Robust Adaptive Estimation for Autonomous Rendezvous in Elliptical Orbit

Christopher D. Karlgaard

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Hanspeter Schaub, Chair
Christopher D. Hall, Co-Chair
Hans Seywald
Cornel Sultan
Craig A. Woolsey

June 28, 2010
Blacksburg, Virginia

Keywords: Robust Estimation; Adaptive Estimation; Rendezvous

Robust Adaptive Estimation for Autonomous Rendezvous in Elliptical Orbit

Christopher D. Karlgaard

(ABSTRACT)

The development of navigation filters that make use of robust estimation techniques is important due to the sensitivity of the typical minimum ℓ_2 norm techniques, such as the Kalman filter, to deviations in the assumed underlying probability distribution. In particular, those distributions with thicker tails than the Gaussian distribution can give rise to erratic filter performance and inconsistency of results. This dissertation discusses the development of an adaptive discrete-time robust nonlinear filtering technique based on a recursive form of Huber's mixed minimum ℓ_1/ℓ_2 norm approach to estimation, which is robust with respect to deviations from the assumed Gaussian error probability distributions inherent to the Kalman filter. This mixed norm approach is applied to a type of Sigma-Point Kalman filter, known as the Divided Difference Filter, which can capture second–order effects of nonlinearities in the system and measurement dynamics.

Additionally, if these assumed parameters of the distribution differ greatly from the true parameters, then the filter can exhibit large errors and possibly divergence in nonlinear problems. This behavior is possible even if the true error distributions are Gaussian. To remedy these problems, adaptive filtering techniques have been introduced in order to automatically tune the Kalman filter by estimating the measurement and process noise covariances, however these techniques can also be highly sensitive to the nature of the underlying error distributions. The Huber–based formulations of the filtering problem also make some assumptions regarding the distribution, namely the approach considers a class of contaminated densities in the neighborhood of the Gaussian density. Essentially the method assumes that the statistics of the main Gaussian density are known, as well as the ratio or percentage of the contamination. The technique can be improved upon by the introduction of a method to adaptively estimate the noise statistics along with the state and state error covariance matrix. One technique in common use for adaptively estimating the noise statistics in real–time filtering applications is known as *covariance matching*. The covariance matching technique is an intuitively appealing approach in which the measurement noise and process noise covariances are determined in such a way that the true residual covariance matches the theoret-

ically predicted covariance. The true residual covariance is approximated in real time using the sample covariance, over some finite buffer of stored residuals. The drawback to this approach is that the presence of outliers and non-Gaussianity can create problems of robustness with the use of the covariance matching technique. Therefore some additional steps must be taken to identify the outliers before forming the covariance estimates. In this dissertation, an adaptive scheme is proposed whereby the filter can estimate the process noise and measurement noise covariance matrices along with the state estimate and state estimate error covariance matrix. The adaptation technique adopts a robust approach to estimating these covariances that can resist the effects of outliers. The particular outlier identification method employed in this paper is based on quantities known as *projection statistics*, which utilize the sample median and median absolute deviation, and as a result are highly effective technique for multivariate outlier identification. These projection statistics are then employed as weights in the covariance matching procedure in order to reduce the influence of the outliers.

The hybrid robust/adaptive nonlinear filtering methods introduced in this dissertation are applied to the problem of 6-DOF rendezvous navigation in elliptical orbit. The full nonlinear equations of relative motion are formulated in spherical coordinates centered on the target orbit. A relatively simple control law based on feedback linearization is used to track a desired rendezvous trajectory. The attitude dynamics are parameterized using Modified Rodrigues Parameters, which are advantageous for both control law development and estimation since they constitute a minimal 3-parameter attitude description. A switching technique which exploits the stereographic projection properties of the MRP coordinate is utilized to avoid singularities which inevitably arise in minimal attitude descriptions. This dissertation also introduces the proper covariance transformations associated with the singularity avoidance switching technique. An attitude control law based on backstepping is employed to track the target vehicle.

A sensor suite consisting of a generic lidar or optical sensor, an Inertial Measurement Unit, consisting of accelerometers and gyroscopes, a star tracker, and a horizon sensor are utilized to provide measurement data to the navigation filters so that the chaser vehicle can estimate its relative state during the rendezvous maneuver. Several filters are implemented for comparison, including the Extended Kalman Filter, First and Second–Order Divided Difference Filters and Huber–based generalizations of these filters that include adaptive techniques for estimating the noise covariances. Monte-Carlo simulations are presented which include both Gaussian and non-Gaussian errors, including mismatches in the assumed noise covariances in the navigation filters in order to illustrate the benefits of the robust/adaptive nonlinear filters. Additionally, computational burdens of the various filters is compared.

Acknowledgments

I am extremely grateful to Dr. Hanspeter Schaub for volunteering his time to serve as my advisor, especially for staying with me after leaving Virginia Tech to take a position at the University of Colorado. I hope that we can continue to collaborate on other research topics in the years to come. I am also grateful to Dr. Chris Hall for serving as my committee co-chair and to Dr. Craig Woolsey and Dr. Cornel Sultan for serving as committee members. I owe a considerable debt of gratitude to Dr. Hans Seywald for taking time from his busy schedule as President of AMA to serve on my committee, including traveling to Blacksburg to participate in my examinations. I am also thankful for the help I have received from Rachel Hall Smith on procedural matters in the department and the university.

Over the course of this research I have had the opportunity to publish several papers in journals. The comments and suggestions by editors and reviewers has been extremely helpful. In particular, I am thankful to Dr. John Crassidis, Dr. Todd Humphreys, Dr. Kathleen Howell, and several anonymous reviewers.

Last but not least I could not have accomplished this research without the support of my friends and family. I am especially thankful to my wonderful wife, Lissa for all her support over the last six years while I have been working full-time at AMA and part-time towards this dissertation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Autonomous Rendezvous of Spacecraft

Rendezvous and docking of spacecraft in orbit has been at the forefront of research and technology development since the beginning of space flight, continuing to the present day. Spacecraft rendezvous and docking has in the past played an important role in the exploration of space, including both crewed and un-crewed missions. For example, the lunar orbit rendezvous mission mode was critical to the success of the Apollo program.[1] The autonomous rendezvous and docking of spacecraft in orbit will remain a critical technology for future space exploration missions.[2] Indeed, autonomous rendezvous and docking was named one of several top technologies required for the future exploration of space in the Exploration Systems Architecture Study.[3] Potential future applications of this technology include robotic sample return missions to the moon and Mars,[4] in–space assembly of modular systems,[5,6] peer-to-peer refueling within constellations,[7] and could potentially serve as an important backup system in the case of an abort scenario involving crewed lunar missions.[8]

Autonomous rendezvous and docking is currently far from being mastered. A recent example in support of this claim can be found in the failure of the Demonstration of Autonomous Rendezvous Technology (DART) mission.[9] This technology demonstration mission called for the DART spacecraft to rendezvous and conduct various proximity operations, with complete autonomy, with the Multiple Paths, Beyond-Line-of-Sight Communications (MUBLCOM) satellite. In addition, the mission goals also included the demonstration of the use of an Advanced Video Guidance Sensor (AVGS) for relative navigation during proximity operations. The DART spacecraft is shown in Fig. 1.1 and and Fig. 1.2. The planned proximity operations are shown in Fig. 1.3.

The DART spacecraft was launched on a Pegasus booster on April 15th 2005, and successfully completed some mid-course maneuvering in preparation for the terminal rendezvous and docking with the MUBLCOM target satellite. After about 11 hours into the mission, the DART spacecraft propellant supply was prematurely depleted, ending the mission with only 11 out of 27 rendezvous and proximity objectives completed.[10] The Mishap Investigation Board (MIB) found that the propellent depletion, and therefore the mission failure, was due primarily to the following factors:[10]

- Poor relative position initialization in the navigation software

*http://www.msfc.nasa.gov/news/dart/index.html, March 7th, 2007

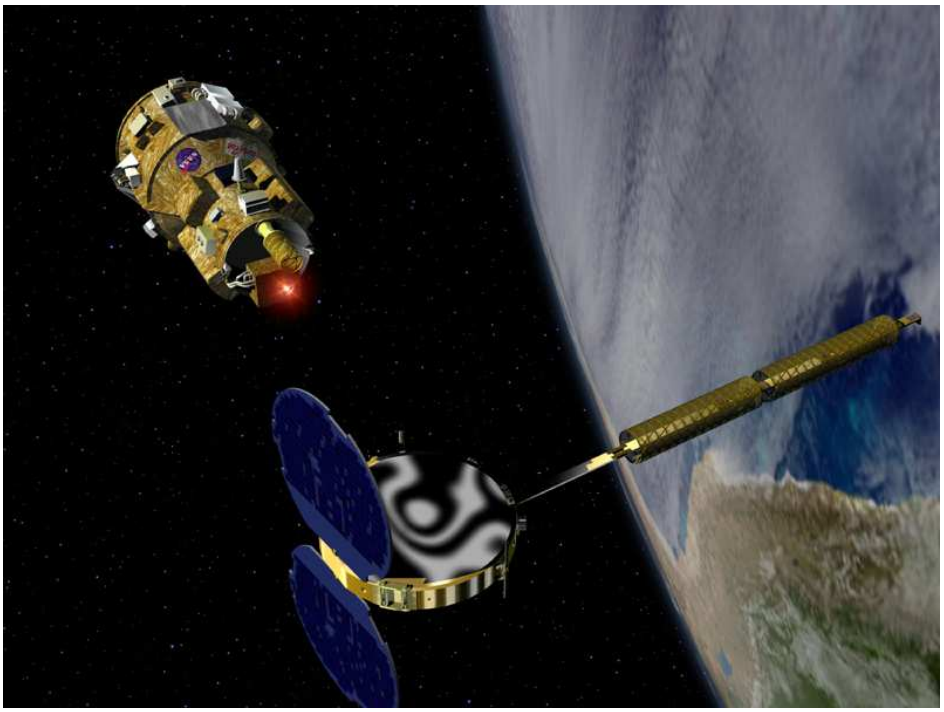


Figure 1.1: Artist Rendering of the DART Spacecraft*

- The introduction of a biased velocity measurement
- Navigation software design that was overly sensitive to erroneous data
- The use incorrect weighting of measurement data in the navigation software

These inadequacies in the DART navigation software led to a buildup of navigation error, which in turn led to a higher than anticipated rate of thruster firing, thus depleting the propellant supply and causing the premature termination of the mission. Some additional factors that contributed to the failure are the guidance system for allowing continual thrusting, as well as faulty calculations of the remaining propellant supply (DART in fact had about 30% fuel remaining when it had computed that the fuel was below the limit, thus terminating the mission[10]).

The DART mishap was not the first instance where navigation sensors and data processing techniques played a role in rendezvous failures or anomalies. The STS–32 mission in January 1990 experienced higher than expected noise in the rendezvous radar measurements, leading to an increase in fuel consumption, although the rendezvous mission was ultimately completed successfully.[14] Investigation into this anomaly found that the radar measurement was biased and was not properly accounted for in the onboard navigation filter software. In another example, the STS–91 mission in June 1998 experienced a series of errors due to faulty processing of Global Positioning System (GPS) measurement data.[14]

[†]http://www.msfc.nasa.gov/news/dart/index.html, March 7th, 2007
[‡]http://www.msfc.nasa.gov/news/dart/index.html, March 7th, 2007

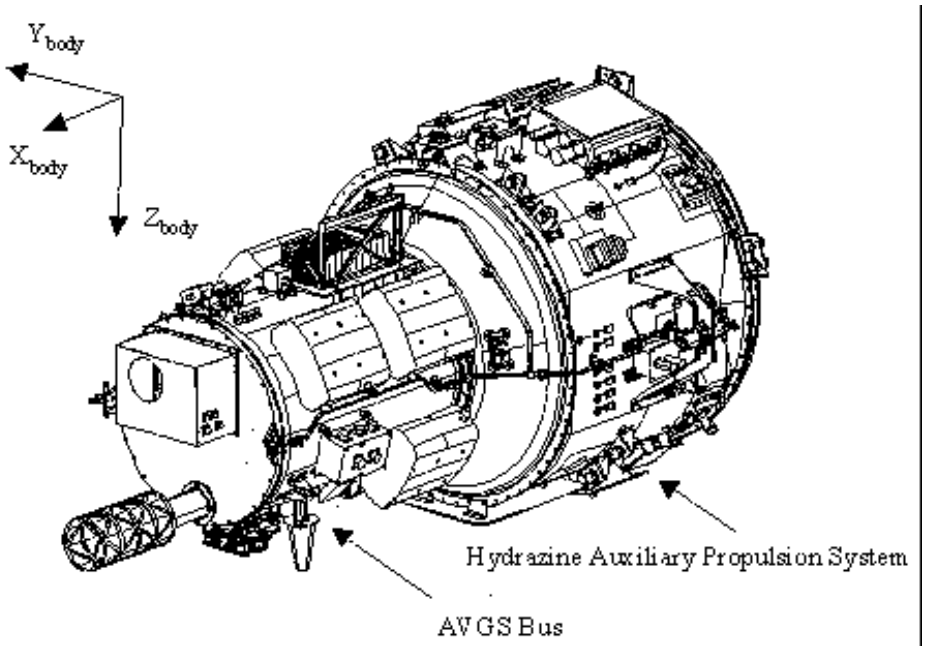


Figure 1.2: DART Spacecraft Configuration[†]

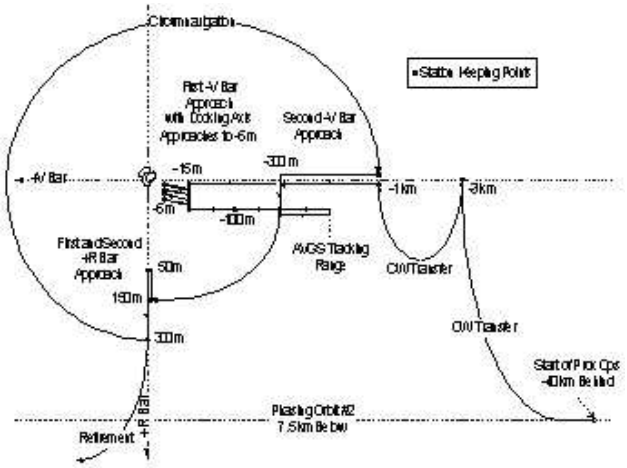
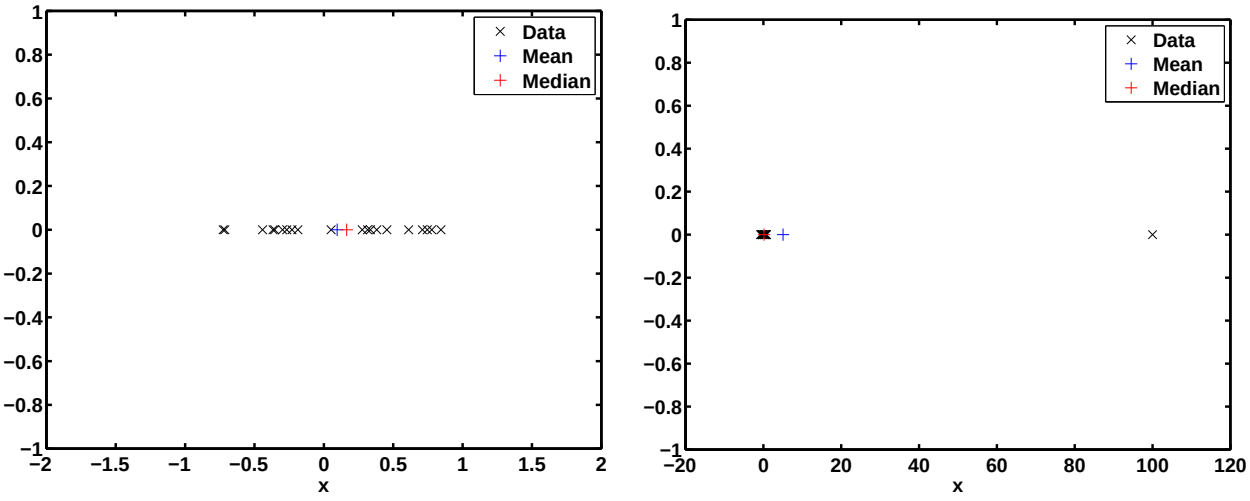


Figure 1.3: DART Proximity Operations[‡]

Based on these rendezvous and docking mission failures and anomalies, there is a need for both improved sensor modeling and the development of navigation filters that are insensitive to data that does not follow the expected distribution of measurement errors. It is also desirable to develop filters that can adapt to changes in these error distributions over time. It is therefore the purpose of this dissertation to develop *robust* filtering and navigation sensor processing techniques for use in spacecraft autonomous rendezvous scenarios.



(a) Data Points

(b) Data Points with Outlier

Figure 1.4: Illustration of the Sensitivity of Least Squares for Point Estimation

1.2 State Estimation

The processing of the various sensor measurements to determine the spacecraft position and velocity relative to the target spacecraft can be accomplished by one of several means. Perhaps the most basic technique in this area of application is the Kalman filter.[15] The Kalman filter is essentially a recursive dynamic weighted least–squares or minimum ℓ_2 norm estimation procedure, and is a maximum likelihood technique assuming that the error statistics follow Gaussian probability distributions.[17,18] The least–squares method is not a robust estimation technique because the procedure can fail to perform adequately when the true error statistics follow non-Gaussian probability distributions, particularly those with much thicker tails than the Gaussian distribution.[19,20] Thickly–tailed probability distributions are those with high probability of large errors compared with the Gaussian distribution. The term *robustness* in a statistical sense was coined by Box[21] to describe procedures that are insensitive to deviations in the assumed underlying probability distribution from which the data are sampled. Formal definitions of robust estimation can be found in Refs. 22–25 and will be summarized in Sec. 2.3.2. Intuitively, robustness means that the estimation error remains bounded for arbitrarily large observation errors.

The sensitivity of least squares can be illustrated with a simple example of a one-dimensional point estimation problem. Suppose 20 samples are taken with uniform random errors in the interval $[-1, 1]$, as shown in Fig. 1.4(a). Here, the mean (least squares estimator) and median (least absolute value estimator) both appear to take on reasonable values, for instance the mean is 0.0962 and the median is 0.1605. Next, the rightmost data point is shifted to the point $x = 100$, as shown in Fig. 1.4(b). In this case, the median value is unchanged whereas the mean is now 5.0539, clearly not representative of the center of the bulk of the data. In this sense, the sample mean is extremely sensitive to erroneous data, whereas the median is said to be robust.

The sensitivity of the least squares technique is also present in higher dimensions. For example, the results of a linear regression problem are shown in Fig. 1.5. In this problem, the data points have an outlier at $x = 25$, $y = 0$. The least squares solution is clearly not a good fit for the bulk

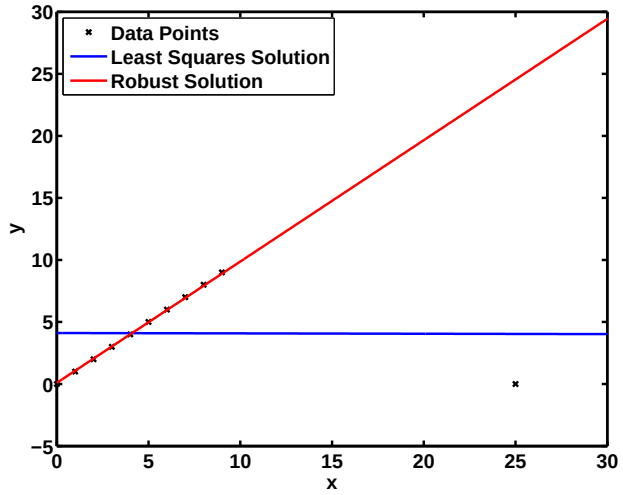


Figure 1.5: Illustration of the Sensitivity of Least Squares for a Linear Regression Problem

of the data, while the robust solution is able to tolerate the presence of the outlier. This example will be discussed in more detail in Sec. 2.7.

1.2.1 Some History of Robust Estimation

The sensitivity of the least squares technique to the true underlying distribution of the data has been known for quite some time. In fact, in the first publication on least squares, Legendre suggests rejecting data that appear erroneous before using least squares on the remaining observations.[29§] Legendre's quote from 1805 is[29]

> *If among these errors are some which appear too large to be admissible, then those observations which produced these error will be rejected, as coming from too faulty experiments, and the unknowns will be determined by means of the other observations, which will then give much smaller errors.*

In other words, Legendre suggests rejecting data that are thicker in the tails than the expected (Gaussian) distribution. Regarding his independent development of the method of least squares published in 1809, Gauss stated in 1821 that[30]

[§]See Plackett[26] and Stigler[27] for a description of the discovery of the least squares method, and subsequent debate between Legendre and Gauss over priority. It is also interesting to note that the least squares technique was independently discovered by Adrain in 1808,[28] *before* Gauss's publication on the method.

> *The author of the present treatise, who in the year 1797 first investigated this problem according to the principles of the theory of probability, soon realized that it was impossible to determine the most probable value of the unknown quantity, unless the function representing the probability of the errors is known. But since it is not, there is no other recourse than to assume such a function in a hypothetical fashion. It seemed most natural to him to take the opposite approach and to look for that function which must be taken as a base in order that for the simplest of all cases a rule is obtained which is generally accepted as a good one, namely that the arithmetic mean of several observations of equal accuracy for one and the same quantity should be considered the most accurate value. This implied that the probability of an error x must be assumed proportional to an exponential expression of the form e^{-hhxx}, and that then just the same method which he found by other considerations already a few years earlier, would become necessary in general. This method, which afterwards, in particular since 1801, he had almost daily opportunity to use in diverse astronomical computations, and which in the meantime also Legendre had happened upon, now is in general use under the name method of least squares.*

It is actually quite shocking that in 1809 Gauss went as far as to claim that the use of the arithmetic mean (minimum ℓ_2 norm) should be regarded as an axiom[31]

> *It has been customary certainly to regard as an axiom the hypothesis that if any quantity has been determined by several direct observations, made under the same circumstances and with equal care, the arithmetical mean of the observed values affords the most probable value, if not rigorously, yet very nearly at least, so that it is always most safe to adhere to it.*

Gauss's claims are questionable, for, even by 1809, it was well known that other estimators were superior to the mean in certain circumstances. In particular, Laplace had shown in 1774 that the median was the minimum–variance estimator for the double exponential (now known as the Laplacian) distribution.[32] It is certainly interesting to note that minimum ℓ_1 norm methods based on weighted medians were in use well before the invention of minimum ℓ_2 norm methods.[33,34] By his own admission, Gauss proposed the normal (Gaussian) distribution out of pure mathematical convenience rather than a statistical or experimental analysis some of random process. Therefore, it should not be expected in reality that *any* process occurring in nature should be Gaussian. In fact, an empirical investigation conducted by Bessel in 1818 showed that astronomical observation data (as one example) was clearly non–Gaussian in nature.[35] In spite of this, a great deal of early estimation theory was rooted in minimum ℓ_2 norm techniques, justified mainly by several elegant mathematical properties of the Gaussian distribution (namely, the Gauss–Markov theorem

and the Central Limit Theorem)[30] ¶. Robust approaches to estimation at the time amounted to cleaning the data with some outlier rejection rule, and applying least squares to the remainder.[29,36] Newcomb[37] seems to have been the first to directly address thickly tailed observational data in 1886 by advocating the use of a mixture of Gaussian densities of differing variance for modeling the underlying distribution.[29,30,36]

A rigorous approach to robust data processing was developed by means of Huber's generalized maximum likelihood estimation theory, first introduced in Ref. 38 in 1964. Huber proposed a combined minimum ℓ_1 and ℓ_2 norm estimation technique, which exhibits robustness with respect to deviations from the commonly assumed Gaussian probability density functions. In particular, Huber considered a class of symmetric contaminated distributions in the neighborhood of the Gaussian, which in some sense can be considered as a generalization of Newcomb's Gaussian mixture models. ‖ The Huber–based estimates are robust in the sense that they minimize the maximum asymptotic estimation variance when applied to contaminated Gaussian densities. The Huber technique was originally developed as a generalization of maximum likelihood estimation, applied first to estimating the center of a probability distribution in Ref. 38 and further generalized to multiple linear regression in Refs. 30, 40 and 41.

The Kalman filter is a recursive minimum ℓ_2 norm technique and therefore exhibits sensitivity to deviations in the true underlying error probability distributions.[20] For this reason, the Huber technique has been further extended to dynamic estimation problems. Boncelet and Dickinson[42] first proposed to solve the robust filtering problem by means of the Huber technique at each measurement point, by expressing the discrete–time filtering problem as a sequence of linear regression problems. The authors do not consider the form of the state error covariance matrix update, nor do they provide any simulation results to demonstrate the benefits of the proposed technique. Kovacevic, *et al*[43] follow the work of Ref. 42 and develop a robust filter using the Huber technique applied to a linear regression problem at each measurement update. Refs. 44–48 express the dynamic filtering problem as a sequential linear regression to be solved by the Huber technique, and apply the filter to underwater vehicle tracking, power system state estimation, speech processing, and spacecraft rendezvous navigation, respectively. The increase in computation due to the use of the Huber technique was found in Ref. 48 to be small, specifically on the order of 10%. Ghandi and Mili[49] develop a robust Huber-based Kalman filter using a pre-whitening filter for outlier identification. It should be noted that Refs. 42–49 apply the Huber methodology to *linearized* filters.

Some other approaches to robust filtering include several ad-hoc techniques for dealing with observation data that do not follow the assumed Gaussian distribution. In particular, that of *censoring* the data such that any observation that differs from the predicted value by some set threshold is discarded entirely. This approach to robust filtering has numerous disadvantages that have been known for quite some time,[50] notably that the non-continuous weighting function can lead to non-robust covariance estimates.[20,22] Another disadvantage of the censoring method is the increase in estimation variance due to the fact that information contained within the residual is discarded entirely and not processed according to a statistical procedure. Another ad-hoc technique

¶The blind adherence to Gaussianity was recently described by Hampel in Ref. 35 as being due to *the mathematicians because they believed it to be an empirical fact, and the users of statistics because the belived it to be a mathematical theorem.*

‖It is interesting to note that, in fact, Newcomb[39] in 1912 proposed an estimator quite similar in form to that proposed by Huber, but without rigorous justification (See Ref. 29).

is to simply inflate the measurement error covariance matrix to account for the perturbing density that leads to the non-Gaussianity. This method also suffers from several drawbacks, namely that *all* measurements processed according to this rule receive decreased weights, not only those that are to be considered as outliers. This reduction in overall weighting in turn leads to an increase in the estimation variance.[20]

Other techniques for robust filtering have been proposed. Sorenson and Alspach[51,52] advocate a Bayesian approach in which the prior and posterior distributions are adaptively approximated using Gaussian sums. The Gaussian sum methodology yields a robust estimation technique, but at an extreme cost in computation. Specifically, the number of terms kept in the Gaussian sum grows exponentially with time. The computation time can be reduced using filter banks and parallel processing techniques,[53–55] but the overall computational complexity across the entire processing system remains substantial. Recent advances in both Gaussian[56] and non–Gaussian[57] particle filtering techniques have done much to reduce the computational burden involving Gaussian sum techniques, but they still may not feasible for real–time implementation in many applications. Specifically, the Gaussian particle filter in Ref. 56 can cost between 15 to over 150 times the computation of the Kalman filter filter in the example problems, depending on the number of particles. Computational cost for the Gaussian sum particle filter in Ref. 57 are not provided but are likely to be even greater.

Masreliez and Martin[58] develop a robust filter by means of a Bayesian approach in which either the state noise is Gaussian and the measurement noise is non-Gaussian, or the state noise is non-Gaussian and the measurement noise is Gaussian. No consideration is made for the case when both noise sources are non-Gaussian, which is allowed for under the Huber filtering approach. Tsai and Kurz[59] propose an adaptive polynomial approximation technique for robustification of the Kalman filter. The approach is only applicable to estimation problems where the measurements are uncorrelated, and also only applies to cases where either the measurement or process noise is Gaussian. Hewer *et al*[60] propose a robust batch prefiltering technique, in which the raw measurement data are smoothed prior to processing in a standard Kalman filter. Meinhold and Singpurwalla[61] also approach the robust Kalman filtering problem from a Bayesian perspective, but instead advocate the use of Student–t distributions for approximating the prior and posterior densities, but the results hold only for scalar systems. Niehsen[62] develops a maximum–likelihood filtering technique based on a generalized Gaussian density.

1.2.2 Adaptive State Estimation

In addition to the form of the error probability density function (the Gaussian), the standard Kalman filter also assumes that the statistics of the distribution, namely its mean and covariance, are known quantities. If these assumed parameters of the distribution differ greatly from the true parameters, then the filter can exhibit large errors and possibly divergence.[63] This behavior is true even if the true error distributions are Gaussian. The Huber–based formulations of the filtering problem also make some assumptions regarding the distribution. As discussed in the previous section, the Huber approach considers a class of contaminated densities in the neighborhood of the Gaussian density. Essentially the Huber approach assumes that the statistics of the main Gaussian density are known, as well as the ratio or percentage of the contamination (although the filter makes no assumption of the nature of the contaminating density other than it be symmetric with

finite variance).

Adaptive filtering techniques have been introduced in which the error statistics are estimated along with the state in real time, in order to reduce the possibility of divergence and to improve the filter performance in the presence of unknown statistics of the underlying noise distributions. There are several approaches to the adaptive state estimation problem. An early review article by Mehra[64] divides these approaches into four basic techniques, namely Bayesian, maximum likelihood, correlation matching, and covariance matching methods. In the Bayesian case and the maximum likelihood case, the techniques become extremely complicated as the dimension of the measurement noise and process noise covariances matrices increase. For this reason, the Bayesian and maximum likelihood approaches can both be collapsed into a problem of estimating the Kalman gain matrix directly, without estimating the measurement and process noise covariances matrices, which saves on a great deal of the computational complexity inherent to the problem. This is approach, however, is disadvantageous in general since it does not directly permit evaluation of sensor performance or model uncertainty, which are parameters of interest in obtaining through the measurement and process noise covariances.

The correlation matching technique operates by attempting to correlate the observed output of a system to the unknown noise covariances. Methods can be developed using either the autocorrelation of the output or that of the residuals. The approach using the output is generally more restrictive, so in practice usually the residuals–based approach is preferred. In both cases, the estimates of the process noise covariance are not unique, and moreover they can only be computed in steady state conditions.

The covariance matching technique is an intuitively appealing approach in which the measurement noise and process noise covariances are determined in such a way that the true residual covariance matches the theoretically predicted covariance. The true residual covariance is approximated in real time using the sample covariance, over some finite window of stored residuals. The solution provided by Mehra[64] leads to a non-unique estimate of the process noise covariance, as with the correlation matching method. For this reason, the author states that, at the time, this approach has exhibited best success when the process noise covariance is known and one only wishes to solve for the measurement noise covariance. An example of this approach is given in Ref. 65.

The covariance matching technique is expanded by Myers and Tapley in Ref. 66. In this approach, the authors are able to determine explicit solutions for both the process noise and the measurement noise covariances by using empirical estimators based on the sample covariance for a finite window of stored observations. The authors also introduce a fading memory weighting parameter in which more recent observations receive more weight than the older observations. The estimators are derived in batch form, but are manipulated into a recursive form suitable for real–time implementation. The approach is not computationally intensive, requiring only 12% more cost than the standard EKF in one sample problem.

Special cases of the Myers–Tapley method appear in the literature. For example, Maybeck *et al*[67] and Whitmore and Leondes[68] propose a covariance matching method to estimate only the process noise covariance matrix, assuming the measurement noise covariance is known. In contrast, Hull *et al*[69] devise a special case of the Myers–Tapley method in which the process noise covariance is known and the measurement noise covariance is estimated in real time.

Few authors have developed combined robust and adaptive approaches to state estimation by blending the Huber approach for robust estimation with the Myers–Tapley approach for adaptive

estimation. In particular, Groutage *et al*[70] approach the adaptive state estimation problem by applying a robust processing technique based on Huber's generalized maximum likelihood estimation to develop a data smoother. This data smoother has a similar structure to that proposed by Myers and Tapley. The robust smoother processes outputs from a standard Kalman filter in order to estimate the process noise variance. The method has a major drawback in that it can only be applied to scalar dynamic systems. Kirlin and Moghaddamjoo[71] develop a similar approach as that of Ref. 70, but include the estimation of unknown step inputs. The adaptive filter in Ref. 71 is essentially the same as the Myers–Tapley method, with the difference that robust scale estimates are used in place of the sample covariance. The approach is more general than that of Ref. 70 in that both process noise and measurement noise covariances can be estimated. The drawback of this approach is that the results are valid only for scalar linear systems.

Durovic and Kovacevic[72] develop a methodology for estimating the measurement and process nosie covariances based on robust principles, and include this procedure in a recursive dynamic Huber filter with linear state dynamics and measurement equations. The approach is essentially a merger of Ref. 42 and Ref. 70. The proposed filter adaptively estimates the noise statistics along with the state in a structure similar to that proposed by Myers and Tapley in Ref. 66, however, the approach is only applicable to scalar measurement and process noise cases. Zou *et al*[73] develop a robust adaptive signal processing technique based on Huber's generalized maximum likelihood estimation with robust adaptive scale estimates used in place of the sample covariance, but the method is only applicable to scalar input and output noise. To date, combined robust and adaptive approaches to the dynamic state estimation problem have only been developed for scalar systems.

1.2.3 Recent Advances in State Estimation

This section describes some recent advances in the area of state estimation, that are not necessarily robust or adaptive in nature, and have been developed using the standard minimum ℓ_2 norm approach to stochastic estimation problems.

One such technique is the Backward Smoothing Extended Kalman Filter (BSEKF),[74] which is a sort of combined filter, smoother, and batch estimator with sliding window of measurement data to be processed. The BESKF essentially operates by relinearizing previous measurement data about the state estimates from the most current observation that are propagated back to the individual measurement points. These relinearized measurement points are then processed according to an EKF–type minimum ℓ_2 norm technique. The simulation results presented in Ref. 74 show that for a spacecraft attitude estimation problem with a reasonable measurement data frame size, the computational requirements are over 150 times greater than the standard EKF. Based on this computational burden, and given the non–robust minimum ℓ_2 norm nature of the method, there does not appear to be any advantage in the use of the BSEKF over, for example, the Gaussian sum particle filter presented in Ref. 57.

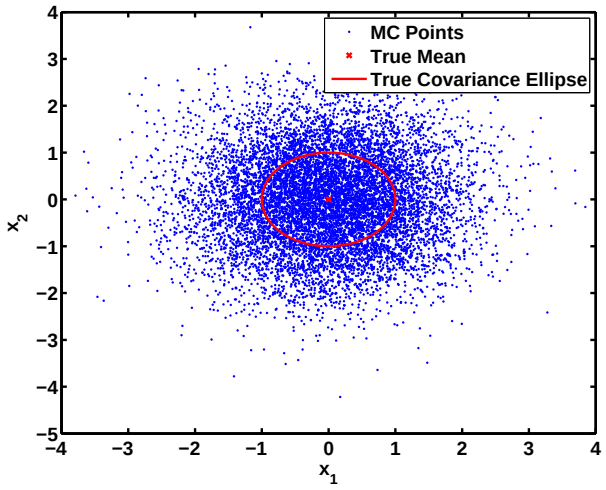
The Two–Step Optimal Estimator (TSOE) is a technique developed initially in Ref. 75 and further developed in Refs. 76–78. The TSOE operates by dividing the filter update into two subproblems to be solved in sequence. The first subproblem is the standard linear Kalman filter, which can be solved explicitly. The second subproblem treats the solution to the first subproblem as "measurements" to be processed using an iterative nonlinear least squares method. The product of this nonlinear second subproblem is then taken as the state estimate following the measurement

update. The filter as originally developed[75] suffered from occasional ill-conditioned covariances matrices,[77] which was remedied in Ref. 78. In Ref. 78, it was found that the TSOE costs about 2-3 times the computation of the EKF for the particular example problem, which is a reasonable penalty for the nonlinear measurement update processing. However, the state dynamics are assumed to be linear and the measurement error distribution is assumed to be Gaussian, which implies the TSOE suffers from the same lack of robustness as the basic Kalman filter.

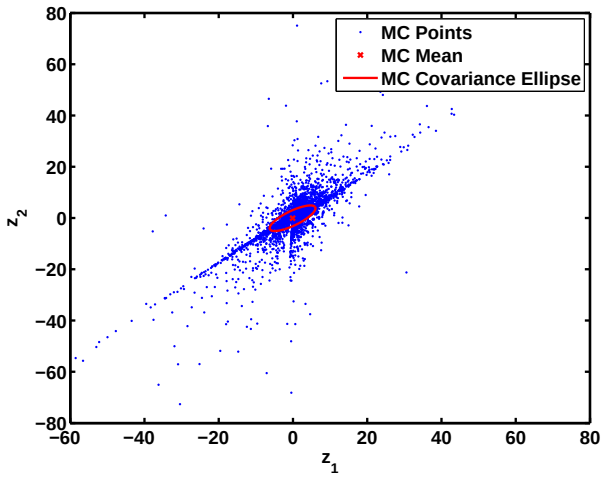
Another recent advancement is an estimator known as the divided difference filter (DDF), which is one of several new estimation techniques that are collectively known as sigma–point Kalman filters (SPKF). The first–order (DD1) and second–order (DD2) divided difference filters[79,80] are generalizations of the filter introduced by Schei[81] and are two examples of SPKF–class estimators; other examples can be found in Refs. 82–85. Like the basic Kalman filter, the SPKFs seek to determine a state estimate that minimizes the ℓ_2–norm of the residuals. The SPKF technique differs from the standard Kalman filter in the sense that the SPKFs do not linearize the dynamic system for the propagation, but instead propagate a cluster of points centered around the current estimate in order to form improved approximations of the conditional mean and covariance. Specifically, the divided difference filters make use of multidimensional interpolation formulas to approximate the nonlinear transformations. As a result of this approach, the filters do not require knowledge or existence of the partial derivatives of the system dynamics and measurement equations. SPKFs have the additional advantage over the standard Kalman filter in that they can easily be extended to determine second–order solutions to the minimum ℓ_2 norm filtering problem, which increases the estimation accuracy when the system and/or measurement equations are nonlinear. It is important to note that the SPKFs use a minimum ℓ_2–norm measurement update and are therefore subject to the same sensitivity to non-Gaussian measurement errors as the Kalman filter. An adaptive SPFK filter that uses the Myers–Tapley covariance matching technique is proposed in Ref. 86.

The improvement found using a SPKF technique in favor of the Kalman filter can be shown in the following simple example. Given some transformation $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$ where $\boldsymbol{x}$ is a random variable with known distribution and $\boldsymbol{f}(\boldsymbol{x})$ is a nonlinear transformation, several methods can be used to approximate the covariance of the transformed variable $\boldsymbol{z}$. One technique to determine a highly accurate approximate solution is a Monte-Carlo approach in which several thousand random choices of $\boldsymbol{x}$ are generated, with each being propagated through the transformation $\boldsymbol{f}(\boldsymbol{x})$. Then the sample covariance of the transformed points can be calculated. This approach gives a good approximation to the true covariance of variable $\boldsymbol{z}$ but at a usually extreme cost due to the number of calculations that must be made. On the other extreme, a crude approximation can be found by linearizing the transformation and using basic matrix transforms to calculate the covariance. This approach, which is essentially the Kalman filter strategy, works well for nearly linear systems but breaks down in the presence of nonlinearity. A third approach is to use a Divided–Difference transformation in which a select few points are propagated through the transformation to calculate the covariance. In this approach, the transformed covariance is calculated more precisely than that found using a linearization, but requires far fewer points than the Monte-Carlo method.

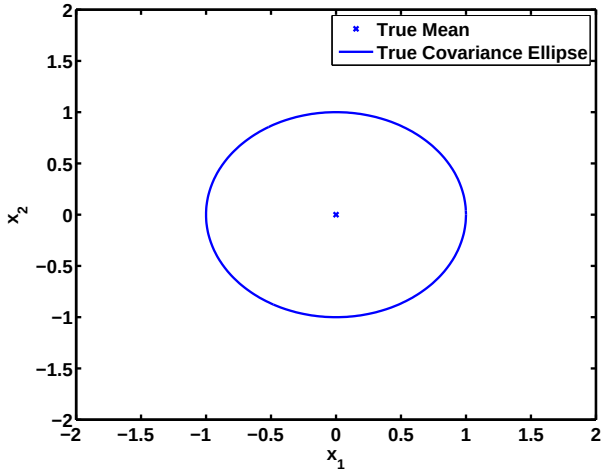
This trade off is illustrated in Fig. 1.6. In this example the initial variable $\boldsymbol{x} = [x_1, x_2]$ where x_1 and x_2 are mutually uncorrelated Gaussian random variables with unit variance. The transformation is given by $z_1 = x_1 + x_1^3 + x_1 x_2$, and $z_2 = x_2 + x_1^3 + x_1 x_2^3 + x_1 x_2$. The "true" covariance of the transformed state $\boldsymbol{z}$ is determined using the Monte-Carlo method shown in Fig. 1.6(a)–(b). In this case, 10,000 points are selected at random and propagated through the transformation in order to



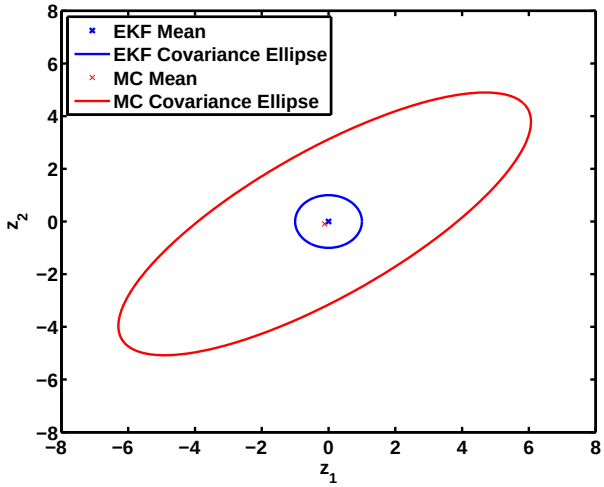
(a) Initial Monte-Carlo Distribution



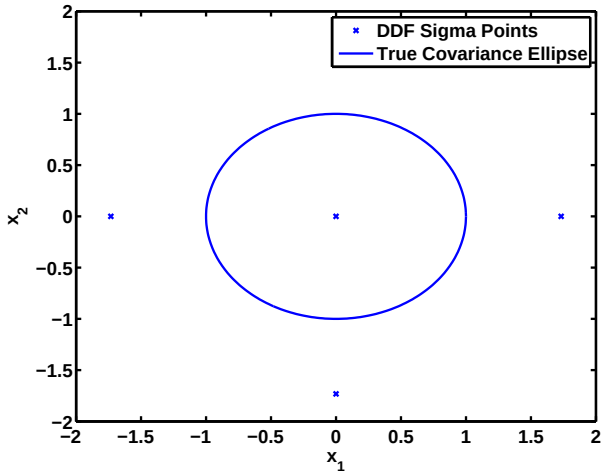
(b) Transformed Monte-Carlo Distribution



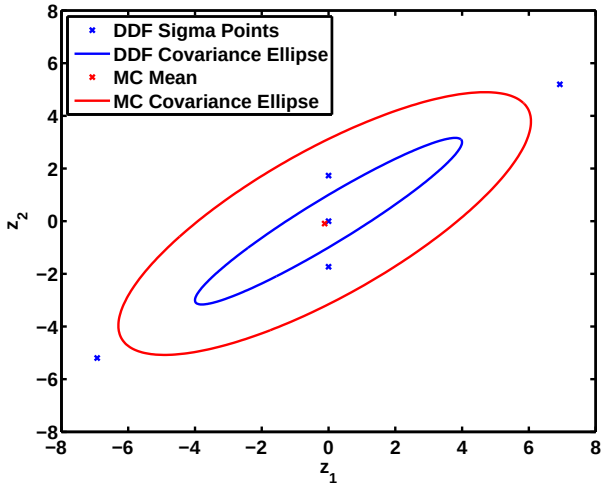
(c) Initial Kalman Filter Distribution



(d) Transformed Kalman Filter Distribution



(e) Initial Divided Difference Filter Distribution



(f) Transformed Divider Difference Filter Distribution

Figure 1.6: Illustration of Divided Difference Filter Transformation

calculate the covariance. Next, a linearized mapping similar to the Kalman filter approach is used to approximate the transformation, shown in Fig. 1.6(c)–(d). In this case the linear transformation clearly does not match the "truth" covariance determined using the Monte-Carlo method. Lastly, the Divided–Difference transformation is used to approximate the transformed covariance, shown in Fig. 1.6(e)–(f). In this case, only 5 points are used in the propagation, and the resulting covariance propagation is much more accurate than the linearized result. This example serves to illustrate the benefits of the DDF over that of the EKF.

1.3 Attitude Estimation

Attitude estimation techniques often make use of quaternions for the representing the attitude, for several reasons including globally nonsingular kinematics and linear state propagation as discussed in the previous section.[87,88] However, techniques making use of quaternions as state variables are complicated by the quaternion constraint. The usual approach to satisfying the constraint is to estimate an error quaternion at each measurement update and then form the true quaternion estimate from the composition of the estimated error quaternion with the predicted quaternion based on the state transition matrix. Assuming small errors allows for the first three components of the quaternion to be estimated independently of the fourth component, which is essentially amounts to a linearization using small angle assumptions. Recently, constrained filtering approaches have been investigated by Zanetti and Bishop[89] and Majji and Mortari[90] (also see Ref. 91). These approaches use a Lagrange multiplier formulation to solve a constrained filtering problem for all four components of the error quaternion, rather than using a linearization in order to enforce the quaternion norm constraint.

Other attitude parameterizations can be used, provided that a singularity avoidance method is employed to provide a valid attitude description at any condition. One representation with several attractive features are the Modified Rodrigues Parameters (MRP).[92] The MRPs have several interesting properties. Firstly, the MRPs constitute a minimal three parameter set of variables that describe the orientation of a rigid body and are nonsingular for any rotation other than multiples of 2π. Tsiotras and Longuski[93] discuss that the MRPs can be viewed as the result of a stereographic projection of the unit quaternion sphere onto a three-dimensional hyperplane. Schaub and Junkins[94] use this insight to formulate a family of attitude coordinates called the Stereographic Orientation Parameters (SOP), which contain the MRPs as one particular solution of symmetric SOPs. As part of this development it is noted that the MRPs are not unique, but rather there are always two possible MRP sets that can describe a particular orientation. This alternate MRP is known as the *shadow* MRP set. The shadow MRP set is singular for zero rotations, but is non-singular for rotations of 2π. This property allows for the development of a singularity avoidance method by switching to and from the shadow MRP set. For example, this switching procedure allows for non–singular optimal attitude control problems to be formulated using a minimal three–parameter family of MRPs as discussed in Ref. 95, in which an analytical mapping is developed for the MRP costates.

The application of MRPs to attitude estimation was first explored in Ref. 96 without discussion of singularity avoidance. Other examples make use of MRPs for representing attitude error rather than the global attitude, preferring to keep track of the quaternion.[97,98] In these cases the MRP singularity is never encountered in practice but the additional computations to transform the MRP

error estimate to the quaternion may not always be desirable. The two MRP sets are applied to attitude estimation problems as a singularity avoidance procedure in Refs. 99–101. In these cases, the transformation of the covariance matrix at the switching point has been ignored, although it is not actually required in the particle filtering approach utilized in Ref. 99.

1.4 Contributions

The purpose of this dissertation is to develop robust adaptive state estimation techniques for nonlinear dynamic systems. The robust technique investigated in this research is based on Huber's generalized maximum likelihood estimation method, which is further generalized for dynamic filtering problems. A new form of the state update equation for first and second–order divided–difference filtering is found by recasting the filtering problem into a linear regression problem at each measurement update and solving the resulting system of equations using Huber's generalized maximum likelihood theory. A new robust adaptive procedure for real–time tuning of the noise covariances in the filter is found, as a modified version of the Myers–Tapley covariance matching approach. This new adaptive technique makes use of multidimensional outlier detection methods to remove spurious observations in the stored residual sequence in order to compute robust estimates of the noise covariance matrices.

This dissertation also introduces a covariance transformation to accompany the shadow MRP mapping for singularity avoidance in attitude estimation problems. The covariance transformation is introduced for Kalman filtering problems by using a first–order analytical mapping of the MRP covariance to and from the shadow MRP set. Subsequently, a divided difference covariance transformation is introduced, suitable for the first and second–order divided difference filters introduced in Refs. 79 and 80. This singularity avoidance technique offers some advantages for spacecraft attitude estimation problems, not the least of which is the fact that a minimal three parameter set of attitude variables can be used a globally nonsingular attitude description.

Lastly, the hybrid robust/adaptive nonlinear filtering methods introduced in this dissertation, along with the MRP attitude formulation, are implemented for 6-DOF rendezvous navigation in elliptical orbit. Relatively simple control laws based on feedback linearization are implemented, with these various filtering approaches implemented as observers inside the control loop. These navigation and control methods are tested in a non-Gaussian simulation setting in order to illustrate performance robustness of the techniques in situations where typical assumptions inherent to the navigation state estimation problem are violated.

1.5 Overview

The remainder of this document is divided as follows. Chapter 2 discusses the generalized maximum likelihood estimation technique as it applies to static linear regression problems. Chapter 3 discusses the application of the Huber–based estimation technique to dynamic state estimation problems, including first and second–order filters and adaptive techniques in which the estimator can solve for the measurement and process noise covariances in real time. Here, outlier identification techniques are introduced to provide robustness to the adaptive filtering approach in the presence of non-Gaussian errors. Chapter 4 develops the 6-DOF equations of motion, guidance and control methods,

and sensor models suitable for the rendezvous problem in elliptical orbits. Chapter 5 provides simulation results to illustrate the performance of the robust and adaptive filtering methods applied to spacecraft attitude estimation (including comparisons between quaternion-based approaches and the MRP-based approach advocated in this dissertation) and to the 6-DOF rendezvous problem in an elliptical lunar orbit.

Chapter 2

Some Concepts in Regression Theory

2.1 Introduction

The purpose of this section is to review some basic concepts in regression theory. The estimation of a static linear system will later be shown to be easily extended to discrete-time dynamic estimation. In this section the linear regression model of the form

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w} \tag{2.1}$$

is considered, where $\boldsymbol{y}$ represents the measurement data, $\boldsymbol{x}$ is the variable to be estimated, $\boldsymbol{H}$ represents the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ and is usually called the design matrix, and $\boldsymbol{w}$ represents the random measurement error.

2.2 Maximum Likelihood Estimation

2.2.1 Overview of the Method

The method of maximum likelihood estimation was introduced by Fisher[102] in a series of papers between 1912 and 1922, although some of the basic ideas had been touched on previously by Bernoulli[103] for semi-circular distributions, and Gauss [104] in connection with an effort to determine the probability distribution that made the least squares estimate the most probable estimate.[30,31] The maximum likelihood method proceeds as follows. Given a sample of m measurement data $\boldsymbol{y} = [y_1, \cdots, y_m]^T$, let the joint probability density function of the measurement data for a given value of $\boldsymbol{x}$ be denoted by $f(\boldsymbol{y}|\boldsymbol{x})$. The likelihood function L is then defined as

$$L(\boldsymbol{x}; \boldsymbol{y}) = f(\boldsymbol{y}|\boldsymbol{x}) \tag{2.2}$$

If the observation data are considered to be independent and identically distributed, then the joint density function is the product of the marginal density functions.[105] Also, the density function reduces to a function only of the residuals between the estimate and the observation data, defined as $\boldsymbol{\zeta} = \boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}$. Under these assumptions, the likelihood function takes the form

$$L(\boldsymbol{x}; \boldsymbol{y}) = \prod_{i=1}^{m} f(\boldsymbol{\zeta}) \tag{2.3}$$

The most probable value of $\boldsymbol{x}$ (the mode of the joint distribution) is the value of $\boldsymbol{x}$ that maximizes the likelihood function. In practice it is more convenient to minimize the natural logarithm of the likelihood function. Such a value of $\boldsymbol{x}$ can be found by minimizing the cost function

$$J(\boldsymbol{x}) = -\sum_{i=1}^{m} \ln\left[f\left(\zeta_i\right)\right] \tag{2.4}$$

If the density function is differentiable, then the solution to the maximum likelihood regression problem can be found from the from the implicit equation

$$\sum_{i=1}^{m} \phi(\zeta_i) \frac{\partial \zeta_i}{\partial \boldsymbol{x}} = \boldsymbol{0} \tag{2.5}$$

where $\phi(\zeta_i) = -f'(\zeta_i)/f(\zeta_i)$. By defining the function $\psi\left(\zeta_i\right) = \phi(\zeta_i)/\zeta_i$, and the matrix $\boldsymbol{\Psi} = \operatorname{diag}\left[\psi\left(\zeta_i\right)\right]$, the implicit equation can be written in matrix form as

$$\boldsymbol{H}^T \boldsymbol{\Psi} \left(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\right) = \boldsymbol{0} \tag{2.6}$$

Note that, in general, the implicit likelihood equation (2.6) is nonlinear, since the matrix $\boldsymbol{\Psi}$ is a function of $\boldsymbol{x}$. Solutions of the likelihood equation can be determined by the application standard iterative techniques for nonlinear systems. The solution of the likelihood equation is the maximum likelihood estimate for $\boldsymbol{x}$ and is denoted by $\hat{\boldsymbol{x}}$.

Once the solution of the maximum likelihood method is obtained, the covariance associated with the estimate can be computed from[22]

$$\hat{\boldsymbol{P}} = \frac{\mathrm{E}\left(\phi^2\right)}{\left[\mathrm{E}\left(\phi'\right)\right]^2} \left(\boldsymbol{H}^T \boldsymbol{H}\right)^{-1} \tag{2.7}$$

It can be shown that estimators of this form are asymptotically normal with mean $\hat{\boldsymbol{x}}$ and covariance $\hat{\boldsymbol{P}}$. Further, the estimator is minimum variance and asymptotically unbiased, when the true measurement error distribution follows the assumed distribution exactly, and when the assumed distribution is unimodal. See Ref. 106 for further details.

2.2.2 Correlated Residuals

The development of the maximum likelihood estimation method made the assumption that the residuals were independent and identically distributed. If in fact the residuals are correlated then the maximum likelihood method is still applicable, but a transformation of variables is required. If the measurement error covariance matrix is given by $\boldsymbol{R}$, then the regression problem can be transformed to uncorrelated coordinates (this transformation is sometimes known as stochastic decoupling) by defining the new variables

$$\widetilde{\boldsymbol{w}} = \boldsymbol{R}^{-1/2}\boldsymbol{w} \tag{2.8}$$

$$\widetilde{\boldsymbol{y}} = \boldsymbol{R}^{-1/2}\boldsymbol{y} \tag{2.9}$$

$$\widetilde{\boldsymbol{H}} = \boldsymbol{R}^{-1/2}\boldsymbol{H} \tag{2.10}$$

Upon defining the above quantities, the regression problem is transformed into

$$\widetilde{\boldsymbol{y}} = \widetilde{\boldsymbol{H}}\boldsymbol{x} + \widetilde{\boldsymbol{w}} \tag{2.11}$$

In the transformed problem, the measurement errors have a covariance matrix given by $\widetilde{\boldsymbol{R}} = \mathbf{I}$, as can be seen by expanding the expectation $\mathrm{E}\left[\widetilde{\boldsymbol{w}}\widetilde{\boldsymbol{w}}^T\right]$. This fact implies that the transformed residuals are linearly independent and hence that the standard maximum likelihood method can be applied to the transformed linear regression problem.

2.2.3 Consistency, Efficiency, and Bias

This section discusses several properties of estimators, namely consistency, efficiency, and bias. These properties are important qualitative measures of estimator performance.

2.2.3.1 Consistency

An estimator $\hat{\boldsymbol{x}}$ is said to be consistent if the value of the estimator converges to the true value of the parameter as the sample size goes toward infinity. Consistency can be expressed mathematically as

$$\lim_{n\to\infty} \hat{\boldsymbol{x}} = \boldsymbol{x} \tag{2.12}$$

where n is the number of observations processed by the estimator $\hat{\boldsymbol{x}}$.

2.2.3.2 Efficiency

A useful metric for comparing estimators is the relative efficiency, which is the ratio of the maximum likelihood asymptotic variance and the asymptotic variance of the estimator. If the asymptotic variance of the maximum likelihood estimator is denoted by $\hat{P}_{MLE}$, then the asymptotic relative efficiency is defined as $e = \hat{P}_{MLE}/\hat{P}$, where $\hat{P}$ is the asymptotic variance of the estimator in question. Note that the best possible asymptotic relative efficiency is $e = 1$, since the maximum likelihood estimator is asymptotically minimum variance at the particular distribution.

2.2.3.3 Bias

An estimator $\hat{\boldsymbol{x}}$ is said to be unbiased if $B\left(\hat{\boldsymbol{x}}\right) = E\left(\hat{\boldsymbol{x}}\right) - \boldsymbol{x} = \mathbf{0}$. For a consistent estimator, the mean–squared error, variance, and bias are related via

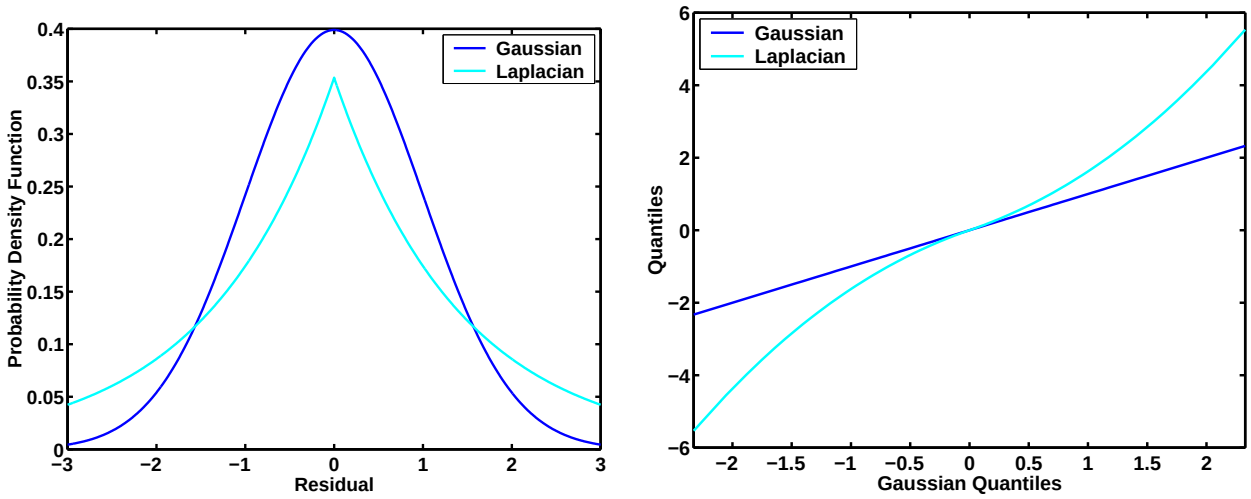
$$\begin{aligned} E\left[\left(\hat{\boldsymbol{x}} - \boldsymbol{x}\right)\left(\hat{\boldsymbol{x}} - \boldsymbol{x}\right)^T\right] &= E\left[\left(\hat{\boldsymbol{x}} - E\left(\hat{\boldsymbol{x}}\right)\right)\left(\hat{\boldsymbol{x}} - E\left(\hat{\boldsymbol{x}}\right)\right)^T\right] + \left(E\left(\hat{\boldsymbol{x}}\right) - \boldsymbol{x}\right)\left(E\left(\hat{\boldsymbol{x}}\right) - \boldsymbol{x}\right)^T \\ &= \hat{\boldsymbol{P}} + B\left(\hat{\boldsymbol{x}}\right) B\left(\hat{\boldsymbol{x}}\right)^T \end{aligned} \tag{2.13}$$

Note that an unbiased estimate implies that the mean-squared error is equal to the variance.

2.2.4 Example: Maximum Likelihood Estimator for the Gaussian Distribution

As an example of maximum likelihood regression, consider the case when the error distributions are Gaussian. Specifically, when the marginal density function takes the form

$$f\left(\zeta_i\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\zeta_i^2}{2\sigma^2}\right] \tag{2.14}$$



(a) Density Functions (b) Theoretical Quantile–Quantile Plot

Figure 2.1: Unit Variance Gaussian and Laplacian Distributions

where σ^2 is the variance of the distribution. Note that the center of the density is assumed, without loss of generality, to be located at the origin. With $f(\zeta_i)$ given by Eq. (2.14), the the function $\phi(\zeta_i)$ takes the form

$$\phi\left(\zeta_i\right) = -\frac{f'(\zeta_i)}{f(\zeta_i)} = \frac{\zeta_i}{\sigma^2} \tag{2.15}$$

and the function $\psi(\zeta_i)$ is clearly $\psi(\zeta_i) = 1/\sigma^2$, which implies that $\boldsymbol{\Psi} = \left(1/\sigma^2\right)\mathbf{I}$. The implicit likelihood equation then becomes

$$\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\right) = \mathbf{0} \tag{2.16}$$

and can be solved to yield the estimate

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{y} \tag{2.17}$$

which is the familiar least squares or minimum ℓ_2 norm solution.

In the correlated multivariate Gaussian case, the maximum likelihood estimate can be written

$$\hat{\boldsymbol{x}} = \left(\widetilde{\boldsymbol{H}}^T\widetilde{\boldsymbol{H}}\right)^{-1}\widetilde{\boldsymbol{H}}^T\boldsymbol{y} = \left(\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{y} \tag{2.18}$$

which is the standard weighted least squares solution.

2.2.5 Example: Maximum Likelihood Estimator for the Laplacian Distribution

The maximum likelihood estimator for the Laplacian distribution can be found by first writing the density function

$$f\left(\zeta_i\right) = \frac{1}{2b}\exp\left[-\frac{|\zeta_i|}{b}\right] \tag{2.19}$$

where b is a scale parameter, such that the variance of the distribution is $2b^2$. The function $\phi(\zeta_i)$ takes the form

$$\phi(\zeta_i) = -\frac{f'(\zeta_i)}{f(\zeta_i)} = \frac{\operatorname{sgn}\left(\zeta_i\right)}{b} \tag{2.20}$$

In the scalar estimation case, the implicit likelihood equation reduces to

$$\sum_{i=1}^{n} \operatorname{sgn}(\zeta_i) = 0 \tag{2.21}$$

which implies that the solution for $\hat{x}$ is such that half the residuals are negative and half the residuals are positive. This result is the sample median.

