

Distributed Estimation Architectures and Algorithms for Formation Flying Spacecraft

by

Milan Mandic

S.B. Aeronautics and Astronautics, Massachusetts Institute of Technology,
2003

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

©2006 Massachusetts Institute of Technology. All rights reserved.

Author

Department of Aeronautics and Astronautics

May 26, 2006

Certified by

Jonathan P. How

Associate Professor of Aeronautics and Astronautics

Thesis Supervisor

Accepted by

Jaime Peraire

Professor of Aeronautics and Astronautics

Chair, Committee on Graduate Students

Distributed Estimation Architectures and Algorithms for Formation Flying Spacecraft

by

Milan Mandic

Submitted to the Department of Aeronautics and Astronautics
on May 26, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Future formation flying missions are being planned for fleets of spacecraft in MEO, GEO, and beyond where relative navigation using GPS will either be impossible or insufficient. To perform fleet estimation for these scenarios, local ranging devices on each vehicle are being considered to replace or augment the available GPS measurements. These estimation techniques need to be reliable, scalable, and robust. However, there are many challenges to implementing these estimation tasks. Previous research has shown that centralized architecture is not scalable, because the computational load increases much faster than the size of the fleet. On the other hand, decentralized architecture has exhibited synchronization problems, which may degrade its scalability. Hierarchic architectures were also created to address these problems. This thesis will compare centralized, decentralized, and hierarchic architectures against the metrics of accuracy, computational load, communication load, and synchronization. It will also briefly observe the performance of these architectures when there are communication delays. It will examine the divergence issue with the EKF when this estimator is applied to a system with poor initial knowledge and with non-linear measurements with large differences in measurement noises. It will analyze different decentralized algorithms and identify the Schmidt-Kalman filter as the optimal algorithmic choice for decentralized architectures. It will also examine the measurement bias problem in the SPHERES project and provide an explanation for why proposed methods of solving the bias problem cannot succeed. Finally, the SPHERES beacon position calibration technique will be proposed as an effective way to make the SPHERES system more flexible to a change of testing environment.

Thesis Supervisor: Jonathan P. How

Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

First of all, I have to thank my family: Cindy, Mom, Dad, Vuk, Pinar and Mina. It is their constant support and love that made this thesis possible. During my masters at MIT, I lived through some of the most difficult experiences, and without their unwavering support, I am not sure this would have been possible.

I especially want to thank my advisor, Professor Jonathan How, for his guidance and support during my graduate research at MIT. Thanks to him I learned a lot not only in my research area, but also how to address the problems and how to seek for an answer. Also, I would like to thank him for offering me assistance and for believing in me during my most challenging days. I will always remember that.

I also want to thank amazing people from my lab: Louis Breger, Arthur Richards, Henry de Plinval, Yoshiaki Kuwata, Mehdi Alighanbari, Luca Bertucelli, Ian Garcia, Ellis King, Megan Mitchell, Georges Aoude, Philip Ferguson, Justin Teo, Han-Lim Choi, Byunghoon Kim, Thomas Chabot, Pal Forus. I also want to thank people from SPHERES lab, Edmund Kong, Simon Nolet, Alvar Saenz-Otero, Serge Tournier and many others. They are not only excellent researchers, but also great people. I also want to thank to professor How's administrative assistants, Kathryn Fischer and Margaret Yoon for all their help during my time in the lab.

This research was funded under NASA Space Communications Project Grant #NAG3-2839 and under SPHERES project. I thank Professor How, Professor Miller and NASA for their continued funding through this research project.

Milan Mandic

Contents

1	Introduction	17
1.1	Thesis Outline	19
2	Mitigating the Divergence Problems of the Extended Kalman Filter	21
2.1	Introduction	21
2.2	Problem Statement	22
2.2.1	Problem Walkthrough	23
2.2.2	Divergence due to Three Divergence Factors	24
2.3	EKF, GSF and Bump-up R Algorithms	25
2.3.1	Gaussian Second Order Filter	27
2.3.2	Bump-up R method	33
2.4	Simulation	35
2.4.1	Plinval's Example revisited	36
2.4.2	The Simulated System	37

2.4.3	The Effect of Varying the HPH^T Bump-up Term	43
2.4.4	The Two-Step Approach	45
2.5	Conclusion	45
3	Analysis of Decentralized Estimation Filters for Formation Flying Space-	
	craft	49
3.1	Introduction	49
3.2	Reduced-order Decentralized Filters	51
3.3	Covariance Comparison	54
3.3.1	Effects Of Using Corrupted Measurements	56
3.3.2	Comparing P_{abu} and P_{bu} With P_{af} and P_f	59
3.3.3	Measure of improvement	62
3.3.4	Simulation	63
3.4	Application of SKF to Hierarchic Architectures	64
3.5	Conclusion	67
4	Improved Comparison of Navigation Architectures for Formation Flying	
	Spacecraft	69
4.1	Introduction	69
4.2	Evaluation Metrics	71
4.3	Improvements to the Simulator	72

4.3.1	Improvement of Scalability	73
4.3.2	Incorporation of the Communication Simulator	74
4.3.3	Performance comparison of the decentralized algorithms with and without communication delay	75
4.3.4	Definition of Hierarchic Architectures	78
4.4	Effect of Communication Delays	79
4.4.1	Centralized Architectures	80
4.4.2	Decentralized Architectures	82
4.4.3	Hierarchic Architectures	83
4.5	Modified Simulation Results and Architecture Comparison	85
4.5.1	Simulation Setup	85
4.5.2	Algorithm Comparison	86
4.5.3	Large scale architecture comparison	89
4.6	Robustness of Estimation Architectures	91
4.6.1	Assumptions	91
4.6.2	Robustness	92
4.6.3	Centralized Architecture	94
4.6.4	Decentralized Architecture	95
4.6.5	Robustness Analysis for Hierarchic Architectures	96
4.6.6	Simulation	101

4.7	Conclusion	102
5	Analysis of SPHERES Bias Problem and Calibration of SPHERES Positioning System	105
5.1	Introduction	105
5.1.1	SPHERES testbed	106
5.1.2	The SPHERES Metrology System	108
5.2	The Techniques of Solving the Bias Problem	109
5.2.1	Overbounding	110
5.2.2	The Bias Estimation	110
5.2.3	The Bias Elimination	111
5.2.4	The Schmidt-Kalman Filter	112
5.2.5	Comparison of Bias Estimation and Elimination	113
5.3	The Sources of the Bias in SPHERES	115
5.3.1	Receiver Biases	116
5.4	Simplified SPHERES Measurement System Setup	118
5.4.1	Measurement Equations	119
5.5	Resolving the Bias Problem in SPHERES	121
5.6	Calibration of the SPHERES Positioning System	123
5.6.1	Calibration Approach and Requirements	124

5.6.2	The Estimation Process	126
5.7	The Calibration Simulation Results	127
5.8	Calibration Conclusion	129
6	Conclusion	131

List of Figures

1-1	A fleet of spacecraft in deep space with relative ranging and communication capabilities	18
1-2	Ground exploration vehicles at the Aerospace Controls Lab, MIT	19
2-1	The Update Step as it should occur (left), and as it actually occurs (right). The curved line is the level line of the range measurement: on this line, the range is constant.	24
2-2	The divergence of the EKF when all three divergence factors are present. On the left, range measurement is much more accurate than the bearing measurement. On the right, the bearing measurement is much more accurate than the range measurement	25
2-3	The range ρ is large, making the B term small. The performance of the EKF and GSF does not differ much, even though the measurements still have a large difference in their noises.	32
2-4	Performance of EKF, GSF and Bump-up R methods with $\sigma_0 = 100$. The GSF and Bump-up R methods perform much better than the original Extended Kalman Filter.	37

2-5	The divergence of the EKF when all three divergence factors are present (bearing measurement error is much larger than range measurement error). On the left, the graph shows the estimation error for three different filters. On the right, the condition number for P is plotted.	39
2-6	The divergence of the EKF when all three divergence factors are present (range measurement error is much larger than bearing measurement error). On the top, the graph shows the estimation error for three different filters. On the bottom, the condition number for P is plotted.	41
2-7	The convergence of the EKF when measurement noises are of the same order of magnitude (not all divergence factors are present). On the top, the graph shows the estimation error for three different filters. On the bottom, the condition number for P is plotted.	42
2-8	On the top: the effect of varying the HPH^T bump-up term when $\sigma_0 = 20$. The HPH^T term is multiplied with constant $\alpha = 1, 2 \dots 7$. The best accuracy is achieved with $\alpha = 4$. On the bottom: The effect of varying the HPH^T bump-up term. In this case, $\sigma_0 = 10$. The HPH^T term is multiplied with constant $\alpha = 1, 2 \dots 7$. The best accuracy is achieved with $\alpha = 6$	44
2-9	The effect of multiplying HPH^T term with a positive constant α	46
2-10	The performance comparison of the regular, continuous Bump-up R method and the Two-Step method with the EKF method starting at time step 15. The EKF alone, without the Bump-up R "help", diverges	47
3-1	ICEKF and Bump-Up R ; at the 20th iteration, new measurements are introduced	55

3-2	Error Covariance Comparison. Differences between the various covariances are all non-negative, which means that $P_{af}^+ > P_{bu}^+ > P_f^+$	57
3-3	$\frac{P_{af}^+ - P_{abu}^+}{P_{af}^+}$ as a function of $\frac{R_{bu}}{R_a}$; When $\frac{R_{bu}}{R_a} = 1$, then $P_{best}^+ = P_{abu}^+$ and the difference is largest. On the other hand when $R_{bu} \approx R_f$ then $R_{bu} \ll R_a$ and the $P_{af}^+ \approx P_{abu}^+$. Graph shows a certain difference, which is due the approximation used for calculating P_{af}^+	63
3-4	Error Covariance Comparison. This figure actually shows what this section has proven: $P_{af} > P_{abu} > P_{best} > P_{bu} > P_f$. The bump-up terms in <i>Bump Up R/SK</i> filters are bringing the P_{af} and P_f to new values P_{abu} and P_{bu} which are closer to the best possible covariance value P_{best}	65
4-1	Profile of the simulation with 16 vehicles. It shows the “take-estim” function as one of the most computationally demanding functions (dark color means the computations are actually performed inside the function as opposed to inside the sub-functions)	74
4-2	Performance comparison of the decentralized algorithms with and without communication delay	76
4-3	Hierarchic Clustering. Super Cluster and the sub-clusters may run different estimation algorithms.	79
4-4	The comparison of various architectures: centralized, decentralized, HCC and HCD. The range of vehicles used in this comparison is from 4 to 25 spacecraft	87
4-5	The comparison of the two hierarchic estimation architectures. The range of vehicles used in this comparison is from 16 to 50 spacecraft	90
4-6	Probability distribution of communication delay	93

4-7	The performance comparison of various filters when communication delays are introduced	102
5-1	SPHERES satellite	106
5-2	SPHERES 2D test at the MSFC	107
5-3	Schematic of SPHERES lab (2D) space, with four beacons mounted on the walls	108
5-4	Receiver triggering schematics. The threshold is raised to a certain level in order to filter the noise.	116
5-5	Error graph, from Serge Tournier (1 beacon, 2 receivers, same beacon-receiver configuration)	117
5-6	Standard deviation graph, from Serge Tournier (1 beacon, 2 receivers, same beacon-receiver configuration)	118
5-7	3D setup (with z component constant) with 2 beacons and 4 receivers on a single SPHERES face	119
5-8	Evolution of error in beacon positions	128
5-9	Performance of the proposed calibration approach. The “x” are true positions of the beacons, and “o” are the estimated beacon positions	129

Chapter 1

Introduction

Future formation flying missions are being planned for fleets of spacecrafts in MEO, GEO and beyond, where relative navigation using GPS would either be impossible or insufficient. To perform fleet estimation for these scenarios, local ranging devices on each vehicle are being considered to replace or augment the available GPS measurements. Besides being identified as an enabling technology for many types of space science missions [9], the concept of formation flying of satellite clusters has also been identified as one of the enabling technologies for the NASA exploration initiative [10]. Examples include ground-exploration in remote destinations where vehicles have to work together to be more efficient than a single vehicle [1-2].

The use of many smaller vehicles instead of one monolithic vehicle can have several benefits:

- improving the science return through longer baseline observation.
- enable faster ground track repeats.
- provide a high degree of redundancy and reconfigurability in the event of a single vehicle failure.

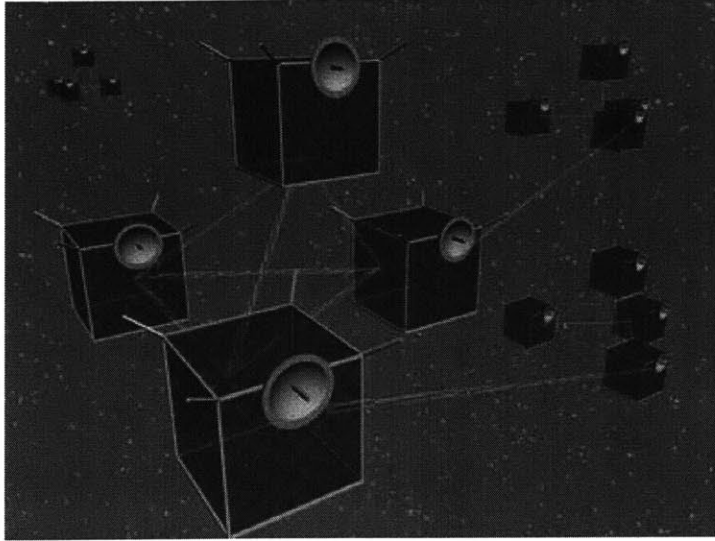


Figure 1-1: A fleet of spacecraft in deep space with relative ranging and communication capabilities

However, the guidance and navigation system for these large fleets is very complicated and requires a large number of measurements. Performing the estimation process in a centralized way can lead to a large computational load that can make the system unusable. Therefore, there is a need for distributing the computational load using decentralized or hierarchic estimation architectures.

This work will focus on extending the previous work of Plinval [1] and Ferguson [22]. It will also provide the analysis that shows the benefits of using Schmidt-Kalman filter as a choice for decentralized estimators.

In addition to this topic, two more ideas will be explored. The first is the analysis of EKF divergence in the presence of so-called divergence factors such as non-linearity in measurements with distinct accuracies and large initial co-variance.

The second is the SPHERES bias estimation problem and potential solutions. Also, the

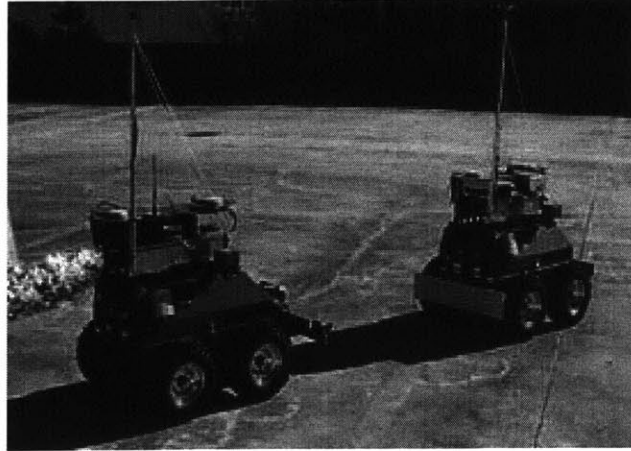


Figure 1-2: Ground exploration vehicles at the Aerospace Controls Lab, MIT

beacon position calibration technique for SPHERES will be introduced and presented as a part of the chapter. This technique will show a simple estimation approach in determining the beacon positions in the SPHERES test environment, which would provide great flexibility for the SPHERES system.

1.1 Thesis Outline

This thesis consists of six chapters. After the initial introduction, the second chapter will focus on the divergence issues with the extended Kalman filter. Previous work will be briefly described, followed by new insights and simulations.

Chapter 3 examines the analysis of decentralized estimation filters for formation flying spacecraft, in which bump-up decentralized algorithms (more specifically, the Schmidt-Kalman filter) are shown to have an advantage over the non-bump-up estimators. The analytical derivation is shown, followed by the simulation results.

Chapter 4 describes the various estimation architectures and compares them against sev-

eral different metrics. This, in essence, is a continuation of the work done by Plinval [1] with an improved simulator. A brief, qualitative discussion of robustness of various estimation architectures is included.

Chapter 5 describes the measurement bias problem in SPHERES and potential approaches to solve it. However, the specific nature of the biases in the SPHERES measurement system creates a much more complex problem, which cannot be solved easily and efficiently using estimation techniques. This chapter also describes the SPHERESs beacon position calibration technique, which allows the SPHERES system to be more flexible when changing test environments.

Finally, the last chapter summarizes the work of the previous chapters.

Chapter 2

Mitigating the Divergence Problems of the Extended Kalman Filter

2.1 Introduction

The Extended Kalman Filter (EKF) is the most common non-linear filter for estimation problems in the aerospace and other industries. The EKF performs very well in solving problems with non-linear measurements and/or non-linear dynamics. However, the filter is not optimal due to the approximation that is performed using the linearization method (i.e. Taylor series expansion). This linearization can also cause undesirable effects on the performance of the filter [3, 4]. We will consider how the EKF can diverge when applied to a system with a large initial state-error covariance that uses non-linear measurements of distinct accuracies [1].

The EKF divergence issues are well known and documented [3, 4, 5, 6]. Several studies have already described the causes of divergence in the EKF for relative navigation prob-

lems [8, 22, 25]. The recent work of Huxel and Bishop [8] addresses the divergence issue of EKF in the presence of large initial state-error covariance, inertial range (large ρ) and relative range (small ρ) measurements. They concluded that EKF divergence is caused by ignoring the large second order linearization terms in B that correspond to relative (short) range measurements. The B term is included in the Gaussian Second Order Filter (GSF), which allows the GSF to converge. Because the measurements used in their analysis were of equal accuracy, they did not explore the effect on the system when some measurements had different accuracies than others. Plinval [1] described the effect of using sensors with different accuracies on the performance of the EKF. He geometrically explained the reason behind the divergence of EKF when the non-linearity in the sensors is coupled with their different accuracies. This chapter will extend that analysis to show that the EKF can have divergence issues when three different divergence factors are combined (see Problem Statement). We will also provide a more general discussion of the problem presented by Plinval and introduce two methods (GSF and Bump-up R) to address the problem.

2.2 Problem Statement

As discussed in Ref. [1], the problem of EKF divergence due to non-linear measurements of substantially different accuracies can be explained geometrically. The problem lies in the process of linearizing the non-linear measurements, in which some information is lost. This means that in every subsequent step, the measurement matrix H in the EKF measurement update equation is incorrect. This error can become even more significant during the estimation process if the measurements being linearized have different accuracies. However, for this effect to actually become significant, the filter needs to rely mostly on the measurements. This will be the case if there is a large initial state-error covariance P_0 . Therefore, the divergence of the EKF can be caused when three “divergence factors” are combined:

- Large initial state-error covariance P_0 . Large means much larger than measurement noise covariance R .
- Significant non-linearity in measurements.
- Large difference in the measurement errors.

2.2.1 Problem Walkthrough

Let us consider a system where a state vector of size 2 is estimated using non-linear sensors of significantly different accuracies and with poor initial knowledge of the states (i.e. large P_0). This is similar to the system used in Ref. [1]. After the first measurements are collected, the state error covariance matrix P can be updated as

$$P^+ = (I - KH)P^- \quad (2.1)$$

Since P_0 is large, the filter will give preference to the measurements rather than the previous knowledge. After the first update, the state error covariance matrix will have a very small eigenvalue in the direction associated with the more accurate measurement. However, it will still have a large eigenvalue in the direction associated with the less accurate measurement. The new measurements collected after the first update will now be linearized around the newly acquired estimate. This means that the directions of the linearized new measurements will be different from the directions of the previous measurements. During the second update, the more accurate measurement will cause a further decrease in state error covariance in the direction corresponding to that measurement. However, this direction is not the same as in the previous update step. This can lead to a significant decrease in the state error covariance in a wrong direction. This can be observed in Figure 2-1. Furthermore, as the state error covariance matrix, P , decreases in directions other than the ones corresponding

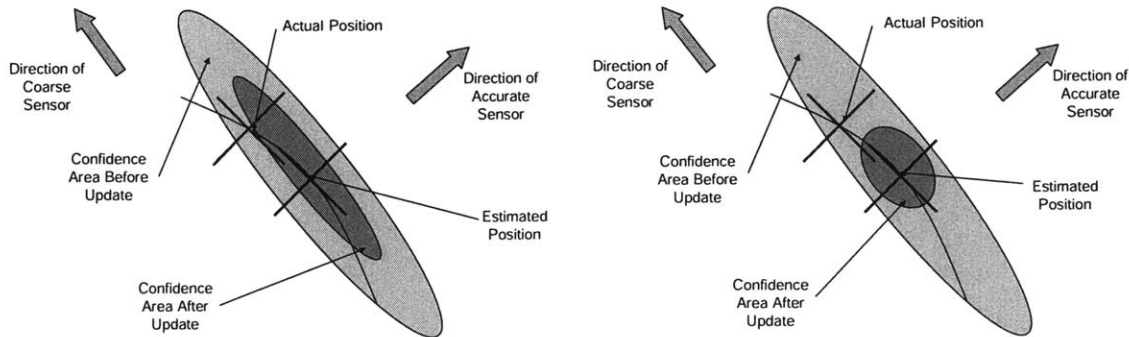


Figure 2-1: The Update Step as it should occur (left), and as it actually occurs (right). The curved line is the level line of the range measurement: on this line, the range is constant.

to the direction of the accurate measurement, the EKF will become decreasingly responsive to measurements in those wrong directions and it will become increasingly confident in its prior state knowledge. If the state error covariance becomes sufficiently small with the state estimates still far away from the true state, the EKF will diverge. This phenomenon is called spill-over of good measurements in the wrong directions.

2.2.2 Divergence due to Three Divergence Factors

In the problem statement, we referred to the three factors that are responsible for the divergence of the EKF. Of course, it is possible to have divergence with only two of those factors, especially when highly non-linear measurements are involved. However, when all three factors are included, the EKF can diverge easily. In order to confirm this, we will show a series of simulations with various factors included in the simulation section. The simulations will show that when one of the factors is accounted for or is minimized, the EKF will converge.

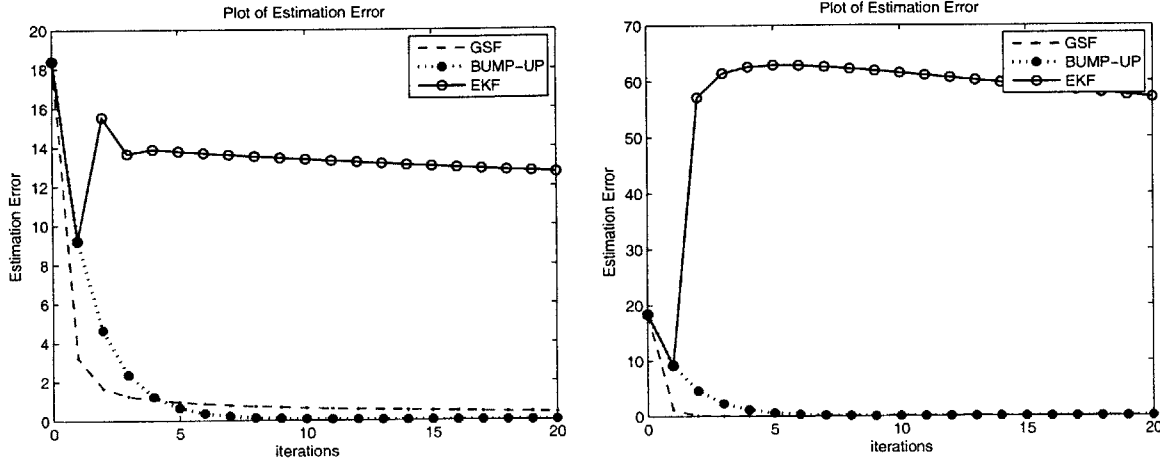


Figure 2-2: The divergence of the EKF when all three divergence factors are present. On the left, range measurement is much more accurate than the bearing measurement. On the right, the bearing measurement is much more accurate than the range measurement

Figure 2-2 shows the performance of the three different filters, EKF, GSF and Bump-up R when all three divergence factors are present (the details of these simulations are given in the Simulation section, 2.4). That is: the EKF filter diverges when non-linear measurements of highly different measurement noises are used for updating the system with large P_0 .

The performance of the GSF and Bump-up R filters is also shown in Figure 2-2. Their performance is superior to the performance of the EKF filter, as they are accounting for one of the three divergence factors. This will be examined in the following sections.

2.3 EKF, GSF and Bump-up R Algorithms

This section will focus on the derivation of the three filters. We will use the system similar to the one described in chapter 6 of [1]. The system is in 2D, and we are trying to estimate the position of the fixed object using two non-linear measurements: range and bearing. In addition, we will use the scalar update approach to show the evolution of the B term

described in the GSF algorithm. System state model is as following:

$$\mathbf{x}_k = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k \quad (2.2)$$

where the states: $x_1 = x$, $x_2 = y$, are coordinates of the fixed object

$$\mathbf{x}_{k+1} = \begin{bmatrix} \phi_1 & 0 \\ 0 & \phi_2 \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_k \quad (2.3)$$

Setting the process noise \mathbf{w} to zero and using the fact that, in order to simplify the analysis, there are no dynamics

$$\Delta \mathbf{x}_k = 0 \Rightarrow x_1^{k+1} = x_1^k \text{ and } x_2^{k+1} = x_2^k \quad (2.4)$$

or equivalently that $\phi_1 = \phi_2 = 1$. The time propagation equations for state error covariance simplified to

$$P_{k+1}^- = P_k^+ \quad (2.5)$$

This allows us to focus on the measurement update equations. The non-linear range and bearing measurements can be written as:

$$h_1 = \rho = \sqrt{\hat{x}_1^2 + \hat{x}_2^2} \quad (2.6)$$

$$h_2 = \theta = \arctan\left(\frac{\hat{x}_2}{\hat{x}_1}\right) \quad (2.7)$$

with corresponding Jacobian:

$$H = \begin{bmatrix} \frac{\hat{x}_1}{\rho} & \frac{\hat{x}_2}{\rho} \\ -\frac{\hat{x}_2}{\rho^2} & \frac{\hat{x}_1}{\rho^2} \end{bmatrix} \quad (2.8)$$

The measurement accuracy is:

$$R = \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \quad (2.9)$$

where σ_ρ^2 and σ_θ^2 are range and bearing measurement noise covariances, respectively. The initial conditions are:

$$x_1 = x_1^0 \quad (2.10)$$

$$x_2 = x_2^0 \quad (2.11)$$

$$P_0 = \begin{bmatrix} \sigma_{p1}^2 & 0 \\ 0 & \sigma_{p2}^2 \end{bmatrix} \quad (2.12)$$

where $\sigma_{p1}^2 = \sigma_{p2}^2 = \sigma_0^2$.

