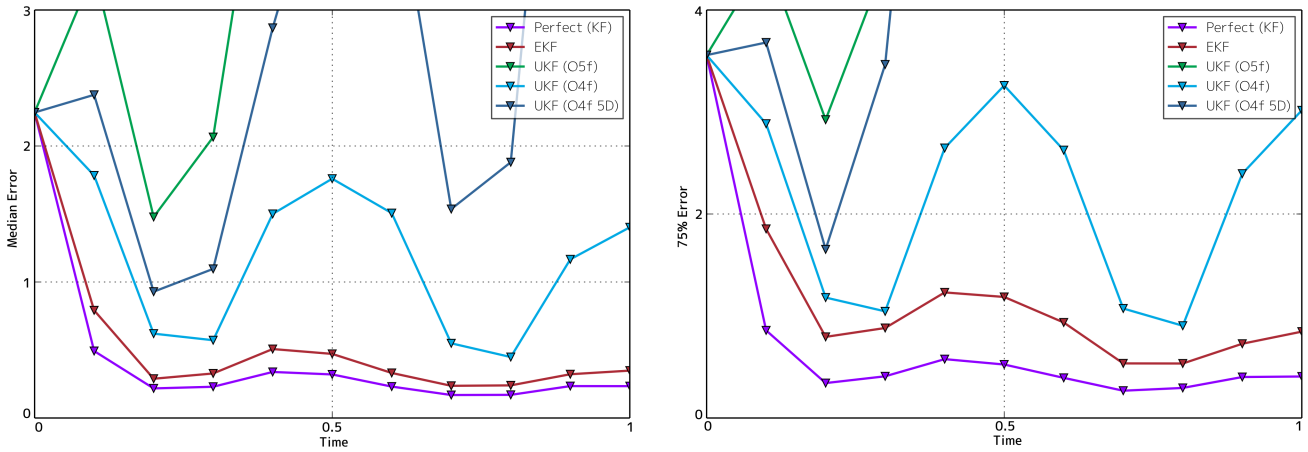

Figure 6.1.5: Performance of Filters with Negative Weights

Negative Weights

Before we move on we need to address the results from the O5f set. The performance in the face of wrapping for σ-point methods is dramatically worse for this set as shown in the 50 and 75 percentile plots of Figure 6.1.5. Examining a similar multivariate moment set, O4f, which was generated using a slightly different strategy, we can see that this not an issue introduced by the inclusion the E_{1122} moment. One of the major differences between these two sets, O5f and O4f, is the inclusion of non zero σ-points with negative weights. At 3 dimensions the O4f set, and in fact all the other sets, have only used a negative weight for the 0 point, O5f, on the other hand, has many points from the canceling mirror set $\begin{bmatrix} -d-1/2 & -d-1/2 \\ 1 & -1 \end{bmatrix}$. The fact that the O4f set does not have any negative nonzero σ-points only holds for relatively small dimension sizes, $d < 5$, at dimension 5 it starts to require a nonzero point with negative weight and when using an O4f designed for a 5D space we can see the inferior performance return.

The problem, as we have alluded to, appears to be related to the nonzero negatively weighted σ-points. The σ-points after mapping are shown in Figure 6.1.6 where we have differentiated those with a negative weight with a star marker. The covariance ellipses associated with the σ-point estimates have been removed because for some of the sets they do not exist. The problem of σ-point estimation when wrapping is present is magnified because when a point we would expect to be in the top right gets wrapped to the bottom left instead of pulling the average down, as it would with a positive weight it instead pushes it up. This gives us means outside the region supported and leads to nonphysical means and covariances. When we use negatively weighted points we are relying on symmetry to balance them with the positive ones. The 0 vector causes us less issue because it is balanced by all the positive points and is centrally

located making it less likely to be skewed.

The accuracy of σ-point methods in estimating nonlinear transforms is theoretically sound only for appropriately low order polynomials, as we saw in the last chapter. For higher order polynomials the incorrect higher order moments will cause errors in the approximation. This way of thinking about σ-point failure does not lead to any intuition about the how they fail. I suggest we examine how negative weights lead to two unintuitive results to form some intuition. The two results of negative weights is the non-convex averaging of points to a mean and the non positive-definiteness of the covariance matrix. First we can examine the non-convexity of the mean. If all the weights are positive then the mean is a convex combination of all the sampled σ-points and lies somewhere in their convex hull. Conversely when we introduce a negative weight this no longer remains the case, i.e. the 'mean' of the σ-points can lie somewhere outside their convex hull. As a consequence of this consider a 1 dimensional observation function which maps all the σ-points into the range of values $[-1, 1]$. With positive weights we know the estimated mean will be somewhere in this range. With negative weights, however, the estimated mean could be outside this range, 3 for example, which seems counterintuitive. The negative weights arose as part of the flexible σ-point construction method we introduced in the last chapter. Consider the O3 sets estimation of the function $f(x) = x_0^2 - 1$ for a standard Gaussian $x \sim N(0, I)$. If the dimension of x is $d = 1$, then we have two points $\begin{bmatrix} \pm 1 \\ 0 \\ \vdots \end{bmatrix}$ which both map to 0 giving us a mean of 0 which is both intuitive and correct. Now instead allow the dimension of the state space to grow $d > 1$ creating a multitude of points of the form $\begin{bmatrix} \vdots \\ 0 \\ \pm 1 \\ 0 \\ \vdots \end{bmatrix}$, all with the same weight as the original two and all of which map to -1. These new points pull the average down to some negative value so we need to introduce the negatively weighted 0 point, which also maps to -1, to push the average back up to the correct mean value, 0. The example is much more complicated for the sets constructed to have correct mixed moments but the result is the same, that negative weights are needed to cancel out the multitude of points from other dimensions which are effectively 0 for functions on some subset of the state vector. The other aspect of this problem is the introduction of non positive-definite components to the covariance matrix. All of the components of the covariance matrix are outer products of the mean subtracted mapped σ-points which with only positive weights guarantees the covariance to be positive-definite. Negative weights introduce negatively-definite components leading to the possibility of non positive-definite covariance estimate. So negative weights are a requirement of the construction methods we have discussed and act as cancellation factors but in non low order polynomial functions they may not properly cancel the terms they were meant to and instead lead to nonphysical estimates. As an alternative construction technique consider some large set of

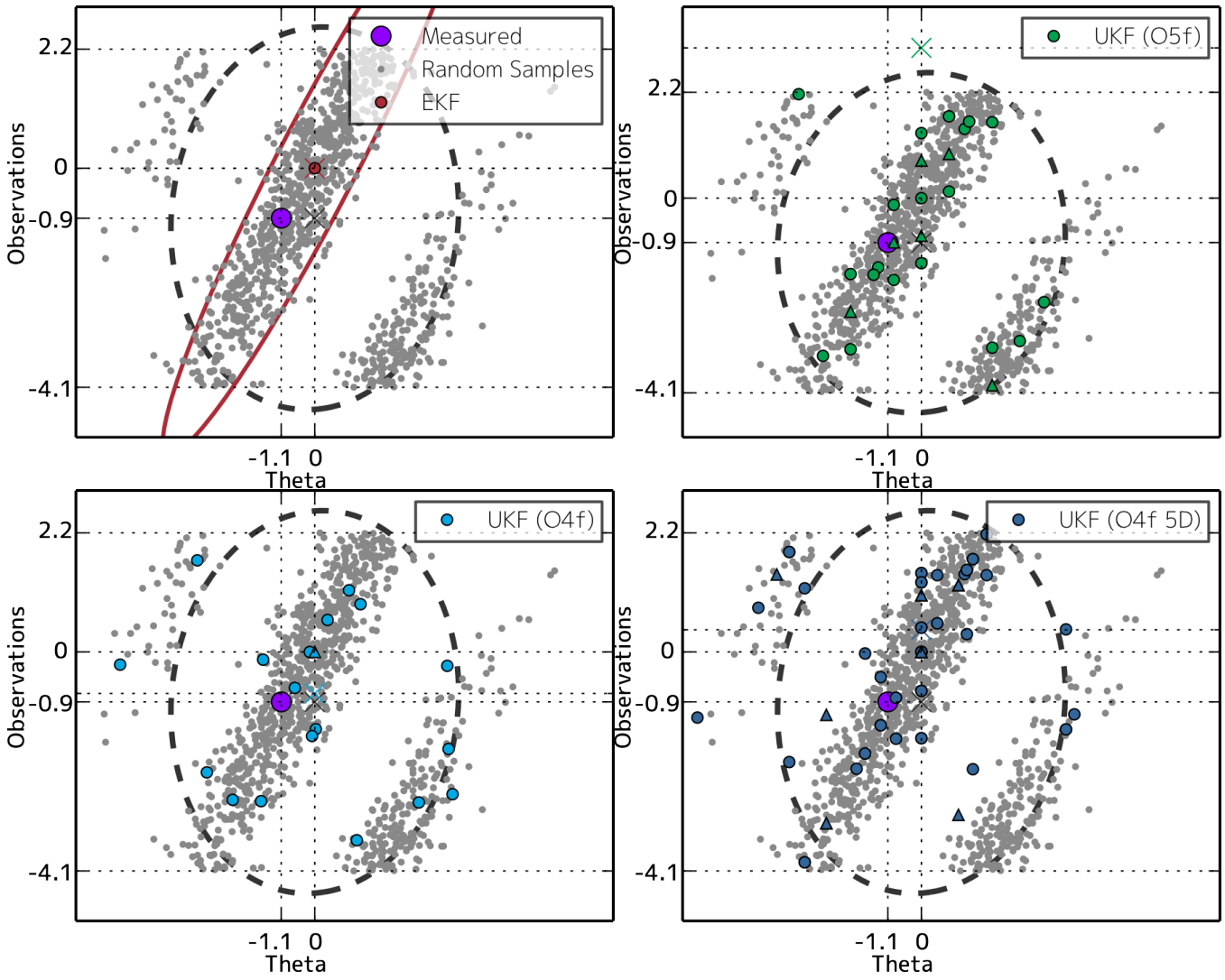

Figure 6.1.6: Periodic Measurement with Filters' Estimated Mean and Covariance

ensemble points all with equal weights. Let these be approximately σ-points which have positive weights and approximate higher order moments dependent on the number of points used. A set of points like this is designed for nothing in particular, which gives it the robustness we expect from the Ensemble method, it does, however, require many more points than the σ-point sets we have been discussing for problems which the σ-points are designed to solve.

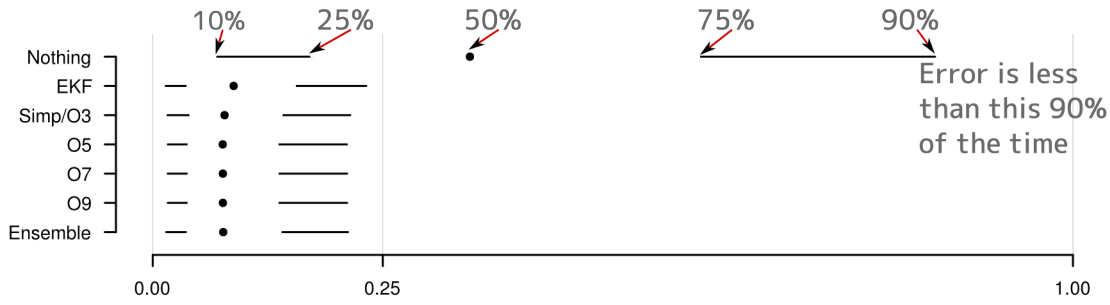

Figure 6.1.7: Quantile Error Plot for $\sigma = 0.5$

6.1.2 Statistical Regions of Approximation and Scaling

As we have seen in the previous section the EKF's linearizion assumption can produce a very good result in problems when the majority of cases lie in the region around the linearization point where the assumption is approximately correct. Let us now consider a case where the statistical distribution of x influences what order of polynomial is needed to well approximate the majority of the distribution.

Consider the single state estimation problem with a single observation.

$$x \sim N\left(0, \sigma^2\right)$$
$$h\left(x, v\right) = \sum_{k=1}^{4} \left(-\frac{x}{k}\right)^k + v$$
$$v \sim N\left(0, 0.001\right)$$

Although the observation function is a high dimensional polynomial, it is by construction well approximated by lower dimensional polynomials within regions about zero. For example within the bound $|x| < 0.575$ the observation function will be well approximated by a linear function. This is the 75% bound for the state distributed with $\sigma = 0.5$ allowing even the EKF to perform well under this scenario. This can be seen in Figure 6.1.7, where we show the distribution of the estimate errors in 5 quartiles $0.1, 0.25, 0.5, 0.75, 0.9$ (a line between the first two, dot for 50th, and a second line between the last two) the NOTHING case refers to the error in state estimate without performing any observation, doing nothing.

If we increase the standard deviation to $\sigma = 2$, the linear model starts to struggle to approximate the function h for the vast majority of the distribution (90% and below). These errors are shown in Figure 6.1.8 where we can start to see the difference between the σ-point methods and the EKF. The EKF's linearization strategy is still working better for the more than 50% of the time compared to the σ-point sets, who's performance is not superior until we get to about the 75% level and above.