2.3 Generalized Maximum Likelihood Estimation

2.3.1 Overview of the Method

Huber in 1964[38] introduced a new class of estimation methods known as the generalized maximum likelihood method, also known as the technique of M-estimation. In this method, one seeks to minimize a function of the residuals of the form

$$J(\boldsymbol{x}) = \sum_{i=1}^{m} \rho(\zeta_i) \tag{2.22}$$

where ρ is an arbitrary function. Note that the generalized maximum likelihood method reduces the standard maximum likelihood method in the special case when $\rho(\zeta_i) = -\ln[f(\zeta_i)]$. The solution to the generalized maximum likelihood problem can be found from the implicit equation

$$\sum_{i=1}^{m} \phi(\zeta_i) \frac{\partial \zeta_i}{\partial \boldsymbol{x}} = \mathbf{0} \tag{2.23}$$

where $\phi(\zeta_i) = \rho'(\zeta_i)$. By defining the function $\psi(\zeta_i) = \phi(\zeta_i)/\zeta_i$, and the matrix $\boldsymbol{\Psi} = \operatorname{diag}[\psi(\zeta_i)]$, the implicit equation can be written in matrix form as

$$\boldsymbol{H}^T \boldsymbol{\Psi} (\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}) = \mathbf{0} \tag{2.24}$$

In generalized maximum likelihood estimation, the function ρ can be chosen to yield an estimator $\hat{\boldsymbol{x}}$ with certain desirable properties. A desirable property of the solution of the generalized maximum likelihood technique is that of robustness with respect to deviations from the assumed underlying probability distribution.

Huber[38] introduces a ρ function of the form

$$\rho(\zeta_i) = \begin{cases} \frac{1}{2}\zeta_i^2 & \text{for } |\zeta_i| < \gamma \\ \\ \gamma|\zeta_i| - \frac{1}{2}\gamma^2 & \text{for } |\zeta_i| \geq \gamma \end{cases} \tag{2.25}$$

where γ is a tuning parameter. This ρ function is a blend of the minimum ℓ_1 and ℓ_2 norm functions, and estimates derived from the use of this ρ function have desirable robustness properties. Specifically, the estimates minimize the maximum asymptotic estimation variance when applied to contaminated Gaussian densities. Note that as $\gamma \to 0$, Eq. (2.25) approaches the ℓ_1 norm, which is equivalent to the median in the scalar case, and that as $\gamma \to \infty$, Eq. (2.25) approaches the ℓ_2 norm, which is equivalent to the mean in the scalar case. The ρ, ϕ, and ψ functions for the ℓ_1, ℓ_2,

and Huber cases are shown in Fig. 2.2 for comparison. Note that the ρ, ϕ, and ψ functions are sometimes known as the score, influence, and weight functions, respectively. The analytic forms of the influence and weighting functions for the Huber estimator are provided as follows.

$$\phi(\zeta_i) = \begin{cases} 1 & \text{for } \zeta_i \geq \gamma \\ \zeta_i & \text{for } |\zeta_i| < \gamma \\ -1 & \text{for } \zeta_i \leq -\gamma \end{cases} \tag{2.26}$$

$$\psi(\zeta_i) = \begin{cases} 1 & \text{for } |\zeta_i| < \gamma \\ 1/|\zeta_i| & \text{for } |\zeta_i| \geq \gamma \end{cases} \tag{2.27}$$

Huber[38] shows that the ρ function given in Eq. (2.25) is asymptotically optimally robust in the ϵ neighborhood of the Gaussian distribution. Explicitly, if the measurement errors follow a perturbed Gaussian density function of the form

$$f(w) = \frac{1-\epsilon}{\sqrt{2\pi}} \exp\left(-\frac{w^2}{2}\right) + \epsilon g(w) \tag{2.28}$$

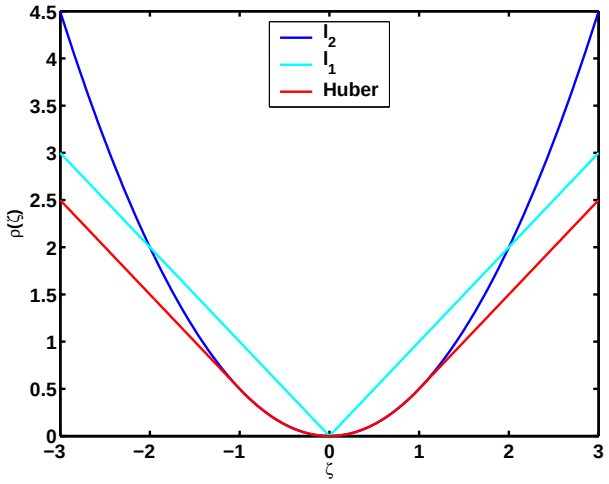
where ϵ is a perturbing parameter and $g(w)$ is an unknown perturbing density function, then the M-estimation technique with ρ function given by Eq. (2.25) minimizes the maximum asymptotic estimation variance in the ϵ neighborhood of the Gaussian distribution. Further, it can be shown that estimators of this form are asymptotically normal and unbiased.[107]

2.3.2 Robustness Concepts and Measures

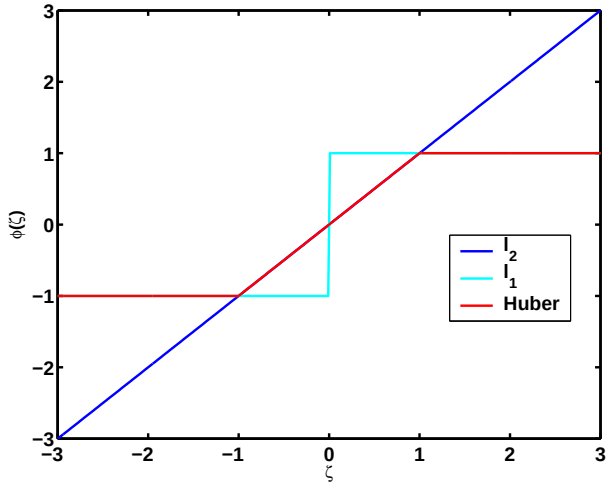
This section discusses some more formal definitions of robustness of an estimator with respect to deviations from some assumed underlying probability distribution. These concepts of robustness of an estimator to such deviations were developed by Hampel,[108,109] and are known as *qualitative robustness*, *global robustness*, and *local robustness*. Local robustness is most important for the purposes of this dissertation, however a short review of all concepts is provided in the following sections.

2.3.2.1 Qualitative Robustness

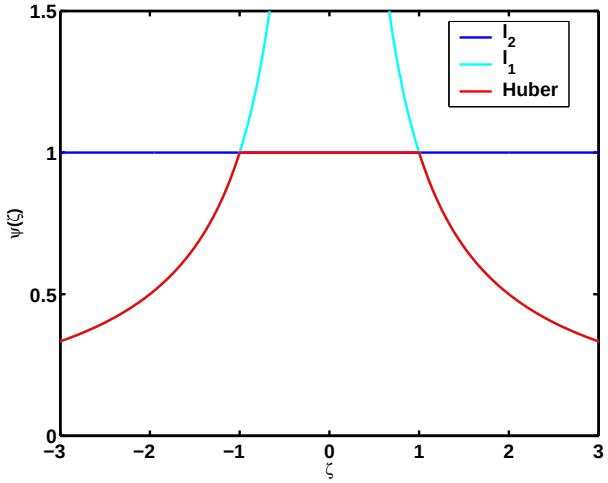
Qualitative robustness considers the impact to the estimator stemming from small deviations from the assumptions under which it was developed. A statistical procedure that is qualitatively robust implies that small changes from the assumed model produce small changes on the estimate, where small changes in the assumed model include cases of large changes to a small subset of the data (i.e. outliers) or small changes to all the data. A more precise mathematical definition is beyond the scope of this dissertation but can be found in Refs. 108 and 109. However, it is clear from discussion and examples presented in previous sections that the least–squares estimation technique is not qualitatively robust.



(a) Comparison of ρ Functions



(b) Comparison of ϕ Functions



(c) Comparison of ψ Functions

Figure 2.2: Comparison of ρ, ϕ, and ψ Functions

2.3.2.2 Global Robustness

The concept of global robustness relates to the level of perturbation that the estimator can handle before breaking down completely. Quantitative measures related to the concept of global robustness are known as the maximum bias curve and the breakdown point. For a contaminated distribution of the form $G = (1-\epsilon)\,F + \epsilon H$, where F is the assumed distribution and H is an unknown contaminating distribution, the maximum bias can be represented by $b_{max}(\epsilon) = \sup|\hat{x}(G) - \hat{x}(F)|$. The breakdown point describes the maximum value of ϵ for which the estimator has a finite maximum bias and is denoted by ϵ^*. An estimator is not globally robust if $\epsilon^* = 0$. Clearly the highest possible value of ϵ^* is $1/2$. Least–squares estimation is not globally robust since a single outlier can take the bias to infinity.

2.3.2.3 Local Robustness

The concept of local robustness concerns the effect of infinitesimal perturbations from the assumed model on the bias and variance. A robustness measure related to the concept of local robustness is known as the influence curve or influence function. The asymptotic influence function in a one-dimensional case can be written as[109]

$$\nu(\zeta) = \left.\frac{\partial \hat{x}(G)}{\partial \epsilon}\right|_{\epsilon=0} = \frac{\phi(\zeta)}{\mathrm{E}[\phi'(\zeta)]} \tag{2.29}$$

where $G = (1-\epsilon)F + \epsilon\Delta$, Δ is a unit mass distribution at the point x. Since $\mathrm{E}[\phi'(\zeta)]$ is a constant, the asymptotic influence function is proportional to $\phi(\zeta)$.

The asymptotic bias is locally related to the influence function via $b(\epsilon) = |\hat{x}(G) - \hat{x}(F)| \approx \epsilon|\nu(\zeta_i)|$. Therefore a locally robust estimator must have a bounded influence function for the bias to remain finite in the presence of contamination, which in turn implies that $\phi(\zeta_i)$ be bounded for all x. By this definition, the least–squares technique is not locally robust whereas the ℓ_1 and Huber techniques are, as can be gathered from Fig. 2.2.

The influence function for a linear regression estimator is somewhat more complicated and takes the form[109]

$$\nu_i(\zeta_i) = \frac{\phi(\zeta_i)}{\mathrm{E}[\phi'(\zeta_i)]}\mathrm{E}\left[\left(\frac{\partial \zeta_i}{\partial \boldsymbol{x}}\right)\left(\frac{\partial \zeta_i}{\partial \boldsymbol{x}}\right)^T\right]^{-1}\frac{\partial \zeta_i}{\partial \boldsymbol{x}} \tag{2.30}$$

In the linear regression case, it is important to note that the influence function can be bounded by an appropriate function ϕ, but can be unbounded depending on $(\partial\zeta_i/\partial\boldsymbol{x})$. Outlying $(\partial\zeta_i/\partial\boldsymbol{x})$ are known as *leverage points* and will be discussed in following sections. It is important to note that linear regression estimators with bounded ϕ functions are indeed locally robust in the absence of leverage points.

2.3.3 Example of a Gaussian Mixture

To illustrate the advantages of the generalized maximum likelihood techniques, consider for a moment the scalar estimation problem found by setting $\boldsymbol{H} = 1$. This problem corresponds with estimating the center of a probability distribution given a sample of measurements. In this example, the measurement data are drawn from a mixture of two Gaussian densities given by the function

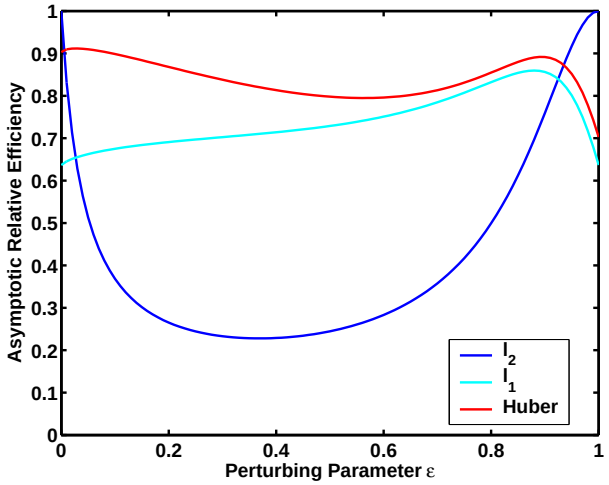
$$f(w) = \frac{1-\epsilon}{\sqrt{2\pi}}\exp\left(-\frac{w^2}{2}\right) + \frac{\epsilon}{\sqrt{2\pi}a}\exp\left(-\frac{w^2}{2a^2}\right) \tag{2.31}$$

where $a^2 > 1$ is the variance of the perturbing density. The mixture of Gaussian densities in Eq. 2.31 was first introduced by Newcomb for the purpose of studying heavy–tailed measurement data.[29,37] Since $\boldsymbol{H} = 1$, the estimation variance is a scalar quantity that simplifies from Eq. (2.7) as $\hat{P} = \mathrm{E}\left(\phi^2\right)/\left[\mathrm{E}\left(\phi'\right)\right]^2$. The ℓ_1, ℓ_2, and Huber estimation variances can be found by carrying out the the necessary integrals. If the asymptotic variance of the ℓ_1, ℓ_2, and Huber estimators are denoted by $\hat{P}_{\ell_1}$, $\hat{P}_{\ell_2}$, and $\hat{P}_H$, respectively, then the results are

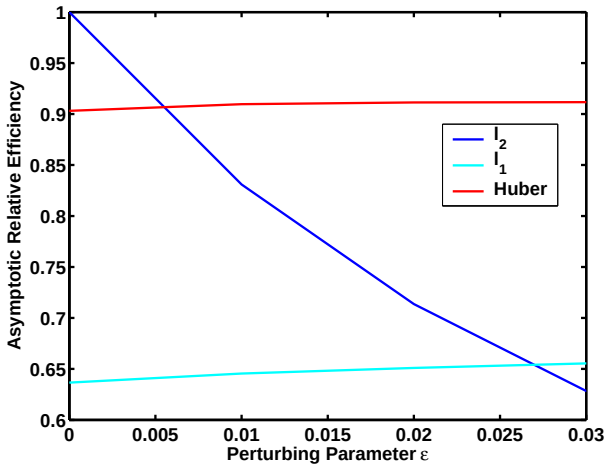
$$\hat{P}_{\ell_1} = \frac{\pi}{2}\left(1-\epsilon+\frac{\epsilon}{a}\right)^{-2} \tag{2.32}$$

$$\hat{P}_{\ell_2} = 1+\epsilon\left(a^2-1\right) \tag{2.33}$$

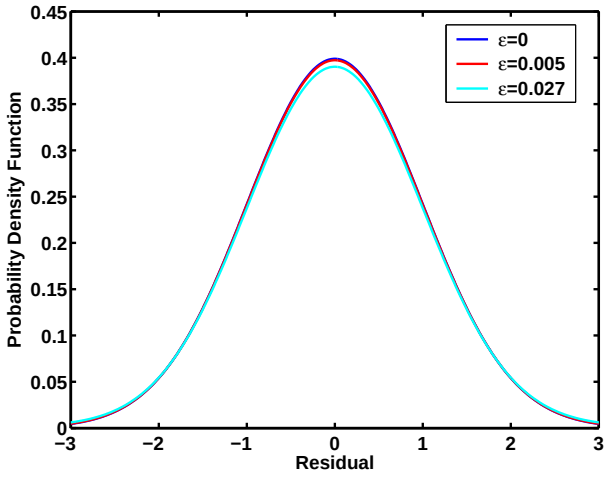
$$\begin{aligned}\hat{P}_H &= \left\{\gamma^2 + (1-\epsilon)\left[(1-\gamma^2)\,\mathrm{erf}\left(\frac{\gamma}{\sqrt{2}}\right) - \sqrt{\frac{2}{\pi}}\gamma\exp\left(\frac{-\gamma^2}{2}\right)\right]\right. \\ &\quad \left. +\epsilon\left[(a^2-\gamma^2)\,\mathrm{erf}\left(\frac{\gamma}{\sqrt{2}a}\right) - \sqrt{\frac{2}{\pi}}a\gamma\exp\left(\frac{-\gamma^2}{2a^2}\right)\right]\right\} \\ &\quad \times\left[(1-\epsilon)\,\mathrm{erf}\left(\frac{\gamma}{\sqrt{2}}\right) + \epsilon\mathrm{erf}\left(\frac{\gamma}{\sqrt{2}a}\right)\right]^{-2} \qquad (2.34)\end{aligned}$$



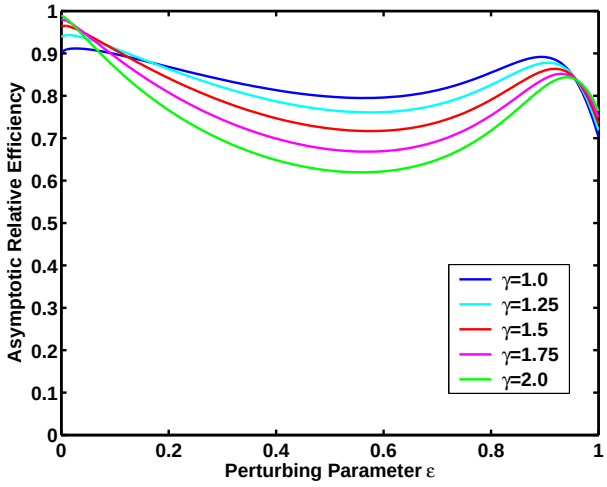
(a) Comparison of Asymptotic Relative Efficiencies



(b) Comparison of Asymptotic Relative Efficiencies (Detail)



(c) Comparison of Density Functions



(d) Comparison of Asymptotic Relative Efficiencies

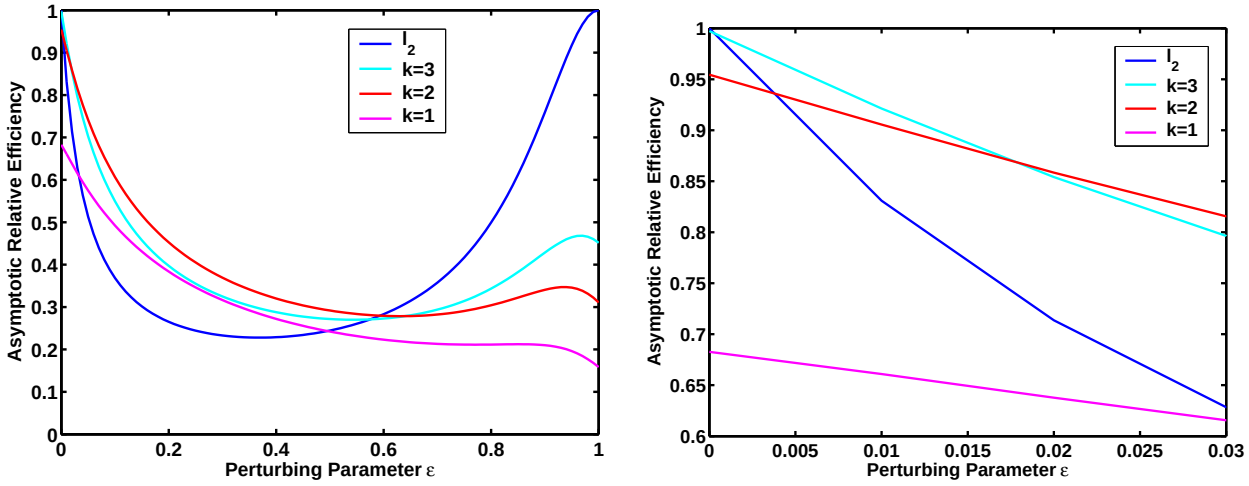
Figure 2.3: Comparison of Asymptotic Relative Efficiencies for Gaussian Mixture

The asymptotic relative efficiencies are shown in Fig. 2.3(a) as a function of the perturbing parameter ϵ for fixed $a = 5$ and $\gamma = 1$. The results for the entire range of ϵ is shown in Fig. 2.3(a) where just the portion of ϵ between 0 and 0.03 is shown in Fig. 2.3(b). From the plots in Fig. 2.3 it is apparent that when the data are perfectly Gaussian, the ℓ_2 estimator is more efficient than

either the ℓ_1 or the Huber estimator. This behavior is to be expected since the ℓ_2 estimator is the maximum likelihood estimator at the Gaussian distribution. When the perturbing parameter ϵ is increased, the ℓ_2 efficiency decreases sharply, but the ℓ_1 and Huber efficiencies show a more shallow decrease than the ℓ_2 case. The Huber and ℓ_1 estimators rapidly show higher efficiencies than the ℓ_2 case as ϵ increases away from zero. The ℓ_1 estimator becomes superior to the ℓ_2 estimator at $\epsilon = 0.027$ and the Huber estimator becomes superior to the ℓ_2 estimator at $\epsilon = 0.005$. As ϵ increases further, the distribution function again becomes nearly Gaussian for which the ℓ_2 estimator has the minimum variance. This example illustrates the point that a small contamination of an assumed Gaussian model by another distribution will lead to a rapid loss of efficiency of the least squares method, whereas the Huber estimator exhibits a reduced sensitivity. In fact, the Huber efficiency is nearly constant across a wide range of ϵ, illustrating the robustness of the technique.

Fig. 2.3(c) shows the density functions for the Gaussian mixture for $\epsilon = 0$ (the perfectly Gaussian case), for $\epsilon = 0.005$ and for $\epsilon = 0.027$. The density corresponding to $\epsilon = 0.005$ can hardly be distinguished from the perfectly Gaussian case. This plot serves to show that apparently extremely small deviations from Gaussianity can have a large impact on the efficiency of the least squares estimator.

The asymptotic relative efficiencies are shown in Fig. 2.3(d) as a function of the perturbing parameter ϵ for fixed $a = 5$ and γ ranging between values of 1.0 and 2.0 in increments of 0.25. These curves show how the asymptotic efficiency varies with the tuning parameter γ, in particular the un-contaminated efficiency ($\epsilon = 0$) and the behavior for moderate contamination levels. It is interesting to note that the results for small values of γ begin with smaller efficiencies but become superior for moderate values of ϵ.



(a) Comparison of Asymptotic Relative Efficiencies

(b) Comparison of Asymptotic Relative Efficiencies (Detail)

Figure 2.4: Comparison of Asymptotic Relative Efficiencies for Gaussian Mixture

The plots in Fig. 2.4 show the asymptotic relative efficiencies for the minimum ℓ_2 norm case and for three cases of data censoring. Recall from the previous section that the data censoring technique simply discards data points with large residuals, rather than processing them according to a statistical technique such as that introduced by Huber. The censoring technique corresponds

to a ψ function of the form $\psi(\zeta_i) = 1$, for $|\zeta_i| < k$ and zero otherwise. The plots in Fig. 2.4 show the resulting asymptotic relative efficiencies for the minimum ℓ_2 norm case and for the data censoring with $k = 1, 2$, and 3.

Several important points can be gleaned from the plots in Fig. 2.4 along with comparison to the results in Fig. 2.3. First, is that the data censoring technique with $k = 3$ maintains a high efficiency for small values of the perturbing parameter, and the censoring results for $k = 2$ and $k = 1$ are corresponding lower than as for $k = 3$. It is interesting to note that the efficiency for the censoring technique with $k = 2$ becomes superior to that for $k = 3$ for $\epsilon > 0.02$, and stays higher until well beyond $\epsilon = 0.5$. The efficiency for data censoring with $k = 1$ is always lower than the other cases. It should also be noted that the results for all three censoring cases become *lower* than that corresponding to the minimum ℓ_2 norm for $\epsilon > 0.6$. It can be seen by way of comparison to Fig. 2.3 that the data censoring results appear attractive only for very small contamination levels. The Huber processing technique provides results with a higher efficiency than the data censoring techniques for cases beyond this level.

2.3.4 Example of a Gaussian–Cauchy Mixture

As a slightly more extreme example, consider the Gaussian–Cauchy mixture given by the density

$$f(w) = \frac{1-\epsilon}{\sqrt{2\pi}} \exp\left(-\frac{w^2}{2}\right) + \frac{\epsilon}{\pi} \frac{1}{1+w^2} \tag{2.35}$$

Note that for any $\epsilon > 0$, the variance of this distribution is infinite. Therefore, the asymptotic variance of the ℓ_2 estimator applied to data drawn from this distribution is also infinite for $\epsilon > 0$. The ℓ_1 and Huber estimator asymptotic variances are finite and given by

$$\hat{P}_{\ell_1} = \left[\sqrt{\frac{2}{\pi}}\,(1-\epsilon) + \frac{2\epsilon}{\pi}\right]^{-2} \tag{2.36}$$

$$\begin{aligned}\hat{P}_H = {} & \left[\gamma^2 + \frac{2\gamma\epsilon}{\pi} + \gamma\sqrt{\frac{2}{\pi}}\,(\epsilon-1)\exp\left(\frac{-\gamma^2}{2}\right) - \frac{2\epsilon}{\pi}\left(1+\gamma^2\right)\arctan\gamma + \right. \\ & \left.\left(1-\gamma^2-\epsilon+\gamma^2\epsilon\right)\operatorname{erf}\left(\frac{\gamma}{\sqrt{2}}\right)\right] \cdot \left[\frac{2\epsilon}{\pi}\arctan\gamma + (1-\epsilon)\operatorname{erf}\left(\frac{\gamma}{\sqrt{2}}\right)\right]^{-2}\end{aligned} \tag{2.37}$$

The ℓ_1 and Huber asymptotic relative efficiencies are shown in Fig. 2.5. Here it is apparent that the Huber estimation technique performs quite well, even when applied to a problem where the random measurement errors have infinite variance.

2.3.5 Choice of Tuning Parameter

Figure 2.6(a) shows the influence of the choice of the tuning parameter γ on the asymptotic relative efficiency of the Huber technique for the Gaussian case ($\epsilon = 0$). The previous examples have shown that for a penalty in asymptotic relative efficiency at the model distribution (the Gaussian case), the relative efficiency is nearly constant in an ϵ neighborhood of the model distribution. The efficiency can vary between one for $\gamma = \infty$ (the ℓ_2 case) and $2/\pi$ for $\gamma = 0$ (the ℓ_1 case). If the value of the parameter ϵ is known (even though the density function $g(w)$ of the perturbing

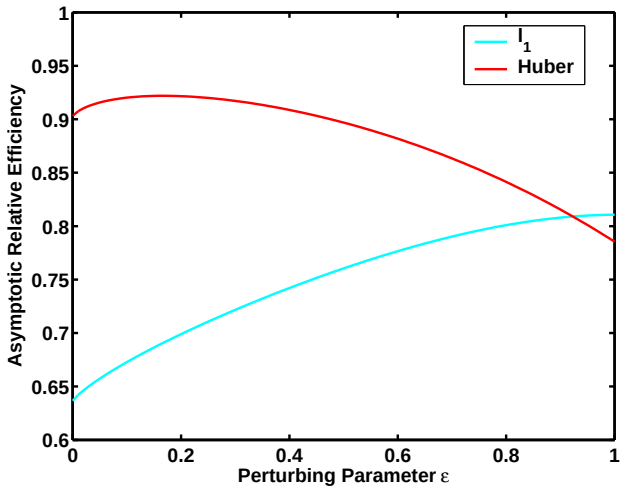
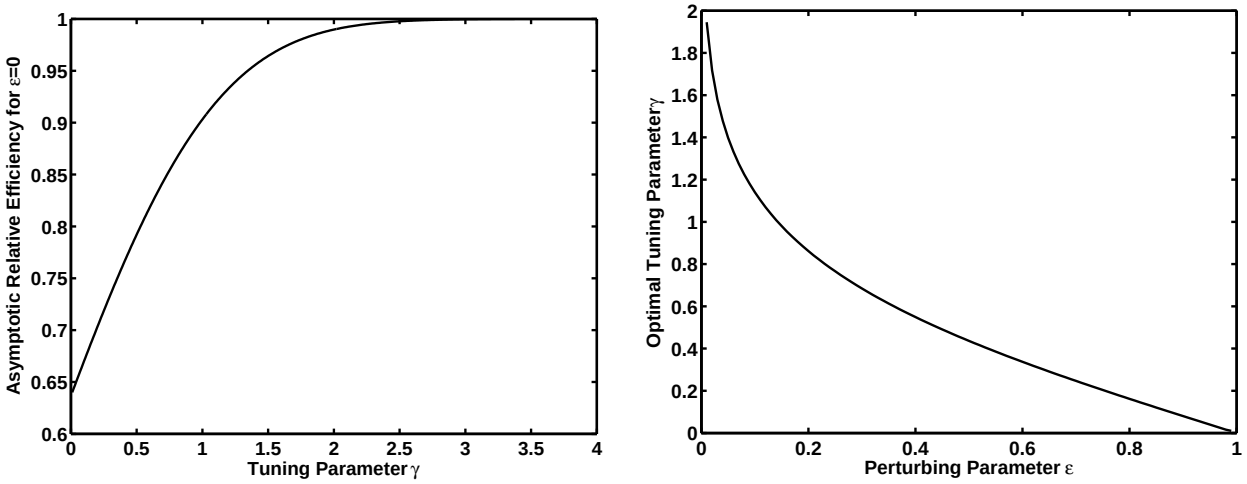


Figure 2.5: Comparison of Asymptotic Relative Efficiencies for Gaussian–Cauchy Mixture



(a) Huber Asymptotic Relative Efficiency for $\epsilon = 0$

(b) Optimal Tuning Parameter γ

Figure 2.6: Huber Gaussian Asymptotic Relative Efficiency and Optimal Tuning Parameter

distribution is unknown), then the best choice of γ has been determined by Huber.[30,38] If $\epsilon = \epsilon_0$ is the known perturbing parameter, then the optimal choice of γ, denoted by $\gamma^\star$, is given by the implicit equation[30,38]

$$\frac{1}{1-\epsilon_0} = \frac{1}{\gamma^\star}\sqrt{\frac{2}{\pi}}\exp\left[-\frac{(\gamma^\star)^2}{2}\right] + \operatorname{erf}\left(\frac{\gamma^\star}{\sqrt{2}}\right) \tag{2.38}$$

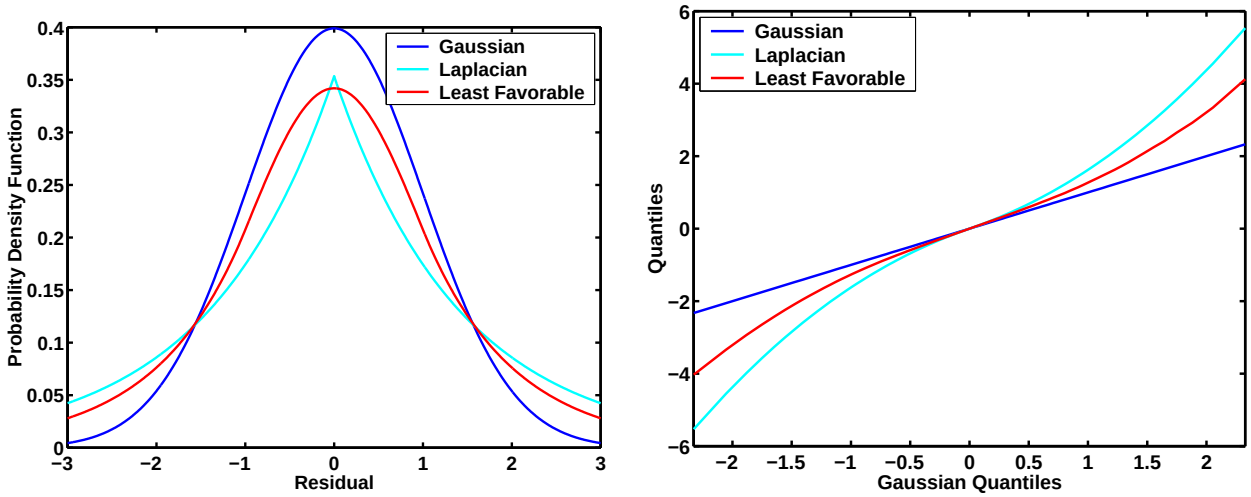
For a given value of ϵ, estimates computed using $\gamma = \gamma^\star$ maximize the asymptotic relative efficiency across the range of all possible values of γ.[30,38]

The optimal value of the tuning parameter $\gamma^\star$ is shown as a function of ϵ_0 in Fig. 2.6(b).

When $\epsilon_0 = 0$ the best tuning parameter is infinite. This value is expected because when $\epsilon_0 = 0$, the distribution is Gaussian and least squares estimation ($\gamma = \infty$) is minimum variance. As $\epsilon_0 \to 1$, the tuning parameter $\gamma^\star \to 0$. This behavior suggests that the median is the best choice of estimation technique when the noise density $f(w) = g(w)$ is completely unknown. However, note that if $\epsilon_0 > 1/2$ then the model distribution should no longer be taken as Gaussian since the perturbing density is in fact greater in proportion.

If the perturbing parameter ϵ is unknown, then the choice of γ is typically motivated by a desired variance at the model distribution. One common choice is $\gamma = 1.345$; for this value the Huber filter will exhibit estimation error variances that are 5% larger than that of the least squares method when the measurement error distributions are truly Gaussian. Generally any value of γ between 1.0 and 2.0 is suggested.[38]

2.3.6 The Huber Estimator as a Maximum Likelihood Estimator



(a) Density Functions

(b) Theoretical Quantile–Quantile Plot

Figure 2.7: Unit Variance Gaussian, Laplacian, and Least Favorable Distributions

This section poses the same question as did Gauss to the method of least squares: For what probability distribution (if any) is the Huber estimator a maximum likelihood estimator? This question can be answered by simply equating the Huber ρ function to the log-likelihood function, as $\rho(\zeta_i) = -\ln[f(\zeta_i)]$. This equation can be solved to yield

$$f(\zeta_i) = \left(\frac{1-\epsilon}{\sqrt{2\pi}}\right) \exp[-\rho(\zeta_i)] \tag{2.39}$$

where $\rho(\zeta_i)$ is as found in Eq. (2.25) with γ is determined from Eq. (2.38) for a given ϵ (or the opposite). The coefficient arises to ensure a unit total area under the density curve. This density function is known as the *least favorable density*, and is shown in Fig. 2.7(a) for $\gamma = 1$, $\epsilon = 0.1428$, along with the unit variance Gaussian and Laplacian densities. A comparison of the theoretical quantile–quantile plot is shown in Fig. 2.7(b). Clearly the least favorable density is something in between the Gaussian and the Laplacian, which is in keeping with the fact that the Huber estimator

is a blended minimum ℓ_2 and ℓ_1 norm technique, which are the maximum likelihood estimators for the Gaussian and the Laplacian distribution, respectively.

2.4 Leverage Points

Although the Huber generalized maximum likelihood technique has been proven to exhibit robustness with respect to deviations from an assumed Gaussian error model, an estimator designed using this approach may still lack robustness in the multiple linear regression case due to *leverage points*, as can be seen from the influence function of linear regression estimators shown in Eq. (2.30). Leverage points are outliers in the design matrix rather than outliers in the measurement errors. Outliers in the design matrix can artificially give extra weight to the associated measurements, whose consequences can be catastrophic in the case that an outlier in the measurement error statistics is combined with an outlier in the design matrix. This section discusses the design of estimators that are robust with respect to leverage points.

In this section, two leverage point identification methods are introduced. The first method is a classical technique that is based on the weighted Euclidean norm of the separation between a possible outlier and the sample mean, known as Mahalanobis Distances, introduced in Ref. 114. The weighting in this method is based on the sample covariance matrix. The second method, known as Projection Statistics, is a robust approach in which the sample mean and covariance are replaced by the sample median and the median absolute deviation, respectively. The latter method is said to be robust because it is insensitive to clusters of outliers, unlike the classical method.

2.4.1 The Mahalanobis Distances

Given a cloud of m points in n dimensions represented by the vectors $\boldsymbol{h}_i$ for $i = 1, \cdots, m$, the Mahalanobis distances are defined as

$$\mathcal{M}_i = \sqrt{\left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}\right)^T \boldsymbol{C}^{-1} \left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}\right)} \tag{2.40}$$

where $\bar{\boldsymbol{h}}$ is the sample mean and $\boldsymbol{C}$ is the sample covariance matrix, given by the equations

$$\bar{\boldsymbol{h}} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{h}_i \tag{2.41}$$

$$\boldsymbol{C} = \frac{1}{m-1} \sum_{i=1}^{m} \left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}\right) \left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}\right)^T \tag{2.42}$$

respectively.

The Mahalanobis distances can also be expressed as the solution to a maximization problem of the form[115]

$$\mathcal{M}_i = \max_{\|\boldsymbol{v}\|=1} \left[\frac{\left| \boldsymbol{h}_i^T \boldsymbol{v} - \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{h}_j^T \boldsymbol{v} \right|}{\sqrt{\frac{1}{m-1} \sum_{k=1}^{m} \left(\boldsymbol{h}_k^T \boldsymbol{v} - \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{h}_j^T \boldsymbol{v} \right)^2}} \right] \tag{2.43}$$

The Mahalanobis Distances represent the surface of an n-dimensional ellipsoid centered at the sample mean. The square of the Mahalanobis distances follow a χ^2 distribution with n degrees of

freedom, assuming that the input data are Gaussian. Therefore, an outlier identification method is to consider all points satisfying

$$\mathcal{M}_i > \sqrt{\chi^2_{n,\alpha}} \tag{2.44}$$

to be outliers, where α is the probability that a value falls inside the ellipse (for example, $\alpha = 0.95$).

While the Mahalanobis distances are simple to conceptualize and easy to compute, the method suffers from sensitivity to clusters of outliers. This sensitivity is due to the masking effect, in which it is possible to find groups of points with nonzero errors that can sum to produce very small residuals. The masking effect in the Mahalanobis distances is related to the sensitivity of the sample mean and covariance, which are not robust estimators. In the case of clustered outliers the sample mean can be pulled toward their direction and away from the main cluster of data, which serves also to increase the size of the sample covariance. In effect, these sensitivities serve to hide or mask the cluster of outliers since their associated Mahalanobis distances will not be larger than the main group of data. See Refs. 116 and 117 for further information on the masking effect and sensitivity of the Mahalanobis distances.

Another common leverage point identification scheme is based on the diagonal elements of the least-squares projection matrix. By substituting the least–squares estimate, $\hat{\boldsymbol{x}} = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{y}$ into Eq. (2.1), then the fitted values of $\boldsymbol{y}$ are given by $\hat{\boldsymbol{y}} = \boldsymbol{\Lambda}\boldsymbol{y}$, where

$$\boldsymbol{\Lambda} = \boldsymbol{H}\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T \tag{2.45}$$

is known as the projection matrix. It can be shown that the diagonal elements of the projection matrix are directly proportional to the square of the Mahalanobis distance.[117] The relationship of the Mahalanobis distances to the least–squares projection matrix underscores the lack of robustness of the Mahalanobis distances, since the least–squares method is itself non-robust. A robust alternative for leverage point identification will be discussed next.

2.4.2 Projection Statistics

A robust approach to the problem of outlier identification is to replace the sample mean and covariance in the equation for the Mahalanobis distances, Eq. 2.43, with the sample median and the median absolute deviation from the median.[115] These estimators of location and scale are known to be robust with respect to outliers and therefore one may expect that the computation of a Mahalanobis–like quantity based on these parameters will also be robust with respect to outliers. These quantities are known as *Projection Statistics*, and are defined as the solution to the maximization problem

$$\mathcal{P}_i = \max_{\|\boldsymbol{v}\|=1}\left[\frac{\left\|\boldsymbol{h}_i^T\boldsymbol{v} - \text{median}\left(\boldsymbol{h}_j^T\boldsymbol{v}\right)\right\|}{c\cdot\text{median}\left(\left\|\boldsymbol{h}_k^T\boldsymbol{v} - \text{median}\left(\boldsymbol{h}_j^T\boldsymbol{v}\right)\right\|\right)}\right] \tag{2.46}$$

where $c = 1.4826$ in the denominator is a correction factor to ensure unbiasedness.[24] The maximization problem can be approximated by considering only the directions that correspond to the unit vectors of the individual data points relative to the median of the point cloud. An algorithm for computing the approximate projection statistics is given in Ref. 118 and is repeated below.

Given a matrix $\boldsymbol{H} = \begin{bmatrix}\boldsymbol{h}_1 & \boldsymbol{h}_2 & \cdots & \boldsymbol{h}_m\end{bmatrix}$, the projection statistics can be computed by means of the following procedure.

1. Let the vector $\boldsymbol{m}$ represent the median of the column space of the matrix $\boldsymbol{H}$.
2. Compute the unit vectors of the individual data points relative to the median. These unit vectors are given by $\boldsymbol{v}_i = (\boldsymbol{h}_i - \boldsymbol{m}) / \|\boldsymbol{h}_i - \boldsymbol{m}\|$ for each $i = 1, \cdots, m$.
3. Compute the standardized projections $\boldsymbol{s}_i = \|\boldsymbol{z}_i - \text{median}\,(\boldsymbol{z}_j)\,\| / \text{mad}(\boldsymbol{z}_i)$, where $\boldsymbol{z}_i = \boldsymbol{h}^T \boldsymbol{v}_i$ and $\text{mad}(\boldsymbol{z}_i) = c \cdot \text{median}\,|\boldsymbol{z}_i - \text{median}\,(\boldsymbol{z}_j)|$.
4. The projection statistics are then found from $\mathcal{P}_j = \max_i (s_{ij})$, where s_{ij} represents the jth element of $\boldsymbol{s}_i$.

Having computed the projection statistics, one could potentially use them directly for multidimensional outlier detection. The projection statistics can also be used to compute *robust distances*, $\mathcal{R}_i$, which are defined as a weighted form of the Mahalanobis distances where the weights are a function of the projection statistics. The robust distances are defined as[116]

$$\mathcal{R}_i = \sqrt{\left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}_r\right)^T \mathbf{C_r}^{-1} \left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}_r\right)} \tag{2.47}$$

where $\bar{\boldsymbol{h}}_r$ and $\mathbf{C}_r$ are a robust mean and covariance matrix, calculated from

$$\bar{\boldsymbol{h}}_r = \left[\sum_{i=1}^{m} w_i \boldsymbol{h}_i\right] \cdot \left[\sum_{i=1}^{m} w_i\right]^{-1} \tag{2.48}$$

$$\mathbf{C}_r = \left[\sum_{i=1}^{m} w_i \left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}_r\right)\left(\boldsymbol{h}_i - \bar{\boldsymbol{h}}_r\right)^T\right] \cdot \left[\sum_{i=1}^{m} w_i\right]^{-1} \tag{2.49}$$

where w_i are weights computed from the projection statistics by means of

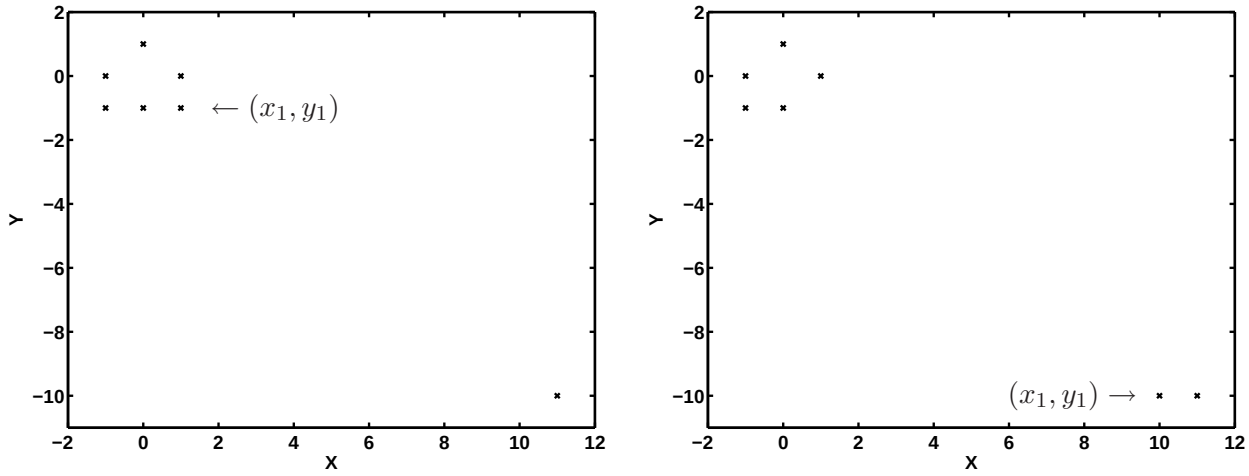
$$w_i = \min\left[\ 1, \quad \left(\frac{\chi^2_{n,\alpha}}{\mathcal{P}_i}\right)^2\ \right] \tag{2.50}$$

The robust distances calculated through the projection statistics algorithm grants an improvement in robustness properties without a great increase in computation, depending on the dimensionality of the system in question. In fact, the projection statistics method may be faster to compute than the classical Mahalanobis distances in high dimensions, since the former does not require the inversion of the sample covariance matrix. The following example case serves to illustrate the advantage of projection statistics in leverage point identification.

2.4.3 Simple Example of Leverage Point Identification

To illustrate the advantages of the projection statistics over the classical Mahalanobis distances, consider a simple example from power system state estimation.[119] In the following example, consider the matrix $\boldsymbol{H}$ given by

$$\boldsymbol{H} = \begin{bmatrix} x_1 & 1 & -1 & 0 & 0 & 11 & -1 \\ y_1 & 0 & 0 & -1 & 1 & -10 & -1 \end{bmatrix}^T \tag{2.51}$$



(a) Point Cloud for $x_1 = 1$, $y_1 = -1$

(b) Point Cloud for $x_1 = 10$, $y_1 = -10$

Figure 2.8: Elements of H Matrix

Table 2.1: Comparison of $\mathcal{M}_i$ and $\mathcal{P}_i$

(a) $x_1 = 1$, $y_1 = -1$

Point	$\mathcal{M}_i$	$\mathcal{P}_i$
1	0.26	0.67
2	1.21	0.67
3	0.76	1.35
4	0.71	0.67
5	1.40	1.35
6	2.25	13.49
7	1.54	1.35

(b) $x_1 = 10$, $y_1 = -10$

Point	$\mathcal{M}_i$	$\mathcal{P}_i$
1	1.47	7.16
2	1.30	0.67
3	0.84	0.75
4	0.75	0.67
5	1.45	1.35
6	1.56	7.57
7	1.53	1.35

The points in this matrix are shown in Fig. 2.8(a) for values $x_1 = 1$, $y_1 = -1$ and in Fig. 2.8(b) for values $x_1 = 10$, $y_1 = -10$. In the first case, one leverage point exists whereas two leverage points in a cluster appear in the second case. The Mahalanobis distances and the projection statistics have been computed for each case and are shown in Table 2.1. If the typical 0.95 χ^2 test is used to identify leverage points, then any value of $\mathcal{M}_i$ or $\mathcal{P}_i$ that exceeds $\left(\chi^2_{2,0.95}\right)^{1/2} \approx 2.45$ should be identified as a leverage point. In the example with $x_1 = 1$, $y_1 = -1$, point 6 is clearly identified as a leverage point the projection statistics, but the value of the Mahalanobis distance associated with this point does not exceed the threshold, although it is the largest value. In the case of $x_1 = 10$, $y_1 = -10$, the situation is worsened because of the clustered leverage points. Here, the projection statistics correctly identifies the leverage points but the Mahalanobis distances do not. In fact, the Mahalanobis distances associated with the two leverage points are not any larger than those associated with the other (non-leverage) points. This behavior is due to the masking effect of clustered leverage points.

2.5 Modified Generalized Maximum Likelihood Estimation

The method of generalized maximum likelihood estimation can be modified to provide robustness against leverage points by means of re-weighting the residuals by functions of the measure of the spread of the point cloud. Two methods are in common use for providing robustness against leverage points in linear regression, which are discussed in the following sections.

2.5.1 Mallows Form

The modified generalized maximum likelihood technique with leverage point identification is said to be in *Mallows form*[120, 121] if the cost function is written as

$$J(\boldsymbol{x}) = \sum_{i=1}^{m} w_i \rho(\zeta_i) \tag{2.52}$$

where w_i are the leverage point weights that are determined according to one of the previously developed leverage point identification methods.

The solution of the Mallows modified generalized maximum likelihood technique can be found by solving for the minimum of the cost function by means of the equation

$$\sum_{i=1}^{m} w_i \phi(\zeta_i) \frac{\partial \zeta_i}{\partial \boldsymbol{x}} = \boldsymbol{0} \tag{2.53}$$

where $\phi(\zeta_i) = \rho'(\zeta_i)$. By defining the function $\psi(\zeta_i) = w_i \phi(\zeta_i)/\zeta_i$, and the matrix $\boldsymbol{\Psi} = \text{diag}[\psi(\zeta_i)]$, the implicit equation can be written in matrix form as

$$\boldsymbol{H}^T \boldsymbol{\Psi} (\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}) = \boldsymbol{0} \tag{2.54}$$

which is in the same form as Eq. (2.24).

The influence function of a Mallows estimator is given by[109]

$$\nu_i(\zeta_i) = \frac{w_i \phi(\zeta_i)}{\text{E}[w_i \phi'(\zeta_i)]} \text{E}\left[\left(\frac{\partial \zeta_i}{\partial \boldsymbol{x}}\right)\left(\frac{\partial \zeta_i}{\partial \boldsymbol{x}}\right)^T\right]^{-1} \frac{\partial \zeta_i}{\partial \boldsymbol{x}} \tag{2.55}$$

The Mallows form of the modified generalized maximum likelihood estimation technique, advocated in Refs. 120–123, successfully reduces the weights of leverage points in the linear regression problem. However, this reduced weighting occurs without concern for the magnitude of the residual. Therefore, leverage points with a small residual receive the same reduced weighting as leverage points with a large residual. Generally, leverage points with small residual are termed *good* leverage points whereas those with large residual are termed *bad* leverage points. In the Mallows form, both good and bad leverage points receive the same weights, when, in fact, it is desirable that the good leverage points receive their full weight since they can be pivotal in determining the fit. The next technique avoids this problem using a different form of the modified generalized maximum likelihood estimation technique.

2.5.2 Schweppe Form

The modified generalized maximum likelihood technique with leverage point identification is said to be in *Schweppe form*[124] if the cost function is written as

$$J\left(\boldsymbol{x}\right) = \sum_{i=1}^{m} w_i^2 \rho\left(\frac{\zeta_i}{w_i}\right) \tag{2.56}$$

where w_i are the leverage point weights.

The solution of the Schweppe modified generalized maximum likelihood technique can be found by solving for the minimum of the cost function by means of the equation

$$\sum_{i=1}^{m} w_i \phi\left(\frac{\zeta_i}{w_i}\right) \frac{\partial \zeta_i}{\partial \mathbf{x}} = \mathbf{0} \tag{2.57}$$

where $\phi(\zeta_i) = \rho'(\zeta_i)$. By defining the function $\psi\left(\zeta_i/w_i\right) = w_i\phi(\zeta_i/w_i)/\zeta_i$, and the matrix $\boldsymbol{\Psi} = \operatorname{diag}\left[\psi\left(\zeta_i/w_i\right)\right]$, the implicit equation can be written in matrix form as

$$\boldsymbol{H}^T \boldsymbol{\Psi}\left(\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\right) = \mathbf{0} \tag{2.58}$$

which is in the same form as Eq. (2.24) and Eq. (2.54).

The influence function of a Schweppe estimator is given by[109]

$$\nu_i\left(\zeta_i\right) = \frac{w_i\phi\left(\zeta_i/w_i\right)}{\mathrm{E}\left[w_i^2\phi'\left(\zeta_i/w_i\right)\right]} \mathrm{E}\left[\left(\frac{\partial \zeta_i}{\partial \boldsymbol{x}}\right)\left(\frac{\partial \zeta_i}{\partial \boldsymbol{x}}\right)^T\right]^{-1} \frac{\partial \zeta_i}{\partial \boldsymbol{x}} \tag{2.59}$$

The Schweppe form of the modified generalized maximum likelihood estimation technique, advocated in Refs. 41, 119, 124–126, reduces the weight of leverage points, but only those with large residuals. If the ρ function is quadratic for small residuals, as is the case with the Huber ρ function, then the ϕ function is linear which in turn implies that the weighting function cancels out in the numerator and denominator of Eq. (2.57). Therefore the data point in question receives the full weight irrespective of its leverage.[119] This cancellation implies that Schweppe estimators only reject bad leverage points.

2.5.3 Weighting Function

In either the Mallows or Schweppe form of the modified generalized maximum likelihood technique, the weighting function for leverage point identification can be written using either the Mahalanobis distances or the projection statistics. In the case of the Mahalanobis distances, the weighting function takes the form

$$w_i = \min\left[\ 1, \quad \left(\chi^2_{n,\alpha}/\mathcal{M}_i^2\right)\ \right] \tag{2.60}$$

In the case of projection statistics, the weighting function is

$$w_i = \min\left[\ 1, \quad \left(\chi^2_{n,\alpha}/\mathcal{P}_i^2\right)\ \right] \tag{2.61}$$

It is worth noting that very few authors working on estimation problems have considered the use of the projection statistics for leverage point identification since the introduction in Ref. 115 in 1982. The projections statistics for leverage point identification in regression problems have appeared in Refs. 115–117, 119, 126, 127.

2.6 Numerical Solution of the Implicit Likelihood Equation

This section discusses a numerical technique of solving the implicit equation resulting from application of the generalized maximum likelihood method. Equation (2.24) can be expanded to yield $\boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{H}\boldsymbol{x} = \boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{y}$, which can be solved for $\boldsymbol{x}$ to give $\boldsymbol{x} = \left(\boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{y}$. Since the matrix $\boldsymbol{\Psi}$ depends on the residuals ζ_i, and hence on $\boldsymbol{x}$, an iterative solution to Eq. (2.24) is expressed as

$$\boldsymbol{x}^{(j+1)} = \left(\boldsymbol{H}^T\boldsymbol{\Psi}^{(j)}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{\Psi}^{(j)}\boldsymbol{y} \tag{2.62}$$

where the superscript (j) refers to the iteration index. The method can be initialized by using the least–squares solution $\boldsymbol{x}^{(0)} = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{y}$. The converged value from the iterative procedure is taken as the state estimate, $\hat{\boldsymbol{x}}$. This technique is known as iteratively reweighted least–squares,[110] and is generally attributed to Beaton and Tukey.[111] This iteration will converge if the ψ function is nonincreasing[25] (for $\zeta_i > 0$), which is the case when using the Huber ρ function in Eq. 2.25. The algorithm can be iterated until convergence or can be carried out through only one fixed iteration step, as discussed by Bickel,[112] and Rousseeuw and Leroy,[24] an approach that captures the robustness properties of the estimator and also saves on the computational costs associated with the iterative solution.

When the measurement errors are correlated, they can be transformed to uncorrelated measurements with unit variance by means of the transformations given in Sec. 2.2. The resulting solution by means of the iteratively reweighted algorithm takes the form

$$\boldsymbol{x}^{(j+1)} = \left(\boldsymbol{H}^T\boldsymbol{R}^{-1/2}\boldsymbol{\Psi}^{(j)}\boldsymbol{R}^{-1/2}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1/2}\boldsymbol{\Psi}^{(j)}\boldsymbol{R}^{-1/2}\boldsymbol{y} \tag{2.63}$$

Note that as $\gamma \to \infty$, the Huber filtering problem reduces to the least squares estimator. Specifically, when $\gamma \to \infty$, the matrix $\boldsymbol{\Psi} \to \mathbf{I}$, and Eq. (2.24) can be solved exactly in one iteration step and is equal to the least squares solution.

2.7 Examples in Linear Regression

2.7.1 A Simple Example Without Leverage Points

This section discusses the application of the robust Huber technique to a sample problem drawn from a simple linear regression involving fitting a line through data points. The Huber technique is used to estimate the slope and intercept of the line. Fig. 2.9 shows the data points, the least squares data fit, and four iterations of the Huber estimator for $\gamma = 1.345$.

In this case, the data points are shown with black circles, the least squares solution is shown with the blue line, and the first though fourth iterates of the Huber technique are shown with cyan, red, magenta, and green lines, respectively. In this problem, it is clear that the Huber technique effectively reduces the effect of the outlier, and also the convergence of the iteratively re-weighted least squares technique is clearly demonstrated. It should be noted that this example problem did not include leverage points, which is the subject of the next example problem.

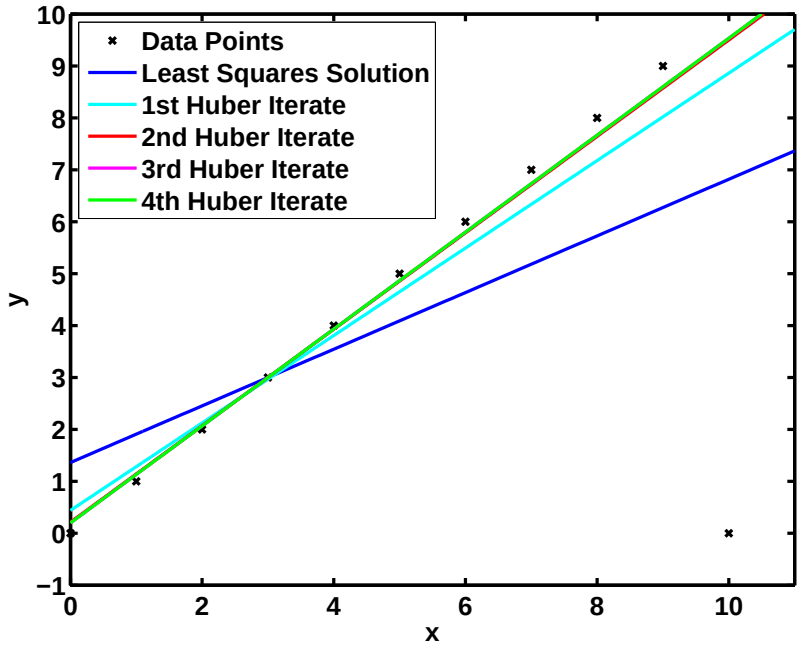


Figure 2.9: Data Points and Regression Results for Example Problem

Table 2.2: Linear Regression Solution

Iteration	Slope	Intercept
Least Squares	0.5455	1.3636
1st Huber Iterate	0.8420	0.4413
2nd Huber Iterate	0.9285	0.2144
3rd Huber Iterate	0.9331	0.2007
4th Huber Iterate	0.9333	0.2000

2.7.2 Example With a Single Bad Leverage Point

This section discusses a similar example problem, but now adds a single leverage point, which is simultaneously an outlier. The least squares, Huber estimator, Mallows estimator using the Mahalanobis distances for leverage point identification, Mallows estimator using the projection statistics for leverage point identification, the Schweppe estimator with both Mahalanobis distances and projection statistics for leverage point identification are implemented for this problem. The results of the estimation problem are shown in Fig. 2.10. In this case, the data points are shown with black circles, the least squares solution is shown with the blue line, the Mallows–Mahalanobis distance solution is shown with the cyan line, the Schweppe–Mahalanobis distance solution is shown with the dashed cyan line, the Mallows–Projection statistics solution is shown with the magenta line, and the Schweppe–Projection statistics solution is shown with the dashed magenta line. The slope, intercept, and iteration count is shown in Table 2.3.

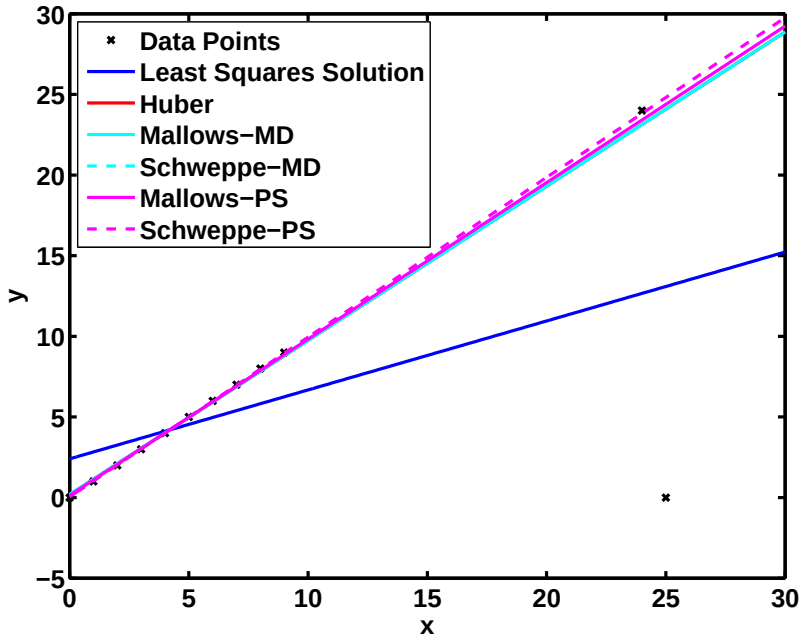


Figure 2.10: Data Points and Regression Results for Example Problem With Single Bad Leverage Point

Table 2.3: Linear Regression Solution With Single Bad Leverage Point

Method	Slope	Intercept	Iterations
Least Squares	-0.0029	4.1096	0
Huber technique	0.7515	1.0182	28
Mallows–Mahalanobis distances	0.8005	0.8175	15
Schweppe–Mahalanobis distance	0.8398	0.6563	12
Mallows–Projection statistics	0.9262	0.3023	8
Schweppe–Projection statistics	0.9781	0.0897	6

From these results it is apparent that the Schweppe form of the modified generalized maximum likelihood technique is superior to the Mallows form, and also that the use of projection statistics is superior to that of the Mahalanobis distances for the weighting functions used in leverage point identification. Interestingly, the number of iterations required of the estimator reduces as the robustness of the estimator increases, using the same convergence tolerance on each particular case (explicitly, the change in slope and intercept are both less than 10^{-5}). The next section discusses a similar problem with multiple leverage points.

2.7.3 Examples With Multiple Leverage Points

This section discusses two example problems involving multiple leverage points in the data for the linear regression problem. The first example involves data with both a good and bad leverage point,

Table 2.4: Linear Regression Solution With One Good and One Bad Leverage Point

Method	Slope	Intercept	Iterations
Least Squares	0.4275	2.4011	0
Huber technique	0.9563	0.1834	10
Mallows–Mahalanobis distances	0.9563	0.1834	10
Schweppe–Mahalanobis distance	0.9563	0.1834	10
Mallows–Projection statistics	0.9705	0.1219	6
Schweppe–Projection statistics	0.9908	0.0381	5

whereas the second example involves data with two bad leverage points.