Having all the initial values, we can proceed with analyzing different filters. The analysis of the EKF and its related divergence problem is closely tied to the results obtained for the GSF case, so only the GSF analysis will be presented.

2.3.1 Gaussian Second Order Filter

the measurement update equations for the Extended Kalman Filter (EKF) and the Gaussian Second Order Filter (GSF) [?] are:

$$K = P^- H^T (H P^- H^T + R)^{-1} \quad \text{for EKF} \quad (2.13)$$

$$K = P^- H^T (H P^- H^T + R + B)^{-1} \quad \text{for GSF} \quad (2.14)$$

where B is the covariance term due to the second order terms. B is calculated using Hessians H'_1 and H'_2 , by calculating each member of matrix B separately:

$$B_{jk} = \frac{1}{2} \text{trace}(H'_j P^- H'_k P^-) \quad j, k = 1, 2 \quad (2.15)$$

Hessians H'_1 and H'_2 are defined as second order derivatives of measurement functions h_1 and h_2 . Note that the Hessian of \mathbf{h} is a tensor:

$$H'_1 = \frac{\partial^2 h_1(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}^2} \quad (2.16)$$

$$H'_2 = \frac{\partial^2 h_2(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}^2} \quad (2.17)$$

In this case H'_1 and H'_2 are:

$$H'_1 = \begin{bmatrix} \frac{\hat{x}_2^2}{\rho^3} & -\frac{\hat{x}_1 \hat{x}_2}{\rho^3} \\ -\frac{\hat{x}_1 \hat{x}_2}{\rho^3} & \frac{\hat{x}_1^2}{\rho^3} \end{bmatrix} \quad (2.18)$$

$$H'_2 = \begin{bmatrix} \frac{2\hat{x}_1 \hat{x}_2}{\rho^4} & -\frac{\hat{x}_1^2 - \hat{x}_2^2}{\rho^4} \\ -\frac{\hat{x}_1^2 - \hat{x}_2^2}{\rho^4} & -\frac{2\hat{x}_1 \hat{x}_2}{\rho^4} \end{bmatrix} \quad (2.19)$$

Then, the initial matrix B can be calculated:

$$B = \begin{bmatrix} \frac{\sigma_0^4}{2\rho^2} & 0 \\ 0 & \frac{\sigma_0^4}{\rho^4} \end{bmatrix} \quad (2.20)$$

Using the initial value of B , P , H , and R , we can proceed and calculate the B term after the first update step. To do this, we will perform the first measurement update step on the GSF, one measurement at the time (scalar update approach [8]). We will then apply the

approximation used by Huxel and Bishop [8], to obtain bounds for B :

$$0 \leq B_{h_{kk}} \leq \frac{1}{2}(\|D_h\|\text{trace}(P^-))^2 \quad (2.21)$$

where $B_{h_{kk}}$ corresponds to B , and D_h corresponds to Hessian in our case. Performing the measurement update with the first (range) measurement:

$$H_1 = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \\ \rho & \rho \end{bmatrix} \quad (2.22)$$

Where H_1 corresponds to the first row of the Jacobian matrix H . Therefore:

$$H_1 P_0 H_1^T = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \\ \rho & \rho \end{bmatrix} \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_0^2 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \rho \\ \hat{x}_2 \\ \rho \end{bmatrix} \quad (2.23)$$

$$= \frac{\sigma_0^2(\hat{x}_1^2 + \hat{x}_2^2)}{\rho^2} \quad (2.24)$$

$$= \sigma_0^2 \quad (2.25)$$

and

$$K = \begin{bmatrix} \frac{\sigma_0^2 \hat{x}_1}{\rho} \\ \frac{\sigma_0^2 \hat{x}_2}{\rho} \end{bmatrix} \left(\sigma_0^2 + \sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2} \right)^{-1} \quad (2.26)$$

where $\frac{\sigma_0^4}{2\rho^2} = B_{11}$, as calculated earlier in equation 2.20. In order to compute the bounds on B we need to calculate the $\text{trace}(P)$. Therefore, the elements on the main diagonal of the state error covariance matrix p_{11}^+ and p_{22}^+ can be calculated as:

$$P^+ = (I - KH)P^- \quad (2.27)$$

$$p_{11}^+ = \sigma_0^2 \left(1 - \frac{\sigma_0^2 \hat{x}_1^2}{\rho^2 \left(\sigma_0^2 + \sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2} \right)} \right) \quad (2.28)$$

$$p_{22}^+ = \sigma_0^2 \left(1 - \frac{\sigma_0^2 \hat{x}_2^2}{\rho^2 \left(\sigma_0^2 + \sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2} \right)} \right) \quad (2.29)$$

Simplifying the expressions for p_{11}^+ and p_{22}^+ :

$$p_{11}^+ = \sigma_0^2 \left(1 - \frac{\sigma_0^2 \hat{x}_1^2}{\rho^2 \left(\sigma_0^2 + \sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2} \right)} \right) \quad (2.30)$$

$$\geq \sigma_0^2 \left(1 - \frac{\sigma_0^2}{\sigma_0^2 + \sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2}} \right) \quad (2.31)$$

$$= \frac{\sigma_0^2 \left(\sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2} \right)}{\sigma_0^2 + \sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2}} \quad (2.32)$$

and the similar result (using \hat{x}_2 instead of \hat{x}_1) can be obtained for p_{22}^+ after the update with the first measurement. A detailed analysis of equation 2.32 reveals several key properties for various important cases. For example, depending on the relative values of σ_0 , σ_ρ and ρ , the expression 2.32 will show what influences p_{11}^+ and p_{22}^+ the most. The range ρ plays an important role in determining the significance of non-linearities in measurements. Therefore, we will observe two different cases with respect to the size of ρ :

- $\rho \gg \sigma_0 \gg \sigma_\rho$
- $\rho \approx \sigma_0 \gg \sigma_\rho$

Case 1: $\rho \gg \sigma_0 \gg \sigma_\rho$

In the first case, $\rho \gg \sigma_0 \gg \sigma_\rho$, the Eq.2.32 collapses to the following expression:

$$p_{11}^+ \approx \frac{\sigma_0^2 \sigma_\rho^2}{\sigma_0^2 + \sigma_\rho^2} \approx \sigma_\rho^2 \quad (2.33)$$

A similar expression can be obtained for p_{22}^+ . This expression shows that for a large range, ρ , the p_{11}^+ term is very small. This is as expected, since when the state error covariance is relatively large compared to the measurement error covariance, the Kalman filter will mostly rely on the new information coming from the measurements, rather than on the previous information. After using the same approach for the second measurement, and applying the equations 2.5 and 2.21, similar results were reached, which were confirmed in the simulations ($\|B\|$ is on the order of measurement noise). The analytical derivation of the measurement update with the second measurement is avoided here since it is long and does not add much weight to the discussion.

It can be concluded that the compensation term due to the non-linearities in the measurements, B will be very small, and as stated in 2.21 the non-linear effect of measurements will be small. The initial B shown in the equation 2.20 is also very small due to large ρ . This means is that the large ρ diminishes the non-linear effect of the non-linear measurements. Therefore, the large ρ can account for one of the divergence factors and EKF can converge, as shown in Figure 2-3. (Large number (50) of initial conditions were tested and Figure 2-3 is a representative sample.)

Case 2: $\rho \approx \sigma_0 \gg \sigma_\rho$

The previous case has shown that the compensation term B may not have a large effect on the performance of the EKF filter when ρ is large. However, in this case, ρ is of the same order of magnitude as σ_0 . The equation 2.32 can not be approximated as in the previous case and it remains:

$$p_{11}^+ = \frac{\sigma_0^2 \left(\sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2} \right)}{\sigma_0^2 + \sigma_\rho^2 + \frac{\sigma_0^4}{2\rho^2}} \quad (2.34)$$

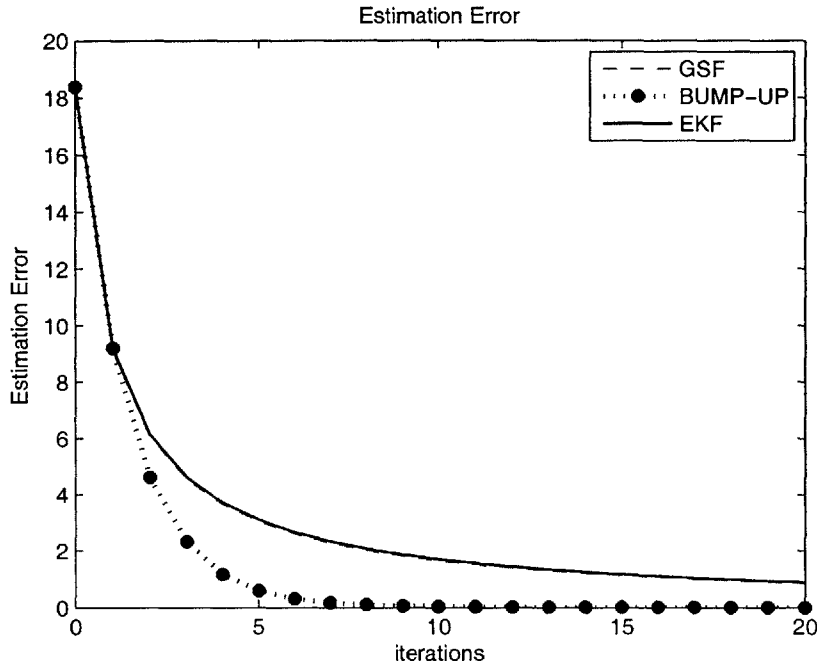


Figure 2-3: The range ρ is large, making the B term small. The performance of the EKF and GSF does not differ much, even though the measurements still have a large difference in their noises.

Therefore, according to Eq.2.21, the linearization compensation term B will become significantly larger than in Case 1. Of course, this is expected since the non-linearity becomes significant for small ranges, and the second order terms compensated by B can not be ignored anymore without risking the divergence of the filter.

GSF Analysis Conclusion

The two cases clearly show an important trait: when P is relatively large compared to R , and the range ρ is of the same order of magnitude as P , then the B term becomes important in order to prevent the EKF from diverging, by avoiding the “spillover” effect discussed earlier. If ρ is much larger than the other values, the EKF can converge even without the use of B .

The reason is that the non-linear effects of the measurements are not significant, as shown earlier in the discussion of Case 1.

Also, when P is small enough $\sigma_\rho \gg \sigma_0$, equation 2.32 collapses to:

$$p_{11}^+ = \sigma_0^2 \quad (2.35)$$

Again, there is no need for the compensation term B , by using the same reasoning as in Case 1.

Since P is usually large initially, it can be concluded that bringing P down slowly by some estimation method other than EKF and then allowing the EKF to take over, as shown in Section 2.4.4, can be beneficial. Of course, there are other ways of solving the divergence issue with EKF, namely the Bump-up R method. However, the true benefit of the GSF and the analysis shown above is to explain how the measurement errors coupled with the non-linearities and large state error covariances can have a negative effect on the filter performance.

2.3.2 Bump-up R method

This method has been shown to be an efficient way of fixing the problem of EKF divergence. The main idea behind the bump-up R method is to set the measurement error covariance R to a value somewhat larger than the original measurement noise in order to compensate for the non-linearities and differences in the measurement noises among various measurements [7]. As proposed in the Ref. ([1]), the bumping-up term R_{bump} that will successfully resolve the divergence problem is:

$$R_{bump} = HP^-H^T \quad (2.36)$$

$$R_{new} = R + R_{bump} \quad (2.37)$$

Plinval further explained how this specific form of R_{bump} actually helps. As shown earlier, the reason why EKF does not converge is explained by the reduction of the state error covariance in a wrong direction after the measurement update. By keeping the measurement error covariance large, this fast reduction is not permitted, but rather the reduction occurs much slower. Ultimately, the goal is to reach a small enough state error covariance P , after which, as shown by Case 2 of the GSF analysis, the filter will not rely on the measurements to the same extent that it does initially (when the P was large).

According to Plinval [1], this form of R_{bump} is selected because it allows the accuracy of the sensors to follow the evolution of the state error covariance P , thus preventing P from becoming ill-conditioned. He shows that this approach slows down the convergence of the filter, but also prevents the divergence, by preventing the over-reduction of P in the direction of the coarse sensor. In fact, the Bump-up R method increases the measurement error covariance, R , so that the filter does not rely solely on the measurements (which, due to their non-linearity, are less accurate than expected) but also on the previous knowledge. Essentially, the Bump-up R method eliminates one of the three divergence factors: $\sigma_0 \gg \sigma_\rho$.

This approach has also been used by Huxel and Bishop [8], with the same R_{bump} term but without the analytic explanation offered, for solving navigation problems involving large state error covariances P and ranging measurements of different orders of magnitude. It was shown that these types of problems can also cause divergence of the EKF, and that the above mentioned R_{bump} term can avoid it. The simulation results will further confirm the benefits of this approach.

Bump-up R Analysis Conclusion

It is interesting to note that the compensation term B in the GSF analysis also behaves as the bump-up term. The main difference is that B is actually computed as the compensation for the second order terms in the Taylor series expansion, that were initially ignored in the EKF approach. The R_{bump} term is an artificial way of slowing the reduction of P , which was analytically shown in [1]. It makes the measurement noise larger, in order to prevent the filter from focusing too much on measurements. Since both terms yield similar results, in the simulation section we will explore how some other similar bump-up terms perform in attempt to prevent the divergence of EKF. Also, it is important to note that the computational load for calculating B is larger than that for R_{bump} , since B is calculated using Hessian tensors of measurement functions h . This makes the Bump-up R method better for practical purposes. The simulations will also show that the bump-up R method tends to be the more accurate of the two methods.

2.4 Simulation

The simulation section will present the performance of the various methods mentioned in this chapter. First, we will describe the system that will be used for simulation. In addition, the results generated with different methods and different initial conditions will be presented. This will include the performance of EKF, GSF and Bump-up R methods; the effects of varying bump-up terms; and a two-step method. Finally, conclusions will be drawn to state how well the simulated results follow the analytical.

2.4.1 Plinval's Example revisited

In the work published by Huxel and Bishop [8], the divergence of EKF was observed when the system involved the measurements whose order of magnitude was of the same order as the state error covariance. Plinval ([1], Chapter 6) described the divergence of EKF in the presence of non-linear measurements with very different measurement noises. Fig. 2-4 shows the example examined by Plinval, and it clearly presents the advantages of GSF and Bump-up R methods over the EKF. The GSF method converges faster, but eventually the Bump-up R method reaches the same and even better accuracy. In this case, the initial state error covariance (P_0) and measurement error covariance (R) are:

$$P_0 = \begin{bmatrix} 100^2 & 0 \\ 0 & 100^2 \end{bmatrix} \quad (2.38)$$

$$R = \begin{bmatrix} 2.5 \cdot 10^{-5} & 0 \\ 0 & 6.0 \cdot 10^{-3} \end{bmatrix} \quad (2.39)$$

Also the initial position estimate was set at:

$$\mathbf{x}_{init} = \begin{bmatrix} 20 \\ 80 \end{bmatrix} \quad (2.40)$$

Since, in this specific case, the range is of similar order as the state error variance $\rho \approx 141.4$ and $\sigma_0 = 100$, this example clearly shows the divergence effect when all three divergence factors are involved. Also, it shows the convergent behavior of the two methods, the GSF and the Bump-up R . Plinval has also shown that this is true for a large number of initial conditions.

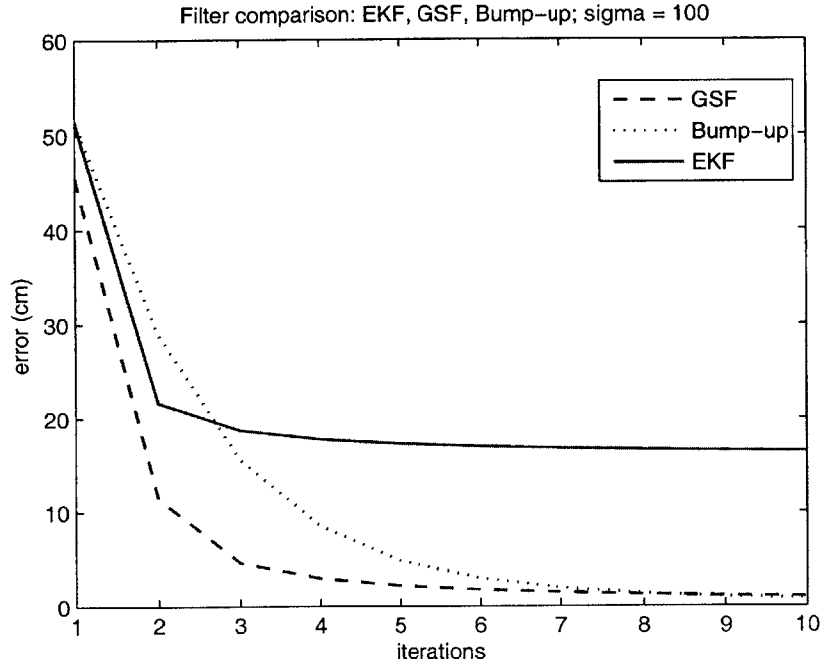


Figure 2-4: Performance of EKF, GSF and Bump-up R methods with $\sigma_0 = 100$. The GSF and Bump-up R methods perform much better than the original Extended Kalman Filter.

2.4.2 The Simulated System

The simulated system is similar to the system described by Plinval ([1], Chapter 6). The equations are already included in the Analysis section (Eqs. 2.2 - 2.12). In order to focus on the effects of the measurements on the performance of the filter, the target has a fixed position and the non-linear measurements involved are range and bearing from the coordinate center. The true position of the target is at:

$$\mathbf{x}_{truth} = \begin{bmatrix} 100 \\ 100 \end{bmatrix} \quad (2.41)$$

The initial position, state error covariance and the relative size of measurement noises will vary. Since the measurements are of different types, the order of magnitude of the measurement noise is not the most representative way of comparing the accuracies of these two measurements. In general the measurement noise in the bearing angles can be much more significant than the ranging noise, especially if the range is of large magnitude.

Performance of EKF, GSF and Bump-up R methods

For Figures 2-5 – 2-7, the plots on the top show the performance of the three filters discussed in this chapter. The plots on the right show the conditional number P , which is identified as one of the symptoms of divergence (or convergence) by Plinval [1]. Different divergence factors are introduced or removed. For example, the plots on the top in Figures 2-5 and 2-6 show the performance of the filters when there is a significant difference in the errors of the two measurements. In both cases, ρ is comparable in size to σ_0 , which is much larger than the measurement errors. This combination of the three divergence factors leads to an ill-conditioned P matrix, and may lead to divergence [1].

The Gaussian Second Order filter accounts for the non-linear effect of the measurements, which effectively removes one of the divergence factors. This translates directly to the convergence of GSF. $\text{Cond}(P)$ is kept low in the initial steps, which was sufficient enough time to allow the filter to converge and avoid significant spill-over.

The Bump-up R method converges somewhat more slowly than the GSF, since it degrades the accuracy of the measurements by a much larger amount than B . This, as described earlier, prevents the filter from focusing too much on measurements, which leads to slower convergence. In the steady-state, the Bump-up R method actually performs better than GSF. Again, the $\text{cond}(P)$ is kept low for long enough to allow the filter to converge.

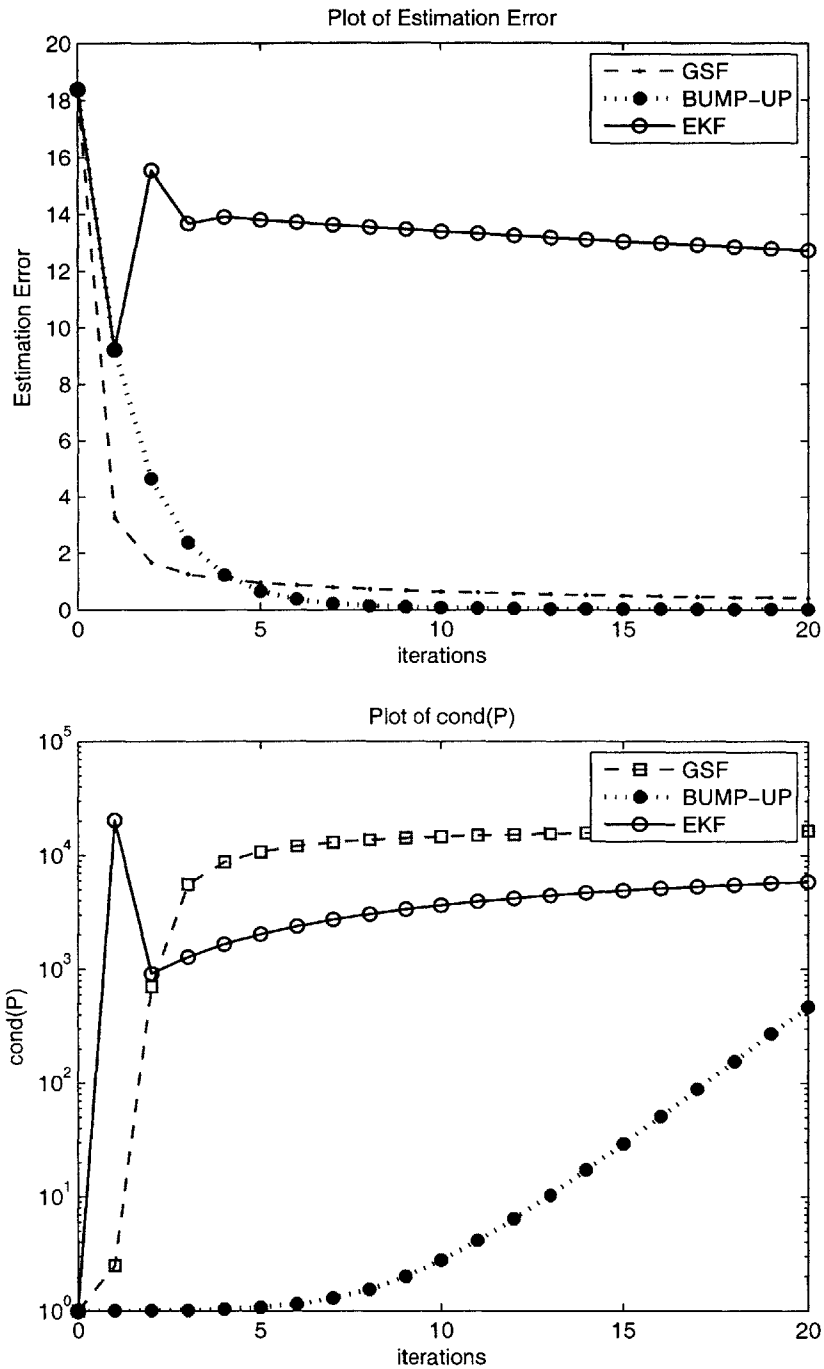


Figure 2-5: The divergence of the EKF when all three divergence factors are present (bearing measurement error is much larger than range measurement error). On the left, the graph shows the estimation error for three different filters. On the right, the condition number for P is plotted.

Table 2.1: Computational load for the EKF, GSF and Bump-up R methods

Method	EKF	GSF	Bump-up R
Computational load (ms)	0.13	0.29	0.13

Figure 2-7 shows the performance of the three filters when the measurement errors are of the same order of magnitude. The EKF in this case converges since not all three divergence factors are present. This effectively confirms our hypothesis that the three factors described in this chapter can easily lead to divergence. Again, the Bump-up R method converges somewhat more slowly than the GSF. The plot of $\text{cond}(P)$ shows that all three filters keep the P well-conditioned, which in this case translates to convergence.

Computational Load

While the first few simulation results show the superiority of the GSF and Bump-up R methods over the Extended Kalman Filter, it is also of great importance to see how practical these methods are. The EKF is widely used, as it can be easily and efficiently implemented on the computers. Table 2.1 shows the computational load of each of the methods while running on the Pentium IV, 2.6GHz processor. The EKF and Bump-up R methods perform better than the GSF method because they do not require the computation of the Hessian tensors of the measurement functions \mathbf{h} , that are required for GSF. The EKF and Bump-up R have about the same computational effort, since the bump-up term in Bump-up R method, HPH^T , is already calculated in the Kalman gain expression.