As we move to larger variances, $\sigma = 8$, the nonlinearities start becoming more evident even for the

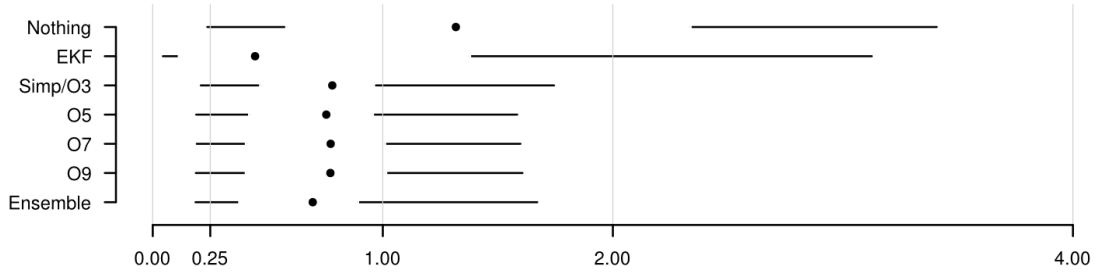

Figure 6.1.8: Quantile Error Plot for $\sigma = 2$

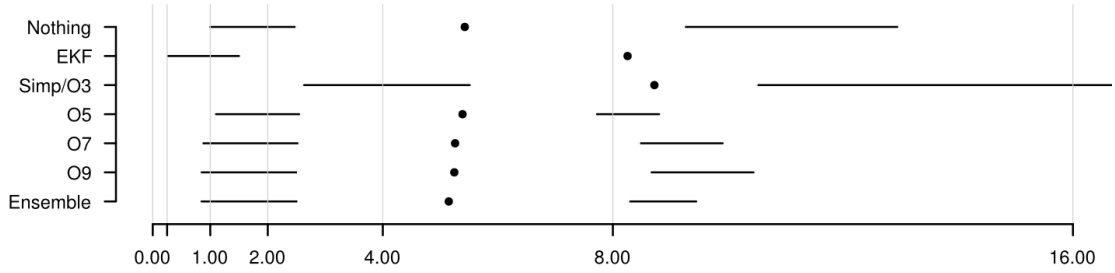

Figure 6.1.9: Quantile Error Plot for $\sigma = 8$

lower quantiles of data as can be seen in Figure 6.1.9. We can see at this point the EKF is doing worse than the no estimator case for the majority of cases but for the 10th and 25th quartiles it is still doing better than the UKFs (other that the Simp/O3 set) which are only outperforming the EKF at the higher quantiles. The complete failure of the UKF using the Simp/O3 points is important because even though it is, arguably, a better estimator (by taking into account up to 3rd order moments) neither the EKF or UKF O3 is capable of correctly estimating the function. But by only considering an infinitely small interval about the expectation, however, the EKF still succeeds for some of the cases whereas the UKF with Simp/O3 set, in its attempt to model a much larger interval of data, simply ends up doing everything poorly.

Clearly in some scenarios it might be better to have a filter which replicates the EKF's local emphasis, either because we want to improve the performance in the lower quartiles at the expense of the higher ones or to accept our inability to model the function for the vast majority of the distribution but still succeed for some of them. In [4] the authors suggest introducing a scaling parameter α to the nonlinear function f to reduce the impact of higher order terms which have gone un-modeled, shown in Equation 6.1.2. For our purposes we would like this parameter to gracefully adapt the UKF between a behavior similar to an Ensemble filter to one of an EKF. In Figure 6.1.10 we can see the quantile performance plot for a UKF as we vary α from 0.1, where performance would be expected to be more like a EKF, to 1, where we expect Ensemble like performance.

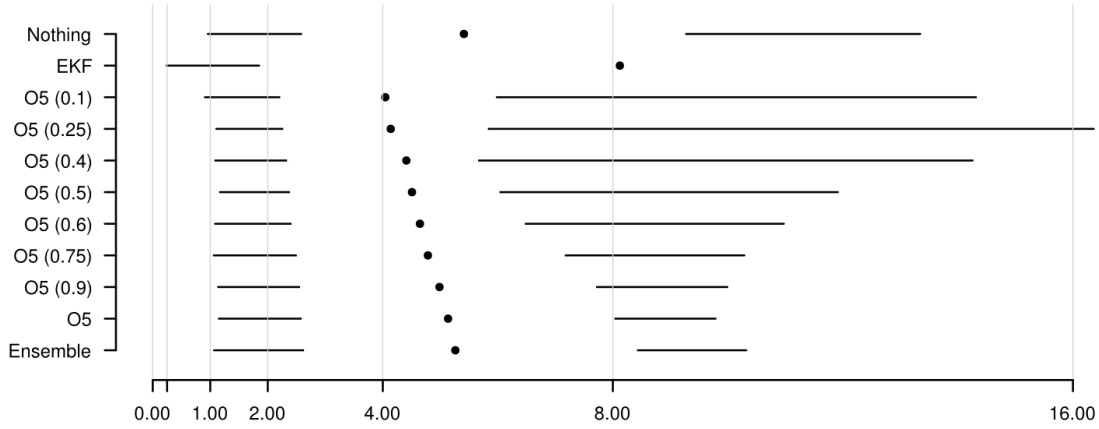

Figure 6.1.10: Quantile Error Plot for $\sigma = 8$ as α Scaling Factor (Indicated in parenthetic values on vertical axis)

$$f(x) \sim \frac{f(\alpha(x - \bar{x}) + \bar{x}) - f(\bar{x})}{\alpha^2} + f(\bar{x}) \tag{6.1.2}$$

Scaling Method

This scaling method does not produce the response we are looking for so I suggest a different one. First let us consider fitting the UKF estimation technique to what we might expect from a linear model. In a linear model we expect that a normal random variable, x, is transformed as follows.

$$\begin{aligned}
x &\sim N(\bar{x}, P_{xx}) \\
w &\sim N(0, P_{ww}) \\
y &= Fx + Lw + u \\
y &\sim N(\bar{y}, P_{yy}) \\
\bar{y} &= F\bar{x} + u \\
P_{yy} &= FP_{xx}F^{\mathrm{T}} + LP_{ww}L^{\mathrm{T}} \\
P_{xy} &= P_{xx}F^{\mathrm{T}} \\
P_{wy} &= P_{ww}L^{\mathrm{T}}
\end{aligned}$$

Using a UKF we estimate the same final parameters, $\bar{y}, P_{yy}, P_{xy}, P_{wy}$, but without the linear matrices F, L. Consider going back and solving for F, L from our covariance matrices.

$$\tilde{F} = \left(P_{xx}^{-1} P_{xy}\right)^{\mathrm{T}}$$
$$\tilde{L} = \left(P_{ww}^{-1} P_{wy}\right)^{\mathrm{T}}$$

Covariance matrices are positive definite so the matrix inverses will exist and be well defined. The only problem with these estimates is that there is no reason to believe that if we reconstructed $\tilde{P}_{yy}$ from them it would match the original UKF estimate, P_{yy}. We know P_{yy} is symmetric by construction so the difference $P_{yy} - \tilde{P}_{yy}$ is also symmetric. I suggest we interpret this matrix as a third noise vector's contribution to the transformation, $n \sim N(0, P_{nn})$, which represents the nonlinear noise as estimated by the the UKF. It should be noted that although P_{nn} is symmetric it is not necessarily positive definite. This gives us the relationship between the two given in Equation 6.1.3.

$$\begin{aligned}
\tilde{F} &= \left(P_{xx}^{-1} P_{xy}\right)^{\mathrm{T}} \\
\tilde{L} &= \left(P_{ww}^{-1} P_{wy}\right)^{\mathrm{T}} \\
P_{nn} &= P_{yy} - \left(\tilde{F} P_{xx} \tilde{F}^{\mathrm{T}} + \tilde{L} P_{ww} \tilde{L}^{\mathrm{T}}\right) \\
u &= \bar{y} - \tilde{F} \bar{x} \\
y &= \tilde{F} x + \tilde{L} w + u + n
\end{aligned} \tag{6.1.3}$$

I should point out that this is simply an interpretation of the UKF's estimates and has not changed any of the actual computations. This does, however, allow us to build a model of the function using a "σ-point" method and then apply that model to a different random variable. Just as we can linearize a function about a different point than the expected one we can now "σ-point" a function about a different random variable than the one we started with. For example we can now scale the input covariance of the UKF by some parameter α, so $\acute{P}_{xx} = \alpha^2 P_{xx}$, use this new covariance to estimate $\tilde{F}, \tilde{L}, P_{nn}$ and then use these to estimate how this function would interact with the original, unscaled, random variable.

$$
\begin{aligned}
\mathfrak{S}\left(f, N\left(\bar{x}, \alpha^2 P_{xx}\right), N\left(0, \alpha^2 P_{ww}\right)\right) &\rightarrow \left(\acute{\bar{y}}, \acute{P_{yy}}, \acute{P_{xy}}, \acute{P_{wy}}\right) \\
\tilde{F} &= \left(\left(\alpha^2 P_{xx}\right)^{-1} \acute{P}_{xy}\right)^{\mathrm{T}} \\
\tilde{L} &= \left(\left(\alpha^2 P_{ww}\right)^{-1} \acute{P}_{wy}\right)^{\mathrm{T}} \\
P_{nn} &= \frac{1}{\alpha^2} \acute{P_{yy}} - \left(\tilde{F} P_{xx} \tilde{F}^{\mathrm{T}} + \tilde{L} P_{ww} \tilde{L}^{\mathrm{T}}\right) \\
&\text{or} \\
P_{nn} &= \acute{P_{yy}} - \alpha^2 \left(\tilde{F} P_{xx} \tilde{F}^{\mathrm{T}} + \tilde{L} P_{ww} \tilde{L}^{\mathrm{T}}\right) \\
u &= \acute{\bar{y}} - \tilde{F} \bar{x} \\
\bar{y} &= \tilde{F} x + u \\
P_{yy} &= \tilde{F} P_{xx} \tilde{F}^{\mathrm{T}} + \tilde{L} P_{ww} \tilde{L}^{\mathrm{T}} + P_{nn} \\
P_{xy} &= P_{xx} \tilde{F}^{\mathrm{T}} \\
P_{wy} &= P_{ww} \tilde{L}^{\mathrm{T}}
\end{aligned}
$$

Using this method for scaling σ-points we get the desired transition between Ensemble and EKF like results as shown in Figure 6.1.11, where we can even see some improvements over the EKF for the value $\alpha = 0.1$. Additionally the scaling has allowed the Simp/O3 set to succeed for quartiles that it can model successfully which can be seen in Figure 6.1.12. Depending on what you are trying to accomplish you may want to use either of the two forms for P_{nn}, the first attempts to scale P_{nn} up for the entire interval and might be a good idea if you are trying to avoid some property of the nonlinear function but still want to model the entire interval. The second attempts to only model the scaled random variables and might be better if you are looking for purely improve lower quartile performance. Both methods approach a simple linearization assumption in the limit of small α for any function where a derivative exists and in this example both are very similar.

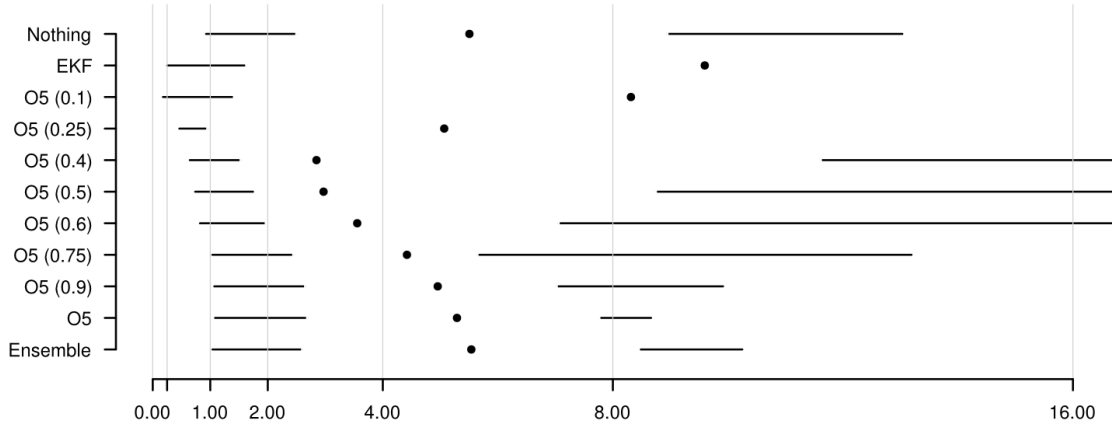

Figure 6.1.11: Quantile Error Plot for $\sigma = 8$ as alternative α varies for O5 set

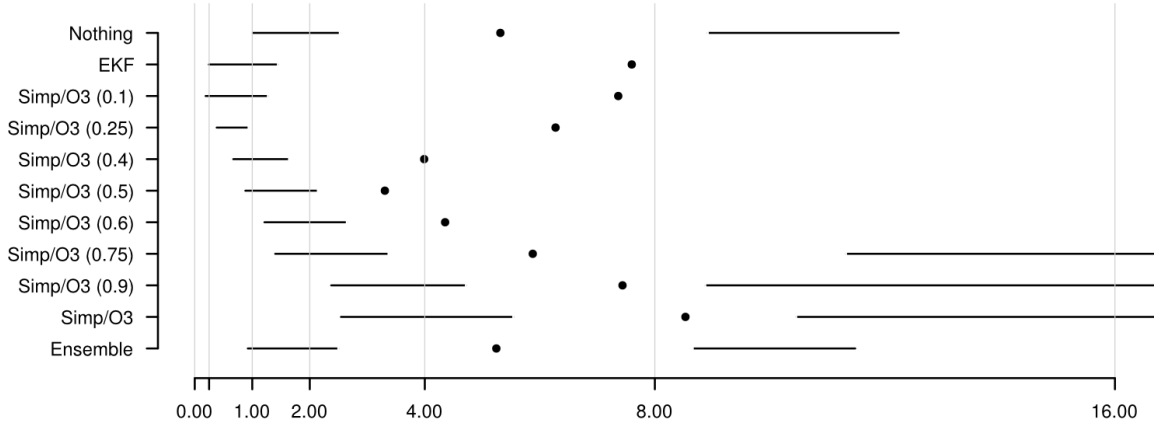

Figure 6.1.12: Quantile Error Plot for $\sigma = 8$ as alternative α varies for O3/Simp set

Sigma Point Set	Largest Point Distance	% of Distribution closer to 0
UKF Simp	1.73	91%
UKF O3	1.00	68%
UKF O5	1.73	91%
UKF O7	2.33	98%
UKF O9	2.86	99.6%

Table 6.1: Sigma Point Sets' Sampling Distance

6.1.3 Wrapping Example with Scaling

We can now go back and reexamine the Wrapping Example of the first section, where we can mimic and even out perform the EKF by decreasing the distance between the σ-points by applying our scaling factor α. Because we are trying to avoid interaction with the discontinuity when it is unlikely I suggest we examine the sampling distances of the different sets. Shown in Table 6.1 is each σ-point method's largest sample point's distance from the mean and the equivalent confidence interval of the input distribution, x, they encompass. In order to improve performance in the majority of cases we could scale these points back so they will not sample farther away than 50%, or 0.67 in standard deviations, so they do not interact, and therefore model, any parts of the function which are outside this region, the wrapping discontinuity in this example. We can see the results of applying this scaling in Figures 6.1.13 and 6.1.14. The improvement of performance over the EKF can be explained by an inbuilt data editing of this new method. When the measurement and expected value, agree the σ-point methods do not interact with the discontinuity and make EKF like corrections. When the measurement and expected value disagree, however, the σ-points begin to interact with the discontinuity and the filter makes less confident corrections as is appropriate because interaction with the wrapping discontinuity is more likely. Notice the improvement in the 50% and 75% error plots for the σ-point methods as compared to their previous, unscaled versions, which has come at the cost of 90% performance.