2.7.3.1 Example with Good and Bad Leverage Points

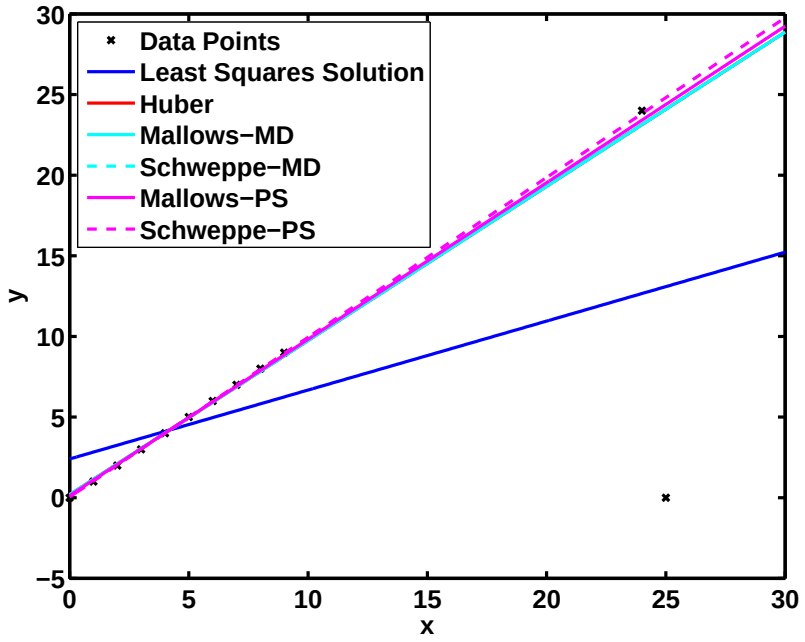


Figure 2.11: Data Points and Regression Results for Example Problem With Good and Bad Leverage Points

The results of a sample problem with one good and one bad leverage point are shown in Fig. 2.11. The slope, intercept, and iteration count for this problem is shown in Table 2.4, for the same group of estimators investigated in the previous example involving one leverage point. These results illustrate that the estimators based on the Mahalanobis distances for leverage point identification falls prey to the clustered leverage points in this problem, even though one point is good and the other is bad, while the estimators based on the projection statistics can successfully handle the clustered leverage points. Clearly, the results using the Huber estimator alone without

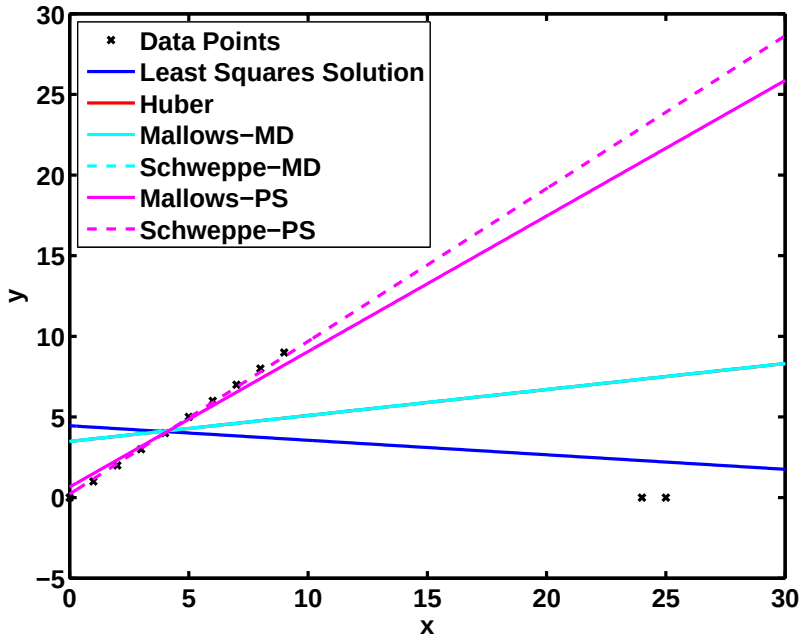


Figure 2.12: Data Points and Regression Results for Example Problem With Two Bad Leverage Points

Table 2.5: Linear Regression Solution With Two Bad Leverage Points

Method	Slope	Intercept	Iterations
Least Squares	-0.0900	4.4553	0
Huber technique	0.1610	3.4811	12
Mallows–Mahalanobis distances	0.1610	3.4811	12
Schweppe–Mahalanobis distance	0.1610	3.4811	12
Mallows–Projection statistics	0.8404	0.6523	13
Schweppe–Projection statistics	0.9473	0.2153	7

any leverage point identification are identical to those using the Mahalanobis distances in both the Mallows and Schweppe form, indicated that the Mahalanobis distance–based leverage point identification method failed to successfully pick out the leverage points in this problem. This problem also illustrates that the performance of the Mallows estimator is degraded compared with that of the Schweppe estimator, since the latter can distinguish between good and bad leverage points.

2.7.3.2 Example with Two Bad Leverage Points

The results of a sample problem with two bad leverage points are shown in Fig. 2.12. The slope, intercept, and iteration count for this problem is shown in Table 2.5. These results illustrate that the estimators based on the Mahalanobis distances for leverage point identification falls prey

to the clustered leverage points in this problem, whereas the estimators based on the projection statistics can successfully handle the clustered leverage points. As in the previous example, the results using the Huber estimator alone without any leverage point identification are identical to those using the Mahalanobis distances in both the Mallows and Schweppe form, indicating that the Mahalanobis distance–based leverage point identification method failed to successfully pick out the leverage points in this problem. This problem also illustrates that the performance of the Mallows estimator is degraded compared with that of the Schweppe estimator even for cases with multiple bad leverage points.

Chapter 3

Dynamic State Estimation

3.1 Introduction

This chapter discusses the problem of estimating the state of the system of ordinary differential equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, t\right) \tag{3.1}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ are the deterministic inputs to the system, and $\boldsymbol{v}$ are random inputs to the system. The mean value of $\boldsymbol{v}$ is $\bar{\boldsymbol{v}} = \mathbf{0}$ and its covariance is $\mathbf{Q}$. It is assumed that the state of the system can be measured at discrete times in the form of a model given as

$$\boldsymbol{y}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) + \boldsymbol{w}_k \tag{3.2}$$

where the subscript k refers to the value of the parameter at time t_k, $\boldsymbol{y}_k$ is the measurement at time t_k, and $\boldsymbol{w}_k$ is the measurement noise at time t_k. The mean value of $\boldsymbol{w}_k$ is $\bar{\boldsymbol{w}}_k = \mathbf{0}$ and its covariance is $\boldsymbol{R}_k$.

3.2 Robust Extended Kalman Filter

The Kalman filter is a well-known technique for estimating the state of systems of differential equations described in the form provided in Eqs. (3.1–3.2). The filter is a predictor–corrector approach in which the state predictions are computed by numerical integration of the dynamic model, and state corrections are obtained by a weighted linear combination of the predicted measurements and the actual measurements. In this approach, the state and covariance predictions are given as

$$\bar{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k-1} + \int_{t_{k-1}}^{t_k} \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, \bar{\boldsymbol{v}}, t\right) \mathrm{d}t \tag{3.3}$$

$$\bar{\boldsymbol{P}}_k = \hat{\boldsymbol{P}}_{k-1} + \int_{t_{k-1}}^{t_k} \left[\mathbf{A}(\hat{\boldsymbol{x}}(t), t)\bar{\boldsymbol{P}}(t) + \bar{\boldsymbol{P}}(t)^T\mathbf{A}(\hat{\boldsymbol{x}}(t), t)^T + \mathbf{B}(\hat{\boldsymbol{x}}(t), t)\mathbf{Q}(t)\mathbf{B}(\hat{\boldsymbol{x}}(t), t)^T\right] \mathrm{d}t \tag{3.4}$$

where $\bar{\boldsymbol{x}}_k$ is the predicted value of the state at time t_k, based on the estimated value of the state at time t_{k-1}, which is $\hat{\boldsymbol{x}}_{k-1}$. Similarly, $\bar{\boldsymbol{P}}_k$ is the predicted state error covariance matrix at time t_k

and $\hat{\boldsymbol{P}}_{k-1}$ is the estimated state error covariance matrix and time t_{k-1}. Also, the matrices $\mathbf{A}$ and $\mathbf{B}$ are given by

$$\mathbf{A}(t) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}(t),\boldsymbol{v}=\bar{\boldsymbol{v}}} \tag{3.5}$$

$$\mathbf{B}(t) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}(t),\boldsymbol{v}=\bar{\boldsymbol{v}}} \tag{3.6}$$

The discrete–time covariance propagation can also be written as

$$\bar{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k \hat{\boldsymbol{P}}_k \boldsymbol{\Phi}_k^T + \tilde{\boldsymbol{Q}}_k \tag{3.7}$$

where $\boldsymbol{\Phi}_k$ is the state transition matrix and $\tilde{\boldsymbol{Q}}_k$ is the process noise covariance matrix. Both of these quantities can be determined jointly through the relation[130,141]

$$\exp\left(\begin{bmatrix} -\boldsymbol{A} & \boldsymbol{B}\boldsymbol{Q}\boldsymbol{B}^T \\ \mathbf{0} & \boldsymbol{A}^T \end{bmatrix} \delta t\right) = \begin{bmatrix} \boldsymbol{X}_{11} & \boldsymbol{X}_{12} \\ \mathbf{0} & \boldsymbol{X}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_{11} & \boldsymbol{\Phi}_k^{-1}\tilde{\boldsymbol{Q}}_k \\ \mathbf{0} & \boldsymbol{\Phi}_k^T \end{bmatrix} \tag{3.8}$$

leading to the result $\boldsymbol{\Phi}_k = \boldsymbol{X}_{22}^T$ and $\tilde{\boldsymbol{Q}}_k = \boldsymbol{\Phi}_k \boldsymbol{X}_{12}$, where $\delta t = t_{k+1} - t_k$. Note that these relationships and the use of the state transition matrix are approximations that are true only when the state dynamics matrix is constant. These approximations generally work well when the process is slowly varying and/or the sampling rate is sufficiently high for a particular problem.

The state correction obtained at the time of the measurement update can be expressed as a linear regression problem between the predicted state and the observed quantity.[128] If the true value of the state is written as $\boldsymbol{x}_k$ and the state prediction error is written as $\boldsymbol{\delta}_k = \boldsymbol{x}_k - \bar{\boldsymbol{x}}_k$, then the state prediction can be expressed as $\bar{\boldsymbol{x}}_k = \boldsymbol{x}_k - \boldsymbol{\delta}_k$. By approximating the measurement equation as

$$\boldsymbol{y}_k \approx \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right) + \boldsymbol{H}_k\left(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\right) \tag{3.9}$$

the regression problem then takes the form

$$\begin{Bmatrix} \boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right) + \boldsymbol{H}_k\bar{\boldsymbol{x}}_k \\ \bar{\boldsymbol{x}}_k \end{Bmatrix} = \begin{bmatrix} \boldsymbol{H}_k \\ \mathbf{I} \end{bmatrix} \boldsymbol{x}_k + \begin{Bmatrix} \boldsymbol{w}_k \\ -\boldsymbol{\delta}_k \end{Bmatrix} \tag{3.10}$$

where the matrix $\boldsymbol{H}_k$ is defined as

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}_k,\boldsymbol{w}=\bar{\boldsymbol{w}}} \tag{3.11}$$

By defining the quantities

$$\boldsymbol{T}_k = \begin{bmatrix} \boldsymbol{R}_k & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{P}}_k \end{bmatrix} \tag{3.12}$$

$$\boldsymbol{z}_k = \boldsymbol{T}_k^{-1/2} \begin{Bmatrix} \boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right) + \boldsymbol{H}_k\bar{\boldsymbol{x}}_k \\ \bar{\boldsymbol{x}}_k \end{Bmatrix} \tag{3.13}$$

$$\boldsymbol{G}_k = \boldsymbol{T}_k^{-1/2} \begin{bmatrix} \boldsymbol{H}_k \\ \mathbf{I} \end{bmatrix} \tag{3.14}$$

$$\boldsymbol{\xi}_k = \boldsymbol{T}_k^{-1/2} \begin{Bmatrix} \boldsymbol{w}_k \\ -\boldsymbol{\delta}_k \end{Bmatrix} \tag{3.15}$$

the linear regression problem is transformed to

$$\boldsymbol{z}_k = \boldsymbol{G}_k \boldsymbol{x}_k + \boldsymbol{\xi}_k \tag{3.16}$$

Note that Eq. (3.16) is in precisely the same form as the linear regression problem given in Eq.(2.11), which can be solved using the robust Huber generalized maximum–likelihood technique. Recall from the previous section that the solution of the Huber estimation problem can be written as the converged solution of the iterative equation

$$\boldsymbol{x}_k^{(j+1)} = \left(\boldsymbol{G}_k^T \boldsymbol{\Psi}^{(j)} \boldsymbol{G}_k \right)^{-1} \boldsymbol{G}_k^T \boldsymbol{\Psi}^{(j)} \boldsymbol{z}_k \tag{3.17}$$

where the superscript (j) refers to the iteration index. The method can be initialized by using the least–squares (Kalman filter) solution $\boldsymbol{x}_k^{(0)} = \left(\boldsymbol{G}_k^T \boldsymbol{G}_k \right)^{-1} \boldsymbol{G}_k^T \boldsymbol{z}_k$. The converged value from the iterative procedure is taken as the corrected state estimate following a measurement update, $\hat{\boldsymbol{x}}_k$.

Finally, the estimated state estimate error covariance matrix following the measurement update is computed from

$$\hat{\boldsymbol{P}}_k = \left(\boldsymbol{G}_k^T \boldsymbol{\Psi} \boldsymbol{G}_k \right)^{-1} \tag{3.18}$$

using the final value of $\boldsymbol{\Psi}$ corresponding to the converged state estimate.

Due to the particular structure of the matrix $\boldsymbol{G}_k$, the discrete time dynamic state estimation technique can be simplified considerably from the static state estimation technique by application of the matrix inversion lemma.[129] It is first useful to decompose the $\boldsymbol{\Psi}$ matrix into two portions $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ corresponding to the state prediction and measurement prediction residuals so that

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_y & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_x \end{bmatrix} \tag{3.19}$$

The state update equation can be rewritten as

$$\hat{\boldsymbol{x}}_k = \left(\boldsymbol{G}_k^T \boldsymbol{\Psi} \boldsymbol{G}_k \right)^{-1} \boldsymbol{G}_k^T \boldsymbol{\Psi} \boldsymbol{z}_k = \left(\widetilde{\boldsymbol{G}}_k^T \widetilde{\boldsymbol{G}}_k \right)^{-1} \widetilde{\boldsymbol{G}}_k \widetilde{\boldsymbol{z}}_k \tag{3.20}$$

where $\widetilde{\boldsymbol{G}}_k = \boldsymbol{\Psi}^{1/2} \boldsymbol{G}_k$ and $\widetilde{\boldsymbol{z}}_k = \boldsymbol{\Psi}^{1/2} \boldsymbol{z}_k$. It follows from Eq. 3.14 that

$$\begin{aligned} \widetilde{\boldsymbol{G}}_k &= \boldsymbol{\Psi}^{1/2} \boldsymbol{T}_k^{-1/2} \begin{bmatrix} \boldsymbol{H}_k \\ \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\Psi}_y^{1/2} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_x^{1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_k^{-1/2} & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{P}}_k^{-1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_k \\ \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\Psi}_y^{1/2} \boldsymbol{R}_k^{-1/2} \boldsymbol{H}_k \\ \boldsymbol{\Psi}_x^{1/2} \bar{\boldsymbol{P}}_k^{-1/2} \end{bmatrix} \end{aligned} \tag{3.21}$$

Then,

$$\begin{aligned} \widetilde{\boldsymbol{G}}_k^T \widetilde{\boldsymbol{G}}_k &= \begin{bmatrix} \boldsymbol{H}_k^T \left(\boldsymbol{R}_k^{-1/2} \right)^T \boldsymbol{\Psi}_y^{1/2} & \left(\bar{\boldsymbol{P}}_k^{-1/2} \right)^T \boldsymbol{\Psi}_x^{1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_y^{1/2} \boldsymbol{R}_k^{-1/2} \boldsymbol{H}_k \\ \boldsymbol{\Psi}_x^{1/2} \bar{\boldsymbol{P}}_k^{-1/2} \end{bmatrix} \\ &= \boldsymbol{H}_k^T \left(\boldsymbol{R}_k^{-1/2} \right)^T \boldsymbol{\Psi}_y \boldsymbol{R}_k^{-1/2} \boldsymbol{H}_k + \left(\bar{\boldsymbol{P}}_k^{-1/2} \right)^T \boldsymbol{\Psi}_x \bar{\boldsymbol{P}}_k^{-1/2} \\ &= \boldsymbol{H}_k^T \widetilde{\boldsymbol{R}}_k^{-1} \boldsymbol{H}_k + \widetilde{\bar{\boldsymbol{P}}}_k^{-1} \end{aligned} \tag{3.22}$$

where $\widetilde{\boldsymbol{R}}_k = \boldsymbol{R}_k^{1/2}\boldsymbol{\Psi}_y^{-1}\left(\boldsymbol{R}_k^{1/2}\right)^T$ and $\widetilde{\bar{\boldsymbol{P}}}_k = \bar{\boldsymbol{P}}_k^{1/2}\boldsymbol{\Psi}_x^{-1}\left(\bar{\boldsymbol{P}}_k^{1/2}\right)^T$. Since $\hat{\boldsymbol{P}}_k^{-1} = \widetilde{\boldsymbol{G}}_k^T\widetilde{\boldsymbol{G}}_k$, then the covariance expansion reduces to

$$\hat{\boldsymbol{P}}_k = \left(\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1}\boldsymbol{H}_k + \widetilde{\bar{\boldsymbol{P}}}_k^{-1}\right)^{-1} \tag{3.23}$$

Making use of the matrix inversion lemma, Eq. (3.23) can be reduced to*

$$\begin{aligned}
\hat{\boldsymbol{P}}_k &= \widetilde{\bar{\boldsymbol{P}}}_k - \widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T + \widetilde{\boldsymbol{R}}_k\right)^{-1}\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k \\
&= \left[\boldsymbol{I} - \widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T + \widetilde{\boldsymbol{R}}_k\right)^{-1}\boldsymbol{H}_k\right]\widetilde{\bar{\boldsymbol{P}}}_k \\
&= \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\widetilde{\bar{\boldsymbol{P}}}_k
\end{aligned} \tag{3.24}$$

where $\boldsymbol{K}_k = \widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T + \widetilde{\boldsymbol{R}}_k\right)^{-1}$ is the reweighted Kalman gain matrix.

Next, the state update equation can be manipulated to form

$$\begin{aligned}
\hat{\boldsymbol{x}}_k &= \left(\widetilde{\boldsymbol{G}}_k^T\widetilde{\boldsymbol{G}}_k\right)^{-1}\widetilde{\boldsymbol{G}}_k\widetilde{\boldsymbol{z}}_k \\
&= \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\widetilde{\bar{\boldsymbol{P}}}_k\left[\begin{array}{cc} \boldsymbol{H}_k^T\left(\widetilde{\boldsymbol{R}}_k^{-1/2}\right)^T & \left(\widetilde{\bar{\boldsymbol{P}}}_k^{-1/2}\right)^T \end{array}\right] \\
&\quad\times\left[\begin{array}{cc} \widetilde{\boldsymbol{R}}_k^{-1/2} & \boldsymbol{0} \\ \boldsymbol{0} & \widetilde{\bar{\boldsymbol{P}}}_k^{-1/2} \end{array}\right]\left\{\begin{array}{c} \boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right) + \boldsymbol{H}_k\bar{\boldsymbol{x}}_k \\ \bar{\boldsymbol{x}}_k \end{array}\right\} \\
&= \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\widetilde{\bar{\boldsymbol{P}}}_k\left\{\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1}\left[\boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right) + \boldsymbol{H}_k\bar{\boldsymbol{x}}_k\right] + \widetilde{\bar{\boldsymbol{P}}}_k^{-1}\bar{\boldsymbol{x}}_k\right\} \\
&= \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1}\left[\boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right) + \boldsymbol{H}_k\bar{\boldsymbol{x}}_k\right] + \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\bar{\boldsymbol{x}}_k
\end{aligned} \tag{3.25}$$

The term $\left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1}$ can be further simplified as

$$\begin{aligned}
\left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1} &= \widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1} - \boldsymbol{K}_k\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1} \\
&= \boldsymbol{K}_k\left(\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T + \widetilde{\boldsymbol{R}}_k\right)\widetilde{\boldsymbol{R}}_k^{-1} - \boldsymbol{K}_k\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1} \\
&= \boldsymbol{K}_k\left[\left(\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T + \widetilde{\boldsymbol{R}}_k\right)\widetilde{\boldsymbol{R}}_k^{-1} - \boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1}\right] \\
&= \boldsymbol{K}_k\left[\boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1} + \boldsymbol{I} - \boldsymbol{H}_k\widetilde{\bar{\boldsymbol{P}}}_k\boldsymbol{H}_k^T\widetilde{\boldsymbol{R}}_k^{-1}\right] \\
&= \boldsymbol{K}_k
\end{aligned} \tag{3.26}$$

Substituting this result into Eq. (3.25) yields

$$\begin{aligned}
\hat{\boldsymbol{x}}_k &= \boldsymbol{K}_k\left[\boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right) + \boldsymbol{H}_k\bar{\boldsymbol{x}}_k\right] + \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right]\bar{\boldsymbol{x}}_k \\
&= \boldsymbol{K}_k\left[\boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right)\right] + \boldsymbol{K}_k\boldsymbol{H}_k\bar{\boldsymbol{x}}_k + \bar{\boldsymbol{x}}_k - \boldsymbol{K}_k\boldsymbol{H}_k\bar{\boldsymbol{x}}_k \\
&= \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k\left[\boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right)\right]
\end{aligned} \tag{3.27}$$

*Recall that the matrix inversion lemma provides the relation $\left(\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{D}}\right)^{-1} = \boldsymbol{\mathcal{A}}^{-1} - \boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}}\left(\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{\mathcal{B}} + \boldsymbol{\mathcal{C}}^{-1}\right)^{-1}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{A}}^{-1}$ where $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{C}}$ are invertible and all matrices have suitable dimensions. To apply the lemma to Eq. (3.23), let $\boldsymbol{\mathcal{A}} = \widetilde{\bar{\boldsymbol{P}}}_k^{-1}$, $\boldsymbol{\mathcal{B}} = \boldsymbol{H}_k^T$, $\boldsymbol{\mathcal{C}} = \widetilde{\boldsymbol{R}}_k^{-1}$, and $\boldsymbol{\mathcal{D}} = \boldsymbol{H}_k$.

Substituting the definitions $\widetilde{\boldsymbol{R}}_k = \boldsymbol{R}_k^{1/2}\boldsymbol{\Psi}_y^{-1}\left(\boldsymbol{R}_k^{1/2}\right)^T$ and $\widetilde{\bar{\boldsymbol{P}}}_k = \bar{\boldsymbol{P}}_k^{1/2}\boldsymbol{\Psi}_x^{-1}\left(\bar{\boldsymbol{P}}_k^{1/2}\right)^T$ back into equations for the state and covariance updates results in the iteratively reweighted Kalman filter algorithm given by

$$\hat{\boldsymbol{x}}_k^{(j+1)} = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k^{(j)}\left[\boldsymbol{y}_k - \boldsymbol{h}\left(\bar{\boldsymbol{x}}_k\right)\right] \tag{3.28}$$

$$\hat{\boldsymbol{P}}_k = \left(\mathbf{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right)\bar{\boldsymbol{P}}_k^{1/2}\boldsymbol{\Psi}_x^{-1}\bar{\boldsymbol{P}}_k^{1/2} \tag{3.29}$$

where

$$\boldsymbol{K}_k = \bar{\boldsymbol{P}}_k^{1/2}\boldsymbol{\Psi}_x^{-1}\bar{\boldsymbol{P}}_k^{1/2}\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\bar{\boldsymbol{P}}_k^{1/2}\boldsymbol{\Psi}_x^{-1}\bar{\boldsymbol{P}}_k^{1/2}\boldsymbol{H}_k^T + \boldsymbol{R}_k^{1/2}\boldsymbol{\Psi}_y^{-1}\boldsymbol{R}_k^{1/2}\right)^{-1} \tag{3.30}$$

Note that as the tuning parameter $\gamma \to \infty$, the matrix $\boldsymbol{\Psi} \to \mathbf{I}$ and the Huber recursive estimation technique reduces to the familiar Kalman filter solution.

3.3 Robust Divided Difference Filtering

The Divided Difference Filter is one of several new estimation techniques that are collectively known as Sigma–Point Kalman Filters (SPKF). Like the basic Kalman filter, the SPKFs seek to determine a state estimate that minimizes the ℓ_2–norm of the residuals. The SPKF technique differs from the basic Kalman filter in that the estimate and covariance propagation prior to determining the state estimate at the time of the measurement update. The divided difference filter arises from an alternate approach to the nonlinear state estimation and filtering problem than the EKF. Whereas the EKF is based on first–order Taylor series approximations, the divided difference filter relies on multidimensional interpolation formulas to approximate the nonlinear transformations. As a result of this approach, the filter does not require knowledge or existence of the partial derivatives of the system dynamics and measurement equations.

The First–Order (DD1) and Second–Order (DD2) Divided Difference Filters[79,80] are generalizations of the filter introduced by Schei,[81] and are two examples of SPKF–class estimators; other examples can be found in Refs. 82–84. This section describes the DD1 and DD2 filters and shows how the measurement update equation can be modified to provide robustness against deviations from the assumed Gaussian error model by use of the Huber technique. The filter summary closely follows that given in Refs. 79 and 80. The filter equations rely upon a discrete representation of the system dynamics, in which the differential equation in Eq. (3.1) is replaced with a difference equation of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\left(\boldsymbol{x}_k, \boldsymbol{v}_k, t_k\right) \tag{3.31}$$

and the measurement equation is given by

$$\boldsymbol{y}_k = \boldsymbol{G}\left(\boldsymbol{x}_k, \boldsymbol{w}_k, t_k\right) \tag{3.32}$$

where $\boldsymbol{y}_k$ is the measurement at time t_k, and $\boldsymbol{w}_k$ is the measurement noise at time t_k. As in Sec. 3.2, the mean value of $\boldsymbol{w}_k$ is $\bar{\boldsymbol{w}}_k = \mathbf{0}$ and its covariance is the matrix $\boldsymbol{R}_k$.

The following square-root decompositions of the predicted state covariance, $\bar{\boldsymbol{P}}_k$, corrected state covariance, $\hat{\boldsymbol{P}}_k$, process noise covariance, $\mathbf{Q}_k$, and measurement noise covariance, $\boldsymbol{R}_k$, are defined as

$$\hat{\boldsymbol{P}}_k = \hat{\boldsymbol{S}}_{x_k}\hat{\boldsymbol{S}}_{x_k}^T \tag{3.33}$$

$$\bar{\boldsymbol{P}}_k = \bar{\boldsymbol{S}}_{x_k}\bar{\boldsymbol{S}}_{x_k}^T \tag{3.34}$$
$$\mathbf{Q}_k = \boldsymbol{S}_{v_k}\boldsymbol{S}_{v_k}^T \tag{3.35}$$
$$\boldsymbol{R}_k = \boldsymbol{S}_{w_k}\boldsymbol{S}_{w_k}^T \tag{3.36}$$

Also, the jth column of $\bar{\boldsymbol{s}}_{x_k}$ is referred to as $\bar{\boldsymbol{s}}_{x_{k_j}}$; likewise for the other matrices.

3.3.1 Overview of the DD1 Filter

The DD1 filter makes use of first–order divided differences to approximate the system and measurement dynamics rather than the first–order Taylor series expansions used in the EKF. The following matrices of first–order divided differences are defined as

$$\boldsymbol{S}'_{x\hat{x}_{k_{i,j}}} = \frac{1}{2c}\left[\boldsymbol{F}_i\left(\hat{\boldsymbol{x}} + c\hat{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{v}}_k, t_k\right) - \boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k - c\hat{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{v}}_k, t_k\right)\right] \tag{3.37}$$
$$\boldsymbol{S}'_{xv_{k_{i,j}}} = \frac{1}{2c}\left[\boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k + c\boldsymbol{s}_{v_j}, t_k\right) - \boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k - c\boldsymbol{s}_{v_j}, t_k\right)\right] \tag{3.38}$$
$$\boldsymbol{S}'_{y\bar{x}_{k_{i,j}}} = \frac{1}{2c}\left[\boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k + c\bar{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{w}}_k, t_k\right) - \boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k - c\bar{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{w}}_k, t_k\right)\right] \tag{3.39}$$
$$\boldsymbol{S}'_{yw_{k_{i,j}}} = \frac{1}{2c}\left[\boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{w}}_k + c\boldsymbol{s}_{w_j}, t_k\right) - \boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{w}}_k - c\boldsymbol{s}_{w_j}, t_k\right)\right] \tag{3.40}$$

where c the divided–difference perturbing parameter.

The state, state root–covariance, measurement, and measurement root-covariance predictions are given by

$$\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{F}\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k, t_k\right) \tag{3.41}$$
$$\bar{\boldsymbol{S}}_{x_{k+1}} = \mathcal{H}\left(\begin{bmatrix} \boldsymbol{S}'_{x\hat{x}_k} & \boldsymbol{S}'_{xv_k} \end{bmatrix}\right) \tag{3.42}$$
$$\bar{\boldsymbol{y}}_k = \boldsymbol{G}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{w}}_k, t_k\right) \tag{3.43}$$
$$\boldsymbol{S}_{y_k} = \mathcal{H}\left(\begin{bmatrix} \boldsymbol{S}'_{y\bar{x}_k} & \boldsymbol{S}'_{yw_k} \end{bmatrix}\right) \tag{3.44}$$

where $\mathcal{H}(\cdot)$ represents a Householder transformation of the argument matrix.[79,80]

The state and root-covariance measurement update equations are given by

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_k\right) \tag{3.45}$$
$$\hat{\boldsymbol{S}}_{x_k} = \mathcal{H}\left(\begin{bmatrix} \bar{\boldsymbol{S}}_{x_k} - \boldsymbol{K}_k\boldsymbol{S}'_{yx_k} & \boldsymbol{K}_k\boldsymbol{S}'_{yw_k} \end{bmatrix}\right) \tag{3.46}$$

where $\boldsymbol{K}_k = \bar{\boldsymbol{S}}_{x_k}\boldsymbol{S}'^T_{y\bar{x}_k}\left(\boldsymbol{S}_{y_k}\boldsymbol{S}_{y_k}^T\right)^{-1}$ is the Kalman gain matrix.

The form of the state root-covariance update can be derived by first defining $\boldsymbol{P}_{y_k} = \boldsymbol{S}_{y_k}\boldsymbol{S}_{y_k}^T = \boldsymbol{S}'_{y\bar{x}_k}\boldsymbol{S}'^T_{y\bar{x}_k} + \boldsymbol{S}'_{yw_k}\boldsymbol{S}'^T_{yw_k}$, and $\boldsymbol{P}_{\bar{x}y_k} = \bar{\boldsymbol{S}}_{x_k}\boldsymbol{S}'^T_{y\bar{x}_k}$. Then, it is clear that the gain matrix can be written as $\boldsymbol{K}_k = \boldsymbol{P}_{\bar{x}y_k}\boldsymbol{P}_{y_k}^{-1}$. The state error covariance matrix update can be computed from $\hat{\boldsymbol{P}}_k = \bar{\boldsymbol{P}}_k - \boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}_k^T$, but in the DD1 implementation it is desirable to update the root-covariance matrix directly. By noting the identities,

$$\boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}_k^T = \boldsymbol{P}_{\bar{x}y_k}\boldsymbol{P}_{y_k}^{-1}\boldsymbol{P}_{y_k}\boldsymbol{K}_k^T = \bar{\boldsymbol{S}}_{x_k}\boldsymbol{S}'^T_{yx_k}\boldsymbol{K}_k^T \tag{3.47}$$
$$= \boldsymbol{K}_k\boldsymbol{P}_{y_k}^{-1}\boldsymbol{P}_{y_k}\boldsymbol{P}_{\bar{x}y_k}^T = \boldsymbol{K}_k\boldsymbol{S}'_{yx_k}\bar{\boldsymbol{S}}_{x_k}^T \tag{3.48}$$
$$= \boldsymbol{K}_k\boldsymbol{S}_{y_k}\boldsymbol{S}_{y_k}^T\boldsymbol{K}_k^T = \boldsymbol{K}_k\boldsymbol{S}'_{y\bar{x}_k}\boldsymbol{S}'^T_{y\bar{x}_k}\boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{S}'_{yw_k}\boldsymbol{S}'^T_{yw_k}\boldsymbol{K}_k^T \tag{3.49}$$

the state covariance update can be manipulated to form

$$\begin{aligned}\hat{\boldsymbol{P}}_k &= \bar{\boldsymbol{P}}_k - \boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}^T - \boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}^T + \boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}^T \\ &= \bar{\boldsymbol{S}}_{x_k}\bar{\boldsymbol{S}}_{x_k}^T - \bar{\boldsymbol{S}}_{x_k}\boldsymbol{S}_{yx_k}'^T\boldsymbol{K}_k^T - \boldsymbol{K}_k\boldsymbol{S}_{yx_k}'\bar{\boldsymbol{S}}_{x_k}^T + \boldsymbol{K}_k\boldsymbol{S}_{y\bar{x}_k}'\boldsymbol{S}_{y\bar{x}_k}'^T\boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{S}_{yw_k}'\boldsymbol{S}_{yw_k}'^T\boldsymbol{K}_k^T \\ &= \left(\bar{\boldsymbol{S}}_{x_k} - \boldsymbol{K}_k\boldsymbol{S}_{yx_k}'\right)\left(\bar{\boldsymbol{S}}_{x_k} - \boldsymbol{K}_k\boldsymbol{S}_{yx_k}'\right)^T + \left(\boldsymbol{K}_k\boldsymbol{S}_{yw_k}'\right)\left(\boldsymbol{K}_k\boldsymbol{S}_{yw_k}'\right)^T \end{aligned} \tag{3.50}$$

Then, the state covariance matrix $\hat{\boldsymbol{P}}_k$ can clearly be factored into $\hat{\boldsymbol{P}}_k = \hat{\boldsymbol{S}}_{x_k}\hat{\boldsymbol{S}}_{x_k}^T$, where $\hat{\boldsymbol{S}}_{x_k}$ takes the form given in Eq. (3.46).

3.3.2 Overview of the DD2 Filter

The DD2 filter makes use of second–order divided differences to approximate nonlinear transformation of the state and covariance. The matrices of second–order divided differences are defined as

$$\begin{aligned}\boldsymbol{S}''_{x\hat{x}_{k_{i,j}}} &= \frac{\sqrt{c^2-1}}{2c^2}\left[\boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k + c\hat{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{v}}_k, t_k\right) + \boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k - c\hat{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{v}}_k, t_k\right)\right. \\ &\quad \left. -2\boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k, t_k\right)\right] \end{aligned} \tag{3.51}$$

$$\begin{aligned}\boldsymbol{S}''_{xv_{k_{i,j}}} &= \frac{\sqrt{c^2-1}}{2c^2}\left[\boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k + c\boldsymbol{s}_{w_j}, t_k\right) + \boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k - c\boldsymbol{s}_{w_j}, t_k\right)\right. \\ &\quad \left. -2\boldsymbol{F}_i\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k, t_k\right)\right] \end{aligned} \tag{3.52}$$

$$\begin{aligned}\boldsymbol{S}''_{y\bar{x}_{k_{i,j}}} &= \frac{\sqrt{c^2-1}}{2c^2}\left[\boldsymbol{G}_i\left(\bar{\boldsymbol{x}} + c\bar{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{w}}_k, t_k\right) + \boldsymbol{G}_i\left(\bar{\boldsymbol{x}} - c\bar{\boldsymbol{s}}_{x_j}, \bar{\boldsymbol{w}}_k, t_k\right)\right. \\ &\quad \left. -2\boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{w}}_k, t_k\right)\right] \end{aligned} \tag{3.53}$$

$$\begin{aligned}\boldsymbol{S}''_{yw_{k_{i,j}}} &= \frac{\sqrt{c^2-1}}{2c^2}\left[\boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{w}}_k + c\boldsymbol{s}_{w_j}, t_k\right) + \boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k\bar{\boldsymbol{w}}_k - c\boldsymbol{s}_{w_j}, t_k\right)\right. \\ &\quad \left. -2\boldsymbol{G}_i\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{w}}_k, t_k\right)\right] \end{aligned} \tag{3.54}$$

The state, state root–covariance, measurement, and measurement covariance predictions are given by

$$\begin{aligned}\bar{\boldsymbol{x}}_{k+1} &= \left(\frac{c^2 - n_x - n_v}{c^2}\right)\boldsymbol{F}\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k, t_k\right) \\ &\quad + \frac{1}{2c^2}\sum_{j=1}^{n_x}\left[\boldsymbol{F}\left(\hat{\boldsymbol{x}}_k + c\hat{\boldsymbol{s}}_{x_{k_j}}, \bar{\boldsymbol{v}}_k, t_k\right) + \boldsymbol{F}\left(\hat{\boldsymbol{x}}_k - c\hat{\boldsymbol{s}}_{x_{k_j}}, \bar{\boldsymbol{v}}_k, t_k\right)\right] \\ &\quad + \frac{1}{2c^2}\sum_{j=1}^{n_v}\left[\boldsymbol{F}\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k + c\boldsymbol{s}_{v_{k_j}}, t_k\right) + \boldsymbol{F}\left(\hat{\boldsymbol{x}}_k, \bar{\boldsymbol{v}}_k - c\boldsymbol{s}_{v_{k_j}}, t_k\right)\right] \end{aligned} \tag{3.55}$$

$$\bar{\boldsymbol{S}}_{x_{k+1}} = \mathcal{H}\left(\begin{bmatrix} \boldsymbol{S}'_{x\hat{x}_k} & \boldsymbol{S}'_{xv_k} & \boldsymbol{S}''_{x\hat{x}_k} & \boldsymbol{S}''_{xv_k} \end{bmatrix}\right) \tag{3.56}$$

$$\begin{aligned}\bar{\boldsymbol{y}}_k &= \left(\frac{c^2 - n_x - n_w}{c^2}\right)\boldsymbol{G}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{w}}_k, t_k\right) \\ &\quad + \frac{1}{2c^2}\sum_{j=1}^{n_x}\left[\boldsymbol{G}\left(\bar{\boldsymbol{x}}_k + c\bar{\boldsymbol{s}}_{x_{k_j}}, \bar{\boldsymbol{w}}_k, t_k\right) + \boldsymbol{G}\left(\bar{\boldsymbol{x}}_k - c\bar{\boldsymbol{s}}_{x_{k_j}}, \bar{\boldsymbol{w}}_k, t_k\right)\right] \end{aligned}$$

$$+\frac{1}{2c^2}\sum_{j=1}^{n_w}\left[\boldsymbol{G}\left(\bar{\boldsymbol{x}}_k,\bar{\boldsymbol{w}}_k+c\boldsymbol{s}_{w_{k_j}},t_k\right)+\boldsymbol{G}\left(\bar{\boldsymbol{x}}_k,\bar{\boldsymbol{w}}_k-c\boldsymbol{s}_{w_{k_j}},t_k\right)\right] \tag{3.57}$$

$$\boldsymbol{S}_{y_k} = \mathcal{H}\left(\left[\begin{array}{cccc}\boldsymbol{S}'_{y\bar{x}_k} & \boldsymbol{S}'_{yw_k} & \boldsymbol{S}''_{y\bar{x}_k} & \boldsymbol{S}''_{yw_k}\end{array}\right]\right) \tag{3.58}$$

where n_x is the size of the state dimension, n_v is the size of the process noise dimension, and n_w is the size of the measurement noise dimension.

Lastly, the state and root-covariance update equations are given by

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k+\boldsymbol{K}_k\left(\boldsymbol{y}_k-\bar{\boldsymbol{y}}_k\right) \tag{3.59}$$

$$\hat{\boldsymbol{S}}_{x_k} = \mathcal{H}\left(\left[\begin{array}{cccc}\bar{\boldsymbol{S}}_{x_k}-\boldsymbol{K}_k\boldsymbol{S}'_{yx_k} & \boldsymbol{K}_k\boldsymbol{S}'_{yw_k} & \boldsymbol{K}_k\boldsymbol{S}'_{yx_k} & \boldsymbol{K}_k\boldsymbol{S}'_{yw_k}\end{array}\right]\right) \tag{3.60}$$

where $\boldsymbol{K}_k=\bar{\boldsymbol{S}}_{x_k}\boldsymbol{S}'^T_{y\bar{x}_k}\left(\boldsymbol{S}_{y_k}\boldsymbol{S}_{y_k}^T\right)^{-1}$ is the Kalman gain matrix.

Note that many of the same state and noise perturbations used to calculate the first–order divided differences are again used to compute the second–order divided differences. This point has important implications with regard to the computational costs, suggesting that the DD2 filter may not require a great deal more computing time than the DD1 filter.

3.3.3 Modification of Measurement Update Equations Using Huber's Technique

This section discusses how the measurement update equations of the DD1 and DD2 filters can be solved using the Huber method. To apply this method, it is first required to recast the measurement update as a regression problem between the observed quantity and the state prediction. Following Sec. 3.2, the state prediction error can be written as $\boldsymbol{\delta}_k=\boldsymbol{x}_k-\bar{\boldsymbol{x}}_k$, then the state prediction can be expressed as $\bar{\boldsymbol{x}}_k=\boldsymbol{x}_k-\boldsymbol{\delta}_k$. By defining the cross–covariance matrix[79] $\boldsymbol{P}_{xy_k}=\bar{\boldsymbol{S}}_{x_k}\left(\boldsymbol{S}'_{y\bar{x}_k}\right)^T$ and the matrix $\boldsymbol{H}_k=\boldsymbol{P}_{xy_k}^T\bar{\boldsymbol{P}}_k^{-1}=\boldsymbol{S}'_{y\bar{x}_k}\left(\bar{\boldsymbol{S}}_{x_k}^T\right)^{-1}$, then the measurement update can then be written in the form of a linear regression problem in same manner as shown in Sec. 3.2.[134]

If the measurement update given in Eq. (3.45) is taken as the initial guess for the state estimate, then a one-step Huber update can be written as

$$\boldsymbol{S}_{y_k}^{(1)} = \mathcal{H}\left(\left[\begin{array}{cc}\boldsymbol{S}'_{y\bar{x}_k}\boldsymbol{\Psi}_x^{-1/2} & \boldsymbol{S}'_{yw_k}\boldsymbol{\Psi}_y^{-1/2}\end{array}\right]\right) \tag{3.61}$$

$$\boldsymbol{K}_k^{(1)} = \bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1}\boldsymbol{S}'^T_{y\bar{x}_k}\left(\boldsymbol{S}_{y_k}^{(1)}\boldsymbol{S}_{y_k}^{(1)^T}\right)^{-1} \tag{3.62}$$

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k+\boldsymbol{K}_k^{(1)}\left(\boldsymbol{y}_k-\bar{\boldsymbol{y}}_k\right) \tag{3.63}$$

$$\hat{\boldsymbol{S}}_{x_k} = \mathcal{H}\left(\left[\begin{array}{cc}\bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1/2}-\boldsymbol{K}_k^{(1)}\boldsymbol{S}'_{yx_k}\boldsymbol{\Psi}_x^{-1/2} & \boldsymbol{K}_k^{(1)}\boldsymbol{S}'_{yw_k}\boldsymbol{\Psi}_y^{-1/2}\end{array}\right]\right) \tag{3.64}$$

The form of the robust DD1 measurement update given in Eqs. (3.61–3.64) can be found from following a similar procedure for the non-robust case. In the Huber case, note that $\boldsymbol{P}_{y_k}=\boldsymbol{S}_{y_k}\boldsymbol{S}_{y_k}^T=\boldsymbol{S}'_{y\bar{x}_k}\boldsymbol{\Psi}_x^{-1}\boldsymbol{S}'^T_{y\bar{x}_k}+\boldsymbol{S}_{yw_k}\boldsymbol{\Psi}_y^{-1}\boldsymbol{S}_{yw_k}^T$. It then follows that $\boldsymbol{P}_{y_k}$ can be factored to give the result in Eq. (3.61). The identities given in Eqs. (3.47–3.49) can be rewritten as

$$\boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}_k^T = \boldsymbol{P}_{\bar{x}y_k}\boldsymbol{P}_{y_k}^{-1}\boldsymbol{P}_{y_k}\boldsymbol{K}_k^T=\bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1}\boldsymbol{S}'^T_{yx_k}\boldsymbol{K}_k^T \tag{3.65}$$

$$= \boldsymbol{K}_k\boldsymbol{P}_{y_k}^{-1}\boldsymbol{P}_{y_k}\boldsymbol{P}_{\bar{x}y_k}^T=\boldsymbol{K}_k\boldsymbol{S}'_{yx_k}\boldsymbol{\Psi}_x^{-1}\bar{\boldsymbol{S}}_{x_k}^T \tag{3.66}$$

$$= \boldsymbol{K}_k\boldsymbol{S}_{y_k}\boldsymbol{S}_{y_k}^T\boldsymbol{K}_k^T=\boldsymbol{K}_k\boldsymbol{S}'_{y\bar{x}_k}\boldsymbol{\Psi}_x^{-1}\boldsymbol{S}'^T_{y\bar{x}_k}\boldsymbol{K}_k^T+\boldsymbol{K}_k\boldsymbol{S}'_{yw_k}\boldsymbol{\Psi}_y^{-1}\boldsymbol{S}'^T_{yw_k}\boldsymbol{K}_k^T \tag{3.67}$$

Using these results, the state covariance update equation can be found by manipulating the equation

$$\begin{aligned}
\hat{\boldsymbol{P}}_k &= \bar{\boldsymbol{P}}_k - \boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}^T - \boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}^T + \boldsymbol{K}_k\boldsymbol{P}_{y_k}\boldsymbol{K}^T \\
&= \bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1}\bar{\boldsymbol{S}}_{x_k}^T - \bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1}\boldsymbol{S}_{yx_k}'^T\boldsymbol{K}_k^T - \boldsymbol{K}_k\boldsymbol{S}_{yx_k}'\boldsymbol{\Psi}_x^{-1}\bar{\boldsymbol{S}}_{x_k}^T \\
&\quad + \boldsymbol{K}_k\boldsymbol{S}_{y\bar{x}_k}'\boldsymbol{\Psi}_x^{-1}\boldsymbol{S}_{y\bar{x}_k}'^T\boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{S}_{yw_k}'\boldsymbol{\Psi}_y^{-1}\boldsymbol{S}_{yw_k}'^T\boldsymbol{K}_k^T \\
&= \left(\bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1/2} - \boldsymbol{K}_k\boldsymbol{S}_{yx_k}'\boldsymbol{\Psi}_x^{-1/2}\right)\left(\bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1/2} - \boldsymbol{K}_k\boldsymbol{S}_{yx_k}'\boldsymbol{\Psi}_x^{-1/2}\right)^T \\
&\quad + \left(\boldsymbol{K}_k\boldsymbol{S}_{yw_k}'\boldsymbol{\Psi}_y^{-1/2}\right)\left(\boldsymbol{K}_k\boldsymbol{S}_{yw_k}'\boldsymbol{\Psi}_y^{-1/2}\right)^T \qquad (3.68) \\
&= \mathcal{H}\left(\left[\begin{array}{cc} \bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1/2} - \boldsymbol{K}_k\boldsymbol{S}_{yx_k}'\boldsymbol{\Psi}_x^{-1/2} & \boldsymbol{K}_k\boldsymbol{S}_{yw_k}'\boldsymbol{\Psi}_y^{-1/2} \end{array}\right]\right) \qquad (3.69)
\end{aligned}$$

Similarly for the DD2 filter, if the measurement update given in Eq. (3.59) is taken as the initial guess for the state estimate, then a one-step Huber update can be written as

$$\begin{aligned}
\boldsymbol{S}_{y_k}^{(1)} &= \mathcal{H}\left(\left[\begin{array}{cccc} \boldsymbol{S}_{y\bar{x}_k}'\boldsymbol{\Psi}_x^{-1/2} & \boldsymbol{S}_{yw_k}'\boldsymbol{\Psi}_y^{-1/2} & \boldsymbol{S}_{y\bar{x}_k}''\boldsymbol{\Psi}_x^{-1/2} & \boldsymbol{S}_{yw_k}''\boldsymbol{\Psi}_y^{-1/2} \end{array}\right]\right) \qquad (3.70) \\
\boldsymbol{K}_k^{(1)} &= \bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1}\boldsymbol{S}_{y\bar{x}_k}'^T\left(\boldsymbol{S}_{y_k}^{(1)}\boldsymbol{S}_{y_k}^{(1)^T}\right)^{-1} \qquad (3.71) \\
\hat{\boldsymbol{x}}_k &= \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k^{(1)}\left(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_k\right) \qquad (3.72) \\
\hat{\boldsymbol{S}}_{x_k} &= \mathcal{H}\left(\left[\begin{array}{cc} \bar{\boldsymbol{S}}_{x_k}\boldsymbol{\Psi}_x^{-1/2} - \boldsymbol{K}_k^{(1)}\boldsymbol{S}_{yx_k}'\boldsymbol{\Psi}_x^{-1/2} & \boldsymbol{K}_k^{(1)}\boldsymbol{S}_{yw_k}'\boldsymbol{\Psi}_y^{-1/2} \\ \boldsymbol{K}_k^{(1)}\boldsymbol{S}_{yx_k}''\boldsymbol{\Psi}_x^{-1/2} & \boldsymbol{K}_k^{(1)}\boldsymbol{S}_{yw_k}''\boldsymbol{\Psi}_y^{-1/2} \end{array}\right]\right) \qquad (3.73)
\end{aligned}$$

In each case, $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ are diagonal matrices computed from the Huber ψ function, with residuals that take the form

$$\boldsymbol{\zeta} = \left[\begin{array}{cc} \boldsymbol{S}_{w_k}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{S}}_{x_k}^{-1} \end{array}\right] \cdot \left\{\begin{array}{c} \boldsymbol{y}_k - \bar{\boldsymbol{y}}_k \\ \\ \hat{\boldsymbol{x}}_k^{(0)} - \bar{\boldsymbol{x}}_k \end{array}\right\} \qquad (3.74)$$

where the superscript (0) refers to the initial state estimate computed from the standard DD1 or DD2 update.

3.4 Adaptive Filtering For Unknown Noise Statistics

Although the Huber technique has been shown to be beneficial in the presence of contaminated Gaussian probability distributions, the approach can still be improved upon by including a method by which the covariance of the main Gaussian distribution can be estimated along with the state and state error covariance matrix. A technique such as this allows the filter to adaptively tune the gain matrix to suite either slow changes in the error statistics, or statistics that are not well known. For example, a rendezvous radar used in a Mars sample return mission is very likely not to behave in the same manner as tested on the Earth, prior to launch, cruise, entry and descent to the Mars surface, and then a launch from the surface of Mars back to an Earth–return vehicle.

3.4.1 Myers–Tapley Method

An intuitive approach to adaptive state estimation is proposed by Myers and Tapley.[66] In this approach, the measurement residual sequence is mined to produce estimates of the measurement noise statistics, and state prediction residuals are mined to compute estimates of the process noise statistics. The estimators make use of a sliding window of N stored measurement noise and process noise residuals to compute the noise statistics. In batch form, the estimator for the measurement noise covariance can be derived by first defining the empirical covariance matrix

$$\boldsymbol{C}_\zeta = \frac{1}{N-1}\sum_{j=1}^{N}\left(\boldsymbol{\zeta}_j - \bar{\boldsymbol{\zeta}}\right)\left(\boldsymbol{\zeta}_j - \bar{\boldsymbol{\zeta}}\right)^T \tag{3.75}$$

where $\boldsymbol{\zeta}_j$ is the jth stored measurement residual, and $\bar{\boldsymbol{\zeta}}$ is the sample mean of the residuals,

$$\bar{\boldsymbol{\zeta}} = \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\zeta}_j \tag{3.76}$$

The expected value of $\boldsymbol{C}_\zeta$ is[66]

$$\mathrm{E}\left[\boldsymbol{C}_\zeta\right] = \boldsymbol{R} + \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{H}_j\bar{\boldsymbol{P}}_j\boldsymbol{H}_j^T \tag{3.77}$$

By substituting Eq. (3.75) into Eq. (3.77), an estimate for the measurement noise covariance matrix is

$$\hat{\boldsymbol{R}} = \frac{1}{N-1}\sum_{j=1}^{N}\left[\left(\boldsymbol{\zeta}_j - \bar{\boldsymbol{\zeta}}\right)\left(\boldsymbol{\zeta}_j - \bar{\boldsymbol{\zeta}}\right)^T - \left(\frac{N-1}{N}\right)\boldsymbol{H}_j\bar{\boldsymbol{P}}_j\boldsymbol{H}_j^T\right] \tag{3.78}$$

In order to form estimates for the process noise statistics, the process noise sample is defined as $\boldsymbol{\lambda}_j = \hat{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}_j$. Then, the empirical covariance matrix for $\boldsymbol{\lambda}$ is

$$\boldsymbol{C}_\lambda = \frac{1}{N-1}\sum_{j=1}^{N}\left(\boldsymbol{\lambda}_j - \bar{\boldsymbol{\lambda}}\right)\left(\boldsymbol{\lambda}_j - \bar{\boldsymbol{\lambda}}\right)^T \tag{3.79}$$

where

$$\bar{\boldsymbol{\lambda}} = \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\lambda}_j \tag{3.80}$$

The expected value of $\boldsymbol{C}_\lambda$ is[67] $\mathrm{E}\left[\boldsymbol{C}_\lambda\right] = \boldsymbol{K}_k\boldsymbol{H}_k\bar{\boldsymbol{P}}_k = \bar{\boldsymbol{P}}_k^\star + \tilde{\boldsymbol{Q}}_k - \hat{\boldsymbol{P}}_k$, where $\bar{\boldsymbol{P}}_k^\star$ is the propagated covariance without the process noise component, given by $\bar{\boldsymbol{P}}_k^\star = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{\Phi}_{k-1}^T$ where $\boldsymbol{\Phi}_{k-1}$ is the state transition matrix computed from Eq. (3.8). By manipulating Eq. (3.7), an estimator for the process noise covariance matrix is

$$\hat{\boldsymbol{Q}} = \frac{1}{N-1}\sum_{j=1}^{N}\left[\left(\boldsymbol{\lambda}_j - \bar{\boldsymbol{\lambda}}\right)\left(\boldsymbol{\lambda}_j - \bar{\boldsymbol{\lambda}}\right)^T - \left(\frac{N-1}{N}\right)\left(\bar{\boldsymbol{P}}_j^\star - \hat{\boldsymbol{P}}_j\right)\right] \tag{3.81}$$

An adaptive filter can function by using some initial guess of the measurement noise and process noise matrices, storing the residuals for the first N frames, and then updating the covariance estimates based on Eqs.(3.78) and (3.81) at each subsequent frame.

It is important to note that the Myers–Tapley method for adaptively estimating the measurement noise and process noise covariance matrices make use of the sample mean and covariance of the stored residuals, which are non–robust estimators. This lack of robustness implies that the performance Myers–Tapley adaptive method can degrade in the presence of non–Gaussianity. Therefore it is of interest to develop a modification of the Myers–Tapley approach that is robust with respect to non–Gaussian distributions, which is the subject of the following section.

3.4.2 Modified Myers–Tapley Method

3.4.2.1 Robust Covariance Estimation

This section discusses a robust modification of the Myers–Tapley adaptive filter in order to protect against the possible presence of non–Gaussian distributions in the measurement and/or process noise. The main issue with the Myers–Tapley method is the use of the sample covariance and the sample mean in forming the noise covariance estimates from the stored residuals. Fortunately, the technique can be modified to use a robust form of the covariance estimate in computing the measurement and process noise statistics.

The proposed technique makes use of the leverage point identification methods introduced in Sec. 2.4. The Myers–Tapley adaptive tuning method can be modified in order to account for non–Gaussianity by means of using the robust covariance estimates based on the Mahalanobis distances or the projection statistics of the stored residuals. In particular, the measurement and process noise covariance estimates can be written as

$$\begin{aligned} \hat{\boldsymbol{R}}^{\star} &= \left[\sum_{i=1}^{N} w_{\zeta_i} - 1\right]^{-1} \cdot \left[\sum_{i=1}^{N} \left(w_{\zeta_i}\boldsymbol{\zeta}_i - \bar{\boldsymbol{\zeta}}_r\right)\left(w_{\zeta_i}\boldsymbol{\zeta}_i - \bar{\boldsymbol{\zeta}}_r\right)^T\right] \\ &\quad - \underset{j \in N}{\text{median}} \left[\left(\boldsymbol{H}_j \bar{\boldsymbol{P}}_j^{1/2}\right) \boldsymbol{\Psi}_x^{-1} \left(\boldsymbol{H}_j \bar{\boldsymbol{P}}_j^{1/2}\right)^T\right] \end{aligned} \tag{3.82}$$

$$\begin{aligned} \hat{\boldsymbol{Q}} &= \left[\sum_{i=1}^{N} w_{\lambda_i} - 1\right]^{-1} \cdot \left[\sum_{i=1}^{N} \left(w_{\lambda_i}\boldsymbol{\lambda}_i - \bar{\boldsymbol{\lambda}}_r\right)\left(w_{\lambda_i}\boldsymbol{\lambda}_i - \bar{\boldsymbol{\lambda}}_r\right)^T\right] \\ &\quad - \underset{j \in N}{\text{median}} \left(\bar{\boldsymbol{P}}_k^{\star} - \hat{\boldsymbol{P}}_j\right) \end{aligned} \tag{3.83}$$

where w_{ζ_i} and w_{λ_i} are the weights based on the Mahalanobis distances or projection statistics of the measurement and process noise residuals, respectively, and

$$\bar{\boldsymbol{\zeta}}_r = \left[\sum_{i=1}^{N} w_{\zeta_i}\right]^{-1} \cdot \left[\sum_{i=1}^{N} w_{\zeta_i}\zeta_i\right] \tag{3.84}$$

$$\bar{\boldsymbol{\lambda}}_r = \left[\sum_{i=1}^{N} w_{\lambda_i}\right]^{-1} \cdot \left[\sum_{i=1}^{N} w_{\lambda_i}\lambda_i\right] \tag{3.85}$$

where w_i are weights computed from the statistics by means of

$$w_i = \min\left[\ 1, \quad \left(\chi^2_{n,\alpha}/\mathcal{M}_i^2\right)\ \right] \tag{3.86}$$

or

$$w_i = \min\left[\ 1, \quad \left(\chi^2_{n,\alpha}/\mathcal{P}_i^2\right)\ \right] \tag{3.87}$$

for some specified probability α.

The matrix $\hat{\boldsymbol{R}}^{\star}$ is related to the measurement noise covariance estimate as

$$\hat{\boldsymbol{R}}^{\star} = \hat{\boldsymbol{R}}^{1/2} \bar{\boldsymbol{\Psi}}_y{}^{-1} \left(\hat{\boldsymbol{R}}^{1/2}\right)^T \tag{3.88}$$

where $\bar{\boldsymbol{\Psi}}_y{}^{-1}$ is the median value of the $\boldsymbol{\Psi}_y$ matrix across the buffer of stored observations and residuals. The estimate of the measurement noise covariance matrix can be determined from a square root decomposition of Eq. (3.82). Specifically,

$$\hat{\boldsymbol{R}} = \left(\hat{\boldsymbol{R}}^{\star}\right)^{1/2} \bar{\boldsymbol{\Psi}}_y \left(\hat{\boldsymbol{R}}^{\star}\right)^{1/2\ \ T} \tag{3.89}$$

Note that in the DD1 and DD2 filter formulation, the quantity $\left(\boldsymbol{H}_j \bar{\boldsymbol{P}}_j^{1/2}\right) \boldsymbol{\Psi}_x^{-1} \left(\boldsymbol{H}_j \bar{\boldsymbol{P}}_j^{1/2}\right)^T$ is replaced by $\boldsymbol{S}'_{y\bar{x}_k} \boldsymbol{\Psi}_x^{-1} \boldsymbol{S}'_{y\bar{x}_k}{}^{\ T}$ (DD1) and $\mathcal{H}\left(\left[\ \boldsymbol{S}'_{y\bar{x}_k} \boldsymbol{\Psi}_x^{-1/2} \quad \boldsymbol{S}''_{y\bar{x}_k} \boldsymbol{\Psi}_x^{-1/2}\ \right]\right)$ (DD2) for the measurement noise covariance estimation. Similarly, the quantity $\bar{\boldsymbol{P}}_k^{\star} = \boldsymbol{\Phi}_{k-1} \hat{\boldsymbol{P}}_{k-1} \boldsymbol{\Phi}_{k-1}^T$ is replaced by $\boldsymbol{S}'_{x\hat{x}_k} \boldsymbol{S}'_{x\hat{x}_k}{}^{\ T}$ (DD1) and $\mathcal{H}\left(\left[\ \boldsymbol{S}'_{x\hat{x}_k} \quad \boldsymbol{S}''_{x\hat{x}_k}\ \right]\right)$ (DD2) for the process noise covariance estimation.

In summary, the proposed modified Myers–Tapley method makes use of a robust technique for outlier identification and weighting based on the projection statistics in forming the estimates of the measurement and process noise covariance matrices.

3.4.2.2 Estimation of the Contamination Parameter

The modified Myers–Tapley approach discussed in the previous section can also offer a crude scheme for estimating the contamination parameter ϵ by using the weighting parameters relating to the stored residual data. In particular, a crude estimate of the contamination parameter is

$$\hat{\epsilon}_k = 1 - \frac{1}{N} \sum_{i=1}^{N} w_{\zeta_i} \tag{3.90}$$

At each frame where the measurement and process noise covariances are computed, the contamination parameter can be estimated directly from the weighting parameters. Then, the optimal tuning parameter $\gamma^{\star}$ can be calculated from Eq. (2.38), which is then used within the Huber filter at each measurement update. Note, however, that it should be expected that the estimated contamination parameter will in general be biased as in cases of large contamination there will be some portion of errors drawn from the contaminating distribution that appear to be drawn from the nominal distribution, in other words false–negatives or the so-called Type II errors in detection theory. Likewise in cases of small contamination there will be some nonzero quantity of data that appear as outliers when they are in fact perfectly valid, in other words the Type I error in detection theory.

The bias of the contamination parameter is not necessarily problematic, so long as upper and lower bounds are set on the value of the tuning parameter γ used in the Huber measurement update. Clearly, the Huber technique is by nature a sub-optimal filter, since the purpose is to find a filter that is consistent across a range of distributions but not necessarily optimal at any one in particular. Therefore one could expect the performance of the technique to improve by having good estimates of the contamination parameter, but so long as the tuning parameter is bounded by some reasonable value then the robustness properties of the estimator will not be compromised by such a bias.