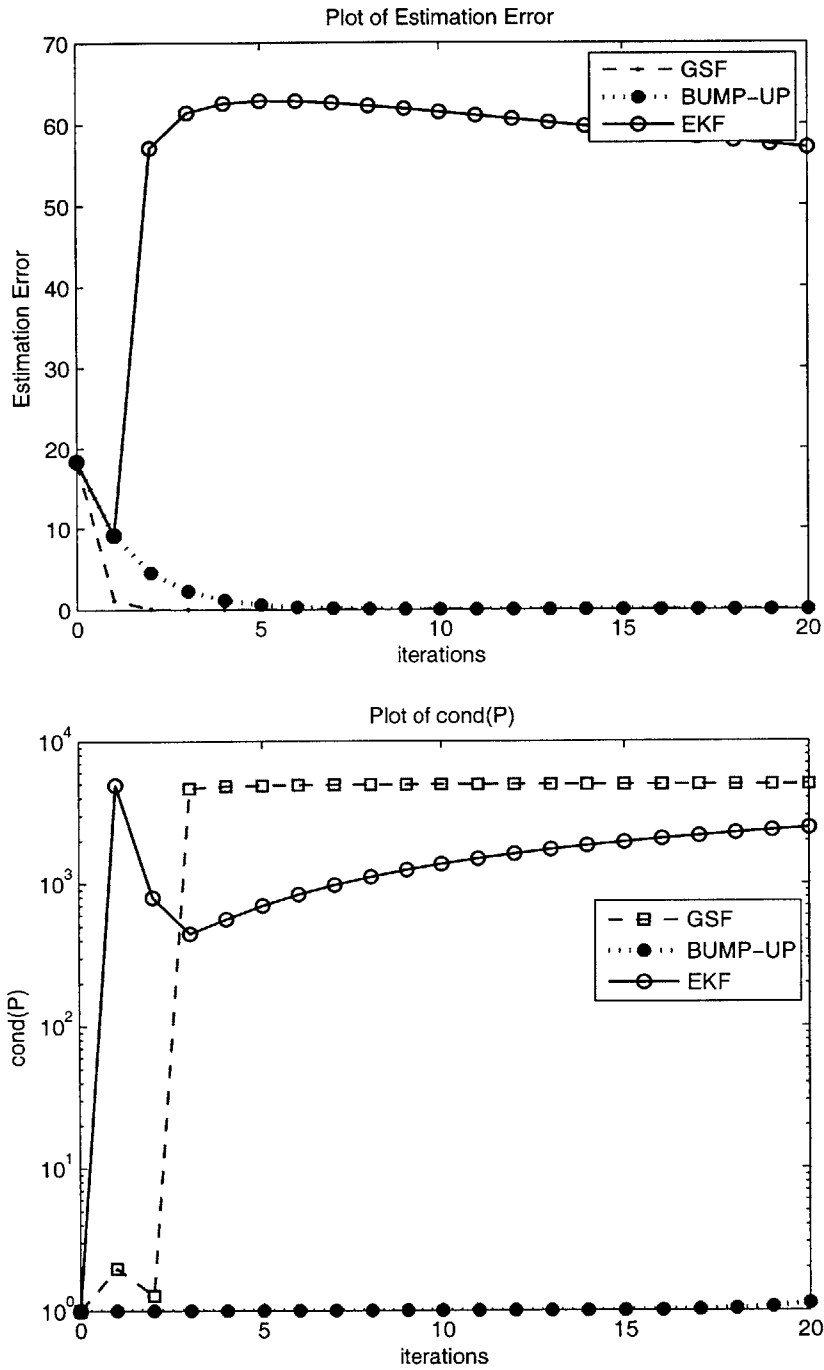


Figure 2-6: The divergence of the EKF when all three divergence factors are present (range measurement error is much larger than bearing measurement error). On the top, the graph shows the estimation error for three different filters. On the bottom, the condition number for P is plotted.

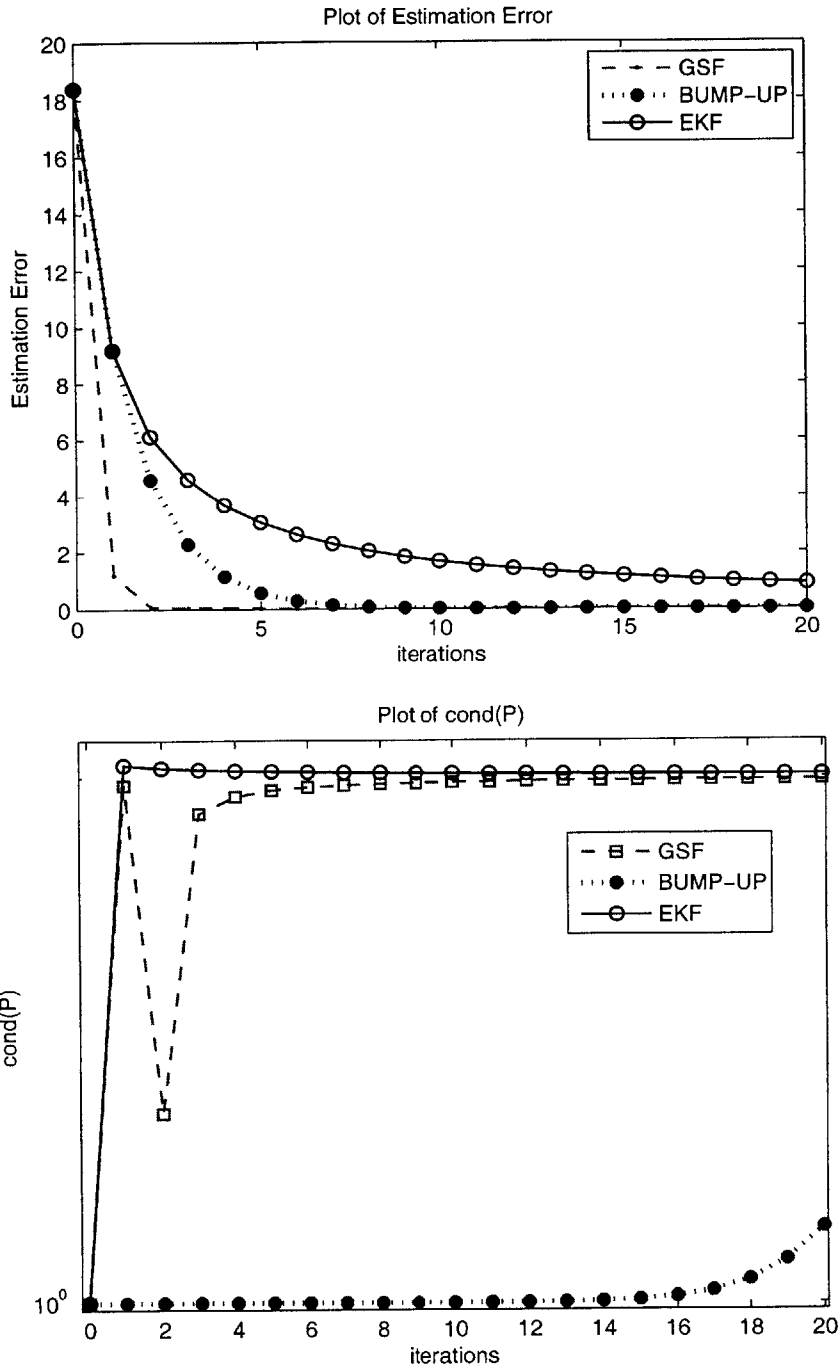


Figure 2-7: The convergence of the EKF when measurement noises are of the same order of magnitude (not all divergence factors are present). On the top, the graph shows the estimation error for three different filters. On the bottom, the condition number for P is plotted.

2.4.3 The Effect of Varying the HPH^T Bump-up Term

As presented in the analysis section, the term B that appears in the derivation of the Gaussian second order filter behaves similarly to the HPH^T term in the Bump-up R method. The previous figures show that these two approaches also prevent the extended Kalman filter from diverging. Therefore, it is of some interest to see how small variations in the HPH^T term can affect the performance of the Bump-up R method.

Figure 2-8 shows the effect of multiplying the HPH^T term in the Bump-up R method with a constant. On the top is the plot when the initial state error standard deviations are set to 20. In the steady-state, it can be seen that the performance of the original Bump-up R method with the HPH^T as a bump-up term performs worse than the methods in which the HPH^T term is multiplied by the constant. In this specific case, the best accuracy is achieved when the multiplier $\alpha = 4$. For $\alpha > 4$ the accuracy degrades. The plot on the bottom in Figure 2-8 shows that the performance of the Bump-up R methods with the varying multipliers (α) actually depends on initial conditions. For the case of $\sigma_0 = 10$, the best performance is achieved with the multiplier $\alpha = 6$. The improvement observed by multiplying the HPH^T term with a constant is very small, but it shows that there is a limit to how far the bumping-up method can go. When α is set to a very large number (i.e. 100 or 1000) the Bump-up R method consistently diverges, similarly to EKF, for various initial conditions.

On the other hand, the plots in Figure 2-9 show the effects on performance of a filter when the HPH^T term is multiplied with a constant $\alpha > 1$ and $0 < \alpha < 1$. Figure 2-9 demonstrates that all three of the bump-up R methods solve the divergence problem of the EKF. The only significant difference among the Bump-up R methods is observed in the bottom plot of Figure 2-9, where the steady states are compared. This figure shows that

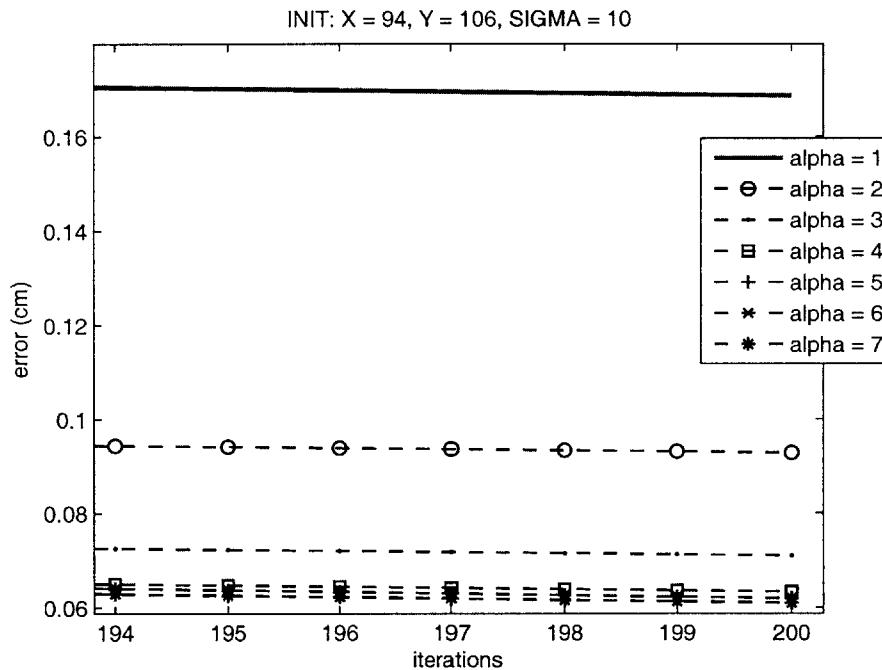
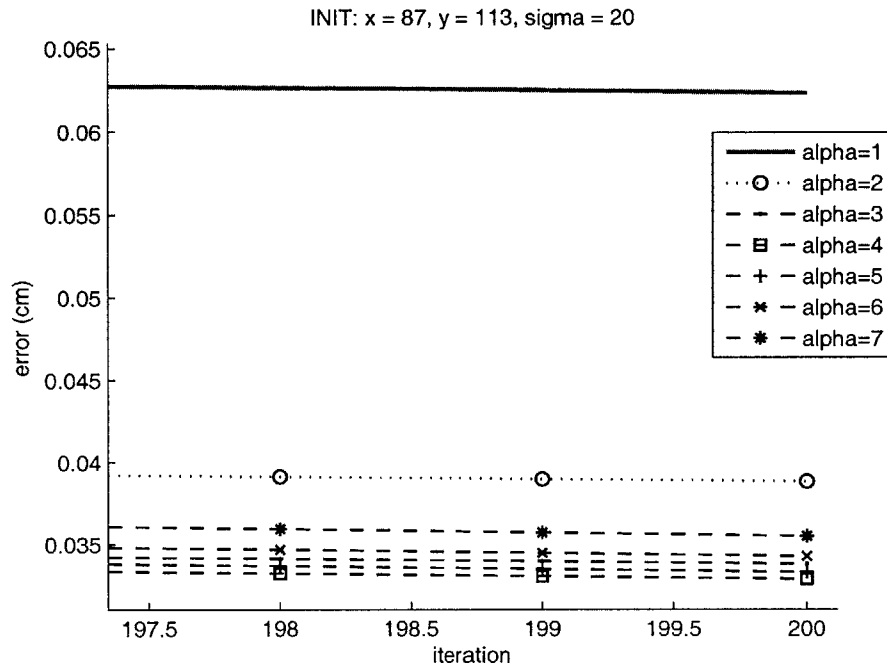


Figure 2-8: On the top: the effect of varying the HPH^T bump-up term when $\sigma_0 = 20$. The HPH^T term is multiplied with constant $\alpha = 1, 2 \dots 7$. The best accuracy is achieved with $\alpha = 4$. On the bottom: The effect of varying the HPH^T bump-up term. In this case, $\sigma_0 = 10$. The HPH^T term is multiplied with constant $\alpha = 1, 2 \dots 7$. The best accuracy is achieved with $\alpha = 6$.

when the HPH^T term is decreased, the steady state performance of the filter degrades.

2.4.4 The Two-Step Approach

The Two-Step method mentioned in the analysis section originates from the observation that the EKF filter will not diverge when initial state-error covariance is low (or in other words, when there is a high confidence in the initial state of the system). A similar observation was made by Huxel and Bishop [8]. In essence, this method effectively removes one of the divergence factors (large P_0) and allows the EKF to converge.

The Two-Step method is not a new concept. It has been explored by Kasdin [2] and discussed in technical comments by Lisano [7]. The basic idea is to bring the initial state error covariance to a sufficiently low level so that the EKF can take over the estimation process, which then will not lead to divergence. Based on the results presented in this chapter, it can be seen that one way of fulfilling the requirement of lowering the state-error covariance is by running the Bump-up R filter during the first few time steps. Following that, one can switch to using the EKF. Figure 2-10 demonstrates this case. This figure compares the performances of the regular Bump-up R method that runs continuously versus the Two-Step method, in which the EKF method starts at time step 15. The EKF converges very well, and its performance is almost as good as the performance of the Bump-up R method.

2.5 Conclusion

This chapter examined the divergence issue of the Extended Kalman Filter (EKF), which occurs in the presence of non-linear measurements with large accuracy differences and large initial state-error covariance. Indeed, the analysis presented here has shown that the con-

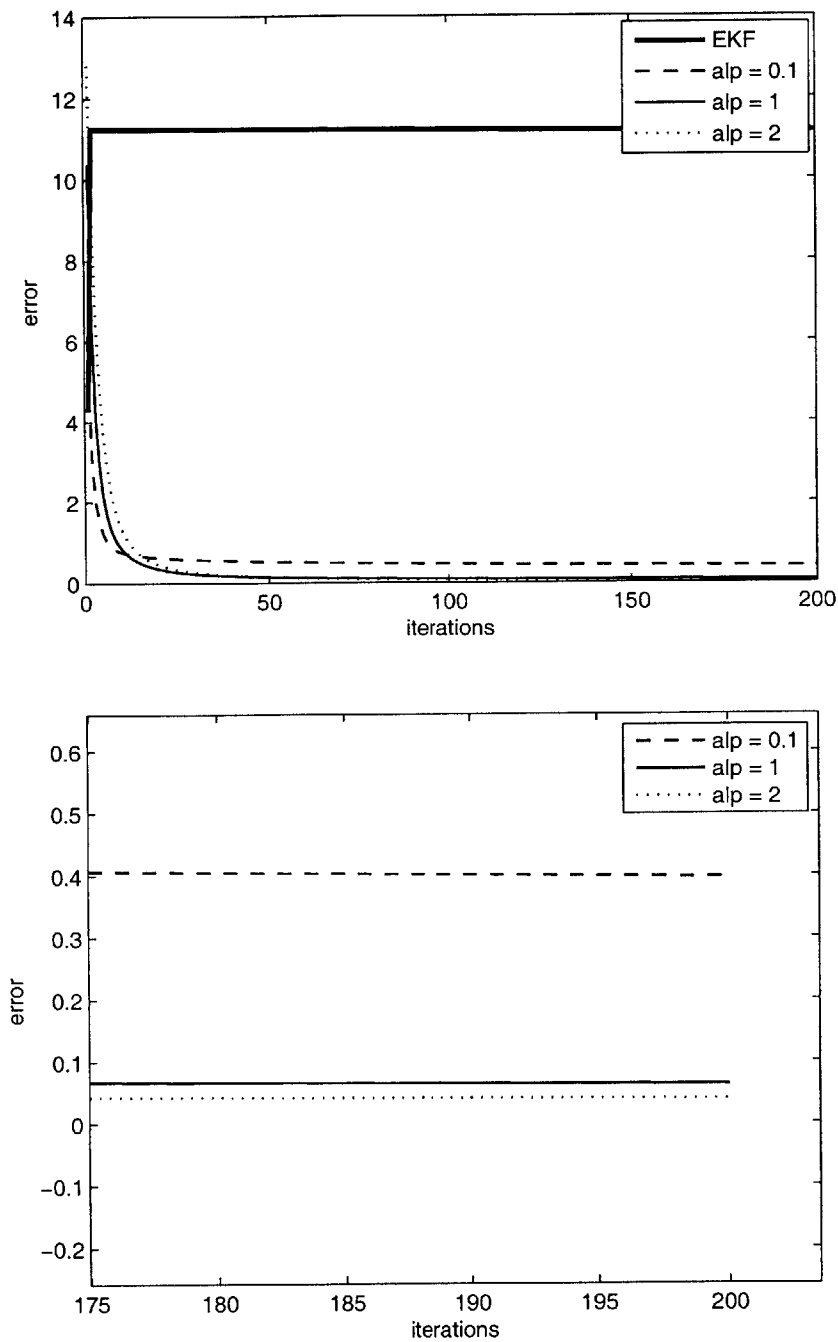


Figure 2-9: The effect of multiplying HPH^T term with a positive constant α

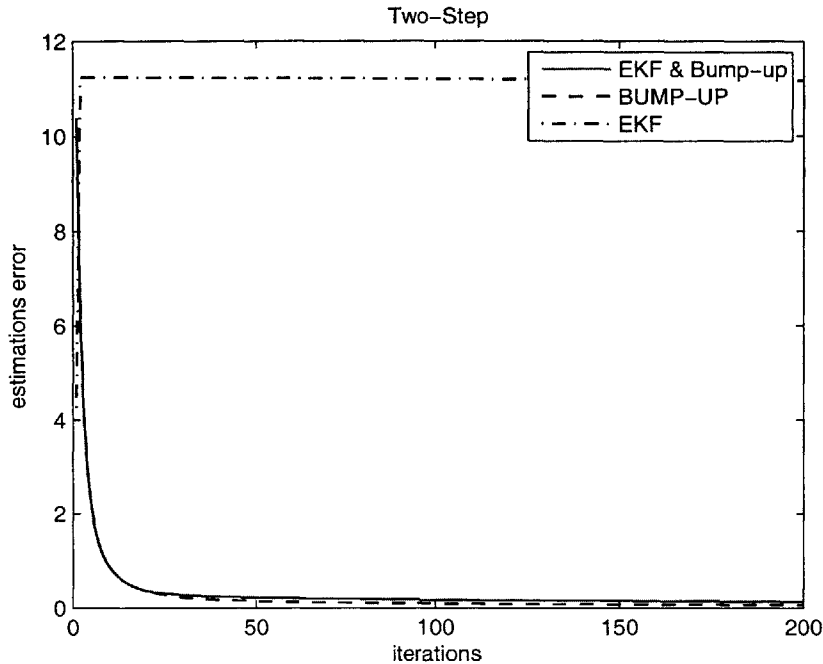


Figure 2-10: The performance comparison of the regular, continuous Bump-up R method and the Two-Step method with the EKF method starting at time step 15. The EKF alone, without the Bump-up R “help”, diverges

vergence of the EKF depends on a combination of factors: the size of the initial state error covariance, P , the relative sizes of the measurements noises and the non-linearity of those measurements. This is also confirmed by Plinval [1], with the use of a geometric argument. The basic idea is that the initial state error covariance needs to be large enough to allow the filter to focus on the measurements. These non-linear measurements are highly dependent on the state estimates, as their linearization is performed in the vicinity of those estimates. Those state estimates can be far off as a result of the spill-over effect described in the Problem Statement section of this chapter and also in [1]. This combination may lead to filter over-reliance on the measurements that are actually less accurate than what their measurement noise covariance shows. As the process iterates based on these corrupt measurements, the EKF can diverge.

A couple of methods have been presented that seem to solve the divergence problem of the EKF. The Gaussian second order filter (GSF) includes the second order terms from the Taylor series expansion (which are ignored by the EKF) and therefore somewhat accounts for one of the factors responsible for the divergence. These second order terms are represented in the filter equations as a compensation or as a bump-up term B . This term is added to the measurement noise covariance, R . The simulations show that this effectively improves the convergence of the filter. However, this method is of less practical value as it is accompanied with a large computational load.

Another way of solving the divergence problem is by developing another bump-up term, HPH^T . Again, this term is added to the measurement noise covariance, R . Similarly to the GSF method, this bump-up term makes measurements less accurate and diverts the filter's attention away from the measurements. This method is more practical, as the computational load is comparable to the one of EKF. Although the convergence is shown to be a bit slower than in the case of the GSF, it is still sufficiently fast. The effect of varying the HPH^T term is also presented in the Simulation section.

Finally, the Two-Step method is presented as the combination of the Bump-up R method and the EKF. The purpose of the Bump-up R method is to bring the error covariance P to a low value, after which the EKF can continue without diverging. The reason is that when P is low, the EKF does not rely on the measurements as much. Essentially, this removes some of the divergence factors mentioned above.

Chapter 3

Analysis of Decentralized Estimation Filters for Formation Flying Spacecraft

3.1 Introduction

The concept of formation flying of satellite clusters has been identified as an enabling technology for many types of space missions [9], [10]¹. In the near future, some formation flying technologies may fit well into the new NASA initiative. An example of this is ground exploration of remote destinations, where having a group of vehicles working together may be far more efficient than a single vehicle. The use of fleets of smaller vehicles instead of one monolithic vehicle should (i) improve the science return through longer baseline observations, (ii) enable faster ground track repeats, and (iii) provide a high degree of redundancy and reconfigurability in the event of a single vehicle failure. The GN&C tasks are very complicated

¹This chapter has been published at the AIAA GNC conference, August 2004 [24]

for larger fleets because of the size of the associated estimation and control problems and the large volumes of measurement data available. As a result, distributing the guidance and control algorithms becomes a necessity in order to balance the computational load across the fleet and to manage the inter-spacecraft communication. This is true not only for the optimal planning, coordination, and control [11], but also for the fleet state estimation, since the raw measurement data is typically collected in a decentralized manner (each vehicle takes its own local measurements).

GPS can be used as an effective sensor for many space applications, but it requires constant visibility of the GPS constellation. In space, GPS visibility begins to breakdown at high orbital altitudes (e.g. highly elliptic, GEO, or at L2). Thus, a measurement augmentation is desired to permit relative navigation through periods of poor visibility and also to improve the accuracy when the GPS constellation is visible [13, 14, 15, 16, 17].

However, the local range measurements taken onboard the spacecraft strongly correlate the states of the vehicles, which destroys the block-diagonal nature of the fleet measurement matrix [12, 18] and greatly complicates the process of decentralizing the algorithms [20]. In contrast to the GPS-only estimation scenario, which effectively decentralizes for reasonable fleet separations, this estimation problem does not decorrelate at any level. As a result, Ref. [20] investigated several methods to efficiently decentralize the estimation algorithms while retaining as much accuracy as possible.

To populate the decentralized architectures, Ref. [20] developed a new approach to estimation based on the **Schmidt Kalman Filter** (SKF). The SKF was shown to work well as a *reduced-order decentralized estimator* because it correctly accounts for the uncertainty present in the local ranging measurements, which is a product of not knowing the location of the other vehicles in the fleet. Since this correction is applied to the covariance of the measurement, this property of the SKF is called the **Schmidt covariance correction**

(SCC).

We extend the covariance comparison in Ref. [20] to consider the transients that occur as ranging measurements are added to the estimator. Ref. [20] performed a similar comparison on the steady-state covariances from the various filter and architecture options. This enabled a comparison of the filter performances, but it could not explain why some of the decentralized techniques performed better than others. The investigation in this paper of what we call the “transient response” of the filter provides further insight on the relative performance of these filters. This analysis also indicates the advantage of using the SCC, which can be extended to other estimation algorithm/architectures.

The following section discusses prior work on reduced-order decentralized filters, which is followed by a detailed investigation of the covariance for different algorithms.

3.2 Reduced-order Decentralized Filters

Recent work by Park [21] introduced the Iterative Cascade Extended Kalman Filter (ICEKF), a reduced-order estimation algorithm for use in decentralized architectures. This filter is used for local ranging augmentation in applications where GPS-only measurements are not sufficient. The ICEKF filter uses an iterative technique that relies on communication between each vehicle in the fleet and continues until a specified level of convergence is reached. It was shown that the ICEKF can incorporate local ranging measurements with GPS levels of accuracy, producing nearly optimal performance. However, Ref. [22] demonstrated that the filter performance can deteriorate when highly accurate local measurements (i.e., more accurate than GPS) are added, and that this performance loss occurs when error/uncertainty in the relative state vectors is not correctly accounted for in the filter.

One way to account for this uncertainty in the relative state is to include it in the measurement noise covariance R , which is the approach taken in the *Bump Up R* method:

$$R_{\text{bump}} = R + JP_{yy}J^T \quad (3.1)$$

where J is the measurement matrix for all non-local measurements in the fleet and P_{yy} is the initial covariance matrix for all non-local states in the fleet state vector. Equation 3.1 implies that the measurements now have larger noise covariance, making them less accurate than was initially assumed.