6.2 Smoothing Equations

The earlier re-interpreting of the σ-point estimates from Equation 6.1.3 also allow us to use a completely different random variable to perform a σ-point method estimate than the one that we are actually propagating in the UKF. This is similar to when we linearize a function about a different point than the mean of the random variable we are propagating, which allowed us to create an iterate, using the smoothed solution, on the EKF. Just as we have done with the EKF we can now use smoothed estimates

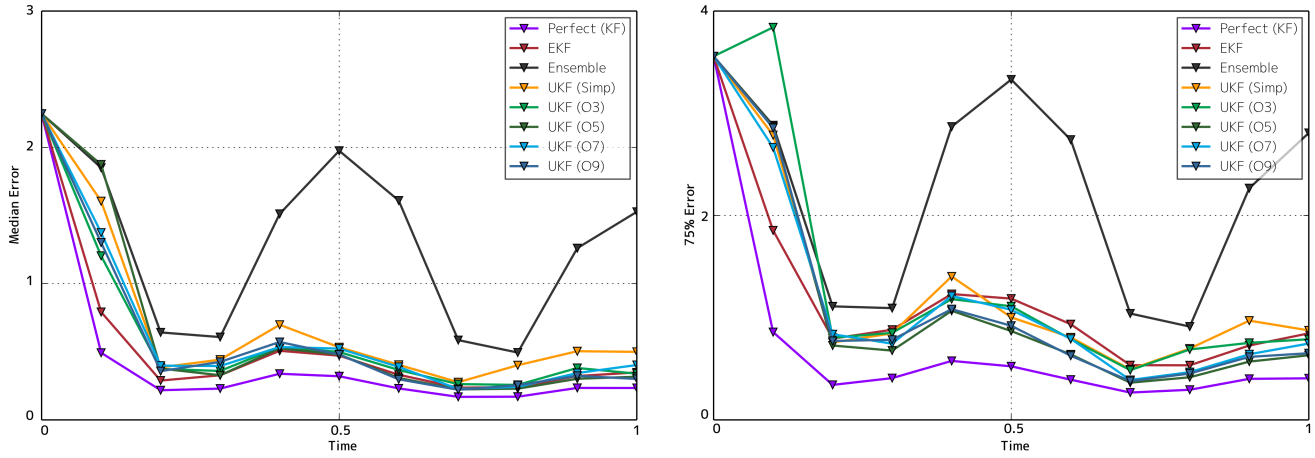

Figure 6.1.13: 50th and 75th Quantile Plots of State Errors for Wrapping Example with Scaling

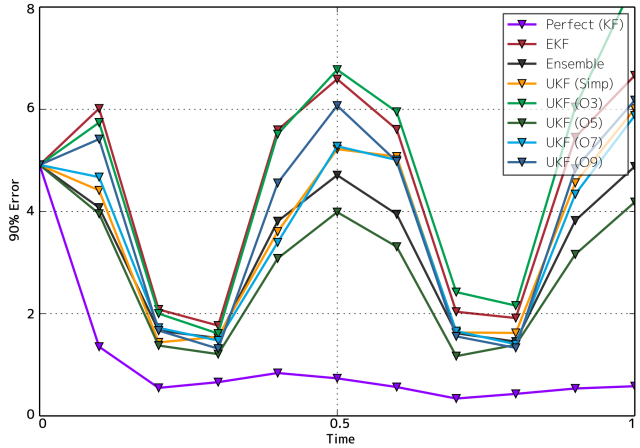

Figure 6.1.14: 90th Quantile Plot of State Error for Wrapping Example with Scaling

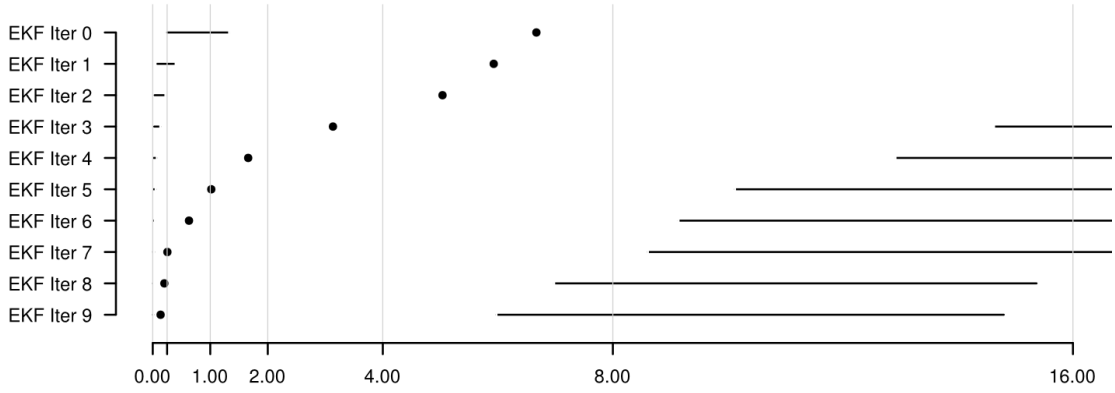

Figure 6.2.1: Quantile Error Plot for $\sigma = 8$ of Iterated EKF

of past random variables to re-"σ-point" past transforms. We can demonstrate this idea on the previous nonlinear observation example, $h(x, v) = \sum_{k=1}^{4} \left(-\frac{x}{k}\right)^k + v$. This function is smooth and with little uncertainty in v we should expect very little error in the estimate of x (if $v = 0$ there would be no uncertainty in x given the measurement as compared to the wrapping example where, due to the wrapping, there would still be ambiguity). The problem is difficult because the filters we are working with, at the core, make a linear assumption of some kind to solve it. When making linear assumptions in nonlinear problems we expect errors but by introducing an iterate, using the post correction estimates of the random variable x to re-estimate the transform, we can expect these errors to reduce and the eventually obtain convergence of estimates. This is straight forward for the EKF, where we form a new estimate of the transform, $H_{\{k\}}, y_{\{k\}}$, based on linearizing about the new estimate of random variable $x_{\{k\}} = \bar{x} + K_{\{k-1\}}\left(y_{\{k-1\}} - H_{\{k-1\}}\bar{x}\right)$. This process forms the Iterative Smoothed previously discussed and the improvement as a function of iteration is shown in Figure 6.2.1.

To form the iterate for the UKF filters we introduce the following form, which is simplified to take into account the linear nature of the introduced noise term, the AGWN v.

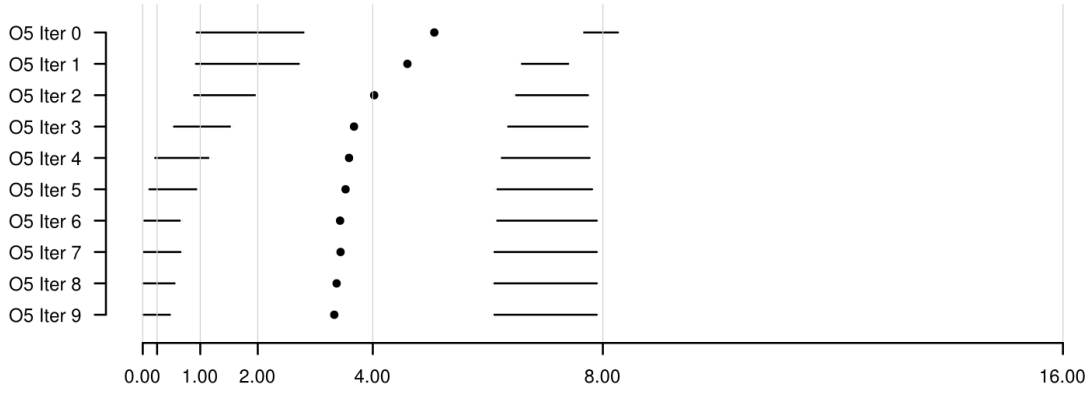

Figure 6.2.2: Quantile Error Plot for $\sigma = 8$ of Iterated UKF

$$
\begin{aligned}
x_{\{0\}} &= \bar{x} = 0 \\
P_{\{0\}} &= P = \sigma^2 \\
\mathfrak{S}\left(h(x,0), N\left(x_{\{i\}}, P_{\{i\}}\right)\right) &\rightarrow \left(\acute{y}_{\{i\}}, \acute{P}_{yy\{i\}}, \acute{P}_{xy\{i\}}\right) \\
\tilde{H}_{k\{i\}} &= \left(P_{\{i\}}^{-1}\acute{P}_{x,y}\right)^{\mathrm{T}} \\
P_{nn\{i\}} &= \acute{P}_{yy\{i\}} - \left(\tilde{H}_{\{i\}} P_{\{i\}} \tilde{H}_{\{i\}}^{\mathrm{T}}\right) \\
K_{\{i\}} &= P\tilde{H}_{\{i\}}^{\mathrm{T}}\left(\tilde{H}_{\{i\}} P \tilde{H}_{\{i\}}^{\mathrm{T}} + R + P_{vv\{i\}}\right)^{-1} \\
x_{\{i+1\}} &= x + K\left(y - \left(\acute{y}_{\{i\}} + \tilde{H}_{\{i\}}\left(x - x_{\{i\}}\right)\right)\right) \\
P_{\{i+1\}} &= \left(I - K_{\{i\}}\tilde{H}_{\{i\}}^{\mathrm{T}}\right) P \left(I - K_{\{i\}}\tilde{H}_{\{i\}}^{\mathrm{T}}\right)^{\mathrm{T}} + K_{\{i\}}\left(R + P_{vv\{i\}}\right) K_{\{i\}}^{\mathrm{T}}
\end{aligned}
$$

This allows the UKF to have the an iterative performance improvement similar to what was achieved with the EKF, as shown in Figure 6.2.2 where its 90% performance is much better the iterated EKF. Additionally by iterating the scaled UKF we have a filter which has superior performance in all quantiles and faster convergence than the EKF as shown in Figure 6.2.3.

6.2.1 The Iterative UKF Smoother

We defined the σ-point propagation of random variables, $N(\bar{x}, P), N(\bar{w}, Q)$ through a function $y = f(x, w)$ as $\mathfrak{S}(f, N(\bar{x}, P), N(\bar{w}, Q)) \rightarrow \left(\bar{y}, P_{y,y}, P_{x,y}, P_{w,y}\right)$ according to Equation 5.1.2[2]. We here define the scaled σ-point function estimator $\mathfrak{P}(\alpha, f, N(x, P), N(w, Q)) \rightarrow (u, F, L, \mathscr{U})$ in Equation

[2]The necessary abstraction here is that although f is a function of two variables x, w it can be thought of as a function of a single random variable $\acute{x} = \left[\begin{smallmatrix} x \\ w \end{smallmatrix}\right] \sim N\left(\left[\begin{smallmatrix} \bar{x} \\ \bar{w} \end{smallmatrix}\right], \left[\begin{smallmatrix} P & 0 \\ 0 & Q \end{smallmatrix}\right]\right)$, with $P_{x,y}, P_{w,y}$ being submatricies of $P_{\acute{x},y}$.

