2015

Censoring and Fusion in Non-linear Distributed Tracking Systems with Application to 2D Radar

Armond S. Conte II
conteas@vcu.edu

Censoring and Fusion in Non-linear Distributed Tracking Systems with Application to 2D Radar

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University

by

Armond Sylvester Conte II
B.S., Virginia Commonwealth University, 2008

Director: Ruixin Niu, PhD,
Assistant Professor, Department of Electrical and Computer Engineering

Virginia Commonwealth University
Richmond, VA
November, 2015

Acknowledgement

I would first like to thank my advisor Dr. Niu for helping me through the long hours of building revisions, derivations, and assisting me through the unique challenges that were presented to me as a part time graduate school student. I would like to thank the members of my thesis committee including Dr. Alen Docef and Dr. QiQi Lu, as well as my two coworkers/non-voting members, John Gray and Dr. Michael Rudzinsky, for taking time out of their schedule to support my defense and for providing me with helpful feedback that ultimately put this work at a level above and beyond its original format. I would like to thank my coworkers Said Sehbaoui and Bill Thompson for understanding when I required flexibility to pursue this degree while working full-time. Last but not least, I would like to thank my mother and father who, among many other gifts, were the first to introduce me to the fields of mathematics and signal processing.

Table of Contents

List of Tables

List of Figures

Abstract

The objective of this research is to study various methods for censoring state estimate updates generated from radar measurements. The generated 2-D radar data are sent to a fusion center using the J-Divergence metric as the means to assess the quality of the data. Three different distributed sensor network architectures are considered which include different levels of feedback. The Extended Kalman Filter (EKF) and the Gaussian Particle Filter (GPF) were used in order to test the censoring methods in scenarios which vary in their degrees of non-linearity. A derivation for the direct calculation of the J-Divergence using a particle filter is provided. Results show that state estimate updates can be censored using the J-Divergence as a metric controlled via feedback, with higher J-Divergence thresholds leading to a larger covariance at the fusion center.

Chapter I: Introduction

The success of the use of tracking methods in radar technology has created a niche in the field of statistical signal processing. Researchers desire to increase the accuracy of available tracks, so that their interpretation of otherwise noisy measurement data is more trustworthy and usable in a wider variety of applications. One method of increasing track accuracy is to fuse data from multiple sensors and take advantage of spatially distributed measurement data. Increasing the number of sensors has the prime disadvantage in that it also requires a greater communication rate between sensors and a fusion center. The goal of this thesis is to investigate and provide various methods that reduce the data-rate from sensors to a distributed fusion network that generates tracks, while maintaining certain level of accuracy.

Research in the field of distributed radar track filtering covers a wide distribution of topics. Some fusion research in the area of radar includes angle only tracking [1] [2] [3] and range only tracking [3] [4] [5]. The sensors in this thesis are radars that take measurements in bearing and range and use them to provide state estimates to a fusion center. This means that each sensor node maintains its own state estimates in Cartesian coordinate space, which transfers the data to a fusion center where the results are fused to form a more accurate estimate. Distributed fusion is an alternative to the purely centralized fusion, where the raw measurement data are sent to the fusion center via the sensors and all tracking and data processing is done there and the decentralized method where there is no fusion center at all. The specifics of the distributed fusion technique with a fusion center are described later in this thesis. Generally speaking distributed fusion is preferred when it is desired to have robustness against a single point of

failure or to gain a savings in communications. For more information on fusion techniques in general, refer to [6] and the tracking handbook [7].

Radar tracking is inherently a non-linear problem. While radar data are taken in polar coordinate systems for 2D Radar (spherical coordinates for 3D) the target trajectory is more easily modelled in Cartesian coordinate systems. In addition, the trajectory itself can also be non-linear. The Kalman filter is an obvious starting point, as its inherent objective is to perform an estimation on a hidden target state (the Cartesian position and velocity of the target) given a set of noisy measurements. However there are two major assumptions incurred by the Kalman filter that inhibit its direct implementation in certain scenarios. These are that:

(1) All statistics are Gaussian

(2) Both the state and measurement equations are either linear or have been linearized

Due to the computational complexity of optimal non-linear filtering, most non-linear methods that are available are suboptimal. Even more for highly non-linear cases, it must also account for non-Gaussianity to gain a better track accuracy. Two methods will be used in this thesis that fulfill the requirement for a non-linear estimator: either the Extended Kalman Filter (EKF) or the Particle Filter (PF). It is also possible to use non-linear fusion techniques in order to compensate for the bias when using linear fusion approaches. All of these techniques are described in more depth in the theory chapters. For more on the EKF and the Kalman Filter (KF), see [7] and [8]. Information on various particle filtering techniques can be found in [9].

The purpose of this thesis is to study a method for gaining communication savings between a set of distributed sensors and a fusion center. To do this, the quality of the data that are transmitted will be rated prior to making a decision on whether it should be censored. The

following papers describe methods to measure the quality of the measurement data before sending the information to a centralized fusion center [10] [11] [12]. The journal paper [10] tested state estimates for new information using the innovation metric. In addition [13] discusses the use of censoring data with the Kullback-Leibler (KL) divergence. The results in this thesis expand upon the techniques described in [10] and [13] using a J-Divergence metric for both the EKF and Gaussian Particle Filter (GPF) for 2D radar.

Novelties introduced as a result of this research include the following. First, the use of J-Divergence thresholding in limiting 2D radar state estimates. Second, the use of the fusion center state estimates as input to only the threshold calculation, later as model II. Finally, a derivation of the J-Divergence for comparing the prior and posterior local state probability density functions (pdfs) of particle filters.

The remainder of this thesis is structured as follows. Chapter II provides the linear Kalman filtering equations and initialization techniques. Chapter III gives an overview of the nonlinear filtering techniques. Chapters IV explains the theory behind the fusion architectures for both centralized and distributed fusion. Chapter V gives the theory behind the KL-Divergence and the J-Divergence and also provides the J-Divergence derivation of the particle filter. Chapter VI explains the censoring and distributed feedback models used in this thesis. Chapter VII describes the parameters and the scenarios used to test the feedback models. Chapter VIII provides the results of the tests. Chapter IX gives a discussion of the results and how they relate to the theory described. Finally Chapter X concludes and provides directions to potential future expansion on the work provided.

Chapter II: Kalman Filter

The Kalman Filter is an algorithm that makes estimates on the hidden state of a system given a set of measurements. It is required that the state-transition equation and the measurement equation be linear, and that all of the noises are Gaussian. This chapter describes the equations used in the Kalman Filter, which is the optimal method to perform linear filtering.

2.1. State and Measurement Equations

The state equation is given as in (1), where $\mathbf{x}_k$ is the state vector, $\mathbf{F}_k$ is the state transition matrix, and $\mathbf{v}_k \sim \mathrm{N}(0, \mathbf{Q}_k)$. k is an independent variable, which in many cases is simply time. Note that this can be expanded to include a control input.

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k \tag{1}$$

The measurement equation is given as in (2) where $\mathbf{z}_k$ is the measurement vector, $\mathbf{H}_k$ is the state transition matrix, and $\mathbf{w}_k \sim \mathrm{N}(0, \mathbf{R}_k)$.

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k \tag{2}$$

2.2. Kalman Filter Prediction

Taking the expected value on both sides of (1) and (2) gives the predicted state and predicted measurements in (3) and (4) respectively.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \tag{3}$$

$$\hat{\mathbf{z}}_{k|k-1} = \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \tag{4}$$

The covariances of (1) and (2) are given as (5) and (6) respectively, where $\mathbf{S}_k$ is known as the covariance of the innovation.