Another approach examined in Ref. [22] is the *Schmidt Kalman Filter* (SKF). This filter also increases the variances in the R matrix, but in contrast to *Bump Up R*, this approach is dynamic and also accounts for the off-diagonal blocks of the error covariance. The SKF eliminates non-local state information, thereby reducing the computational load on the processor. This elimination is accomplished by partitioning the measurement and propagation equations:

$$\begin{bmatrix} x \\ y \end{bmatrix}_{k+1} = \begin{bmatrix} \phi_x & 0 \\ 0 & \phi_y \end{bmatrix}_k \begin{bmatrix} x \\ y \end{bmatrix}_k + \begin{bmatrix} w_x \\ w_y \end{bmatrix}_k \quad (3.2)$$

$$z_k = \begin{bmatrix} H & J \end{bmatrix}_k \begin{bmatrix} x \\ y \end{bmatrix}_k + \nu_k \quad (3.3)$$

$$P_k = \begin{bmatrix} P_{xx} & P_{yx} \\ P_{xy} & P_{yy} \end{bmatrix}_k \quad (3.4)$$

where x represents the vector containing the states of interest (called the *local state*,

which includes the positions, velocities, and time of the vehicle relative to the fleet origin) and y represents the remaining states (the positions, velocities, and time of all other vehicles relative to the fleet origin). After partitions of Eqs. 5.8 and 5.9 are applied to the general Kalman filter equations, each block is solved, and the gain for the y states is set to zero [23]:

$$\alpha_k = H_k P_{xx_k}^- H_k^T + H_k P_{xy_k}^- J_k^T + J_k P_{yx_k}^- H_k^T + J_k P_{yy_k}^- J_k^T + R_k \quad (3.5)$$

$$K_k = (P_{xx_k}^- H_k^T + P_{xy_k}^- J_k^T) \alpha_k^{-1} \quad (3.6)$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^- - J_k \hat{y}_0) \quad (3.7)$$

$$P_{xx_k}^+ = (I - K_k H_k) P_{xx_k}^- - K_k J_k P_{yx_k}^- \quad (3.8)$$

$$P_{xy_k}^+ = (I - K_k H_k) P_{xy_k}^- - K_k J_k P_{yy_k}^- \quad (3.9)$$

$$P_{yx_k}^+ = P_{xy_{k+1}}^{-T} \quad (3.10)$$

$$P_{yy_k}^+ = P_{yy_k}^- \quad (3.11)$$

Schmidt-Kalman Time Update

$$\hat{x}_{k+1}^- = \phi_{x_k} \hat{x}_k^+ \quad (3.12)$$

$$P_{xx_{k+1}}^- = \phi_{x_k} P_{xx_k}^+ \phi_{x_k}^T + Q_{x_k} \quad (3.13)$$

$$P_{xy_{k+1}}^- = \phi_{x_k} P_{xy_k}^+ \phi_{y_k}^T \quad (3.14)$$

$$P_{yx_{k+1}}^- = P_{xy_{k+1}}^{-T} \quad (3.15)$$

$$P_{yy_{k+1}}^- = \phi_{y_k} P_{yy_k}^+ \phi_{y_k}^T + Q_{y_k} \quad (3.16)$$

In order for the SKF to compute an appropriate amount to increase R , each spacecraft communicates both its local state vector and its local error covariance matrix to the next spacecraft in the fleet. This change to R is called the *Schmidt Covariance Correction*.

The additional error covariance information, which is not transmitted when using *Bump Up R*, allows a more appropriate correction, but also requires additional inter-spacecraft communication.

The analysis performed in Ref. [22] showed that the error covariance in the ICEKF method is relatively close to the error covariance of the optimal, centralized case. However, this observation was described as misleading, because it was not a good indicator of the filter’s performance. The filter’s unrealistically high trust in the measurements, due to the assumed low error covariance, was conjectured to be the primary reason for this poor performance. The following section explores this point in more detail. It also presents an equivalent derivation for the SKF approach, which demonstrates how increasing the measurement noise covariance improves the actual filter performance.

3.3 Covariance Comparison

Previous research in Refs. [19, 20] and [21] showed that results from the ICEKF are worse than might be expected. The ICEKF produces these poor results, because it makes unrealistic assumptions about the uncertainty associated with the ranging measurements from the other vehicles, since this filter does not include the position uncertainty of other vehicles. These unrealistic assumptions are captured in the measurement noise covariance (R), which provides a measure of the “quality” of each measurement. One way to investigate this problem is to analyze the error covariance matrix (P), and to understand the true impact we take the approach of investigating the transients immediately after ranging measurements are added to the estimator. This approach differs from that of Ref. [20], which compares steady-state covariances. The steady state values are a good way to compare the overall performance values, but they tend to obscure the reasons why some filters diverge and oth-

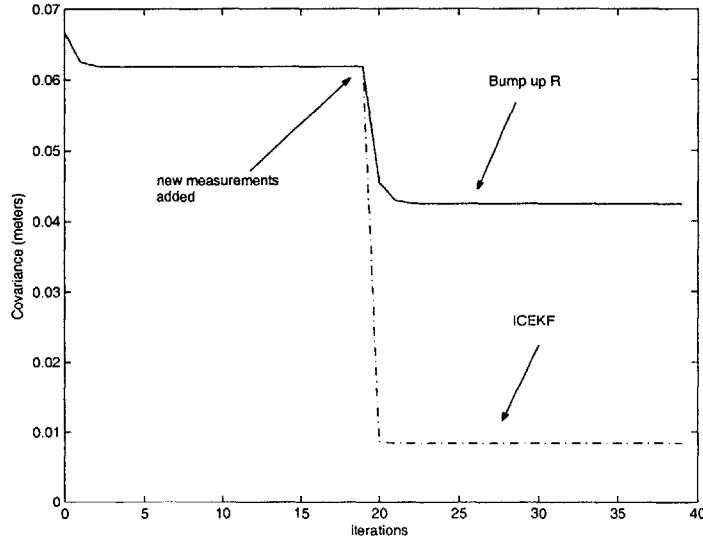


Figure 3-1: ICEKF and Bump-Up R; at the 20th iteration, new measurements are introduced. The ICEKF filter error covariance is much lower than it should be at this stage of the estimation, thereby corrupting all future measurement updates. Using the transient covariance analysis more clearly shows the difference in covariances that occur when new, corrupt measurements are introduced to the filter, as is shown in Figure 3-1. This figure was obtained for a SISO system, for which *Bump Up R* is essentially equivalent to the Schmidt-Kalman Filter (SKF). Analyzing P across the transient step should show how the incorrect modeling of R impacts the filter's confidence in the state estimates.

The ICEKF filter error covariance is much lower than it should be at this stage of the estimation, thereby corrupting all future measurement updates. The analytic derivation of this phenomenon is provided in Section 3.3.1. The results also show that this problem can be partially alleviated by increasing the R value in the algorithm using a systematic approach, such as the *Bump Up R* or SKF formulations. For the scalar case, the error covariances resulting from the two methods are related by:

$$P_{\text{bu}}^+ > P_{\text{f}}^+ \quad (3.17)$$

where the error covariance matrix P_f is analyzed using R_f with the gain based on R_f , and P_{bu} is analyzed using R_{bu} with gain based on R_{bu} . After the update step, the error covariance using the *Bump Up R* approach (P_{bu}^+) is larger than the ICEKF result (P_f^+), so *Bump Up R* should avoid the problems with the ICEKF technique.

Since these analytic results were based on several approximations, a simulation was conducted to confirm that the *Bump Up R* method performed better than ICEKF regardless of the assumptions. This simulation computed the transient behavior of the error covariance using various filters that were based on different assumptions about the measurement noise covariances R . The results are shown in Figures 3-2 and 3-4. These results show that the *Bump Up R* method provides a better prediction of the best possible filter performance, confirming the analytical predictions. Similar results would be expected for the SKF, due to its equivalence with the *Bump Up R* method in a scalar case.

3.3.1 Effects Of Using Corrupted Measurements

The purpose of this section is to analyze the impact of adding new ranging measurements to the estimator, and in particular, to determine how the covariance matrix changes. The bumped-up case is included, because our ultimate goal is to show the effect of using the SCC on incorporating new measurements in a filter. The Schmidt-Kalman filter is a dynamic version of the *Bump Up R* filter, in the sense that it uses a better technique to increase the noise covariance. First, the analysis will be restricted to the scalar case, which should provide adequate insights into the result. Second, the values associated with the actual values are noted with the subscript a , filter values with the subscript f , and bumped-up values with the subscript bu . In the following derivation, $R_a > R_{bu} > R_f$. The derivation begins with a

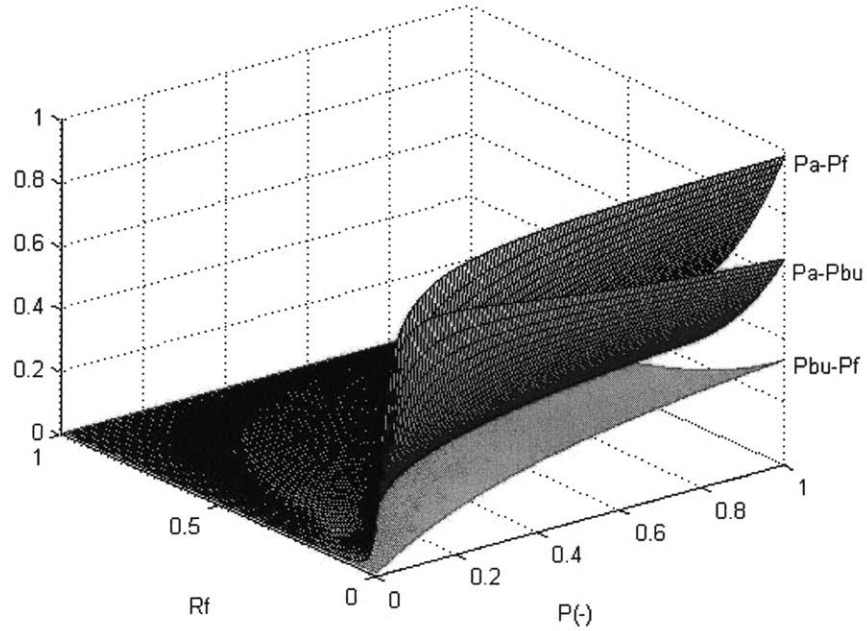


Figure 3-2: Error Covariance Comparison. Differences between the various covariances are all non-negative, which means that $P_{af}^+ > P_{bu}^+ > P_f^+$.

measurement update for the error covariance [23]

$$P^+ = (I - KH)P^- (I - KH)^T + KRK^T \quad (3.18)$$

where K is the Kalman gain, I is an identity matrix, and H is the measurement matrix. The symbols I and H represent scalars in this case. The error covariance P^- represents the filter error covariance based on the measurements available prior to adding the set of ranging measurements. These measurements could be obtained from GPS or other external sources. Since the new, added ranging measurements are more accurate than the previous, the error covariance should substantially decrease. It is essential to observe the transient behavior of the covariance in the first step after the new measurements are added, because the filter performance is heavily impacted at that time. For example, if the measurement noise (R) is

too low, the filter might continue to have a high confidence in erroneous measurements and the estimation results will degrade. In the scalar case, Eq. 3.18 can be rewritten as

$$P^+ = (I - KH)^2 P^- + K^2 R \quad (3.19)$$

The corresponding Kalman gain is

$$K = P^- H^T (H P^- H^T + R)^{-1} = \frac{P^- H}{H^2 P^- + R} \quad (3.20)$$

Thus, for a filter using the incorrect covariance $R = R_f$, the gain matrix would be

$$K = \frac{P^- H}{H^2 P^- + R_f} \quad (3.21)$$

in which case Eq. 3.19 can be rewritten as

$$P_{af}^+ = \left(1 - \frac{P^- H^2}{H^2 P^- + R_f}\right)^2 P^- + \left(\frac{P^- H}{H^2 P^- + R_f}\right)^2 R_a \quad (3.22)$$

where P_{af} is used to designate that this corresponds to the “actual” error covariance that one might expect when using this filter. Essentially the gain is based on the assumed R_f , but the error analysis is based on the more realistic R_a . To proceed, two quantities are defined

$$M \equiv \left(1 - \frac{P^- H^2}{H^2 P^- + R_f}\right)^2 P^- \quad (3.23)$$

$$N \equiv \left(\frac{P^- H}{H^2 P^- + R_f}\right)^2 \quad (3.24)$$

So that Eq. 3.22 becomes

$$P_{af}^+ = M + N R_a \quad (3.25)$$

and similarly

$$P_f^+ = M + NR_f \quad (3.26)$$

so, from Eqs. 3.25 and 3.26

$$P_{af}^+ - P_f^+ = N(R_a - R_f) > 0 \quad \Rightarrow \quad P_{af}^+ > P_f^+ \quad (3.27)$$

3.3.2 Comparing P_{abu} and P_{bu} With P_{af} and P_f

The second step in this derivation is to compare P_{abu} and P_{bu} with P_{af} and P_f . Consider the case where a modified value of the measurement covariance R_{bu} is developed using the *Bump Up R* algorithm. Starting from the equations for the error covariance in Eq. 3.22

$$P_{af}^+ = \left(1 - \frac{P^- H^2}{H^2 P^- + R_f}\right)^2 P^- + \left(\frac{P^- H}{H^2 P^- + R_f}\right)^2 R_a \quad (3.28)$$

$$P_{bu}^+ = \left(1 - \frac{P^- H^2}{H^2 P^- + R_{bu}}\right)^2 P^- + \left(\frac{P^- H}{H^2 P^- + R_{bu}}\right)^2 R_{bu} \quad (3.29)$$

several assumptions and approximations are made to compare these error covariances. For the scalar case, define $\gamma = P^- H^2$, then

$$P_{af}^+ = \left(1 - \frac{\gamma}{\gamma + R_f}\right)^2 P^- + \left(\frac{\gamma}{\gamma + R_f}\right)^2 \frac{R_a}{H^2} \quad (3.30)$$

$$P_{bu}^+ = \left(1 - \frac{\gamma}{\gamma + R_{bu}}\right)^2 P^- + \left(\frac{\gamma}{\gamma + R_{bu}}\right)^2 \frac{R_{bu}}{H^2} \quad (3.31)$$

Now if $R_f \ll \gamma$, which is equivalent to assuming that the new measurements are much more accurate than the previous, then the following is true

$$\frac{y}{y+x} \approx 1 - \frac{x}{y}, \quad x \ll y \quad (3.32)$$

Then Eq. 3.28 reduces as follows

$$P_{af}^+ \approx \left(1 - \left(1 - \frac{R_f}{\gamma}\right)\right)^2 P^- + \left(1 - \frac{R_f}{\gamma}\right)^2 \frac{R_a}{H^2}; \quad \frac{R_f}{\gamma} \ll 1 \quad (3.33)$$

$$= \left(\frac{R_f}{\gamma}\right)^2 P^- + \left(1 - \frac{R_f}{\gamma}\right)^2 \frac{R_a}{H^2} \approx \frac{R_a}{H^2} \quad (3.34)$$

The error covariance for the bumped up method, Eq. 3.29, becomes

$$P_{bu}^+ = \left(\frac{\gamma + R_{bu} - \gamma}{\gamma + R_{bu}}\right)^2 P^- + \left(\frac{\gamma}{\gamma + R_{bu}}\right)^2 \frac{R_{bu}}{H^2} \quad (3.35)$$

$$= \frac{R_{bu}^2 P^-}{(\gamma + R_{bu})^2} + \frac{\gamma^2 R_{bu}}{H^2 (\gamma + R_{bu})^2} \quad (3.36)$$

$$= \frac{R_{bu}^2 \gamma + \gamma^2 R_{bu}}{H^2 (\gamma + R_{bu})^2} = \frac{R_{bu} \gamma (\gamma + R_{bu})}{H^2 (\gamma + R_{bu})^2} = \frac{R_{bu} \gamma}{H^2 (\gamma + R_{bu})} \quad (3.37)$$

$$= \frac{R_{bu} P^-}{\gamma + R_{bu}} \quad (3.38)$$

With these results, a comparison can be drawn between P_{af}^+ and P_{bu}^+ . Assuming

$$P_{af}^+ - P_{bu}^+ > 0 \quad (3.39)$$

which, using Eqs. 3.34 and 3.38, can be rewritten as

$$\frac{R_a}{H^2} > \frac{R_{bu} P^-}{\gamma + R_{bu}} \Rightarrow \frac{R_a \gamma}{H^2} + \frac{R_a R_{bu}}{H^2} > R_{bu} P^- \quad (3.40)$$

$$\Rightarrow P^- \left(\frac{R_a H^2}{H^2} - R_{bu}\right) + \frac{R_a R_{bu}}{H^2} > 0 \quad (3.41)$$

$$\Rightarrow P^- (R_a - R_{bu}) + \frac{R_a R_{bu}}{H^2} > 0 \quad (3.42)$$

which is correct if $R_a > R_{bu}$. Note that if R_f had been used instead of R_a , a similar expression is obtained

$$P^-(R_f - R_{bu}) + \frac{R_f R_{bu}}{H^2} \approx P^-(R_f - R_{bu}) < 0 \quad (3.43)$$

which is correct if $R_f \ll R_{bu} < 1$. These results lead to following set of inequalities

$$P_{af}^+ > P_{bu}^+ > P_f^+ \quad (3.44)$$

It is also important to show that $P_{af}^+ > P_{abu}^+ > P_{bu}^+$

$$P_{af}^+ \approx \frac{R_a}{H^2} \quad (3.45)$$

$$P_{abu}^+ = \left(1 - \frac{\gamma}{\gamma + R_{bu}}\right)^2 P^- + \left(\frac{\gamma}{H(\gamma + R_{bu})}\right)^2 R_a \quad (3.46)$$

Similar to the derivation of P_{bu}^+ :

$$P_{abu}^+ = \frac{R_{bu}^2 \gamma + \gamma^2 R_a}{H^2 (\gamma + R_{bu})^2} \quad (3.47)$$

Therefore,

$$P_{af}^+ - P_{abu}^+ = \frac{R_a}{H^2} - \frac{R_{bu}^2 \gamma + \gamma^2 R_a}{H^2 (\gamma + R_{bu})^2} \quad (3.48)$$

$$= \frac{R_a \gamma^2 + 2R_a R_{bu} \gamma + R_a R_{bu}^2 - R_{bu}^2 \gamma - \gamma^2 R_a}{H^2 (\gamma + R_{bu})^2} \quad (3.49)$$

$$= \frac{\gamma R_{bu} (2R_a - R_{bu}) + R_a R_{bu}^2}{H^2 (\gamma + R_{bu})^2} > 0 \quad (3.50)$$

Using a similar approach, the following result can be reached

$$P_{abu}^+ - P_{bu}^+ = \frac{R_{bu}^2 \gamma + R_a \gamma^2 - R_{bu}^2 \gamma - R_{bu} \gamma^2}{H^2 (\gamma + R_{bu})^2} \quad (3.51)$$

$$= \frac{\gamma^2(R_a - R_{bu})}{H^2(\gamma + R_{bu})^2} > 0 \quad (3.52)$$

In conclusion,

$$P_{af}^+ > P_{abu}^+ > P_{bu}^+ > P_f^+ \quad (3.53)$$

or equivalently

$$P_{af}^+ - P_f^+ > P_{abu}^+ - P_{bu}^+ \quad (3.54)$$

which confirms that after this update step, the error covariance using the *Bump Up R* approach (P_{bu}^+) is larger than the ICEKF result (P_f^+) and is a better indicator of the actual filter covariance. The main factor that makes P_{abu} closer to P_{bu} than P_{af} is to P_f is the amount by which the noise covariance has been magnified. Since the actual noise covariance R_a is not known, the bumped-up noise covariance attempts to estimate the value of actual noise covariance.

3.3.3 Measure of improvement

To measure the improvement obtained by the *Bump Up R* method, Equation 3.52 can be normalized. First assume $\alpha R_{bu} = R_a$ for some $\alpha \geq 1$

$$\frac{P_{af}^+ - P_{abu}^+}{P_{af}^+} = \frac{H^2 \gamma R_{bu} (2R_a - R_{bu}) + R_a R_{bu}^2}{R_a H^2 (\gamma + R_{bu})^2} \quad (3.55)$$

Substitute $R_{bu} = \frac{R_a}{\alpha}$, and Eq. 3.55 becomes

$$\frac{P_{af}^+ - P_{abu}^+}{P_{af}^+} = 1 - \frac{R_a \gamma + \alpha^2 \gamma^2}{(\alpha \gamma + R_a)^2} \quad (3.56)$$

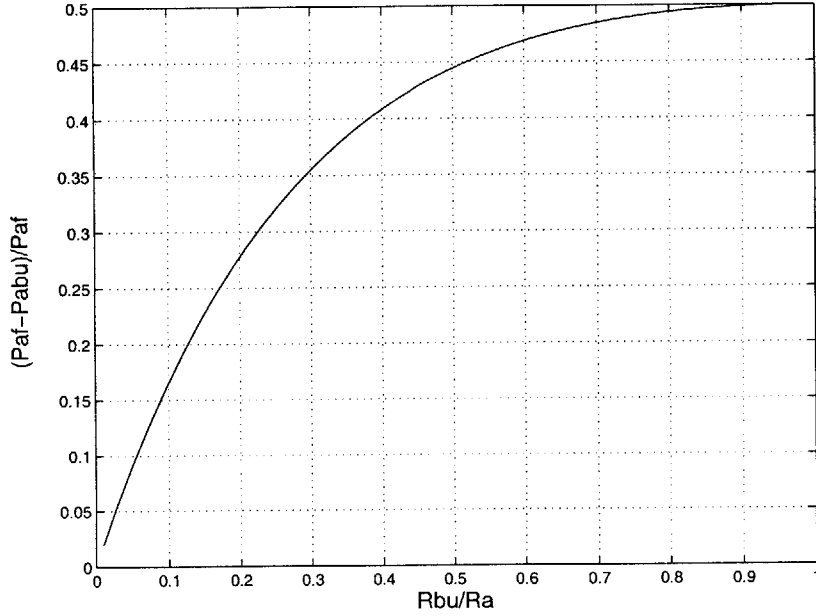


Figure 3-3: $\frac{P_{af}^+ - P_{abu}^+}{P_{af}^+}$ as a function of $\frac{R_{bu}}{R_a}$; When $\frac{R_{bu}}{R_a} = 1$, then $P_{best}^+ = P_{abu}^+$ and the difference is largest. On the other hand when $R_{bu} \approx R_f$ then $R_{bu} \ll R_a$ and the $P_{af}^+ \approx P_{abu}^+$. Graph shows a certain difference, which is due the approximation used for calculating P_{af}^+

Figure 3-3 shows the normalized difference between the actual covariances, P_{abu} and P_{af} , with $\gamma = 1$ and $R_a = 1$. The value of the normalized difference decreases with increased α , which means that as R_{bu} decreases, P_{abu} gets closer to P_{af} , which agrees with the analytic derivation. Then, for $\alpha = 1$, $P_{abu} = P_{best}$ and the difference is the largest (in this specific case the difference is 50%). The strength of the SKF approach is that it calculates the best possible value for bumping up the measurement noise covariance at every step.

3.3.4 Simulation

A simulation was conducted to validate approximations made when obtaining the analytic results and verify that the *Bump Up R* method performs better than ICEKF. The analysis computed the transient behavior of the error covariance using various filters that were based

on different assumptions about the measurement noise covariances R . The measurement covariance matrices used in this study are

- The actual value R_a
- What is assumed in the ICEKF R_f , which is lower than actual R_a
- Increased (*Bump Up R*) value R_{bu} , which takes on values between the actual R_f and the filter R_f

Several different cases were examined:

1. The actual error covariance matrix P_a is computed using R_a , but when the gain is calculated with R_f (R_{bu}) it is called P_{af} (P_{abu}).
2. The error covariance matrix P_f is analyzed using R_f with the gain based on R_f . The same relationship holds for P_{bu} and R_{bu} .
3. P_{best} corresponds to the optimal result with the filter designed and analyzed using R_a .

The results obtained from comparing these cases are shown in Figs. 3-4. The results confirm the analytical comparisons for a wide range of possible R_f and P^- values. They also show that P_{abu} is much closer to P_{best} than P_{af} , indicating that P_{abu} is a much better predictor of the best possible filter performance. A similar result is expected for the SKF.

3.4 Application of SKF to Hierarchic Architectures

Previous work has shown the relative merits of centralized and decentralized navigation architectures [22]. The principal disadvantages of centralized architectures are high computational load and susceptibility to single-point failure. In a centralized architecture, the

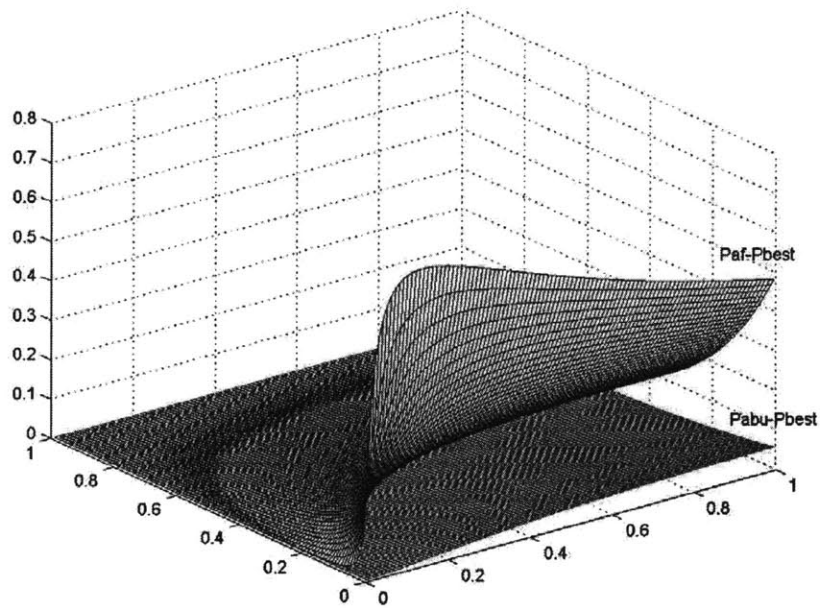
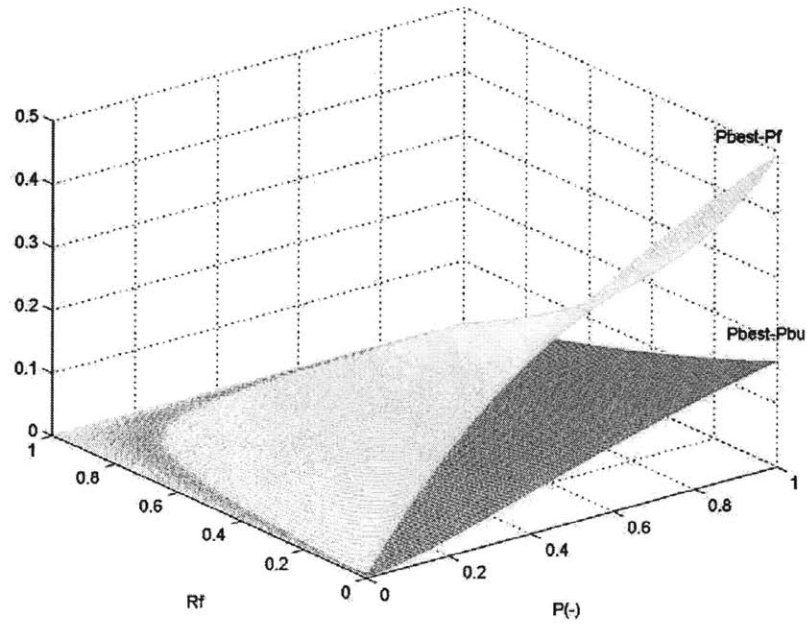


Figure 3-4: Error Covariance Comparison. This figure actually shows what this section has proven: $P_{af} > P_{abu} > P_{best} > P_{bu} > P_f$. The bump-up terms in *Bump Up R/SK* filters are bringing the P_{af} and P_f to new values P_{abu} and P_{bu} which are closer to the best possible covariance value P_{best}

primary computational burden is placed on a single spacecraft, severely limiting the size of a formation. Likewise, routing all communication and computation through a single spacecraft creates the potential for a single point failure for the entire formation, should that spacecraft malfunction. Decentralized architectures can avoid both problems by distributing computational load across the formation, with very little performance loss relative to the centralized architecture [22]. Section 3.3 showed that of the decentralized estimators considered, the Schmidt Kalman Filter provides the best performance. This performance advantage is derived from the way the SKF shares and incorporates new information into the estimate. The approach used by the SKF to share specific information between satellites can be applied to other filters, including those using more centralized architectures. This is achieved using the Schmidt covariance correction (SCC), which allows the various estimation architectures/algorithms to correctly account for errors in the range measurements *without* needing to estimate the states of those vehicles. This can greatly improve the performance and adaptability of the estimation approach, both of which are important properties for reconfigurable networks.