3.4.2.3 Fading Memory Filter

In the course of computing the estimates of the measurement and process noise covariance matrices, as well as the contamination parameter estimates, it is useful to introduction a "forgetting" factor, k_f, in order to smooth the estimate histories. The filter can be implemented as

$$\tilde{\boldsymbol{R}}_k = k_f \tilde{\boldsymbol{R}}_{k-1} + (1 - k_f)\, \hat{\boldsymbol{R}}_k \tag{3.91}$$

$$\tilde{\boldsymbol{Q}}_k = k_f \tilde{\boldsymbol{Q}}_{k-1} + (1 - k_f)\, \hat{\boldsymbol{Q}}_k \tag{3.92}$$

$$\tilde{\epsilon}_k = k_f \tilde{\epsilon}_{k-1} + (1 - k_f)\, \hat{\epsilon}_k \tag{3.93}$$

In this approach, the estimates based on the current set of stored residuals is averaged with the previous estimate, with k_f as a weighting parameter. In general the recommended value of k_f depends on the particular application, but a reasonable range is $k_f \in \left[\begin{array}{cc} 1/2 & 1 \end{array}\right]$.

Chapter 4

Spacecraft Rendezvous Guidance, Navigation, and Control

4.1 Introduction and Overview

This section describes the development of the 6 degree of freedom (6-DOF) nonlinear equations of motion for a maneuvering spacecraft with respect to an elliptical reference orbit. Coordinate systems are defined, which are then used to develop the equations of motion based on Newton–Euler mechanics. These equations of motion are used to develop nonlinear control methodologies based on feedback linearization techniques. Sensor models suitable for rendezvous navigation processing are also introduced in this chapter.

4.2 Coordinate Systems

The development of the 6–DOF rendezvous equations of motion requires the definition of several coordinate systems. First, a planet–centered inertial frame, $\mathcal{I}$, is introduced. This frame is aligned with the target spacecraft orbit (assuming no perturbations) such that the x-axis is oriented towards the periapsis, the z-axis is oriented along the positive orbit normal, and the y-axis completes a right-handed system. The local, or $\mathcal{L}$, frame has its origin located at the position of the target spacecraft and is defined such that the z-axis is oriented toward the center of the planet, the y-axis is oriented along the negative orbit normal, and the x-axis is in the transverse direction, completing the right-handed system. The $\mathcal{I}$ and $\mathcal{L}$ frames are illustrated in Fig. 4.1.

The chaser body frame is denoted by $\mathcal{B}$ with right-handed axes. It is assumed that the various sensors carried onboard the chaser spacecraft produce outputs referenced to this body frame. These sensors are introduced in later sections in this chapter, following the development of the equations of motion.

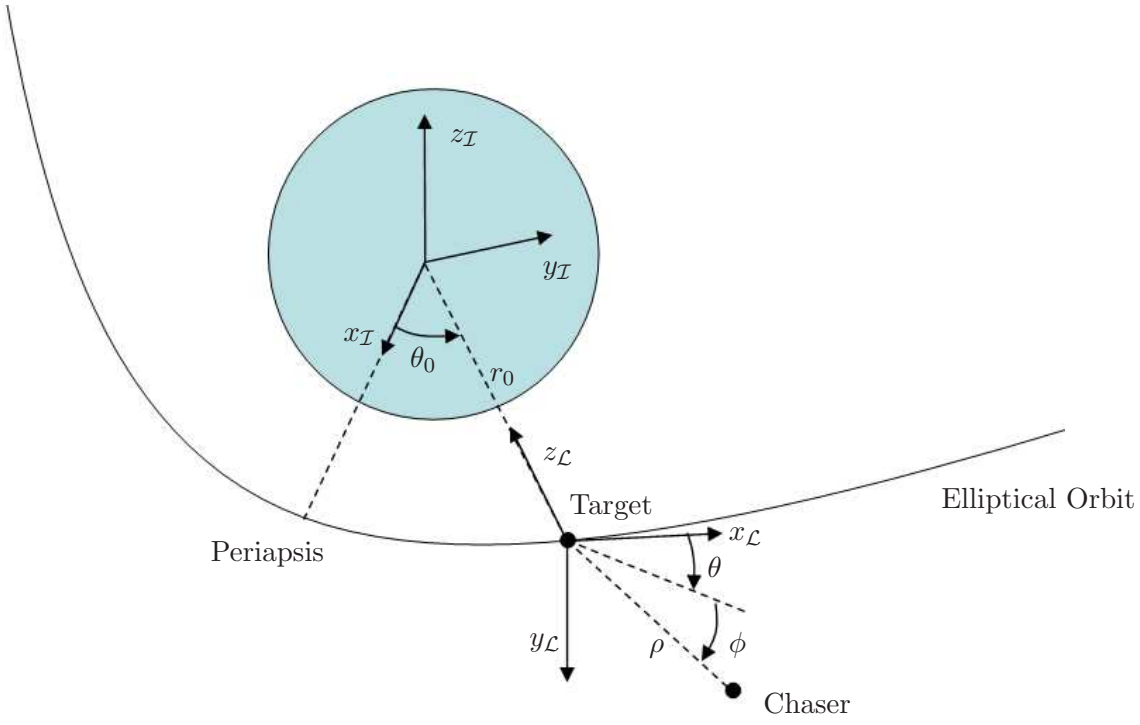


Figure 4.1: Illustration of $\mathcal{I}$ and $\mathcal{L}$ Coordinate Frames

4.3 Rotational Dynamics

4.3.1 Attitude Representations and Kinematics

There are many different attitude representations available in the literature to use for modeling spacecraft attitude and relative orientation, too numerous to review here in detail. Two attitude representations in particular are discussed in this chapter for application to the spacecraft rendezvous navigation problem. First, the quaternion representation is discussed, followed by the modified Rodrigues parameters.

4.3.1.1 Quaternions

The attitude dynamics of spacecraft is well understood and can be represented in many forms. Perhaps the most attractive form for attitude estimation is the quaternion representation, which is the smallest non-singular attitude parameterization. The quaternion can be expressed in terms of the Euler axis, $\boldsymbol{e}$, and the Euler angle, θ, of the transformation between a reference frame fixed in the body and the inertial frame. The quaternion elements are defined as $q_1 = e_1 \sin(\theta/2), q_2 = e_2 \sin(\theta/2), q_3 = e_3 \sin(\theta/2), q_4 = \cos(\theta/2)$. With knowledge of the quaternion elements, the

transformation matrix $\boldsymbol{T}$ relating the body frame to the inertial frame can be expressed as

$$\boldsymbol{T} = \begin{bmatrix} 1-2\left(q_1^2+q_3^2\right) & 2\left(q_1q_2+q_3q_4\right) & 2\left(q_1q_3-q_2q_4\right) \\ 2\left(q_1q_2-q_3q_4\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_1q_4+q_2q_3\right) \\ 2\left(q_1q_3+q_2q_4\right) & 2\left(q_2q_3-q_1q_4\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix} \tag{4.1}$$

By defining $\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$ it can be seen that the quaternion representation must satisfy the constraint equation $\boldsymbol{q}^T\boldsymbol{q} = 1$. Quaternions obey the multiplication rules

$$\boldsymbol{q}' \otimes \boldsymbol{q} = \boldsymbol{\Upsilon}\left(\boldsymbol{q}'\right)\boldsymbol{q} = \boldsymbol{\Xi}\left(\boldsymbol{q}\right)\boldsymbol{q}' \tag{4.2}$$

where

$$\boldsymbol{\Upsilon}\left(\mathbf{q}'\right) = \begin{bmatrix} q_4' & q_3' & -q_2' & q_1' \\ -q_3' & q_4' & q_1' & q_2' \\ q_2' & -q_1' & q_4' & q_3' \\ -q_1' & -q_2' & -q_3' & q_4' \end{bmatrix}, \quad \boldsymbol{\Xi}\left(\mathbf{q}\right) = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_3 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \tag{4.3}$$

and the inversion rule $\boldsymbol{q}^{-1} = \begin{bmatrix} -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix}^T$.

The quaternion kinematic equations can be written as[92]

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}\right)\boldsymbol{q} \tag{4.4}$$

where

$$\boldsymbol{\Omega}\left(\boldsymbol{\omega}\right) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 0 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{4.5}$$

and $\boldsymbol{\omega}$ is the angular velocity of the body frame with respect to the inertial frame.

Quaternions have the special property of bilinear kinematics. If high-rate measurements of the spacecraft angular velocity are available, from a gyroscope system for instance, then the angular velocity can reasonably be assumed to be constant over the sampling interval. In this case, the quaternion kinematic equations become constant coefficient ordinary differential equations and so the solution can be determined from the matrix exponential. Explicitly, the solution for the predicted quaternion at time $k+1$ can be written as[140]

$$\bar{\boldsymbol{q}}_{k+1} = \left[\mathbf{I}\cos\left(\hat{\omega}_k\Delta t\right) + \frac{\sin\left(\hat{\omega}_k\Delta t/2\right)}{\hat{\omega}_k}\boldsymbol{\Omega}\left(\hat{\boldsymbol{\omega}}_k\right)\right]\hat{\boldsymbol{q}}_k \tag{4.6}$$

where $\hat{\boldsymbol{q}}_k$ is the estimated quaternion at time k, $\bar{\boldsymbol{q}}_{k+1}$ is the predicted quaternion at time $k+1$, and $\hat{\omega}_k = \|\hat{\boldsymbol{\omega}}_k\|$ is the assumed constant angular velocity between time k and $k+1$.

The filtering of the attitude quaternion is complicated by the unit norm constraint. One standard approach for handling the constraint is to use a multiplicative method for the measurement update. In this approach, the attitude quaternion is replaced by an error quaternion, $\delta\boldsymbol{q}$, which allows for the processing of the error quaternion as a three component vector of small angles, denoted by $\delta\boldsymbol{\alpha}$. In this approach, the quaternion estimate is propagated to the star tracker measurement at time k by means of Eq. (4.6). Then an error quaternion relating the star tracker measurement quaternion to the predicted quaternion based on propagation from the previous measurement is

defined as $\delta\bar{\boldsymbol{q}}_k\left(\delta\bar{\boldsymbol{\alpha}}_k\right) = \boldsymbol{q}_{m_k} \otimes \bar{\boldsymbol{q}}_k^{-1}$. The first three components of this error quaternion are processed as independent small angles with measurement sensitivity matrix $\boldsymbol{H} = [\ \boldsymbol{I} \quad \boldsymbol{0}\ ]$.

The small angle error and gyroscope bias estimate error dynamics are given by

$$\delta\dot{\boldsymbol{\alpha}} = -\hat{\boldsymbol{\omega}}^{\times}\delta\boldsymbol{\alpha} - \delta\boldsymbol{\beta} + \boldsymbol{\eta}_{\omega} \tag{4.7}$$

$$\delta\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_{\beta} \tag{4.8}$$

where

$$\hat{\boldsymbol{\omega}}^{\times} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{4.9}$$

By setting $\boldsymbol{x} = [\ \delta\boldsymbol{\alpha} \quad \delta\boldsymbol{\beta}\ ]^T$, then these differential equations are in an appropriate form for filter implementation. The quaternion and gyroscope bias estimates following the measurement update are given by $\hat{\boldsymbol{q}}_k = \delta\boldsymbol{q}\left(\delta\hat{\boldsymbol{\alpha}}_k\right) \otimes \bar{\boldsymbol{q}}_k$ and $\hat{\boldsymbol{\beta}}_k = \bar{\boldsymbol{\beta}}_k + \delta\hat{\boldsymbol{\beta}}_k$.

4.3.1.2 Modified Rodrigues Parameters

The MRPs are defined in terms of the quaternions (q_1, q_2, q_3, q_4) as[92,94,145–147]

$$\boldsymbol{\sigma} = \frac{\boldsymbol{q}}{1+q_4} = \boldsymbol{e}\tan\left(\frac{\theta}{4}\right) \tag{4.10}$$

where $\boldsymbol{q} = (q_1, q_2, q_3)$ is the vector part of the quaternion, q_4 is the scalar part of the quaternion, $\boldsymbol{e}$ is the principal rotation axis, and θ is the principal rotation angle.

The shadow MRP set is defined as[92,94]

$$\boldsymbol{\sigma}^S = -\frac{\boldsymbol{\sigma}}{\boldsymbol{\sigma}^T\boldsymbol{\sigma}} = \boldsymbol{e}\tan\left(\frac{\theta - 2\pi}{4}\right) \tag{4.11}$$

Note that the MRP set $\boldsymbol{\sigma}$ behaves nearly linearly (with respect to θ) near the zero rotation and grows infinitely large after a revolution, while the shadow MRP set $\boldsymbol{\sigma}^S$ behaves linearly about 2π and is singular about the zero rotation. Further, while $\|\boldsymbol{\sigma}\| < 1$ (or > 1), $\boldsymbol{\sigma}$ describes the short (or long) rotation back to the origin, the shadow set $\boldsymbol{\sigma}^S$ describes the opposite rotation. The MRP and shadow MRP set can also be described as the *inner* and *outer* MRPs,[148] respectively, where inner refers to the MRP set within the unit sphere ($\|\boldsymbol{\sigma}\| < 1$) and outer refers to the MRP set outside the unit sphere ($\|\boldsymbol{\sigma}\| > 1$). Both inner and outer sets lie on the unit sphere when $\|\boldsymbol{\sigma}\| = 1$.

As proposed in Ref. 94, the shadow MRP set can be exploited to yield a globally non–singular attitude description with a minimal three–parameter coordinate set at the expense of a discontinuity. To avoid the singularity, the MRP set is switched to the shadow set before reaching the singularity. A convenient switching condition is the unit magnitude surface $\|\boldsymbol{\sigma}\| = 1$, such that the composite MRP description always satisfies $\|\boldsymbol{\sigma}(t)\| \leq 1$. This surface represents all possible orientations where the body has performed a principal rotation relative to the origin of $\theta = \pi$. Note that on this surface there are two possible MRP sets that describe the same attitude.

The coordinate transformation matrix involving the MRPs can be expressed as[92]

$$\boldsymbol{T} = \boldsymbol{I} - \frac{4\left(1 - \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)}{\left(1 + \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)^2}\boldsymbol{\sigma}^{\times} + \frac{8}{\left(1 + \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)^2}\left(\boldsymbol{\sigma}^{\times}\right)^2 \tag{4.12}$$

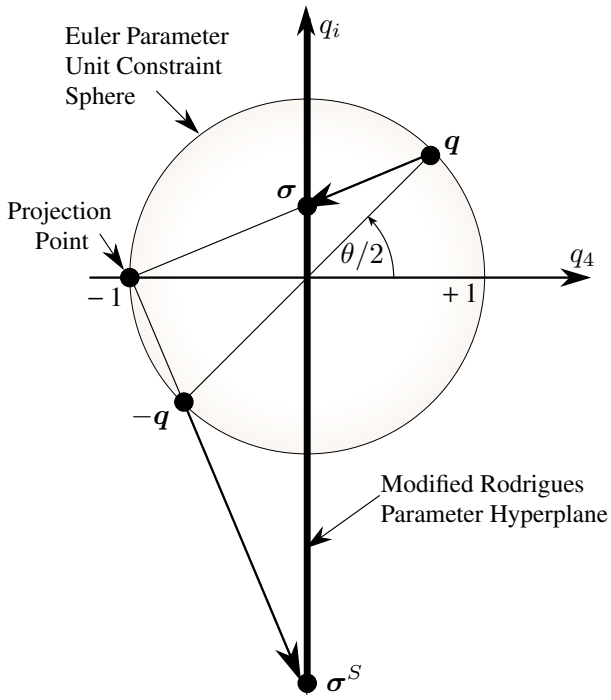


Figure 4.2: MRP Illustration as the Result of a Stereographic Projection

where $\boldsymbol{\sigma}^{\times}$ is the skew-symmetric cross product matrix. MRPs have additive and subtractive composition rules,

$$\boldsymbol{\sigma} = \delta\boldsymbol{\sigma} \oplus \boldsymbol{\sigma}_0 = \frac{\left(1-\boldsymbol{\sigma}_0^T\boldsymbol{\sigma}_0\right)\delta\boldsymbol{\sigma} + \left(1-\delta\boldsymbol{\sigma}^T\delta\boldsymbol{\sigma}\right)\boldsymbol{\sigma}_0 - 2\delta\boldsymbol{\sigma}^{\times}\boldsymbol{\sigma}_0}{1+\left(\boldsymbol{\sigma}_0^T\boldsymbol{\sigma}_0\right)\left(\delta\boldsymbol{\sigma}^T\delta\boldsymbol{\sigma}\right) - 2\boldsymbol{\sigma}_0^T\delta\boldsymbol{\sigma}} \tag{4.13}$$

$$\delta\boldsymbol{\sigma} = \boldsymbol{\sigma} \ominus \boldsymbol{\sigma}_0 = \frac{\left(1-\boldsymbol{\sigma}_0^T\boldsymbol{\sigma}_0\right)\boldsymbol{\sigma} - \left(1-\boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)\boldsymbol{\sigma}_0 + 2\boldsymbol{\sigma}^{\times}\boldsymbol{\sigma}_0}{1+\left(\boldsymbol{\sigma}_0^T\boldsymbol{\sigma}_0\right)\left(\boldsymbol{\sigma}^T\boldsymbol{\sigma}\right) - 2\boldsymbol{\sigma}_0^T\boldsymbol{\sigma}} \tag{4.14}$$

$$\tag{4.15}$$

Both sets of MRPs satisfy the same kinematic differential equation[92]

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}\boldsymbol{B}\left(\boldsymbol{\sigma}\right)\boldsymbol{\omega} = \frac{1}{4}\left[\left(1-\boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)\boldsymbol{I} + 2\boldsymbol{\sigma}^{\times} + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^T\right]\boldsymbol{\omega} \tag{4.16}$$

where $\boldsymbol{\omega}$ is the angular velocity. The matrix $\boldsymbol{B}$ has the useful property[92]

$$\boldsymbol{B}^{-1}\left(\boldsymbol{\sigma}\right) = \frac{1}{1+\boldsymbol{\sigma}^T\boldsymbol{\sigma}}\boldsymbol{B}^T\left(\boldsymbol{\sigma}\right) \tag{4.17}$$

Another useful property involving the matrix $\boldsymbol{B}$ is[155]

$$\boldsymbol{\sigma}^T\boldsymbol{B}\left(\boldsymbol{\sigma}\right)\boldsymbol{\omega} = \left(1+\boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)\boldsymbol{\sigma}^T\boldsymbol{\omega} \tag{4.18}$$

Aside from providing a non–singular attitude description, another advantage of the combined MRP set restricted to $\|\boldsymbol{\sigma}(t)\| \leq 1$ is that they behave nearly linearly for a large set of orientations. Figure 4.3 illustrates $\tan\left(\theta/4\right)$ and the linearized $\theta/4$ for rotations up to $\theta = \pi$.

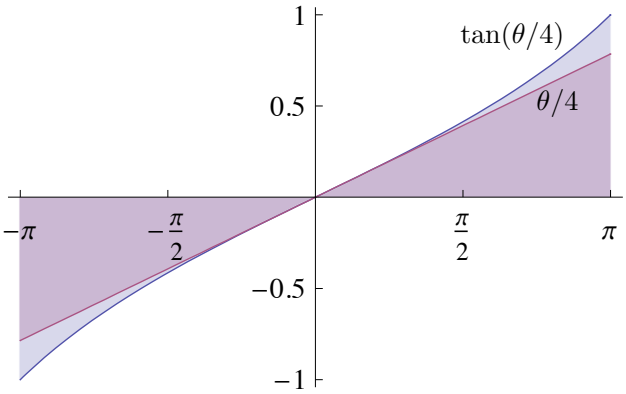


Figure 4.3: Illustration of the weakly nonlinear behavior of the MRPs restricted to $\|\boldsymbol{\sigma}\| \leq 1$.

Having the analytical mapping between two possible MRP sets allows for two attitude motion descriptions to be solved simultaneously, using only one integration of the kinematic equations. After integrating the kinematic equations, the MRP set can be switched if $\|\boldsymbol{\sigma}\| \geq 1$ and then the integration can continue. Note that the mapping in Eq. (4.11) is valid for any non–singular switching point. This observation allows the integration procedure to avoid the need to track the $\|\boldsymbol{\sigma}\| = 1$ surface crossing precisely. Instead, the mapping step is performed only if the MRP set falls outside this surface.

Note that in general, the MRP can switched to the shadow set at any surface of $\|\boldsymbol{\sigma}\| \geq 1$. The shadow MRP mapping cannot be performed at conditions $\|\boldsymbol{\sigma}\| < 1$. For example, suppose a switching condition of $\|\boldsymbol{\sigma}\| = 1/2$ is specified. It follows from Eq. (4.11) that $\|\boldsymbol{\sigma}^S\| = 2$. Since $2 > 1/2$, the MRP must immediately be switched back again and the cycle continues indefinitely. The most convenient switching condition is $\|\boldsymbol{\sigma}\| \geq 1$ since that corresponds to the principal rotation angle of π. However there may be certain circumstances where other switching surfaces are favorable for a particular application. Therefore the covariance transformations developed in the following section are kept to the general case of any switching surface greater than one.

Note that it is possible to construct other minimal attitude coordinate sets which are even more linear with respect to the principal rotation angle θ than the MRPs. Reference 149 calls them the Higher Order Rodrigues Parameters (HORPs). Parameters $\boldsymbol{\vartheta}$ can be developed which are written as

$$\boldsymbol{\vartheta} = \boldsymbol{e} \tan\left(\frac{\theta}{2N}\right) \tag{4.19}$$

where $N \geq 1$ is an integer value. These HORPs also contain multiple sets of possible values which can be used to avoid singular attitude descriptions. The MRP covariance mapping methods developed in this paper could be used for the HORP descriptions as well, but are not developed in this work.

4.3.1.3 Nonsingular Attitude Filtering Using Modified Rodrigues Parameters

In order to use the MRP shadow set singularity avoidance technique for attitude estimation, a mapping must also be developed in order to transform the MRP state estimate error covariance

matrix into the shadow set MRP state estimate error covariance matrix. In previous applications of the MRPs to attitude estimation problems, the state covariance matrix has implicitly been kept fixed during this switching to the shadow set.[101] The following section describes the application of the shadow MRP set for singularity avoidance in the Kalman filter, including a first–order covariance transformation to accompany the MRP singularity avoidance mapping.

During the course of state propagation or following the state update, the state can be switched to the shadow state if certain conditions are met, namely if $\|\boldsymbol{\sigma}\| > \sigma_r$ where σ_r is a threshold value. The shadow set transformation is given by $\boldsymbol{x}^S = \boldsymbol{\lambda}(\boldsymbol{x})$, where

$$\boldsymbol{\lambda}(\boldsymbol{x}) = \left\{ \begin{array}{c} -\left(\boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)^{-1}\boldsymbol{\sigma} \\ \boldsymbol{\beta} \end{array} \right\} \tag{4.20}$$

To examine the covariance transformation at the switching point, let the covariance matrix $\bar{\boldsymbol{P}}_k$ be decomposed into sub-matrices with the structure

$$\bar{\boldsymbol{P}}_k = \left[\begin{array}{cc} \boldsymbol{P}_{\sigma\sigma} & \boldsymbol{P}_{\sigma\beta} \\ \boldsymbol{P}_{\sigma\beta}^T & \boldsymbol{P}_{\beta\beta} \end{array} \right] \tag{4.21}$$

where $\boldsymbol{P}_{\sigma\sigma}$ is the covariance matrix of the MRP state, $\boldsymbol{P}_{\beta\beta}$ is the covariance matrix of the bias state, and $\boldsymbol{P}_{\sigma\beta}$ is the cross-correlation matrix between the MRP and the bias state. It follows that the covariance mapping to the shadow MRP set in the neighborhood of the reference MRP condition is given by

$$\begin{aligned} \bar{\boldsymbol{P}}_k^S = \boldsymbol{\Lambda}\bar{\boldsymbol{P}}_k\boldsymbol{\Lambda}^T &= \left[\begin{array}{cc} \boldsymbol{\Lambda}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{array} \right] \left[\begin{array}{cc} \boldsymbol{P}_{\sigma\sigma} & \boldsymbol{P}_{\sigma b} \\ \boldsymbol{P}_{\sigma\beta}^T & \boldsymbol{P}_{\beta\beta} \end{array} \right] \left[\begin{array}{cc} \boldsymbol{\Lambda}_{11}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{array} \right] \\ &= \left[\begin{array}{cc} \boldsymbol{\Lambda}_{11}\boldsymbol{P}_{\sigma\sigma}\boldsymbol{\Lambda}_{11}^T & \boldsymbol{\Lambda}_{11}\boldsymbol{P}_{\sigma\beta} \\ \boldsymbol{P}_{\sigma b}^T\boldsymbol{\Lambda}_{11}^T & \boldsymbol{P}_{\beta\beta} \end{array} \right] \end{aligned} \tag{4.22}$$

where

$$\boldsymbol{\Lambda} = \frac{\partial\boldsymbol{\lambda}}{\partial\boldsymbol{x}} = \left[\begin{array}{cc} \left(2\sigma^{-4}\boldsymbol{\sigma}\boldsymbol{\sigma}^T - \sigma^{-2}\boldsymbol{I}\right) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{array} \right] \tag{4.23}$$

and $\boldsymbol{\Lambda}_{11} = 2\sigma^{-4}\boldsymbol{\sigma}\boldsymbol{\sigma}^T - \sigma^{-2}\boldsymbol{I}$.

Note that this covariance mapping scales all MRP components. Assume that $\sigma = \|\boldsymbol{\sigma}\|$ is small, and the associated covariance components are small as well indicating good attitude knowledge. Then the corresponding shadow MRP set is stretched toward infinity due to σ being near zero. The associated covariance matrix for the shadow set is large as well, reflecting the large changes in coordinate values in the neighborhood of the singularity. The rate bias covariance is held constant during the MRP mapping, which is expected since the bias estimate itself is held constant in Eq. (4.20).

Following the development in Ref. 79 and 80, a first–order divided difference transformation of the state covariance matrix to the shadow state covariance matrix suitable for the DD1 filter is given by

$$\hat{\boldsymbol{P}}_k^{S1} = \frac{1}{4c^2}\sum_{j=1}^{n}\left[\boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k + c\hat{\boldsymbol{s}}_{x_j}\right) - \boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k - c\hat{\boldsymbol{s}}_{x_j}\right)\right]\left[\boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k + c\hat{\boldsymbol{s}}_{x_j}\right) - \boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k - c\hat{\boldsymbol{s}}_{x_j}\right)\right]^T \tag{4.24}$$

Similarly the second–order transformation suitable for the DD2 filter is

$$\begin{aligned}\hat{\boldsymbol{P}}_k^{S2} &= \hat{\boldsymbol{P}}_k^{S1} + \frac{c^2-1}{4c^4}\sum_{j=1}^{n}\left[\boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k + h\hat{\boldsymbol{s}}_{x_j}\right) + \boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k - c\hat{\boldsymbol{s}}_{x_j}\right) - 2\boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k\right)\right] \\ &\quad \cdot\left[\boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k + c\hat{\boldsymbol{s}}_{x_j}\right) + \boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k - h\hat{\boldsymbol{s}}_{x_j}\right) - 2\boldsymbol{\lambda}\left(\hat{\boldsymbol{x}}_k\right)\right]^T \end{aligned} \tag{4.25}$$

Following these covariance transformations at the switching point, the square–root decompositions of the state covariance can be calculated from Eqs. (3.33) and (3.34), which are in turn used to continue the state propagations forward in time according to Eqs. (3.37), (3.39), (3.51), (3.53), (3.55) and (3.57) until the next measurement update.

4.3.2 Attitude Dynamics

The dynamic equations describing the change in angular momentum are the well–known Euler equations,

$$\dot{\boldsymbol{\omega}} = -\mathbb{I}^{-1}\boldsymbol{\omega}^{\times}\mathbb{I}\boldsymbol{\omega} + \mathbb{I}^{-1}\boldsymbol{\tau} \tag{4.26}$$

where $\boldsymbol{\omega}$ is the inertial angular velocity of the body, $\mathbb{I}$ is the inertia tensor of the vehicle, and $\boldsymbol{\tau}$ is the net control and/or disturbance torque. The relative angular velocity between the chaser and the $\mathcal{L}$ frame is $\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \boldsymbol{\Omega}$ where $\boldsymbol{\Omega}$ is the angular velocity of the $\mathcal{L}$ frame (with all quantities represented in the spacecraft $\mathcal{B}$ frame). By solving for $\boldsymbol{\omega}$ and substituting into Eq. (4.26), the dynamic equation for the relative angular velocity becomes

$$\delta\dot{\boldsymbol{\omega}} = -\mathbb{I}^{-1}\delta\boldsymbol{\omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega} - \mathbb{I}^{-1}\delta\boldsymbol{\omega}^{\times}\mathbb{I}\boldsymbol{\Omega} - \mathbb{I}^{-1}\boldsymbol{\Omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega} - \mathbb{I}^{-1}\boldsymbol{\Omega}^{\times}\mathbb{I}\boldsymbol{\Omega} + \dot{\boldsymbol{\Omega}} + \mathbb{I}^{-1}\boldsymbol{\tau} \tag{4.27}$$

The angular velocity $\boldsymbol{\Omega}$ is given by

$$\boldsymbol{\Omega} = \boldsymbol{T}(\boldsymbol{\sigma})\left\{\begin{array}{c} 0 \\ -\omega_0 \\ 0 \end{array}\right\} \tag{4.28}$$

where $\boldsymbol{\sigma}$ is the MRP representing the transformation from the $\mathcal{L}$ frame to the $\mathcal{B}$ frame and ω_0 is the angular velocity of the reference orbit.

4.4 Translational Dynamics

The translation equations of motion for a spacecraft relative to an elliptical Keplerian reference orbit are well known and can be written as[92]

$$\ddot{x} - 2\omega_0\dot{z} - \dot{\omega_0}z - \omega_0^2 x + \frac{\mu x}{\left[x^2+y^2+(r_0-z)^2\right]^{3/2}} = u_x + v_x \tag{4.29}$$

$$\ddot{y} + \frac{\mu y}{\left[x^2+y^2+(r_0-z)^2\right]^{3/2}} = u_y + v_y \tag{4.30}$$

$$\ddot{z} + 2\omega_0\dot{x} + \dot{\omega}_0 x - \omega_0^2 z + \frac{\mu}{r_0^2} - \frac{\mu(r_0-z)}{\left[x^2+y^2+(r_0-z)^2\right]^{3/2}} = u_z + v_z \tag{4.31}$$

$$\ddot{r}_0 - r_0\omega_0^2 + \frac{\mu}{r_0^2} = 0 \tag{4.32}$$

$$\dot{\omega}_0 + \frac{2\dot{r}_0\omega_0}{r_0} = 0 \tag{4.33}$$

Here, x is the in-track position component, z is the position component along the negative radial direction, and y is the position along the negative orbit normal direction. The variables r_0 and ω_0 correspond to the radius and angular velocity of the reference orbit, and μ is the gravitational parameter of the planet. It is convenient to introduce a transformation of variables such that the equations of motion are in spherical coordinates centered at the reference point.[139] Such a transformation can be found by setting

$$x = \rho\cos\phi\cos\theta \tag{4.34}$$

$$y = \rho\sin\phi \tag{4.35}$$

$$z = -\rho\cos\phi\sin\theta \tag{4.36}$$

where ρ is the range between the target spacecraft and the maneuvering spacecraft, θ is an azimuth angle in the reference orbit plane measured from the x-axis, positive toward the $-z$ direction, and ϕ is an out of plane angle measured from the $x-z$ plane positive toward the y-axis. After the substitution of this transformation the equations of motion become

$$\begin{aligned}\ddot{\rho} &= \left(\omega_0 - \dot{\theta}\right)^2\rho\cos^2\phi + \rho\dot{\phi}^2 + \frac{\mu}{r_0^2}\sin\theta\cos\phi \\ &\quad -\frac{\mu\left(\rho + r_0\cos\phi\sin\theta\right)}{\left(r_0^2 + 2r_0\rho\cos\phi\sin\theta + \rho^2\right)^{3/2}} + u_\rho + v_\rho \end{aligned} \tag{4.37}$$

$$\begin{aligned}\ddot{\theta} &= \dot{\omega}_0 + 2\left(\omega_0 - \dot{\theta}\right)\frac{\dot{\rho}}{\rho} - 2\left(\omega_0 - \dot{\theta}\right)\dot{\phi}\tan\phi + \frac{\mu\cos\theta\sec\phi}{r_0^2\rho} \\ &\quad -\frac{\mu r_0\cos\theta\sec\phi}{\rho\left(r_0^2 + 2r_0\rho\cos\phi\sin\theta + \rho^2\right)^{3/2}} + u_\theta + v_\theta \end{aligned} \tag{4.38}$$

$$\begin{aligned}\ddot{\phi} &= -\frac{1}{2}\left(\omega_0 - \dot{\theta}\right)^2\sin 2\phi - \frac{2\dot{\phi}\dot{\rho}}{\rho} - \frac{\mu\sin\theta\sin\phi}{r_0^2\rho} \\ &\quad +\frac{\mu r_0\sin\theta\sin\phi}{\rho\left(r_0^2 + 2r_0\rho\cos\phi\sin\theta + \rho^2\right)^{3/2}} + u_\phi + v_\phi \end{aligned} \tag{4.39}$$

where the control inputs are

$$u_\rho = u_x\cos\theta\cos\phi + u_y\sin\phi - u_z\sin\theta\cos\phi \tag{4.40}$$

$$\begin{aligned}u_\theta &= -u_x\left[\frac{2r_0\rho\sin^2\theta + \left(r_0^2 + \rho^2\right)\sin\theta\sec\phi}{\rho\left(r_0^2 + 2r_0\rho\cos\phi\sin\theta + \rho^2\right)}\right] \\ &\quad -u_z\left[\frac{2r_0\rho\cos\theta\sin\theta + \left(r_0^2 + \rho^2\right)\cos\theta\sec\phi}{\rho\left(r_0^2 + 2r_0\rho\cos\phi\sin\theta + \rho^2\right)}\right] \end{aligned} \tag{4.41}$$

$$u_\phi = -u_x\frac{\cos\theta\sin\phi}{\rho} + u_y\frac{\cos\phi}{\rho} + u_z\frac{\sin\theta\sin\phi}{\rho} \tag{4.42}$$

with similar relations for the process noise inputs.

The translation equations of motion can be written in the form

$$\ddot{\boldsymbol{\eta}} = \boldsymbol{f}\left(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, t\right) + \boldsymbol{u} + \boldsymbol{v} \tag{4.43}$$

where $\boldsymbol{\eta} = \begin{bmatrix} \rho & \theta & \phi \end{bmatrix}^T$, $\boldsymbol{u} = \begin{bmatrix} u_\rho & u_\theta & u_\phi \end{bmatrix}^T$ and $\boldsymbol{v} = \begin{bmatrix} v_\rho & v_\theta & v_\phi \end{bmatrix}^T$.

4.5 Navigation Sensor Models

Vehicles performing autonomous rendezvous maneuvers must make use of a wide variety of navigation sensors to estimate the trajectory and perhaps other relevant parameters in order to successfully complete the mission. Having developed the dynamics of spacecraft rendezvous, this chapter discusses the modeling of sensor systems whose data can be processed to estimate the relative position, velocity, and orientation of the chaser vehicle.

4.5.1 Inertial Measurement Unit

An inertial measurement unit (IMU) is a device which measures applied accelerations and angular rates by using a system of accelerometers and gyroscopes. The mathematical modeling of such a system can be relatively complex when systematic errors are included in the model.[153,154] For the purposes of real–time error parameter estimation in the navigation filter, reduced–order models that capture the net error effects are preferred in order to reduce the computational complexity.

4.5.1.1 Gyroscope Model

The gyroscope system can be represented mathematically by using Farrenkopf's model. In this model, the sensed angular velocity if expressed as the true angular velocity with an additive bias and white noise. The bias term is itself a slowly varying parameter driven by white noise. The model can be expressed as

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\eta}_\omega \tag{4.44}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_\beta \tag{4.45}$$

where $\tilde{\boldsymbol{\omega}}$ is the sensed inertial angular velocity, $\boldsymbol{\omega}$ is the true inertial angular velocity, $\boldsymbol{\beta}$ is the measurement bias, and $\boldsymbol{\eta}_\omega$ and $\boldsymbol{\eta}_\beta$ are unbiased and uncorrelated random vectors with variances given by σ_ω^2 and σ_β^2, respectively. Discrete–time simulated gyroscope measurements can be generated according to this model by use of the equations[141]

$$\tilde{\boldsymbol{\omega}}_{k+1} = \boldsymbol{\omega}_k + \frac{1}{2}\left(\boldsymbol{\beta}_{k+1} + \boldsymbol{\beta}_k\right) + \left(\frac{\sigma_\omega^2}{\Delta t} + \frac{1}{12}\sigma_\beta^2 \Delta t\right)^{1/2} \boldsymbol{n}_\omega \tag{4.46}$$

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \sigma_\beta \left(\Delta t\right)^{1/2} \boldsymbol{n}_\beta \tag{4.47}$$

where k refers to the time increment, $\Delta t = t_{k+1} - t_k$ is the sampling interval, and $\boldsymbol{n}_\omega$ and $\boldsymbol{n}_\beta$ are unbiased, uncorrelated, unit–variance random vectors.

4.5.1.2 Accelerometer Model

An accelerometer system measures the accelerations applied to the spacecraft. Typically these measurements are corrupted by noise in addition to systematic error such as bias, scale factor, and misalignment. Bias can easily be accounted for by subtracting the accelerometer reading during quiescent periods of flight without thrust or any other disturbance acceleration. During thrusting maneuvers, scale factor uncertainties become important. A model for acceleration measurements including scale factor and noise errors is

$$\tilde{\boldsymbol{a}}_m = (\boldsymbol{I} + \boldsymbol{S})\, \boldsymbol{a}_m + \boldsymbol{\eta}_a \tag{4.48}$$

where $\boldsymbol{a}_m$ is the true acceleration at the IMU location, $\tilde{\boldsymbol{a}}_m$ is the sensed acceleration, $\boldsymbol{\eta}_a$ is the measurement noise, and $\boldsymbol{S}$ is a diagonal matrix of constant scale factor errors. The accelerations at the vehicle center of mass can be calculated from

$$\boldsymbol{a} = \boldsymbol{a}_m - \left(\dot{\boldsymbol{\omega}}^{\times} + \boldsymbol{\omega}^{\times}\boldsymbol{\omega}^{\times}\right) \boldsymbol{r}_m \tag{4.49}$$

where $\boldsymbol{a}$ is the center of mass acceleration and $\boldsymbol{r}_m$ is the position of the IMU with respect to the vehicle center of mass.

4.5.2 Star Tracker Sensors

It is assumed that a star tracker or some other generic attitude sensor is available to provide corrections to the attitude estimates formed by direct numerical integration of the angular velocity measurements, which are subject to error buildup due to integrating errors in the estimated bias and the random noise. The star tracker is assumed to output an estimated quaternion that relates the orientation of the body to the inertial frame. The quaternion estimates are assumed to be unbiased but with a superimposed random measurement noise. The output from such a sensor can be expressed as $\boldsymbol{q}_m = \delta\boldsymbol{q}\,(\delta\boldsymbol{\alpha}) \otimes \boldsymbol{q}$ where $\boldsymbol{q}$ is the quaternion representing the true orientation, $\boldsymbol{q}_m$ is the measured quaternion, and $\delta\boldsymbol{q}\,(\delta\boldsymbol{\alpha})$ is an error quaternion parameterized by a random angular error $\delta\boldsymbol{\alpha}$.

Similarly, the attitude sensing device can be assumed to output an estimated MRP that relates the orientation of the body to the inertial frame. The estimates are assumed to be unbiased but with a superimposed random measurement noise. The output from such a sensor can be expressed as $\tilde{\boldsymbol{\sigma}}_I = \boldsymbol{\sigma}_I + \delta\boldsymbol{\sigma}$ where $\boldsymbol{\sigma}_I$ is the MRP representing the true orientation of the vehicle with respect to inertial space, $\tilde{\boldsymbol{\sigma}}_I$ is the "measured" MRP, and $\delta\boldsymbol{\sigma}$ is an error MRP with covariance matrix denoted by $\boldsymbol{R}$. For instance, the measured MRP could be an output from the algorithm described in Ref. 150, involving vector measurements.

The inertial attitude MRP, $\boldsymbol{\sigma}_I$, can be written as a function of the relative attitude, $\boldsymbol{\sigma}$, and the inertial orbital position angle θ_0. First, note that the matrix of the transformation from the $\mathcal{I}$ frame to the $\mathcal{L}$ frame can be written as a sequence of two elementary rotations, $\boldsymbol{T}_1\,(-\pi/2) \cdot \boldsymbol{T}_3\,(\theta_0 + \pi/2)$ where $\boldsymbol{T}_i\,(\varsigma)$ represents the matrix for an elementary rotation of angle ς about axis i. Using the axis-angle representation of the MRP given in Eq. (4.10), the transformation sequence can be expressed as the composite MRP,

$$\boldsymbol{\sigma}_{\theta_0} = \begin{pmatrix} \tan(-\pi/8) \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ \tan\left[(1/4)\,(\theta_0 + \pi/2)\right] \end{pmatrix}$$

$$= \frac{1}{1+\tan^2(\pi/8)\tan^2\left[(1/4)(\theta_0+\pi/2)\right]} \begin{pmatrix} \left\{\tan^2\left[(1/4)(\theta_0+\pi/2)\right]-1\right\}\tan(\pi/8) \\ -2\tan(\pi/8)\tan\left[(1/4)(\theta_0+\pi/2)\right] \\ \left[1-\tan^2(\pi/8)\right]\tan\left[(1/4)(\theta_0+\pi/2)\right] \end{pmatrix} \quad (4.50)$$

The inertial attitude MRP can then be expressed as $\boldsymbol{\sigma}_I = \boldsymbol{\sigma} \oplus \boldsymbol{\sigma}_{\theta_0}$.

4.5.3 Laser/Radar Navigation Sensors

A radar or laser sensing system provides range, azimuth, elevation measurements between the sensor and the target. These measurements can be modeled as

$$\varrho \begin{Bmatrix} \cos\alpha\cos\varepsilon \\ \sin\alpha\cos\varepsilon \\ \sin\varepsilon \end{Bmatrix} = \boldsymbol{T}(\boldsymbol{\sigma})\,\boldsymbol{r} - \delta\boldsymbol{r}_s \quad (4.51)$$

where ϱ is the range, α is the azimuth angle, and ε is the elevation angle. Here, $\boldsymbol{r} = [x,\ y,\ z]^T$ is the cartesian relative position in the local frame and $\delta\boldsymbol{r}_s$ is the sensor position in the body frame. Assuming the target spacecraft has either a system of retro-reflectors,[157,158] a network of light emitting diodes (LEDS),[156] or a passive geometric sensing system,[159] that provides multiple range, azimuth, and elevation measurements, then the relative attitude between the target and chaser can also be determined in addition to range and bearing. This relative attitude is modeled in the same manner as the inertial attitude sensor given in Sec. 4.5.2, namely that $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma} + \delta\boldsymbol{\sigma}$ where $\tilde{\boldsymbol{\sigma}}$ is the "measured" relative MRP, $\boldsymbol{\sigma}$ is the true relative attitude MRP and $\delta\boldsymbol{\sigma}$ is an error MRP.

4.5.4 Orbit Sensor

A wide variety of sensors can be used to estimate the orbit of the spacecraft. For example, Global Position System (GPS) orbit determination sensors can be used to Earth orbiting and some lunar orbit cases.[160] In other cases, such as lunar orbit, autonomous orbit determination can be accomplished using optical sensors and landmark tracking.[161] For the purposes of this dissertation, it is assumed that some general orbit determination sensor is available for use in the state estimator. This sensor is assumed to provide position data in the form of radius r_0 and orbit angle θ_0 measurements.

4.6 Rendezvous Guidance and Control

This section develops 6-DOF guidance and control schemes for rendezvous in elliptical orbit. These guidance and control schemes are fairly straightforward, as will be seen presently, but they are adequate for the purposes of illustrating the navigation filter performance during rendezvous maneuvers. Considerably more advanced control schemes for spacecraft rendezvous have been published recently, for instance the adaptive control formulations in Refs. 162 and 163. If required, these control laws could be implemented to improve performance, however such work is beyond the scope of this research.

4.6.1 Translational Guidance and Control

Given a commanded translational reference trajectory, $\boldsymbol{\eta}_c$, a translation error state can be defined as $\delta\boldsymbol{\eta} = \boldsymbol{\eta} - \boldsymbol{\eta}_c$. This error state obeys the differential equations

$$\delta\ddot{\boldsymbol{\eta}} = \ddot{\boldsymbol{\eta}} - \ddot{\boldsymbol{\eta}}_c = \boldsymbol{f}\left(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, t\right) + \boldsymbol{u} + \boldsymbol{v} - \ddot{\boldsymbol{\eta}}_c \tag{4.52}$$

This system can be controlled by using a feedback linearization of the form

$$\boldsymbol{u} = \ddot{\boldsymbol{\eta}}_c - \boldsymbol{f}\left(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, t\right) - \boldsymbol{K}_\eta \delta\boldsymbol{\eta} - \boldsymbol{K}_{\dot{\eta}} \delta\dot{\boldsymbol{\eta}} \tag{4.53}$$

By substituting Eq. (4.53) into Eq. (4.52), the closed–loop system becomes

$$\delta\ddot{\boldsymbol{\eta}} + \boldsymbol{K}_\eta \delta\boldsymbol{\eta} + \boldsymbol{K}_{\dot{\eta}} \delta\dot{\boldsymbol{\eta}} = \boldsymbol{v} \tag{4.54}$$

This closed–loop system is asymptotically stable in the absence of disturbance inputs for any gain matrices $\boldsymbol{K}_\eta > 0$ and $\boldsymbol{K}_{\dot{\eta}} > 0$. These gains can be specified by using a number of different approaches. For example, an approach based on desired closed–loop damping ratio and natural frequency is investigated in Ref. 48. Note that non–zero disturbance accelerations result from both model error and feedback error due to erroneous state estimates from the navigation outputs. This mixture of deterministic and stochastic disturbances suggests a gain design using a suboptimal H_∞ approach. To this end, the closed–loop system can first be written in the standard state space form

$$\dot{\boldsymbol{x}}_t = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \left[\boldsymbol{u}_t + \boldsymbol{d}_t\right] \tag{4.55}$$

where $\boldsymbol{x}_t = \left[\begin{array}{cc} \delta\boldsymbol{\eta} & \delta\dot{\boldsymbol{\eta}} \end{array}\right]^T$ is the translational state, $\boldsymbol{u}_t$ is the acceleration input, $\boldsymbol{d}_t$ is a disturbance input, and

$$\boldsymbol{A}_t = \left[\begin{array}{cc} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{array}\right], \qquad \boldsymbol{B}_t = \left[\begin{array}{c} \boldsymbol{0} \\ \boldsymbol{I} \end{array}\right] \tag{4.56}$$

The infinite horizon full information state feedback H_∞ control results from the solution of the Ricatti equation[129]

$$\boldsymbol{X}_t \boldsymbol{A}_t + \boldsymbol{A}_t^T \boldsymbol{X}_t - \boldsymbol{X}_t \left(\boldsymbol{B}_t \boldsymbol{B}_t^T - \frac{1}{\kappa^2} \boldsymbol{B}_t \boldsymbol{B}_t^T\right) \boldsymbol{X}_t + \boldsymbol{C}_t = \boldsymbol{0} \tag{4.57}$$

where $\boldsymbol{C}_t$ is a state error weighting matrix and κ is the the H_∞ performance bound. Once the algebraic Ricatti equation is solved for matrix $\boldsymbol{X}_t$, the state feedback control law is

$$\boldsymbol{u}_t = -\boldsymbol{B}_t^T \boldsymbol{X}_t \boldsymbol{x}_t \tag{4.58}$$

If the state error weighting matrix $\boldsymbol{C}_t$ is chosen to be diagonal, then the state feedback gain matrix can be decomposed into the following structure to solve for the gains $\boldsymbol{K}_\eta$ and $\boldsymbol{K}_{\dot{\eta}}$,

$$\boldsymbol{B}_t^T \boldsymbol{X}_t = \left[\begin{array}{cc} \boldsymbol{K}_\eta & \boldsymbol{K}_{\dot{\eta}} \end{array}\right] \tag{4.59}$$

It is assumed here that a suitable reference trajectory $\boldsymbol{\eta}_c$ is available to provide the guidance commands. In general this trajectory can be specified using a multitude of approaches depending on the specific nature of the rendezvous and docking sequence and path constraints for a particular spacecraft application. For example, the rendezvous guidance can be formulated as either minimum fuel or minimum time optimal trajectories with path constraints,[164] as nonoptimal glideslope approaches,[165] or using non-linear guidance techniques as described in Ref 166.

4.6.2 Attitude Guidance and Control

Attitude guidance commands are generated by solving for the MRP $\boldsymbol{\sigma}_c$ that provides zero azimuth and elevation angles so that the radar boresight is aligned with the target vehicle. The following system of nonlinear algebraic equations is solved at each guidance cycle

$$\begin{aligned} \alpha_c(\boldsymbol{\sigma}_c) &= 0 && (4.60)\\ \varepsilon_c(\boldsymbol{\sigma}_c) &= 0 && (4.61)\\ \varphi_c(\boldsymbol{\sigma}_c) &= 0 && (4.62) \end{aligned}$$

The quantity φ_c specifies the relative roll angle and is defined by

$$\begin{aligned} \varphi_c &= \boldsymbol{y}_{\mathcal{L}} \cdot \boldsymbol{z}_{\mathcal{B}} \\ &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \boldsymbol{T}(\boldsymbol{\sigma})^T \cdot \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \\ &= \frac{8\sigma_2\sigma_3 + 4\sigma_1\left(1 - \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)}{\left(1 + \boldsymbol{\sigma}^T\boldsymbol{\sigma}\right)^2} \end{aligned} \tag{4.63}$$

A zero-order hold is applied to the MRP attitude commands in order to generate a MRP rate command. The rate command is computed using a backward difference derivative between the current MRP command and the lagged MRP command. The commanded MRP and MRP rate are the inputs to the attitude control law.

Given the commanded MRP and MRP rate, $\boldsymbol{\sigma}_c$ and $\dot{\boldsymbol{\sigma}}_c$, respectively, an attitude control law to track these commands can be constructed using the multi–input backstepping method.[167] Here, the control design is split into a sequence of two sub-problems. First, the angular velocity is assumed to be a control, which is chosen to suitably stabilize the attitude kinematics. Next, torque commands are generated in order to track the desired angular velocity. In order to construct this controller, an error MRP is defined as $\delta\boldsymbol{\sigma} = \boldsymbol{\sigma} \ominus \boldsymbol{\sigma}_c$. The error MRP obeys the kinematic differential equation

$$\delta\dot{\boldsymbol{\sigma}} = \frac{1}{4}\boldsymbol{B}(\delta\boldsymbol{\sigma})\,\delta\boldsymbol{\omega} \tag{4.64}$$

Assuming the relative angular rate $\delta\boldsymbol{\omega} = \boldsymbol{\nu}(\delta\boldsymbol{\sigma})$ can be chosen arbitrarily, a Lyapunov function can easily be constructed in order to stabilize the relative MRP kinematics. To this end, a candidate Lyapunov function is

$$V(\delta\boldsymbol{\sigma}) = 2k_v \ln\left(1 + \delta\boldsymbol{\sigma}^T\delta\boldsymbol{\sigma}\right) \tag{4.65}$$

where $k_v > 0$ is a gain.

Making use of the identity in Eq. (4.18), the candidate Lyapunov function rate is

$$\dot{V} = \frac{4k_v}{1 + \delta\boldsymbol{\sigma}^T\delta\boldsymbol{\sigma}}\delta\boldsymbol{\sigma}^T\delta\dot{\boldsymbol{\sigma}} = \frac{k_v}{1 + \delta\boldsymbol{\sigma}^T\delta\boldsymbol{\sigma}}\delta\boldsymbol{\sigma}^T\boldsymbol{B}(\delta\boldsymbol{\sigma})\,\boldsymbol{\nu}(\delta\boldsymbol{\sigma}) = k_v\delta\boldsymbol{\sigma}^T\boldsymbol{\nu}(\delta\boldsymbol{\sigma}) \tag{4.66}$$

By choosing the function $\boldsymbol{\nu}(\delta\boldsymbol{\sigma}) = -\boldsymbol{K}_\sigma\delta\boldsymbol{\sigma}$, where $\boldsymbol{K}_\sigma > 0$, the Lyapunov function rate becomes $\dot{V} = -k_v\delta\boldsymbol{\sigma}^T\boldsymbol{K}_\sigma\delta\boldsymbol{\sigma} < 0$. According to Theorem 4.2 in Ref. 167, the relative kinematics are asymptotically stable with this choice of function $\boldsymbol{\nu}(\delta\boldsymbol{\sigma})$ since the function $V(\delta\boldsymbol{\sigma})$ is positive definite with negative definite rate. Moreover, the origin is exponentially stable as can be seen

from the following expansion of the kinematics differential equation. The closed-loop kinematics are

$$\begin{aligned}\delta\dot{\boldsymbol{\sigma}} &= -\frac{1}{4}\boldsymbol{B}\left(\delta\boldsymbol{\sigma}\right)\boldsymbol{K}_{\sigma}\delta\boldsymbol{\sigma} \\ &= -\frac{1}{2}\left(\boldsymbol{I}-\delta\boldsymbol{\sigma}^{\times}+\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^{T}-\frac{1+\delta\boldsymbol{\sigma}^{T}\delta\boldsymbol{\sigma}}{2}\boldsymbol{I}\right)\boldsymbol{K}_{\sigma}\delta\boldsymbol{\sigma} \\ &= -\frac{1}{4}\boldsymbol{K}_{\sigma}\delta\boldsymbol{\sigma}+\frac{1}{2}\left(\delta\boldsymbol{\sigma}^{\times}+\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^{T}-\frac{1+\delta\boldsymbol{\sigma}^{T}\delta\boldsymbol{\sigma}}{2}\boldsymbol{I}\right)\boldsymbol{K}_{\sigma}\delta\boldsymbol{\sigma} \end{aligned} \tag{4.67}$$

Clearly the linearization of the closed–loop kinematics is Hurwitz since $\boldsymbol{K}_{\sigma}>0$, and therefore the origin $\delta\boldsymbol{\sigma}=\boldsymbol{0}$ is exponentially stable, in view of Theorem 4.15 in Ref. 167.

Next, control torques must be provided that cause the spacecraft angular velocity to track the desired angular velocity $\delta\boldsymbol{\omega}=\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)=-\boldsymbol{K}_{\sigma}\delta\boldsymbol{\sigma}$. An augmented Lyapunov function can be introduced as

$$V_{a}\left(\delta\boldsymbol{\sigma},\delta\boldsymbol{\omega}\right)=V\left(\delta\boldsymbol{\sigma}\right)+\frac{1}{2}\left[\delta\boldsymbol{\omega}-\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)\right]^{T}\left[\delta\boldsymbol{\omega}-\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)\right] \tag{4.68}$$

The Lyapunov function rate is

$$\begin{aligned}\dot{V}_{a}\left(\delta\boldsymbol{\sigma},\delta\boldsymbol{\omega}\right) &= \frac{\partial V}{\partial\delta\boldsymbol{\sigma}}\delta\dot{\boldsymbol{\sigma}}+\left[\delta\boldsymbol{\omega}-\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)\right]^{T}\left[\delta\dot{\boldsymbol{\omega}}-\frac{\partial\boldsymbol{\nu}}{\partial\delta\boldsymbol{\sigma}}\delta\dot{\boldsymbol{\sigma}}\right] \\ &= \frac{1}{4}\frac{\partial V}{\partial\delta\boldsymbol{\sigma}}\boldsymbol{B}\left(\delta\boldsymbol{\sigma}\right)\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)+\frac{1}{4}\frac{\partial V}{\partial\delta\boldsymbol{\sigma}}\boldsymbol{B}\left(\delta\boldsymbol{\sigma}\right)\left[\delta\boldsymbol{\omega}-\nu\left(\delta\boldsymbol{\sigma}\right)\right] \\ &\quad+\left[\delta\boldsymbol{\omega}-\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)\right]^{T}\left[-\mathbb{I}^{-1}\delta\boldsymbol{\omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega}-\mathbb{I}^{-1}\delta\boldsymbol{\omega}^{\times}\mathbb{I}\boldsymbol{\Omega}-\mathbb{I}^{-1}\boldsymbol{\Omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega}\right. \\ &\quad\left.-\mathbb{I}^{-1}\boldsymbol{\Omega}^{\times}\mathbb{I}\boldsymbol{\Omega}+\dot{\boldsymbol{\Omega}}+\mathbb{I}^{-1}\boldsymbol{\tau}-\frac{1}{4}\frac{\partial\boldsymbol{\nu}}{\partial\delta\boldsymbol{\sigma}}\boldsymbol{B}\left(\delta\boldsymbol{\sigma}\right)\delta\boldsymbol{\omega}\right] \end{aligned} \tag{4.69}$$

By choosing the control input

$$\begin{aligned}\boldsymbol{\tau} &= \delta\boldsymbol{\omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega}+\delta\boldsymbol{\omega}^{\times}\mathbb{I}\boldsymbol{\Omega}+\boldsymbol{\Omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega}+\boldsymbol{\Omega}^{\times}\mathbb{I}\boldsymbol{\Omega}-\mathbb{I}\dot{\boldsymbol{\Omega}} \\ &\quad+\frac{1}{4}\mathbb{I}\frac{\partial\boldsymbol{\nu}}{\partial\delta\boldsymbol{\sigma}}\boldsymbol{B}\left(\delta\boldsymbol{\sigma}\right)\delta\boldsymbol{\omega}-\frac{1}{4}\mathbb{I}\left[\frac{\partial V}{\partial\delta\boldsymbol{\sigma}}\boldsymbol{B}\left(\delta\boldsymbol{\sigma}\right)\right]^{T}-\mathbb{I}\boldsymbol{K}_{\omega}\left[\delta\boldsymbol{\omega}-\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)\right] \end{aligned} \tag{4.70}$$

where $\boldsymbol{K}_{\omega}>0$, the Lyapunov function rate becomes

$$\dot{V}_{a}=-\delta\boldsymbol{\sigma}^{T}\boldsymbol{K}_{\sigma}\delta\boldsymbol{\sigma}-\left[\delta\boldsymbol{\omega}-\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)\right]^{T}\boldsymbol{K}_{\omega}\left[\delta\boldsymbol{\omega}-\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)\right]<0 \tag{4.71}$$

Since the function V_{s} is positive definite and radially unbounded with a negative definite rate, it follows from Lyapunov's direct method (Theorem 4.3 in Ref. 167) that the origin of the closed–loop system is globally asymptotically stable.

By substituting the appropriate expressions for $V\left(\delta\boldsymbol{\sigma}\right)$ and $\boldsymbol{\nu}\left(\delta\boldsymbol{\sigma}\right)$, the control torque τ is

$$\begin{aligned}\boldsymbol{\tau} &= \delta\boldsymbol{\omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega}+\delta\boldsymbol{\omega}^{\times}\mathbb{I}\boldsymbol{\Omega}+\boldsymbol{\Omega}^{\times}\mathbb{I}\delta\boldsymbol{\omega}+\boldsymbol{\Omega}^{\times}\mathbb{I}\boldsymbol{\Omega}-\mathbb{I}\dot{\boldsymbol{\Omega}} \\ &\quad-\frac{1}{4}\mathbb{I}\boldsymbol{K}_{\sigma}\boldsymbol{B}\left(\delta\boldsymbol{\sigma}\right)\delta\boldsymbol{\omega}-\frac{k_{v}}{2}\mathbb{I}\delta\boldsymbol{\sigma}-\mathbb{I}\boldsymbol{K}_{\omega}\left[\delta\boldsymbol{\omega}+\boldsymbol{K}_{\sigma}\delta\boldsymbol{\sigma}\right] \end{aligned} \tag{4.72}$$

Substituting this control law into Eq. (4.27) and linearizing about $\left(\delta\boldsymbol{\sigma},\delta\boldsymbol{\omega}\right)=\left(\boldsymbol{0},\boldsymbol{0}\right)$ yields

$$\delta\dot{\boldsymbol{\sigma}} = \frac{1}{4}\delta\boldsymbol{\omega} \tag{4.73}$$

$$\delta\dot{\boldsymbol{\omega}} = -\left(\frac{k_{v}}{2}\boldsymbol{I}+\boldsymbol{K}_{\omega}\boldsymbol{K}_{\sigma}\right)\delta\boldsymbol{\sigma}-\left(\boldsymbol{K}_{\omega}+\frac{1}{4}\boldsymbol{K}_{\sigma}\right)\delta\boldsymbol{\omega} \tag{4.74}$$

Following a similar logic as that discussed in the translational controller section, the gains can be chosen according to a suboptimal H_∞ controller development. First, the linearized rotational system can be written in the form

$$\dot{\boldsymbol{x}}_r = \boldsymbol{A}_r \boldsymbol{x}_r + \boldsymbol{B}_r \left[\boldsymbol{u}_r + \boldsymbol{d}_r\right] \tag{4.75}$$

where $\boldsymbol{x}_r = \begin{bmatrix} \delta\boldsymbol{\sigma} & \delta\boldsymbol{\omega} \end{bmatrix}^T$ is the rotational state, $\boldsymbol{u}_r$ is the angular acceleration input, $\boldsymbol{d}_r$ is a disturbance input, and

$$\boldsymbol{A}_r = \begin{bmatrix} \boldsymbol{0} & \frac{1}{4}\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{B}_r = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \tag{4.76}$$

The H_∞ full information state feedback control law is

$$\boldsymbol{u}_r = -\boldsymbol{B}_r^T \boldsymbol{X}_r \boldsymbol{x}_r \tag{4.77}$$

where $\boldsymbol{X}_r$ is the solution to a Ricatti equation of the same form shown in Eq. (4.57) corresponding to the rotational system. By choosing a diagonal state error weighting matrix, the state feedback gain matrix can be decomposed into

$$\boldsymbol{B}_r^T \boldsymbol{X}_r = \begin{bmatrix} \left(\frac{k_v}{2}\boldsymbol{I} + \boldsymbol{K}_\omega \boldsymbol{K}_\sigma\right) & \left(\boldsymbol{K}_\omega + \frac{1}{4}\boldsymbol{K}_\sigma\right) \end{bmatrix} \tag{4.78}$$

Note that this approach leaves three unknowns, namely the gains k_v, $\boldsymbol{K}_\sigma$, and $\boldsymbol{K}_\omega$, with only two equations. The implication is that families of solutions can be obtained for given values of one parameter. In this work, it is convenient to specify the gain k_v and then solve for $\boldsymbol{K}_\sigma$ and $\boldsymbol{K}_\omega$ from Eq. (4.78).

Chapter 5

Simulation Results

This chapter discusses simulation results of the robust and adaptive filtering techniques introduced in Chapter 3. These results include radar tracking of an entry vehicle, adaptive estimation illustrated using the multiplicative quaternion filtering technique, with both Gaussian and non–Gaussian noise, attitude filtering using the modified Rodrigues parameters, and finally, the full 6-DOF rendezvous navigation simulation.