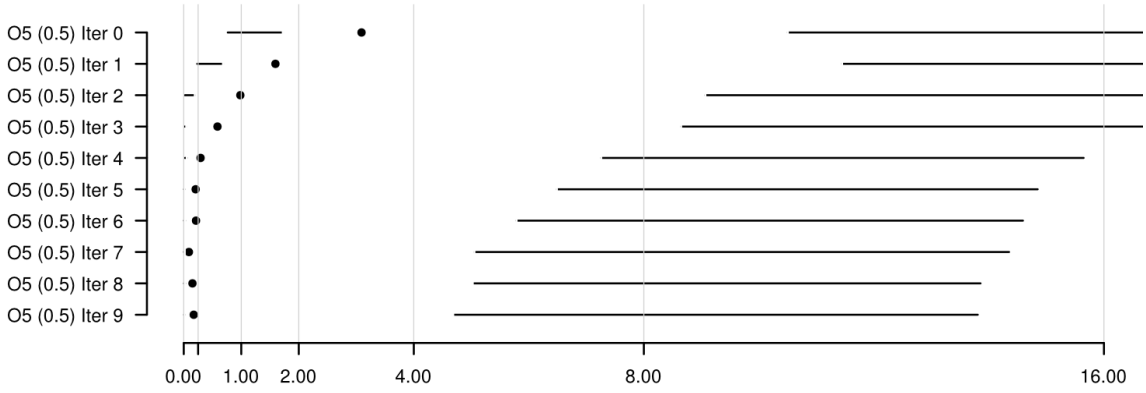

Figure 6.2.3: Quantile Error Plot for $\sigma = 8$ of Iterated UKF with Scaling

$$\begin{aligned}
\mathfrak{S}\left(f, N\left(x, \alpha^2 P\right), N\left(w, \alpha^2 Q\right)\right) &\rightarrow \left(y, P_{y,y}, P_{x,y}, P_{w,y}\right) \\
F &= \left(\left(\alpha^2 P\right)^{-1} P_{x,y}\right)^{\mathrm{T}} \\
L &= \left(\left(\alpha^2 Q\right)^{-1} P_{w,y}\right)^{\mathrm{T}} \\
u &= y - Fx - Lw \\
\mathscr{N} &= \frac{1}{\alpha^2} P_{y,y} - \left(FPF^{\mathrm{T}} + LQL^{\mathrm{T}}\right) \\
&\text{or} \\
\mathscr{N} &= P_{y,y} - \alpha^2 \left(FPF^{\mathrm{T}} + LQL^{\mathrm{T}}\right) \\
\mathfrak{P}\left(f, N\left(x, P\right), N\left(w, Q\right)\right) &\rightarrow \left(u, F, L, \mathscr{N}\right)
\end{aligned}$$

Although we have formulated this with the notation from the propagation it works equally well for observation $\mathfrak{P}\left(h, N\left(x, P\right), N\left(v, R\right)\right) \rightarrow \left(z, H, J, \mathscr{V}\right)$. We can now simply substitute the σ-point function estimator in for the linearization process in the EKF to form the UKF. The propagation step becomes,

$$\begin{aligned}
\mathfrak{P}\left(f_k, N\left(x_{k|k}, P_{k|k}\right), N\left(0, Q_k\right)\right) &\rightarrow \left(u_k, F_k, L_k, \mathscr{W}_k\right) \\
x_{k+1|k} &= F_k x_{k|k} + L_k w_k + u_k \\
P_{k+1|k} &= F_k P_{k|k} F_k^{\mathrm{T}} + L_k Q_k L_k^{\mathrm{T}} + \mathscr{W}_k
\end{aligned}$$

and update step,

$$\begin{aligned}
\mathfrak{P}\left(h_k, N\left(x_{k|k-1}, P_{k|k-1}\right), N\left(0, R_k\right)\right) \rightarrow & \left(n_k, H_k, J_k, \mathscr{V}_{k1}\right) \\
\mathscr{Z}_k &= \left(H_k P_{k|k-1} H_k^{\mathrm{T}} + J_k R_k J_k^{\mathrm{T}} + \mathscr{V}_k\right) \\
K_k &= P_{k|k-1} H_k^{\mathrm{T}} \mathscr{Z}_k^{-1} \\
x_{k|k} &= x_{k|k} + K_k\left(y_k - \left(n_k + H_k x_{k|k-1}\right)\right) \\
P_{k|k} &= \left(I - K_k H_k\right) P_{k|k-1}.
\end{aligned}$$

To form an iterative smoother based on this process we only need to substitute in our smoothed versions of variables. This makes the propagation step,

$$\begin{aligned}
\mathfrak{P}\left(f_k, N\left(x_{k|\eta\{i\}}, P_{k|\eta\{i\}}\right), N\left(w_{k|\eta\{i\}}, Q_{k|\eta\{i\}}\right)\right) \rightarrow & \left(u_{k\{i\}}, F_{k\{i\}}, L_{k\{i\}}, \mathscr{W}_{k\{i\}}\right) \\
x_{k+1|k\{i\}} &= F_{k\{i\}} x_{k|k\{i\}} + L_{k\{i\}} w_k + u_{k\{i\}} \\
P_{k+1|k\{i\}} &= F_{k\{i\}} P_{k|k\{i\}} F_{k\{i\}}^{\mathrm{T}} + L_{k\{i\}} Q_k L_{k\{i\}}^{\mathrm{T}} + \mathscr{W}_{k\{i\}}
\end{aligned}$$

and the update step,

$$\begin{aligned}
\mathfrak{P}\left(h_k, N\left(x_{k|\eta\{i\}}, P_{k|\eta\{i\}}\right), N\left(v_{k|\eta\{i\}}, R_{k|\eta\{i\}}\right)\right) \rightarrow & \left(n_{k\{i\}}, H_{k\{i\}}, J_{k\{i\}}, \mathscr{V}_{k\{i\}}\right) \\
\mathscr{Z}_{k\{i\}} &= \left(H_{k\{i\}} P_{k|k-1\{i\}} H_{k\{i\}}^{\mathrm{T}} + J_{k\{i\}} R_{k\{i\}} J_{k\{i\}}^{\mathrm{T}} + \mathscr{V}_{k\{i\}}\right) \\
K_{k\{i\}} &= P_{k|k-1\{i\}} H_k^{\mathrm{T}} \mathscr{Z}_k^{-1} \\
x_{k|k\{i\}} &= x_{k|k\{i\}} + K_{k\{i\}}\left(y_k - \left(n_{k\{i\}} + H_{k\{i\}} x_{k|k\{i\}}\right)\right) \\
P_{k|k\{i\}} &= \left(I - K_{k\{i\}} H_{k\{i\}}\right) P_{k|k-1}.
\end{aligned}$$

We have already seen equations which provide reverse iterates for the various state and noise vectors. To find the related covariances we simply infer the covariance from the linear transform nature of Gaussian random variables. In the Iterative UKF Smoother we can use the original, un-smoothed, variables for any of the components to avoid the complexity of solving for their smoothed equivalents. This simplification may not drastically change the performance of the filter, as discussed in Chapter 4 where smoothed versions of the propagation noise did little to improve the EKF's IS results, allowing us to use the original $N\left(0, Q_k\right)$ instead of $N\left(w_{k|\eta\{i\}}, Q_{k|\eta\{i\}}\right)$. In cases where the noise is additive, and therefore there is no nonlinear factor to be re-estimated, as is the case in the observation noise, there is no benefit to solving for the smoothed versions. The state reverse equation with covariance becomes,

$$\begin{aligned}
S_{k-1} &= P_{k-1|k-1\{i\}} F_{k\{i\}}^{\mathrm{T}} P_{k|k-1\{i\}}^{-1} \\
x_{k-1|\eta\{i\}} &= x_{k-1|k-1\{i\}} + S_{k-1}\left(x_{k|\eta\{i\}} - x_{k|k-1\{i\}}\right) \\
P_{k-1|\eta\{i\}} &= P_{k-1|k-1\{i\}} + S_{k-1}\left(P_{k|\eta\{i\}} - P_{k|k-1\{i\}}\right) S_{k-1}^{\mathrm{T}}
\end{aligned}$$

the propagation noise,

$$
\begin{aligned}
T_k &= Q_k L_{k\{i\}}^{\mathrm{T}} P_{k|k-1\{i\}}^{-1} \\
w_{k|\eta\{i\}} &= Q_k L_{k\{i\}}^{\mathrm{T}} P_{k|k-1\{i\}}^{-1} \left(x_{k|\eta\{i\}} - x_{k|k-1\{i\}} \right) \\
Q_{k|\eta\{i\}} &= Q_k + T_k \left(P_{k|\eta\{i\}} - P_{k|k-1\{i\}} \right) T_k^{\mathrm{T}}
\end{aligned}
$$

and observation noise,

$$
\begin{aligned}
D_k &= R_k J_{k\{i\}}^{\mathrm{T}} \left(J_{k\{i\}} R_k J_{k\{i\}}^{\mathrm{T}} + \mathscr{V}_{k\{i\}} \right)^{-1} \\
v_{k|\eta\{i\}} &= D_k \left(y_k - \left(n_{k\{i\}} + H_{k\{i\}} x_{k|\eta\{i\}} \right) \right) \\
R_{k|\eta\{i\}} &= \left(I - D_k J_{k\{i\}} \right) R_k
\end{aligned}
$$

Chapter 7

Sigma Point/UKF Application to the Blind Tricyclist Problem

> *Now Jurgen recognized the feeling perfectly. He had often had it in his sleep, in dreams wherein he would bend his legs at the knees so that his feet came up behind him, and he would pass through the air without any effort. Then it seemed ridiculously simple, and he would wonder why he never thought of it before. And then he would reflect: "This is an excellent way of getting around. I will come to breakfast this way in the morning, and show Lisa how simple it is. How it will astonish her, to be sure, and how clever she will think me!" And then Jurgen would wake up, and find that somehow he had forgotten the trick of it.*
>
> - James Branch Cabell, *Jurgen*

With the theory developed in the previous two chapters we now have a suitable alternative to the the EKF for attempting to solve the Blind Tricyclist Problem with improved performance. If we simply apply a UKF with a standard set of σ-points, without taking into account the additional developments introduced in Chapter 6, we do not see the dramatic improvement expected as shown in Figure 7.0.1. As evident from this figure the simple application of the UKF does not yield performance even as good as the EKF in many scenarios. This is proven out in the analysis of error statistics, shown in Figure 7.0.2. As we can see in these figures, the performance is inferior to the EKF of Chapter 4, so we will need to start examining the problem using the advanced techniques we have developed.

7.1 Scaling in the Blind Tricyclist Problem

The observation function in the Blind Tricyclist Problem, because of the inclusion of the arctangent function, has a wrapping property similar to the one we have already examined. This along with other

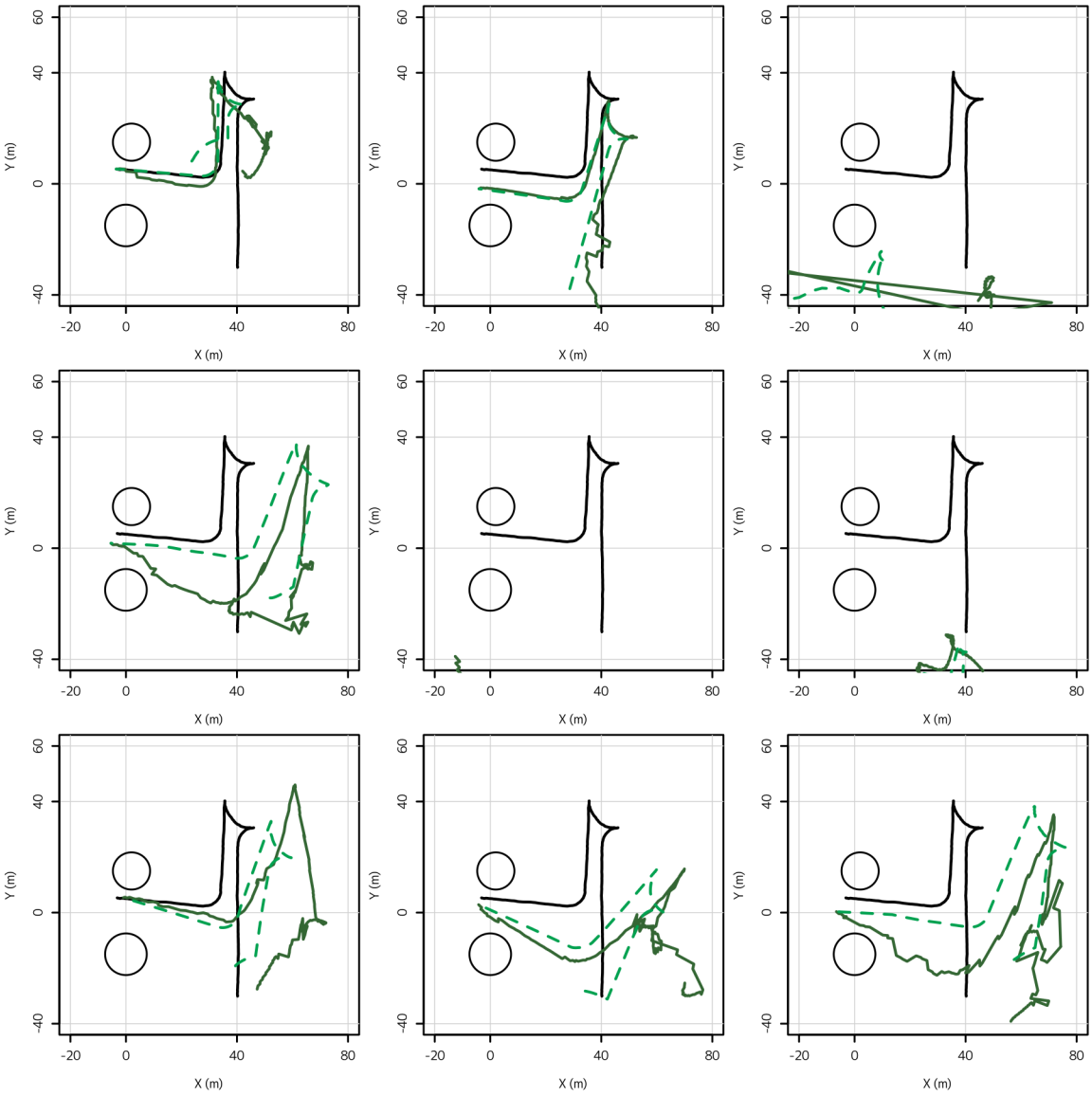

Figure 7.0.1: UKF with O5 Example Tracks (Truth in black, UKF in dark green, Smoothed UKF green dashed)

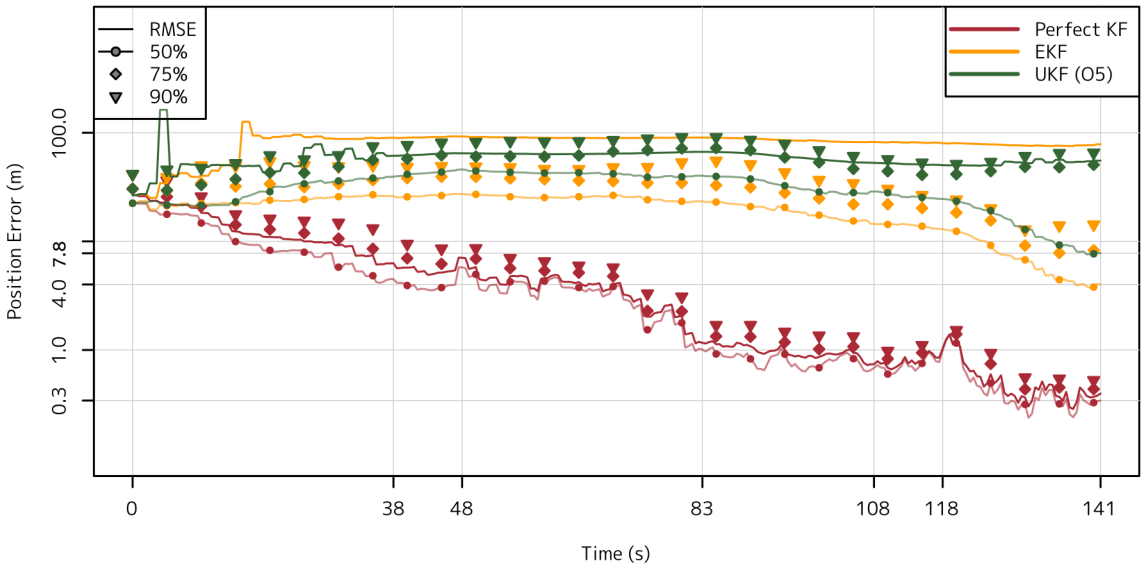

Figure 7.0.2: UKF O5 Error Compare

factors in the problem will require us to apply the scaling method discussed in Chapter 6 to improve performance. The appropriate scaling parameter, however, is somewhat problem specific as we have seen so we will need to tune it. So that we do not over specialize our filter to this specific dataset, we will determine the scaling parameter by examining its impact on similar, simpler, scenarios. We will examine simple scenarios with 3 different starting points with 2 different movement sets. The six scenarios to be considered are shown in Figure 7.1.1. I have constructed the scenarios to represent movements and positions we might expect from the tricyclist to allow us to find a tuning which takes into account the specifics of the problem. Each of these scenarios includes only 10 time increments and the measurement sequence has been sped up to one measurement every other step, $\begin{bmatrix} - & 1 & - & 2 \end{bmatrix}$, from one every 3 in the original scenario.