Although using decentralized architectures improves fleet scalability and robustness, it adds complexity to communications and information sharing. This puts limits on the size of the fleet running the decentralized filter. We are currently exploring into the possibilities of developing a new type of hierarchic architecture that will incorporate the use of SCC. This architecture will be different from a traditional hierarchy, because spacecraft will be allowed to communicate with vehicles in their local cluster and with vehicles in other clusters. We will accomplish this without having to increase the estimator size, by using the Schmidt Covariance Correction. The SCC will enable each spacecraft to range off the spacecraft in other clusters (cross-team ranging) without having to estimate their relative states. The SCC essentially provides each spacecraft with a way to receive and correctly implement new

information, regardless of where this information is coming from.

3.5 Conclusion

This chapter investigates various approaches to design highly distributed estimators for formation flying applications. It presents a detailed investigation of the covariances for the different distributed estimation algorithms, showing that the Schmidt Kalman Filter and *Bump Up R* approaches are much better predictors of the best possible filter performance. Finally, we indicate that the main concept behind the Schmidt Kalman Filter (called the *Schmidt covariance correction*) can be used to develop a reduced-order hierarchic estimator that offers distributed computation and can improve the scalability limitations of the centralized and decentralized architectures. The comparison of the different architectures is presented in the following chapter. The comparison is also made for the hierarchic architectures, including hierarchic centralized-centralized (HCC) and hierarchic centralized-decentralized (HCD). The decentralized algorithm of the HCD is running in the sub-clusters and uses the Schmidt covariance correction.

Chapter 4

Improved Comparison of Navigation Architectures for Formation Flying Spacecraft

4.1 Introduction

This chapter focuses on the comparison of different navigation architectures for formation flying spacecraft. It is an extension from the comparison work of Plinval [1]. In his work, Plinval developed a metric system to compare several different architectures. The metric consists of several different parameters in order to properly identify the advantages and disadvantages of various architectures. The metrics are:

- Accuracy
- Computational Complexity
- Communication load
- Level of Synchronization

The following architectures were analyzed:

- Centralized
- Decentralized
- Hierarchic Centralized-Centralized
- Hierarchic Centralized-Decentralized (not considered in [1])

Each of these architectures have issues associated with them. One of the issues with the centralized architectures is the lack of scalability, due to the computational effort required to perform estimation for a large number of state variables. Also, the centralized architecture is often considered to have low robustness in the case of master spacecraft failure [22]. The decentralized architecture also has a problem with scalability due to the high *synchronization* requirement, defined in section 4.2. Splitting the time step into many sub-steps during which each of the spacecraft in the decentralized architecture needs to complete its estimation process puts a high demand on the communication subsystem to deliver important information at specific time intervals. Although the decentralized architecture is considered more robust to single point failures than the centralized architecture, it still has robustness issues with respect to communication delays, which can damage the performance of the estimation process due to the high synchronization requirement. This concern will be addressed in the section on robustness 4.6. However, in order to test our assumptions we needed to develop a communication simulation, that can be used to show the effects of communication delays on the performance of various architectures.

The comparison of these architectures based on the metrics mentioned above, was previously done on a simulator involving up to 24 spacecraft. One of the contributions of this chapter is expanding the scale of the fleet to 50 spacecraft by improving the efficiency of the simulator. This will also allow us to perform a relevant comparison of different hierarchic architecture. A fleet size of 24 is not large enough to allow more detailed comparison of

hierarchical architectures, since the sub-cluster sizes are small (4–5 spacecraft). Increasing the number of spacecraft to 7–8 per cluster can provide us with an answer to how different hierarchical architectures respond to the increased fleet size.

Finally, this chapter will give a brief qualitative description of robustness of various estimation architectures to communication delays.

4.2 Evaluation Metrics

This section briefly describes the evaluation metrics used in evaluating the performance of different navigation architectures, as a more detailed description is available in [1]. The evaluation metrics consists of accuracy, computational complexity, synchronization and communication.

Accuracy Metric The accuracy metric is defined in two possible ways:

- Average accuracy, which is computed as the error over time, after the steady state is reached
- Worst-case accuracy is defined as the average of errors before the measurement updates occur. It is the worst error that can be produced by an algorithm, over time.

Computational Metric The computational complexity metric presents the computational effort exerted by the spacecraft in order to compute the desired estimates. It is computed as the average of the maximum time it takes for each computational loop to finish. In centralized architecture, it is the time it takes for the master of the fleet to compute the estimates for all the spacecraft. In decentralized architecture it is the maximum time

required for a single spacecraft to complete its estimation process.

Synchronization Metric The synchronization metric is defined as the total number of times a spacecraft needs to wait for other spacecrafts in order to be able to complete its task. For example, in the centralized architecture, the master spacecraft has to wait for all the other vehicles to send the relative measurements to it ($N - 1$), and since only the master computes the estimates and communicates them to the other spacecrafts ($N - 1$), the resulting synchronization will be $2(N - 1)$.

Communication Metric Finally, the communication metric is used to capture the amount of information exchange within the fleet. The communication metric is defined as the average amount of information (size of data) exchanged between the spacecraft per iteration.

4.3 Improvements to the Simulator

In his thesis, Plinval explored and compared the performance of different estimation architectures, with the main focus on centralized, decentralized and hierarchic centralized-centralized architectures. He included several measurement metrics which are mentioned earlier and explained in more detail in his thesis. However, while the comparison did show the differences among the different architectures, it still lacked several important factors that could influence the performance of the filters and perhaps change the overall conclusions reached in his thesis. The simulator has been modified in several areas:

- Scalability Improvement
- Communication Simulator
- Addition of Hierarchic Centralized-Decentralized Architecture

This chapter briefly describes the modifications to the simulator and present the performance of the above mentioned architectures under new conditions.

4.3.1 Improvement of Scalability

The previous simulation included fleets with up to 24 spacecraft. While the number is large indeed, when the fleet is split in clusters, the number of spacecrafts in each cluster becomes small, up to 5 spacecraft per cluster. If the cluster has less than 4 spacecraft, the number of relevant measurements may become a problem. We wanted to develop a fleet of larger size that would allow a wider spectrum of comparison among the architectures, where more than 50 spacecraft could operate under a single architecture. This will also allow us to run hierarchic architectures with up to 8 spacecraft per cluster, which in return allows us to make scale comparison of hierarchic architectures, and determine how scalable these architectures are.

The main problem with scalability of this simulator was the fact that the simulator had a high computational load that was increasing dramatically with the increase of the number of spacecraft. While this is expected to cause very slow simulation of the centralized architecture, it appeared that the decentralized architecture was also performing very slowly. After a thorough investigation, we determined that the lag was due to the inefficient part of the code which ran nested loops. The simulator, coded as a Matlab program, was therefore running into large delays as the number of spacecraft increased. Using the Matlab profiler, we identified exactly where the inefficiency occurred Figure 4-1. By vectorizing the code, we were able to decrease the order of complexity from $O(N^2)$ to $O(N)$ for that segment of the code. This made a significant improvement in the efficiency of the simulator and allowed us to run simulations of much larger fleet size. The limit of 60 spacecraft is reached due to the

Profile Summary

Generated 07-Feb-2006 15:18:21 using real time.







Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<u>Spheres3D</u>	1	27.031 s	0.016 s	
<u>HierCentDec3D</u>	1	26.953 s	2.269 s	
<u>take estim</u>	5472	12.433 s	12.433 s	
<u>Measupdate hcd3D</u>	684	5.157 s	5.063 s	
<u>Measurements3D</u>	684	3.512 s	2.525 s	
<u>Time update decent3D</u>	684	2.252 s	2.252 s	
...

Figure 4-1: Profile of the simulation with 16 vehicles. It shows the “take-estim” function as one of the most computationally demanding functions (dark color means the computations are actually performed inside the function as opposed to inside the sub-functions)

limited computer memory.

4.3.2 Incorporation of the Communication Simulator

The original simulation included the measurement of the communication load, but the communication delay that is associated with the communication systems was not included in the simulator. In other words, the communication load was calculated, but had no effect on performance of the filters. The addition of the communication simulator, which uses the TCP/IP protocol for transmitting data, allows us to more properly compare the performances of different architectures. Figure 4-2 shows the effect of communication delays on the accuracy levels of the decentralized algorithm. One area that will be examined specifically is the relative performance of the centralized (low level of communication) and decentralized (high level of communication) architecture.

Table 4.1: Communication delay

amount of data (class 'double')	size (bytes)	average time for transmission (sec)
100	800	0.084
182	1456	0.1520
200	1600	0.1630
500	4000	0.2590
1000	8000	0.3626
2000	16000	0.4497

Also, with the addition of the communication simulator, we have the ability to control the behavior of the communication system. We are able to simulate the communication glitches and their effect on the performance of the filters. Moreover, it allows us to develop the analysis of robustness of the system to various failures of the communication channels. We can inspect whether the decentralized algorithms, which depend heavily on communication, are indeed very robust systems and if they can recover from the serious glitches in communication. Table 4.1 shows how communicating large amounts of data can indeed cause significant delays. This is achieved by using the communication simulator and measuring the time it takes to complete the exchange of information.

4.3.3 Performance comparison of the decentralized algorithms with and without communication delay

Another important addition is the hierarchic centralized-decentralized architecture. This architecture has a special significance, as it combines two different type of architectures at two levels of hierarchy. Increased size of fleet will allow us to make comparison between already existing hierarchic centralized-centralized architecture and newly developed hierarchic centralized-decentralized architecture. Moreover, the addition of the communication

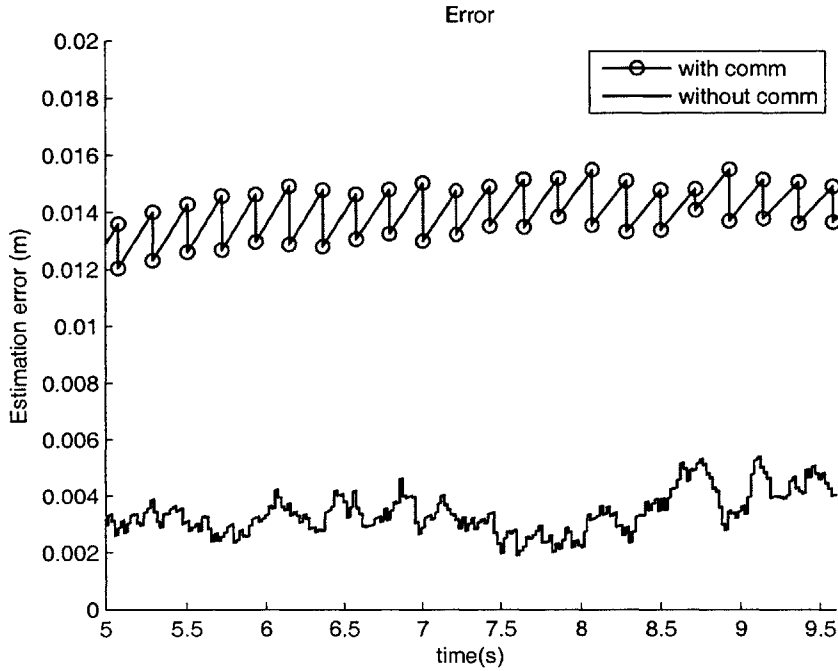


Figure 4-2: Performance comparison of the decentralized algorithms with and without communication delay

simulator will allow us to control the communication, which is a much more significant part of the estimation process in decentralized than in centralized architecture.

Initially only the centralized and decentralized architectures were considered [22]. It was noticed that centralized architecture lacked scalability due to the overwhelming computational load imposed on the central data-processing vehicle. The decentralized architecture was shown to effectively distribute the computational load, at the expense of worse accuracy. Deterioration of accuracy was the result of not having all the information available at every node/spacecraft as in the case of centralized architecture. This lack of information led to the suboptimal decentralized filter compared to the centralized filter, which was considered theoretically to have the most optimal results.

While the lack of optimality and consequent degradation of accuracy (fig. accuracy his-

togram) was a small price for achieving better scalability of the system, the decentralized scheme does have a drawback with respect to scalability. In order to distribute the computational load, some additional information has to be distributed in order to permit successful filter operation at all nodes of the fleet. Indeed, for the Schmidt Kalman Filter used for the decentralized architecture, current estimates and current partitions of the state error covariance have to be transferred to designated nodes in order to have improved performance of the decentralized filter [20, 24]. Therefore, the communication system becomes a very important segment of the overall performance of the estimation architectures.

As the size of the fleet increased [1] it was noticed that the communication requirements of these decentralized architectures were putting a strong pressure on proper and fast performance of the communication system. In this case, the time-step becomes longer due to the delays caused by the communication of required information to members of the fleet. Eventually the step size was forced to be long enough that the nonlinear effects of the measurements and the dynamical model interfered with the proper functioning of the estimation filter. Furthermore, possible failures in the communication system may lead to undesirable effects depending on the severity and size of the failures. As mentioned earlier, this is one of the main reasons for the introduction of the communication simulator, which will allow us to simulate the effect of communication failures on the performance of the filters.

Therefore, one way to approach this problem is to develop the Hierarchic Clustering. The idea was initially proposed in [22] and [24], and further explored in the work of Plinval [1]. Plinval developed a hierarchic centralized-centralized architecture. The main characteristic of this architecture is two levels of hierarchy, with each of them running the centralized estimation filter. This thesis presents the hierarchic centralized-decentralized architecture and compare its performance with the other architectures, namely: centralized, decentralized and hierarchic centralized-centralized.

4.3.4 Definition of Hierarchic Architectures

The idea of hierarchic architecture is based on splitting the fleet into several sub-clusters, with one member of each sub-cluster belonging to the higher level of hierarchy, called the “super-cluster”, Figure 4-3. Both levels of the hierarchy independently perform their own estimation filters, giving rise to several architectural possibilities. The two levels of hierarchy can exchange information, which allows proper performance of the overall estimation process. This in essence splits the fleet into several independent clusters working separately from each other and their link is through the super-cluster. There is also the option of using the dynamical hierarchic architecture, which allows the sub-cluster members to range off the members of non-local sub-clusters. Each of the sub-clusters needs to have a sufficient number of measurements to work properly. However, for a small number of spacecraft, the hierarchic architecture may not be necessary. The reason is, as mentioned earlier, that the need for the hierarchic architecture appears when fleet scalability became a problem in centralized and decentralized architectures. This is one of the main reasons the simulation had to be scaled up to include more than 24 spacecraft.

The two hierarchic architectures to be described are: centralized-centralized and centralized-decentralized. The main difference between the two schemes is in the sub-clusters, which run different estimators.

The hierarchic centralized-centralized (HCC) architecture has two levels of hierarchy. The super-cluster is formed with a spacecraft from each cluster. Both levels of the hierarchy run centralized filter, represented by the EKF, described in chapter 2. In this case, the only necessary information sent to the central vehicles of each cluster are the absolute and relative measurements for the non-master vehicles. The full explanation of the cluster is given in [1].

The hierarchic centralized-decentralized (HCD) architecture is identical in the form to

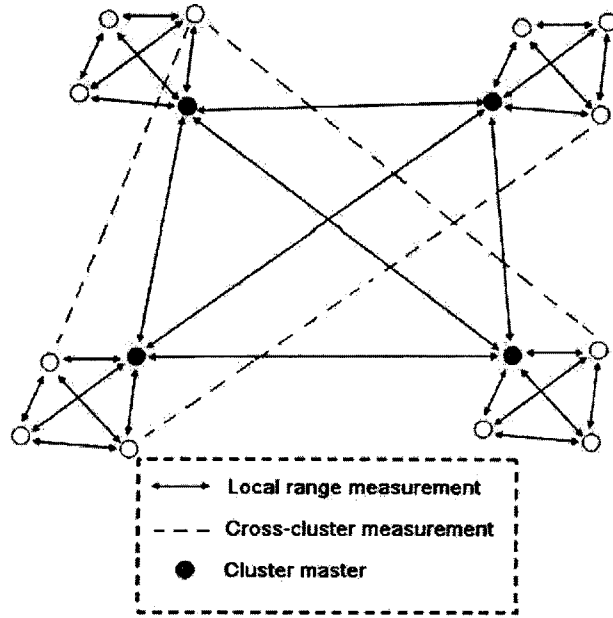


Figure 4-3: Hierarchic Clustering. Super Cluster and the sub-clusters may run different estimation algorithms.

the HCC architecture. The primary difference is that the sub-clusters in the HCD are all running decentralized algorithms. The decentralized algorithm running in the sub-clusters is identical to the decentralized algorithms also examined in this chapter. This algorithm is using the Schmidt-Kalman filter that was identified as the best performing algorithm in the work of Plinval [1]. The information circulating within the subcluster are absolute and relative measurements, states and corresponding blocks of state error covariance P . The equations for Schmidt-Kalman filter are presented in chapter3.

4.4 Effect of Communication Delays

Communication delay can play a major role in the performance of estimation algorithms for formation flying satellites that use relative measurements. Depending on the architecture,

different kinds of information needs to be transmitted from one location to the other. For example, the relative measurements (i.e. ranges and bearing angles) are obtained among all the possible spacecrafts. Therefore, certain range measurements are not available to every spacecraft in the fleet/cluster. This limitation applies to architectures with a centralized filter such as fully centralized architecture or hierarchic centralized-centralized and hierarchic centralized-decentralized (only in super-cluster) architectures, but not to decentralized architecture.

Sometimes the estimated states of a spacecraft need to be communicated to other members of the fleet, as is the case in decentralized architecture. More specifically, for the decentralized architecture using Schmidt-Kalman filter, parts of the state error covariance matrix need to be communicated to the other members of the fleet/cluster in order to avoid the problems described in [22] and proved in Chapter 3.

The next section describes the effects of communication delays on centralized, decentralized, hierarchic centralized-centralized, and hierarchic centralized-decentralized architectures.

4.4.1 Centralized Architectures

In general, centralized estimation architecture for formation flying spacecraft does not have a large communication load. However, for relative measurements to be incorporated in architectures with a centralized filter, they have to be communicated to the central processor (i.e. the master vehicle) of the fleet/cluster. Also, it is needed for those measurements to be sent at specific time intervals so that they are available for the processor before the next estimation process. Therefore, in order for the estimation process to proceed to the next estimation step, the central processor needs to complete its current estimation process and

wait for the relative ranges and bearing angles among the other vehicles to be communicated to it.

In most practical applications, the estimations need to be sent back to the remainder of the fleet/cluster for purposes other than the estimation process. This additional communication adds to the overall communication load of the centralized architecture. In the purely estimation scenario, the central processor does not need to send the estimation results back to the other members of the fleet/cluster. We will only focus on the communication requirement with respect to the estimation process.

Analysis The communication burden is defined as the amount of data transmitted throughout the whole fleet during one estimation step. In the centralized case, the data transmitted can be separated into two different sections. One is the transmission of all the measurements among all the spacecrafts to the central vehicle. The other is the transmission of calculated estimates from the central processor to the rest of the fleet.

During each estimation step, each vehicle collects relative measurements from other vehicles in the fleet and also collects absolute measurements. Since each vehicle has $N-1$ relative ranges, $N-1$ relative bearing angles, four absolute attitude measurements, and two absolute range measurements, this amounts to $2N + 4$ measurements. In total, there are $(N - 1) * (2N + 1)$ measurements that need to be communicated to the central processor. Since in the simulation we are using a type "double," which corresponds to 8 bits, the total amount of data sent is $8 * (N - 1) * (2N + 4)$.

As mentioned earlier, in centralized architecture, for estimation purposes, it is not necessary to communicate the calculated estimates from the central processor to the rest of the fleet. If this is necessary, then the state estimate vector of size 13 will be communicated back to all the members of the fleet, which would amount to $8 * 13 * (N - 1)$ bits of data.

4.4.2 Decentralized Architectures

Decentralized architecture was developed to address the issue of computational burden associated with centralized filters and, in effect, improve the scalability of the system. Various decentralized algorithms have been developed to populate decentralized architecture. Some algorithms (e.g. iterative cascade extended Kalman filter, ICEKF) requires each member of the fleet/cluster to communicate its calculated state estimate to the rest of the fleet/cluster. Since this is an iterative process, the high frequency of communication limits the amount of time spent on each computation. In [22], it was shown that communicating only state estimates in a decentralized scheme produced incorrect measurement matrix H and Kalman gain K , which led to unreliable results of the ICEKF filter.

To address this problem, Schmidt-Kalman filter was introduced as another decentralized filter, which required communicating blocks of state-error covariance matrix P . This increased amount of communication puts additional pressure on the already limited amount of time allowed for computation. Therefore, the communication delay associated with communicating pieces of state-error covariance matrix and state estimates can play a major role in the performance of a decentralized filter. Furthermore, it is this communication delay that prevents greater scalability.

In previous work [1], the effect of communication delays was not included in examining the performance of simulations of various filters. We intend to use a communication simulator to repeat the simulations and measure the effect of communication delays on various filters.

Analysis In decentralized architecture, each spacecraft runs its own estimation filter based on measurements available only to that vehicle. Therefore, unlike in the centralized case, the spacecraft that is performing the estimation does not receive communication of all the

measurements available to the rest of the fleet.

However, in the case of a decentralized filter such as the Schmidt-Kalman filter, each spacecraft needs to send its state estimate and state-error covariance matrix to every other vehicle in the fleet. Since this is an iterative process, this communication can occur several times during a single measurement update cycle. Therefore, if the number of iterations is denoted with k and the size of the state-error covariance matrix is 13×13 , then the amount of data communicated is $8 * k * N * (N - 1) * (13 + 13 * 13)$.

4.4.3 Hierarchic Architectures

The hierarchic architecture was developed as an attempt to address the scalability issue associated with centralized and decentralized architectures. By splitting fleets into clusters, we are allowing estimation filters to operate on a smaller number of spacecraft, therefore eliminating the need for a very scalable algorithm. Each sub-cluster is populated with an estimation algorithm and all the sub-clusters are coordinated among themselves through the super-cluster, which runs its own estimation algorithm. This gives rise to several filter combinations depending on which level of hierarchy the filter is incorporated; we will examine two of them: hierarchic centralized-centralized and hierarchic centralized-decentralized.

Hierarchic C-C Architectures

In hierarchic centralized-centralized architecture, each cluster (sub-clusters and super-clusters) runs a centralized filter. Likewise, it is affected by communication delays in a similar way as a fully centralized architecture is affected, with some differences. In addition to communicating relative measurements within clusters, hierarchic centralized-centralized architecture requires communication between the two levels of hierarchy. Communication between the two levels

of hierarchy is necessary in order to coordinate all the sub-clusters among themselves. If the inter-cluster relative measurements are allowed, this further adds to the communication load of the hierarchic centralized-centralized architecture.

Analysis In the hierarchic centralized-centralized architecture, the communication mostly occurs within the clusters themselves. Therefore, if the number of clusters is C , then the number of vehicles per cluster is equal to N/C .

Within a sub-cluster, each vehicle communicates all of its measurements to the master of the cluster. This amounts to $8 * [(N/C - 1) * (2N/C + 4)]$. If the cluster master sends the information back to the rest of the cluster, that would amount to $8 * 13 * (N/C - 1)$.

In the super-cluster, there will be C spacecrafts. Therefore, analogous to the sub-cluster analysis, the amount of information communicated to the master of the super-cluster is $C * (2C + 4)$. Again, if the super-cluster master sends computed state estimates back to the rest of the super-cluster, then the additional communication load is equal to $8 * 13 * (C - 1)$.

The total communication is therefore equal to the sum of the communication within the sub-clusters and the super-cluster.

Hierarchic C-D Architectures

Similarly to hierarchic centralized-centralized architecture, hierarchic centralized-decentralized architecture requires communication between the two levels of hierarchy. The only difference is that the sub-clusters run decentralized filters (e.g. Schmidt-Kalman filters, SKF). Therefore, the effect of communication delays on sub-clusters is similar to that described in decentralized architecture while the effect on the super-cluster is similar to that described in centralized architecture.

Analysis In hierarchic centralized-decentralized architecture, we have the combination of the centralized filter at the super-cluster level and the decentralized filter at the sub-cluster level. The communication load observed in this architecture can easily be calculated by combining the two analyzes of centralized and decentralized architectures.

In the super-cluster of the hierarchic centralized-decentralized architecture, the communication load is equal to $(C - 1) * (2C + 4)$. Again, if the super-cluster master sends computed state estimates back to the rest of the super-cluster, then the additional communication load is equal to $8 * 13 * (C - 1)$.

At the sub-cluster level where the decentralized filter runs, the communication load is equal to $8k * [N/C * (N/C - 1) * (13 + 13 * 13)]$.