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1} {\mathbf{F}_k}^{\mathrm{T}} + \mathbf{Q}_k \tag{5}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} {\mathbf{H}_k}^{\mathrm{T}} + \mathbf{R}_k \tag{6}$$

2.3. Kalman Filter Update

Under the condition of orthogonality shown in (7),

$$E\left((\mathbf{x}_k - \hat{\mathbf{x}}_k){\mathbf{z}_k}^{\mathbf{T}}\right) = \mathbf{0} \tag{7}$$

a gain factor, known as the Kalman gain can be derived such that when a new measurement comes in, the state estimate can be updated so that it is more accurate and the covariance can be reduced accordingly. This is done using the innovation, which is given as $\boldsymbol{\nu}_k$ in (8) and whose covariance was given in (6).

$$\boldsymbol{\nu}_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1} \tag{8}$$

Taking the ratio between the cross-covariance between the state and measurement given in (9), and the covariance of the innovation in (6) gives the Kalman gain in (10).

$$\mathbf{P}_{k|k-1} {\mathbf{H}_k}^{\mathbf{T}} \tag{9}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} {\mathbf{H}^{\mathbf{T}}}_k (\mathbf{H}_k \mathbf{P}_{k|k-1} {\mathbf{H}^{\mathbf{T}}}_k + \mathbf{R}_k)^{-1} \tag{10}$$

The Kalman filter gain can then be used to update the predicted state estimate and covariance acting as a weight on the innovation for both the state in (11) and the covariance in (12).

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \boldsymbol{\nu}_k \tag{11}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k {\mathbf{K}_k}^{\mathbf{T}} \tag{12}$$

A simple interpretation of the Kalman filter gain is that when the true measurement error is high relative to the predicted measurement covariance, we will trust the prediction more and the observed innovation less, the opposite of which is true when the measurement error is low relative to the predicted measurement covariance.

By performing the prediction equations (3) – (6) and then when a new measurement comes in using equations (8) – (12), the Kalman filter can be sequentially updated to provide a better estimate of the state given the initial assumptions.

2.4. Initialization

In order to perform Kalman Filtering, the state estimate must be initialized. There are several different ways to do this, but one of the more practical methods is known as two-point differencing. If the measurement conversion is linear in a case where the state $\mathbf{x}$ is a two dimensional vector in 1D Cartesian coordinate space, where the first element $x_{p,0}$ is the initial state position and the second element $x_{v,0}$ is the initial state velocity and z_{-1} and z_0 are the first two consecutive measurements on the position, then two-point differencing can be performed using equations (13) – (15) as adapted from [8].

$$\hat{x}_{p,0} = z_0 \tag{13}$$

$$\hat{x}_{v,0} = \frac{z_0 - z_{-1}}{T} \tag{14}$$

The covariance matrix of $\hat{\mathbf{x}}_{0|0}$ is calculated directly as shown in (15), as $\mathbf{P}_{0|0}$, where R is the measurement variance and T is the time difference between the measurements z_0 and z_{-1}.

$$\mathbf{P}_{0|0} = \begin{bmatrix} \mathrm{R} & \frac{\mathrm{R}}{\mathrm{T}} \\ \frac{\mathrm{R}}{\mathrm{T}} & \frac{2\mathrm{R}}{\mathrm{T}^2} \end{bmatrix} \tag{15}$$

Two-point differencing is the basis for the more complex initialization that is performed for the case of non-linear filtering as shown later in Section 3.4. There are also other ways available to perform initialization, even with a single measurement point, for more information see [7] and [8].

Chapter III: Nonlinear Filtering

The reason that nonlinear filters are required for 2D radar, instead of the Kalman filter, is because of the non-linear conversion in coordinate systems between how the target motion is modelled (Cartesian) and how the measurements are taken (polar). This chapter describes two non-linear techniques, the Extended Kalman Filter (EKF) and the Particle Filter (PF) that can be used to perform tracking in 2D radar, along with the non-linear initialization technique that is used in this thesis.

3.1. Extended Kalman Filter

The need for the EKF arises when either the measurement and/or the state transition equations are non-linear, but the statistics are still Gaussian as shown in equations (16) and (17).

$$\mathbf{x}_k = f(\mathbf{x}_{k-1,}\mathbf{v}_t) \tag{16}$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{w}_k) \tag{17}$$

The problem with using a simple Kalman filter for this is that the state transition and measurement matrices are required in order to calculate the predicted covariance and the updated state/covariance. In the EKF a 1st order Taylor series expansion can be used to perform this approximation. To do this the Jacobian of the state and measurement matrices is taken and used in their place. In other words redefine $\mathbf{F}_k$ and $\mathbf{H}_k$ as shown in (18) and (19) respectively, where N_x is size of the state vector and N_z is the size of the measurement vector. In (18) $\mathbf{F}_k$ is evaluated at $\hat{\mathbf{x}}_{k-1|k-1}$ and has the size of N_x x N_x while in (19) $\mathbf{H}_k$ is evaluated at $\hat{\mathbf{x}}_{k|k-1}$ and has the size of N_z x N_x.

$$\mathbf{F}_k = \begin{bmatrix} \frac{\partial f_{1,k}}{\partial x_{1,k}} & \cdots & \frac{\partial f_{1,k}}{\partial x_{N_x,k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{N_x,k}}{\partial x_{1,k}} & \cdots & \frac{\partial f_{N_x,k}}{\partial x_{N_x,k}} \end{bmatrix} \tag{18}$$

$$\mathbf{H}_k = \begin{bmatrix} \frac{\partial h_{1,k}}{\partial x_{1,k}} & \cdots & \frac{\partial h_{1,k}}{\partial x_{N_x,k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{N_z,k}}{\partial x_{1,k}} & \cdots & \frac{\partial h_{N_z,k}}{\partial x_{N_x,k}} \end{bmatrix} \tag{19}$$

Using these equations can create a linearization error and it is important to note that the approximated matrices in (18) and (19) should not be used in the state transition and measurement equations (1) and (2), but rather (16) and (17) should be used directly when possible. Eq. (18) and (19) allow for Eq. (9) – (12) to be calculated when $\mathbf{F}_k$ and $\mathbf{H}_k$ are not otherwise available.

Depending on the degree of noise and non-linearity of the equations present in the system, the linearization errors in the EKF may become an issue. One way to account for this in the EKF, in the case of both 2-D and 3-D radars, is shown in [14]. The choice solution for the purposes of this research is to use the particle filter which eliminates the Gaussian assumption at the cost of computational complexity, and is described as a Markov Chain Monte-Carlo type of technique.

3.2. Particle Filter

Particle filters have the same purpose as a Kalman Filter of making an estimate on the hidden state of a system that produces noisy measurements. It does this by modelling a

continuous probability distribution as a set of discrete weighted points known as particles that are randomly drawn from a so called proposal distribution. The resulting set of particles can take any shape and with updates from measurements and enough particles, the particles and their weights will produce a more accurate discrete representation of the continuous distribution. A simple variation of the sequential importance sampling (SIS) particle filter algorithm, from which other particle filters are derived from follows below. More in depth information on particle filters can be found in [9].

(1) Initialize particles and weights, drawing a set of particles from an importance density given as q, where the number of particles is typically based upon the complexity of the problem as well as the state dimension. This proposal density can be a Gaussian distribution with mean and covariance obtained using the two-point differencing approach described in (13) – (15)

(2) When a measurement comes in, propagate each particle using the prediction equation in (16) to generate the prior discrete distribution.

(3) Update each weight w_k^i, where i is the particle index and N_p is the number of particles, using equation (20) below and normalize the weights such that their sum is equal to one. Note that if q is set as in (21), then we simply need to multiply the prior weights by the likelihood to perform an update.

$$\mathrm{w}_k^i = \mathrm{w}_{k-1}^i \frac{p\left(\mathbf{z}_k \middle| \mathbf{x}_k^i\right) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}, \qquad i = 1, \dots, N_p \tag{20}$$

$$q\left(\mathbf{x}_k^i \middle| \mathbf{x}_{k-1}^i, \mathbf{z}_k\right) = p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i) \tag{21}$$

(4) Repeat steps 2 and 3 each time a new measurement is available

From this, we can obtain the first two moments of the posterior state pdf using equations (22) and (23) below, where N_p is the total number of particles, $\boldsymbol{\mu}_k$ is the estimated mean and $\mathbf{P}_k$ is the estimated covariance.

$$\boldsymbol{\mu}_k = \sum_{i=1}^{N_p} \mathrm{w}_k^i \mathbf{x}_k^i \tag{22}$$

$$\mathbf{P}_k = \sum_{i=1}^{N_p} \mathrm{w}_k^i (\mathbf{x}_k^i - \boldsymbol{\mu}_k)(\mathbf{x}_k^i - \boldsymbol{\mu}_k)^T \tag{23}$$

3.2.1. Particle Degeneracy

The SIS algorithm is in many applications not practical because of particle degeneracy. Since a finite number of particles are being used in order to describe the shape of a probability distribution, this means that the more particles there are, the more intricacies of the shape of the posterior pdf can be captured. Unfortunately, more particles also imply a greater computational complexity. Due to computing power limitations, particle filter are limited in the total number of particles that can be used. This issue is examined in much greater detail in publications that involve algorithms to improve the particle filter to counter the curse of dimensionality [15] [16]. Unfortunately even with a large number of particles, a case called particle degeneracy will occur where as more updates are provided, the spread of the distribution is likely to tighten over time to the point that all weights except for one are approximated as zero. The next section discusses a few variations of the particle filter that address the issue of particle degeneracy.