5.1 Radar Tracking Problem

This section discusses the application of the robust filters to the problem of estimating the trajectory of a target using range measurements recorded from a radar tracking station. The example problem is to estimate the trajectory of a mass falling through an exponential atmosphere with a constant, yet unknown, drag coefficient. The gravitational acceleration acting on the body is neglected in the dynamic model. This truncated model is valid for high initial velocities that cause the aerodynamic acceleration to dominate over the gravitational acceleration. This benchmark nonlinear filtering problem was initially studied in Ref. 131 and has been repeated numerous times in the literature. This problem is especially of interest for non–Gaussian estimation problems given the known non–Gaussian nature of radar noise, including for instance that of glint.[60, 132] Fig. 5.1 shows the geometry of the problem.

5.1.1 Dynamic Model and Measurement Equations

The dynamic model for the radar tracking problem described above can be derived by writing

$$\dot{x}_1 = -x_2 \tag{5.1}$$

$$\dot{x}_2 = -\frac{C_D A \rho}{2m} x_2^2 \tag{5.2}$$

where x_1 represents the altitude of the mass, x_2 its downward velocity, C_D is the drag coefficient, A is the cross-sectional area, m is the mass, and ρ is the atmospheric density, which is assumed to follow an exponential function of the form

$$\rho = \rho_0 \mathrm{e}^{-\eta x_1} \tag{5.3}$$

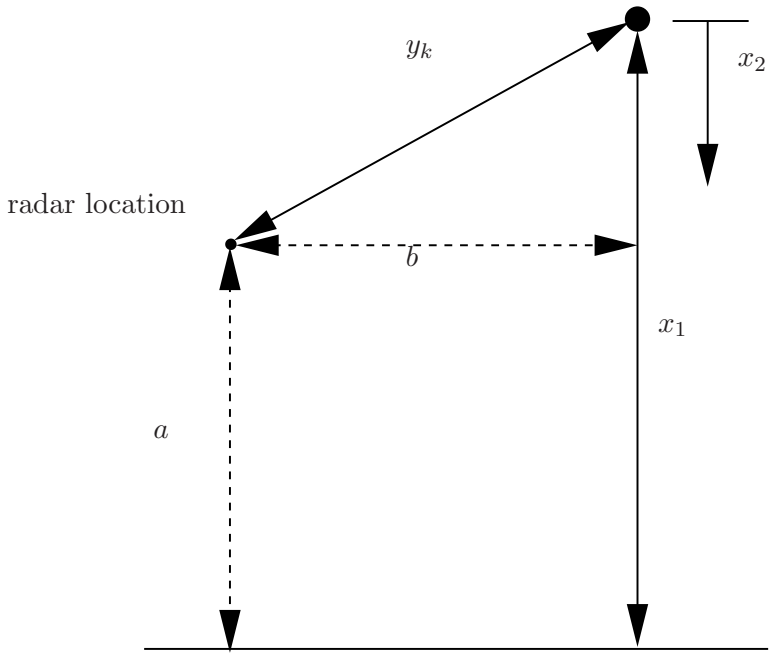


Figure 5.1: Geometry of Example Problem

where the quantity η is the known constant inverse density scale height. If the aerodynamics of the target are considered to be an unknown parameter to be estimated in real time, then a third state variable can be expressed as

$$x_3 = \sqrt{\frac{C_D A \rho_0}{2m}} \tag{5.4}$$

Here, the square-root of $C_D A \rho_0 / 2m$ is used since this term must always be positive, in other words since the quantity x_3^2 is always non-negative.

In summary, the complete dynamic model for this problem is expressed as

$$\dot{x}_1 = -x_2 \tag{5.5}$$
$$\dot{x}_2 = -x_3^2 x_2^2 \mathrm{e}^{-\eta x_1} \tag{5.6}$$
$$\dot{x}_3 = 0 \tag{5.7}$$

The radar measurement equation is

$$y_k = \sqrt{b^2 + \left[x_1\left(t_k\right) - a\right]^2} + w_k \tag{5.8}$$

where w_k represents zero-mean random error, with probability density function $f(w_k)$. Random measurement errors are drawn from the mixture of zero–mean Gaussian probability distributions, defined by the probability density function

$$f\left(w_k\right) = \left(\frac{1-\epsilon}{\sigma_1\sqrt{2\pi}}\right) \exp\left[-\left(\frac{w_k^2}{2\sigma_1^2}\right)\right] + \left(\frac{\epsilon}{\sigma_2\sqrt{2\pi}}\right) \exp\left[-\left(\frac{w_k^2}{2\sigma_2^2}\right)\right] \tag{5.9}$$

Table 5.1: Simulation Parameters

Parameter	Value
a, km	30.5
b, km	30.5
η, m^{-1}	$1.64 \cdot 10^{-4}$
σ_1, m	30.5
γ	1.345
c^2	3.0

Table 5.2: Initial Conditions

Initial State	True Value	Estimated Value	Standard Deviation
$x_1(0)$, km	91.5	91.5	0.31
$x_2(0)$, km/s	6.1	6.1	0.06
$x_3(0)$, $1/\sqrt{\mathrm{m}}$	0.06	0.01	0.02

where σ_1 and σ_2 are the standard deviations of the individual Gaussian distributions, and ϵ is a perturbing parameter that represents error model contamination. The standard deviations σ_1 are chosen according to Table 5.1 and σ_2 is chosen as $\sigma_2 = 5\sigma_1$. The measurements are assumed to occur at a frequency of 1 Hz. The model parameters and initial conditions for the problem are summarized in Table 5.1. The true trajectory for these initial conditions is shown in Fig. 5.2.

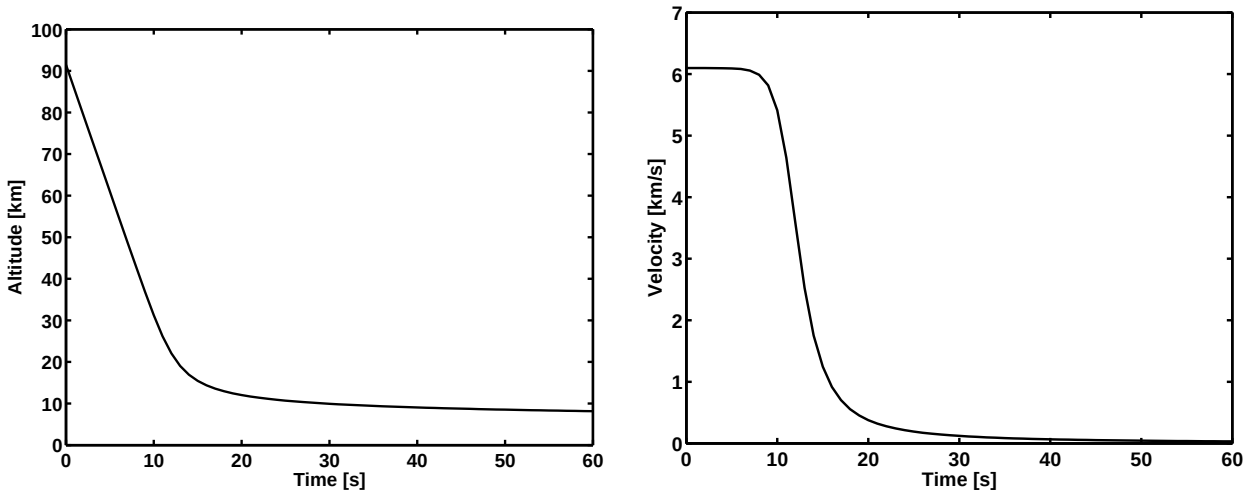
5.1.2 Results of Example Problem: Perfect Noise Case

This section discusses the results of applying several filters to the benchmark tracking problem with both Gaussian and non–Gaussian measurement noise cases. These filters include the EKF, DD1, DD2, and the robust versions of each using the one–step Huber update. The results of a Monte–Carlo simulation are shown in the following figures. In this simulation, 2000 trial cases have been conducted, each case terminating after an elapsed time of 60 s. The results presented in this subsection are for the case with perfect knowledge of the measurement error variance σ_1^2 with no adaptive tuning.

5.1.2.1 Comparison of Leverage Point Identification Methods

This section discusses the impact of the leverage point identification methods on the state estimates using the Huber discrete–time filtering technique. Fig. 5.3 shows a comparison of the average state estimate errors for the case $\epsilon = 0$, and Fig. 5.4 shows the average state estimate errors for the case $\epsilon = 0.5$. In these plots, the estimation errors found by using the projection statistics are shown in the solid curve, the errors found by using the Mahalanobis distances are shown in the dashed curve, and estimation errors without using leverage point identification are shown in the dash-dot curve.

These results show that the leverage point identification does not have a significant impact on the state estimation accuracy for this problem. In the cases of the Mahalanobis distances,



(a) True Position (b) True Velocity

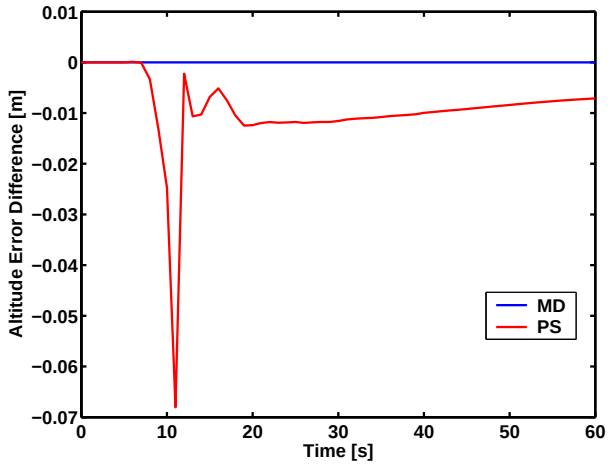
Figure 5.2: True Trajectory of Target

the results are exactly equal to the cases that do not use a leverage point identification method. The cases using projection statistics leads to an increased overall estimation accuracy. In each case, the leverage point identification using the Mahalanobis distances costs approximately 1% more computation than the case without a leverage point identification, whereas the case using projection statistics costs approximately 5% more computation. In this application, the benefits of the use of projection statistics for detecting and weighting the leverage points does not seem worth the cost in computation. In problems of higher dimension, however, it is anticipated that the benefits of the use of projection statistics will increase.

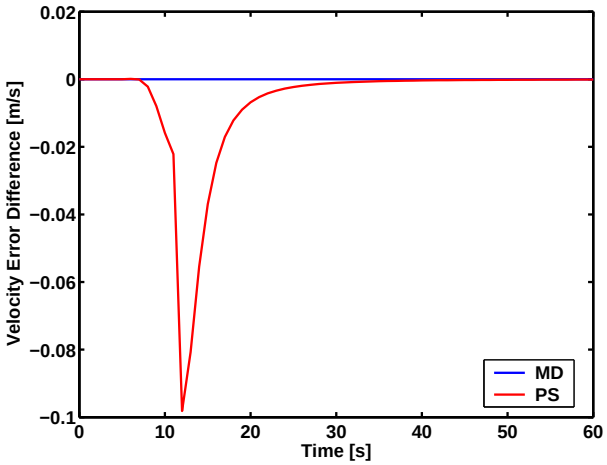
5.1.2.2 Comparison of Filters

The absolute value of the median errors are shown for the case of $\epsilon = 0$ in Figs. 5.5(a)–(c). In this case the DD2 filter gives a smaller estimation error than the other robust and nonrobust filters. The DD1 filter exhibits slightly larger errors than the EKF. These results are not surprising based on the results given in Ref. 79 and 80. The robust filters do not perform as well as their non-robust counterparts in this case, because the Huber update does not minimize the ℓ_2 norm during the measurement update. The increase in the estimation error for the robust filters is to be expected in a perfectly Gaussian simulation since the minimum ℓ_2 norm is the maximum likelihood estimator in this case.

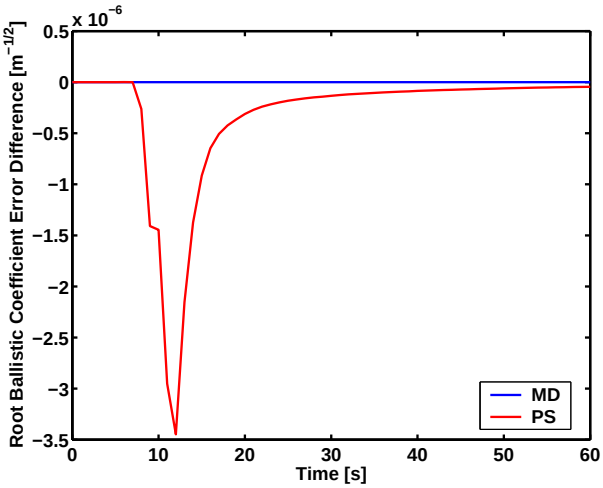
The absolute value of the median errors are shown in Figs. 5.5(d)–(f) for the case $\epsilon = 0.5$, for the case where the measurement errors are highly non–Gaussian. In this case, the Huber–EKF and DD2 filters give comparable results to each other for the position and velocity errors, but the Huber–EKF gives a smaller error in the estimate of the ballistic parameter. Both Huber–EKF and DD2 are superior to the EKF and DD1 filter in this case. The Huber–DD2 filter exhibits the smallest errors, because it captures both nonlinearity and non-Gaussianity. The DD1 and Huber–DD1 filters do not perform as well as the EKF and the Huber–EKF, respectively, in the non–Gaussian case, which follows the behavior from the Gaussian case. In the non-Gaussian case,



(a) Estimated Position Errors



(b) Estimated Velocity Errors

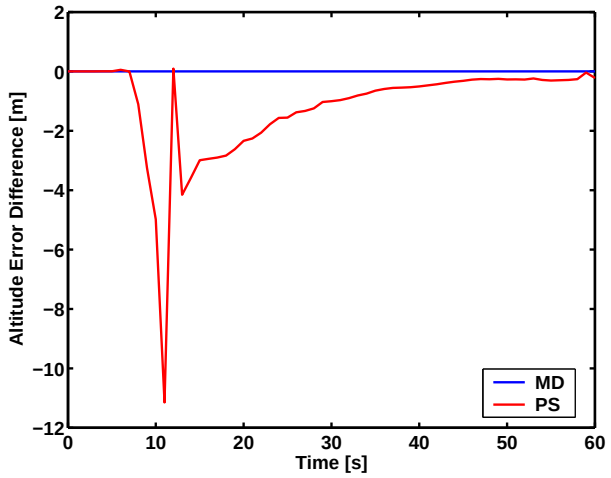


(c) Estimated Ballistic Coefficient Errors

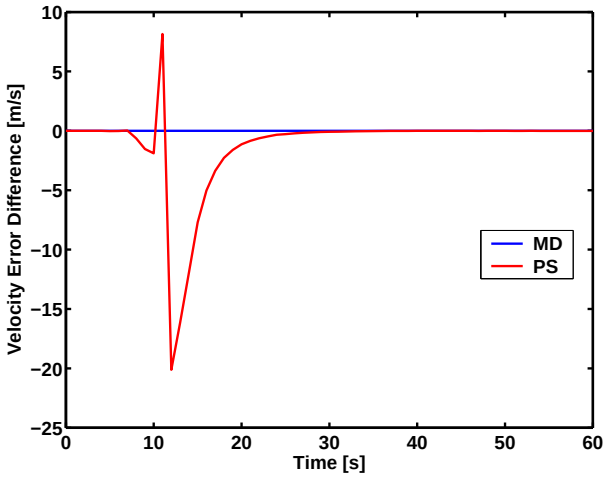
Figure 5.3: Estimated Trajectory Errors for $\epsilon = 0.0$: Comparison of Leverage Point Identification Methods

six cases of the EKF and two cases of the DD1 methods diverged completely, while the robust counterparts, Huber-EKF and Huber-DD1 did not exhibit divergence. Divergence problems with the Kalman filter have been noted previously.[133] In this case, the ballistic parameter initialization is poor, which is compounded by the presence of non-Gaussian errors. The DD2 filter, being a second–order filter and therefore not as sensitive to initialization, did not diverge in any of the Monte-Carlo cases, but the median error is clearly reduced by making use of the Huber update method.

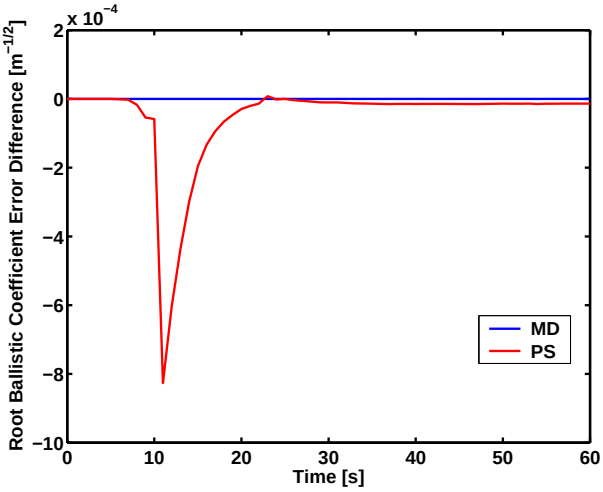
The computational cost associated with implementing the each filter is summarized in Table 5.3. In this table, the computational costs are divided by the EKF processing time to provide a ratio of the cost relative to that associated with the EKF. These ratios are based on the median computation time for each filter during the Monte-Carlo simulation. The results show that the Huber–EKF filter requires the smallest relative computational cost whereas the DD2 and Huber–DD2 filters require a



(a) Estimated Position Errors



(b) Estimated Velocity Errors

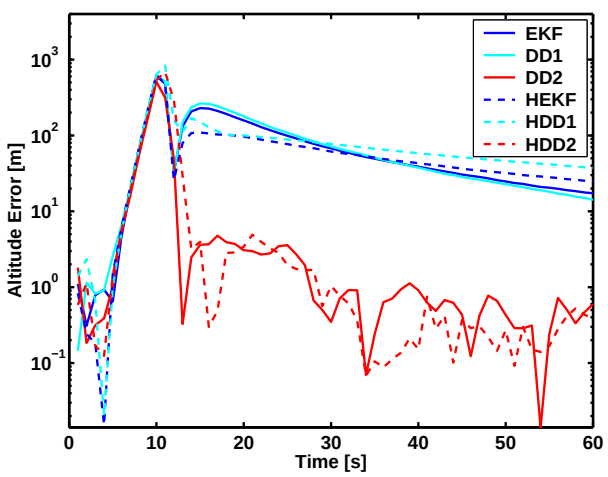


(c) Estimated Ballistic Coefficient Errors

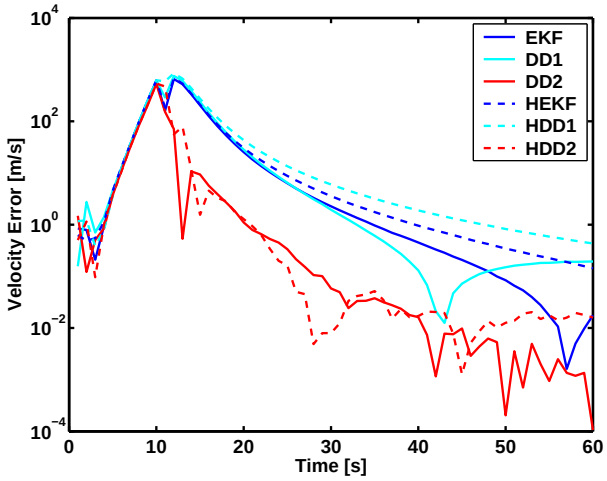
Figure 5.4: Estimated Trajectory Errors for $\epsilon = 0.5$: Comparison of Leverage Point Identification Methods

more: 3.12 and 3.15 times the total computation, respectively. It is not surprising that the Huber–DD2 filter has the largest cost, because it has the smallest errors for the non-Gaussian case, but it is interesting to note that the similar levels of accuracy in the non-Gaussian case can be found by use of the Huber–EKF filter over that of DD2 filter, for only a fraction of the computation time.

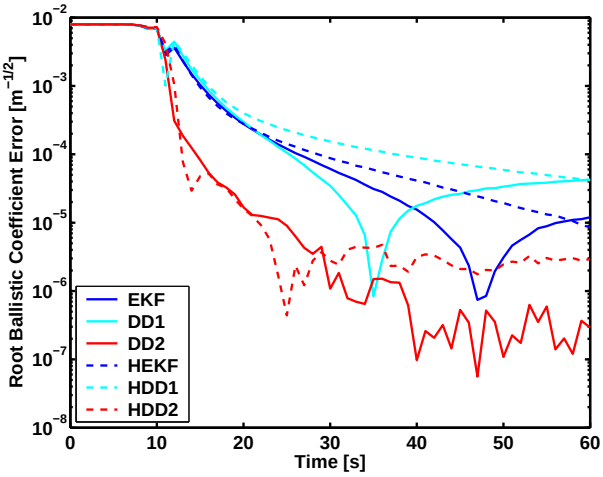
The results show that for perfectly Gaussian error distributions the standard DD2 filter exhibits the lowest estimation error time history, which was the expected outcome based on previously published results. The DD1 filter exhibited slightly larger estimation errors than that of the EKF. The filters with the Huber update technique produced larger errors than the standard update, since the standard form of the update is a the maximum likelihood estimate for the perfectly Gaussian case. However, for non–Gaussian error distributions, the modified filters with the Huber update outperformed the standard filters. The modified DD2 filter with the Huber update equation



(a) Estimated Position Errors for $\epsilon = 0.0$



(b) Estimated Velocity Errors for $\epsilon = 0.0$



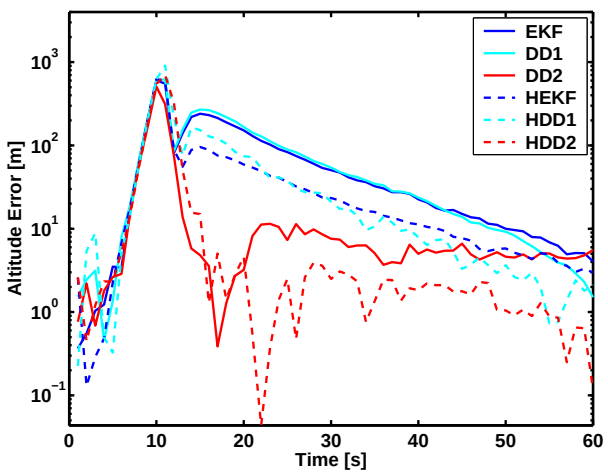
(c) Estimated Ballistic Coefficient Errors for $\epsilon = 0.0$

Figure 5.5: Estimated Trajectory Errors for $\epsilon = 0$

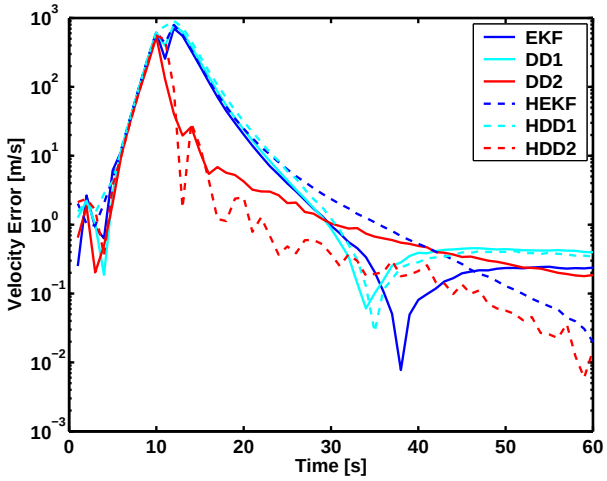
Table 5.3: EKF-Relative Computation Ratios

Filter	Computation Ratio
H-EKF	1.08
DD1	2.95
H-DD1	3.14
DD2	3.02
H-DD2	3.19

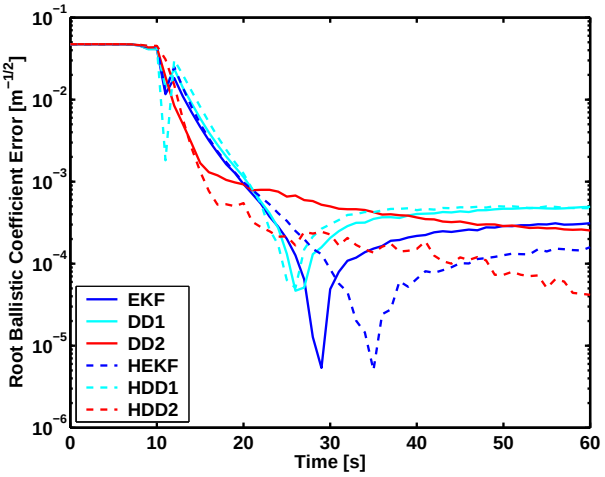
exhibited the smallest errors in the non-Gaussian numerical simulations conducted. The Huber-EKF and the Huber-DD1 filters were able to mitigate divergence problems in their non-robust counterparts.



(a) Estimated Position Errors for $\epsilon = 0.5$



(b) Estimated Velocity Errors for $\epsilon = 0.5$



(c) Estimated Ballistic Coefficient Errors for $\epsilon = 0.5$

Figure 5.6: Estimated Trajectory Errors for $\epsilon = 0.5$

Comparisons of the computational costs associated with each filter show that the Huber–EKF filter is able to process data at a rate approximately three times faster than the standard DD2 filter, and produces similar accuracy levels in the non–Gaussian case. Therefore, for non-Gaussian cases, the Huber–EKF filter is superior to the standard DD2 filter. If computation costs are not a concern for the particular application, then the Huber–DD2 filter exhibits the best performance.

5.2 Adaptive Multiplicative Quaternion Filtering

This section describes the results of the application of the filtering techniques discussed in previous sections to the spacecraft attitude estimation problem using gyroscopes and star trackers. The standard Kalman filter, Myers–Tapley adaptive Kalman filter, Huber filter, and the proposed modified Myers–Tapley adaptive Huber filter are applied to the attitude estimation problem. The

example includes both Gaussian and non–Gaussian noise densities with unknown statistics.

In this example, the true spacecraft angular velocity is $\boldsymbol{\omega} = [\ \omega_0 \quad \omega_0 \quad \omega_0\ ]^T$, for $\omega_0 = 10^{-3}$ rad/s. The angular velocity is sampled at a rate of 10 Hz, with star tracker updates at a rate of 1 Hz. Noise samples for the gyroscope and star tracker measurements are drawn from a mixture model with a probability density function of the form

$$f(w) = \frac{1-\epsilon}{\sqrt{2\pi}} \exp\left(-\frac{w^2}{2}\right) + \frac{\epsilon}{2b} \exp\left(-\frac{|w|}{b}\right) \tag{5.10}$$

where $b = 5/\sqrt{2}$.

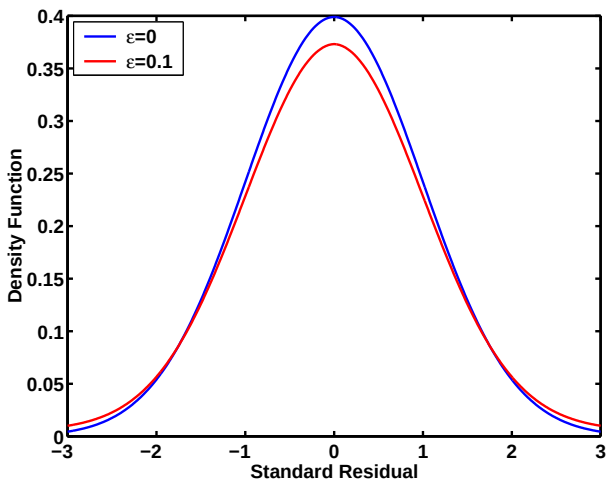
Note that this density is a mixture of a nominal Gaussian with a Laplacian contaminating density. A comparison of the probability density and distribution functions for the cases $\epsilon = 0$ (no contamination) and $\epsilon = 0.1$ (with 10 % contamination) is shown in Fig. 5.7. A comparison of pure Gaussian with the contaminated Gaussian densities is shown in Fig. 5.7(a), the distribution functions are shown in Fig. 5.7(b), and a quantile–quantile plot (or QQ plot) in Fig. 5.7(c). From the density function and distribution function plots it is not entirely obvious that the contaminating density is in fact non-Gaussian; it appears to have Gaussian characteristics but with a larger variance. The differences between the two distributions is much more apparent in the QQ plot in Fig. 5.7(c). A QQ plot is a plot of the scaled ordered data against the quantiles of a comparison distribution. Since the data are scaled prior to creating the plot, the result is independent of the variance, rendering only the shape of the distribution to be important. A linear result with unit slope indicates that the data follows the same distribution as the comparison distribution (the Gaussian, in this case). The QQ plot of the contaminated Gaussian indicates that the data follows a roughly Gaussian distribution in the middle ranges but with a greater thickness in the extremes. This result indicates that the contaminated distribution has a higher probability of generating extreme points.

The sensor uncertainty specifications are $\sigma_\omega^2 = 2.25 \cdot 10^{-13}$ rad^2/s, $\sigma_\beta^2 = 2.25 \cdot 10^{-19}$ rad^2/s^3, and $\sigma_s^2 = 3.81 \cdot 10^{-8}$ rad^2. It is assumed that the known sensor uncertainties differ from the true sensor uncertainties by a factor of 50% of the standard deviation. In addition, the star tracker measurement errors are initially assumed to be uncorrelated in each axis whereas the true errors are correlated with a correlation coefficient of 0.5 in each axis.

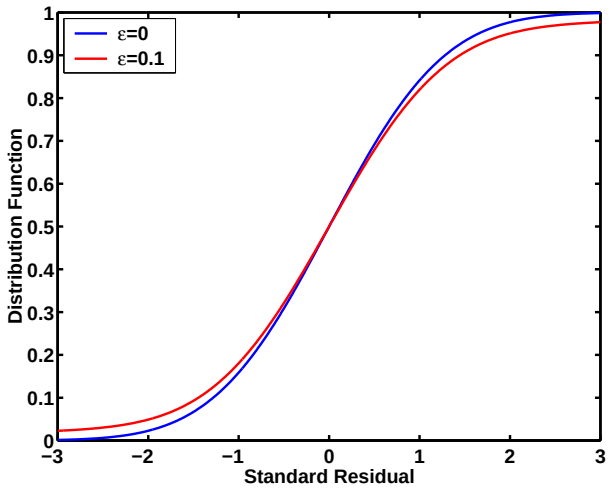
The Huber filter and adaptive Huber filter use an initial value for the tuning parameter of $\gamma = 1.5$. The tuning parameter is specified to be bounded such that $1 \leq \gamma \leq 2$ within the adaptive Huber filter. The adaptive filtering techniques are implemented in such a way that they begin estimating the noise statistics after 25 samples and store up to a specified maximum threshold of observations. This maximum threshold is varied in later sections to show the improvements that can be achieved through larger samples. The adaptive filtering techniques use a value $k_f = 0.9$ to smooth the covariance and contamination parameters estimates.

5.2.1 Gaussian Simulation

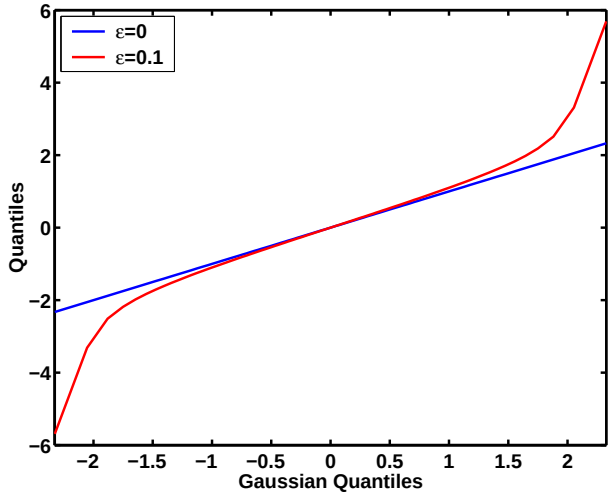
This section describes the results of a 2000 case Monte-Carlo simulation involving pure Gaussian random errors ($\epsilon = 0$) with the specified noise variances discussed in the previous section. The adaptive filtering techniques make use of a maximum buffer size of 100 observations in forming the estimates of the process and measurement noise covariance matrices.



(a) Density Functions



(b) Distribution Functions

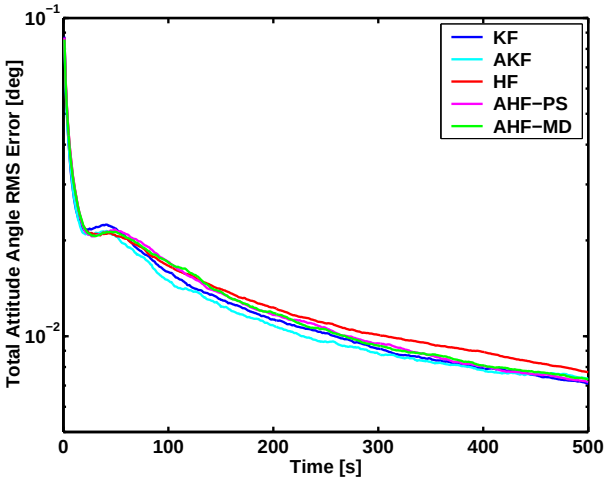


(c) Quantile–Quantile Plot

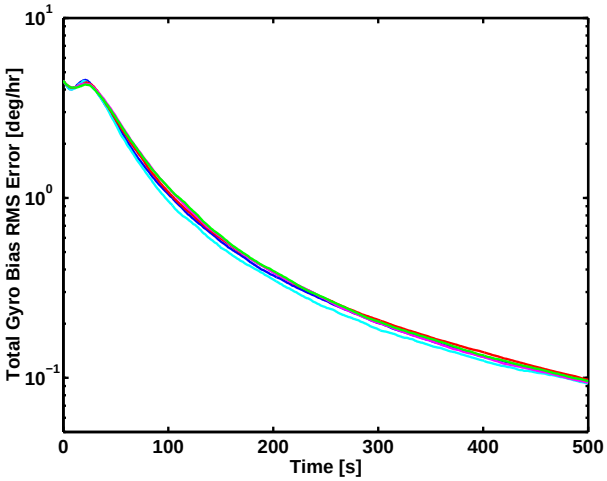
Figure 5.7: Comparison of Probability Density and Distribution Functions for $\epsilon = 0$ and $\epsilon = 0.1$

The results in the form of root mean square (RMS) errors are shown in Fig. 5.8. The total attitude angle RMS error is shown in Fig. 5.8(a), the vector norm of the gyroscope bias RMS error is shown in Fig. 5.8(b). The error in the RMS error prediction of the total attitude angle is shown in Fig. 5.8(c), and that of the gyroscope bias is shown in Fig. 5.8(d). The (1,1) component of the star tracker variance estimate RMS error is shown in Fig. 5.8(e), and the RMS error of the contamination parameter estimate is shown in Fig. 5.8(f). In each case, the Kalman filter results are shown with the blue curve, the adaptive Kalman filter results are shown with the cyan curve, and the Huber filter results are shown with the red curve. For the adaptive Huber filter cases, the filter based on the projection statistics is shown with the magenta curve while that based on the Mahalanobis distances is shown with the green curve.

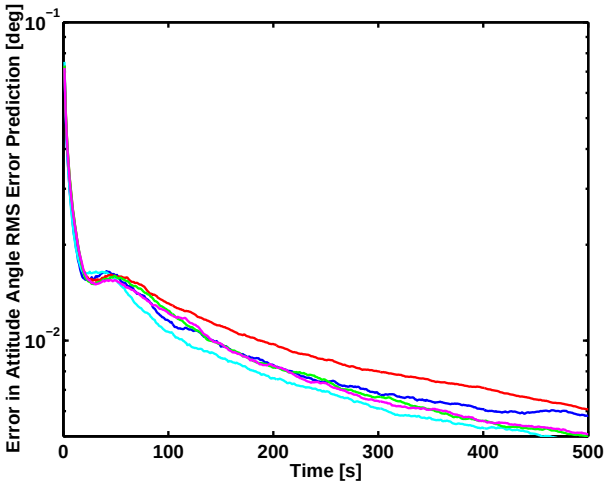
In this case, the attitude angle errors are not significantly different between the various tech-



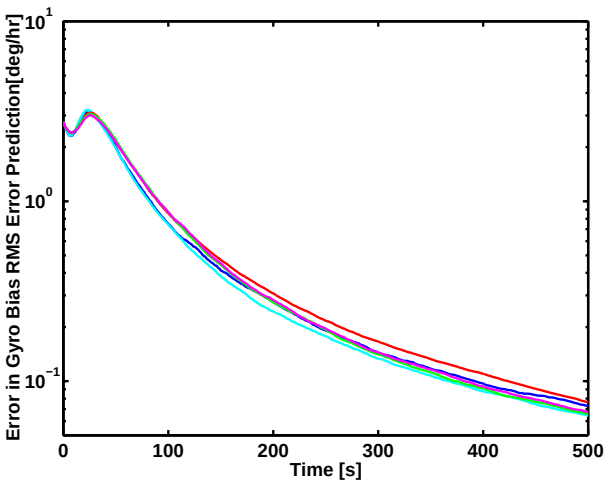
(a) Attitude Angle RMS Error



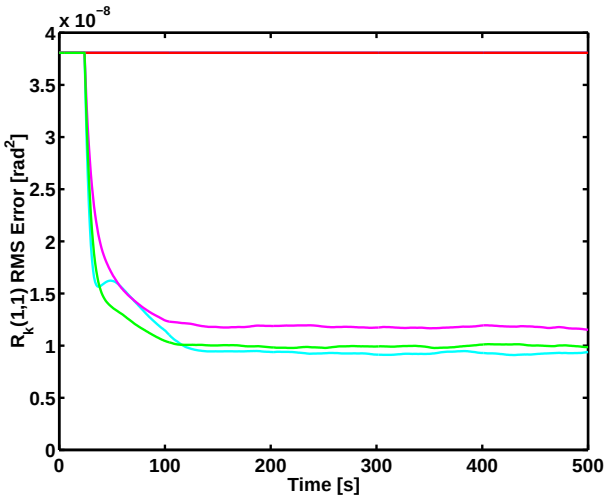
(b) Gyro Bias RMS Error



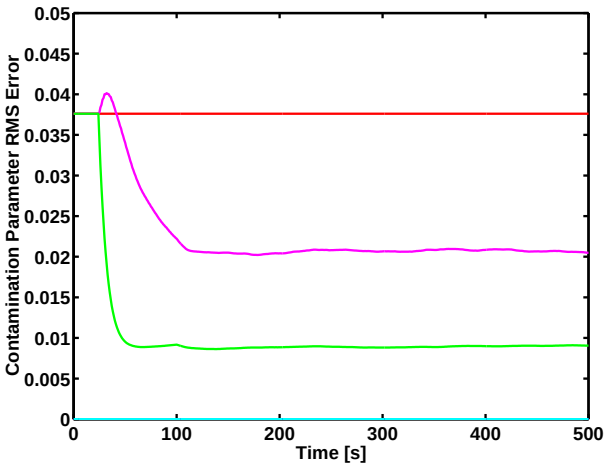
(c) Error in Attitude Angle RMS Error Prediction



(d) Error in Gyro Bias RMS Error Prediction



(e) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error



(f) Contamination Parameter RMS Error

Figure 5.8: Gaussian Simulation RMS Errors

niques even though the measurement statistics assumed by the filters are erroneous. The Kalman and adaptive Kalman filter provide similar performance, slightly superior to the Huber approaches, which is to be expected since the errors are perfectly Gaussian. The adaptive Huber filters show slightly better performance than the non–adaptive case. The adaptive Huber filter based on the Mahalanobis distances performs better than that based on the projection statistics, which is to be expected since the sample mean and covariance used in computing the Mahalanobis distances are maximum likelihood estimates for perfectly Gaussian problems. The same trends and relative performance between filters is shown in the gyroscope bias RMS error.

The comparison of the error predictions in Figs. 5.8(c) and (d) show that the adaptive Kalman filter outperforms the other filters for both the attitude error prediction and rate bias error prediction. The predicted error estimates of the Huber–based adaptive filters are similar, and superior to the non–adaptive Huber and Kalman filters. The improved estimates of the attitude and bias errors for the adaptive filters is to be expected since these filters attain an improved knowledge of the measurement and process noise covariances. Naturally, the adaptive Kalman filter is superior since this problem is Gaussian.

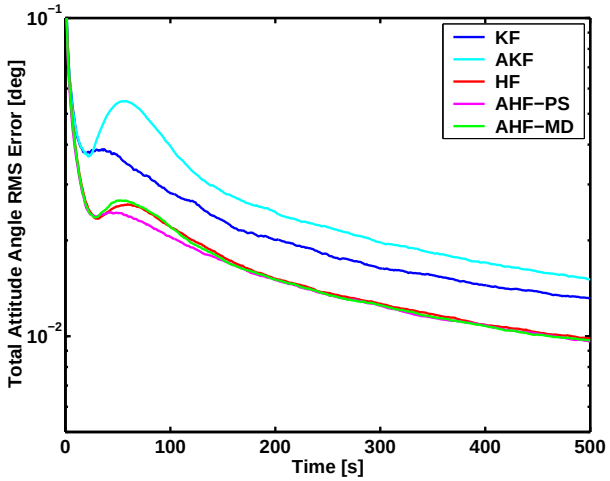
The $\tilde{\boldsymbol{R}}_k(1,1)$ RMS error results in Fig. 5.8(e) show that the adaptive filters are able to reduce the error in the assumed measurement statistics. The adaptive Kalman filter shows superior performance than the adaptive Huber filter for this case, which is expected since the errors are purely Gaussian in nature. For the same reason, the adaptive Huber filter based on the Mahalanobis distances is superior to that based on the projection statistics. In passing it should be mentioned that the other components of the measurement and process noise covariance estimates exhibit the same behavior as the $\tilde{\boldsymbol{R}}_k(1,1)$ estimate, and therefore only this one particular quantity is shown as an example.

Finally, the contamination parameter RMS error shows that the adaptive Huber filter is able to reduce the error in the assumed contamination parameter by means of the formula based on the robust weighting parameters in Eq. 3.90. Note that the adaptive Huber filter is the only filter investigated in this dissertation that can estimate the contamination parameter, which is evident from the fact that the other filter results show a constant RMS error. In this case, both the Kalman and adaptive Kalman filter show no error in the contamination parameter estimate, as these filters implicitly assume the errors are Gaussian and this simulation was conducted with only Gaussian errors. The Huber filter shows a constant RMS error at the value of the contamination parameter corresponding to the tuning parameter $\gamma = 1.5$. Due to Gaussianity, the adaptive Huber filter based on the Mahalanobis distances is superior to the filter based on the projection statistics.

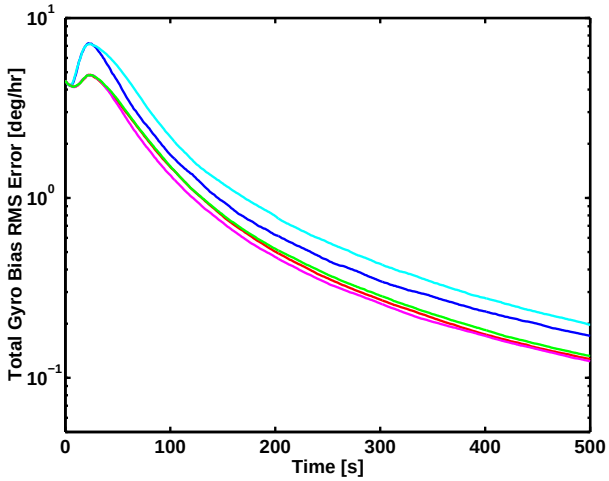
5.2.2 Non–Gaussian Simulation

This section describes the results of a 2000 case Monte-Carlo simulation involving non–Gaussian errors, specifically for the mixture model in Eq. (5.10) with $\epsilon = 0.1$. The RMS error results corresponding to this simulation are shown in Fig. 5.9. The adaptive filters again use a buffer size of $N = 100$ stored residuals in order to compute the measurement and process noise covariance matrices.

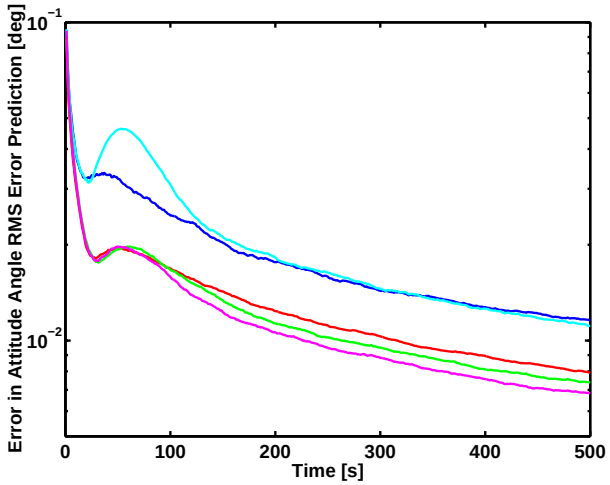
For this case, the total attitude RMS errors clearly show the superiority of the Huber–based filtering methods. The adaptive Huber method exhibits the smallest RMS error, followed by the non–adaptive Huber method. The projection statistics-based adaptive Huber filter is superior



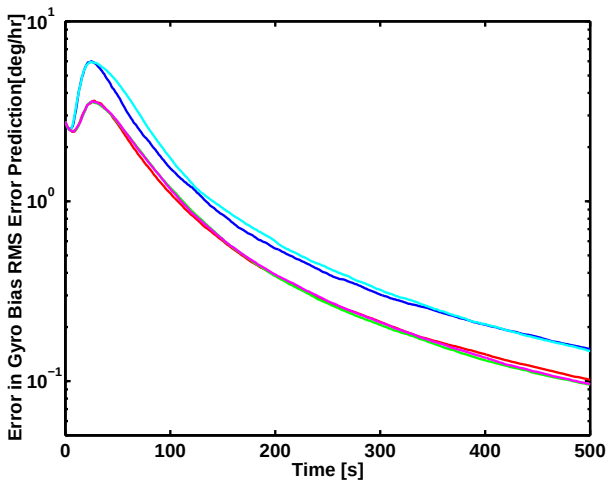
(a) Attitude Angle RMS Error



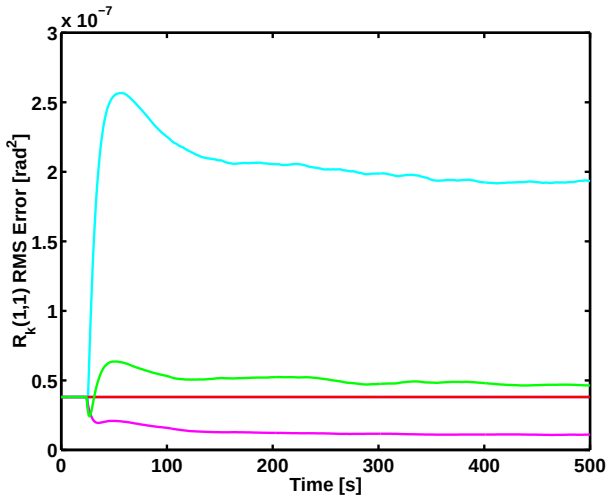
(b) Gyro Bias RMS Error



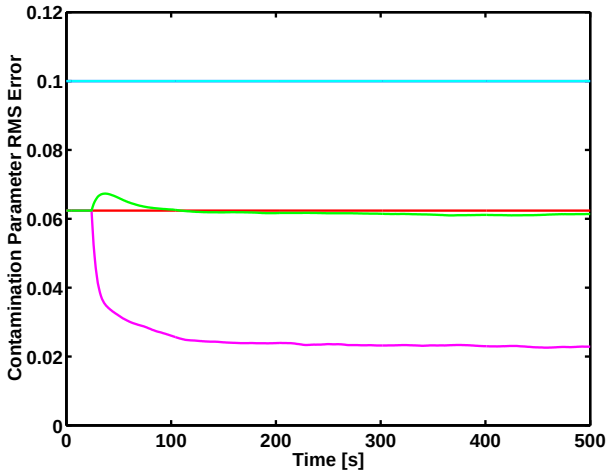
(c) Error in Attitude Angle RMS Error Prediction



(d) Error in Gyro Bias RMS Error Prediction



(e) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error



(f) Contamination Parameter RMS Error

Figure 5.9: Contaminated Gaussian Simulation RMS Errors

Table 5.4: Comparison of Identification Techniques for Clustered Outliers

Point	$\mathcal{M}_i$	$\mathcal{P}_i$
A	24.07	166.53
B	22.15	166.96
C	18.22	126.07

to the filter based on the Mahalanobis distances, due to the non–Gaussianity in this example. Interestingly, the standard, non–adaptive Kalman filter exhibits smaller errors than that of the adaptive Kalman filter. This behavior is explored in more detail below. As with the previous case, the same trends seen in the attitude angle error appear in the gyroscope bias estimate RMS error, as expected.

The predicted attitude and bias estimate errors show that the adaptive Huber–based filters are better predictors of the estimation errors than the other filters for the non–Gaussian case. As expected, the filter based on the projection statistics performs the best for both the attitude and bias error prediction. Of the non–adaptive filters, the Huber filter exhibits a better error prediction than the Kalman filter.

The RMS error in the $\tilde{\boldsymbol{R}}_k(1,1)$ estimate is quite interesting. Here, the adaptive Huber technique based on the projection statistics is able to reduce the error associated with the assumed measurement noise covariance, whereas the adaptive Huber filter using the Mahalanobis distances and the adaptive Kalman technique exhibit an *increased* error. This increase in the measurement noise error is the cause of the increased error in the attitude angle estimate as previously noted. Essentially, the adaptive Kalman technique associates the contaminated measurements with an increase in the measurement covariance, therefore causing the assumed covariance to inflate in order to account for the contamination. This covariance inflation is known to cause an increase in the estimation error due to the fact that *all* measurements are processed as if they were outliers, receiving less weight, than a technique that is able to distinguish outlying measurements and weight them accordingly.[20] Likewise, the adaptive Huber technique using the Mahalanobis distances suffers from the inherent limitations of the sample mean and covariance when applied to non-Gaussian problems.

As discussed in the previous section, the only technique described in this dissertation to adaptively estimate the contamination parameter associated with the noise densities are the adaptive Huber-based filters. In the non–Gaussian case, the adaptive Huber filter is successfully able to the reduce the error in the assumed contamination parameter. However, the results of the filter based on the projection statistics is far superior to that of the Mahalanobis distances. Essentially, the filter based on the Mahalanobis distances attempts to find a compromise between the crude covariance inflation of the adaptive Kalman filter and the Huber filter. The result is that the contamination parameter estimate error does not improve substantially over that of the non–adaptive Huber filter, and the covariance estimate inflates slightly but not to the extent of the adaptive Kalman filter. All other filters assume (explicitly or implicitly) a constant contamination parameter. In the case of the adaptive and non–adaptive Kalman filters, the contamination is assumed to be zero, thus the error is constant at 0.1 for this case. The Huber filter error again corresponds to the assumption leading to the tuning parameter $\gamma = 1.5$.

Figure 5.10 shows a scatter plot of the stored measurement residuals from the adaptive Huber

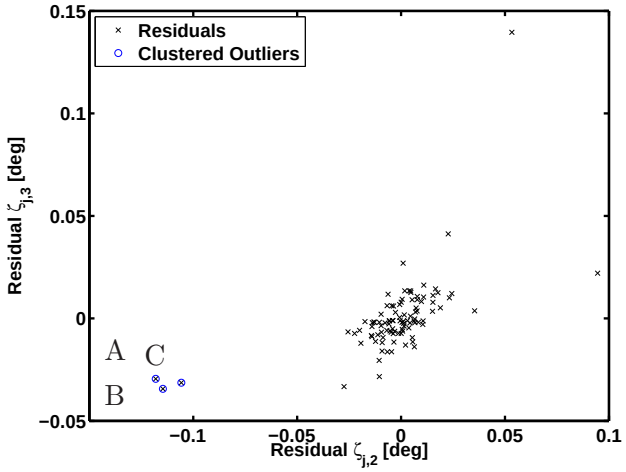


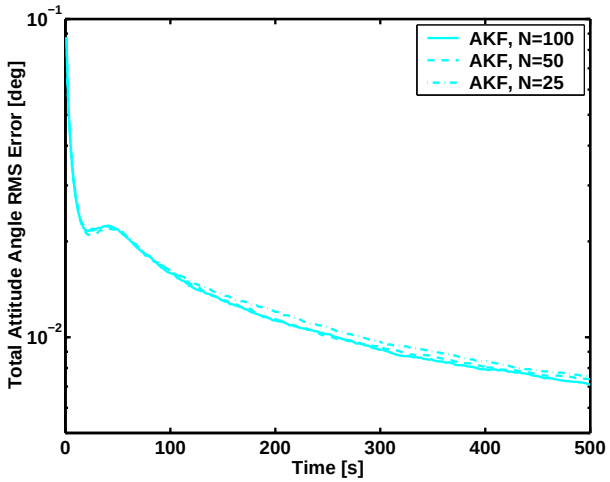
Figure 5.10: Example of Clustered Outliers in Adaptive Filter

technique, from an arbitrary stopping point in a non-Gaussian simulation, projected into the (2,3) plane. This sample includes 100 measurement residuals, with several apparent outliers. In particular, a group of 3 outliers appears in a cluster as indicated in the plot. The Mahalanobis Distances and Projection Statistics have been computed for this set of residuals, and the values corresponding to the clustered outliers is shown in Table 5.4. Compared with the $\chi^2_{3,0.95} = 7.81$ threshold, both methods are able to detect the clustered outliers in this example. However, the Projection Statistics is not influenced by the clustering and assigns these point much smaller weights than the Mahalanobis Distances. The Projection Statistics provide a better estimate of the contamination parameter, giving a value of 0.104, opposed to the estimate based on the Mahalanobis Distances of 0.051, for this set of residuals.

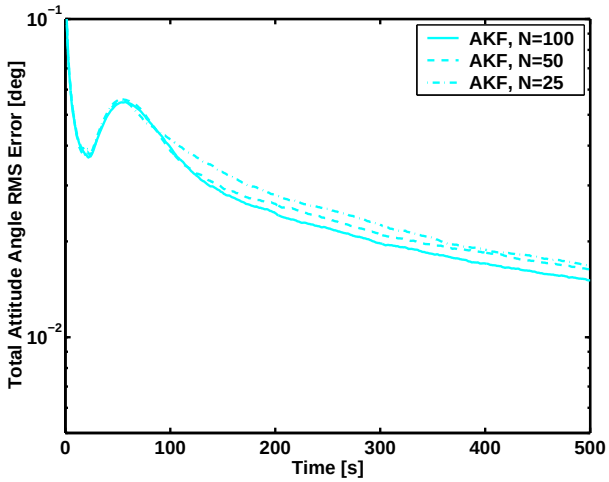
5.2.3 Sensitivity to the Adaptive Filter Buffer Size

The previous two sections have discussed the application of the various filtering techniques to both Gaussian and non–Gaussian cases, using a maximum buffer size of 100 stored residuals for the adaptive filters. This section explores the trade off in performance found by using a smaller buffer size, with the expectation that processing a smaller number of stored residuals reduces the computational costs of the filter. Additional 2000 case Monte–Carlo studies were conducted for $N = 50$ and $N = 25$ for the cases $\epsilon = 0$ and $\epsilon = 0.1$.

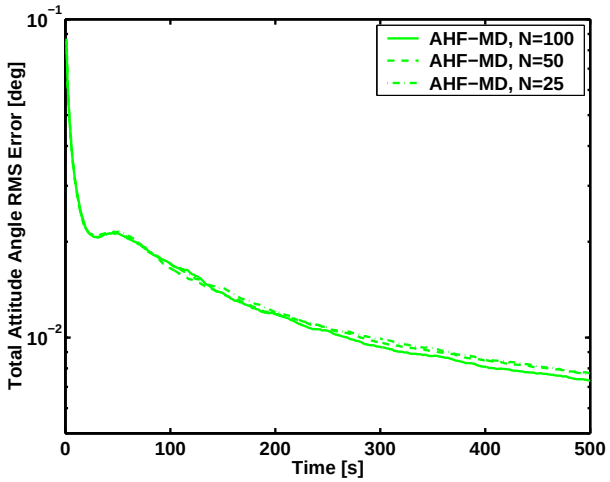
Figure 5.11 shows the results of the attitude angle RMS error for each adaptive filter investigated for this problem. Each subplot shows the result of a particular filter for a range of N for the given value of ϵ. For instance, Fig. 5.11(a) shows results for the adaptive Kalman filter for a range of N and for $\epsilon = 0$. These results show that, for this particular problem, the buffer size can be made as low as $N = 25$ without severe degradation in the filter performance for the attitude error estimates. It is also interesting to note that the scales are all the same in each subplot in Fig. 5.11, and the sensitivity of the adaptive Kalman filter can clearly been detected for the $\epsilon = 0.1$ case compared with the $\epsilon = 0.0$ results. Likewise, the insensitivity of the Huber–based approaches is readily apparent.