On each of these scenarios we can compare the end state position error of the different σ-point sets with a course sampling of scaling parameters. The results are shown in Figures 7.1.2, 7.1.3, 7.1.4, 7.1.5, 7.1.6, 7.1.7. Each of these scenarios behaves differently under each of the σ-point sets and scaling parameters and when comparing them to chose a σ-point set we are fortunate that there is no set which works markedly better in some but worse in others. When comparing the sets I would suggest avoiding those that require smaller scaling factors, their EKF behavior is something we are trying to avoid, which eliminates O5f[1]. Additionally I suggest we avoid sets which appear sensitive to the scaling parameter as we

[1]Doubly so because the arctangent wrapping problem is especially bad with this set

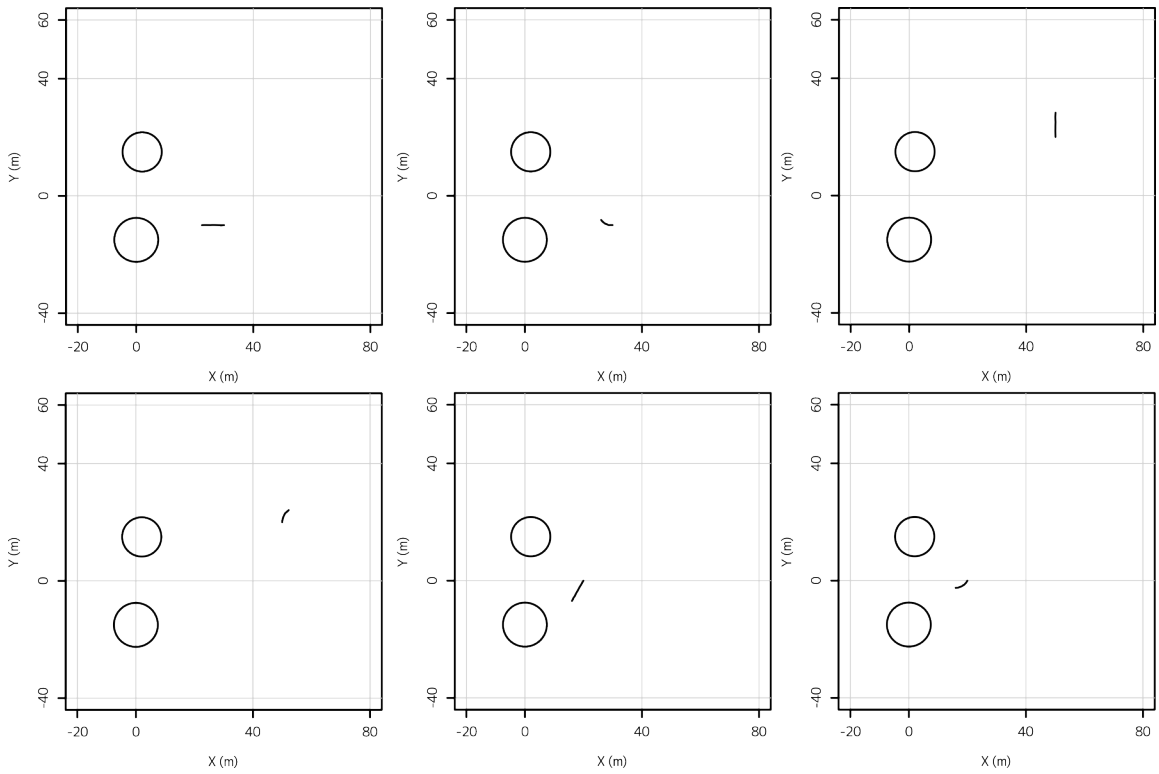

Figure 7.1.1: Simple Scenarios for Parameter Tuning

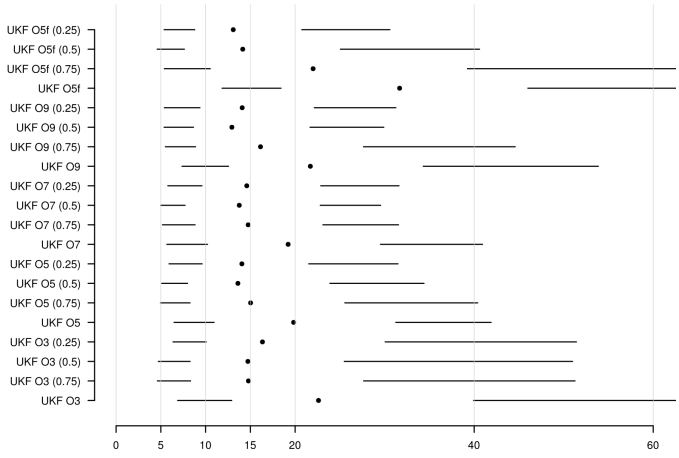

Figure 7.1.2: End State Error for Scenario 1 (Point 1 Straight Movement)

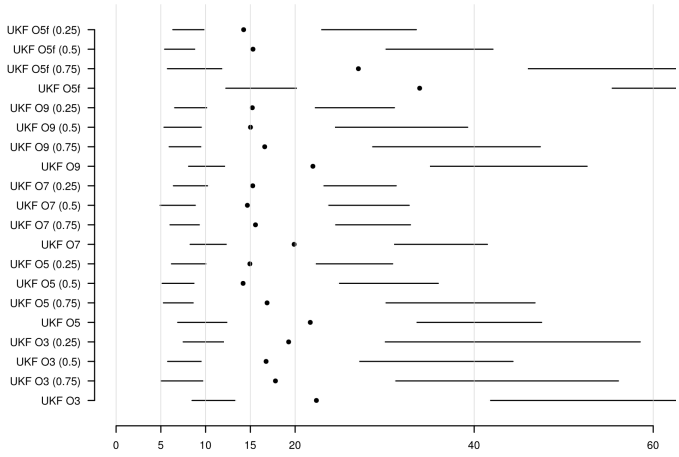

Figure 7.1.3: End State Error for Scenario 2 (Point 1 Curved Movement)

will not be able to find a robust method necessarily applicable to more general conditions that might arise in the full problem. With this in mind, I suggest we work with the O7 set as it performs the best in most scenarios and close to best in the rest.

Now that we have a candidate set we can take a closer look at the scaling parameter. We have increased the resolution for Scenario 1 and 5 and shown the results in Figures 7.1.8 and 7.1.9 respectively. These scenarios both have different responses to the scaling parameter, α, the first improving with larger α and the second with smaller. They, visually, have a cross over point in performance at $\alpha = 0.4$ so this, O7 with a scaling factor of 0.4, is the set I would recommend.

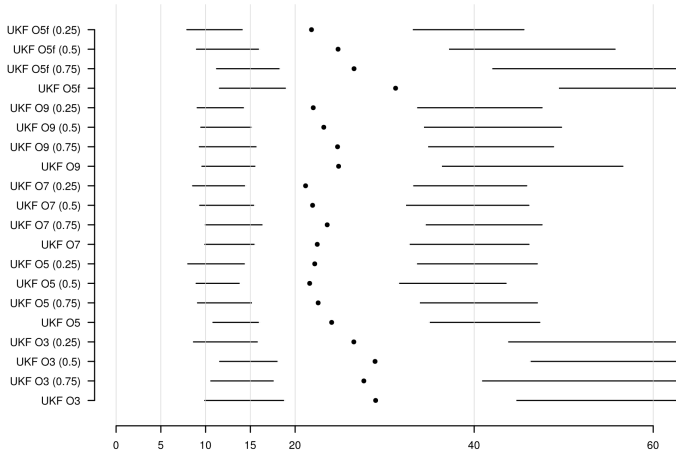

Figure 7.1.4: End State Error for Scenario 3 (Point 2 Straight Movement)

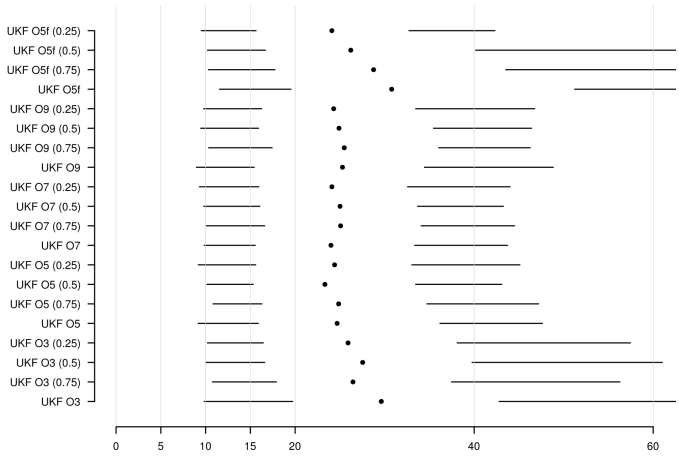

Figure 7.1.5: End State Error for Scenario 4 (Point 2 Curved Movement)

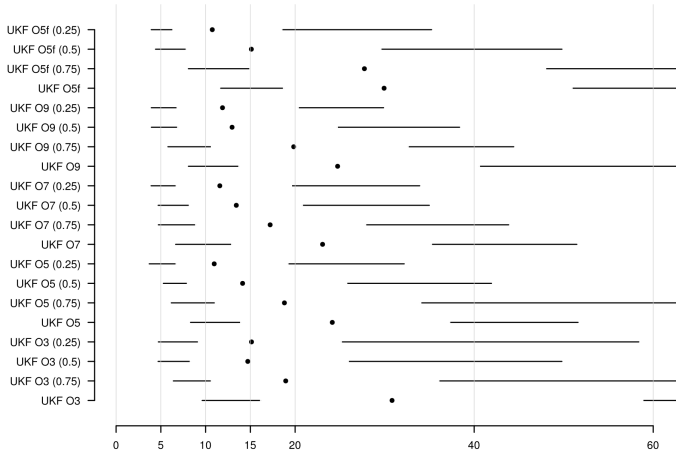

Figure 7.1.6: End State Error for Scenario 5 (Point 3 Straight Movement)

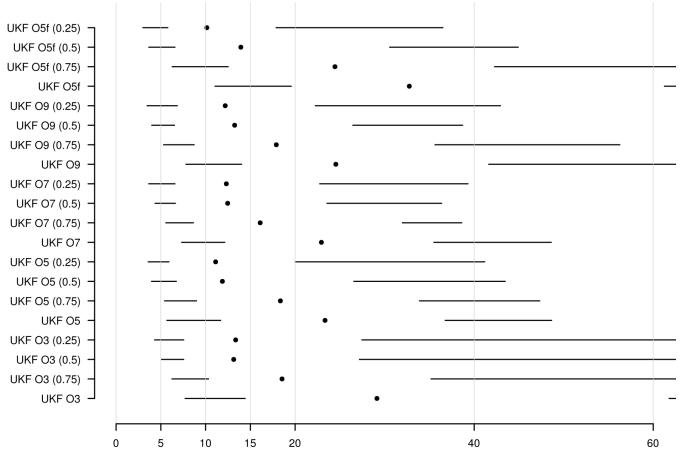

Figure 7.1.7: End State Error for Scenario 6 (Point 3 Curved Movement)

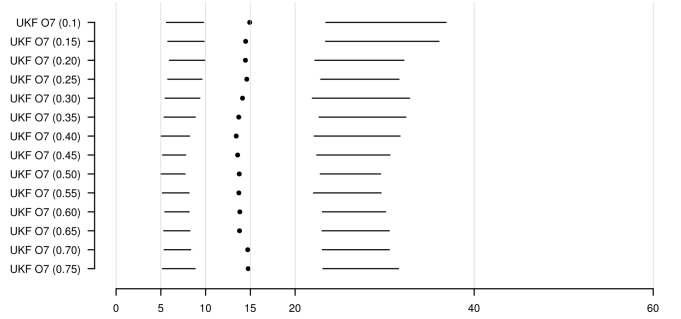

Figure 7.1.8: End State Error for Scenario 1 (Point 1 Straight Movement)

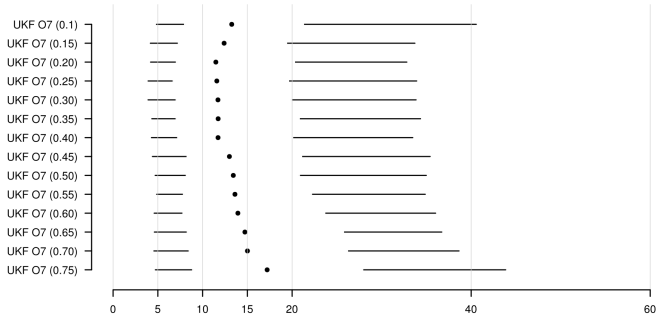

Figure 7.1.9: End State Error for Scenario 5 (Point 3 Straight Movement)

7.2 New Results in the Blind Tricyclist Problem

Now that we have picked a σ-point set and scaling factor we can start to analyze its performance on the original Blind Tricyclist scenario. In Figure 7.2.1 we have shown a couple of example runs for this filter. The error statistics for the filter are shown in Figure 7.2.2 where it can be seen to be performing better than any of the previous filters.