3.2.2. Variations

There are a few methods of getting by the problem of particle degeneracy, including the bootstrap filter and the Gaussian particle-filter. The bootstrap filter involves deleting low weighted particles and then resampling from the currently known distribution. Resampling can create a drastic increase in the computational complexity of the particle filter and also results in a change in the distribution, but it ensures that particles with low weights are not taken into account. Unfortunately, this can also create a different, but similar effect of the SIS filter where all of the particles occupy a single point and the shape of the distribution is lost. While there are ways to correct this using techniques such as the regularized particle filter, the added computations from the resampling step make the bootstrap filter unattractive for many practical purposes.

The Gaussian particle filter [17] reduces the approximation of the posterior pdf propagated between updates and predictions to a Gaussian (or other known) distribution. While this adds a redraw step to the simple SIS algorithm, this eliminates the need to resample particles because all of the particles are drawn from a Gaussian distribution at each iteration of the filter. The disadvantage is that the shape of the prior pdf is limited to Gaussian. Since the Gaussian particle filter does not assume that the posterior pdf of the distribution must be linear and/or Gaussian, this allows for the particle filter to cover both the nonlinearity in both the trajectory and the radar measurements.

3.3. Applicability of Non-linear Filtering to 2-D Radar Tracking

In 2-D radar the primary source of non-linearity is that the measurements are taken in polar coordinates, while the state estimates are given in Cartesian coordinates. If bearing and range are measured, then assuming a state vector as shown in (24) the measurement matrix is given by (25).

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \tag{24}$$

$$\mathbf{H}_k = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2}} & 0 & \frac{y}{\sqrt{x^2+y^2}} & 0 \\ -\frac{y}{x^2+y^2} & 0 & \frac{x}{x^2+y^2} & 0 \end{bmatrix}\Bigg|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}} \tag{25}$$

The performance of the tracking filters and their associated fusion will vary based upon the type of radar used. Two parameters that directly affect how accurately targets will be able to be tracked include the measurement accuracy and update rate. In the literature, [18] describes some generic tracking frame times for various types of targets that can be tracked by a multi-function array radar. A condensed version of the table with the applicable frame times is provided below in Table 1.

Radar Function (distance)	Frame Time (seconds)
Long-range search (200km)	10
Medium-range search (60km)	6
Horizon search (40km)	3
Tracking (Air Traffic Control) (80km)	2
Tracking (Hostile) (80km)	0.5
Guidance (80km)	0.1

Table 1: List of Nominal Frame Times for a Multi-Function Phased Array Radar

Frame time indicates the amount of time dedicated to revisiting an individual target. The difference between search and tracking is that in tracking, additional time is dedicated to the target's location. For the purposes of this research, the 2 second frame time from Table 1 was used to simulate a realistic air traffic control scenario.

3.4. Initialization for Non-linear Filtering

The Kalman filtering initialization technique described in Chapter 2.4 cannot be always be performed when the state/measurement equations are non-linear. This thesis uses a Converted Measurement Kalman Filter (CMKF) to perform the two point differencing portion for both the EKF and PF. The methods used to create the initialization using the CMKF for initialization of a 2D radar filter are described below as adapted from [19].

(1) Measurements are taken for both range (R_{-1} and R_0) and bearing (θ_{-1} and θ_0) at two consecutive times (k_{-1} and k_0).

(2) The state estimate is then given by (26)-(29) where the noises are assumed to be Gaussian distributed and t_λ and t_λ' are respectively given in (30) and (31) and L_x and L_y respectively are the x and y positions of a stationary sensor.

$$\hat{p}_x = \frac{R_0\, cos(\theta_0) + L_x}{t_\lambda} \tag{26}$$

$$\hat{v}_x = \frac{p_x - (R_{-1}\cos(\theta_{-1}) + L_x)}{t_\lambda(k_0 - k_{-1})} \tag{27}$$

$$\hat{p}_y = \frac{R_0\sin(\theta_0) + L_x}{t_\lambda} \tag{28}$$

$$\hat{v}_y = \frac{p_y - (R_{-1}\sin(\theta_{-1}) + L_y)}{t_\lambda(k_0 - k_{-1})} \tag{29}$$

$$t_\lambda = e^{-2\sigma_\theta^2} \tag{30}$$

$$t_\lambda' = {t_\lambda}^2 \tag{31}$$

If the corresponding covariance matrix $\mathbf{P}$, calculated via 2-point differencing is given by (32), taken by calculating the variance elements directly, then the standard deviation elements, $\sigma_{xx,k}$, $\sigma_{yy,k}$, and $\sigma_{xy,k}$ are given by (33), (34), or (35) as shown in [19].

$$\mathbf{P} = \begin{bmatrix} \sigma_{xx,0}^2 & \frac{\sigma_{xx,0}^2}{\mathrm{T}} & \sigma_{xy,0}^2 & \frac{\sigma_{xy,0}^2}{\mathrm{T}} \\ \frac{\sigma_{xx,0}^2}{\mathrm{T}} & \frac{\sigma_{xx,0}^2 + \sigma_{xx,-1}^2}{\mathrm{T}^2} & \frac{\sigma_{xy,0}^2}{\mathrm{T}} & \frac{\sigma_{xy,0}^2 + \sigma_{xy,-1}^2}{\mathrm{T}^2} \\ \sigma_{xy,0}^2 & \frac{\sigma_{xy,0}^2}{\mathrm{T}} & \sigma_{yy,0}^2 & \frac{\sigma_{yy,0}^2}{\mathrm{T}} \\ \frac{\sigma_{xy,0}^2}{\mathrm{T}} & \frac{\sigma_{xy,0}^2 + \sigma_{xy,-1}^2}{\mathrm{T}^2} & \frac{\sigma_{yy,0}^2}{\mathrm{T}} & \frac{\sigma_{yy,0}^2 + \sigma_{yy,-1}^2}{\mathrm{T}^2} \end{bmatrix} \tag{32}$$

$$\sigma_{xx,k} = \sqrt{R_k^2\sigma_\theta^2 sin^2(\theta_k) + \sigma_R^2 cos^2(\theta_k)} \tag{33}$$

$$\sigma_{yy,k} = \sqrt{R_k^2\sigma_\theta^2 cos^2(\theta_k) + \sigma_R^2 sin^2(\theta_k)} \tag{34}$$

$$\sigma_{xy,k} = \sqrt{(\sigma_R^2 - R_k^2 sin^2(\theta_k))\sin(\theta_k)\cos(\theta_k)} \tag{35}$$

If the same initialization procedure is not performed in every scenario, then the problem may be biased due to a too accurate, or inaccurate initial value for the filter. This method is an unbiased and repeatable procedure for selecting the filter value, and is performed in each scenario for both the EKF and the GPF.

Chapter IV: Fusion Theory

It is desired to take measurement data from multiple sensors in order to obtain a better estimate of the target state by taking advantage of diversity of the sensors. Some general references that build the basic theory behind data fusion, specifically for tracking, include [6] , [7] , and [20]. If the sensors provide different information, then these pieces of information can be combined to generate a more overall encompassing view of the state estimate. Spatial diversity for the case of 2D Radar in particular increases the observability of the target state, when the measurements and their associated noises are taken in range and azimuth. Centralized and distributed fusion are two different architectures that are analyzed in the following sections.

4.1. Centralized

In centralized fusion, sensors provide their raw measurements directly to a fusion center. The fusion center then combines all of the measurements and generates a state estimate. According to [6] in a linear setting, this is the optimal fusion method in terms of accuracy. Since the sensors are required to transmit all of their data to the fusion center, and then the fusion center is required to combine all of the estimates, this can create much larger communication costs than other methods that are available. See Figure 1 for an overview of how centralized fusion works.