4.5 Modified Simulation Results and Architecture Comparison

In this section, different architecture have been compared using the simulation results. The simulation setup will be described and the comparison will be performed against several metrics mentioned earlier in this chapter.

4.5.1 Simulation Setup

The simulation was performed using the core of the simulator presented in the work by Plinval [1] . The setup represents the SPHERES testbed with test space of $1m^3$. The SPHERES testbed is explained in more detail in chapter 6. All of the initial conditions presented in Plinval's work remain the same, except the initial velocity, which was increased in

order to put more emphasis on the dynamics of the system, as this can affect the performance of the systems with large computational or communication delays. The main additions to the simulator have already been described in previous sections of this chapter.

The measurements used are satellite-to-satellite range and elevation, and also range measurements from satellite-to-wall beacons (two beacons). The number of vehicles is varied between 4 and 25 for the comparison of all four algorithms. For the comparison of the two hierarchic algorithms the number of vehicles was varied between 16 and 50, since the size of the clusters in the hierarchic architecture increases with a square root of the total number of the vehicles. The simulation is performed in the in 3D space. The measurement covariance remains at 10^{-4} and process noise at 10^{-6} . The process noise was kept slightly above zero in order to allow the system to always track the true system by never disregarding the measurements [26].

4.5.2 Algorithm Comparison

Accuracy Comparison In the accuracy performance we can notice that for the low number of vehicles, Centralized architecture performs the best. This is as expected since the computational time for the small number of vehicles is comparable to the computational times for the other architectures. The decentralized architecture also performs worse for smaller systems, since not all information is available to all vehicles and the communication delay plays an important role.

The decentralized architecture performs very well for the larger number of vehicles. While the centralized architecture is affected by the large computational load, the decentralized architecture benefits by distributing this load. However, we do see that the communication and synchronization level in the decentralized case are quite high, which implies a high risk

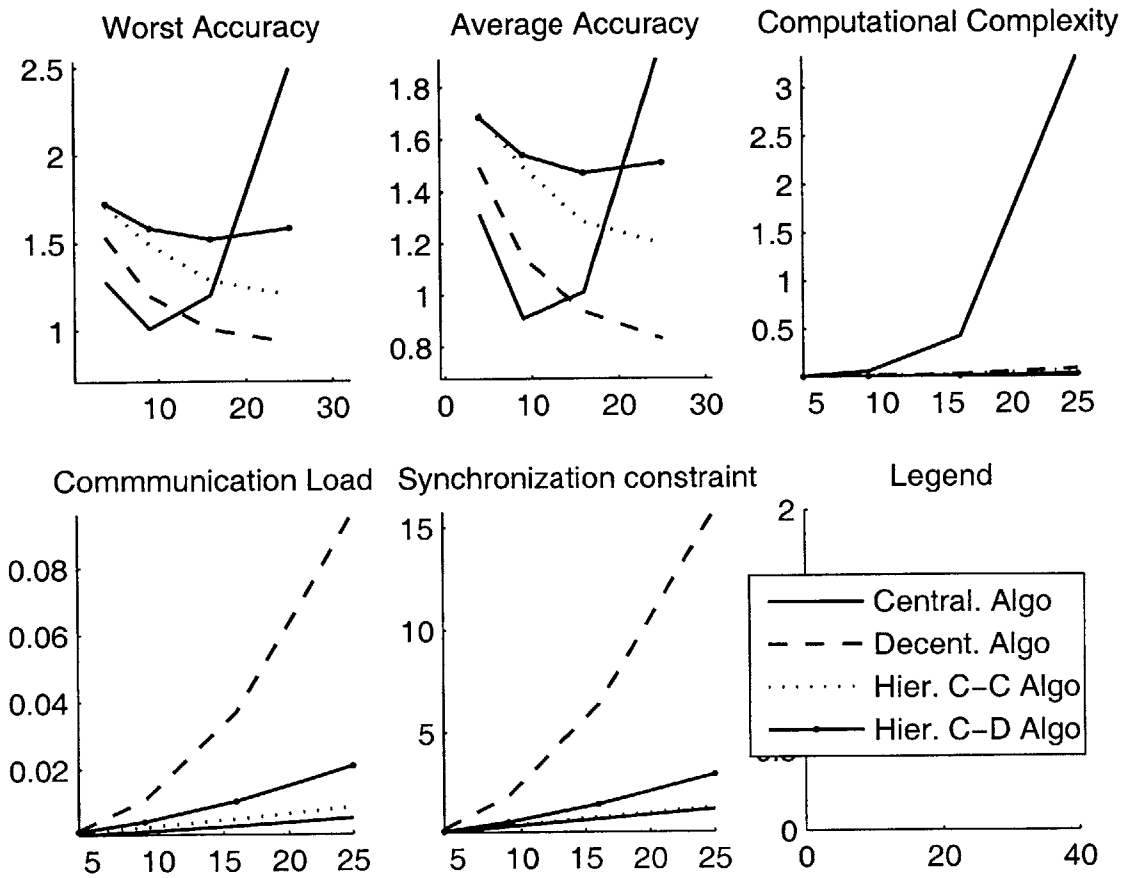


Figure 4-4: The comparison of various architectures: centralized, decentralized, HCC and HCD. The range of vehicles used in this comparison is from 4 to 25 spacecraft

of running into problems if the system has communication delays. In this simulation the communication was running without problems, and that allowed good results.

The two hierarchic architectures perform also very well and in general they have the best overall performance. Their accuracy is good and comparable with the other two architectures, in great deal thanks to the distributed computational load. Their lack of larger number of measurements (due to clustering) also results in a slightly degraded accuracy, but as the number of vehicles increases, this accuracy improves.

Comparison of the Computational loads As expected, the computational load for a centralized architecture is very large for large fleets. When the number of vehicles is lower, the computational load for the centralized architecture is somewhat comparable to the other architectures, which allows it to perform so well in those cases. However, as the computational effort increases the performance of the centralized architecture worsens.

All other architectures, including hierarchic centralized-centralized architecture have low computational loads. In the decentralized case, the computational load is spread across the fleet, while in the hierarchic cases the load is spread among the clusters.

Comparison of the Communication loads The communication load comparison reveals that the decentralized architecture indeed makes the highest strain on its communication systems. As the number of vehicles increases, the information flow increases. The amount of communication required by the hierarchic centralized-decentralized architecture is significantly smaller, as the decentralized scheme is running only on the sub-cluster level. The centralized and hierarchic centralized-centralized architecture are with the lowest communication loads as the primary source of communication load is from communicating the measurements to the cluster masters.

Synchronization comparison The synchronization graph shows that the decentralize architecture has the tightest synchronization. Of course, this means that small, unexpected perturbations to the system can cause the decentralized architecture to perform worse than it is shown in this simulation. On the other hand, the synchronization is significantly reduced in the hierarchic centralized-decentralized architecture, thanks to the cluster division of the fleet. The centralized and hierarchic centralized-centralized architectures have the lowest synchronization requirement. Thanks to this fact and also to the communication comparison,

we can see that hierarchic architectures distribute the computation, communication and synchronization in the most optimal way while not taking too much damage in performance, unlike the centralized architecture for example.

Summary This section shows that the centralized architecture achieves best accuracy for a small number of vehicles, but it quickly shows its lack of scalability due to enormous increase in computational load as the fleet size increases. The decentralized architecture performs the best in terms of accuracy when the number of vehicles increases. However, the decentralized scheme does have a large communication load and tight synchronization, which may lead performance degradation if the system is not well synchronized or if there are problems with communication. The hierarchic architectures are introduced to address these issues with centralized and decentralized architecture. The hierarchic centralized-centralized architecture performs best overall, keeping the accuracy high while not taking a hit on computational or synchronization performance. The hierarchic centralized-decentralized also has the overall advantage over the centralized and decentralized scheme, but does not perform as well as HCC.

4.5.3 Large scale architecture comparison

The simulator improvements allow us to run larger fleet sizes. This is particularly important, considering that one of the objectives of this technology is improvement in scalability. For fleet sizes up to 50 vehicles, we compared the two best overall architectures, the hierarchic centralized-centralized (HCC) and hierarchic centralized-decentralized (HCD) architectures.

As we can see the accuracy of the HCC actually improves as the number of vehicles increases. The reason is that the computation time is still very low, as the cluster size is

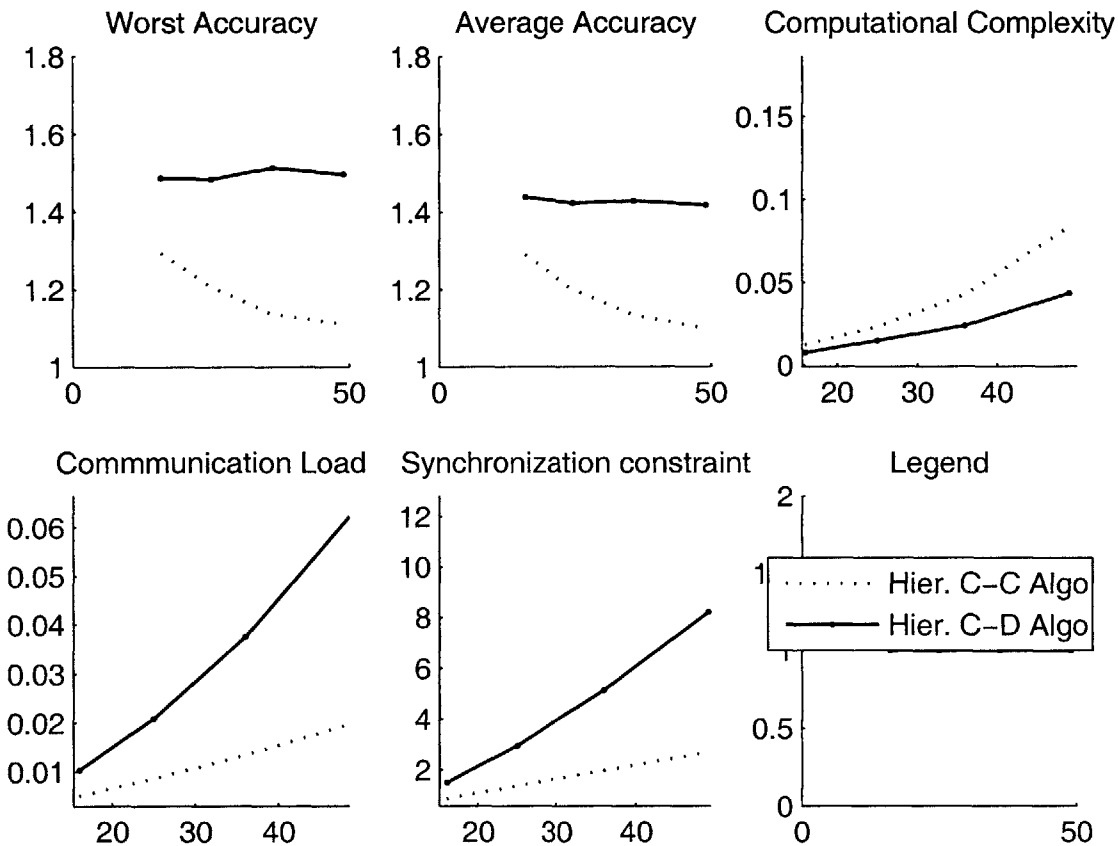


Figure 4-5: The comparison of the two hierarchic estimation architectures. The range of vehicles used in this comparison is from 16 to 50 spacecraft

approximately the square root of the total number of vehicles in the fleet. Therefore, while still having a low computational load, we are able to increase the number of measurements and the fleet size, while actually improving the performance of the fleet. The communication and synchronization levels remain low. Similarly, the HCD architecture performs quite well, following the trends established in the previous, smaller size fleet simulation. However, even for the larger fleet sizes, HCC performs better than the HCD. The main advantage for HCD architecture is the fact that the estimator doesn't rely on the single spacecraft in the cluster,

which may play an important role in the case of cluster master vehicle failures. This problem in HCC and centralized architectures can also be addressed by redistributing the tasks and capabilities to the next available vehicle in the fleet/cluster from the master vehicle that is experiencing failures.

4.6 Robustness of Estimation Architectures

In order to qualify certain system or algorithm as robust, robustness need to be defined. More specifically, the idea is to determine what the systems or algorithms need to be robust to. In our case, we want to focus on the robustness of estimation algorithm to communication failures and communication delays that may lead to temporary break in information exchange. It is well known that centralized architectures are extremely vulnerable to failures of leader vehicle. The answer to this vulnerability is to decentralize the fleet using decentralized estimators. However, decentralized estimators rely heavily on communication system. Therefore, this section will address the effect of communication problems on the performance of decentralized architectures. First of all, in order to qualitatively engage in this issue, some assumptions about the system need to be set.

4.6.1 Assumptions

1. We are considering an N-member fleet with the mission requirement of having M vehicles operational.
2. Probability of communication failure is p_{comm} .
3. All CPU's are the same, one per vehicle. Probability of CPU failure is p_{cpu} .
4. Probability of single vehicle failure due to causes other than communication and CPU

failure is p_{veh} .

5. Centralized fleet has only one vehicle capable of being leader (while having redundancy in that sense is possible, it is not considered in this analysis).
6. Probability of communication delay is modeled based on the amount of time delay (shorter delays occur more often - an assumption):

$$p_d = p_g e^{-T_d}, \quad (4.1)$$

where T_d is duration of time delay in *sec* and p_g is the constant that keeps the area under curve equal to 1 (Fig. 4-6). That is, the probability that a glitch occurs and delays communication for *at least* the time period of T_d is p_d . In order to simplify things, we will ignore the communication path delay $T_{path} = LD_{path}$, where D_{path} is approximately the communication path delay per km ($D_{path} \approx 3.5 \frac{\mu sec}{km}$).

7. Time required for regular communication between 2 satellites for the centralized case is T_{comm}^{cen} and for the decentralized is T_{comm}^{dec} , where $T_{comm}^{dec} > T_{comm}^{cen}$ due to larger information packages sent in the decentralized case.
8. Computation time for the centralized case is T_{cpu}^{cen} and in the decentralized T_{cpu}^{dec} , with T_{cpu}^{cen} generally being much larger than T_{cpu}^{dec} .
9. Time step T_{TS} is the shortest amount of time required for the filter to produce reasonable/usable results before causing science interruption.

4.6.2 Robustness

The robustness issue is very case-dependent and there are many different aspects of robustness. We will address the two most obvious and diverse cases of architectures – centralized

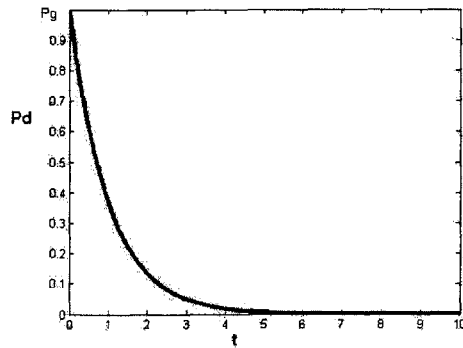


Figure 4-6: Probability distribution of communication delay

using EKF, and decentralized using Schmidt Kalman Filter – and also we will address different types of hierarchic architectures. We are looking at robustness with respect to failures that can cause either a mission failure, or a science experiment interruption. Also, for communication issues, one has to determine the wait-time after which if no data is being sent, one can conclude that the link is broken.

Some possible impacts of communication delays on mission performance are shown by J.A. Leitner et al [28] and Yu-Han Chen [29]. These two papers show that there indeed can be a serious impact on the performance.

In this analysis the robustness to the following failures will be considered:

1. Single vehicle failure.
2. Single communication delay
3. Communication channel failure.
4. CPU failure.

4.6.3 Centralized Architecture

We are considering an N-member fleet with the mission requirement of having M operational vehicles. In centralized case, it can be seen that type 3 failure can have quite similar effect as type 1 failure. If the *leader* vehicle loses the ability to communicate, or if its CPU fails, this is essentially the cause for mission failure and/or science interruption. However, a filter can adjust to failures of *follower* vehicles by removing their states and covariance blocks from computation as long as there are enough measurements available.

Therefore for a *leader* failure, the probability that the whole mission will fail or that there is a science interruption is:

$$P_{fail} = (p_{comm} + p_{cpu} + p_{veh}) \quad (4.2)$$

Otherwise:

$$P_{fail} = \binom{N-1}{M} (p_{comm} + p_{cpu} + p_{veh})^{N-M-1} \quad (4.3)$$

For the type 2 failure, or the communication delay, we have to note that in general this problem will not cause the failure of the mission, but it could cause a science interruption event. In this case, the maximum amount of time for data transmission is $T_{TS} - T_{cpu}^{cen}$. Since the *followers* are not doing any computation, that means each of them has $T_{TS} - T_{cpu}^{cen} - T_{comm}^{cen}$ amount of time for communication delay. We then conclude that the probability of a single spacecraft having a communication failure due to communication delay is

$$p_d = p_g e^{-T_d} \quad (4.4)$$

$$= p_g e^{-(T_{TS} - T_{cpu}^{cen} - T_{comm}^{cen})} \quad (4.5)$$

Therefore, the probability of having a science interruption event will be:

$$P_{si} = \binom{N-1}{M} p_d^{N-M-1} \quad (4.6)$$

since we need at least M vehicles to still communicate with the master in order for the mission to operate. Since we know that the computational time for centralized filters is not well scalable with the number of the satellites in the fleet, we can conclude that the communication delay can have a serious effect. However, since T_{cpu}^{cen} is in general much larger than T_d , the science interruption event is much more likely to happen due to large T_{cpu}^{cen} . This is a much larger problem that needs to be addressed, which is done by decentralization. In next section we will take a look how T_d will gain on importance due to the synchronization issue.

4.6.4 Decentralized Architecture

In the case of iterative decentralized approach, a CPU, communication or vehicle failures will have similar consequences. Since we are looking at the iterative method, once the failure is detected, the next vehicle can remove this satellite from the iteration and proceed with the previous information. Therefore, the mission is operational, as long as there are M working vehicles.

$$P_{fail} = \binom{N}{M} (p_{comm} + p_{cpu} + p_{veh})^{N-M} \quad (4.7)$$

There is also type 2 failures or the communication delays. Each vehicle needs to do some computation, and it has to iterate within the single time step. Assuming that we need k iterations to converge to a solution, then the maximum time that a single spacecraft can

experience a communication delay is very scarce:

$$T_d = \frac{T_{TS} - kN(T_{cpu}^{dec} + T_{comm}^{dec})}{kN} \quad (4.8)$$

$$= \frac{T_{TS}}{kN} - T_{cpu}^{dec} - T_{comm}^{dec} \quad (4.9)$$

Therefore the probability of a single spacecraft having a communication failure due to its communication delay is

$$p_d = p_g e^{-T_d} \quad (4.10)$$

$$= p_g e^{-\left(\frac{T_{TS}}{kN} - T_{cpu}^{dec} - T_{comm}^{dec}\right)} \quad (4.11)$$

This shows that the computational effort plays a very important role in iterative decentralized case as well, while before it was considered to mainly affect the performance of the centralized algorithms. T_{cpu}^{dec} is kept pretty much constant as N increases, but the maximum time allowed for computation is decreasing. The equation 4.11 gives a clear explanation as to why iterative decentralized estimation filters are not scalable, which their constant computation time would suggest. The mission can have a science interruption event due to communication delays with a probability:

$$P_{si} = \binom{N}{M} p_d^{N-M} \quad (4.12)$$

4.6.5 Robustness Analysis for Hierarchic Architectures

The notion of robustness can be applied to different hierarchic architectures. The motivation for the decentralized architectures arose from the need for a more scalable and single-point failure robust fleet, which was unachievable with the centralized architecture due to the

high computational efforts and single processor. Using the decentralized approach, we were able to distribute the computational load across the fleet; however, new problems were introduced in the decentralization procedure. The high level of synchronization required for the decentralized architectures leads to the introduction of hierarchic architectures. In the following robustness analysis, we will show how hierarchic architecture has advantages over the decentralized architecture with respect to synchronization issues.

At the very beginning, we will make an assumption that the fleet can be split into two levels of clusters, super-cluster and sub-clusters. We will assume for simplicity reasons that the total number of spacecrafts is $N = Q^2$, where Q is number of spacecrafts in both the super-cluster and sub-cluster. Also, we will focus mostly on the effects of communication failures on the robustness of the system, as the other types of robustness can be easily determined using the equations derived in the centralized and decentralized architecture sections of this chapter.

Centralized-Centralized Architecture

The Centralized-Centralized architecture may be defined in two different ways with respect to the synchronization of the super-cluster and sub-clusters. If the two levels of clusters are working in parallel, then the time delay for each of the clusters can be presented separately. Since both clusters are using centralized architecture, this example boils down to two centralized architecture analyzes presented earlier. For the super-cluster:

$$p_d = p_g e^{-T_d} \quad (4.13)$$

$$= p_g e^{-(T_{RS} - T_{sup_{cpu}^{cen}} - T_{sup_{comm}^{cen}})} \quad (4.14)$$

where T_{sup} refers to the time values of the super-cluster. Similarly for the sub-cluster:

$$p_d = p_g e^{-T_d} \quad (4.15)$$

$$= p_g e^{-(T_{RS} - T_{sub_{cpu}^{cen}} - T_{sub_{comm}^{cen}})} \quad (4.16)$$

where T_{sub} refers to the time values of the sub-cluster. It is important to realize that the values for T_{sup} and T_{sub} are much smaller than in the purely centralized case, since the number of the spacecraft is square root of the total number of the vehicles in the fleet [1]. These two equations show that the resulting probability of science interruption in either of the two levels of clusters is:

$$P_{si} = \binom{Q-1}{R} p_d^{Q-R-1} \quad (4.17)$$

where R is the number of vehicles that need to still communicate with the master in order for the mission to operate.

The second option would be that the two levels of clusters work sequentially. This would mean that the super-cluster waits for the sub-clusters to finish their estimation step, collect the needed information, and perform the update again. This would force the masters of each of the clusters (which are also the members of the super-cluster) to perform the estimation twice within the single time step, and that would lead to having a better estimate. The probability of having a delay due to the communication failure will be:

$$p_d = p_g e^{-T_d} \quad (4.18)$$

$$= p_g e^{-(T_{RS} - T_{sup_{cpu}^{cen}} - T_{sup_{comm}^{cen}} - T_{sub_{cpu}^{cen}} - T_{sub_{comm}^{cen}})} \quad (4.19)$$

Again, the probability of science interruption will be:

$$P_{si} = \binom{Q-1}{R} p_d^{Q-R-1} \quad (4.20)$$

While this approach makes the system less robust than the parallel approach, it is still an improvement over the centralized architecture.

Centralized-Decentralized Architecture

The Centralized-Decentralized architecture uses two different algorithms to solve the estimation problem on different levels of hierarchy. Again, we will observe two different approaches in synchronizing the two levels of clusters. The super-cluster running the centralized filter can run in parallel or in sequence with the sub-clusters that are running the decentralized architecture.

In the case of parallel execution, the equations for the super-cluster remain the same as in the previous section:

$$p_d = p_g e^{-T_d} \quad (4.21)$$

$$= p_g e^{-(T_{TS} - T_{sup_{cpu}^{cen}} - T_{sup_{comm}^{cen}})} \quad (4.22)$$

with the probability of science interruption occurring due to the communication problems at the super-cluster level:

$$P_{si} = \binom{Q-1}{R} p_d^{Q-R-1} \quad (4.23)$$

On the sub-cluster side, the decentralized algorithm will lead to the following equation:

$$p_d = p_g e^{-T_d} \quad (4.24)$$

$$= p_g e^{-\left(\frac{T_{TS}}{kQ} - T_{cpu}^{dec} - T_{comm}^{dec}\right)} \quad (4.25)$$

and the probability of science interruption at this level is:

$$P_{si} = \binom{Q}{R} p_d^{Q-R} \quad (4.26)$$

If the two levels of clusters are working sequentially, the probability of the communication delay occurring would be different for the two levels of clusters. In the super-cluster case the super-master needs to wait until the decentralized sub-clusters finish their estimation step:

$$p_d = p_g e^{-T_d} \quad (4.27)$$

$$= p_g e^{-\left(T_{TS} - T_{sup_{cpu}}^{cen} - T_{sup_{comm}}^{cen} - kQ(T_{sub_{cpu}}^{dec} + T_{sub_{comm}}^{dec})\right)} \quad (4.28)$$

On the other hand, the probability of communication delay occurring on a member of the sub-clusters is as follows:

$$p_d = p_g e^{-T_d} \quad (4.29)$$

$$= p_g e^{-\left(\frac{T_{TS} - T_{sup_{cpu}}^{cen} - T_{sup_{comm}}^{cen}}{kQ} - T_{cpu}^{dec} - T_{comm}^{dec}\right)} \quad (4.30)$$

While it may seem that this makes a very tight synchronization constraint, it is still better than in the iterative decentralized case. Also, $T_{sup_{cpu}}$ is still a reasonable amount of time, as the super-cluster master is working on estimating the states of Q vehicles, instead of Q^2 vehicles. The equations for the probability of science interruption are equivalent to the ones in the case of parallel execution.

Summary This section has shown how different delay terms, such as computational and communication time, may affect the probability of having science interrupt. It also shows that the decentralized architectures are not very robust to delays in communication, since the communication time may already pose significant pressure on the size of the time-step. This was also shown in Figure 4-4. Next section, 4.6.6, will show how frequent communication problems can affect the performance of the decentralized schemes.

4.6.6 Simulation

The simulation section presents the performance of estimation algorithms when there are problems with the communication system. The simulator remains the same as described earlier in section 4.5. The only difference now is that the communication among the vehicles is being interrupted randomly. The probability of communication interrupt is varied.

The communication system interrupts are introduced when the system reaches the steady state. When the probability of communication interrupts is kept low, the effect on the decentralized estimation scheme was very small. The decentralized estimator was able to perform very well even with occasional satellite not communication with the rest of the fleet for a short period of time.

However, Figure 4-7, shows the effect of the communication interruption on the decentralized and hierarchic centralized-decentralized architectures when the probability of having communication problems is high. This leads to long periods of time where large number of spacecrafts has no information from parts of the fleet. Since these two architectures highly depend on the communication of state estimate and state error covariance, their performance deteriorates. Clearly, under such conditions, the hierarchic centralized-centralized architecture shows the best performance.