We can apply the smoothing operation described in the previous chapter to this problem in an attempt to improve performance. As described in Chapter 4 there is not much point to using new information on the propagation noise, w, because there is not much information to be gained. Additionally because there is no nonlinear effect from the observation noise, v, there is nothing to be gained by using smoothed estimates for this variable. Using a UKF smoother with 20 passes gives us a dramatic improvement over what we have seen before, as is shown in Figure 7.2.3.

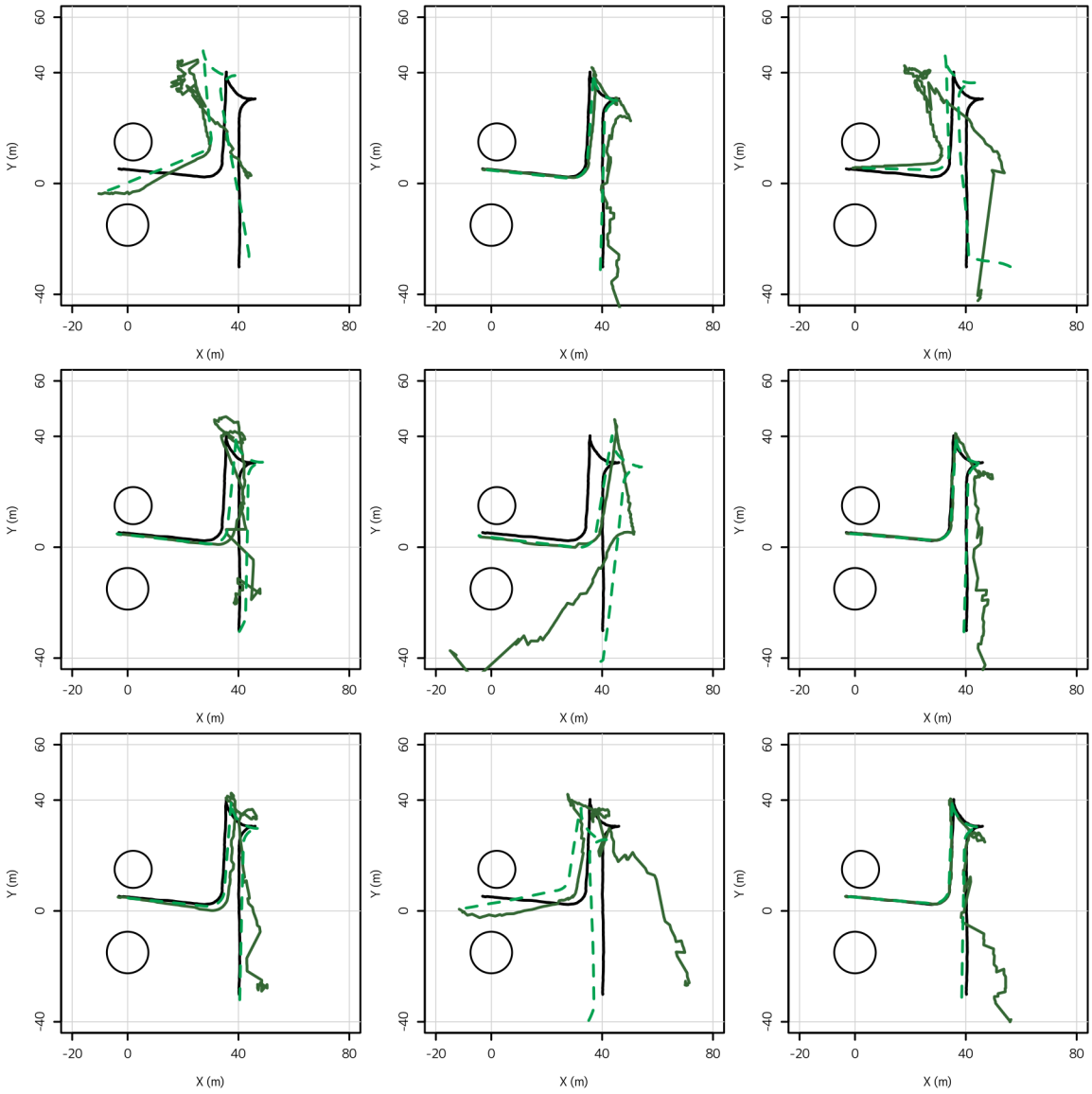

Figure 7.2.1: UKF with O7 (0.4 scaling) Example Tracks

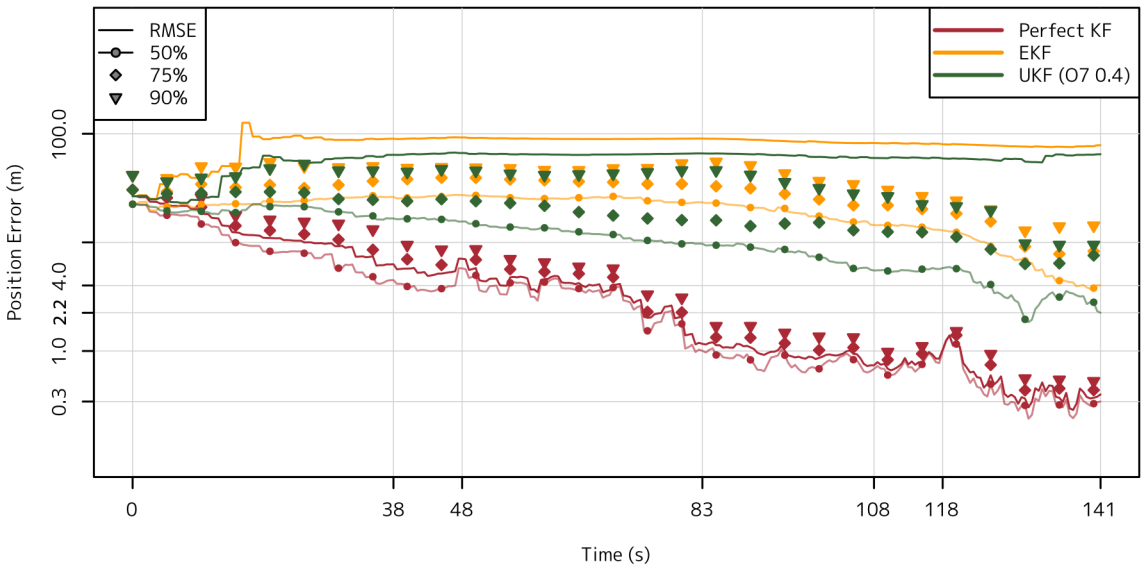

Figure 7.2.2: UKF O7 (0.4 Scaling) Error Compare

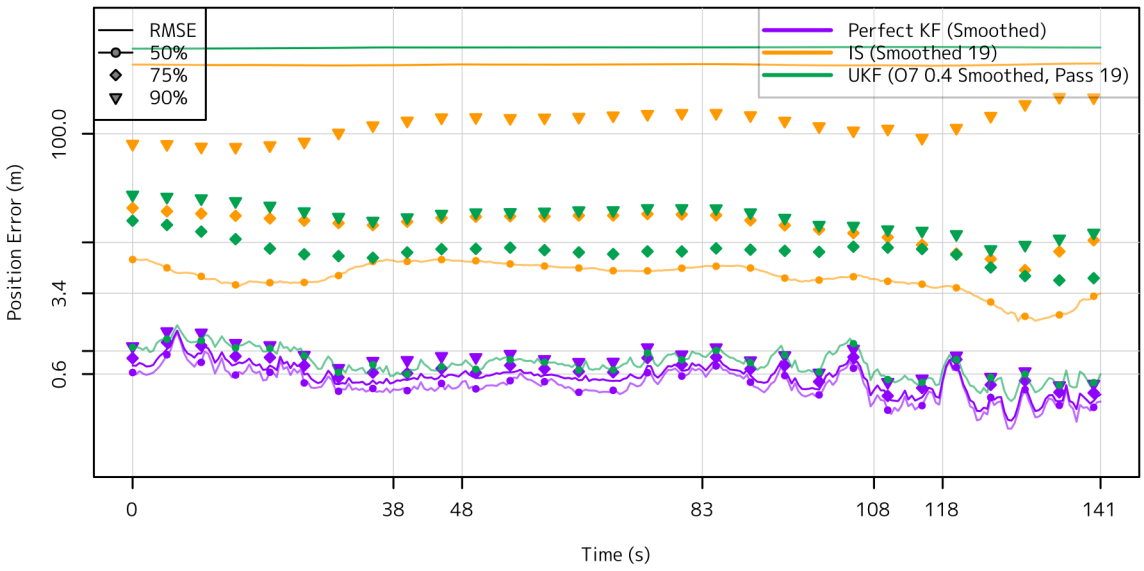

Figure 7.2.3: UKF O7 (0.4 Scaling) with Smoothing Error Compare

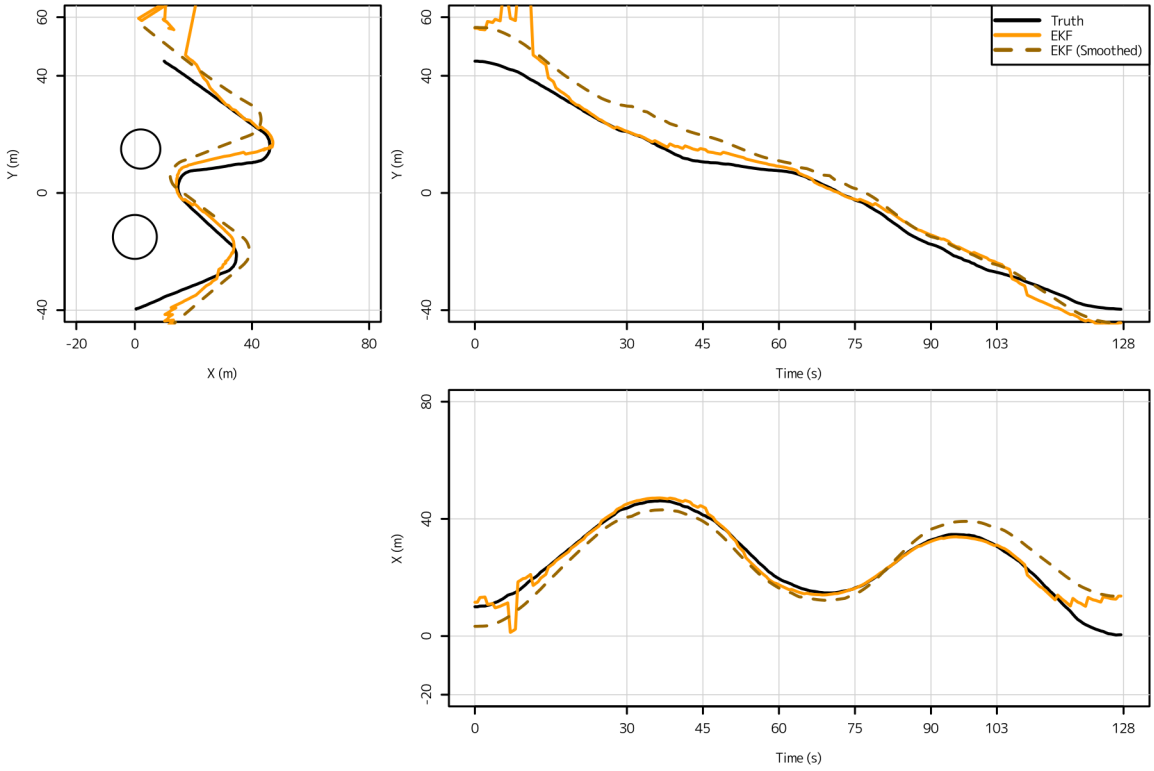

Figure 7.3.1: Alternate Scenario EKF Example

7.3 Alternate Scenario

As a final examination of this problem we can create an alternate movement set. Shown in Figure 7.3.1 is the movement and EKF performance on this new data set. This scenario is similar to the first in length but its movement should be different enough to strengthen the generality of the results produced.

Again in this example the EKF, even with an iterative smoother, fails to perform at the levels we might expect, its statistical errors are shown in Figure 7.3.2. By applying the tuned UKF with smoothing we can again achieve results similar to the perfectly linearized case, shown in Figure 7.3.2 in the majority of cases.