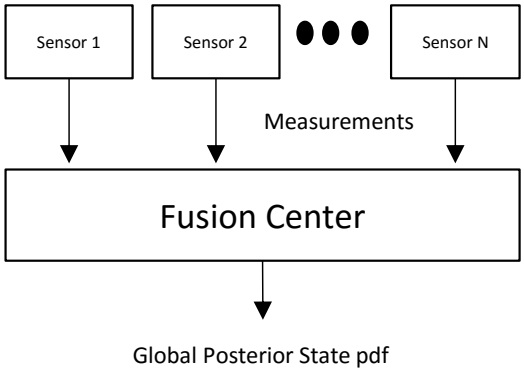

Figure 1: Centralized Fusion Architecture

4.2. Distributed

Another type of fusion is called distributed. In distributed fusion, the local sensors generate their own state estimates based upon the local measurements. Only the state estimates are then sent to the fusion center, unlike in the centralized case. This creates an advantage in distributed fusion over centralized fusion, because the state-estimates do not need to be transmitted by the local sensors at every sampling interval. On the other-hand, if the state estimates are not transmitted at each interval, distributed fusion is sub-optimal in comparison with centralized fusion. See Figure 2 for the basic data flow structure of the distributed fusion architecture.

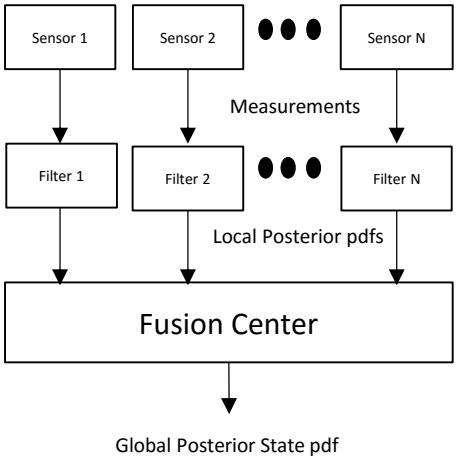

Figure 2: Distributed Fusion Architecture

4.3. Fusion Center

The purpose of the fusion center is to combine the local estimates into a more accurate global estimate. A method known as feedback, where the fusion center provides its estimate to the local sensors for use in the filter update, can be used to perform even more accurate filtering with a fusion center. At a general level there is a linear and a nonlinear method available to perform filtering, as described in [6]. Fusion changes both of these methods slightly because the additional prior information from the local sensors must be subtracted out. The linear fusion equations without feedback are described below for the information matrix (inverse of the covariance matrix) as (36) and the state fusion equation, which must be multiplied by the global covariance in (37). Note that in equations (36) – (41), i is the sensor index, N is the total number of sensors, and the subscripted variables indicate local while the non-subscripted indicate global.

$$\mathbf{P}_{k|k}^{-1} = \mathbf{P}_{k|k-1}^{-1} + \sum_{i=1}^{N} \left[\mathbf{P}_{i,k|k}^{-1} - \mathbf{P}_{i,k|k-1}^{-1}\right] \tag{36}$$

$$\mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} = \mathbf{P}_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1} + \sum_{i=1}^{N}\left[\mathbf{P}_{i,k|k}^{-1}\hat{\mathbf{x}}_{i,k|k} - \mathbf{P}_{i,k|k-1}^{-1}\hat{\mathbf{x}}_{i,k|k-1}\right] \tag{37}$$

With feedback, the global prior is included in each of the updated state estimates, so it must be subtracted out $N-1$ times in order to ensure that it is not double counted.

$$\mathbf{P}_{k|k}^{-1} = \sum_{i=1}^{N}\left[\mathbf{P}_{i,k|k}^{-1} - (N-1)\mathbf{P}_{i,k|k-1}^{-1}\right] \tag{38}$$

$$\mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^{N}\left[\mathbf{P}_{i,k|k}^{-1}\hat{\mathbf{x}}_{i,k|k} - (N-1)\mathbf{P}_{i,k|k-1}^{-1}\hat{\mathbf{x}}_{i,k|k-1}\right] \tag{39}$$

The nonlinear fusion equations are given by (40) and (41), for no-feedback and feedback respectively. In equations (40) and (41), the constant C is equal to the integral of the other terms on the right hand side of the equation and effectively is known as a normalizing constant and $\mathbf{Z}_k$indicates the total set of measurements leading up to time k.

$$p[\mathbf{x}_k|\mathbf{Z}_k] = C^{-1}[\mathbf{Z}_k]\left\{\prod_{i=1}^{N}\frac{p\left[\mathbf{x}_k|\mathbf{Z}_{i,k}\right]}{p\left[\mathbf{x}_k|\mathbf{Z}_{i,k-1}\right]}\right\}p[\mathbf{x}_k|\mathbf{Z}_{k-1}] \tag{40}$$

$$p[\mathbf{x}_k|\mathbf{Z}_k] = C^{-1}[\mathbf{Z}_k]\left\{\prod_{i=1}^{N}p\left[\mathbf{x}_k|\mathbf{Z}_{i,k}\right]\right\}\{p[\mathbf{x}_k|\mathbf{Z}_{k-1}]\}^{-(N-1)} \tag{41}$$

Chapter V: Comparing Probability Distributions

Another aspect which is touched on in this thesis is the concept of utilizing the distance between two different probability distributions as the method of quantifying the amount of new information between the two.

5.1. Kullback-Leibler Distance

One type of metric is known as the Kullback-Leibler (KL) Distance [21]. The general purpose of the KL Distance is to provide two different probability distributions with the same support a non-symmetric distance between these two distributions. By non-symmetric, it is meant that if the two probability distributions are input into the KL Distance equation in two different orders, a different metric will be obtained. The general formula for the KL distance is given as (42).

$$KL(p(x)||q(x)) = \int_{-\infty}^{\infty} p(x) log \frac{p(x)}{q(x)} dx \tag{42}$$

5.2. J-Divergence

J-Divergence, also known as the Jensen-Shannon Divergence, transforms the Kullback-Leibler Distance into a symmetric metric. In the literature, [21] describes the J-Divergence as a quantity proportional to the power 2 mean of the two versions of the KL distances, see equation (43), but some sources simply describe it as an arithmetic mean [22] (see (44)), or simply a sum [23]. The general purpose is to avoid the non-symmetric biasing from the KL Distance. Since each serves approximately the same function, the proper J-Divergence can be selected based upon

the problem at hand, with the necessary calculations taken into account. The general formula for J-Divergence as the power mean is (43) and as the arithmetic mean is (44).

$$J(p(\mathbf{x})||q(\mathbf{x})) = \sqrt{KL(p(\mathbf{x})||q(\mathbf{x}))^2 + KL(q(\mathbf{x})||p(\mathbf{x}))^2} \tag{43}$$

$$J(p(\mathbf{x})||q(\mathbf{x})) = \frac{KL(p(\mathbf{x})||q(\mathbf{x}))}{2} + \frac{KL(q(\mathbf{x})||p(\mathbf{x}))}{2} \tag{44}$$

When $p(\mathbf{x})$ and $q(\mathbf{x})$ are Gaussian distributed, with mean described as $\bar{\mathbf{x}}_{p(\mathbf{x})}$and $\bar{\mathbf{x}}_{p(\mathbf{x})}$, the covariance described as $\mathbf{P}_{p(\mathbf{x})}$ and $\mathbf{P}_{q(\mathbf{x})}$, d denoting the dimensionality of the state vector $\mathbf{x}$, tr denoting the trace function, and $\ln$ denoting the natural logarithm, a closed form for the KL distance is given in [24] as (45).

$$\begin{aligned} &KL(p(\mathbf{x})||q(\mathbf{x})) \\ &\quad = \frac{1}{2}\, tr\left(\mathbf{P}_{q(\mathbf{x})}{}^{-1}\mathbf{P}_{p(\mathbf{x})}\right) \\ &\quad - \frac{1}{2}\ln\frac{\left|\mathbf{P}_{p(\mathbf{x})}\right|}{\left|\mathbf{P}_{q(\mathbf{x})}\right|} - \frac{d}{2} + \frac{1}{2}(\bar{\mathbf{x}}_{q(\mathbf{x})} - \bar{\mathbf{x}}_{p(\mathbf{x})})^T\mathbf{P}_{p(\mathbf{x})}^{-1}(\bar{\mathbf{x}}_{q(\mathbf{x})} - \bar{\mathbf{x}}_{p(\mathbf{x})}) \end{aligned} \tag{45}$$

In this thesis, (45) is used in (43) for the simulations described in Chapter VII.