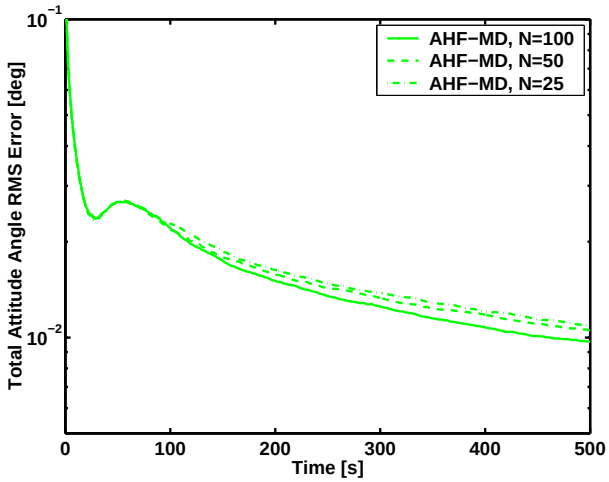
(a) Attitude Angle RMS Error: AKF ($\epsilon = 0$)



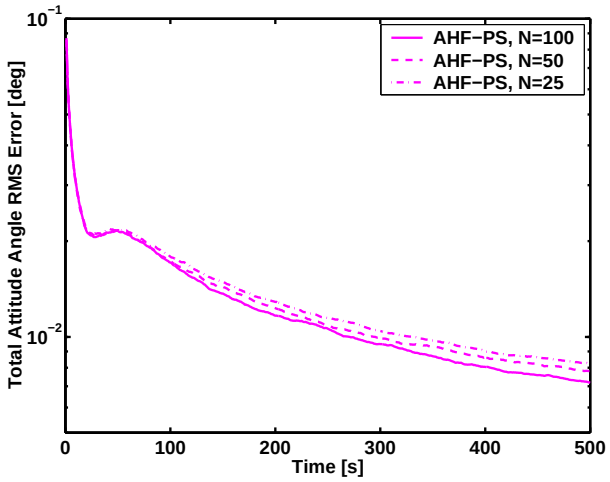
(b) Attitude Angle RMS Error: AKF ($\epsilon = 0.1$)



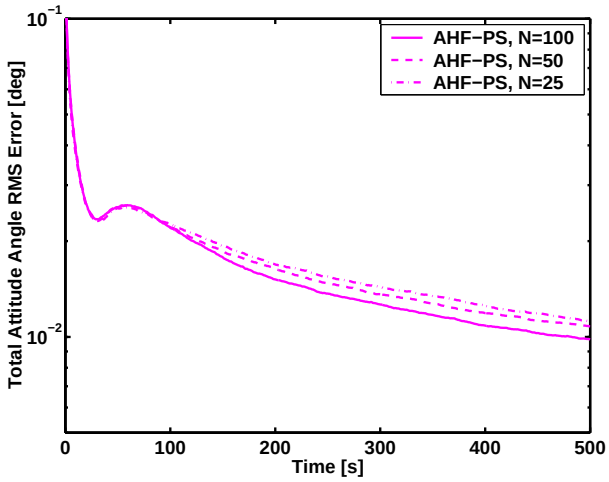
(c) Attitude Angle RMS Error: AHF–MD ($\epsilon = 0$)



(d) Attitude Angle RMS Error: AHF–MD ($\epsilon = 0.1$)

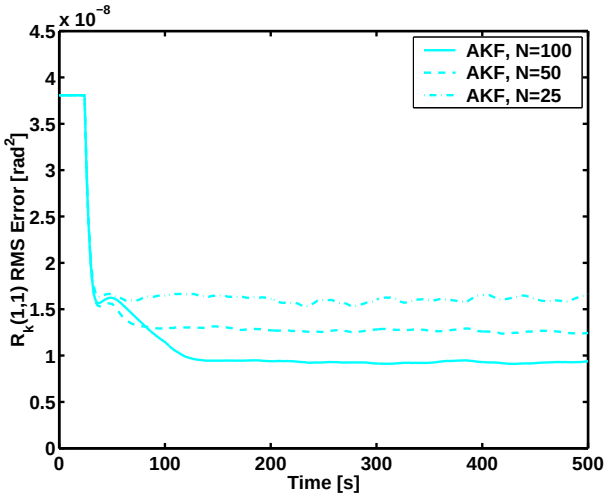


(e) Attitude Angle RMS Error: AHF–PS ($\epsilon = 0$)

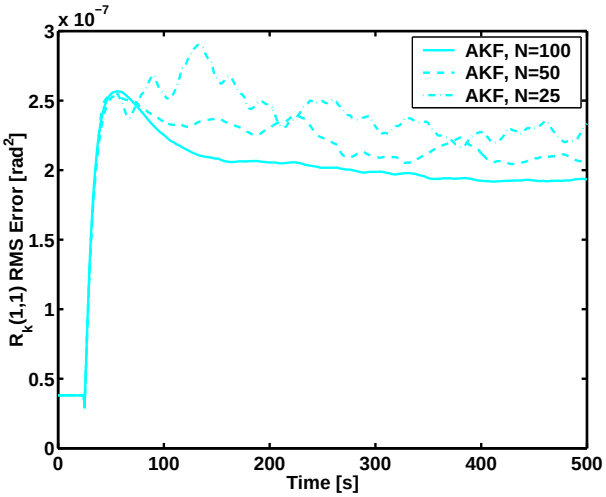


(f) Attitude Angle RMS Error: AHF–PS ($\epsilon = 0.1$)

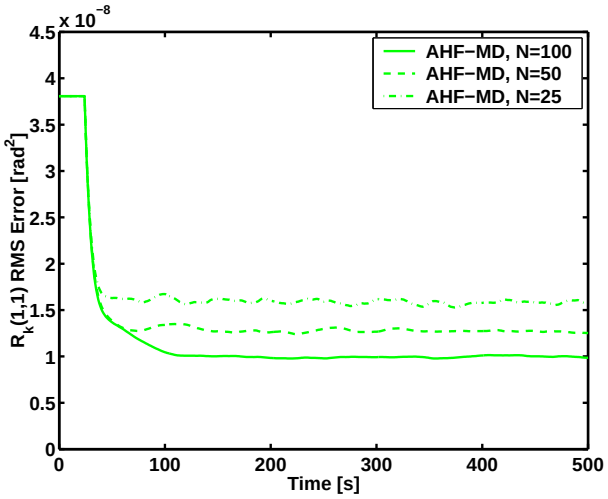
Figure 5.11: Attitude Angle RMS Errors for Varying Buffer Size



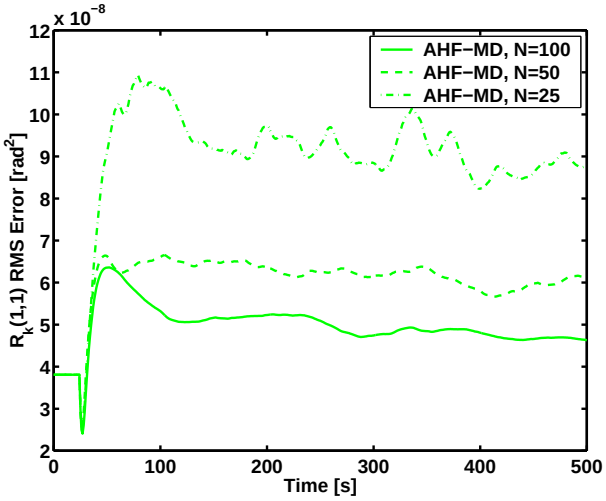
(a) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error: AKF ($\epsilon = 0$)



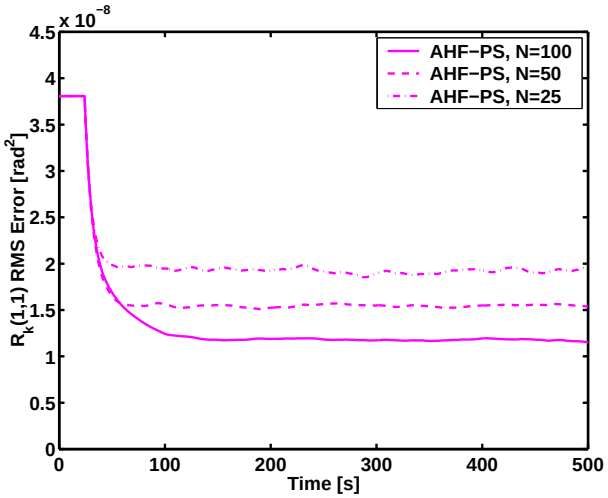
(b) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error: AKF ($\epsilon = 0.1$)



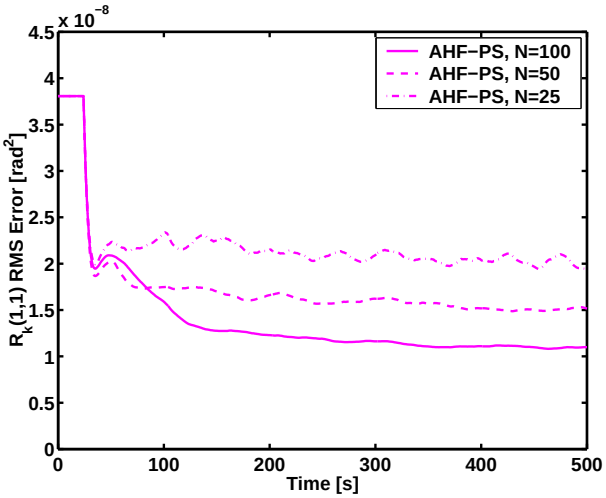
(c) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error: AHF–MD ($\epsilon = 0$)



(d) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error: AHF–MD ($\epsilon = 0.1$)

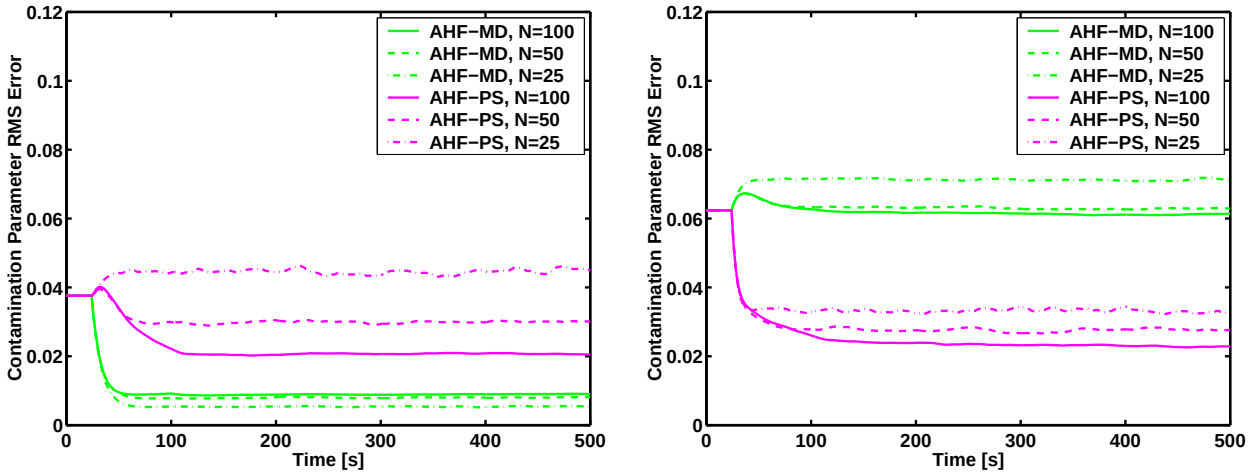


(e) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error: AHF–PS ($\epsilon = 0$)



(f) $\tilde{\boldsymbol{R}}_k(1,1)$ RMS Error: AHF–PS ($\epsilon = 0.1$)

Figure 5.12: $\boldsymbol{R}_{11}$ RMS Errors for Varying Buffer Size



(a) Contamination Parameter RMS Error ($\epsilon = 0$)

(b) Contamination Parameter RMS Error ($\epsilon = 0.1$)

Figure 5.13: Contamination Parameter RMS Errors for Varying Buffer Size

Figure 5.12 shows the $\tilde{\boldsymbol{R}}_k(1,1)$ RMS errors for the adaptive filters for a range of N for both Gaussian and non–Gaussian cases. Differences are apparent in the filter performance depending on the type of filter and the parameter N, which controls the amount of data used in the adaptive filters for estimating the covariance matrices. In all cases, the $\boldsymbol{R}_{11}$ RMS error decreases as the parameter N is increased.

Figure 5.13 shows the contamination parameter estimates for the Huber filters based on the Mahalanobis Distances and the Projection Statistics. These results show interesting behavior in estimates based on the Mahalanobis distances in that they are actually more accurate for smaller sample sizes in the case of no contamination. This effect implies that the Mahalanobis distances tend to be optimistic regarding the contamination in small samples. This phenomenon is another well known limitation of the Mahalanobis distances: the small sample Mahalanobis distances tend to regard the entire sample as Gaussian, thus setting a lower contamination ratio, which in the Gaussian case leads to a lower error level.[116] In the non–Gaussian case, the opposite trend occurs and the error increases since the true distribution has nonzero contamination. The estimates based on the projection statistics are not as accurate for Gaussian cases but they do follow the expected trends of increasing accuracy with an increasing buffer size. In the small sample case (N=25) without contamination, the projection statistics tend to over predict the ratio of contamination.

5.2.4 Computational Costs

Based on the results shown in the previous section, the Huber–based filtering methods show a clear advantage over traditional Kalman filter methods when applied to contaminated Gaussian measurement distributions. In the Gaussian case, the Huber–based methods do not suffer appreciably from any kind of mistuning associated with incorrect assumptions regarding the true underlying statistics of the problem. This section compares the computational costs associated with the Huber–based approaches compared with the Kalman filtering techniques. Table 5.5 shows the computational costs for each filter discussed in this dissertation. Here, the computational time is divided by that corresponding to the standard Kalman filter in order to provide a relative comparison of the

Table 5.5: Relative Computation Ratios

Filter	Nonadaptive	N=25	$N = 50$	$N = 100$
Kalman Filter	1.00	–	–	–
Adaptive Kalman Filter	–	1.13	1.16	1.24
Huber Filter	1.33	–	–	–
Adaptive Huber Filter (Mahalanobis Distances)	–	1.75	1.89	2.15
Adaptive Huber Filter (Projection Statistics)	–	2.86	4.29	7.18

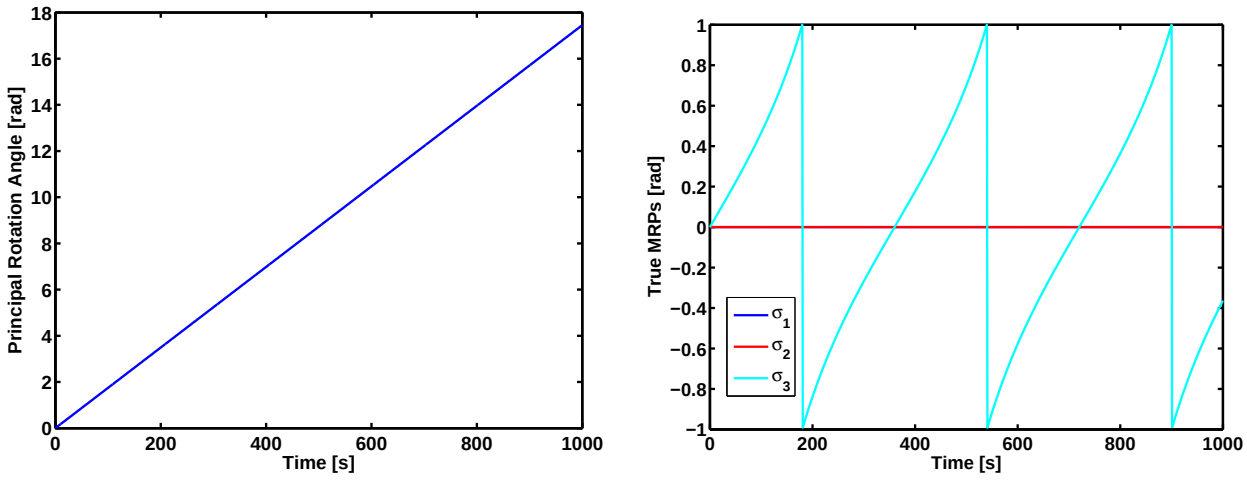
computational burden associated with the particular filter.

For this problem, the adaptive Kalman filter and the Huber filter are comparable in computational cost relative to the standard Kalman filter. The adaptive Huber filters proposed in this dissertation require more computation, well beyond that of the adaptive Kalman filter or the nonadaptive Huber filter as can be seen from Table 5.3. As expected, the computation associated with the adaptive filters is reduced as the buffer size reduced. Note that the computational cost of the filter based on the projection statistics is reduced at a higher rate than the filter based on the Mahalanobis distances. The adaptive Huber filter based on the projection statistics has good properties when applied to contaminated Gaussian distributions, however, one must trade off the computational costs to determine the feasibility of implementing the technique for any particular application.

5.3 Attitude Filtering Using Modified Rodrigues Parameters

This section describes an example problem that illustrates the MRP–based estimation techniques using the shadow set transformation for singularity avoidance. In this problem, consider a spacecraft rotating with an angular velocity of 1 deg/s about the body z-axis over a period of 1000 s. The simulation parameters are shown in Table 5.6. The true principal rotation angle and true MRP time history are shown in Fig. 5.14. Note that there are several shadow set transformations apparent in Fig. 5.14(b) in order to keep the MRP value within the unit sphere.

The results of a 2000 case Monte-Carlo simulation are shown Fig. 5.15. Figure 5.15(a) shows the root mean square (RMS) total attitude angle error and Fig. 5.15(b) shows RMS of the norm of the gyroscope bias estimate error. The Monte-Carlo simulations involve five filtering techniques: a standard Quaternion Multiplicative Extended Kalman Filter (QM-EKF),[87] a Quaternion Constrained Extended Kalman Filter (QC-EKF),[90] an extended Kalman filter based on the MRP formulation discussed in this dissertation (MRP-EKF), and first and second–order divided difference filters using the MRP formulation (MRP-DD1 and MRP-DD2, respectively). In these plots, the RMS errors of the filters are shown in the solid curves while the predicted RMS error based on the filter covariance matrix are shown in the dashed curves. In this case, the QM–EKF and the MRP–EKF exhibit nearly the same overall performance. This result is not a surprise because both filters involve similar first–order approximations of the state dynamics and measurement noise. However, it can be seen in the detailed plot over the first 50 s of the simulation, Fig. 5.15(c) and (d), that the MRP–EKF converges faster than the QM–EKF to the steady state error level. This enhanced convergence rate is due to the fact that the MRP formulation does not require a linearization in



(a) True Principal Rotation Angle

(b) True MRPs

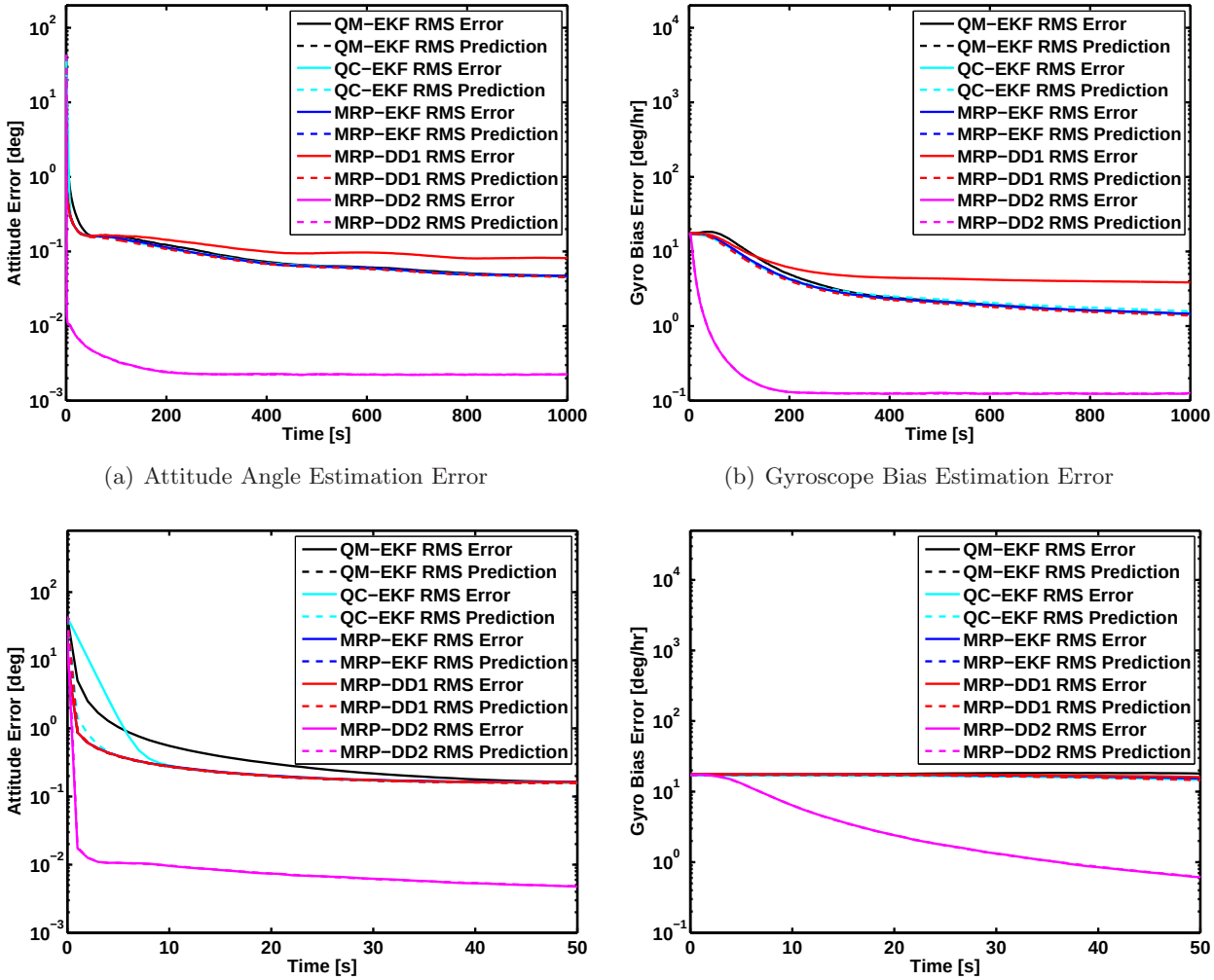
Figure 5.14: True Principal Rotation Angle and MRPs

Table 5.6: Simulation Parameters

Variable	Value
Gyroscope Sample Rate	10 Hz
MRP Sample Rate	1 Hz
σ_ω^2	10^{-13} rad^2/s
σ_β^2	10^{-15} rad^2/s^3
σ_s^2	$7.16{\cdot}10^{-5}$ rad^2
$\hat{\boldsymbol{P}}_{\sigma\sigma_0}$	$\text{diag}\left(\begin{bmatrix} 0.0122 & 0.0122 & 0.0122 \end{bmatrix}\right)$ rad^2
$\hat{\boldsymbol{P}}_{\beta\beta_0}$	$\text{diag}\left(\begin{bmatrix} 2.35 & 2.35 & 2.35 \end{bmatrix}\right) \cdot 10^{-9}$ rad^2/s^2
$\hat{\boldsymbol{P}}_{\sigma\beta_0}$	$\mathbf{0}$ rad^2/s
$\boldsymbol{\sigma}_0$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ rad
$\boldsymbol{\beta}_0$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ rad/s

order to enforce the quaternion norm constraint. Similarly, the QC–EKF converges to the steady state error faster then the QM–EKF over all, though its initial convergence rate is slower. The MRP–DD1 filter does not meet the same level of performance as that of the MRP–EKF case. This result is not particularly bothersome since the DD1 filter performance is usually worse than that of the EKF as seen in Refs. 79, 80 and 134. The MRP–DD2 filter exhibits the best performance overall, which is to be expected since it is a second–order filter and as a result can better capture the system nonlinearities. The uncertainty predictions based on the covariance matrix do not match the actual RMS for any of the filter results. The uncertainties can be tuned to better match the actual performance either offline or by using an adaptive approach to estimate the process noise covariance.[66]

As discussed in earlier sections, the MRP switching condition can occur for any value of $\sigma_r \geq 1$. Figure 5.16 shows the estimator performance for values of σ_r ranging from 1 to 1000. The results are



(a) Attitude Angle Estimation Error

(b) Gyroscope Bias Estimation Error

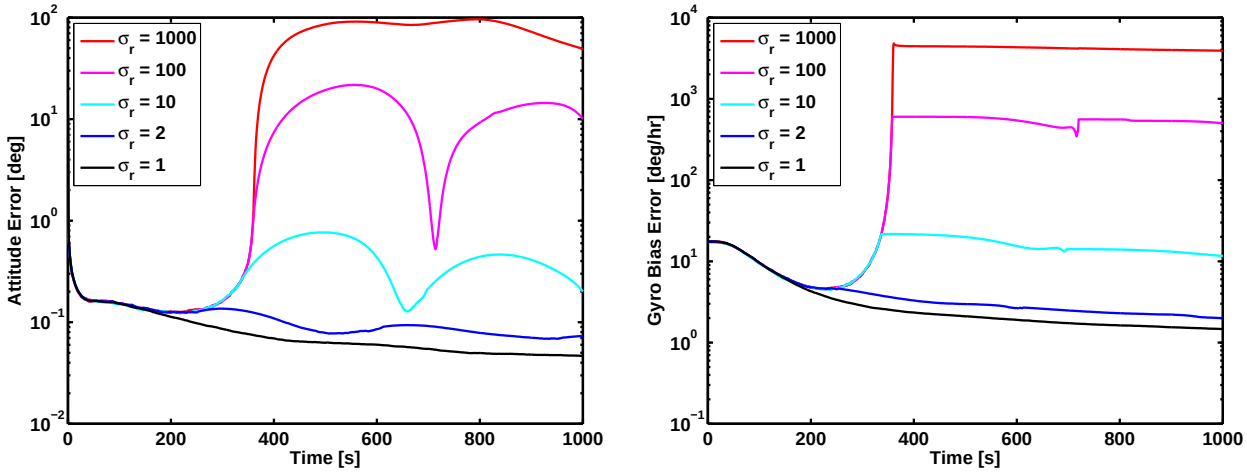
(c) Attitude Angle Estimation Error (Detail)

(d) Gyroscope Bias Estimation Error (Detail)

Figure 5.15: Comparison of MRP-based filters and Quaternion–based filter

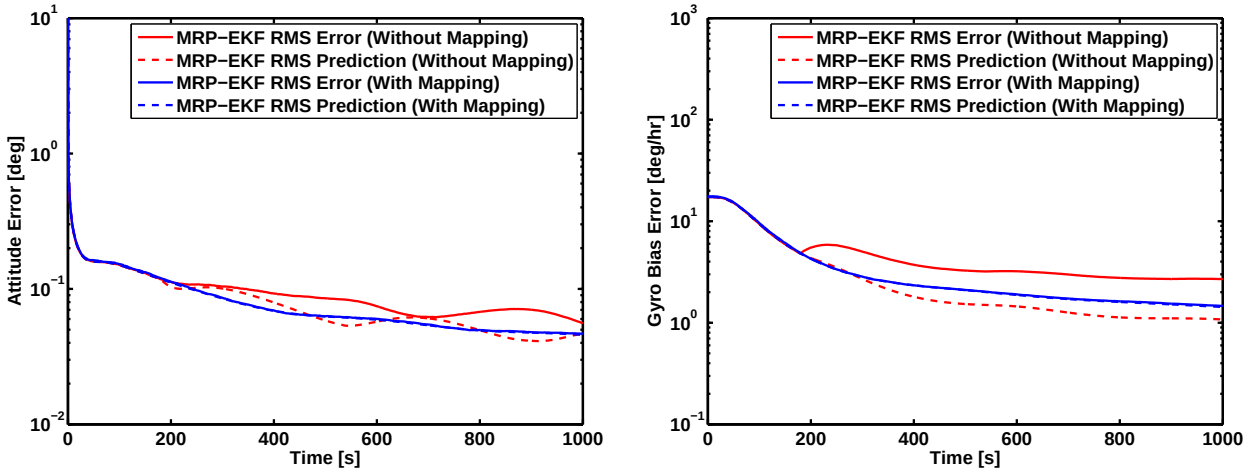
shown only for the EKF formulation of the MRP attitude filter. Clearly the estimator performance degrades as the switching surface grows in magnitude, and it can be inferred from the results that the limiting case $\sigma_r \to \infty$ leads to infinite estimation error since the MRP is reaching the neighborhood of the singularity. Similar trends occur for the DD1 and DD2 formulations. Based on these results there does not seem to be any benefit for using a MRP switching surface greater than the unit sphere but for some particular applications it may be preferable to do so. Having a general MRP covariance switching solution, however, also use to switch at any time where $\|\boldsymbol{\sigma}\| > 1$. It is not required to intercept the $\|\boldsymbol{\sigma}\| = 1$ surface precisely, making the numerical implementation far easier.

Previous applications of the MRP singularity avoidance based on the shadow set transformation have neglected the covariance mapping associated with the transformation. Fig. 5.17 shows a comparison of the MRP–based EKF with and without the covariance transformation to illustrate the issues associated with neglecting the transformation. At the first switching point a sharp bend



(a) Attitude Angle Estimation Error (b) Gyroscope Bias Estimation Error

Figure 5.16: Comparison of MRP-based filter with varying σ_r



(a) Attitude Angle Estimation Error (b) Gyroscope Bias Estimation Error

Figure 5.17: Comparison of MRP-based filter with and without the covariance transformation

can clearly be seen in the case without the covariance transformation after which the estimator performance is degraded relative to the case that includes the proper covariance transformation. This bend is due to the fact that elements of the covariance matrix must change sign during the shadow mapping since the MRP state representation changes sign during the mapping. Therefore the estimates that neglect the covariance transformation develop systematic error and are no longer optimal. The results are shown only for the EKF–based filter, similar behavior is found for the DD1 and DD2 filters.

Table 5.7 shows a comparison of the computational costs of each filter applied to this problem. The mean computation time is calculated for each Monte-Carlo set and then divided by the QM–EKF time to provide a relative cost comparison ratio. Also the standard deviation of the com-

Table 5.7: Computational Cost

Filter	Mean Computation Time	Standard Deviation
QM–EKF	1.000	0.015
QC–EKF	1.089	0.029
MRP–EKF	0.977	0.024
MRP–DD1	10.795	0.316
MRP–DD2	11.062	0.274

putation times are provided to show the confidence intervals. The MRP–based EKF formulation described in this dissertation requires slightly less computation on average than the quaternion–based EKF. These cost savings are consistent with the results of Ref. 152, which found a reduced computation using the Rodrigues parameters for attitude estimation compared with the quaternion filter. The DD1 and DD2 filters require roughly the same computational cost which is consistent with Ref. 134. In this case the divided difference filters are each about an order of magnitude more expensive than the EKF.

5.4 Rendezvous Navigation in Elliptical Orbit

5.4.1 Overview and Simulation Setup

This section described the application of the robust/adaptive filtering algorithms to the problem of 6-DOF rendezvous navigation and control in elliptical orbit. Here, the navigation filters are used "real–time" inside the control loop as state observers. The previous examples have considered only open loop estimation problems where the state estimator does not interact with the dynamics of the problem.

The specific example discussed in this section involves terminal rendezvous maneuver in a 15km by 75km altitude lunar orbit. This orbit is typical of intermediate phasing orbit during ascent from the lunar surface. Hypothetically, the lunar ascent vehicle could have failed, leaving it stranded in such an orbit. Therefore, a CEV/Orion-like vehicle must maneuver from a circular parking orbit, typically 100km altitude, to rendezvous with the lunar ascent vehicle. This problem assumes that the midcourse maneuvering and phasing has already been accomplished such that the initial conditions of the maneuvering vehicle are 1km behind the target vehicle in the in-track direction. The rendezvous trajectory begins with a constant closure rate from 1km to 100m over a duration of half an orbital period (in this problem, one orbital period is approximately 113 minutes). At this point, the vehicle is commanded to execute a circumnavigation of the target vehicle at a constant range of 100m over one orbital period. Finally, the vehicle executes a glideslope maneuver using a Space Shuttle–based exponentially decaying range rate guidance scheme[170] with decay rate of 0.2% over a time span of half an orbital period. These guidance commands are shown in Fig. 5.18. Attitude commands are generated as discussed in Sec. 4.6.2, in which commands are given such that the vehicle remains pointed at the target throughout the maneuver. In a nominal scenario with no out of plane motion, the attitude commands are purely in the pitch plane, resulting in MRP and MRP-rate commands shown in Fig. 5.19. Note that the MRPs are used exclusively

for the attitude representation in this problem, making use of the singularity avoidance methods discussed in Sec. 4.3.1.3 for globally non-singular attitude estimation. The guidance commands are generated at a rate of 4 Hz. The control laws and navigation filter update rates are at 4 Hz. The inertial navigation sensors operate at 20 Hz. The controller gains are computed according to the H_∞ development discussed in Sec. 4.6. The translational gains are computed using the state error weighting matrix $\boldsymbol{C}_t = \left(10^3\right)\boldsymbol{I}$ and the rotational gains are computed using $\boldsymbol{C}_r = \boldsymbol{I}$, with $k_v = 1\ \mathrm{s}^{-2}$. In both cases, the H_∞ performance bound is set to $\kappa = 5$. Control force and torque limits were set to 1779.2 N and and 5337.6 Nm, respectively.

The nominal vehicle mass properties correspond to a total mass of $1.1 \cdot 10^5$ kg with inertia tensor given by

$$\boldsymbol{I} = 10^4 \cdot \begin{bmatrix} 2.9441 & 0.0368 & 0.3680 \\ 0.0368 & 3.6801 & 0.0074 \\ 0.3680 & 0.0074 & 3.6801 \end{bmatrix} \ \mathrm{kg} \cdot \mathrm{m}^2 \tag{5.11}$$

Monte-Carlo simulations have been conducted for this problem for several different navigation filters. In particular, the extended Kalman filter and first and second order divided difference filters, including Huber implementations, Myers–Tapley adaptive implementations, and combined Huber–Myers–Tapley adaptive filtering methods. Results are shown in the following subsections for both Gaussian and non-Gaussian cases.

5.4.2 Gaussian Simulation

Rendezvous simulation conducted with pure Gaussian errors are described in this section. The initial conditions of the maneuvering vehicle are at the time of perilune passage of the target vehicle and are provided in Table 5.8. Table 5.8 shows the true value of the initial conditions, the estimate of the initial conditions for filter initialization, and the components of the initial variance matrix for filter initialization. The initial state errors are uncorrelated in this simulation. The vehicle mass properties are dispersed by multiplying the total mass by a Gaussian random variable with standard deviation of 0.5% and a inertia tensor uncertainty by the mass multiplier coupled with a uncorrelated random axis uncertainty with 0.5 deg standard deviation in yaw-pitch-roll Euler angles.

The navigation sensor uncertainties are summarized in Table 5.9, which shows the measurement standard deviations for the gyro parameters ($\boldsymbol{\eta}_\omega$ and $\boldsymbol{\eta}_\beta$), the accelerometer errors ($\boldsymbol{\eta}_a$ and $\boldsymbol{S}$), the rendezvous lidar sensor errors (ϱ, α, ε, and $\delta\boldsymbol{\sigma}$), the star tracker errors ($\delta\boldsymbol{\sigma}_I$) and the orbit sensor errors (r_0 and θ_0). For this simulation, the errors are cast as uncorrelated Gaussian random numbers with standard deviations given in Table 5.9. The filter measurement noise and process noise matrices are set according to the sensor errors in Table 5.9 without error in the assumed values. In this simulation, the accelerometer scale factor errors are not modeled in the filter formulation as state variables in order to provide a mismatch between the true measurements and the modeled measurements in the filter.

Figures 5.20–5.22 show the results of 100 Monte-Carlo cases involving Gaussian errors. In these cases, the navigation system is operating in pure-inertial mode, in which no measurement data are processing in the navigation filter to improve accuracy and reduce the drift inherent in inertial navigation systems. Figure 5.20 shows the position and velocity RMS error, Fig. 5.21 shows the attitude and gyro bias RMS errors, and Fig. (5.22) shows the orbital position and velocity RMS

Table 5.8: Initial Conditions

Initial State	Mean	Standard Deviation
$\rho(0)$, km	1.0	0.005
$\theta(0)$, deg	180.0	0.25
$\phi(0)$, deg	0.0	0.25
$\dot{\rho}(0)$, m/sec	-0.266	0.05
$\dot{\theta}(0)$, deg/sec	0.0	0.003
$\dot{\phi}(0)$, deg/sec	0.0	0.003
σ_1, rad	0.0	0.0175
σ_2, rad	0.0	0.0175
σ_3, rad	0.0	0.0175
β_1, deg/hr	0.0	1.0
β_2, deg/hr	0.0	1.0
β_3, deg/hr	0.0	1.0
r_0, km	1753.1	0.01
θ_0, deg	0.0	0.001
$\dot{r}_0$, m/s	0.0	$1.0 \cdot 10^{-5}$
$\dot{\theta}_0$, deg/s	$9.619 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$

Table 5.9: Sensor Specifications

Measurement	Standard Deviation
$\boldsymbol{\eta}_\omega$, deg/s	$1.8 \cdot 10^{-5}$ $\boldsymbol{I}$
$\boldsymbol{\eta}_\beta$, deg/s^2	$1.8 \cdot 10^{-8}$ $\boldsymbol{I}$
$\boldsymbol{\eta}_a$, m/s^2	0.1 $\boldsymbol{I}$
$\boldsymbol{S}$, ppm	500
ϱ, m	0.1
α, deg	0.05
ε, deg	0.05
$\delta\boldsymbol{\sigma}$, deg	0.05 $\boldsymbol{I}$
$\delta\boldsymbol{\sigma}_I$, deg	0.05 $\boldsymbol{I}$
r_0, m	0.1
θ_0, deg	$3.3 \cdot 10^{-5}$

errors. These figures clearly indicate that the rendezvous maneuver cannot be accomplished without processing measurement data in the filter due to the large buildup of error in the navigation solution.

Figures 5.23–5.28 show the RMS error results of the Gaussian simulation involving the use of several navigation filters for processing measurement data to aid the pure inertial solution. Results are shown from the first and second order divided difference filters (DD1 and DD2), including Huber implementations (H-DD1 and H-DD2), Myers–Tapley adaptive implementations (A-DD1 and A-DD2), and combined Huber–Myers–Tapley adaptive filtering methods (AH-DD1 and AH-DD2). The adaptive filters use a buffer size of 500 samples in the procedures to estimate the measurement and process noise covariances. Results from the Extended Kalman Filter approaches are essentially identical to the first–order divided difference filtering approaches, both being first-order filters, and therefore are omitted from the RMS error plots in an effort to keep the results clear. In Figures 5.23–5.28, the DD1 results are shown with the blue curve, the DD1 results with the red curve, the H-DD1 results in cyan, the H-DD2 results in magenta, the A-DD1 results in green, and A-DD2 results in yellow, the AH-DD1 results in the solid black curve, and the AH-DD2 results in the dashed black curve.

Figure 5.23 shows the position RMS error results. The left column shows the actual RMS error plots for the position variables ρ, θ, and ϕ while the right column shows the predicted RMS errors based on the filter covariance matrix. Although the differences between the performance various filters in this case is small, it is possible to discern a slightly better performance in the Gaussian-based filters such as the DD1 and DD2. Note in this simulation that the measurement and process noise cases were initialized without error. Therefore, the adaptive filters do not perform as well as the non-adaptive filters since their measurement and process noise covariances are not exact estimates due to the finite sample sizes used in the buffering technique.

The predicted RMS error results show the trend that the adaptive Gaussian filters (A-DD1 and A-DD2) exhibit the smallest error prediction. This trend is due to the adaptive tuning of the filter in real time, which has the effect of reducing the state covariance matrix. This reduction is erroneous however, since the actual RMS error results are higher due to the introduction of error into the measurement and process noise covariances. The Huber–based filters (H-DD1 and H-DD2) correctly predict that the RMS error should be higher than the other filters. The adaptive Huber filters (AH-DD1 and AH-DD2) have a slightly lower RMS error prediction than the non-adaptive Huber filters due to the adaptive tuning.

The actual RMS error and predicted RMS error results for the velocity variables $\dot{\rho}$, $\dot{\theta}$, and $\dot{\phi}$ is shown in Fig. 5.24. As in the position plots, the actual RMS error from the Monte-Carlo simulation is shown on the left column and the predicted RMS error is shown in the right column. As expected, the velocity performance trends between the various filters matches with the position performance trends.

The MRP attitude RMS errors are shown in Fig. 5.25. As in the previous results, the differences between the various filters is small yet it can be seen that the Gaussian filters perform slightly better than their non–Gaussian counterparts. In particular the adaptive filters tend to under predict the accuracy of the attitude solution. The adaptive DD1 filters exhibit the lowest RMS predictions, which is consistent with recent analytical results that conclude that the DD1 filter can produce overly optimistic covariance estimates.[171] The MRP shadow set switching point for the singularity avoidance can be seen in the RMS error prediction plots at the time of approximately 1.9 hrs. The rate bias estimate RMS errors are shown in Fig. 5.26. Expectedly, the rate bias RMS trends follow

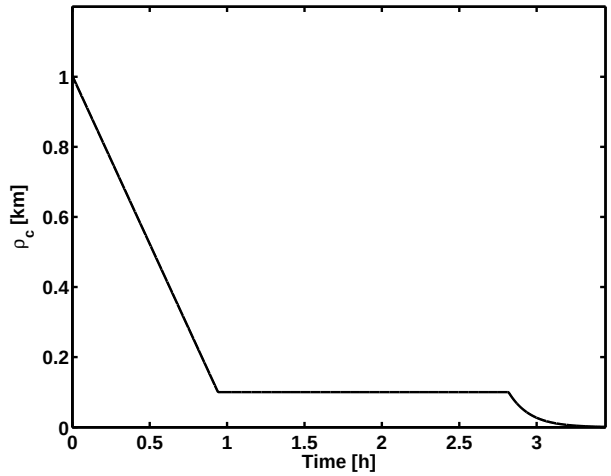
the MRP attitude RMS error trends.

The RMS errors of the reference orbit position estimates are shown in Fig. 5.27, and those for the orbit velocity estimates are shown in Fig. 5.28. These results follow similar trends observed in the relative position, velocity, and attitude estimates.

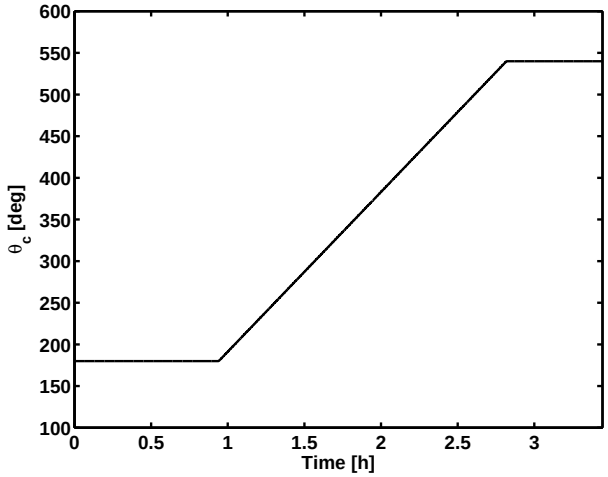
Figure 5.29 shows a comparison of results for the adaptive filters investigated in this study. Figure 5.29(a) shows the RMS estimation error of the (1,1) component of the measurement noise covariance matrix, which corresponds to the range measurement from the lidar sensor. Here, the A-DD2 exhibits the best performance, which is expected since the errors are purely Gaussian and since the DD2 filter has the benefits of capturing second–order terms in the state covariance estimates. The Huber–based estimates exhibit higher errors due to the reduced statistical efficiency of the projection statistics algorithm for purely Gaussian errors. Similarly, the RMS error of the (5,5) component of the measurement noise covariance matrix is shown in Fig. 5.29(b). This element corresponds to an inertial star tracker MRP measurement. The A-DD1 and A-DD2 estimates of this parameter are nearly identical, with both estimates being superior to the Huber–based estimates. Figure 5.29(c) shows the (1,1) component of the process noise covariance estimate. In this case differences between the various adaptive filters is not as obvious, thought the A-DD2 filter shows slightly better performance than the others. Figure 5.29(d) shows the results of estimating the contamination parameter in the Huber–based adaptive filters. Here, both filters AH-DD1 and AH-DD2 are able to reduce the error in the assumed contamination parameter.

Finally, the closed-loop system performance results are shown in Fig. 5.30. These plots show the true position and attitude RMS error results in the left and right columns, respectively. The relative performance of the system is not significantly different between the various filters, although Gaussian filters DD1 and DD2 show slight improvement. This result is expected since the errors are purely Gaussian and the filters have perfect knowledge of the measurement and process noise covariance matrices.

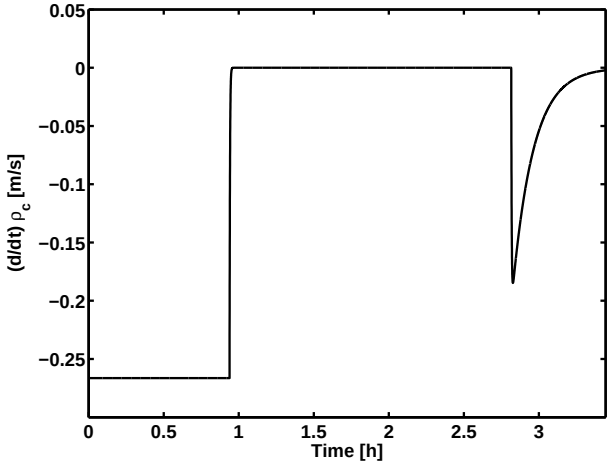
To summarize the results of the Gaussian simulation, the overall trends follow the expectation that the Gaussian DD1 and DD2 filters should have the best overall performance since the errors in the simulation are purely Gaussian and these filters have perfect knowledge of the measurement and process noise covariance matrices. However, the differences between the various filters is fairly small, implying that for the benign Gaussian case the navigation filters are almost identical in overall performance.



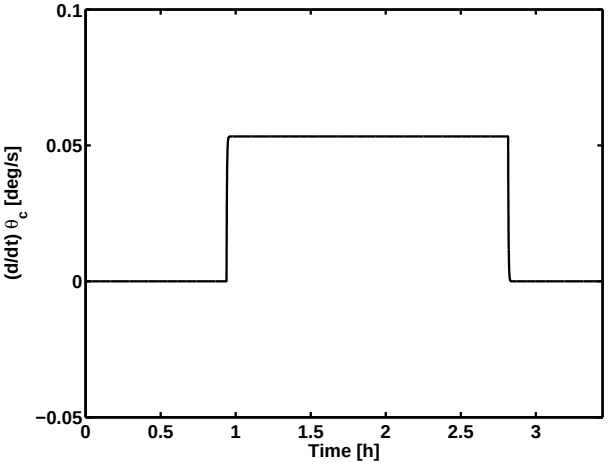
(a) ρ_c



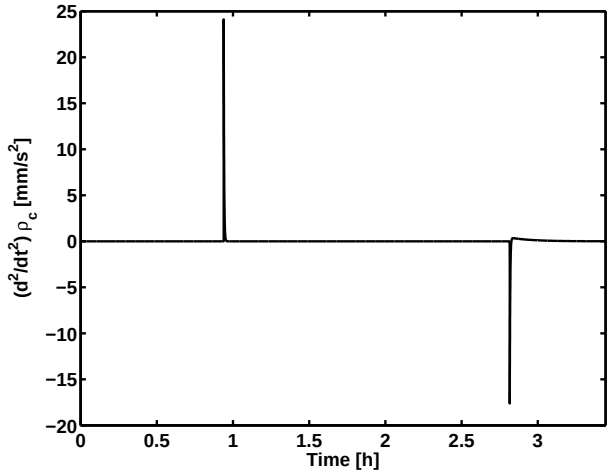
(b) θ_c



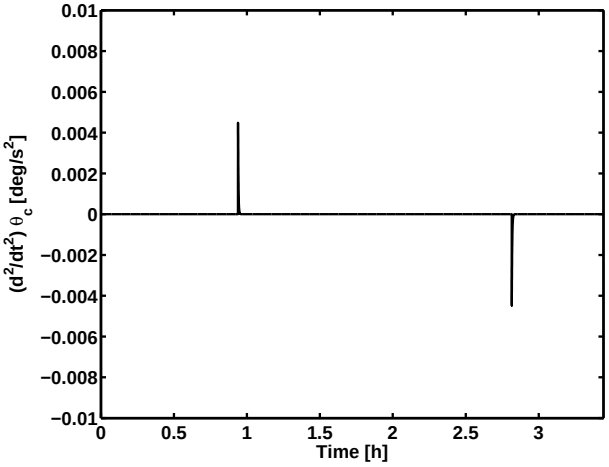
(c) $\dot{\rho}_c$



(d) $\dot{\theta}_c$

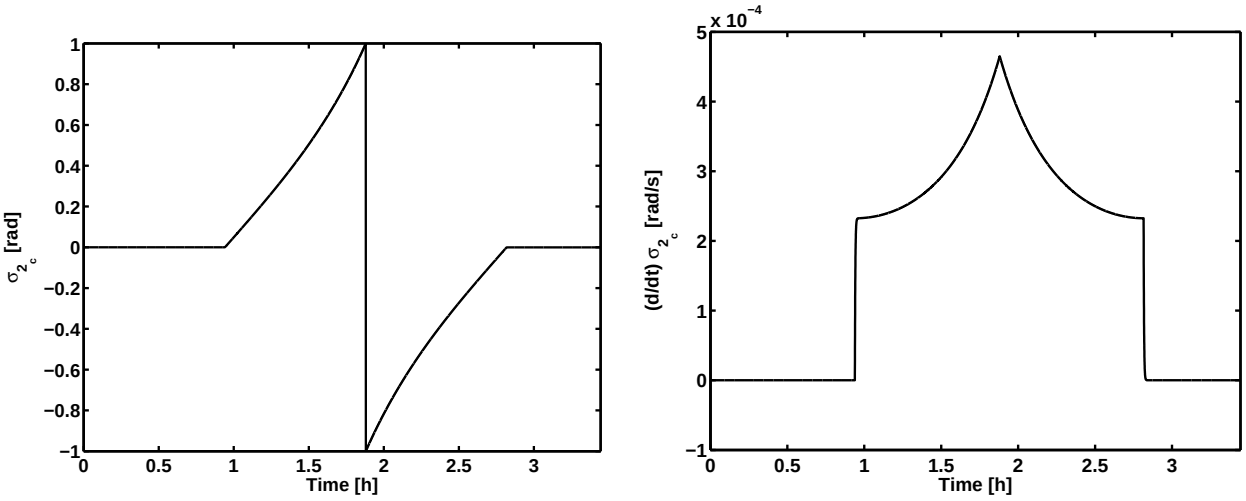


(e) $\ddot{\rho}_c$



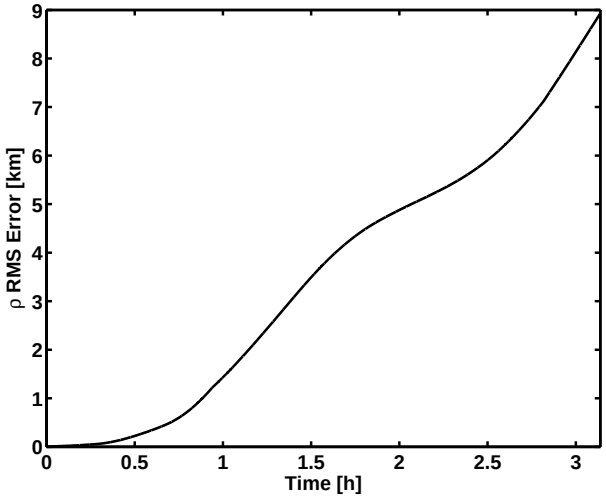
(f) $\ddot{\theta}_c$

Figure 5.18: Translational Guidance Commands

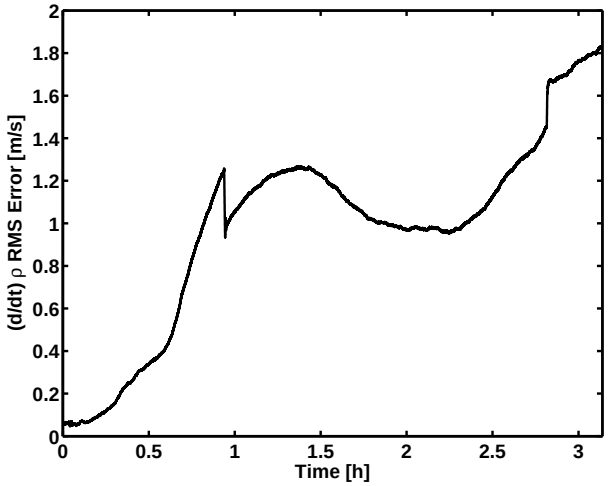


(a) σ_{2_c} (b) $\dot{\sigma}_{2_c}$

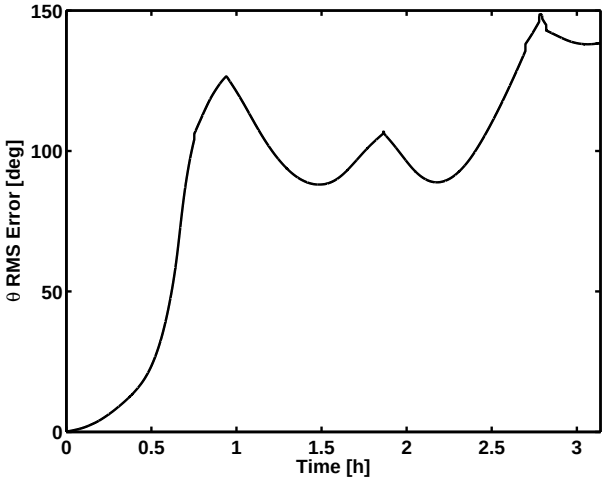
Figure 5.19: Attitude Guidance Commands



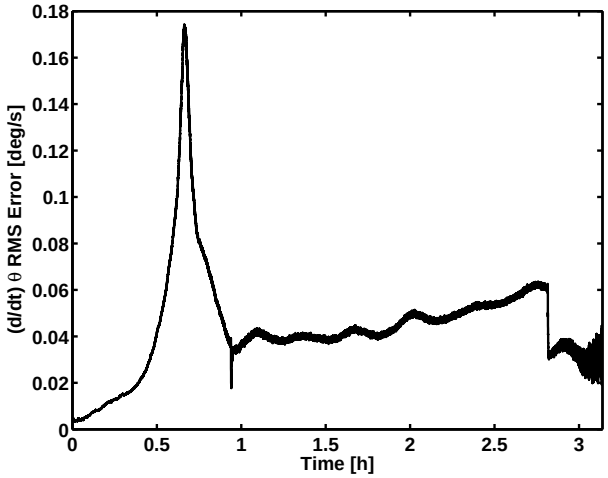
(a) ρ RMS Error



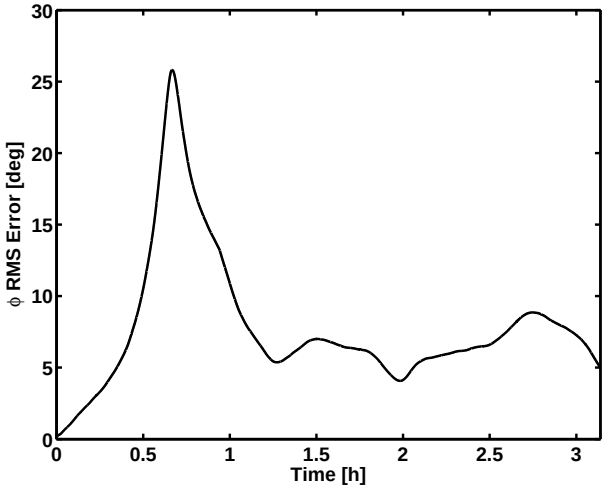
(b) $\dot{\rho}$ RMSError



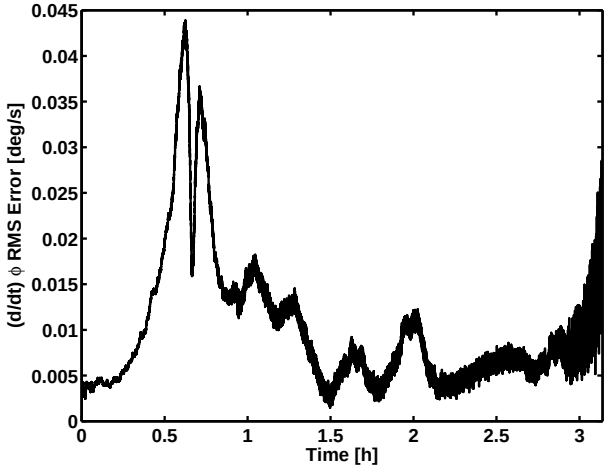
(c) θ RMS Error



(d) $\dot{\theta}$ RMS Error

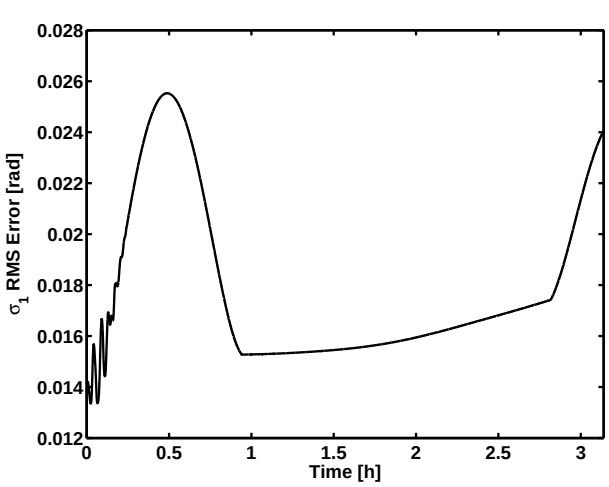


(e) ϕ RMS Error

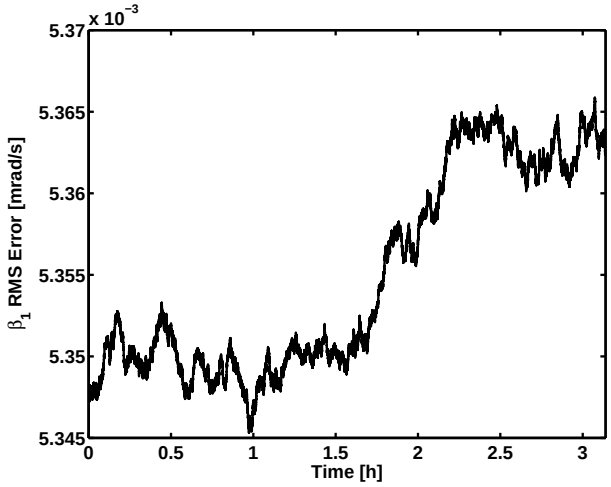


(f) $\dot{\phi}$ RMS Error

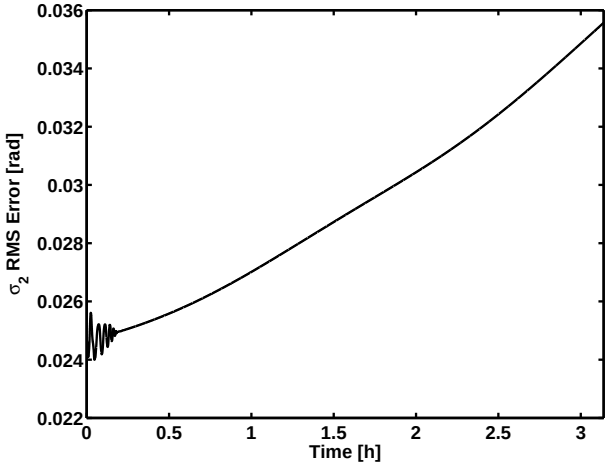
Figure 5.20: Pure Inertial Navigation Results: Position and Velocity



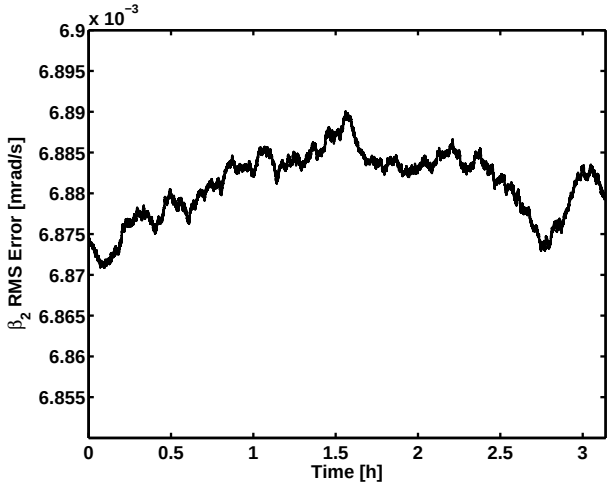
(a) σ_1 RMS Error



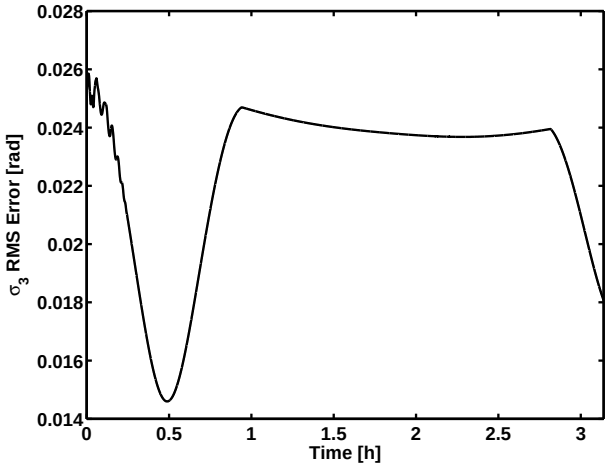
(b) β_1 RMS Error



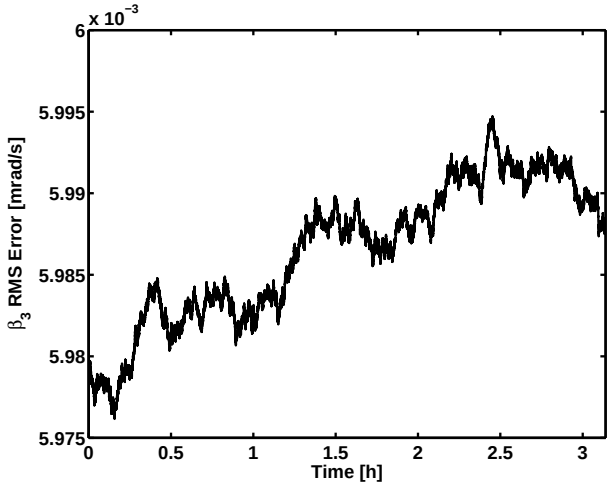
(c) σ_2 RMS Error



(d) β_2 RMS Error

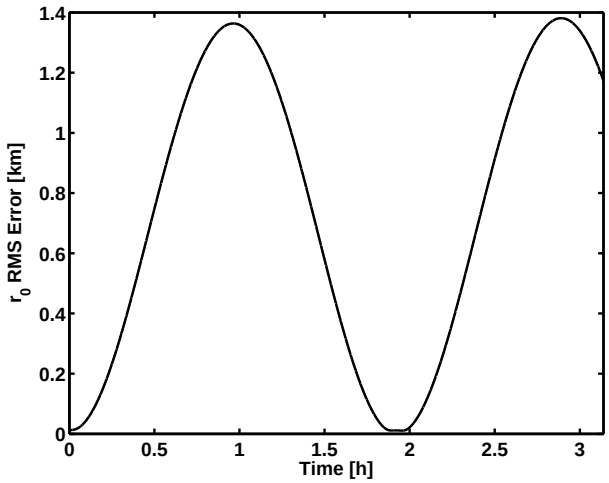


(e) σ_3 RMS Error

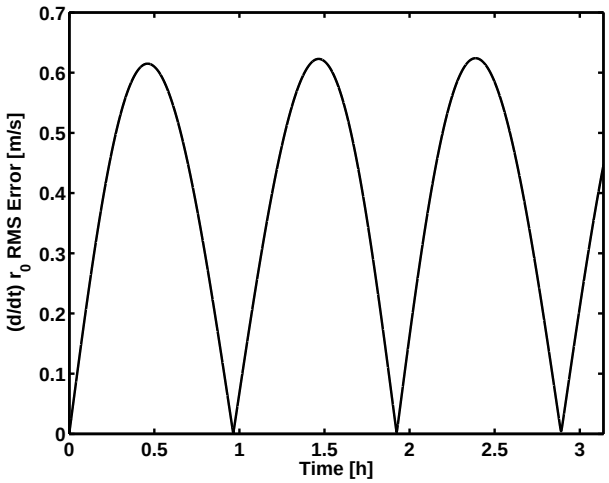


(f) β_3 RMS Error

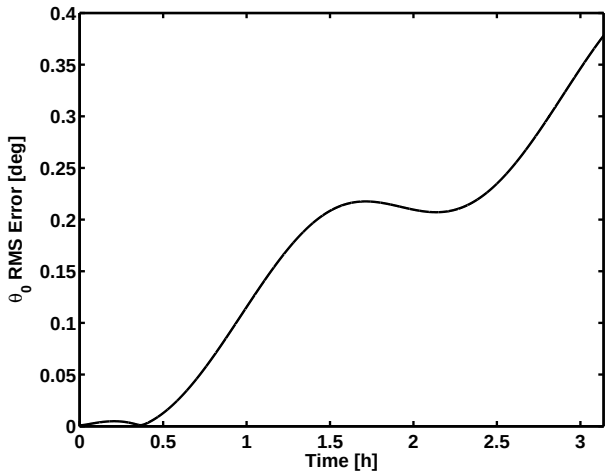
Figure 5.21: Pure Inertial Navigation Results: Attitude and Rate Bias



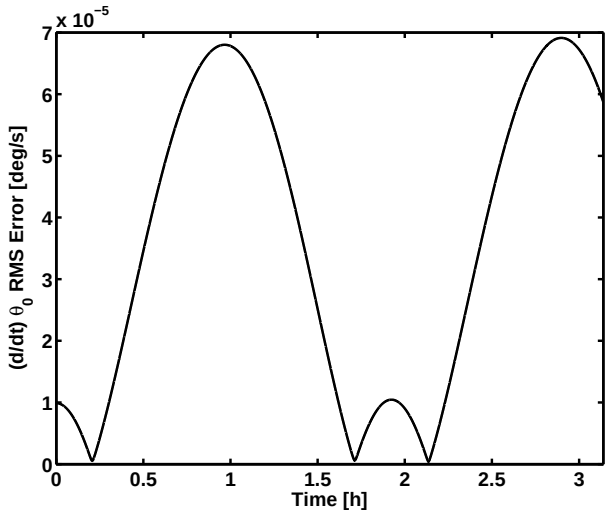
(a) r_0 RMS Error



(b) $\dot{r}_0$ RMS Error

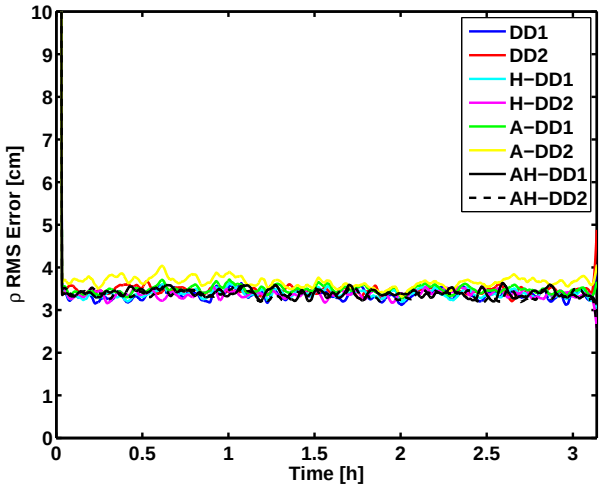


(c) θ_0 RMS Error

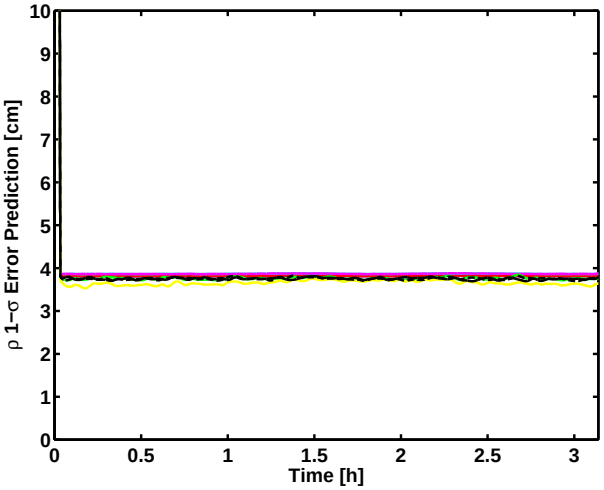


(d) $\dot{\theta}_0$ RMS Error

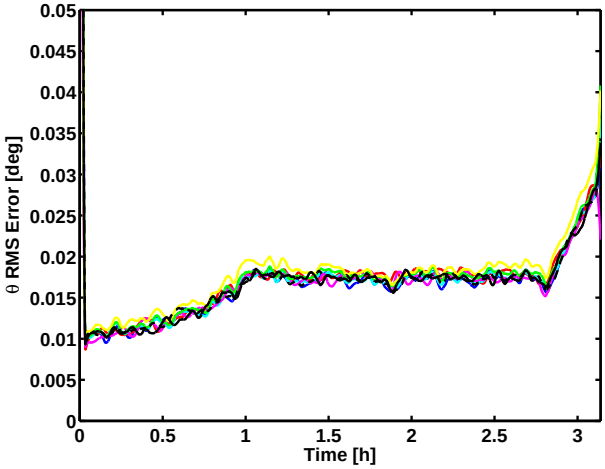
Figure 5.22: Pure Inertial Navigation Results: Orbit