7.4 Conclusions on the Blind Tricyclist Problem

We have seen, through the examination of 2 different possible movements each under many different observation scenarios, that between the EKF, which provides a common baseline for performance in nonlinear problems like this, and a perfectly linearized KF, which gives us an approximate best case performance, there is a lot of room for improvement. By utilizing the combination of smoothers, which

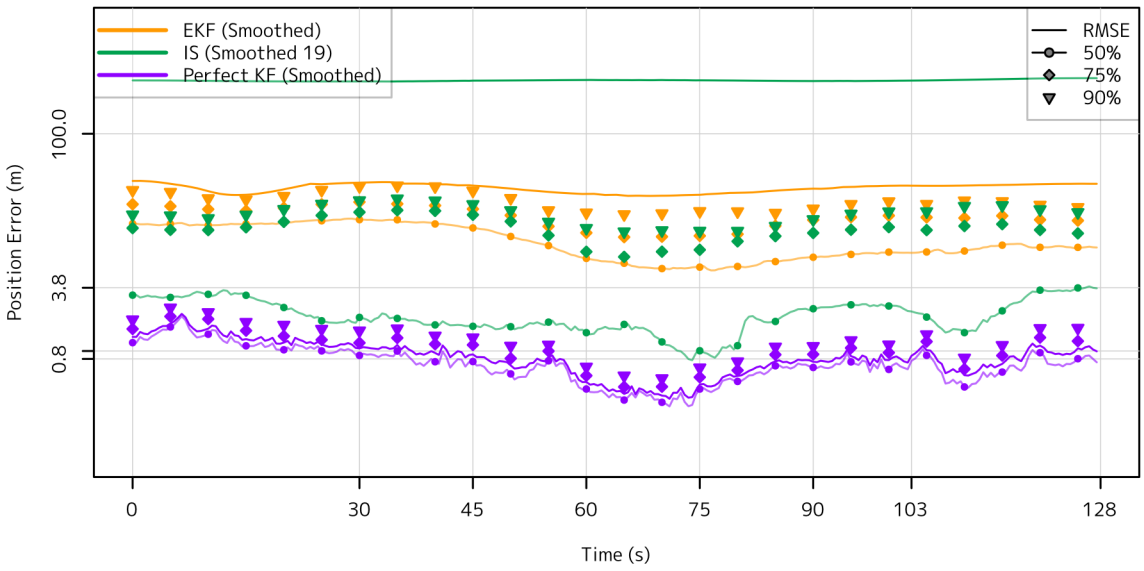

Figure 7.3.2: EKF Error Statistics for Alternate Scenario

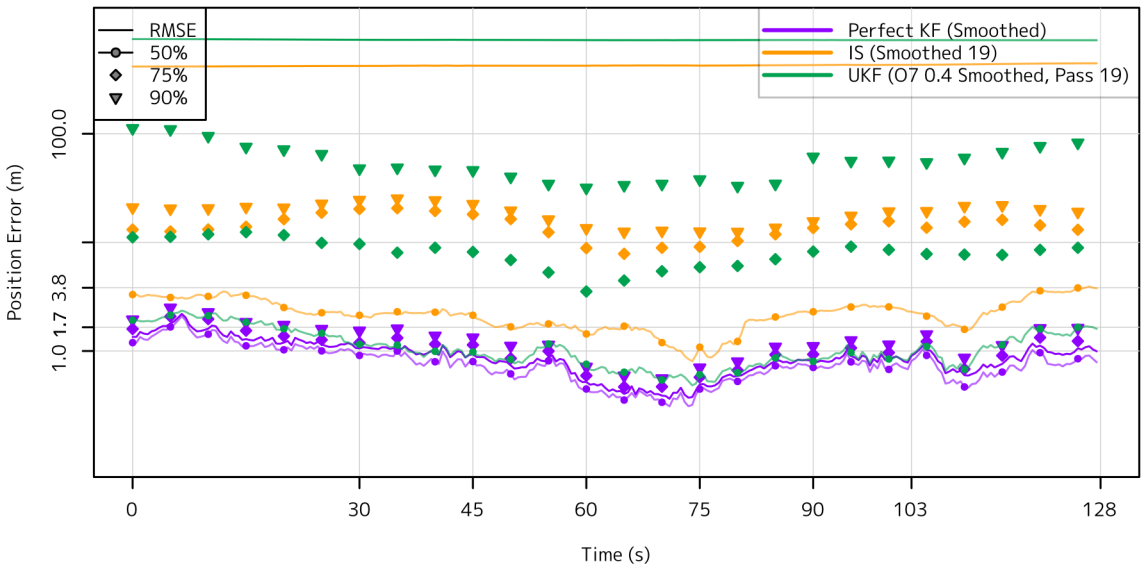

Figure 7.3.3: UKF Error Statistics for Alternate Scenario

re-approximate the problem using complete solutions, and the UKF, which approximates the problem using statistically strategic sampling, we have found a method that improves performance in the majority of cases. This method does require an element of problem specific tuning but we have shown that this is possible to find by considering similar, simpler, cases.

Chapter 8

Conclusions

> *Steel screams when it's forged, it gasps when it's quenched. It creaks when it goes under load. I think even steel is scared, son.*
>
> - Walter M. Miller, Jr., *A Canticle For Leibowitz*

Just as the introduction chapter provides a synopsis of what each chapter would be tackling this chapter will be exploring some of broader results of each chapter.

Chapter 2 The State/Dual Kalman Filter Represent and Chapter 3 Smoothing Solutions Both of these chapters are about the Symplectic Kalman Filter so their results will be discussed together. The use of the Symplectic form allows the construction of a smoother without the need to preserve and use the covariance matrices from each time step, which is a new result of this work. Although its usefulness is compromised by the numerical issues discussed, the methods contributed in this work can reduce the computational complexity by potentially trimming the the number of matrix inversions necessary.

Chapter 4 Smoothed Solution Convergence in the Blind Tricyclist Problem This chapter reviews the Blind Tricyclist Problem, which is presented as an example of a challenging nonlinear tracking problem. When approaching a problem like this in practice there is often a performance goal, for example, to reach below 1 m error by the end of the track. All estimators have a probability of failure but as we make the performance goal tighter or introduce assumptions necessary for constructing simplified or computationally frugal implementations the chance may increase dramatically. In this study, rather than introduce a fixed performance target, I have described the error in quantiles. From these quantiles we can estimate how certain filters would perform with different error targets. Beyond the limitations on performance dictated by the noise itself the other main contributor to errors for the filters in this work is the linearization

process itself. This process is suboptimal even when given correct linearization points but even worse when given incorrect ones, resulting in increased error and, in turn, increased probability of failure. If the performance requirement is easy enough, such as below 5 m of error 50% of the time, the simple EKF may be enough to obtain the desired result. But assuming that we want to achieve a result closer to the approximate performance bound of the perfectly linearized Kalman Filter, we will need to introduce various iterative smoothing techniques. One of the take aways from this chapter is that when considering what variables (state, process noise, observation noise) to include in the linearization iterate we need to consider both their potential impact and our ability to estimate them. In this example, and in most, we have very little ability to estimate the process noise w, which coupled with its negligible impact on the problem (as seen by comparing the performance of filters linearizing about its truth versus its expected value), gives it very little value as a relinearization parameter[1]. The performance of the IS therefore is primarily limited by false minima, which is evident in the performance of an IS initialized with the truth value as its linearization points.

One of the things not discussed in this chapter is the notion of breaking the data into smaller intervals to be processed rather than considering the entire batch as a whole. In the original work [10] as each new data point is added the entire iterated smoothing process is performed for a smaller interval at the cost of ~1000 times the computation time of the single EKF pass. This gives us the greatest chance of avoiding being caught in a local minima, and is, I believe, the main contributor in the performance increase demonstrated in the original reference, as opposed to the inclusion of process noise in the relinearization. Within any given sub interval, all of the discussion of this work is relevant and considering results across a range of interval sizes would introduce an entire new set of parameters and add a lot of clutter into an already complicated chapter. For that reason, sub-interval smoothing was not explored in this work.

Chapter 5 Sigma Point/Unscented Methods Geometry There are a couple of things I feel need to be pointed out. First, all the sets, other than the simplex set, are constructed using the given strategy but are not necessarily the minimal set for the properties desired. Although I present a parameterization for the σ-point's space, it turns out to be incredibly difficult to actually search it. I had hoped, from all the progress in the development of a new descriptive geometry, I would have found a more robust way of generating point sets than the ones initially implemented. As an example of a partial result in this direction; to meet 3rd order constraints for n variables it appears that you need $2n$ σ-points. Examining their any 3rd order moment matrix A_1 as defined in 5.3.2 on page 68 reveals a block form, $A_1 = \left[\begin{smallmatrix} B & C \\ C^\intercal & D \end{smallmatrix}\right]$,

[1]The observation noise on the other hand is something we can estimate but in this example, because it is simple Gaussian noise introduced linearly, the non-linear extensions have no potential impact.

where the upper diagonal block is the, known to be zero, matrix of 3rd order constraints, $B \in \mathbb{R}^{(n+1)\times(n+1)}$, the lower diagonal block is set to zero $D = 0$ and C is left free. Diagonalizing A_1 matrices of this form generates sets of σ-points which have correct, 0, 3rd order moments and require one less point than the O3 set, found in 5.4.2 on page 74 which meets the same constraints.

Additionally the 1D sets may not behave quite as expected because all the sets are generated for noise statistics of the standard Gaussian, $N(0, I)$, and are transformed into $N(\mu, P)$ via $\acute{\chi} = \sqrt{P}\chi + \mu$, where $\sqrt{P}$ is the matrix square root of P. The 1D sets do not initially contain cross moments, but it should be clear that after mixing with $\sqrt{P}$ it is likely that there is some cross state moments and some error in the 1D state moments. As an example of this consider the case where $\mu = 0$ and $P = I$ except that we pick an orthogonal matrix $\mathcal{P}$ to be the $\sqrt{P}$. The points, after the affine transform, should still be valid σ-points for the random variable $N(0, I)$ and they have the same weights as the original set meaning its prototype form is related by some orthogonal transform $\left[\begin{smallmatrix} 1 & 0 \\ 0 & \mathcal{R} \end{smallmatrix}\right]$ causing its moments to be changed through the tensor equation discussed in the simplex set derivation, Equation 5.4.1 on page 73. This partially explains the lack of performance increase for the O5f set for problems in the Angle Tracking example. Thus, while we have made significant advances to the theory of σ-points sets, with regard to both of minimal size and higher order moment preserving properties, there is more work left to be done in future investigations.

Chapter 6 Sigma Point/Unscented Smoothing The most important take away from this chapter is that all of the linearized Kalman Filters, EKF, UKF, and EnKF, represent different points on the spectrum between correction strength (very accurate results sometimes) and robustness (less accurate results most of the time). On the two extremes there is the the EKF, which represents a very aggressive strategy, trying to extract the most information in a local sense, and the Ensemble KF, which is more passive, working on the entire statistical region. Neither of these are strictly better, they are simply different strategies, it's possible that a EKF approach is more appropriate for a given problem if an occasional good result is more valuable than consistently mediocre ones. The contribution of the new scaling method gives us a fine control between these two extremes, and would be applicable to the ensemble method.

The other major contribution is the Iterative UKF. This iterate is powerful because although the UKF starts by making corrections taking into account the necessary amount of nonlinearity for the original, pre-smoothing, random variables, both state and appropriate noise, as the estimate of these variables improves, through observations, the filter re-estimates the amount of nonlinearity noise using the post-smoothing covariances, resulting in less nonlinear noise.

Consider the following 1D example with no propagation but one observation. The state and observation noise are Gaussian random variables, $x \sim N(0, 0.49)$, $v \sim N(0, 0.0025)$, and the observation function

is given by $1/2\left(x^3 + 3/2x^2 - 1/4x - 1/8\right) + v$. First imagine a realization where the truth state and noise are given by $x = -1, v = 0.02$, giving us an observed value of $y = -0.92$. When the UKF first estimates the function h its σ-points extends into and includes the region $x \approx (-0.5, 2)$ where the observation function is highly nonlinear. Recognizing this the σ-point process initially estimates the nonlinear noise as $N(0, 0.38)$ with a covariance ellipse shown in Figure 8.0.1 in crimson centered on the crimson 'x'. The correction, because of this large nonlinear noise, is relatively weak moving the mean and shrinking the state's variance very little. After this correction however we can re-estimate the transform, h, using this update mean and covariance. This new estimate, shown in green, interacts with the nonlinear region less and in turn estimates less nonlinear noise allowing the filter to use a stronger correction. As this process continues the filter's estimate of the state becomes more and more confident until, at iteration 5, it estimates the nonlinear noise at $N\left(0, 8.4 \times 10^{-6}\right)$ allowing a practically linear observation giving us an excellent final state estimate of $x \sim N(-1, 0.00032)$. This all was done without any new observations and is similar to how an IS would iterate to the solution without the inclusion of the covariances. This process, the Iterative UKF, differs vastly from the IS when we consider a different problem realization, $x = 1, v = 0.02$ giving us an observed value of $y = -0.17$. The iterative process for this shown in Figure 8.0.2. This case, just as the first, initially estimates the transform, h, with a nonlinear noise of $N(0, 0.38)$. However, as it iterates it cannot avoid the strongly nonlinear region and so even by iteration 5 its estimate of the state is poor and includes a lot of variance, $x \sim N(-0.094, 0.31)$; the transform still being estimated as still having significant nonlinear noise component, $N(0, 0.19)$. This demonstrates the method's ability to iterate to very confident solutions when possible and to maintain a strong level of uncertainty when it is not. An Iterated EKF would have found one of the measurement's crossing points with the function and estimated its covariance with a linearization about that point, failing to maintain the covariance necessary to include the other possibilities.