5.3. Particle Filter J-Divergence Calculation

Assuming $KL(p(x)||q(x))$ and $KL(q(x)||p(x))$ are two versions of the Kullback-Leibler divergence as defined by (42) and the variation of the J-Divergence in (44) is adopted. The J-Divergence can be expanded by inputting the definition in (42) into (44), and then rearranging to get (46) and then (47).

$$J(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{2}\left[\int p(x)[\log p(\mathbf{x}) - \log q(\mathbf{x})]dx + \int q(x)[\log q(\mathbf{x}) - \log p(\mathbf{x})]dx\right] \tag{46}$$

$$J(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{2}\left[\int [p(\mathbf{x}) - q(\mathbf{x})][\log p(\mathbf{x}) - \log q(\mathbf{x})]dx\right] \tag{47}$$

Assume that $p(\mathbf{x})$ and $q(\mathbf{x})$ are respectively the posterior and prior probability distribution functions. In a particle filter, $p(\mathbf{x})$ and $q(\mathbf{x})$ are approximated as shown in (48) and (49), where $\mathbf{x}^i$ is a particle with weight w^i, i is the particle index and $N_p$is the total number of particles.

$$p(\mathbf{x}) \approx \sum_{i=1}^{N_p} w_{new}^i \delta(\mathbf{x} - \mathbf{x}^i) \tag{48}$$

$$q(\mathbf{x}) \approx \sum_{i=1}^{N_p} w_{old}^i \delta(\mathbf{x} - \mathbf{x}^i) \tag{49}$$

Adding the approximations in (48) and (49) into the first part of (47), we get (50).

$$J(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{2}\left[\int \left[\left(\sum_{i=1}^{N_p} w_{new}^i \delta(\mathbf{x} - \mathbf{x}^i)\right) - \left(\sum_{i=1}^{N_p} w_{old}^i \delta(\mathbf{x} - \mathbf{x}^i)\right)\right]\left[\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\right]dx\right] \tag{50}$$

Using the sifting property of the delta function we get (51).

$$J(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{2}\left[\sum_{i=1}^{N_p}\left((w_{new}^i - w_{old}^i)\log\frac{p(\mathbf{x}^i)}{q(\mathbf{x}^i)}\right)\right] \tag{51}$$

Substituting the particle approximations of (48) and (49) for the second half of (51), we get (52)

$$J(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{2}\left[\sum_{i=1}^{N_p}\left(\left(w_{new}^{i} - w_{old}^{i}\right)\log\frac{\sum_{j=1}^{N_p} w_{new}^{j}\delta(\mathbf{x}^{i} - \mathbf{x}^{j})}{\sum_{j=1}^{N_p} w_{old}^{j}\delta(\mathbf{x}^{i} - \mathbf{x}^{j})}\right)\right] \tag{52}$$

Since the summation of $\delta(\mathbf{x}^{i} - \mathbf{x}^{j})$ only has support when $i = j$, the summation and delta functions can be removed from (52), giving a closed form for the J-divergence of a particle filter as shown in (53).

$$J(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{2}\sum_{i=1}^{N_p}\left(\left[w_{new}^{i} - w_{old}^{i}\right]\log\left(\frac{w_{new}^{i}}{w_{old}^{i}}\right)\right) \tag{53}$$

There is a problem in using this equation when the support changes between particle filter iterations. This occurs when the Gaussian particle filter or resampling is performed. Due to this issue, of the models shown in Chapter VI, only Model I allows for the particle filter J-Divergence to be used.

Chapter VI: Feedback Models

Three different feedback models have been used for this research. Feedback from the fusion center to the local sensors improves tracking performance, by giving the local state estimators a better state estimate as its prior.

6.1. Model I

In the first model there is no feedback performed. The obvious disadvantage is that the fusion center can easily diverge. This is likely to occur in cases where the target maintains the same system model over a long period of time. The advantage to this is that the transmitting cost is reduced because the fusion center does not need to transmit its state estimate back to the sensors.

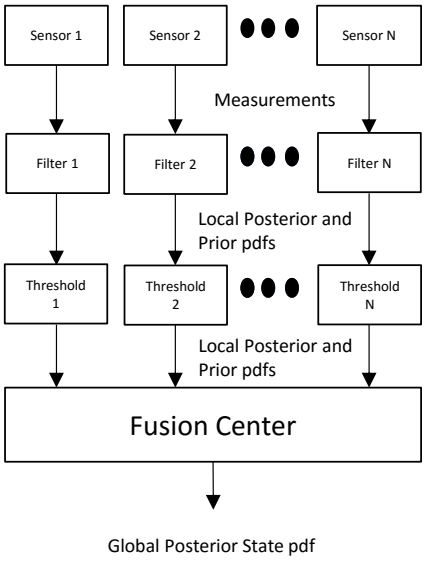

Figure 3: Model I - No Feedback

6.2. Model II

In Model II, the fusion center sends its global state estimate back to the local sensors. The sensors then, using their own prior state estimate generate a posterior state estimate and compares it against the global prior state estimate. Compared to Model I, this offers a degree of control to the fusion center, enabling the fusion center to "call" for a new measurement when its state estimate begins to diverge. Model II also enables the sensors to develop their own estimates which protects them from faults that may occur at the Fusion Center.

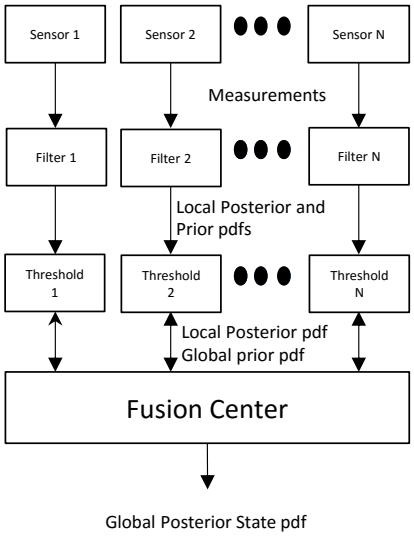

Figure 4: Model II - Feedback to Local Thresholds

6.3. Model III

Model III is similar to Model II in that the global state estimates from the fusion center are sent back to the local sensors for processing. The difference is that the sensors then replace their prior with that of the global prior. This enables a degree of control that is different and possibly greater from that seen in Model II. More specifically, the local posterior is generated from the global prior and the comparison is made between the new local posterior and the global prior.

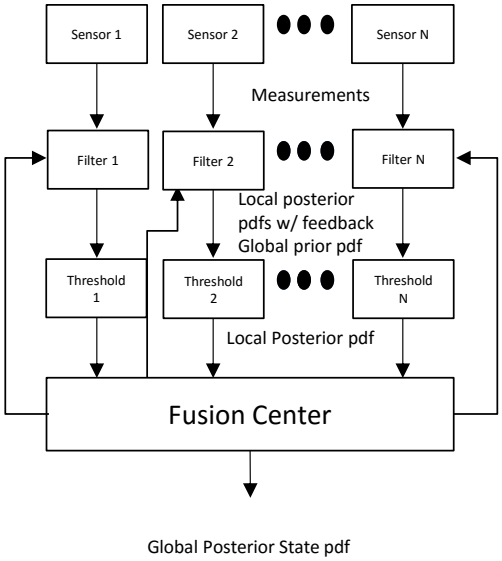

Figure 5: Model III - Feedback to Local State Estimators

6.4. Fusion Model Table

Table 2 provides a general overview of the inputs to the threshold and the local state estimate filter.

MODEL #	Threshold Input	Local Filter Input
I	Sensor prior pdf	Sensor prior pdf
II	Global prior pdf	Sensor prior pdf
III	Global prior pdf	Global prior pdf

Table 2: Overview of Model Inputs

Chapter VII: Experiment Settings

7.1. State and Measurement Equations

This subsection denotes the various models that are used for the scenarios performed. In 2-D Radar with a near-constant-velocity model the state vector is defined as (54) with its elements in Cartesian coordinates.