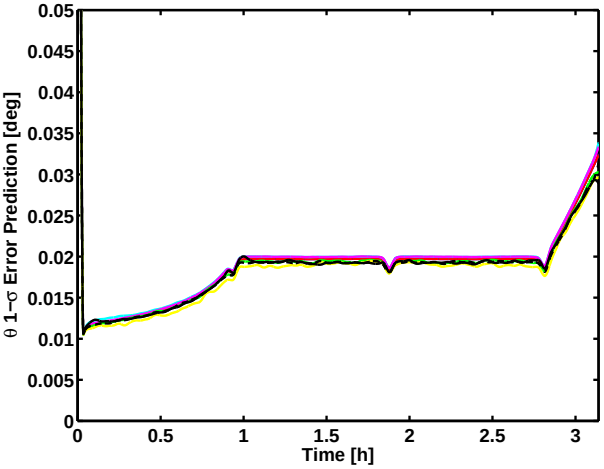
(a) ρ RMS Error



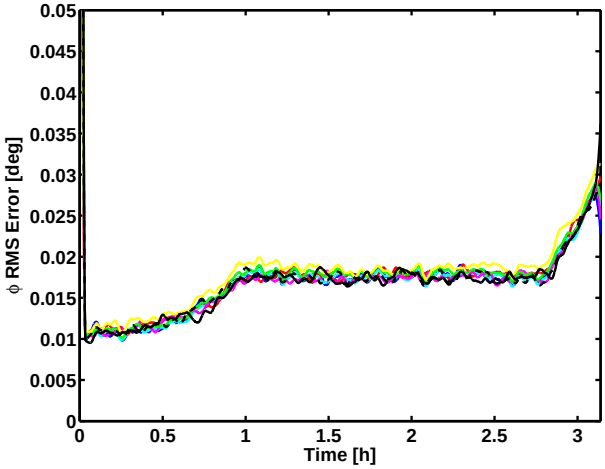
(b) ρ RMS Error Prediction



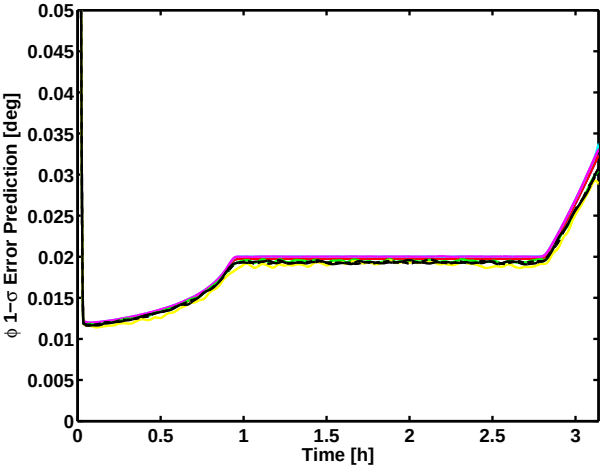
(c) θ RMS Error



(d) θ RMS Error Prediction

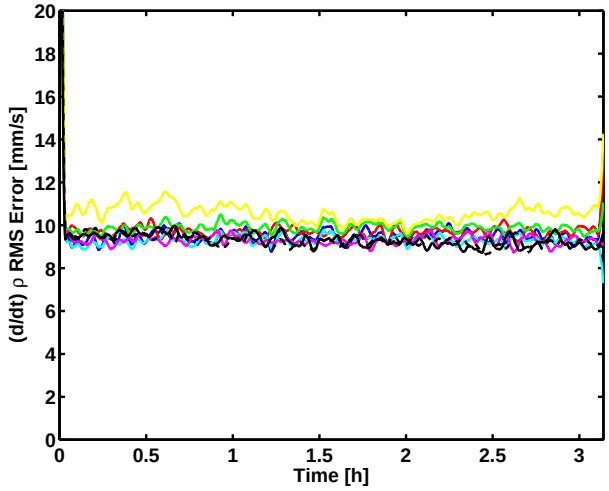


(e) ϕ RMS Error

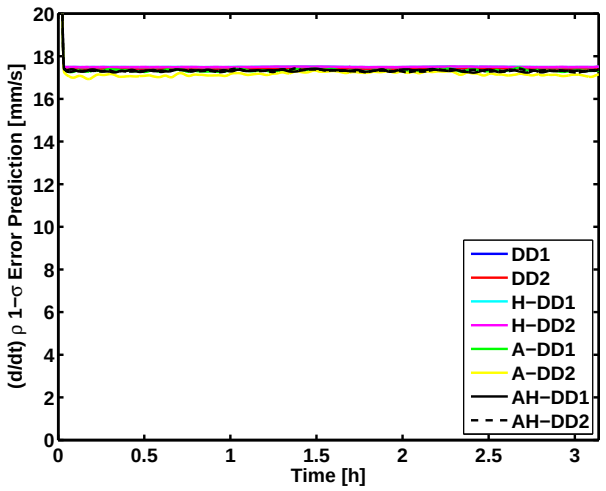


(f) ϕ RMS Error Prediction

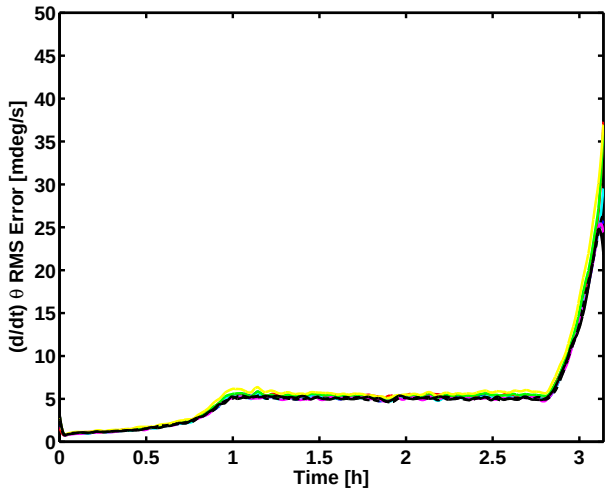
Figure 5.23: Gaussian Simulation Results: Position



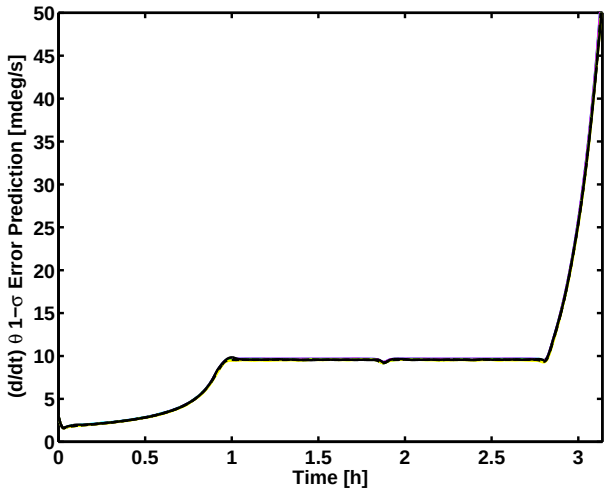
(a) $\dot{\rho}$ RMS Error



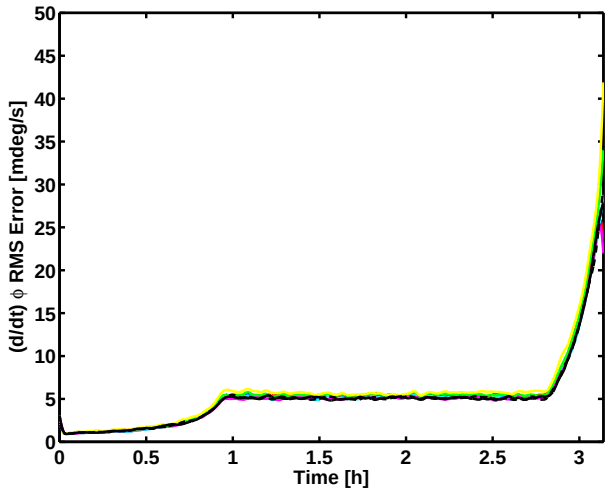
(b) $\dot{\rho}$ RMS Error Prediction



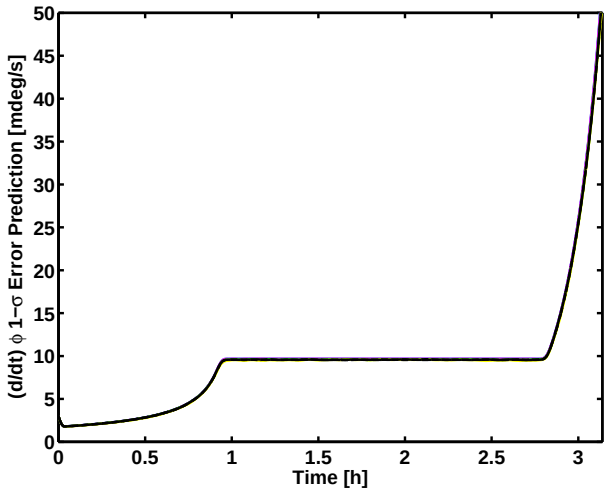
(c) $\dot{\theta}$ RMS Error



(d) $\dot{\theta}$ RMS Error Prediction

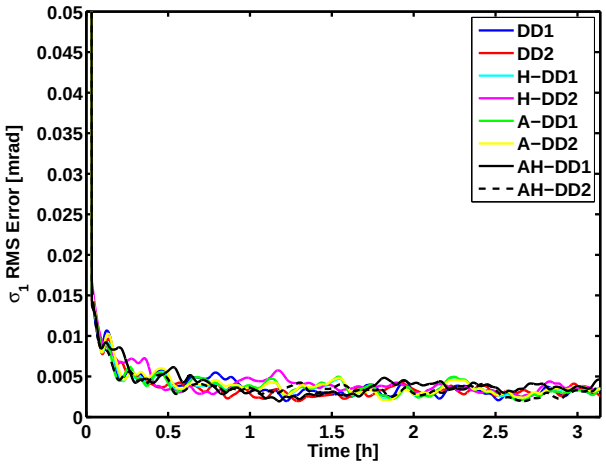


(e) $\dot{\phi}$ RMS Error

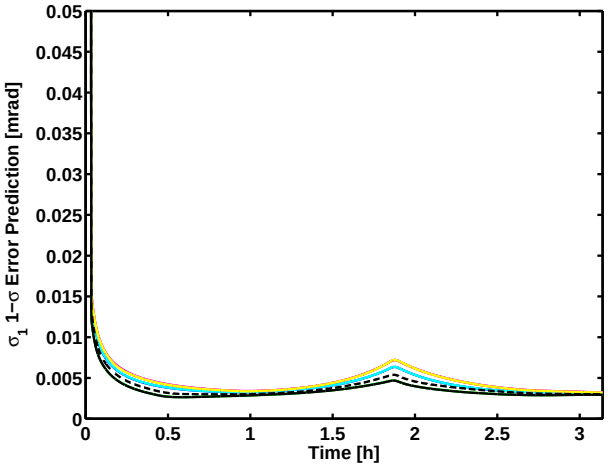


(f) $\dot{\phi}$ RMS Error Prediction

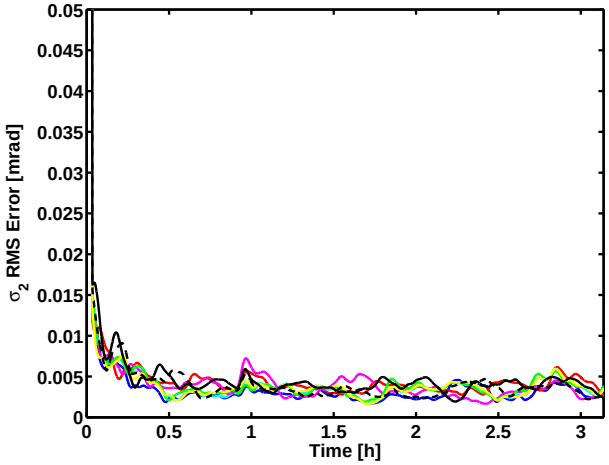
Figure 5.24: Gaussian Simulation Results: Velocity



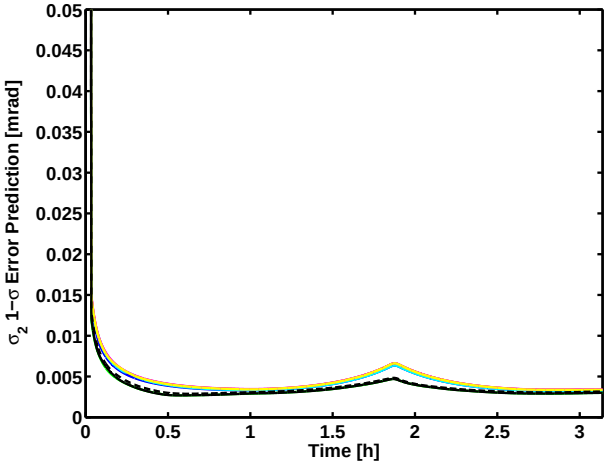
(a) σ_1 RMS Error



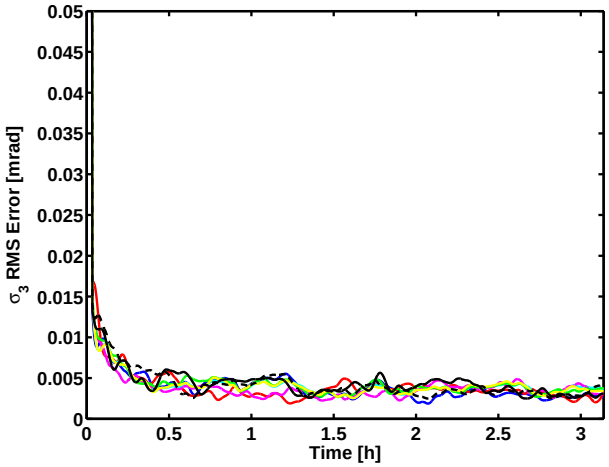
(b) σ_1 RMS Error Prediction



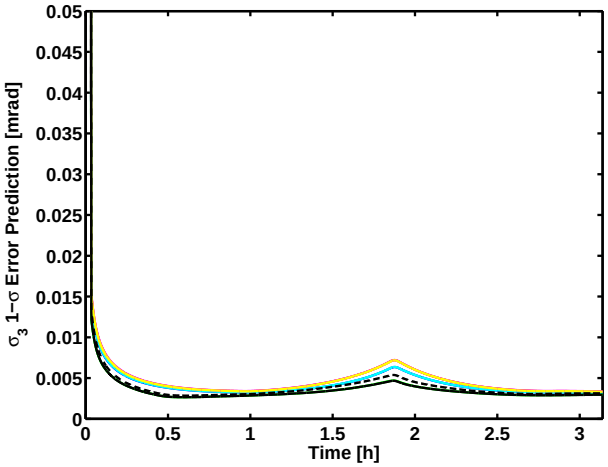
(c) σ_2 RMS Error



(d) σ_2 RMS Error Prediction

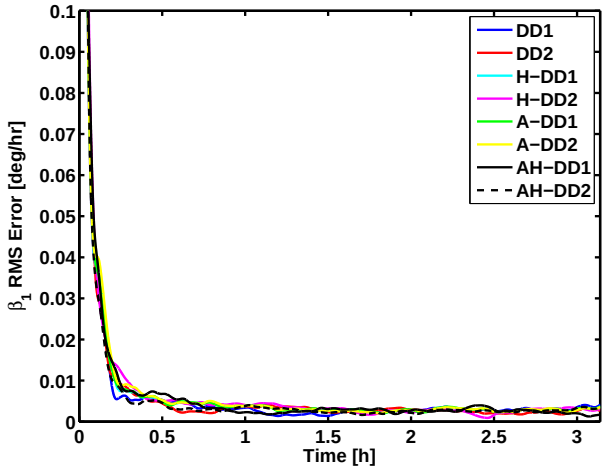


(e) σ_3 RMS Error

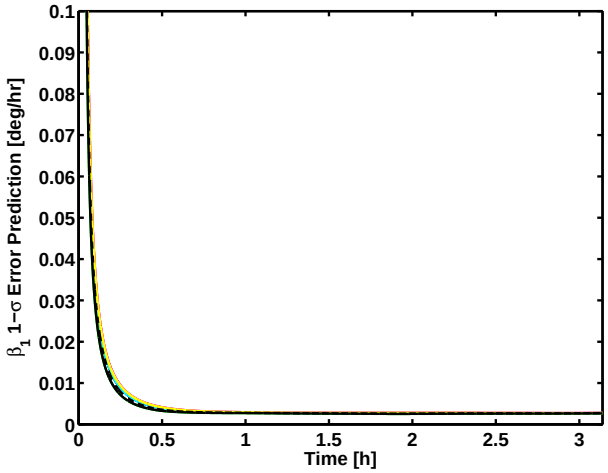


(f) σ_3 RMS Error Prediction

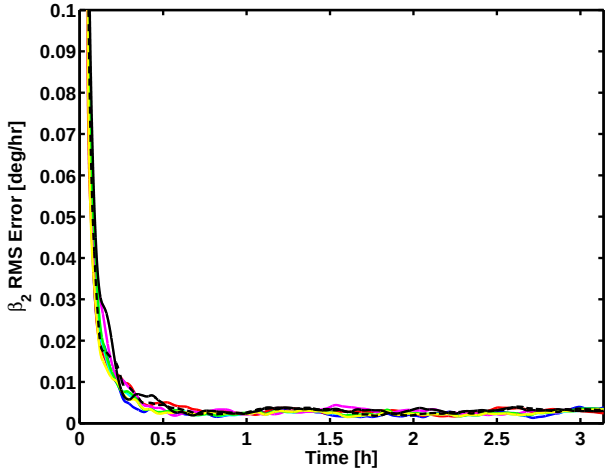
Figure 5.25: Gaussian Simulation Results: Attitude



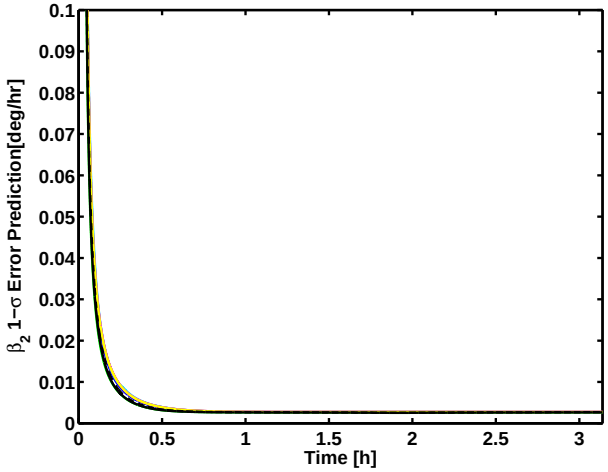
(a) β_1 RMS Error



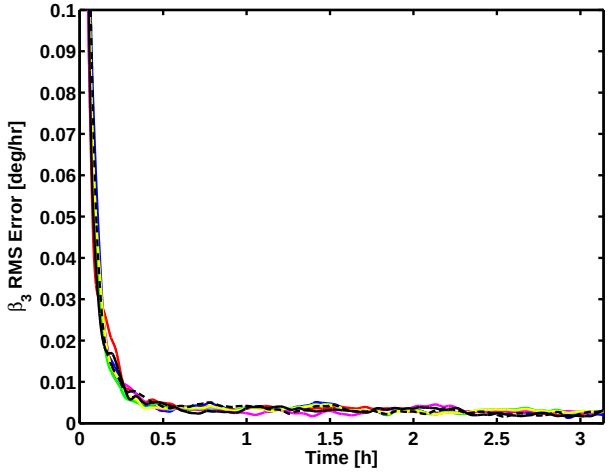
(b) β_1 RMS Error Prediction



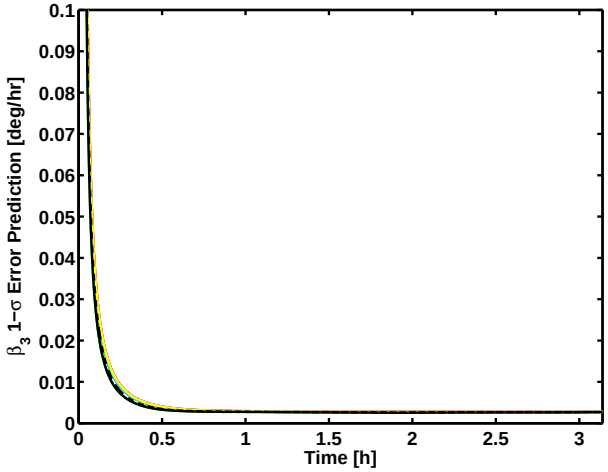
(c) β_2 RMS Error



(d) β_2 RMS Error Prediction

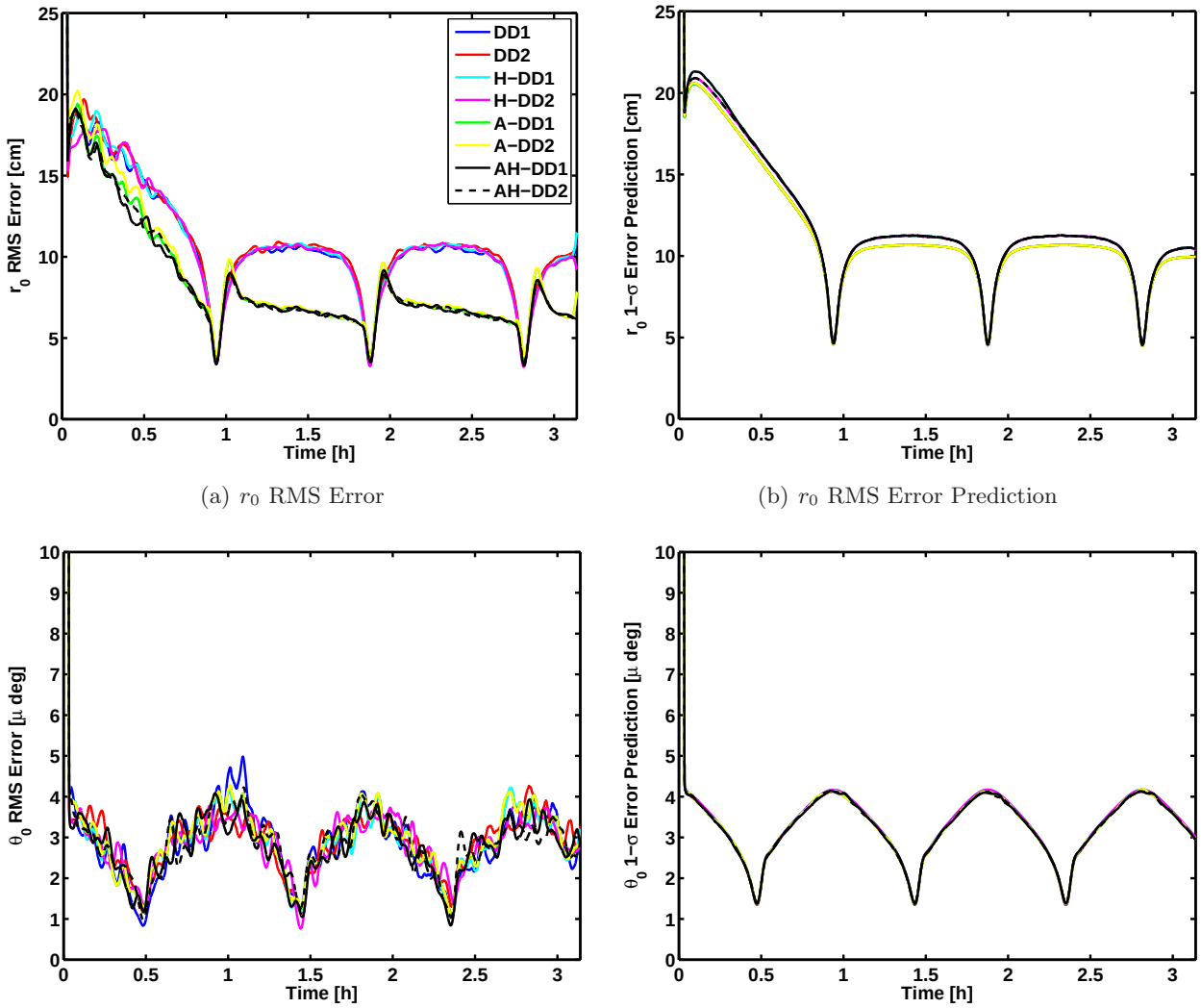


(e) β_3 RMS Error



(f) β_3 RMS Error Prediction

Figure 5.26: Gaussian Simulation Results: Rate Bias



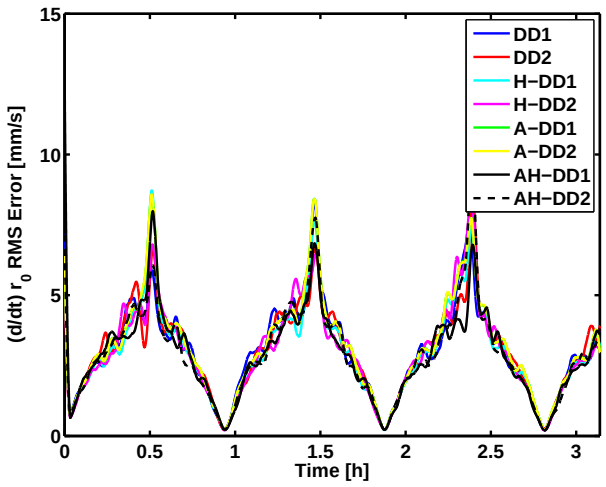
(a) r_0 RMS Error

(b) r_0 RMS Error Prediction

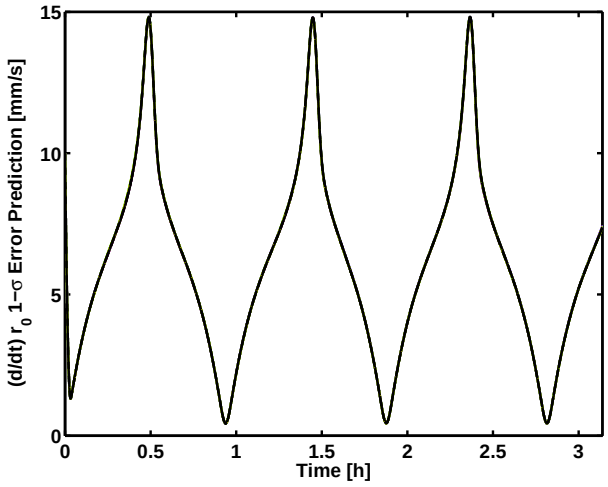
(c) θ_0 RMS Error

(d) θ_0 RMS Error Prediction

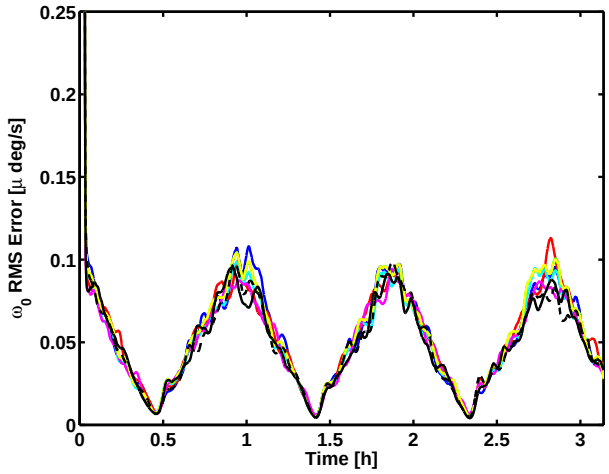
Figure 5.27: Gaussian Simulation Results: Orbit Position



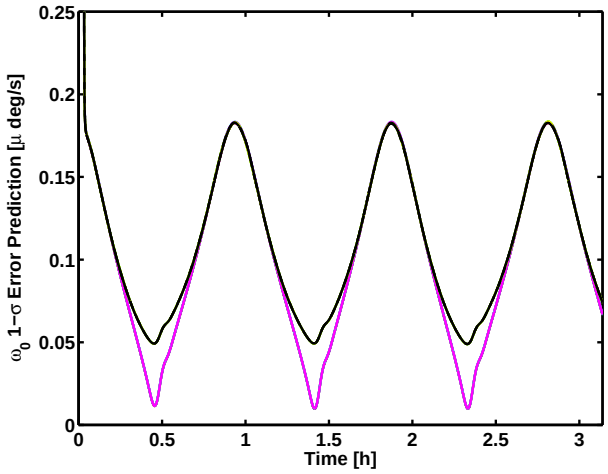
(a) $\dot{r}_0$ RMS Error



(b) $\dot{r}_0$ RMS Error Prediction

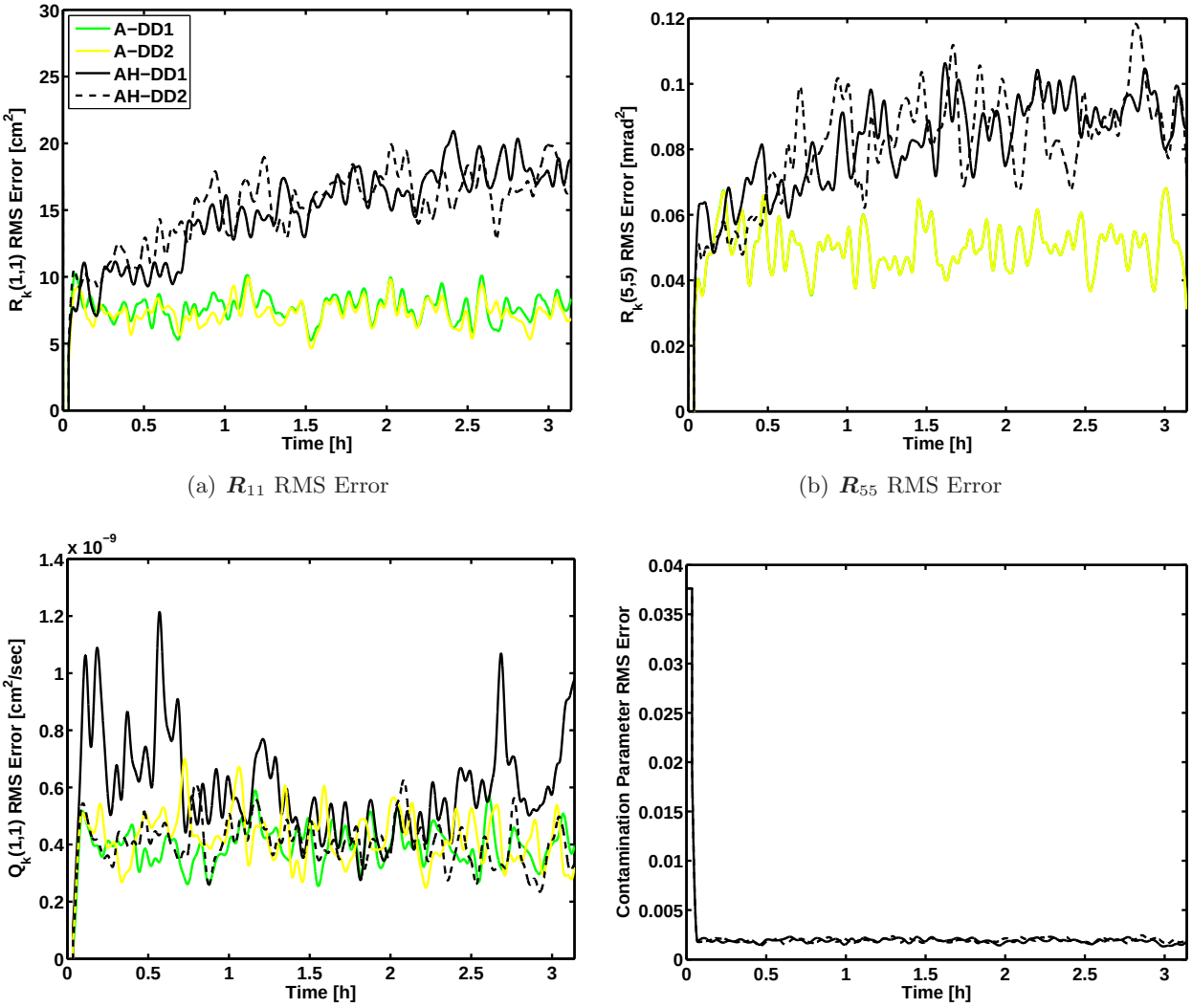


(c) $\dot{\theta}_0$ RMS Error



(d) $\dot{\theta}_0$ RMS Error Prediction

Figure 5.28: Gaussian Simulation Results: Orbit Velocity



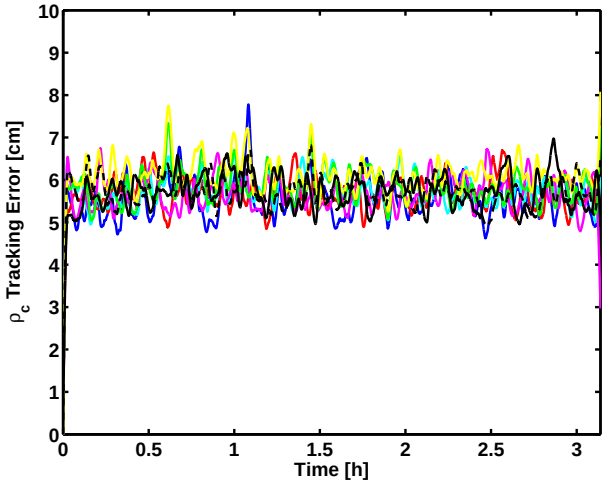
(a) $\boldsymbol{R}_{11}$ RMS Error

(b) $\boldsymbol{R}_{55}$ RMS Error

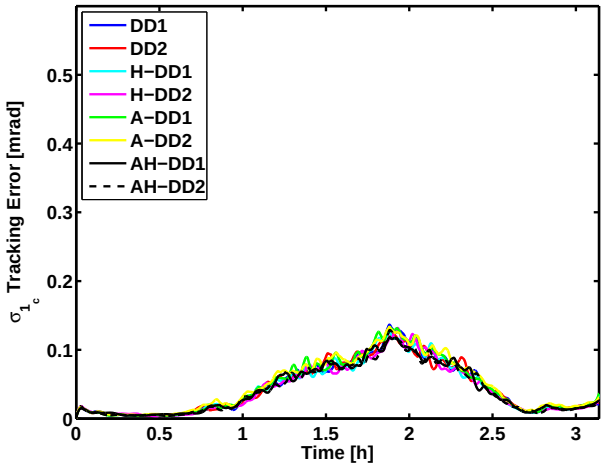
(c) $\boldsymbol{Q}_{11}$ RMS Error

(d) $\hat{e}$ RMS Error

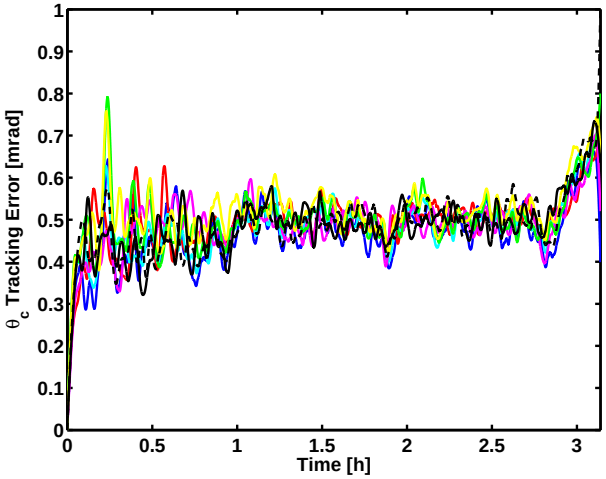
Figure 5.29: Gaussian Simulation Results: Covariance and Contamination



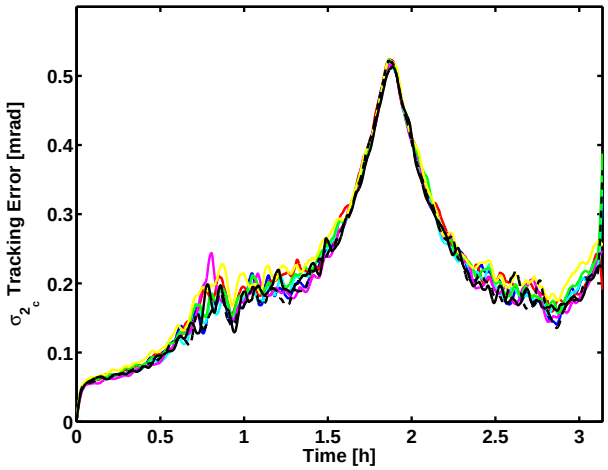
(a) ρ_c RMS Error



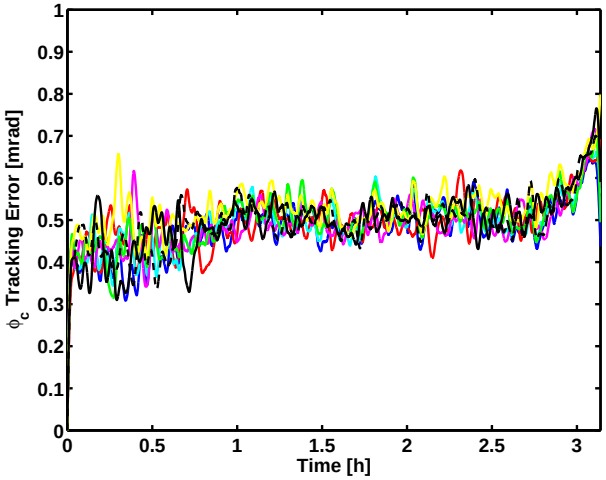
(b) σ_{1_c} RMS Error



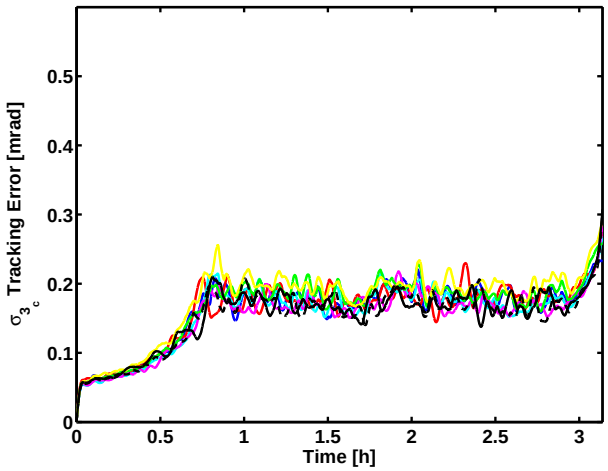
(c) θ_c RMS Error



(d) σ_{2_c} RMS Error



(e) ϕ_c RMS Error



(f) σ_{3_c} RMS Error

Figure 5.30: Gaussian Simulation Results: Closed–Loop System Performance

5.4.3 Non-Gaussian Simulation

This section describes the results of Monte-Carlo simulations conducted for the same rendezvous scenario as described in the previous section, but with non-Gaussian noise components. In this case, the noise samples are drawn from a Gaussian mixture model with 15% contamination from the higher variance distribution. In an effort to stress the filters, the standard deviations of the primary Gaussian density of the mixture model were set to ten times the values provided in Table 5.9, with the contaminating densities an additional five times higher. Additionally, the true standard deviations are scaled according to a uniform distribution to be up to twice as high as the assumed standard deviations in the navigation filter, and uniform random correlation coefficients were assigned to the true errors on each run. The mean initial conditions of this simulation is the same as that given in Table 5.8, but the true random initial errors are generated according to a randomly correlated covariance matrix with standard deviations set to ten times the value in Table 5.8.

The RMS errors of the position variables are shown in Fig. 5.31. As in the previous section, the actual RMS errors calculated from the Monte-Carlo simulation are shown in the left column whereas the mean predicted RMS errors based on the state covariance matrix are shown in the right column. In this non-Gaussian problem, the differences between the various filters becomes much more apparent. In particular the non-robust filters suffer greatly from the contamination present in this simulation. In the range variable ρ, the DD1 filters shows the worst overall performance, closely followed by the DD2 filter. Interestingly, the A-DD1 filters exhibits large errors over the first 0.5 hrs of the simulation before finally settling to the steady state error. The A-DD2 filter benefits from the inclusion of second-order terms in the system and measurement dynamics and as a result, does not exhibit the large initial errors. The A-DD2 shows similar performance to the the Huber-based filters, H-DD1 and H-DD2, which are nearly identical in this problem, with the differences between these two filters being comparable to the differences between the Gaussian DD1 and DD2 filters, though the RMS errors are smaller by roughly 25%. The best performing filters are the AH-DD1 and AH-DD2 filters, which are essentially identical in this variable. The predicted RMS errors show that the Gaussian filters DD1 and DD2 produce highly optimistic error estimates, having the smallest error predictions yet the largest actual RMS error. The non-adaptive Huber estimates, H-DD1 and H-DD2, correctly show slightly larger error estimates. The adaptive filter A-DD1 has large initial error estimates, which correctly reflect the large actual errors although the magnitudes are quite different, with the predicted error being much smaller in magnitude. The A-DD2, AH-DD1, and AH-DD2 filters produce the most accurate error predictions in this particular variable. These same trends can be observed in the other position variables θ and ϕ, as shown in Fig. 5.31, as well as the velocity variables shown in Fig. 5.32.

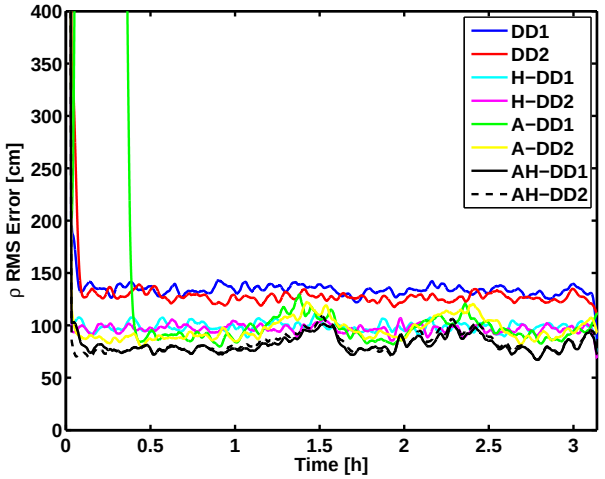
The MRP attitude RMS error components are shown in Fig. (5.33). Here the DD1 filter exhibits large attitude errors over the first hour of the simulation. The DD1 filter also has poor covariance performance in the sense that the actual attitude errors are large yet the predicted errors are the smallest of all the filters implemented for this example. The A-DD1 and A-DD2 filters also exhibit large errors but for the most part they are superior to the DD1 filter. The robust AH-DD2, AH-DD1, H-DD2, and H-DD1 filters exhibit the best attitude estimation performance. The adaptive filters have higher covariance estimates which better match the actual filter performance. These same trends and relative performance comparisons are also evident in the rate bias estimation errors

shown in Fig. (5.34).

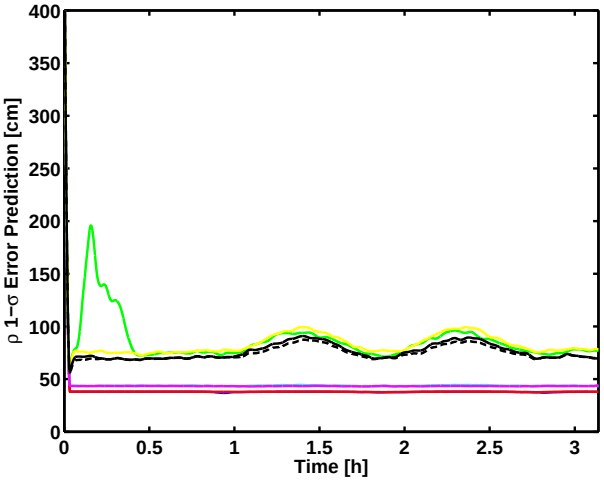
The reference orbit position estimates are shown in Fig. (5.35). Here the A-DD1 and DD1 filters exhibit the worst overall performance. The adaptive Huber–based filters AH-DD1 and AH-DD2 filters perform the best. The nonadaptive Huber–based filters H-DD1 and H-DD2 have similar performance to the adaptive counterparts but with slightly higher errors. The H-DD2 and AH-DD2 filters reach their steady state errors much more rapidly than the other filters due to the inclusion of second–order terms in the filter. This behavior is even more apparent in the velocity components shown in Fig. (5.36).

Results of the covariance and contamination estimation for the adaptive filters are shown in Fig. (5.37). The same covariance elements are chosen for comparison here as in the Gaussian simulation results shown in Fig. (5.29), namely the (1,1) and (5,5) element of the measurement covariance matrix, the (1,1) element of the process noise covariance matrix, and the contamination parameter estimate. The RMS errors in these plots indicate that the A-DD1 filter has relatively inaccurate covariance estimates over the first 0.5 hrs of simulation time for range measurement error and the range process noise. Once converged to its steady state error level the magnitudes remain higher than the A-DD2 filter, which is expected since the A-DD2 filter has the benefit of second-order terms in the process and measurement transformations. The AH-DD1 and AH-DD2 filters are nearly identical in performance with the AH-DD2 filter performing slightly better.

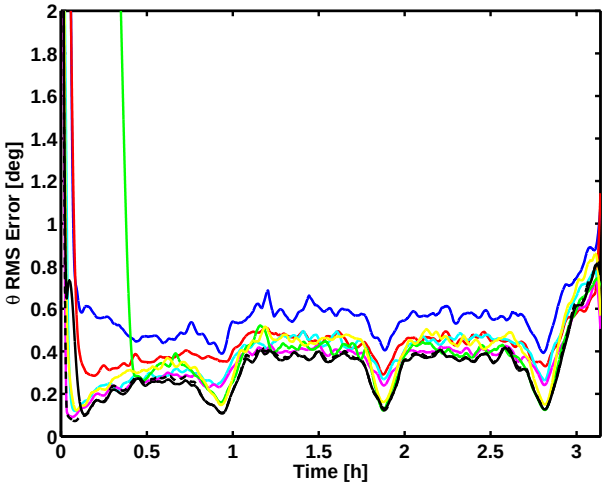
The closed-loop system performance results are shown in Fig. (5.38). Here it can be seen that the DD1, DD2, A-DD1, and H-DD1 filter results have poor performance and essentially diverge in the presence of the non-Gaussian noise. The A-DD2 filter also gives erratic performance, which is more apparent in the attitude response. The H-DD2, AH-DD1, and AH-DD2 filters exhibit the best performance and are essentially able to successfully execute the rendezvous maneuver even in the presence of a high degree of non-Gaussian noise with uncertain statistics. This result is interesting since the H-DD2 filter is non-adaptive it is still better able to cope with the errors than the other filters investigated in this study.



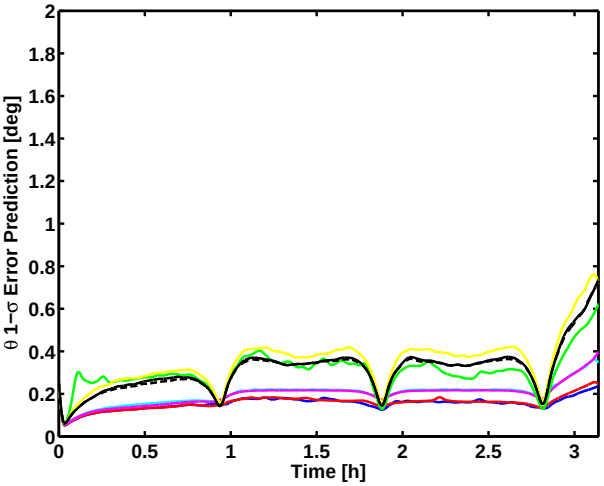
(a) ρ RMS Error



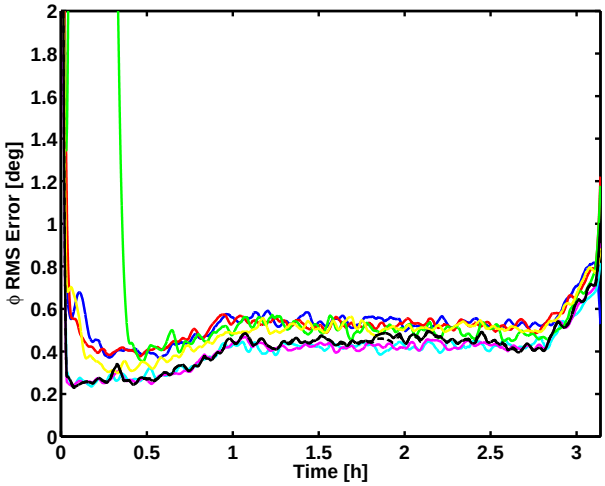
(b) ρ RMS Error Prediction



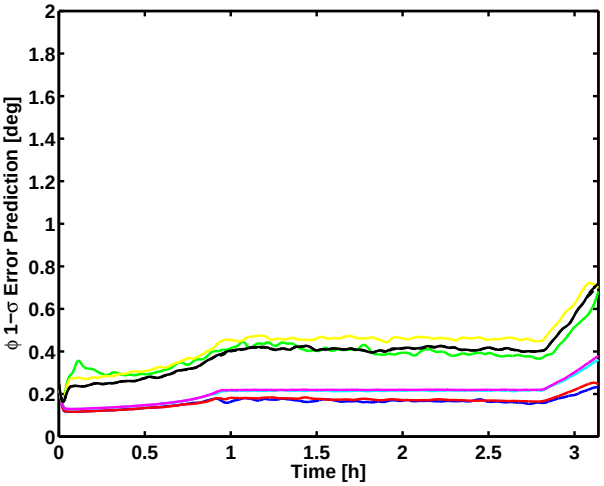
(c) θ RMS Error



(d) θ RMS Error Prediction

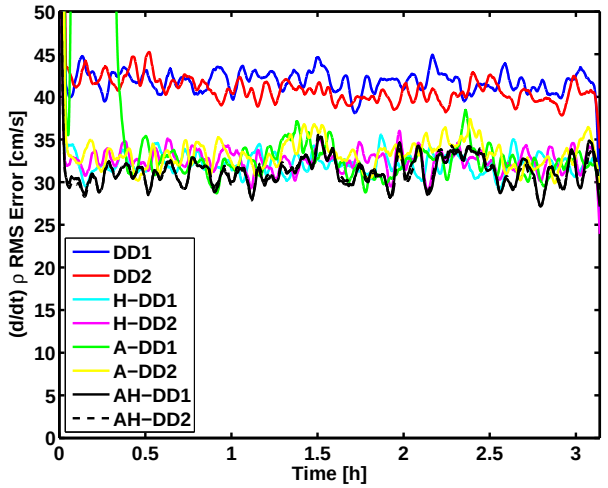


(e) ϕ RMS Error

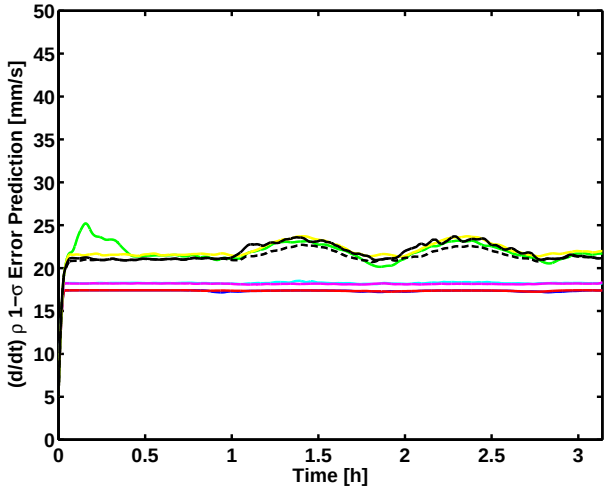


(f) ϕ RMS Error Prediction

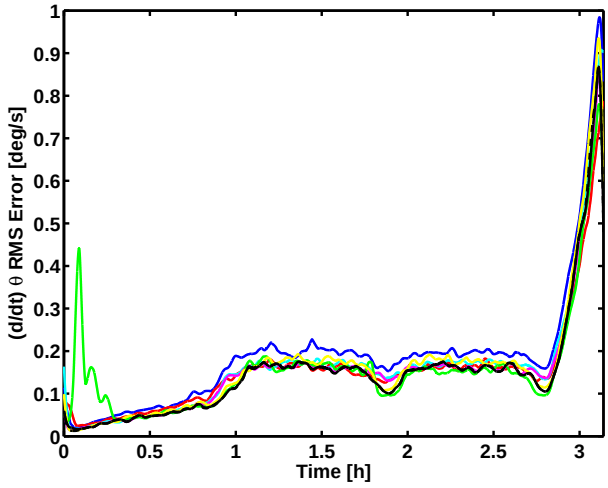
Figure 5.31: Non-Gaussian Simulation Results: Position



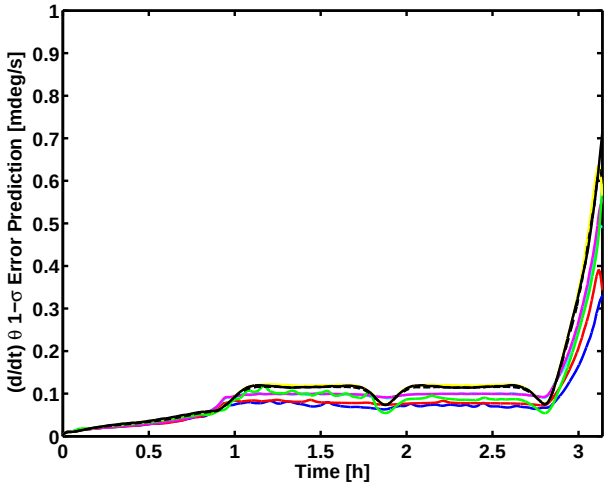
(a) $\dot{\rho}$ RMS Error



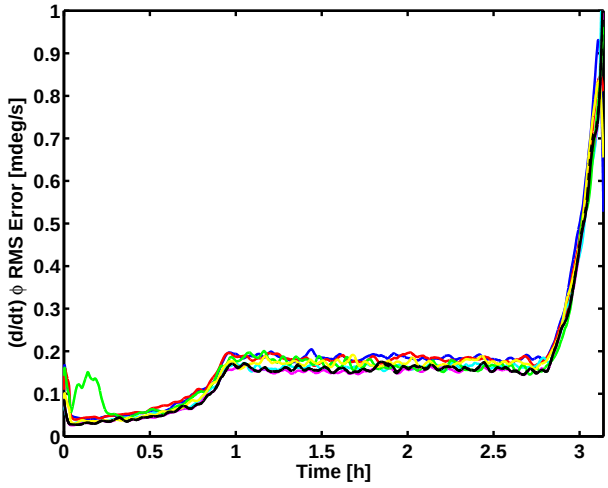
(b) $\dot{\rho}$ RMS Error Prediction



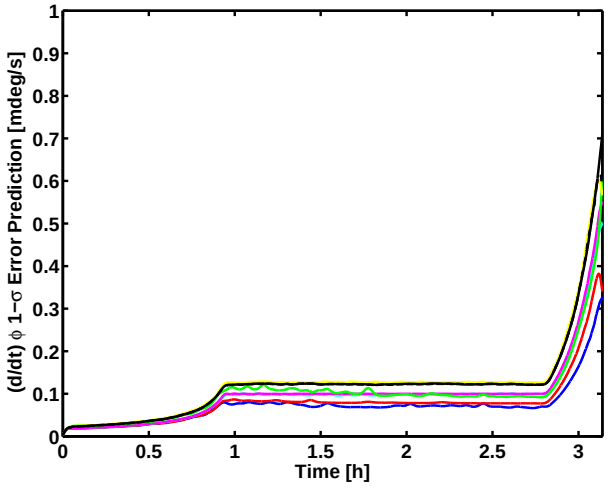
(c) $\dot{\theta}$ RMS Error



(d) $\dot{\theta}$ RMS Error Prediction

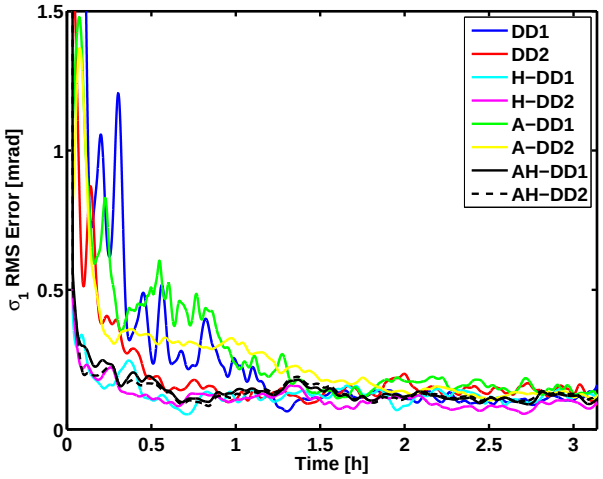


(e) $\dot{\phi}$ RMS Error

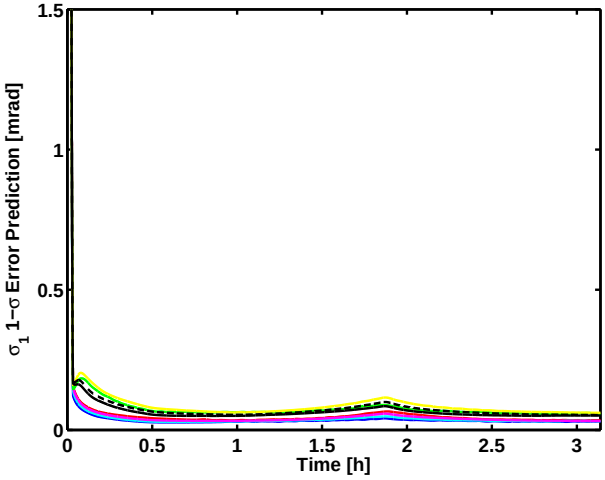


(f) $\dot{\phi}$ RMS Error Prediction

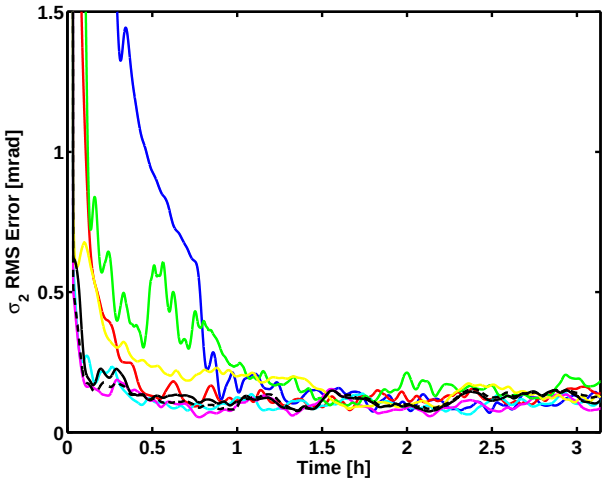
Figure 5.32: Non-Gaussian Simulation Results: Velocity