Chapter 7 Sigma Point/UKF Application to the Blind Tricyclist Problem The results of this chapter speak for themselves for the most part, but I want to mention that I tried to construct the tuning as fairly as possible, trying to mimic what I might have known going into the problem, before the tricyclist actually pedaled the path. It seems likely that someone approaching the problem would have known about the approximate area the tricyclist would be in and been able to pick out a couple of possible points within that area. They also would have known the approximate speed of the tricyclist and its capability to turn. The tuning goal process is intended to get the filter to work as well in the general type of scenario it will be facing by considering the specifics of a given problem, the approximate region traversed by the tricyclist and approximate speed in this example, without over specializing it, by considering only straight moving

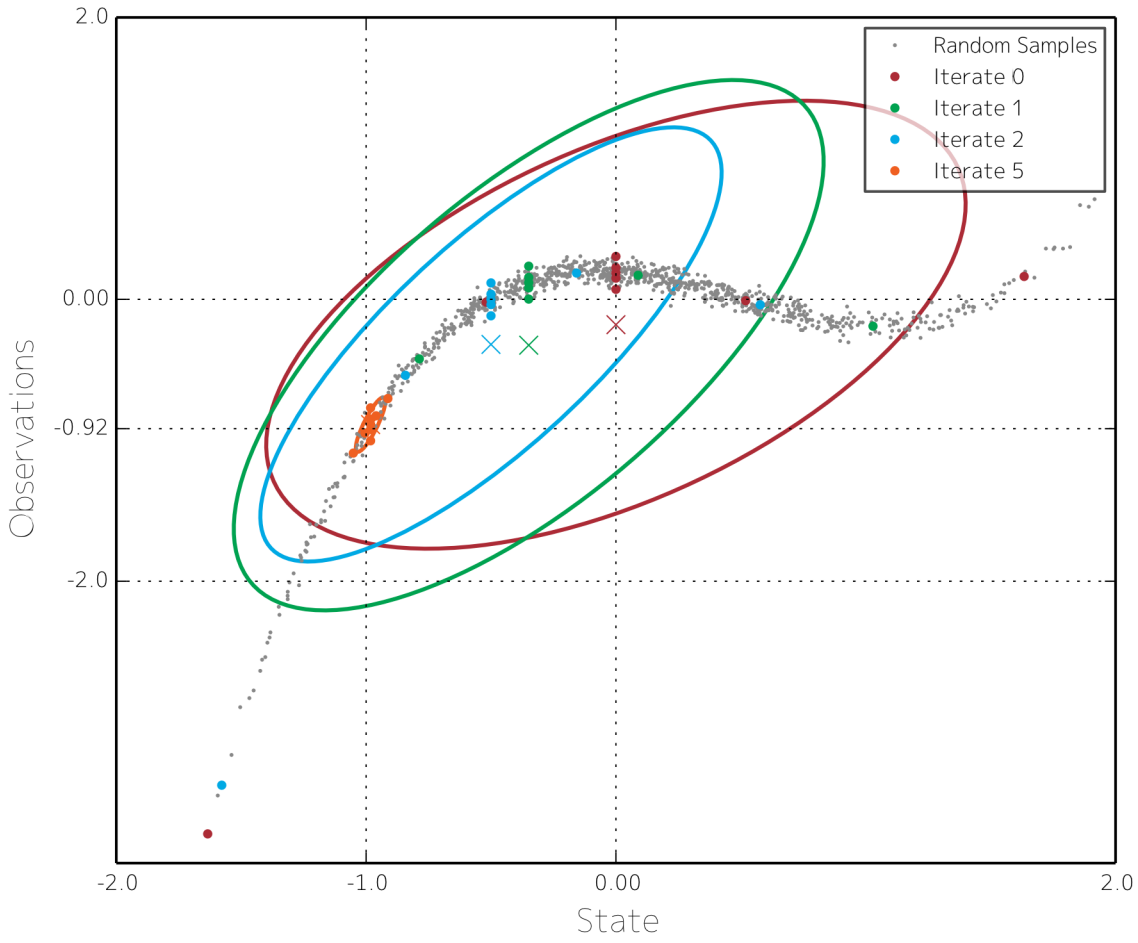

Figure 8.0.1: Iterative UKF Example Realization 1

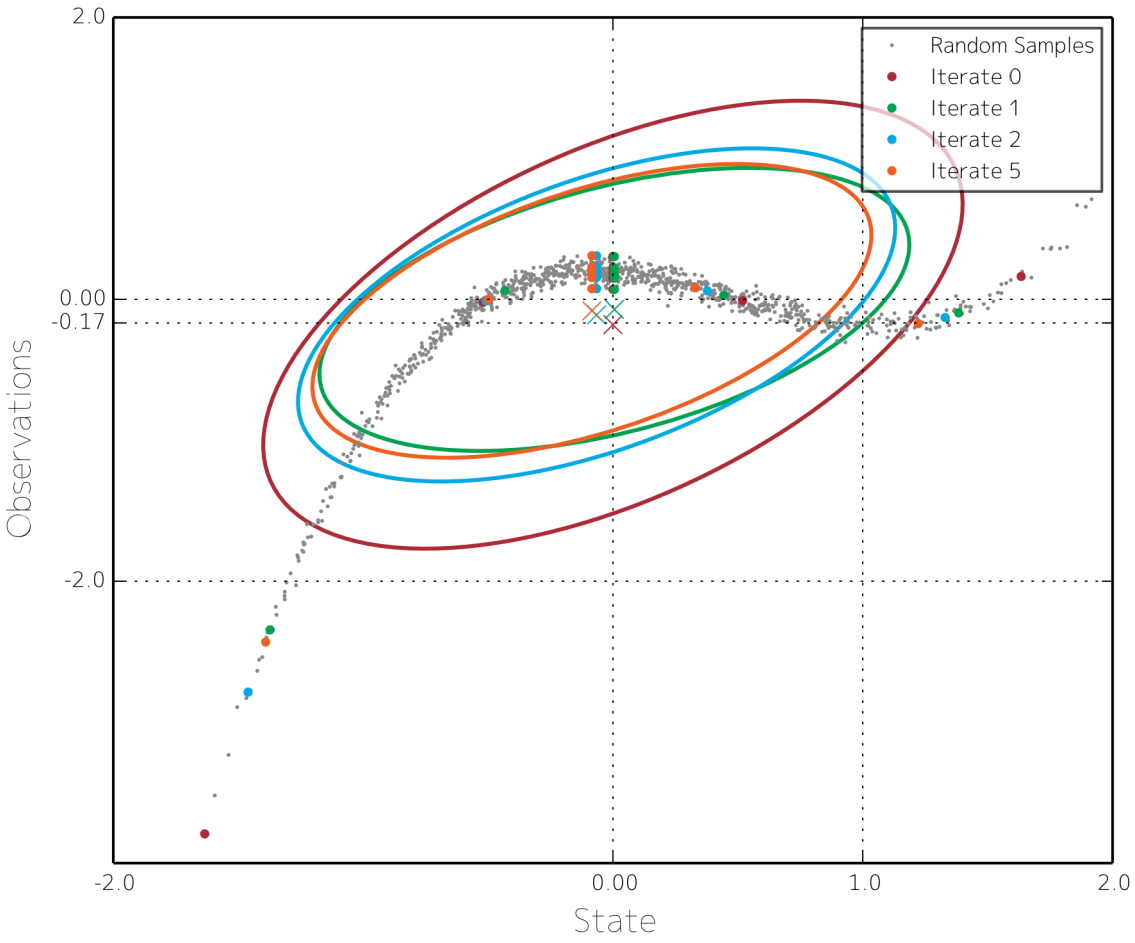

Figure 8.0.2: Iterative UKF Example Realization 2

trajectories close to the merry-go-rounds. The inclusion of the alternative track is a demonstration that this tuning works well on other, similar, scenarios.

Final Remarks This work has brought to light many of the inherent difficulties of solving nonlinear problems with practical solutions based on the Kalman Filter. Its main contributions are in the area of Unscented Kalman Filtering, where it has introduced an entire new way of thinking about the σ-points, which make up the core of the filter, creating sets designed for p-order moments and expanding them to nD; a new way of computing estimates which have a nature and properties between the UKF and the EKF (the benefits of such a scaling are also demonstrated in this work); and by formulating an iterate for the UKF which preserves its benefits. The results of these contributions is remarkable, significantly improving performance in a problem which has challenged other cutting edge methods. The new geometrically derived way of representing and generating σ-points introduced here is certainly an area which requires more examination; the geometries introduced are compelling if not difficult to work with. Also the scaling method introduced requires an element of problem specific tuning which has only been naively considered in the past. This work introduces strategy and new ways to achieve a trade off between accuracy and robustness with the parametric scaling. I consider this still preliminary and suggest tuning, and automatic-tuning are rich areas for further research. These contributions expand the capability of UKF methods to solve more challenging problems, opening up entirely new applications.

Appendix A

Additional SP Lemma

Lemma. *If a set of p σ-points for an n-dimensional system, $\{w, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_n\}$, has b negative weights then* $p \geq n + b + 1$

Proof. To demonstrate this let us consider counting the number of positive entries of w, which will be easiest if we consider counting the number of positive eigenvalues of the diagonal matrix constructed from the entries of w, D_w. Keeping in mind that the transformation BD_wB^{T} maintains the sign of the eigenvalues of original matrix A we construct a basis including $\{w, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_n\}$, $B = \left[w, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_n, \beta_1, \ldots, \beta_{p-n-1}\right]$. This transformation contains many known elements,

$$BD_wB^{\mathrm{T}} = \begin{bmatrix} w\cdot(w\star w) & w\cdot(w\star\mathcal{Y}_2) & w\cdot(w\star\mathcal{Y}_2) & & w\cdot(w\star\mathcal{Y}_n) & w\cdot(w\star\beta_1) & & w\cdot\left(w\star\beta_{p-n-1}\right) \\ w\cdot(w\star\mathcal{Y}_1) & w\cdot(\mathcal{Y}_1\star\mathcal{Y}_1) & w\cdot(\mathcal{Y}_1\star\mathcal{Y}_2) & \cdots & w\cdot(\mathcal{Y}_1\star\mathcal{Y}_n) & w\cdot(\mathcal{Y}_1\star\beta_1) & \cdots & w\cdot\left(\mathcal{Y}_1\star\beta_{p-n-1}\right) \\ w\cdot(w\star\mathcal{Y}_2) & w\cdot(\mathcal{Y}_1\star\mathcal{Y}_2) & w\cdot(\mathcal{Y}_2\star\mathcal{Y}_2) & & w\cdot(\mathcal{Y}_2\star\mathcal{Y}_n) & w\cdot(\mathcal{Y}_2\star\beta_1) & & w\cdot\left(\mathcal{Y}_2\star\beta_{p-n-1}\right) \\ & \vdots & & & & & & \\ w\cdot(w\star\mathcal{Y}_n) & w\cdot(\mathcal{Y}_1\star\mathcal{Y}_n) & w\cdot(\mathcal{Y}_2\star\mathcal{Y}_n) & & w\cdot(\mathcal{Y}_2\star\mathcal{Y}_n) & w\cdot(\mathcal{Y}_n\star\beta_1) & & w\cdot\left(\mathcal{Y}_n\star\beta_{p-n-1}\right) \\ w\cdot(w\star\beta_1) & w\cdot(\mathcal{Y}_1\star\mathcal{Y}_n) & w\cdot(\mathcal{Y}_2\star\beta_1) & & w\cdot(\mathcal{Y}_2\star\beta_1) & w\cdot(\beta_1\star\beta_1) & & w\cdot\left(\beta_1\star\beta_{p-n-1}\right) \\ & \vdots & & & & & & \\ w\cdot\left(w\star\beta_{p-n-1}\right) & w\cdot(\mathcal{Y}_1\star\mathcal{Y}_n) & w\cdot\left(\mathcal{Y}_2\star\beta_{p-n-1}\right) & & w\cdot\left(\mathcal{Y}_2\star\beta_{p-n-1}\right) & w\cdot\left(\beta_1\star\beta_{p-n-1}\right) & & w\cdot\left(\beta_{p-n-1}\star\beta_{p-n-1}\right) \end{bmatrix}$$

Most of the elements in the top left block are known from the identity $w\cdot\left(\mathcal{Y}_i\star\mathcal{Y}_j\right) = \delta_{i,j}$ reducing it to nearly an identity block. If we knew this block was invertible and had all positive eigenvalues we could use a block LDL^{T} decomposition to show that the matrix D_w has at least $n+1$ positive eigenvalues, which paired with its b negative ones means would have at least $b+n+1$ values. To this end consider the substitution of the vector of all ones, 1, for w in B. This substitution will not change the structure of B as w was originally orthogonal to the other required elements $\mathcal{Y}$ and$1\cdot w = \sum w = 1$, meaning $\{w, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_n\}$ and $\{1, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_n\}$ span the same space. The new upper block is an identity matrix.

$$\begin{bmatrix} w \cdot (1 \star 1) & w \cdot (1 \star \mathscr{Y}_1) & w \cdot (1 \star \mathscr{Y}_2) & & w \cdot (1 \star \mathscr{Y}_n) \\ w \cdot (1 \star \mathscr{Y}_1) & 1 & 0 & & 0 \\ w \cdot (1 \star \mathscr{Y}_2) & 0 & 1 & \cdots & 0 \\ & \vdots & & & \\ w \cdot (1 \star \mathscr{Y}_n) & 0 & 0 & & 1 \end{bmatrix} = I$$

□

Bibliography

[1] axiom. Axiom, the scientific computation system, 2013.

[2] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Achademic Press, Inc., New York, 1970.

[3] S.J. Julier and J.K. Uhlmann. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *American Control Conference, 2002. Proceedings of the 2002*, volume 2, pages 887–892 vol.2, 2002.

[4] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Mar 2004.

[5] macaulay2. Macaulay2, a software system for algebraic geometry research. version 1.6, 2013.

[6] Jing PENG, Falin WU, Ming ZHU, Feixue WANG, and Kefei ZHANG. An improved gps/rfid integration method based on sequential iterated reduced sigma point kalman filter. *IEICE Transactions on Communications*, E95.B(7):2433–2441, 2012.

[7] Ksenia Ponomareva and Paresh Date. Higher order sigma point filter: A new heuristic for nonlinear time series filtering. *Applied Mathematics and Computation*, 221(0):662 – 671, 2013.

[8] Samuel J. Prentice. Robust range-based localization and motion planning under uncertainty using ultra-wideband radio. Master's thesis, Massachusetts Institute of Technology, 2007.

[9] M. L. Psiaki. Backward-smoothing extended kalman filter. *Journal of Guidance, Control, and Dynamics*, 28(5), 2005.

[10] M. L. Psiaki. The blind tricyclist problem and a comparative study of nonlinear filters. *IEEE Control Systems Magazine*, 33(2), June 2013.

[11] K. V. Ramachandra. *Kalman Filtering Techniques for Radar Tracking*. CRC Press, 2000.

[12] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[13] Dan Simon. *Optimal State Estimation*. John Wiley and Sons, Hoboken, New Jersey, 2006.

[14] Robert F. Stengel. *Optimal Control and Estimation*. Dover Publications, New York, 1986.

[15] D. Tenne and T. Singh. The higher order unscented filter. In *2003 American Control Conference*, Denver, Colorado, June 2003.

[16] Paul Waltman. *A Second Course in Elementary Differential Equations*. Dover Publications, Mineola, New York, 2004.