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \tag{54}$$

The measurement vector is defined in range and bearing as shown in (55), with elements in polar coordinates.

$$\boldsymbol{z} = \begin{bmatrix} R \\ \theta \end{bmatrix} \tag{55}$$

The state evolves linearly according to (56). Using a linear white noise acceleration motion model [8], the state-transition matrix is defined in (57), where T is defined as the difference in time between state estimate updates. $\mathbf{v}$ is the process noise with zero-mean and a covariance which is defined in (58), where $\tilde{q}$ is the power spectral density of the continuous process noise before its discretization over time. In equations (56) – (63), k is an independent variable denoting time and i is the sensor index.

$$\mathbf{x}_{k+1|k} = \mathbf{F}\mathbf{x}_{k|k} + \mathbf{v}_k \tag{56}$$

$$\mathbf{F} = \begin{bmatrix} 1 & \mathrm{T} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathrm{T} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{57}$$

$$\mathbf{Q} = \begin{bmatrix} \mathrm{T} & \frac{\mathrm{T}^2}{2} & 0 & 0 \\ \frac{\mathrm{T}^2}{2} & \mathrm{T} & 0 & 0 \\ 0 & 0 & \frac{\mathrm{T}^3}{3} & \frac{\mathrm{T}^2}{2} \\ 0 & 0 & \frac{\mathrm{T}^2}{2} & \mathrm{T} \end{bmatrix} \tag{58}$$

The measurements in 2-D Radar are nonlinear because of the transformation between Cartesian and polar coordinate systems. Assuming that the position of the target at time k is given as (x_k, y_k) and the position of radar i is given as (x_i, y_i). Then we define $x_{k,i}$ as in (59) and $y_{k,i}$ as in (60).

$$\mathrm{x_{k,i}} = \mathrm{x_k} - \mathrm{x_i} \tag{59}$$

$$\mathrm{y_{k,i}} = \mathrm{y_k} - \mathrm{y_i} \tag{60}$$

Similarly the measurement equation is defined in (70) as

$$\mathbf{z}_{\mathrm{k,i}} = \begin{bmatrix} \sqrt{\mathrm{x_{k,i}^2} + \mathrm{y_{k,i}^2}} \\ \mathrm{atan}\left(\frac{\mathrm{y_{k,i}}}{\mathrm{x_{k,i}}}\right) \end{bmatrix} + \mathbf{w}_{\mathrm{k,i}} \tag{61}$$

where $\mathbf{w}_{k,i}$ is the measurement noise with a covariance defined as $\mathbf{R}$ in (71).

$$\mathbf{R} = \begin{bmatrix} \sigma_{\mathrm{R}}^2 & 0 \\ 0 & \sigma_{\theta}^2 \end{bmatrix} \tag{62}$$

While this is not needed in the Gaussian particle filter, the EKF requires a first order approximation of the measurement matrix $\mathbf{H}$, which is the Jacobian of the matrix shown in equation (61). This is given as equation (63) and is evaluated using (59) and (60).

$$\mathbf{H}_{k,i} = \begin{bmatrix} \frac{x_{k,i}}{\sqrt{x_{k,i}^2 + y_{k,i}^2}} & 0 & \frac{y_{k,i}}{\sqrt{x_{k,i}^2 + y_{k,i}^2}} & 0 \\ -\frac{y_{k,i}}{x_{k,i}^2 + y_{k,i}^2} & 0 & \frac{x_{k,i}}{x_{k,i}^2 + y_{k,i}^2} & 0 \end{bmatrix} \tag{63}$$

7.2. Test Scenario

7.2.1. Assumptions

For each scenario, the following assumptions are made:

(1) Measurements are taken in Range and Bearing, where zero degrees is aligned with the positive x-axis as opposed to the y-axis. Thus in a more real-life scenario the x-axis will correspond to North and the negative y-axis corresponds to East.

(2) The sensors are stationary.

(3) The target is fully detectable throughout the run and measurements are taken iteratively and are processed in the order in which they were taken.

(4) A linear state transition function is used in both the EKF and the PF and $\tilde{q}$ is set to 2 for all runs.

(5) The fusion center can use either the general linear combination or a particle filter implementation.

7.2.2. Geometry

A target with a linear trajectory moves through the center area of a set of three spatially diverse sensors, see Fig. 6.

Sensor #	Pos. (x,y) (km)
1	(0,10)
2	(10,25)
3	(25,10)

Table 3: Sensor Locations

The target follows a near constant velocity model with a set start position and velocity. For the purposes of this simulation this is given as follows.

Tar. Param	Value
x – start	5 km
y – start	20 km
x – velocity	300 m/s
y - velocity	-100 m/s

Table 4: Target Initial State

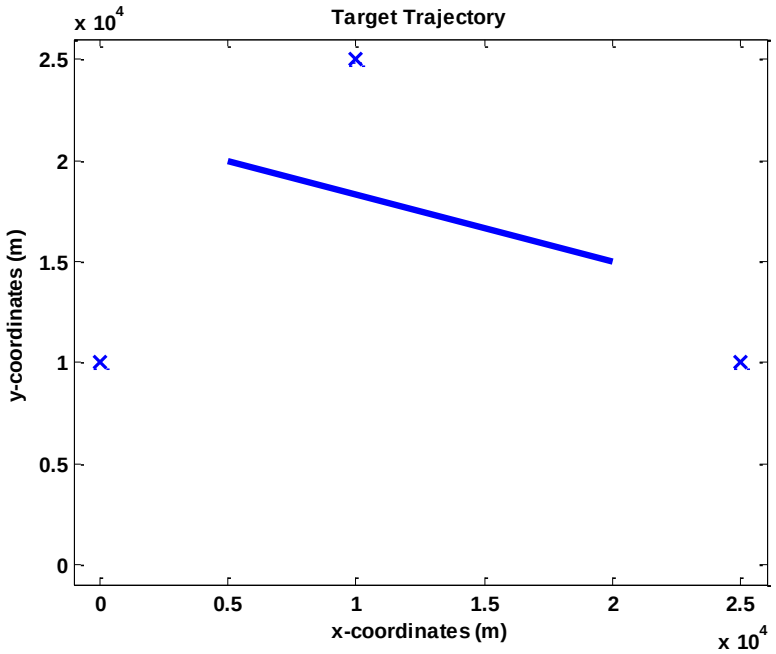

Figure 6: Target Trajectory

7.2.3. Test Descriptions

7.2.3.1. Test 1

Since the EKF has the ability to run at a much faster rate than the GPF, we are not as computationally limited and can run problems with higher sampling rates over a longer period of time. A scenario was performed to illustrate running the censoring scenarios with a large number of data points, by using a 100Hz sampling rate (this is unreasonably high, see Table 1). The remainder of the scenarios utilize a sampling rate of 10Hz. The test scenario for this case was run over the course of 50 seconds. The sensor standard deviation in range is 100m and the standard deviation in angle is 1 degree. This is to simulate weak non-linearity in the measurement distribution as seen in [25].

7.2.3.2. Test 2

In the second test, the sampling rate is reduced to 0.5 Hz, which is nominal for air traffic control operations in radar, see Table 1. Both the EKF and GPF algorithms are used as the GPF is no longer as computationally constrained. In terms of fusion, the linear equations are used in the EKF and GPF for models 1, 2, and 3. The sensor accuracies all remain the same. This run allows for a comparison of the EKF and the GPF under more linear conditions than Test 3 and enables the GPF to actually be run unlike in Test 1.

7.2.3.3. Test 3

In the third test, all of the parameters are the same as the second test, with the exception that the accuracies in the sensor are changed to that of the strong non-linearity described in [25], in other words the range standard deviation is decreased to 10m and the bearing standard deviation is increased to 3 degrees. The purpose of this test is to further test the capabilities of the GPF, as the EKF is expected to fail due to the high degree of non-linearity.