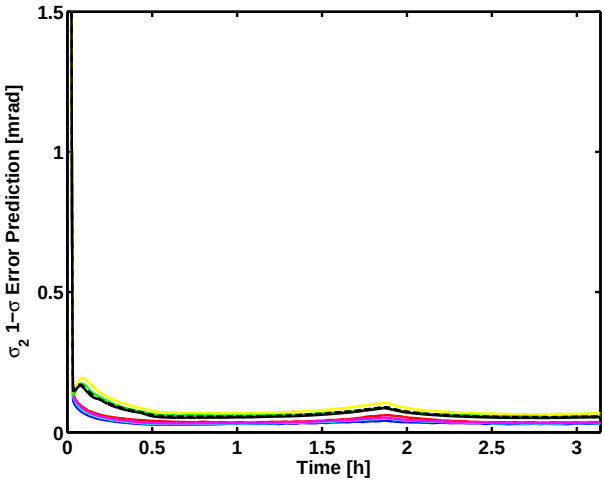
(a) σ_1 RMS Error



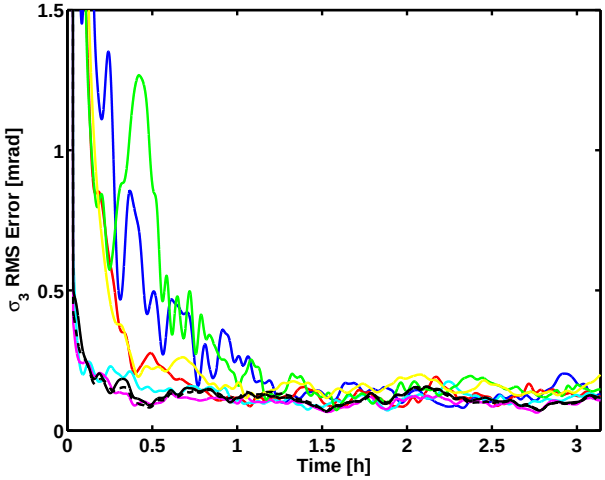
(b) σ_1 RMS Error Prediction



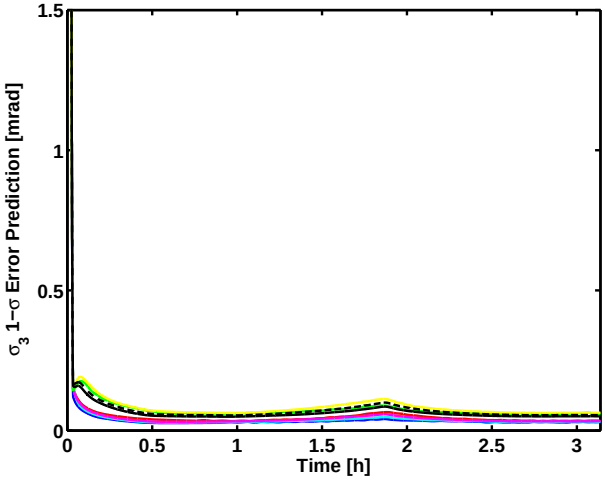
(c) σ_2 RMS Error



(d) σ_2 RMS Error Prediction

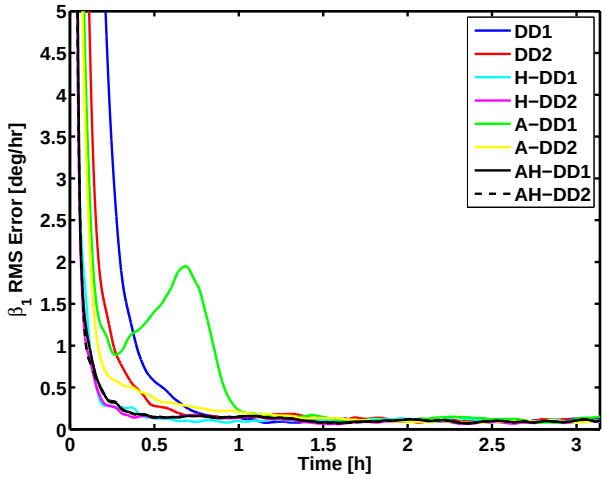


(e) σ_3 RMS Error

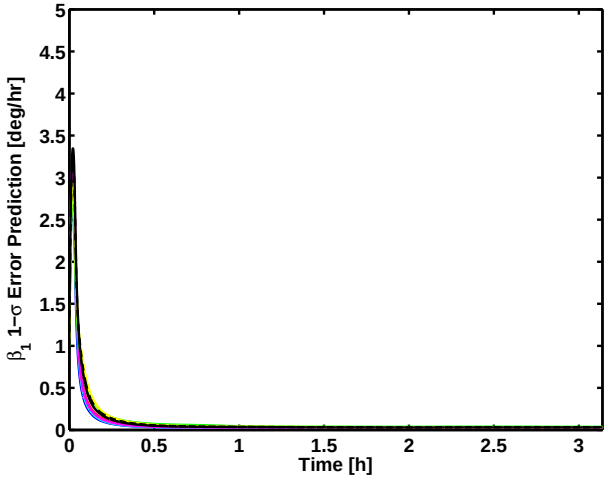


(f) σ_3 RMS Error Prediction

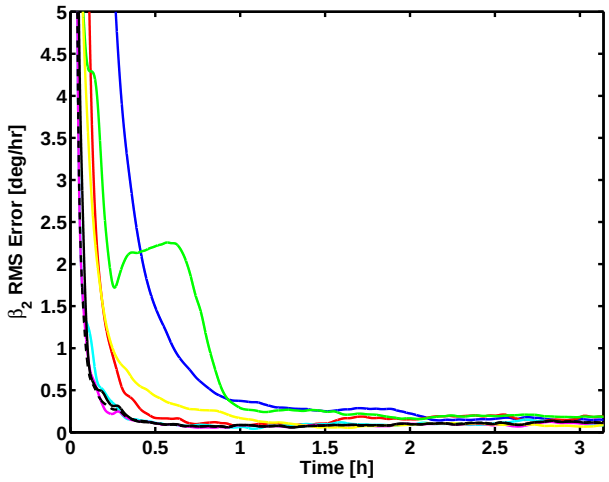
Figure 5.33: Non-Gaussian Simulation Results: Attitude



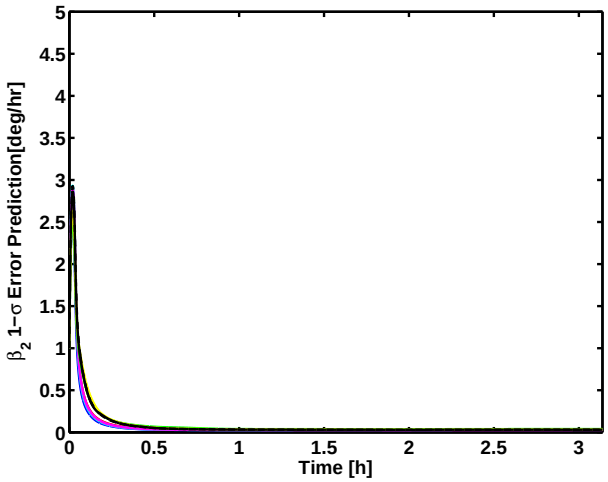
(a) β_1 RMS Error



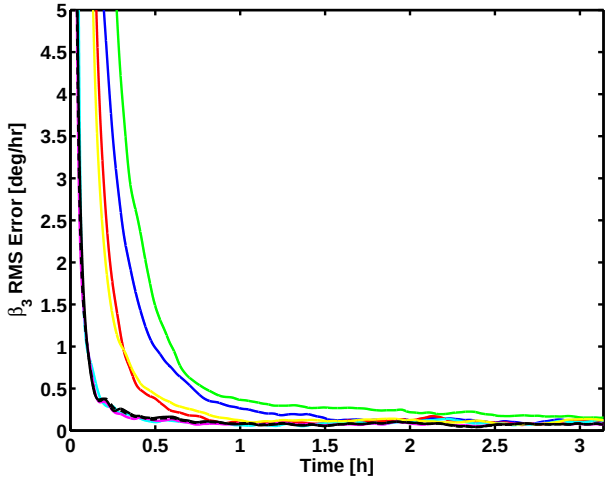
(b) β_1 RMS Error Prediction



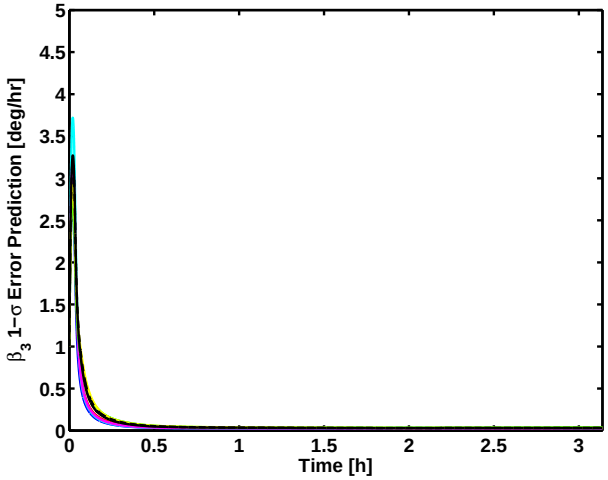
(c) β_2 RMS Error



(d) β_2 RMS Error Prediction

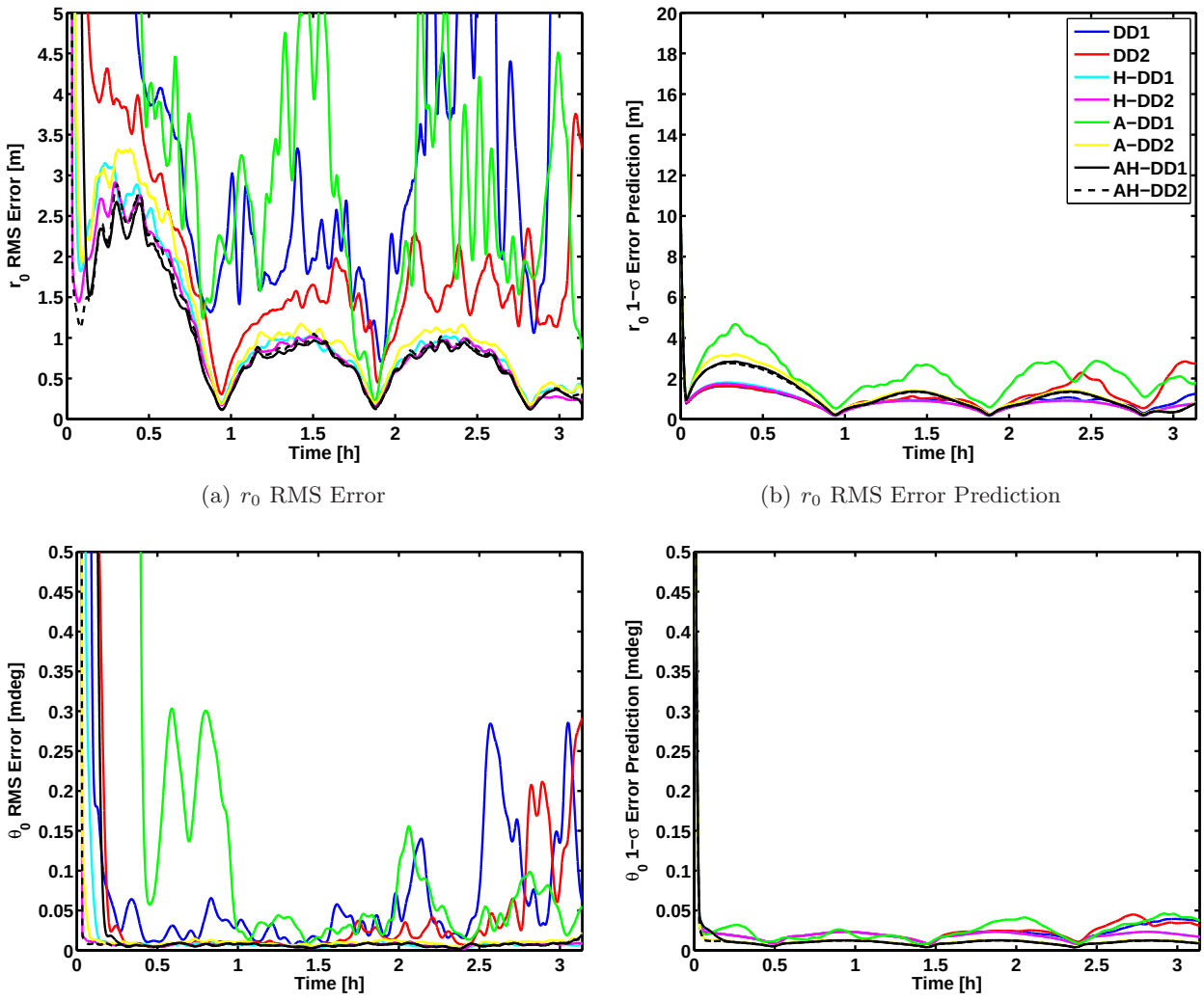


(e) β_3 RMS Error



(f) β_3 RMS Error Prediction

Figure 5.34: Non-Gaussian Simulation Results: Rate Bias



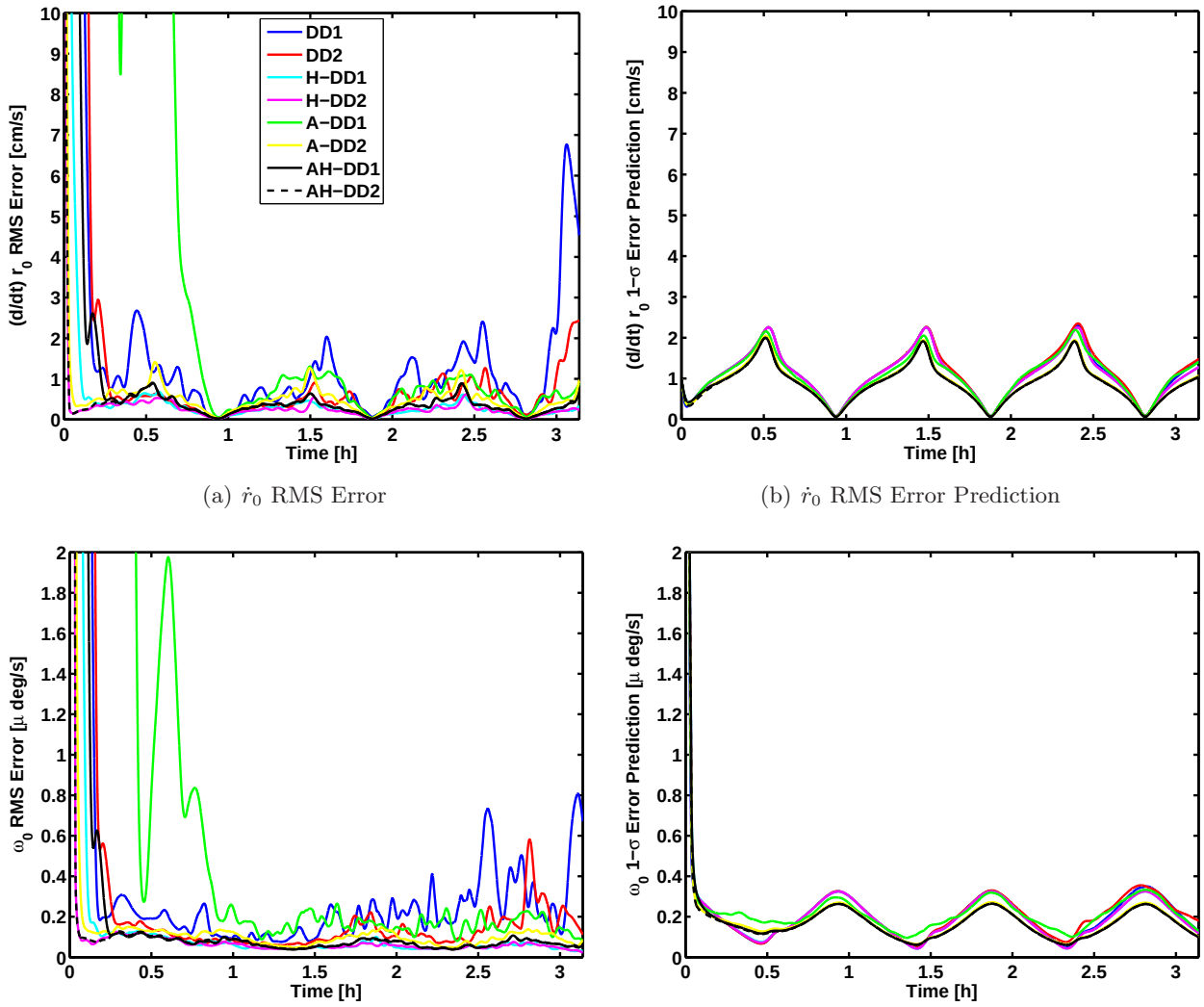
(a) r_0 RMS Error

(b) r_0 RMS Error Prediction

(c) θ_0 RMS Error

(d) θ_0 RMS Error Prediction

Figure 5.35: Non-Gaussian Simulation Results: Orbit Position



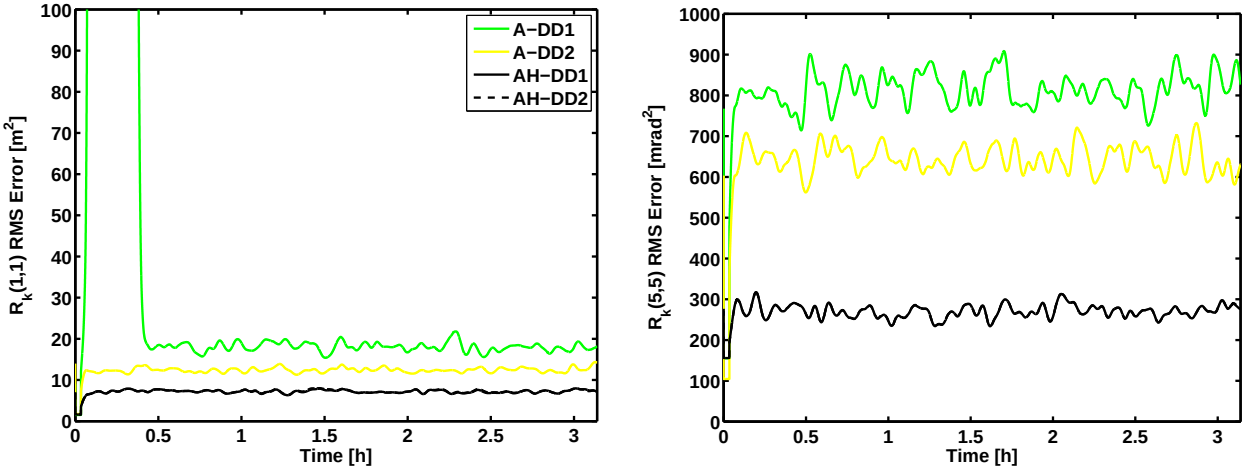
(a) $\dot{r}_0$ RMS Error

(b) $\dot{r}_0$ RMS Error Prediction

(c) $\dot{\theta}_0$ RMS Error

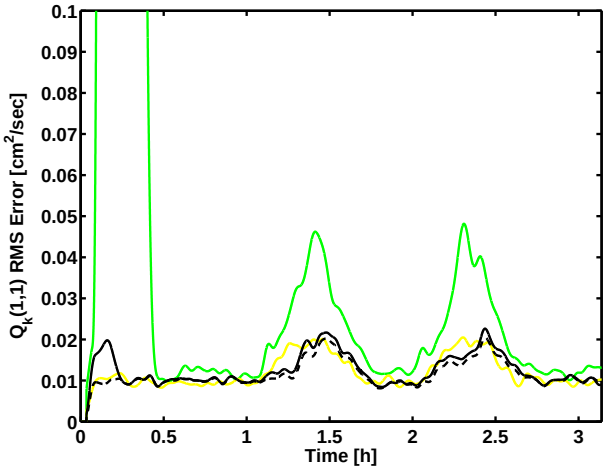
(d) $\dot{\theta}_0$ RMS Error Prediction

Figure 5.36: Non-Gaussian Simulation Results: Orbit Velocity

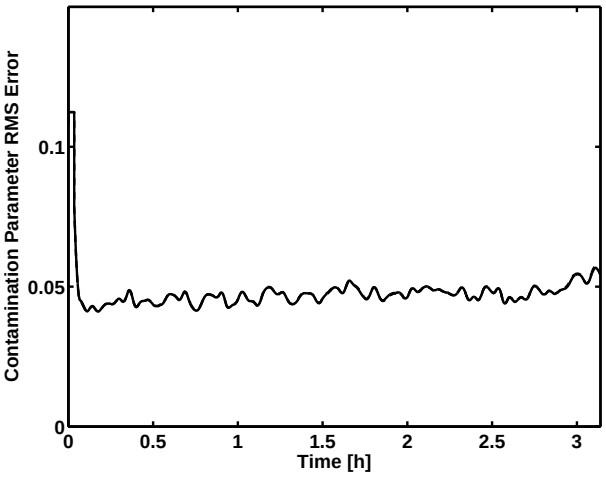


(a) $\boldsymbol{R}_{11}$ RMS Error

(b) $\boldsymbol{R}_{55}$ RMS Error

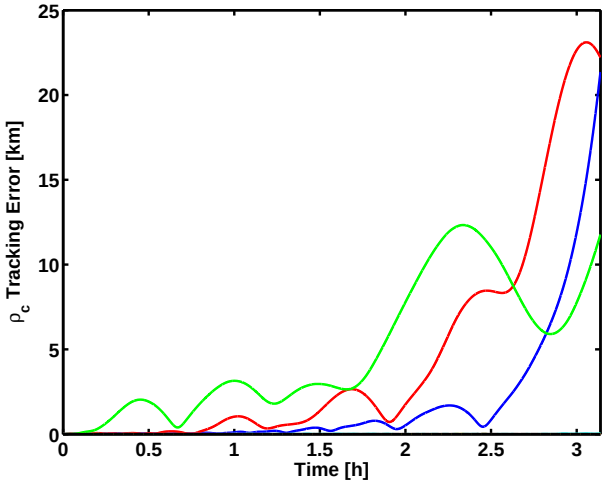


(c) $\boldsymbol{Q}_{11}$ RMS Error

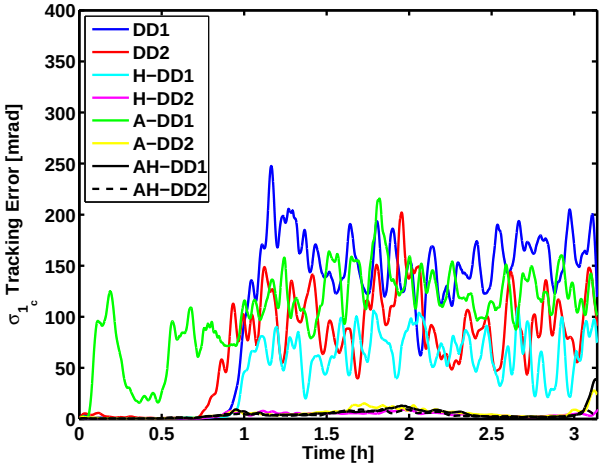


(d) $\hat{e}$ RMS Error

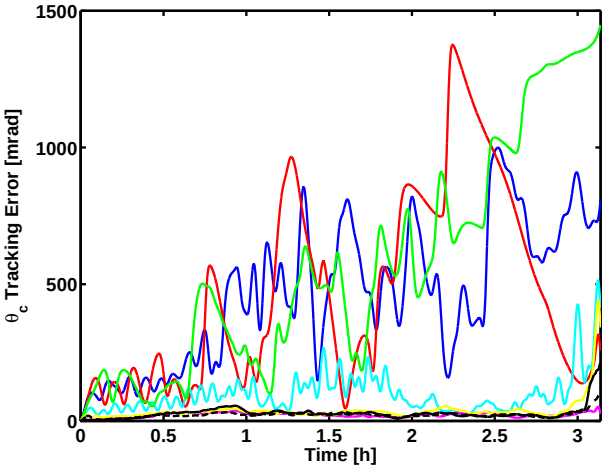
Figure 5.37: Non-Gaussian Simulation Results: Covariance and Contamination



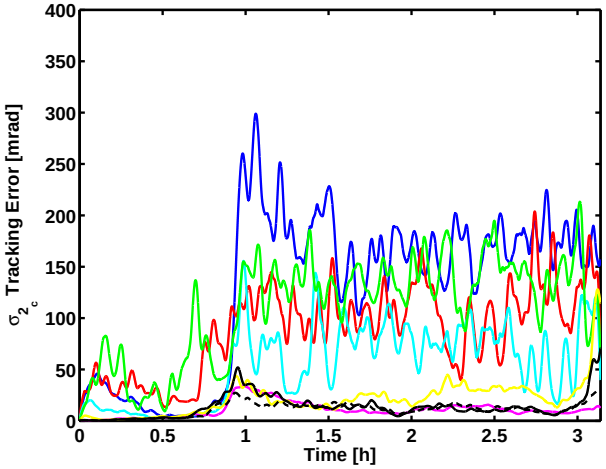
(a) ρ_c RMS Error



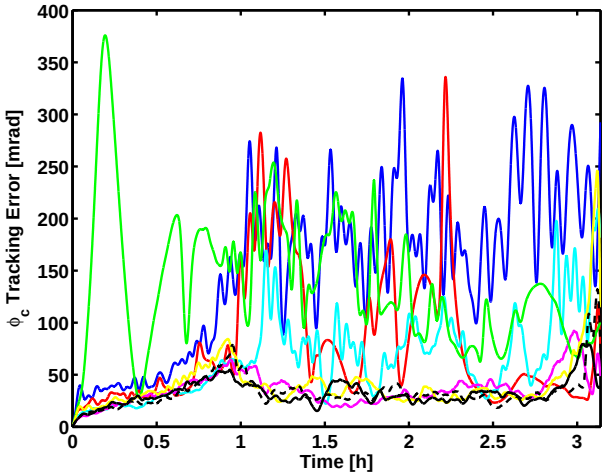
(b) σ_{1_c} RMS Error



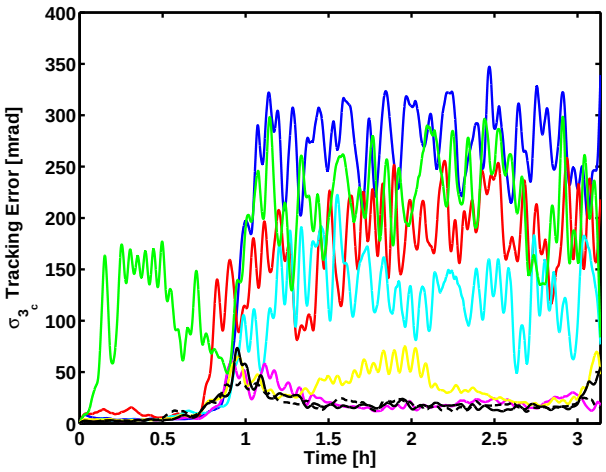
(c) θ_c RMS Error



(d) σ_{2_c} RMS Error



(e) ϕ_c RMS Error



(f) σ_{3_c} RMS Error

Figure 5.38: Non-Gaussian Simulation Results: Closed–Loop System Performance

Table 5.10: Computational Time Comparisons

Filter	EKF	DD1	DD2
Kalman	1.000	1.938	2.124
Huber-Kalman	1.066	2.086	2.272
Adaptive-Kalman	1.526	2.302	2.526
Adaptive-Huber-Kalman	2.961	4.330	4.659

5.4.4 Computational Comparisons

The relative computational costs of the various filters is provided in Table 5.10. Here, the median computational time is computed for each filter and then normalized by the EKF median computational time to provide a relative cost comparison. The trends observed follow closely with those provided in previous sections. In particular, the DD1 and DD2 filters cost roughly twice that of the EKF. The DD2 filter costs only slightly more than the DD1 filter. The Huber-based filters each cost roughly 7% more than the standard Kalman implementations. In the adaptive case, the adaptive Huber filters cost roughly 80% more than the standard Myers-Tapley adaptive implementations.

5.4.5 Summary

In summary, this section provided numerical examples of the robust/adaptive filters applied to the elliptical orbit rendezvous problem for both Gaussian and non-Gaussian noise cases. In the Gaussian case, the performance of the robust filters was found to be nearly the same as that of the standard Kalman type implementations. For the non-Gaussian case, however, the robust filters performed considerably better than the Kalman implementations. The closed-loop system performance results indicate that the DD1, DD2, A-DD1, A-DD2, and H-DD1 filters produced poor rendezvous maneuvering results whereas the H-DD2, AH-DD1, and AH-DD2 filters were able to perform much more successfully. The performance of the H-DD2 filter is especially interesting since it is able to successfully navigate even with erroneous assumed values of the noise statistics and is able to do so with a lower computational cost. The adaptive-Huber filters do provide the extra capabilities to estimate the noise statistics and contamination levels which, aside from robustness considerations, are of interest for sensor performance evaluation and depending on the application the extra computational costs may be deemed worthwhile.

Chapter 6

Conclusions

This dissertation discusses the development of an adaptive discrete-time robust filtering technique based on a recursive form of Huber's mixed minimum ℓ_1/ℓ_2 norm approach, which is robust with respect to deviations from the assumed Gaussian error probability distributions inherent to the Kalman filter. The recursive estimation procedures are developed by recasting the discrete-time filtering problem into the form of a linear regression between the state prediction and the observed quantity, to be solved at each measurement update. The robust estimation approach is developed and applied to both the standard Extended Kalman Filter as well to the First and Second–Order Divided Difference Filter framework, which is an example of a relatively new class of nonlinear filter called the Sigma-Point Kalman Filter. The research described in this dissertation marks the first instance of Huber's robust estimation technique to nonlinear filters of this type.

Additionally, adaptive schemes based on generalized covariance matching is introduced whereby the filter can estimate the process noise and measurement noise covariance matrices along with the state estimate and state estimate error covariance matrix using a buffer of stored residuals. The adaptation technique adopts a robust approach to estimating these covariances that can resist the effects of outliers, based on the use of *projection statistics*, which have been previously developed for robust outlier identification as generalizations of the classical Mahalanobis distance measures but have not been applied to the adaptive state estimation problem in such a manner before.

The robust filters were applied to several benchmark problems, including the estimation of the trajectory of an entry body from discrete–time, noisy range measurement data provided by a radar tracking system and the Multiplicative Quaternion attitude estimation problem. The simulations are conducted using Monte–Carlo techniques with both Gaussian and non–Gaussian error distributions in order to asses the performance of the filtering techniques. The results show that the proposed adaptive Huber filer can achieve greater accuracies than the Kalman filter in situations where unexpected non–Gaussian contaminating noise is present. The amount of improvement depends on the ratio of contamination but for the case of the 10% contamination, the adaptive Huber technique can improve accuracies of the Kalman filter by 50% or better. Furthermore, the adaptive Huber-based filters can successfully estimate the noise covariances in the non-Gaussian case as well as the contamination ratio whereas the adaptive Kalman filter is not capable of doing so. Therefore, the adaptive Huber filter has better consistency and is self–tuning.

Next, the hybrid robust/adaptive filtering approaches were applied to the 6 degree of freedom elliptical orbit rendezvous problem. In this problem, the navigation filters are used "real–time"

inside the control loop as state observers whereas the previous examples have considered only open loop estimation problems where the state estimator does not interact with the dynamics of the problem. The full nonlinear equations of relative motion are formulated in spherical coordinates centered on the target orbit. A relatively simple control law based on feedback linearization is used to track a desired rendezvous trajectory. The attitude dynamics are parameterized using Modified Rodrigues Parameters, which are advantageous for both control law development and estimation since they constitute a minimal 3-parameter attitude description. A switching technique which exploits the stereographic projection properties of the MRP coordinate is utilized to avoid singularities which inevitably arise in minimal attitude descriptions. This dissertation also introduces the proper covariance transformations associated with the singularity avoidance switching technique, which provides a novel means for globally non-singular attitude estimation with a minimal attitude description. An attitude control law based on backstepping is employed to track the target vehicle.

A sensor suite consisting of a generic lidar or optical sensor, an Inertial Measurement Unit, consisting of accelerometers and gyroscopes, a star tracker, and a horizon sensor are utilized to provide measurement data to the navigation filters so that the chaser vehicle can estimate its relative state during the rendezvous maneuver. Several filters will be implemented for comparison, including the Extended Kalman Filter, First and Second–Order Divided Difference Filters and Huber–based generalizations of these filters that include adaptive techniques for estimating the noise covariances. Monte-Carlo simulations are presented which include both Gaussian and non-Gaussian errors, including mismatches in the assumed noise covariances in the navigation filters in order to illustrate the benefits of the robust/adaptive nonlinear filters. Additionally, computational burdens of the various filters is compared. The results indicate that the robust filters give rise to essentially the same closed-loop performance as their non-robust counterparts for purely Gaussian noise simulations. However for the non-Gaussian problem, the results of the hybrid robust/adaptive filters is superior to the non-robust filters by a considerable margin.

It is anticipated that the techniques introduced in this dissertation will be beneficial to a wide range of linear and non-linear filtering, estimation, navigation, and sensor fusion problems in order to reduce the sensitivity of the technique with respect to non-Gaussian noise components with uncertain statistics. Also the globally non-singular technique for attitude estimation using Modified Rodrigues Parameters is applicable to a variety of applications including spacecraft and aircraft attitude estimation.

Bibliography

[1] Houbolt, J. C., "Problems and Potentialities of Space Rendezvous," *Astronautica Acta*, Vol. 7, Nos. 5/6, 1961, pp. 406–429.

[2] Polites, M. E., "Technology of Automated Rendezvous and Capture in Space," *Journal of Spacecraft and Rockets*, Vol. 36, No. 2, 1999, pp. 280–291.

[3] "NASA's Exploration Systems Architecture Study," NASA/TM–2005–214062, November 2005.

[4] Ruoff, C., "Technologies for Mars Exploration and Sample Return," American Institute of Aeronautics and Astronautics, AIAA Paper 1999–4447, September 1999.

[5] Moses, R. W., Van Laak, J., Johnson, S. L., Chytka, T. M., Amundsen, R. M., Dorsey, J. T., Doggett, W. R., Reeves, J. D., Todd, B. K., Stambolian, D. B., and Moe, R. V., "Analysis of In–Space Assembly of Modular Systems," American Institute of Aeronautics and Astronautics, AIAA Paper 2005–2504, January 2005.

[6] Zimpfer, D., Kachmar, P., and Tuohy, S., "Autonomous Rendezvous, Capture, and In-Space Assembly: Past, Present, and Future," American Institute of Aeronautics and Astronautics, AIAA Paper 2005-2523, January 2005.

[7] Shen, H. and Tsiotras, P., "Peer-to-Peer Refuelling for Circular Satellite Constellations," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 6, 2005, pp. 1220–1230.

[8] Machula, M. F. and Sandhoo, G. S., "Rendezvous and Docking for Space Exploration," American Institute of Aeronautics and Astronautics, AIAA Paper 2005-2716, January 2005.

[9] Rumford, T. E., "Demonstration of Autonomous Rendezvous Technology (DART) Project Summary," *Space Systems and Operations*, Tchoryk, P. and Shoemaker, J., (Eds.), SPIE Proceedings, Vol. 5088, 2003, pp. 10–19.

[10] "Overview of the DART Mishap Investigation Results," NASA Marshall Space Flight Center, May 15, 2006.

[11] Ruth, M. and Tracy, C., "Video–Guidance Design for the DART Rendezvous Mission," *Spacecraft Platforms and Infrastructure*, SPIE Proceedings, Vol. 5419, 2004, pp. 92–106.

[12] Clohessy, W. H. and Wiltshire, R. S., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 5, 1960, pp. 653–658, 674.

[13] Kawano, I., Mokuno, M., Kasai, T., and Suzuki, T., "Result of Autonomous Rendezvous Docking Experiment of Engineering Test Satellite VII," *Journal of Spacecraft and Rockets*, Vol. 38, No. 1, 2001, pp. 105–111.

[14] Goodman, J., "Lessons Learned From Seven Space Shuttle Missions," NASA/CR–2007–213697, January 2007.

[15] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of*

Basic Engineering, Vol. 82, No. 1, 1960, pp. 35–50.

[16] Swerling, P., "First–Order Error Propagation in a Stagewise Smoothing Procedure for Satellite Observations," *Journal of the Astronautical Sciences*, Vol. 6, No. 3, 1959, pp. 46–52.

[17] Sorenson, H. W., "Least–Squares Estimation: From Gauss to Kalman," *IEEE Spectrum*, Vol. 7, No. 7, 1970, pp. 63–68.

[18] Meinhold, R. J. and Singpurwalla, N. D., "Understanding the Kalman Filter," *American Statistician*, Vol. 37, No. 2, 1983, pp. 123–127.

[19] Tukey, J. W., "A Survey of Sampling from Contaminated Distributions," *Contributions to Probability and Statistics*, Olkin, I., Ghurye, S., Hoeffding, W., Madow, W., and Mann, H. (Eds.), Stanford University Press, Stanford, CA, 1960, pp. 448–485.

[20] Schick, I. C. and Mitter, S. K., "Robust Recursive Estimation in the Presence of Heavy-Tailed Observation Noise," *Annals of Statistics*, Vol. 22, No. 2, 1994, pp. 1045–1080.

[21] Box, G. E. P., "Non–Normality and Tests on Variances," *Biometrika*, Vol. 40, Nos. 3/4, 1953, pp. 318–335.

[22] Huber, P. J., *Robust Statistics*, Wiley, New York, 1981, pp. 43–106, 153–198.

[23] Hampel, F. R., Rousseeuw, P. J., Ronchetti, E., and Stahel, W. A., *Robust Statistics: The Approach Based on Influence Functions*, Wiley, New York, 1986, pp. 78–186.

[24] Rousseeuw, P. J. and Leroy, A. M., *Robust Regression and Outlier Detection*, John Wiley and Sons, New York, NY, 1987, pp. 158–174.

[25] Maronna, R. A., Martin, R. D., and Yohai, V. J., *Robust Statistics: Theory and Methods*, John Wiley & Sons, Ltd., West Sussex, England, 2006.

[26] Plackett, R. L., "The Discovery of the Method of Least Squares," *Biometrika*, Vol. 59, No. 2, 1972, pp. 239–251.

[27] Stigler, S. M., "Gauss and the Invention of Least Squares," *Annals of Statistics*, Vol. 9, No. 3, 1981, pp. 465–474.

[28] Stigler, S. M., "Mathematical Statistics in the Early States," *Annals of Statistics*, Vol. 6, No. 2, 1978, pp. 239–265.

[29] Stigler, S. M., "Simon Newcomb, Percy Daniell, and the History of Robust Estimation 1885–1920," *Journal of the American Statistical Association*, Vol. 68, No. 344, 1973, pp. 872–879.

[30] Huber, P. J., "Robust Statistics: A Review," *Annals of Mathematical Statistics*, Vol. 43, No. 4, 1972, pp. 1041–1067.

[31] Eisenhart, C., "Laws of Error II: The Gaussian Distribution," *Encyclopedia of Statistical Science*, Kotz, S. and Johnson, N. L., (Eds.), Wiley, New York, NY, 1983, pp. 547–566.

[32] Stigler, S. M., "Laplace's 1774 Memoir on Inverse Probability," *Statistical Science*, Vol. 1, No. 3, 1986, pp. 359–363.

[33] Stigler, S. M., "Boscovich, Simpson, and 1760 Manuscript Note on Fitting a Linear Relation," *Biometrika*, Vol. 71, No. 3, 1984, pp. 615–620.

[34] Portney, S. and Koenker, R., "The Gaussian Hare and the Laplacian Tortoise: Computability of Squared–Error versus Absolute–Error Estimators," *Statistical Science*, Vol. 12, No. 4, 1997, pp. 279–296.

[35] Hampel, F., "Robust Statistics: A Brief Introduction and Overview," Symposium on Robust Statistics and Fuzzy Techniques in Geodesy and GIS, ETH Zurich, March 2001.

[36] Beckman, R. J. and Cook, R. D., "Outliers," *Technometrics*, Vol. 25, No. 2, 1983, pp. 119–149.

[37] Newcomb, S., "A Generalized Theory of the Combination of Observations so as to Obtain the

Best Result," *American Journal of Mathematics*, Vol. 8, No. 4, 1886, pp. 343–366.

[38] Huber, P. J., "Robust Estimation of a Location Parameter," *Annals of Mathematical Statistics*, Vol. 35, No. 2, 1964, pp. 73–101.

[39] Newcomb, S., "Researches on the Motion of the Moon, II," *Astronomical Papers*, U.S. Nautical Almanac Office, Vol. 9, pp. 1–249, 1912.

[40] Huber, P. J., "Robust Regression: Asymptotics, Conjectures and Monte Carlo," *Annals of Statistics*, Vol. 1, No. 5, 1973, pp. 799–821.

[41] Huber, P. J., "Minimax Aspects of Bounded–Influence Regression," *Journal of the American Statistical Association*, Vol. 78, No. 381, 1983, pp. 66–72.

[42] Boncelet, C. G. and Dickinson, B. W., "An Approach to Robust Kalman Filtering," *22nd IEEE Conference on Decision and Control*, Institute of Electrical and Electronics Engineers, New York, NY, 1983, pp. 304–305.

[43] Kovacevic, B., Durovic, Z., and Glavaski, S., "On Robust Kalman Filtering," *International Journal of Control*, Vol. 56, No. 3, 1992, pp. 547–562.

[44] El-Hawary, F. and Jing, Y., "Robust Regression-Based EKF for Tracking Underwater Targets," *IEEE Journal of Oceanic Engineering*, Vol. 20, No. 1, 1995, pp. 31–41.

[45] Durgaprasad, G. and Thakur, S. S., "Robust Dynamic State Estimation of Power Systems Based on M-Estimation and Realistic Modeling of System Dynamics," *IEEE Transactions on Power Systems*, Vol. 13, No. 4, 1998, pp. 1331–1336.

[46] Yang, T., Lee, J., Lee, K. Y., and Sung, K., "On Robust Kalman Filtering with Forgetting Factor for Sequential Speech Analysis," *Signal Processing*, Vol. 63, No. 2, 1997, pp. 151–156.

[47] Ledoux, C., "Robust Speech Filtering in Impulsive Noise Environments," M.S. Thesis, Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, December 1999.

[48] Karlgaard, C. D., "Robust Rendezvous Navigation in Elliptical Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, 2006, pp. 495–499.

[49] Ghandi, M. A. and Mili, L., "Robust Kalman Filter Based on a Generalized Maximum–Likelihood–Type Estimator," *IEEE Transactions on Signal Processing*, Vol. 58, No. 5, 2010, pp. 2509–2520.

[50] Jeffreys, H., "An Alternative to the Rejection of Observations," *Proceedings of the Royal Society of London, Series A*, Vol. 137, No. 831, 1932, pp. 78–87.

[51] Sorenson, H. W. and Alspach, D. L., "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, Vol. 7, No. 4, 1971, pp. 465–479.

[52] Alspach, D. L. and Sorenson, H. W., "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, Vol. 17, No. 4, 1972, pp. 439–448.

[53] Tanaka, M. and Katayama, T., "Robust Kalman Filter for Linear Discrete–Time System with Gaussian Sum Noises," *International Journal of Systems Science*, Vol. 18, No. 9, 1987, pp. 1721–1731.

[54] Plantaniotis, K. N., Androutsos, D., and Venetsanopoulos, A. N., "Nonlinear Filtering of Non-Gaussian Noise," *Journal of Intelligent and Robotic Systems*, Vol. 19, No. 2, 1997, pp. 201–231.

[55] Tam, W. I., Plantaniotis, K. N., and Hatzinakos, D., "An Adaptive Gaussian Sum Algorithm for Radar Tracking," *Signal Processing*, Vol. 77, No. 1, 1999, pp. 85–104.

[56] Kotecha, J. H. and Djuric, P. M., "Gaussian Particle Filtering," *IEEE Transactions on Signal Processing*, Vol. 51, No. 10, 2003, pp. 2592–2601.

[57] Kotecha,J. H. and Djuric, P. M., "Gaussian Sum Particle Filtering," *IEEE Transactions on Signal Processing*, Vol. 51, No. 10, 2003, pp. 2602–2612.

[58] Masreliez, C. J. and Martin, R. D., "Robust Bayesian Estimation for the Linear Model and Robustifying the Kalman Filter," *IEEE Transactions on Automatic Control*, Vol. 22, No. 3, 1977, pp. 361–371.

[59] Tsai, C. and Kurz, L., "An Adaptive Robustizing Approach to Kalman Filtering," *Automatica*, Vol. 19, No. 3, 1983, pp. 279–288.

[60] Hewer, G. A., Martin, R. D., and Yeh, J., "Robust Preprocessing for Kalman Filtering of Glint Noise," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 23, No. 1, 1987, pp. 120–128.

[61] Meinhold, R. J. and Singpurwalla, N. D., "Robustification of Kalman Filter Models," *Journal of the American Statistical Association*, Vol. 84, No. 406, 1989, pp. 479–486.

[62] Niehsen, W., "Robust Kalman Filtering with Generalized Gaussian Measurement Noise," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 4, 2002, pp. 1409–1412.

[63] Fitzgerald, R. J., "Divergence of the Kalman Filter," *IEEE Transactions on Automatic Control*, Vol. 16, No. 6, 1971, pp. 736–747.

[64] Mehra, R. K., "Approaches to Adaptive Filtering," *IEEE Transactions on Automatic Control*, Vol. 17, No. 5, 1972, pp. 693–698.

[65] Tapley, B. D. and Born, G. H., "Sequential Estimation of the State and the Observation–Error Covariance Matrix," *AIAA Journal*, Vol. 9, No. 2, 1971, pp. 212–217.

[66] Myers, K. A. and Tapley, B.D., "Adaptive Sequential Estimation with Unknown Noise Statistics," *IEEE Transactions on Automatic Control*, Vol. 21, No. 4, 1976, pp. 520–523.

[67] Maybeck, P. S., Jensen, R. L., Harnly, D. A., "An Adaptive Extended Kalman Filter for Target Image Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 17, No. 2, 1981, pp. 173–180.

[68] Whitmore, S. A. and Leondes, C. T., "Formulation and Implementation of a Practical Algorithm for Non–Stationary Adaptive State Estimation," *International Journal of Control*, Vol. 44, No. 3, 1986, pp. 767–775.

[69] Hull, D. G., Speyer, J. L., and Greenwell, W .M., "Adaptive Noise Estimation for Homing Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 3, 1984, pp. 322–328.

[70] Groutage, F. D., Jacqout, R. G., and Smith, D. E., "Adaptive State Variable Estimation Using Robust Smoothing," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 106, No. 4, 1984, pp. 335–341.

[71] Kirlin, R. L. and Moghaddamjoo, A., "Robust Adaptive Kalman Filtering for Systems with Unknown Step Inputs and Non–Gaussian Measurement Errors," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 24, No. 2, 1986, pp. 252–263.

[72] Durovic, Z. M. and Kovacevic, B. D., "Robust Estimation with Unknown Noise Statistics," *IEEE Transactions on Automatic Control*, Vol. 44, No. 6, 1999, pp. 1292–1296.

[73] Zou, Y., Chan, S. C., and Ng, T. S., "Robust M–Estimate Adaptive Filtering," *IEE Proceedings of Vision, Image, and Signal Processing*, Vol. 148, No. 4, 2001, pp. 289–294.

[74] Psiaki, M. L., "Backward–Smoothing Extended Kalman Filter," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, 2005, pp. 885–894.

[75] Haupt, G. T., Kasdin, N. J., Keiser, G. M., and Parkinson, B. W., "Optimal Recursive Iterative Algorithm for Discrete Nonlinear Least–Squares Estimation," *Journal of Guidance, Control,*

and Dynamics, Vol. 19, No. 3, 1996, pp. 643–649.

[76] Kasdin, N. J. and Haupt, G. T., "Second–Order Correction and Numerical Considerations for the Two–Step Optimal Estimator," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 2, 1997, pp. 362–369.

[77] Garrison, J. L., Axelrad, P., and Kasdin, N. J., "Ill–Conditioned Covariance Matrices in the First–Order Two–Step Estimator," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 754–760.

[78] Kasdin, N. J., "New, Guaranteed Positive Time Update for the Two–Step Optimal Estimator," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 2, 2000, pp. 215–221.

[79] Nørgaard, M., Poulsen, N. K. and Ravn, O., "Advances in Derivative-Free State Estimation for Nonlinear Systems," Technical Report IMM-REP-1998-15, Department of Mathematical Modelling, Technical University of Denmark, revised April 2000.

[80] Nørgaard, M., Poulsen, N. K. and Ravn, O., "New Developments in State Estimation for Nonlinear Systems," *Automatica,* Vol. 36, No. 11, 2000, pp. 1627–1638.

[81] Schei, T. S., "A Finite–Difference Method for Linearization in Nonlinear Estimation Algorithms," *Automatica*, Vol. 33, No. 11, 1997, pp. 2053–2058.

[82] Julier, S., Uhlmann, J., and Durrant–Whyte, H. F., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, 2000, pp. 477–482.

[83] Julier, S. J. and Uhlmann, J. K., "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422.

[84] Ito, K. and Xiong, K., "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control,* Vol. 45, No. 5, 2000, pp. 910–927.

[85] Quine, B. M., "A Derivative-Free Implementation of the Extended Kalman Filter," *Automatica*, Vol. 42, No. 11, 2006, pp. 1927-1934.

[86] Li. W., Wei, P., and Xian, X., "An Adaptive Nonlinear Filter of Discrete–Time System with Uncertain Covariance Using Unscented Kalman Filter," International Symposium on Communications and Information Technology, October 2005.

[87] Lefferts, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, 1982, pp. 417–429.

[88] Crassidis, J. L., Markley, F. L., and Cheng, Y., "Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 12–28.

[89] Zanetti, R. and Bishop, R., "Quaternion Estimation and Norm Constrained Kalman Filtering," American Institute of Aeronautics and Astronautics, AIAA Paper 2006–6164, August 2006.

[90] Majji, M. and Mortari, D., "Quaternion Constrained Kalman Filter," American Astronautical Society, AAS Paper 08–215, January 2008.

[91] Zanetti, R., Majji, M., Bishop, R. H., and Mortari, D., "Norm–Constrained Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 5, 2009, pp. 1458–1465.

[92] Schaub, H. and Junkins, J. L., *Analytical Mechanics of Space Systems*, American Institute of Aeronautics and Astronautics, AIAA Education Series, Reston, VA, 2003.

[93] Tsiotras, P. and Longuski, J. M., "A New Parameterization of the Attitude Kinematics," *Journal of the Astronautical Sciences*, Vol. 43, No. 3, 1995, pp. 243–262.

[94] Schaub, H. and Junkins, J. L., "Stereographic Orientation Parameters for Attitude Dynamics: A Generalization of the Rodrigues Parameters," *Journal of the Astronautical Sciences*, Vol. 44,

No. 1, 1996, pp. 1–19.

[95] Schaub, H., Robinett, R. D., and Junkins, J. L., "New Penalty Functions for Optimal Control Formulation for Spacecraft Attitude Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 428–434.

[96] Crassidis, J. L. and Markley, F. L., "Attitude Estimation Using Modified Rodrigues Parameters," Proceedings of the Flight Mechanics/Estimation Theory Symposium, NASA-Goddard Space Flight Center, Greenbelt, MD, May 1996, pp. 71-83.

[97] Markley, F. L., "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 311–317.

[98] Crassidis, J. L. and Markley, F. L., "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, 2003, pp. 536–542.

[99] Cheng, Y. and Crassidis, J. L., "Particle Filtering for Sequential Spacecraft Attitude Estimation," American Institute of Aeronautics and Astronautics, AIAA Paper 2004–5337, August 2004.

[100] Chen, J., Yuan, J., and Fang, Q., "Flight Vehicle Attitude Determination Using the Modified Rodrigues Parameters," *Chinese Journal of Aeronautics*, Vol. 21, No. 5, 2008, pp. 433–440.

[101] Lee, D. and Alfriend, K. T., "Additive Divided Difference Filtering for Attitude Estimation Using Modified Rodrigues Parameters," American Astronautical Society, AAS Paper 08-283, June 2008.

[102] Aldrich, J., "R. A. Fisher and the Making of Maximum Likelihood, 1912-1922," *Statistical Science*, Vol. 12, No. 3, 1997, pp. 162–176.

[103] Kendall, M. G., "Daniel Bernoulli on Maximum Likelihood," *Biometrika*, Vol. 48, Nos. 1/2, 1961, pp. 1–18.

[104] Hald, A., "On the History of Maximum Likelihood in Relation to Inverse Probability and Least Squares," *Statistical Science*, Vol. 14, No. 2, 1999, pp. 214–222.

[105] Leon–Garcia, A., *Probability and Random Processes for Electrical Engineering*, Addison–Wesley, Reading, MA, 1994, pp. 207–231.

[106] Sorenson, H. W., *Parameter Estimation: Principles and Problems*, Marcel Dekker, Inc., New York, 1980, pp. 183–198.

[107] Huber, P. J., "The Behavior of Maximum–Likelihood Estimates Under Nonstandard Conditions," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, LeCam, L. M. and Neyman, J. (Eds.), University of California Press, Berkeley, CA, Vol. 1, 1967, pp. 221–233.

[108] Hampel, F. R., "A General Qualitative Definition of Robustness," *Annals of Mathematical Statistics*, Vol. 42, No. 6, 1971, pp. 1887–1896.

[109] Hampel, F. R., "The Influence Curve and Its Role in Robust Estimation," *Journal of the American Statistical Association*, Vol. 69, No. 346, 1974, pp. 383–393.

[110] Holland, P. W. and Welsch, R. E., "Robust Regression Using Iteratively Reweighted Least Squares," *Communications in Statistics: Theory and Methods*, Vol. 6, No. 9, 1977, pp. 813–827.

[111] Beaton, A. E. and Tukey, J. W., "The Fitting of Power Series, Meaning Polynomials, Illustrated on Band–Spectroscopic Data," *Technometrics*, Vol. 16, No. 2, 1974, pp. 147–185.

[112] Bickel, P. J., "One-Step Huber Estimates in the Linear Model," *Journal of the American Statistical Association*, Vol. 70, No. 350, 1975, pp. 428-434.

[113] Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., and Tukey, J. W.,

Robust Estimates of Location: Survey and Advances, Princeton University Press, Princeton, NJ, 1972.

[114] Mahalanobis, P.C., "On The Generalized Distance in Statistics," *Proceedings of the National Institute of India*, Vol. 12, 1938, pp. 49–55.

[115] Gasko, M. and Donoho, D. L., "Influential Observations in Data Analysis," *Proceedings of the Business and Economic Statistics Section*, American Statistical Association, 1982, pp. 104-110.

[116] Rousseeuw, P. J. and van Zomeren, B. C., "Unmasking Multivariate Outliers and Leverage Points," *Journal of the American Statistical Association*, Vol. 85, No. 411, 1990, pp. 633–639.

[117] Rousseeuw, P. J. and van Zomeren, B. C., "Robust Distances: Simulations and Cutoff Values," In Stahel, W. and Weisberg, S. (Eds.), *Directions in Robust Statistics and Diagnostics, Part II*, Vol. 34, The IMA Volumes in Mathematics and its Applications, Springer-Verlag, New York, 1991, pp. 195–203.

[118] Des Rosiers, A. P., Schoenig, G. N., and Mili, L., "Robust Space–Time Adaptive Processing Using Projection Statistics," International Conference on Radar Systems, Toulouse, France, October 2004.

[119] Mili, L., Cheniae, M. G., Vichare, N. S., and Rousseeuw, P. J., "Robust State Estimation Based on Projection Statistics," *IEEE Transactions on Power Systems*, Vol. 11, No. 2, 1996, pp. 1118–1125.

[120] Mallows, C. L., "On Some Topics in Robustness," Technical Memorandum, Bell Telephone Laboratories, Murray Hill, NJ, 1975.

[121] Mallows, C. L., "Some Theory of Nonlinear Smoothers," *Annals of Statistics*, Vol. 8, No. 4, 1980, pp. 695–715.

[122] Krasker, W. S. and Welsch, R. E., "Efficient Bounded–Influence Regression Estimation," *Journal of the American Statistical Association*, Vol. 77, No. 379, 1982, pp. 595–604.

[123] Petrus, P., "Robust Huber Adaptive Filter," *IEEE Transactions on Signal Processing*, Vol. 47, No. 4, 1999, pp. 1129–1133.

[124] Handschin, E., Schweppe, F. C., Kohlas, J., and Fiechter, A., "Bad Data Analysis for Power System State Estimation," *IEEE Transactions on Power Apparatus and Systems*, Vol. 94, No. 2, 1975, pp. 329–337.

[125] Coakley, C. W. and Hettmansperger, T. P., "A Bounded Influence, High Breakdown, Efficient Regression Estimator," *Journal of the American Statistical Association*, Vol. 88, No. 423, 1993, pp. 872–880.

[126] Ghandi, M. A., Ledoux, C., and Mili, L., "Robust Estimation Methods for Impulsive Noise Suppression in Speech," IEEE International Symposium on Signal Processing and Information Technology, 2005, pp. 755–760.

[127] Thomas, L. and Mili, L., "A Robust GM–Estimator for the Automated Detection of External Defects on Barked Hardwood Logs and Stems," *IEEE Transactions on Signal Processing*, Vol. 55, No. 7, 2007, pp. 3568–3576.

[128] Duncan, D. B. and Horn, S. D., "Linear Dynamic Recursive Estimation from the Viewpoint of Regression Analysis," *Journal of the American Statistical Association*, Vol. 67, No. 340, 1972, pp. 815–821.

[129] Burl, J. B., *Linear Optimal Control*, Addison–Wesley, Reading, MA, 1999, pp. 437.

[130] Van Loan, C. F., "Computing Integrals Involving the Matrix Exponential," *IEEE Transacctions on Automatic Control*, Vol. 23, No. 3, 1978, pp. 395–404.

[131] Athans, M., Wishner, R. P. and Bertolini, A., "Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements," *IEEE Transactions on Automatic Control,* Vol. 13, No. 5, 1968, pp. 504–514.

[132] Wu, W., "Maximum Likelihood Identification of Glint Noise," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 32, No. 1, 1996, pp. 41–51.

[133] Schlee, F. H., Standish, C. J., and Toda, N. F., "Divergence in the Kalman Filter," *AIAA Journal*, Vol. 5, No. 6, 1967, pp. 1114–1120.

[134] Karlgaard, C. D. and Schaub, H., "Huber–Based Divided Difference Filtering," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 3, 2007, pp. 885–891.

[135] Farrenkopf, R. L., "Analytic Steady–State Accuracy Solutions for Two Common Spacecraft Attitude Estimators," *Journal of Guidance and Control*, Vol. 1, No. 4, 1978, pp. 282–284.

[136] Kuhl, C. T. F., "Combined Earth–/Star Sensor for Attitude and Orbit Determination of Geostationary Satellites," Ph.D. Dissertation, Institute fur Flugmechanik und Flugregelung, Universitat Stuttgart, March 2005.

[137] Oshman, Y. and Carmi, A., "Attitude Estimation from Vector Observations Using Genetic-Algorithm Embedded Quaternion Particle Filter," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 4, 2006, pp. 879–891.

[138] Lam, Q., Woodruff, C., Ashton, S., and Martin, D., "Noise Estimation for Star Tracker Calibration and Enhanced Precision Attitude Determination," *Proceedings of the Fifth International Conference on Information Fusion*, Vol. 1, 2002, pp. 235–242.

[139] Eggleston, J. M. and Dunning, R. S., "Analytical Evaluation of a Method of Midcourse Guidance for Rendezvous with Earth Satellites," NASA TN D–883, June 1961.

[140] Kane, T. R., "Solution of Kinematical Differential Equations for a Rigid Body," *Journal of Applied Mechanics*, Vol. 40, No. 1, 1973, pp. 109–113.

[141] Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[142] Mohanty, N. C., "Autonomous Navigation for High Altitude Satellites," *Information Sciences*, Vol. 30, No. 2, 1983, pp. 125–150.

[143] Mokuno, M., Kawano, I., and Suzuki, T., "In–Orbit Demonstration of Rendezvous Laser Radar for Unmanned Autonomous Rendezvous Docking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, No. 2, 2004, pp. 617–626.

[144] Mili, L., Phaniraj, V., and Rousseeuw, P. J., "Robust Estimation Theory For Bad Data Diagnostics in Electric Power Systems," *Control and Dynamic Systems*, Vol. 37, 1990, pp. 271–325.

[145] Weiner, T. F., "Theoretical Analysis of Gimballess Inertial Reference Equipment Using Delta-Modulated Instruments," Sc.D. Thesis, Massachusetts Institute of Technology, March 1962.

[146] Marandi, S. R. and Modi, V. J., "A Preferred Coordinate System and the Associated Orientation Representation in Attitude Dynamics," *Acta Astronautica*, Vol. 15, No. 11, 1987, pp. 833–843.

[147] Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517.

[148] Hurtado, J. E., "Interior Parameters, Exterior Parameters, and a Cayley–like Transform," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, 2009, pp. 653–657.

[149] Tsiotras, P., Junkins, J. L., and Schaub, H., "Higher Order Cayley Transforms with Appli-

cations to Attitude Representations," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 528–536.

[150] Bruccoleri, C. and Mortari, D., "MRAD: Modified Rodrigues Vector Attitude Determination," *Journal of the Astronautical Sciences*, Vol. 54, No. 3–4, 2006, pp. 383–390.

[151] Setoodeh, P., Khayatian, A., and Farjah, E., "Attitude Estimation By Divided Difference Filter–Based Sensor Fusion," *Journal of Navigation*, Vol. 60, No. 1, 2007, pp. 119–128.

[152] Idan, M., "Estimation of Rodrigues Parameters from Vector Observations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 32, No. 2, 1996, pp. 578–586.

[153] Savage, P. G., "Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 19–28.

[154] Savage, P. G., "Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 208–221.

[155] Tsiotras, P., "Stabilization and Optimality Results for the Attitude Control Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 772–779.

[156] Gunnam, K. K., Hughes, D. C., Junkins, J. L., and Kehtarnavaz, N., "A Vision–Based DSP Embedded Navigation Sensor," *IEEE Sensors Journal*, Vol. 2, No. 5, 2002, pp. 428–442.

[157] Creamer, N. G., Gilbreath, G. C., Meehan, T. J., Vilcheck, M. J., Vasquez, J. A., Rabinovich, W. S., Goetz, P. G., and Mahon, R., "Interspacecraft Optical Communication and Navigation Using Modulating Retroreflectors," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 1, 2004, pp. 100–106.

[158] Hintze, G. C., Cornett, K. G., Rahmatiour, M. H., Heaton, A. F., Newman, L. E., Fleischmann, K. D., and Hamby, B. J., "AVGS, AR&D for Satellites, ISS, the Moon, Mars and Beyond," American Institute of Aeronautics and Astronautics, AIAA Paper 2007-2883, May 2007.

[159] Monda, M. J. and Schaub, H., "Spacecraft Relative Motion Estimation using Visual Sensing Techniques," American Institute of Aeronautics and Astronautics, AIAA Paper 2005-7116, September 2005.

[160] Stadter, P. A., Duven, D. J., Kantsiper, B. L., Sharer, P. J., Finnegan, E. J., and Weaver, G. L., "A Weak-signal GPS Architecture for Lunar Navigation and Communication Systems," IEEE Aerospace Conference, March 2008.

[161] Hur-Diaz, S., Bamford, B., and Gaylor, D., "Autonomous Lunar Orbit Navigation Using Optical Sensors," American Astronautical Society, AAS Paper 07-312, August 2007.

[162] Singla, P., Subbarao, K., and Junkins, J. L., "Adaptive Output Feedback Control for Spacecraft Rendezvous and Docking Under Measurement Uncertainty," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 4, 2006, pp. 892–902.

[163] Subbarao, K. and Welsh, S., "Nonlinear Control of Motion Synchronization for Satellite Proximity Operations," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1284–1294.

[164] Shen, H. and Karlgaard, C. D., "An Optimal Rendezvous and Docking Simulator for Elliptial Orbits," American Astronautical Society, AAS Paper 07-210, January 2007.

[165] Pearson, D. J., "The Glideslope Approach," American Astronautical Society, AAS Paper 89-162, 1989.

[166] Wang, P. K. C., "Non-Linear Guidance Laws for Autonomous Orbital Rendezvous," *Inter-*

national Journal of Control, Vol. 42, No. 3, 1985, pp. 651–670.

[167] Khalil, H. K., *Nonlinear Systems, Third Edition*, Prentice–Hall, Upper Saddle River, NJ, 2002.

[168] Kluever, C. A., "Feedback Control for Spacecraft Rendezvous and Docking," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 4, 1999, pp. 609–611.

[169] Yoon, H. and Agrawal, B. N., "Novel Expressions of Equations of Relative Motion and Control in Keplerian Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, 2009, pp. 664–669.

[170] Brody, A. R., "Evaluation of the '0.1% Rule' for Docking Maneuvers," *Journal of Spacecraft and Rockets*, Vol. 27, No. 1, 1990, pp. 7–8.

[171] Simandl, M. and Dunik, J., "Derivative–Free Estimation Methods: New Results and Performance Analysis," *Automatica*, Vol. 45, No. 7, 2009, pp. 1749–1757.