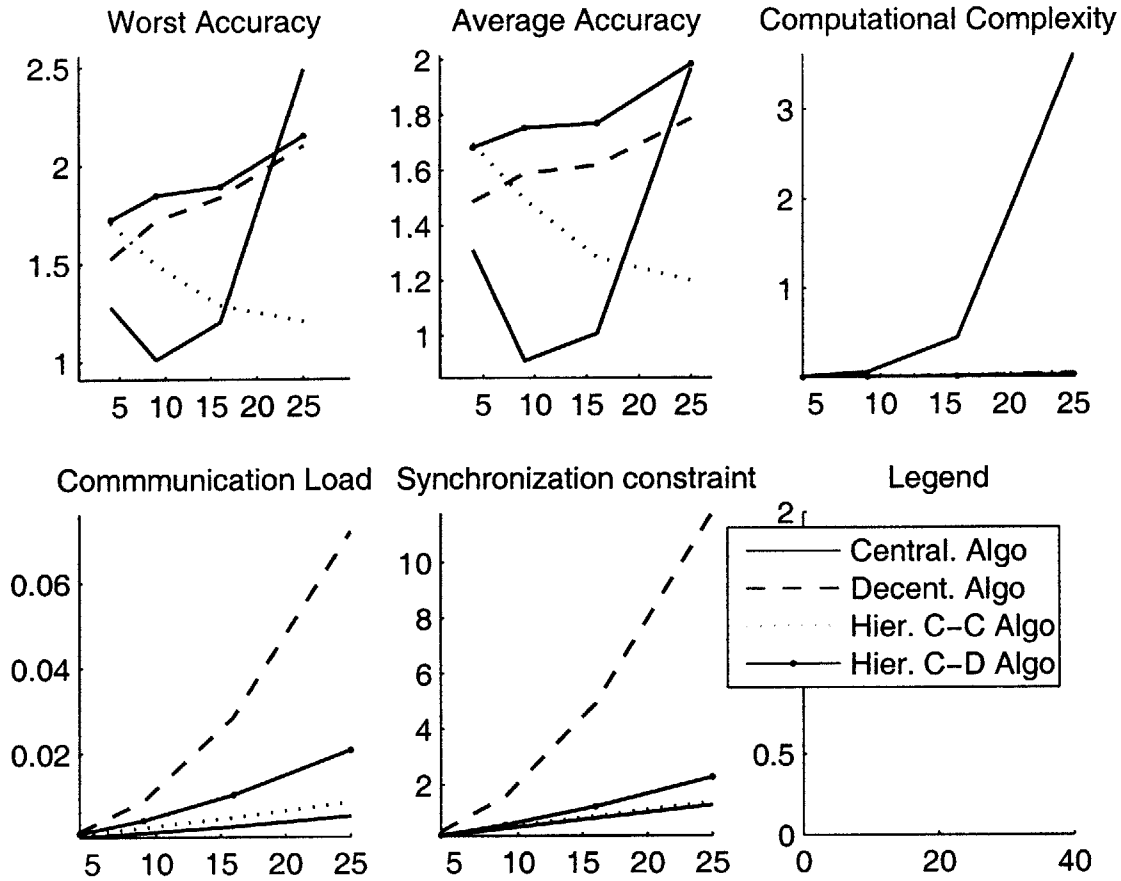


Figure 4-7: The performance comparison of various filters when communication delays are introduced

4.7 Conclusion

This chapter presented three improvements to the simulator presented in [1]. The first improvement increased the computational efficiency of the simulator, which allowed the fleets to run with double the size. Secondly, the communication simulator showed the effect of communication delays on the performance of various algorithms. It also allowed for brief analysis of robustness of various architectures to communication delays. Finally, the hierarchic

centralized-decentralized architecture was developed in order to compare the performance of two different hierarchic schemes.

In conclusion, the hierarchic centralized-centralized architecture seems to have the best overall performance against the five metrics used in this analysis. The HCC architecture also appears to be the most scalable architecture among the four discussed in this chapter.

The qualitative discussion of robustness of various architectures to communication delays showed that architectures incorporating decentralized architectures tend to have degraded performance when those delays are significant.

Chapter 5

Analysis of SPHERES Bias Problem and Calibration of SPHERES Positioning System

5.1 Introduction

This chapter presents work done on the SPHERES system. This chapter focuses on two main topics:

- The SPHERES bias problem
- Calibration of the SPHERES positioning system

First we describe the SPHERES testbed and its metrology system. This is followed by a discussion of the general bias problems and possible solutions for it. Also, we explain why those proposed methods of solving the bias problem are not applicable to the SPHERES bias problem.



Figure 5-1: SPHERES satellite

The chapter also presents a new calibration method for determining the positions of the wall beacons in order to facilitate the initialization of the metrology system and to make the overall system more flexible.

5.1.1 SPHERES testbed

The SPHERES testbed was developed by the MIT Space Systems Laboratory. This multiple-spacecraft testbed, designed to test and mature satellite formation flight control and estimation algorithms, is to be launched onboard the International Space Station (ISS) where the tests can be performed in a real environment [27, 35]. As such, this testbed is intended to mitigate the risk associated with the distributed autonomous spacecraft control, by providing a risk-tolerant medium for the development of control and estimation algorithms [30] The SPHERES testbed consists of several spacecraft, 5 ultrasonic beacons distributed around the test space and a desktop control station(Figure 5-2). Each of the spacecraft (also called "spheres", Figure 5-1) contains an onboard propulsion system, power system, communica-

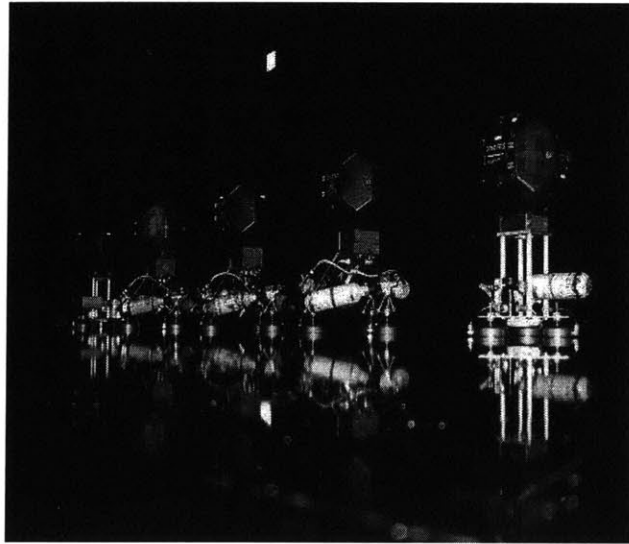


Figure 5-2: SPHERES 2D test at the MSFC

tions system and metrology system. The metrology system consists of 24 ultrasonic sensors distributed on the sphere's surface, and one ultrasonic beacon placed on the sphere's face corresponding to the negative X-axis. Currently, the testbed is located in the Space Systems Laboratory, in the 2-dimensional setup. The detailed description of the entire metrology system is provided in the Guest Scientist Paper [33, 34]; however, for the purposes of this chapter, the metrology system is described in more detail.

This chapter focuses on two major issues with this system. The first issue is related to the metrology system, which contains a significant amount of bias, which tends to spoil the measurements. The chapter presents the problem and several options that were considered in solving this problem. It also explains in detail the complex nature of these biases and why the offered solution would not be applicable and successful.

Also, we look at the SPHERES beacon position calibration technique. Currently, no procedure exists to automatically perform this calibration. The goal is to automatically determine the positions of the beacons in the test space, only by using the available metrology

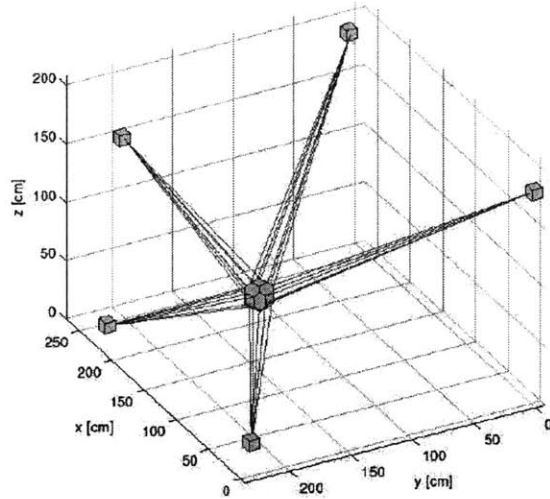


Figure 5-3: Schematic of SPHERES lab (2D) space, with four beacons mounted on the walls system and the estimation algorithms. This is a feasible idea, especially since relatively good initial beacon locations guesses, that are required for this technique, can be obtained from previously performed manual measurements.

5.1.2 The SPHERES Metrology System

The SPHERES metrology system is known as the Position and Attitude Determination System (PADS). It has inertial and global elements which provide information about the sphere's position and attitude. Inertial navigation sensors are accelerometers and rate gyroscopes. The global navigation sensors, which provide low frequency measurements of the sphere's position and orientation with respect to the laboratory reference frame consist of five fixed "wall" ultrasonic beacons and 24 ultrasonic receivers (microphones) mounted onboard of the spheres. The PADS global metrology provides each sphere with range measurements from five wall beacons to the receivers on the spheres surface. (Figure 5-3).

The process of updating the global measurements is initiated with the infrared signal. Once the signal is received by the spheres, they wait for the ultrasonic signal sent from the beacons, which transmit the signal sequentially, in specific time intervals. Once the signals are detected on the ultrasonic receiver end, spheres measure the “time of flight”, that is the time elapsed between the moment of beacon signal transmission and the signal reception on the spheres’ ultrasonic receivers. The schematic of the process by which the ultrasonic signal triggers the receiver is shown in (Figure 5-4). Also, each of the spheres is equipped with the onboard beacon, which allows determining inter-satellite the range and bearing. This is an important characteristic of the spheres system, as it allows to perform the wall beacon position calibration in the manner described in calibration section of this chapter. The receivers positions relative to the sphere’s center is given in [33].

5.2 The Techniques of Solving the Bias Problem

This section tries to address the bias problem occurring the PADS global metrology system. In the previous work with the spheres, [31], [30] it was determined that there are sources of biases in the range measurements between the wall beacons and receivers in the spheres. These biases were held responsible for degraded performance of the estimator. The spheres estimator was built robustly, meaning that the pre-filter was developed, which rejected all the poor measurement data that were entering the measurement matrix. This allowed the estimator to work with more relevant data. However, the biases in the states were still present.

There are several approaches to addressing the bias issue. Those are:

- Overbounding
- The Bias Estimation

- The Bias Elimination
- Schmidt-Kalman Filter

5.2.1 Overbounding

The overbounding method incorporates the bias in the states that are being estimated. This leads to a degraded performance of the estimator, since it cannot distinguish the bias from the state. The measurement equation for overbounding approach can be presented as following:

$$y = x + b + v \quad (5.1)$$

$$= x + v' \quad (5.2)$$

where x is the state of interest, b is the bias, v is the measurement error and $v' = v + b$ is the measurement error “bumped up” with the bias. When the system can permit degraded accuracy of the measurement, this could be the method to account for biases.

5.2.2 The Bias Estimation

Another standard approach to solving the bias problem is to attempt to estimate the biases [26]. This can be done by augmenting the state vector with biases, which are now treated as states themselves. Also, it requires sufficient number of measurements which allows these extra states (biases) to be estimated. The downside of this approach is that the state vector is increased, which will add to the computational burden and to the time required to perform the measurement update. The measurement equations corresponding to this approach are:

$$y = x + b + v \quad (5.3)$$

$$\text{and } x_1 = x, \quad x_2 = b \quad (5.4)$$

where x_1 and x_2 are the states, and v is a measurement error with a covariance R . Therefore, in this simplified case:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_k = \begin{bmatrix} c_1 & 1 \\ c_2 & 1 \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_k \quad (5.5)$$

where v_1 and v_2 are measurement errors with covariances R_1 and R_2 respectively.

5.2.3 The Bias Elimination

The biases can be eliminated if there are sufficient measurements of the same state. If the bias is considered non-variable, then by differencing the measurement equations, the biases can be discarded. This reduces the number of measurement equations, which is acceptable since we are eliminating the states in the same process. The main goal is to avoid estimating the biases, and to reduce the computational burden. Therefore:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_k = \begin{bmatrix} c_1 & 1 \\ c_2 & 1 \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_k \quad (5.6)$$

After differencing y_1 and y_2 :

$$\delta y = (c_1 - c_2)x_1 + \delta v \quad (5.7)$$

where δv is a measurement error with covariance $R = R_1 + R_2$. If the biases are very small compared to the measurement error, then this increase in measurement error covariance can be significant, and even further degrade the measurements. However, if the biases are significant, then this approach can be beneficial in attempting to decrease the state vector

size and therefore the computation associated with the estimation process.

5.2.4 The Schmidt-Kalman Filter

Another approach is the *Schmidt Kalman Filter* (SKF) [23]. The original application of the SKF is to eliminate the states of no interest from the estimation process, while still using some information about those states through the error covariance matrix. This allows reduction of the computational load on the processor. This elimination is accomplished by partitioning the measurement and propagation equations:

$$\begin{bmatrix} x \\ y \end{bmatrix}_{k+1} = \begin{bmatrix} \phi_x & 0 \\ 0 & \phi_y \end{bmatrix}_k \begin{bmatrix} x \\ y \end{bmatrix}_k + \begin{bmatrix} w_x \\ w_y \end{bmatrix}_k \quad (5.8)$$

$$z_k = \begin{bmatrix} H & J \end{bmatrix}_k \begin{bmatrix} x \\ y \end{bmatrix}_k + \nu_k \quad (5.9)$$

$$P_k = \begin{bmatrix} P_{xx} & P_{yx} \\ P_{xy} & P_{yy} \end{bmatrix}_k \quad (5.10)$$

where x represents the vector containing the states of interest. More detailed explanation of the SKF filter has been shown in Chapter 3. This approach is closely related to the bias estimation approach described earlier. The main difference is that the new states, that is the biases which augmented the state vector, are not being estimated as they have no practical importance for the system. However, the information about the biases is conserved through the error covariance matrix. Also, Schmidt-Kalman Filter requires observability of the biases for it to be successful.

5.2.5 Comparison of Bias Estimation and Elimination

The performance of the Schmidt-Kalman Filter is known to be sub-optimal since the approach deliberately eliminates states of no interest from the estimation process. As the equations in Chapter 3 show, the computed Kalman gain is sub-optimal. In addition to this, it is of interest to compare the performances of the two other methods: the bias estimation and the bias elimination.

To compare the performance of these two approaches, it is necessary to see how their error covariance behave in a steady state. In this case, a simplified example is observed using the system state model:

$$\mathbf{x}_k = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k \quad (5.11)$$

where: $x_1 = x$, $x_2 = b$, with b being the bias,

$$\mathbf{x}_{k+1} = \begin{bmatrix} \phi_1 & 0 \\ 0 & \phi_2 \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_k \quad (5.12)$$

and measurement equations:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_k = \begin{bmatrix} c_1 & 1 \\ c_2 & 1 \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_k \quad (5.13)$$

The assumptions are that the process error covariance is $Q = 0$ and that

$$\Delta \mathbf{x}_k = 0 \Rightarrow x_1 = \text{const and } x_2 = \text{const} \quad (5.14)$$

so that

$$x_1^{k+1} = x_1^k \text{ and } x_2^{k+1} = x_2^k \quad (5.15)$$

and $\phi_1 = \phi_2 = 1$. This set of equations simplify the time propagation equations for error covariance to

$$P_{k+1}^- = P_k^+ \quad (5.16)$$

Steady state derivation for the **bias estimation** only consists of the measurement update, because the states are random constants in this example:

$$H = \begin{bmatrix} c_1 & 1 \\ c_2 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \quad (5.17)$$

After setting coefficients to some arbitrary values, to simplify the derivation, $c_1 = 2$, $c_2 = 1$, $R_1 = R_2 = r$, $P_{10} = P_{20} = p$ (initial condition), these values can be inserted in the equation for Kalman gain:

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \quad (5.18)$$

The Kalman gain is then used to update the error covariance matrix:

$$P_k^+ = (I - K_k H_k) P_k^- \quad (5.19)$$

After iterating this process the resulting steady state error covariance is:

$$P_{ss} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \quad (5.20)$$

where the variances for states x_1 and x_2 are:

$$P_{11} = \frac{rp(2kp + r)}{k^2p^2 + 7kpr + r^2} \quad (5.21)$$

$$P_{11} = \frac{rp(5kp + r)}{k^2p^2 + 7kpr + r^2} \quad (5.22)$$

which was confirmed using Matlab symbolic toolbox. This shows that the error covariance tends to zero in steady state.

Steady state derivation for the **bias elimination** is also consisting of only measurement update. However, the system simplifies to scalar problem:

$$H = c_1 - c_2 = h, \quad R = R_1 + R_2 = r \text{ and } P_0 = p \quad (5.23)$$

After substituting these values in the equations 5.18 and 5.19, the steady state covariance becomes:

$$P_{ss} = \frac{pr}{kph^2 + r} \quad (5.24)$$

This result shows that in the error covariance behaves similarly in both cases when k grows very large, so there is no steady state performance disadvantage if biases are eliminated in the proposed way. On the other hand, if the biases are eliminated, the state vector is smaller and therefore the estimation process can be done faster and more efficiently.

5.3 The Sources of the Bias in SPHERES

Typical sources of bias in sensing systems are due to imperfect hardware. Examples of imperfect hardware are the analog receivers used in the SPHERES project. As it turned out, those analog receivers were a source of bias in the SPHERES project. Another source of bias originated from the bearing angles of the beacons with respect to the SPHERES satellites. Since this second type of bias was removed by creating a calibration map, this section focuses on the more important source of bias, which originates from the analog nature of the receivers.

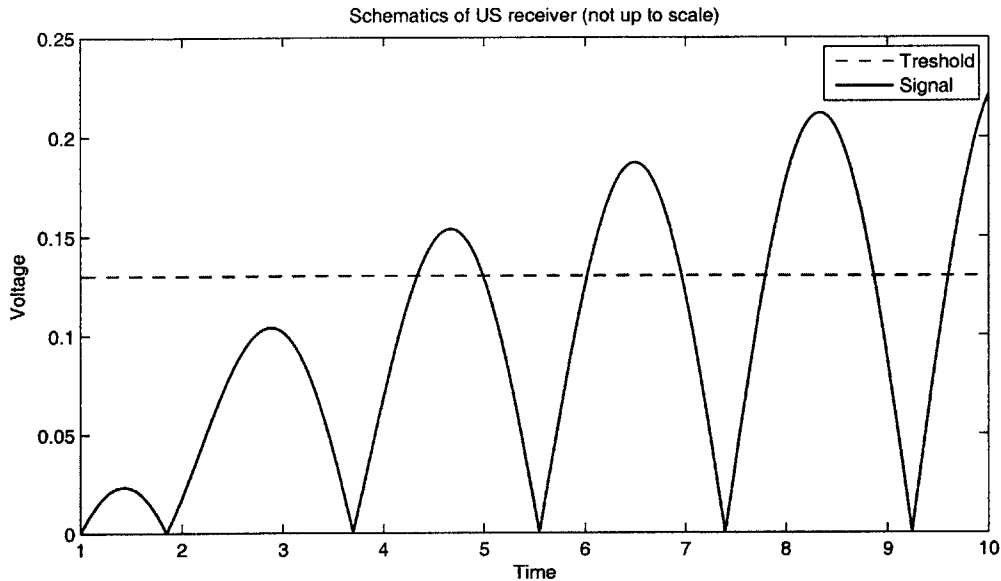


Figure 5-4: Receiver triggering schematics. The threshold is raised to a certain level in order to filter the noise.

5.3.1 Receiver Biases

The biases originating from the ultrasonic receivers are due to their analog nature (Figure 5-4), as confirmed by results obtained from the experiments performed by Serge Tournier. These biases are not constant but rather distance-dependent. The threshold level shown in Figure 5-4 indicates the signal intensity required for the measurement to be detected by the ultrasonic receiver. Since the signal strength is inversely proportional to the square of the distance, the signal strength drops quickly as the distance between the receiver and beacon increases. That means according to Figure 5-4, the time required for the signal to be detected increases as the distance between the receiver and beacon increases. The bias is equal to the time elapsed between the moment the signal reaches the receiver and when the receiver actually detects the signal.

As the distance between the beacon and the receiver increases and thus, the signal

strength decreases, it requires more half-waves to elapse before their amplitude reaches the threshold level (as shown in Figure 5-4). This leads to the variable bias that is dependent on the distance, as observed on Figure 5-5.

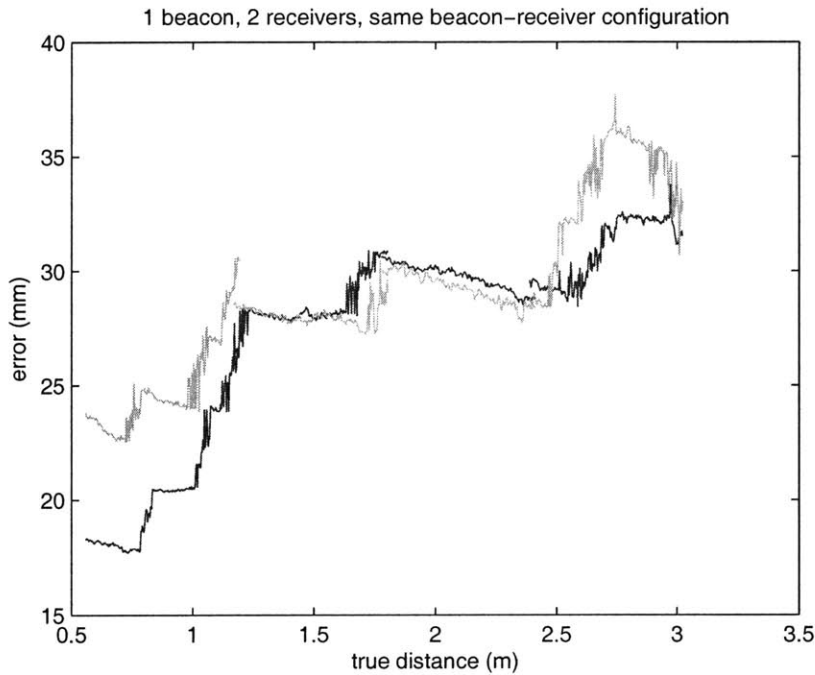


Figure 5-5: Error graph, from Serge Tournier (1 beacon, 2 receivers, same beacon-receiver configuration)

In addition to this type of bias, this hardware and the setup lead to another type of bias, which is of smaller scale and also low observability. This bias can be explained using the following example. When the distance is such that the signal reaches the threshold with exactly the tip of one of its half-waves, the slightest perturbation can cause the signal to be detected at the next half-wave. This is shown in the Figure 5-6. Since the distance between the two half-waves corresponds to approximately 5mm, this can cause additional bias in the measurement system.

These highly unobservable biases cause many problems when trying to account for them

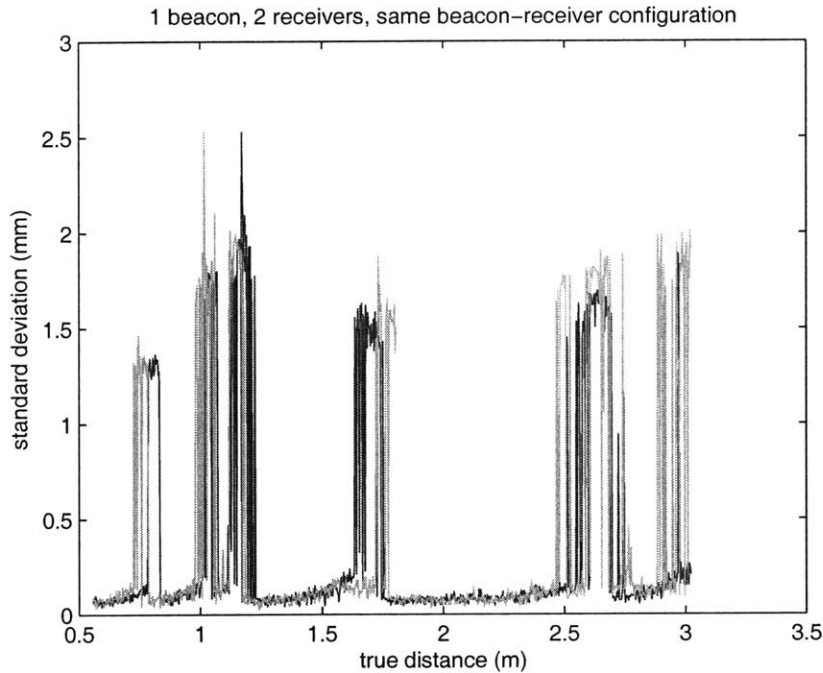


Figure 5-6: Standard deviation graph, from Serge Tournier (1 beacon, 2 receivers, same beacon-receiver configuration)

in an estimation process. As seen in Figure 5-5, not all receivers exhibit the same bias at each position. This means that there is a large number of biases in the measurement system. As a result, it is extremely time-consuming and computationally inefficient to attempt to identify the biases by using the bias estimation technique. The following section shows a simplified model of the SPHERES measurement system, which will study the possibility of solving the bias problem using one of the methods described earlier.

5.4 Simplified SPHERES Measurement System Setup

The setup consists of two beacons of known positions and one face of the SPHERES satellite. The SPHERES satellite includes four ultrasonic receivers located at the four corners of the

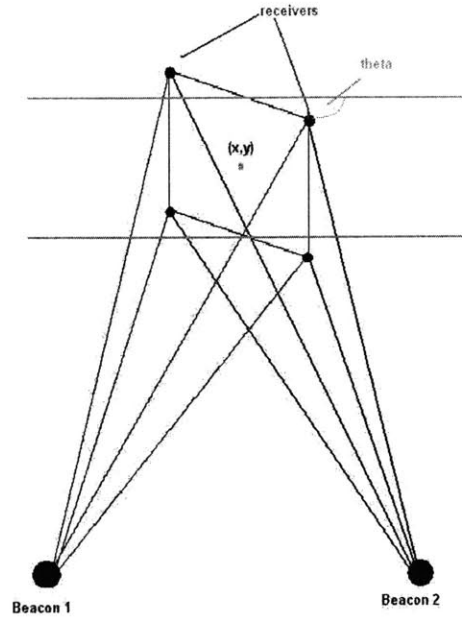


Figure 5-7: 3D setup (with z component constant) with 2 beacons and 4 receivers on a single SPHERES face

face. The setup is in two dimensions, where the z-axis is fixed to represent the lab setup of the SPHERES. The detailed configuration is schematically shown on figure (Fig. 5-7). Using this setup, we develop the measurement equations to analyze the system and to determine if any of the proposed methods can address the bias problem.