7.2.3.4. Test Matrix

Parameter	**Test 1**	**Test 2**	**Test 3**
Range std	100 m	100 m	10 m
Bearing std	1 deg	1deg	3 deg
Sampling Rate	100Hz	0.5 Hz	0.5Hz
EKF Used?	Yes	Yes	Yes
GPF Used	No	Yes	Yes

Table 5: Test Matrix

Chapter VIII: Results

Results are provided by the Root Mean Squared Error (RMSE) curves, covariance consistency plots, and also a chart with the percentage of unused updates. The covariance consistency plots were calculated using the Normalized Estimation Error Squared (NEES) between the actual target state and the state/covariance as described in [8]. The average NEES over a total of 100 Monte-Carlo runs is given in (64), where $\tilde{\mathbf{x}}$ is the difference between the state $\mathbf{x}_k$ and the updated state estimate $\hat{\mathbf{x}}_{k|k}$ and ϵ_k is the value of the NEES, all at time k. i indicates the index of the Monte-Carlo run. In ideal cases (with a linear-Gaussian assumption), the NEES are chi-squared distributed. This gives the 95% confidence interval, with 100 Monte-Carlo runs, and 4 degree of freedom in the state estimate n_x as [3.46, 4.57]. In the plots, the 95% confidence interval is represented by two horizontal red lines.

$$\bar{\epsilon_k} = \frac{1}{100}\sum_{\mathrm{i}=1}^{100} \tilde{\mathbf{x}}_{\mathrm{i,k|k}}^{\mathbf{T}} \mathbf{P}_{\mathrm{i,k|k}}^{-\mathbf{1}} \tilde{\mathbf{x}}_{\mathrm{i,k|k}} \tag{64}$$

There is no readily available equation that can map the J-Divergence to a level of unused estimates, so the J-Divergence thresholds, denoted as ξ, were empirically determined based upon trial and error. While the two-way communication parameter was not calculated, it can be inferred that Model I has additional communication savings due to the fusion center not being required to send estimates back to the local sensors.

Also for Test 3 not all of the plots could be generated due to quick divergence in covariance of the particle filters, leading to run-time errors due to the need to draw values from a random number generator. This is part of the nature of the highly non-linear model.

8.1. Test 1:

J-Divergence I/II/III	Percentage of Unused Updates		
	Model I	Model II	Model III
0/0/0	0	0	0
0.01/2/0.01	65.05	6.83	64.83
0.02/4/0.02	89.55	78.13	80.82
0.03/6/0.03	93.54	89.75	86.39
0.04/8/0.04	95.2	94.84	89.32
0.05/10/0.05	96.17	96.14	91.15