5.4.1 Measurement Equations

The measurement equations between the two beacons and four receivers are:

$$h_{11} = \sqrt{(x - x_1 - L \cos \theta)^2 + (y - y_1 + L \sin \theta)^2} + c_1 + b(\varphi) + b_{R1}(r_1) + v_1 \quad (5.25)$$

$$h_{21} = \sqrt{(x - x_1 + L \cos \theta)^2 + (y - y_1 - L \sin \theta)^2} + c_1 + b(\varphi) + b_{R2}(r_1) + v_2 \quad (5.26)$$

$$h_{31} = \sqrt{(x - x_1 + L \cos \theta)^2 + (y - y_1 - L \sin \theta)^2} + c_2 + b(\varphi) + b_{R3}(r_1) + v_3 \quad (5.27)$$

$$h_{41} = \sqrt{(x - x_1 - L \cos \theta)^2 + (y - y_1 + L \sin \theta)^2} + c_2 + b(\varphi) + b_{R4}(r_1) + v_4 \quad (5.28)$$

$$h_{12} = \sqrt{(x - x_2 - L \cos \theta)^2 + (y - y_2 + L \sin \theta)^2} + c_1 + b(\gamma) + b_{R1}(r_2) + v_5 \quad (5.29)$$

$$h_{22} = \sqrt{(x - x_2 + L \cos \theta)^2 + (y - y_2 - L \sin \theta)^2} + c_1 + b(\gamma) + b_{R2}(r_2) + v_6 \quad (5.30)$$

$$h_{32} = \sqrt{(x - x_2 + L \cos \theta)^2 + (y - y_2 - L \sin \theta)^2} + c_2 + b(\gamma) + b_{R3}(r_2) + v_7 \quad (5.31)$$

$$h_{42} = \sqrt{(x - x_2 - L \cos \theta)^2 + (y - y_2 + L \sin \theta)^2} + c_2 + b(\gamma) + b_{R4}(r_2) + v_8 \quad (5.32)$$

(Note: Subscripts for h are (receiver number, beacon number)) Therefore, the linearized measurements are:

$$z_1 = \frac{\partial h_{11}}{\partial x} x + \frac{\partial h_{11}}{\partial y} y + \frac{\partial h_{11}}{\partial \theta} \theta + b(\varphi) + b_{R1}(r_1) + v_1 \quad (5.33)$$

$$z_2 = \frac{\partial h_{21}}{\partial x} x + \frac{\partial h_{21}}{\partial y} y + \frac{\partial h_{21}}{\partial \theta} \theta + b(\varphi) + b_{R2}(r_1) + v_2 \quad (5.34)$$

$$z_3 = \frac{\partial h_{31}}{\partial x} x + \frac{\partial h_{31}}{\partial y} y + \frac{\partial h_{31}}{\partial \theta} \theta + b(\varphi) + b_{R3}(r_1) + v_3 \quad (5.35)$$

$$z_4 = \frac{\partial h_{41}}{\partial x} x + \frac{\partial h_{41}}{\partial y} y + \frac{\partial h_{41}}{\partial \theta} \theta + b(\varphi) + b_{R4}(r_1) + v_4 \quad (5.36)$$

$$z_5 = \frac{\partial h_{12}}{\partial x} x + \frac{\partial h_{12}}{\partial y} y + \frac{\partial h_{12}}{\partial \theta} \theta + b(\gamma) + b_{R1}(r_2) + v_5 \quad (5.37)$$

$$z_6 = \frac{\partial h_{22}}{\partial x} x + \frac{\partial h_{22}}{\partial y} y + \frac{\partial h_{22}}{\partial \theta} \theta + b(\gamma) + b_{R2}(r_2) + v_6 \quad (5.38)$$

$$z_7 = \frac{\partial h_{32}}{\partial x} x + \frac{\partial h_{32}}{\partial y} y + \frac{\partial h_{32}}{\partial \theta} \theta + b(\gamma) + b_{R3}(r_2) + v_7 \quad (5.39)$$

$$z_8 = \frac{\partial h_{42}}{\partial x} x + \frac{\partial h_{42}}{\partial y} y + \frac{\partial h_{42}}{\partial \theta} \theta + b(\gamma) + b_{R4}(r_2) + v_8 \quad (5.40)$$

The terms are:

1. X: x position of center of the SPHERES face

2. Y : y position of center of the SPHERES face
3. θ : Angle of rotation about z -axis
4. x_1, x_2, y_1, y_2 : x and y positions of the centers of the beacons 1 and 2
5. c_1, c_2 : constants for z -axis displacement of receivers
6. $b(\varphi)$: bias due to bearing angle (φ) of beacon 1
7. $b(\gamma)$: bias due to bearing angle (γ) of beacon 2
8. $b_{Ri}(r_1)$: bias due to receiver i ($i = 1, \dots, 4$) at distance r_1 from beacon 1
9. $b_{Ri}(r_2)$: bias due to receiver i ($i = 1, \dots, 4$) at distance r_2 from beacon 2
10. v_j : measurement error ($j= 1..8$)

We have 8 equations and 13 unknowns, but only 3 states (x , y , and θ) are of interest. The assumption is that the beacon-to-receiver distance is much larger than receiver-to-receiver, so that the bearing angle φ from beacon 1 to each receiver is approximately the same (similar assumption can be applied for bearing angle γ from beacon 2). Trying to go to 3D increases the number of unknowns, and make bias analysis even harder. Another difficulty arises from the variable biases, although they are not explicitly included in the equations. Even the fact that $b(\varphi)$ and $b(\gamma)$ have been removed using the calibration map does not improve the situation since there are still more unknowns than the equations.

5.5 Resolving the Bias Problem in SPHERES

In previous parts of the chapter 5.2 we described several different methods of solving estimation problems with biased measurements. All of these cases require that the biases of the

measurements are observable. For example, in the bias estimation method, the biases are considered states and therefore, in order for the estimation process to be successful, we need a sufficient number of measurements.

We also showed the source of bias in the SPHERES system. The tests done by Tournier show that due to the analog nature of the receivers and the measurement system in general, these biases are not only dependent on the distance but can even oscillate at certain distances. The simplified 2D model presented in this chapter also shows all the measurement equations and corresponding biases according to the measurement system used in SPHERES.

The immediate conclusion is that there are more unknowns than the measurements available from that system. Unfortunately, when faced with such a scenario, very little can be done in the estimation process to account for those biases. Therefore, all of the proposed methods fail to enhance the estimation results. For the bias estimation method, the cause of failure has already been described (lack of measurements). Similarly, in the bias elimination method, we also lack measurements, because when we try to eliminate a bias, we also lose a measurement. The Schmidt-Kalman filter also needs more measurements, because it requires observability of the biases [23].

The only method that does not fail is the over-bounding method, because it ignores the biases, or more precisely, they are included as part of the measurements. This method has essentially already been used on SPHERES and produced relatively poor results, which led to the search for better methods.

The only remaining approach to solving this problem is to create a look-up table. This table is used before the measurements are included in the estimation process. In this table, values of range measurements correspond with values of bias, which are measured with Tourniers approach. While this approach leads to eliminating the most significant biases

that are dependent on the distance, the variable bias that occurs at certain distances will still be present.

5.6 Calibration of the SPHERES Positioning System

In general, the calibration of the positioning systems is essential for their proper performance. Otherwise the system is very inflexible, and poorly transferable. Calibration of the SPHERES measurement system can be an important part of the SPHERES initialization process. In this calibration process, we attempt to determine the positions of the wall beacons using the measurement system itself. This can be done, with certain restraints, by placing the sphere in various positions and collecting measurements. After a sufficient number of measurements, we are able to determine the positions of all beacons.

The calibration process has not been used on SPHERES, since it was believed that the beacon positions were well-known in the lab space. This assumption turned out to be false with one of the beacons, creating the need for the calibration process. Furthermore, the SPHERES are sent on the International Space Station, where they perform tasks in a different environment that will likely require different beacon positions. The calibration process bypasses the need to know exact positions of the beacons, thus making the SPHERES system much more flexible.

In this section, we present the requirements that need to be satisfied in order for the calibration process to be successful. We will also describe the calibration process and the idea behind it, and finally, we will present the simulation results.

5.6.1 Calibration Approach and Requirements

As mentioned before, the calibration is performed in the following way. First, a sphere is placed at a random position in the test space. The measurement system is then initiated and the measurements are collected. Following that, the sphere is moved to a different position and a new set of measurements are collected. After a certain number of measurements are collected, the number of unknowns accumulated during the position-changing process will be equal to or smaller than the number of measurements collected. For this to be possible, at every sphere position where the measurements are collected, the number of new measurements must be greater than the number of new unknowns. If we are working in the 3D space, the number of new measurements acquired at each test position must be greater than 3. For example, in the SPHERES case, there are 5 wall beacons. This means that every time the sphere changes its position in the test space, we acquire 5 new range measurements and 3 new unknowns. Therefore, in the SPHERES case, this calibration process is possible.

Also, it is important to determine how many test positions are required for collecting enough measurements. This can be obtained using following derivation:

$$3n_b + 3N_p \leq n_b N_p \quad (5.41)$$

$$N_p \geq \frac{3n_b}{n_b - 3} \quad (5.42)$$

where N_p is the number of required test positions for the sphere, and n_b is the number of beacons for which we are determining positions and $n_b > 3$. Therefore, if we apply this equation to the SPHERES system ($n_b = 5$) we get:

$$N_p \geq 7.5 \Rightarrow N_p^{min} = 8 \quad (5.43)$$

It is, however, possible to decrease the number of unknowns and therefore the number of necessary test positions, by using the beacons to determine a coordinate system:

1. The coordinate center is placed at the position of one the beacons
2. The beacon at the coordinate center and another beacon define an x-axis
3. The two beacons on the x-axis and another beacon define an x-y plane

This procedure decreases the number of unknowns by 6, and Equation 5.45 becomes:

$$3n_b - 6 + 3N_p \leq n_b N_p \quad (5.44)$$

$$N_p \geq \frac{3n_b - 6}{n_b - 3} \quad (5.45)$$

$$\geq 3 + \frac{3}{n_b - 3} \quad (5.46)$$

Therefore, in the SPHERES case ($n_b = 5$):

$$N_p \geq 4.5 \Rightarrow N_p^{min} = 5 \quad (5.47)$$

In addition to this, the SPHERES should be positioned in order to maximize the observability of the system. Otherwise the estimation filter might not be able to converge. For example, two test positions should not overlap, as no new information can be provided from a second test position.

One benefit of the SPHERES measurement system is that it allows us to treat the spheres as a single point. The spheres are capable of determining the position of a beacon in its own coordinate system, thus allowing us to collect the exact range measurement from the center of the sphere to the beacon. This is possible because the face of each sphere has multiple

receivers due to which the spheres are capable of computing the two bearing angles from a beacon to the sphere in addition to the range measurement [32].

5.6.2 The Estimation Process

To perform the calibration, the collected measurements need to be fed to the estimation filter in order to determine the positions of the beacons. The system has no dynamics, the measurements are non-linear and are all collected before running the filter. Therefore, one good approach to solving this problem is using the weighted non-linear least squares approach. The algorithm is very similar to the linear least squares approach with the measurement matrix H being replaced by H_x which is the Jacobian of $\mathbf{h}(\mathbf{x})$.

Weighted Non-linear Least Squares Algorithm

This algorithm is in the essence the Newton-Ralphson method. It is laid out as follows [36]:

1. Make an initial guess for \mathbf{x}
2. Compute $\mathbf{h}(\mathbf{x})$ and H_x
3. Compute the covariance of the estimate, P using:

$$P_1 = (P_0^{-1} + H_x^T R^{-1} H_x)^{-1} \quad (5.48)$$

4. Compute the gradient, G_R at current \mathbf{x} , using definition of G_R :

$$G_R = P_0^{-1}(\mathbf{x} - \mathbf{x}_0) - H_x^T R^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x})) \quad (5.49)$$

where P_0 is the covariance of the prior estimate, \mathbf{x}_0 is the prior estimate, and R is the measurement error covariance.

5. Optimal value of \mathbf{x} is computed when $G_R = 0$. To make $G_R = 0$ we compute:

$$\mathbf{dx} = -P_1 G_R \quad (5.50)$$

6. Stop and exit the loop if $\|\mathbf{dx}\| < \epsilon$. This means that the algorithm stops when the change in \mathbf{x} is insignificant.
7. Otherwise, add \mathbf{dx} to \mathbf{x} .

This algorithm has been applied to this problem and produced good results, which are shown in the simulation section.

5.7 The Calibration Simulation Results

The weighted non-linear least squares method has been simulated using Matlab. The test positions have been selected randomly. The following table shows the performance of the algorithm described earlier with minimum (in our case, $N_P = 5$) or more test points used and also with the measurements accuracies σ_ρ varied. The performance is measured with the average error, which is the sum of distances of each of the final beacon position estimate to its true position divided by the number of beacons. It is also averaged over a large number of simulations (with the non-convergent cases excluded). The results are presented in Table 5.1, which lead to several observations. First, if a larger number of test locations for a sphere are considered, the results tend to be more accurate. Also, the method converges more often with a larger number of test locations, as it can work with a larger amount of information,

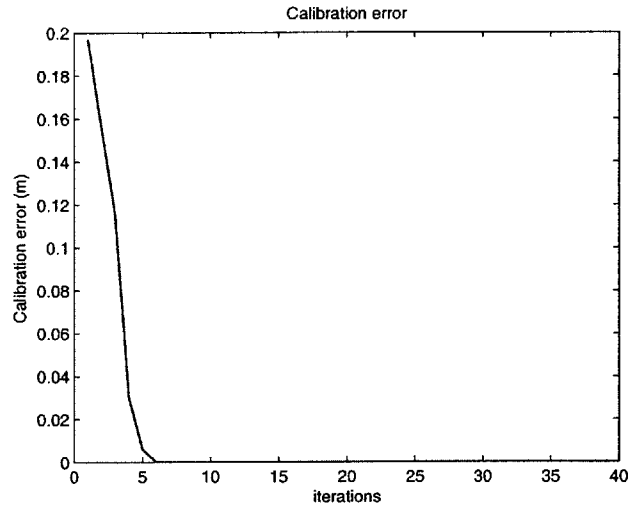


Figure 5-8: Evolution of error in beacon positions

Table 5.1: Calibration results

N_p	$\sigma_\rho(m)$	Average Error (m)
5	0.01	0.098
5	0.001	0.021
8	0.01	0.043
8	0.001	0.004
12	0.01	0.029
12	0.001	0.002

and therefore be less sensitive to possible overlap of test locations.

Another conclusion can be drawn from the calibration table. If the measurement system is more sophisticated and provides a better accuracy, the calibration method gives better results. For example, the average error for the case of 12 test location and $\sigma_\rho = 0.001$ leads to an average error of only slightly more than 0.2cm, Figure 5-8.

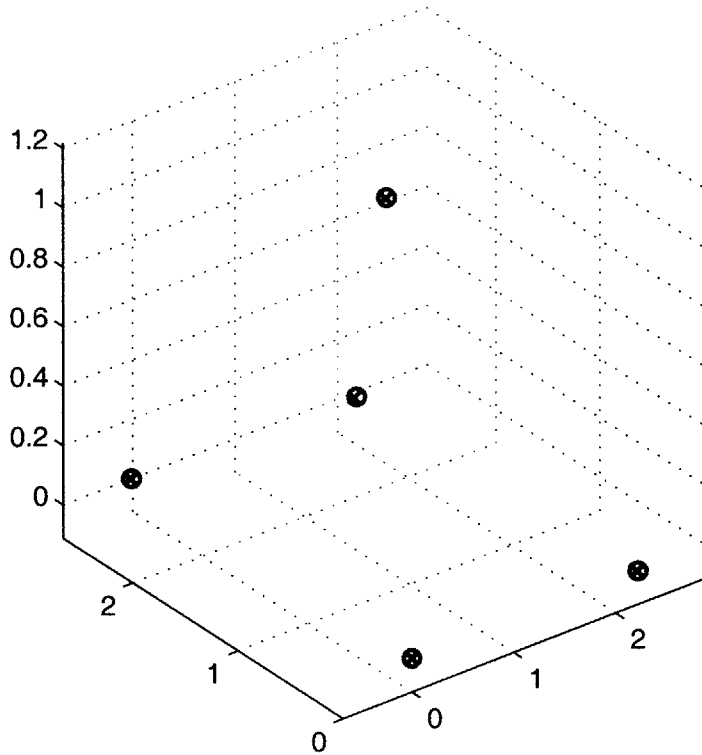


Figure 5-9: Performance of the proposed calibration approach. The “x” are true positions of the beacons, and “o” are the estimated beacon positions

5.8 Calibration Conclusion

Section 5.6 offered a new way of defining the coordinate system of the SPHERES test space. Currently, the positions of the transmitters are determined for the lab space, without the use of the positioning system. Using this calibration method, the SPHERES have a much better flexibility in the event of possible change of the test space, either for the ground tests, the KC-135 tests or for the incoming ISS testing. The calibration using the weighted non-linear least squares method has performed very well with the use of better measurement system and with a larger number of test points. Figure 5-9 shows the performance of this approach.

Chapter 6

Conclusion

This thesis investigates various topics. First, it focuses on divergence issues with the extended Kalman filter (EKF), which is extensively used in space science missions. The EKF is also used for centralized architecture, which is compared along with decentralized, hierarchic centralized-centralized, and hierarchic centralized-decentralized architectures. Several metrics are used to compare the architectures: accuracy, computational load, communication load, and synchronization. This work shows that centralized architecture is poorly scalable due to the rapidly increasing computational load as the fleet increases. Decentralized architecture attempts to distribute this computational load at the expense of a small reduction in accuracy. The analysis of various decentralized algorithms that populate the decentralized architecture is also presented in this work and it singles out the bump-up method (the Schmidt-Kalman filter) as the most optimal way of distributing the computational load throughout the fleet.

However, decentralized architectures also require a lot of communication and tight synchronization. The robustness discussion in chapter four explains the disadvantages of decentralized architecture due to possible communication delays. The hierarchic architecture

appears to have the most optimal balance between the performances against the four metrics. This specifically applies to hierarchic centralized-centralized architecture.

Finally, some special topics are presented for the SPHERES system. The SPHERES measurement bias problem is described along with possible approaches to solve it. However, due to the specific nature of the problem, these estimation approaches failed to solve it. Also, the SPHERESs beacon positioning calibration technique is investigated and shown to be a very good approach to determining the positions of beacons in the SPHERES testing area. This approach can provide the SPHERES system with great flexibility for future experiments.

Bibliography

- [1] Plinval, H. *Analysis of Relative Navigation Architectures for Formation Flying Spacecraft*. S.M. thesis, Massachusetts Institute of Technology, Dept. of Aero/Astro, Jan. 2006.
- [2] Kasdin N.J. *New, Guaranteed Positive Time Update for the Two-Step Optimal Estimator*. Journal of GNC, Vol.23, No.6, Nov.-Dec. 2000.
- [3] N. F. Toda, F. H. Schlee, and P. Obsharsky, "Region of Kalman Filter Convergence for Several Autonomous Navigation Modes," AIAA Journal, Vol. 7, April 1969, pp. 622627.
- [4] F. H. Schlee, C. J. Standish, and N. F. Toda, "Divergence in the Kalman Filter," AIAA Journal, Vol. 5, June 1967, pp. 11141120.
- [5] W. F. Denham and S. Pines, "Sequential Estimation When Measurement Function Nonlinearity Is Comparable to Measurement Error," AIAA Journal, Vol. 4, June 1966, pp. 10711076.
- [6] W. S. Widnall, Enlarging the Region of Convergence of Kalman Filters Employing Range Measurements, AIAA Journal, Vol. 11, March 1973, pp. 283287.
- [7] Lisano M.E. *Comment on "New, Guaranteed Positive Time Update for the Two-Step Optimal Estimator"*. Journal of GNC, Vol.23, No.6, Nov.-Dec. 2000.

- [8] Huxel P.J., Bishop R.H. *Fusing Inertial and Relative Range Measurements for Inertial Navigation in the Presence of Large State Error Covariances*. Proceedings of the 16th AAS/AIAA Space Flight Mechanical Conference, Jan.22-26, 2006.
- [9] J. Leitner, F. Bauer, D. Folta, R. Carpenter, M. Moreau, and J. How, "Formation Flight in Space," *GPS World*, Feb. 2002, pp. 22-31.
- [10] E.C. Aldridge, Jr et al "A Moon, Mars and Beyond. A Journey to Inspire, Innovate and Discover," Report of the Presidents Commission on Implementation of United States Space Exploration Policy, June 2004.
- [11] L. Breger, P. Ferguson, J. P. How, S. Thomas, and M. Campbell, "Distributed Control of Formation Flying Spacecraft Built on OA," Proceedings of the *AIAA Guidance, Navigation, and Control Conf.*, August 2003. (Paper 2003-5366)
- [12] F. D. Busse, *Precise Formation-State Estimation in Low Earth Orbit using Carrier Differential GPS*. Ph.D. thesis, Stanford University, Dept. of Aero/Astro, Nov. 2002.
- [13] T. Corazzini, *Onboard Pseudolite Augmentation for Spacecraft Formation Flying*. Ph.D. Dissertation, Stanford University, Dept. of Aeronautics and Astronautics, Aug. 2000.
- [14] C.-W. Park, J. How, and L. Capots, "Sensing Technologies for Formation Flying Spacecraft in LEO Using Inter-Spacecraft Communications System," the *Navigation J.* of the Institute of Navigation, Vol. 49, No. 1, Spring 2002, pp. 45-60.
- [15] J. R. Carpenter, C. Gramling *et al*, "Relative Navigation of Formation-Flying Satellites," from the proceedings of the *International Symposium on Formation Flying*, Toulouse France, October 2002.
- [16] P. Stadter, R. Heins, et al., "Enabling Distributed Spacecraft Systems with the Crosslink Transceiver," *AIAA Space Conference and Exposition*, Aug. 2001. Paper 2001-4670.

- [17] G. Purcell, D.Kuang, S. Lichten, S.C. Wu and L. Young, "Autonomous Formation Flyer (AFF) Sensor Technology Development," TMO Progress Report 42-134, August 1998.
- [18] C. Park and J. P. How, "Precise Relative Navigation using Augmented CDGPS," *ION-GPS Conference*, Sept. 2001.
- [19] P. Ferguson, T. Yang, M. Tillerson and J. How, "New Formation Flying Testbed for Analyzing Distributed Estimation and Control Architectures." Presented at the *AIAA GNC*, Monterey, CA, August 2002.
- [20] P. Ferguson and J. P. How, "Decentralized Estimation Algorithms for Formation Flying Spacecraft," *Proceedings of the AIAA Guidance, Navigation, and Control Conf.*, August 2003. (Paper 2003-5442)
- [21] C. W. Park, *Precise Relative Navigation using Augmented CDGPS*. Ph.D. thesis, Stanford University, Dept. of Mech. Eng., June 2001.
- [22] P. Ferguson, *Distributed Estimation and Control Technologies for Formation Flying Spacecraft*. S.M. thesis, Massachusetts Institute of Technology, Dept. of Aero/Astro, Jan. 2003.
- [23] R. Brown, P. Hwang, Introduction to Random Signals and Applied Kalman Filtering. Third Ed., pp. 366-7, John Wiley & Sons, 1997.
- [24] M. Mandic, L. Breger, and J. P. How, "Analysis of Decentralized Estimation Filters for Formation Flying Spacecraft," presented at the *AIAA GNC Conf.*, Aug. 2004 (AIAA-2004-5135).
- [25] D. Kelbel, T. Lee, A. Long, J. Russell Carpenter and C. Gramling, "Relative Navigation Algorithms for Phase 1 of the MMS Formation," *Flight Mechanics Symposium*, 25-30 Oct. 2003, Greenbelt, MD, USA.

- [26] A. Gelb, Applied Optimal Estimation, Massachusetts, MIT press, 1974.
- [27] <http://ssl.mit.edu/spheres/>, last accessed jan. 2005.
- [28] J.A. Leitner, G.E. Piper and J.M. Watkins “On the Impact of Cross-link Delays on Spacecraft Formation Control,” AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas, Aug. 11-14, 2003
- [29] Yu-Han Chen, “Effect of Communication Delays On the Performance of Vehicle Platoons,” MS thesis, Mechanical Eng., University of California Berkeley, May 1995.
- [30] Alvar Saenz-Otero “Design Principles for the Development of Space Technology Maturation Laboratories Aboard the International Space Station” Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Aero/Astro, June 2005
- [31] Serge Tournier “A reverse problem: Determination of the beacons positions on the SPHERES testbed”, MIT class project, 2004
- [32] Mark O. Hilstad, “A Multi-Vehicle Testbed and Interface Framework for the Development and Verification of Separated Spacecraft Control Algorithms.” Massachusetts Institute of Technology, Master Of Science Thesis, June 2002.
- [33] Mark O. Hilstad, John P. Enright, and Arthur G. Richards, “The SPHERES Guest Scientist Program,” <http://ssl.mit.edu/spheres/gsp/>, MIT, Oct. 2003.
- [34] J. Enright, M. Hilstad, A. Saenz-Otero, and D.W. Miller, “The SPHERES Guest Scientist Program: Collaborative Science On the ISS,” *IEEE Aerospace Conference*, Big Sky, Montana, March 2004.
- [35] <http://ssl.mit.edu/spheres/motivation.html>, last accessed may. 2006.

- [36] How, J. H., Deyst J., “Lecture Notes for Advanced Estimation for GPS and Inertial Navigation”,MIT Aerospace department graduate course, 16.324, Spring 2005.