Table 6: Test 1 J-Divergence Unused Updates Chart

8.1.1. Model I

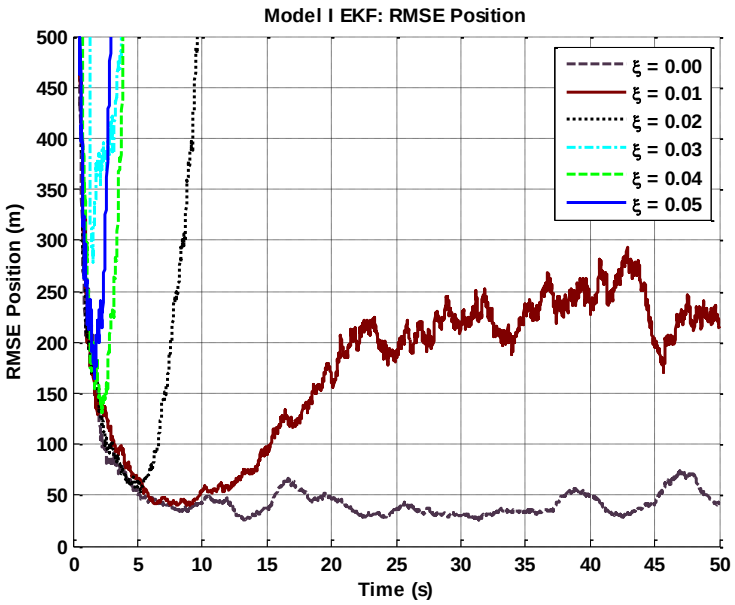

Figure 7: RMSE Position Test 1 Model I

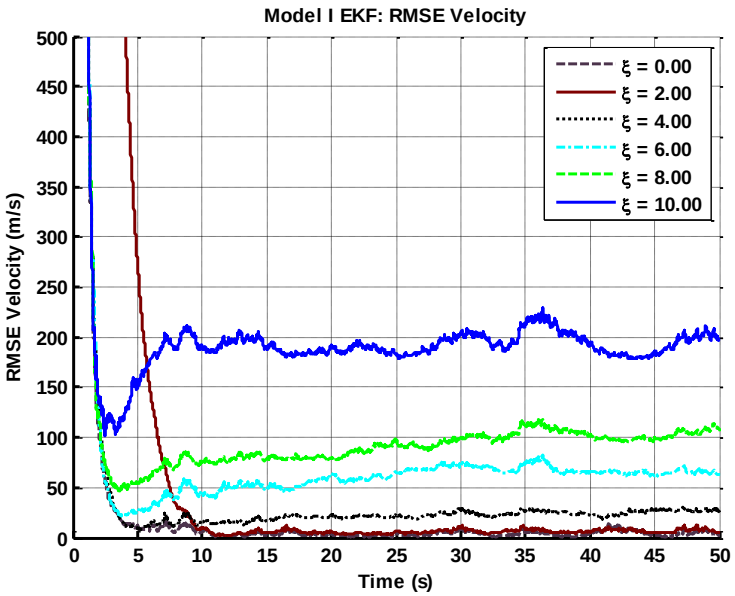

Figure 8: RMSE Velocity Test 1 Model I

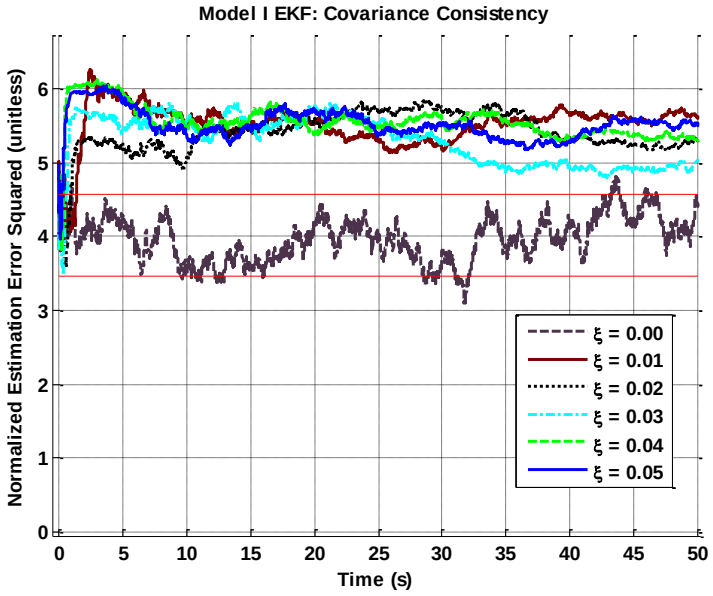

Figure 9: Covariance Consistency Test 1 Model I

8.1.2. Model II

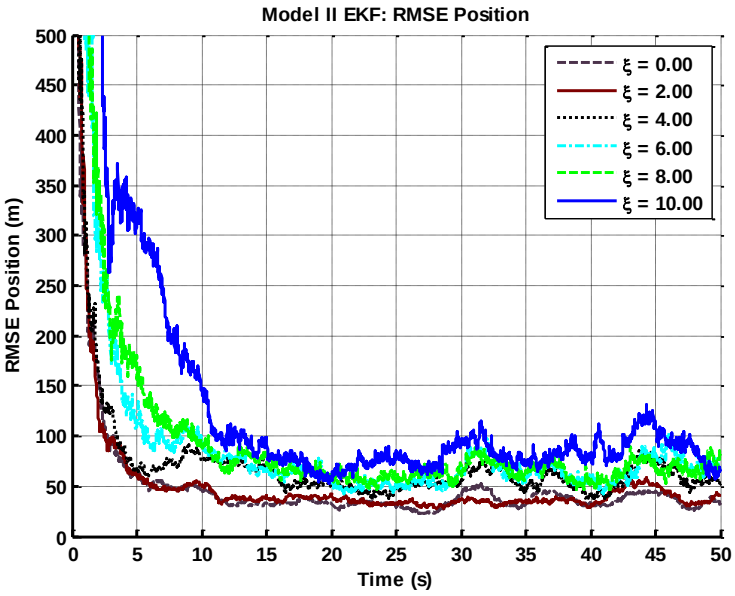

Figure 10: RMSE Position Test 1 Model II

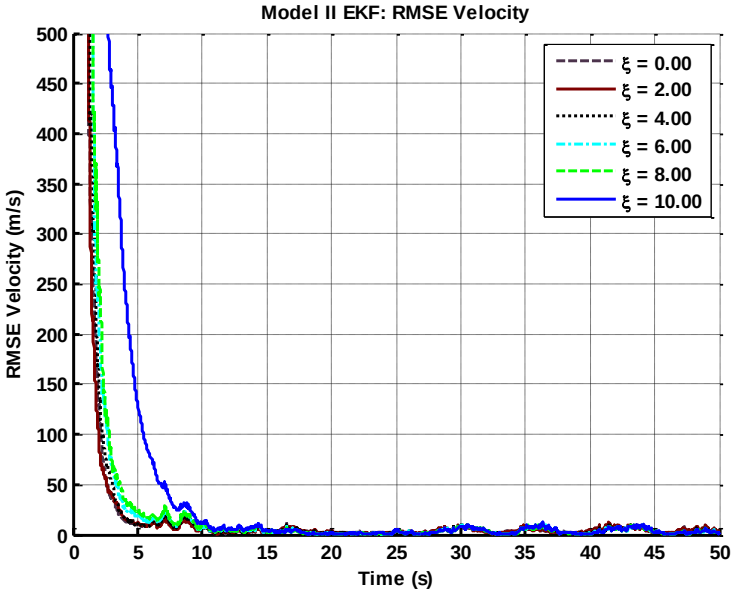

Figure 11: RMSE Velocity Test 1 Model II

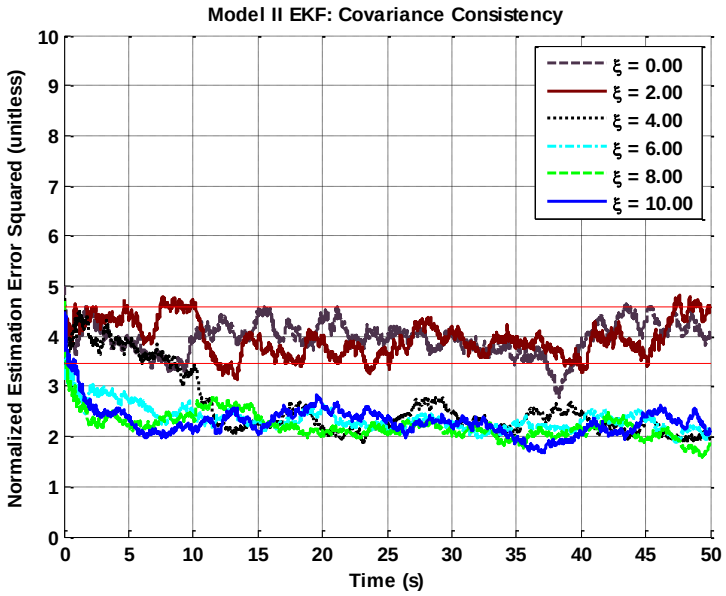

Figure 12: Covariance Consistency Test 1 Model II

8.1.3. Model III

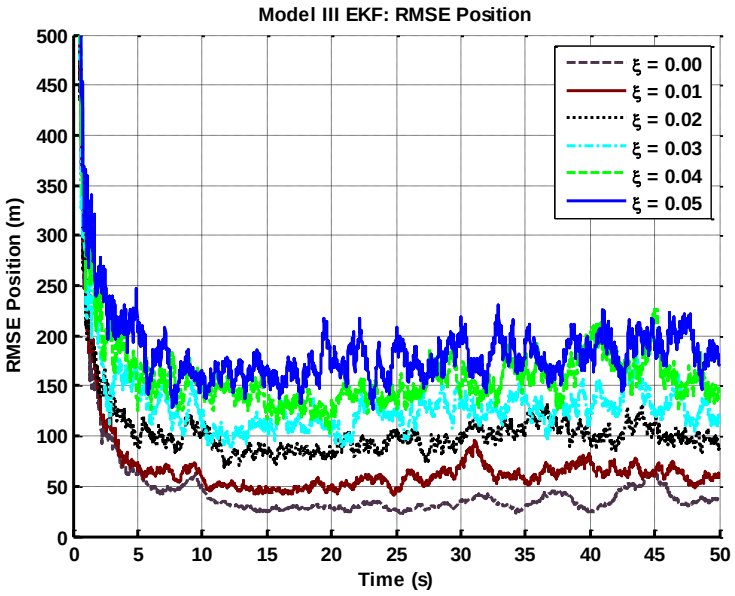

Figure 13: RMSE Position Test 1 Model III

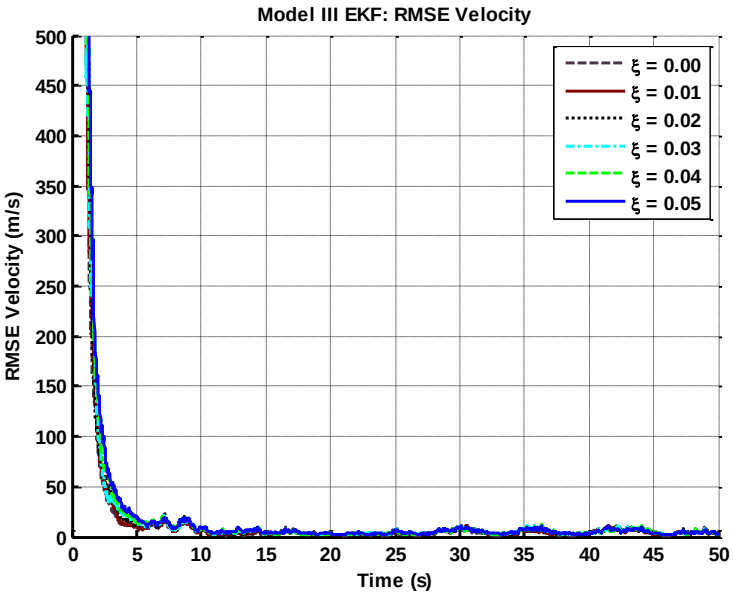

Figure 14: RMSE Velocity Test 1 Model III

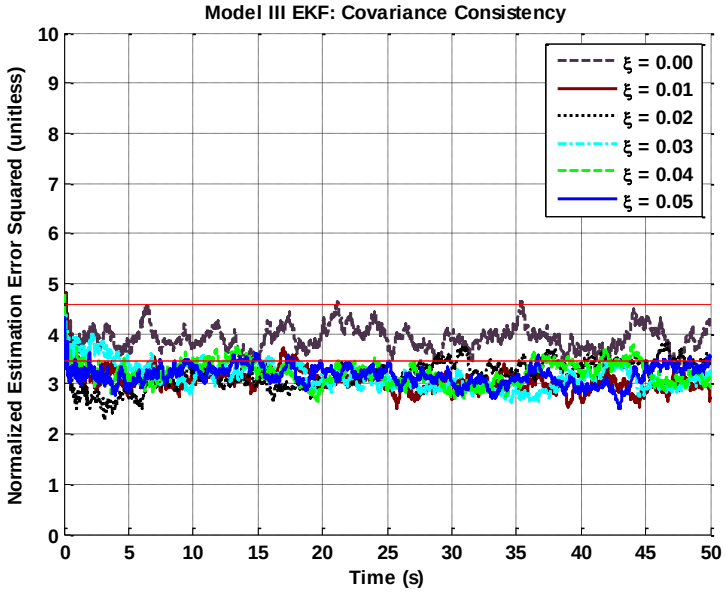

Figure 15: Covariance Consistency Test 1 Model III

8.2. Test 2

8.2.1. Extended Kalman filter

J-Divergence I/II/III	Percentage of Unused Updates		
	Model I	Model II	Model III
0/0/0	0	0	0
2/10/0.5	11.29	63.53	41.88
4/20/1.0	60.97	74.68	64.28
6/30/1.5	75.95	81.79	73.44
8/40/2.0	82.78	85.72	77.44
10/50/2.5	87.42	88.42	80.33

Table 7: Test 2 EKF J-Divergence Unused Updates Chart

8.2.1.1. Model I

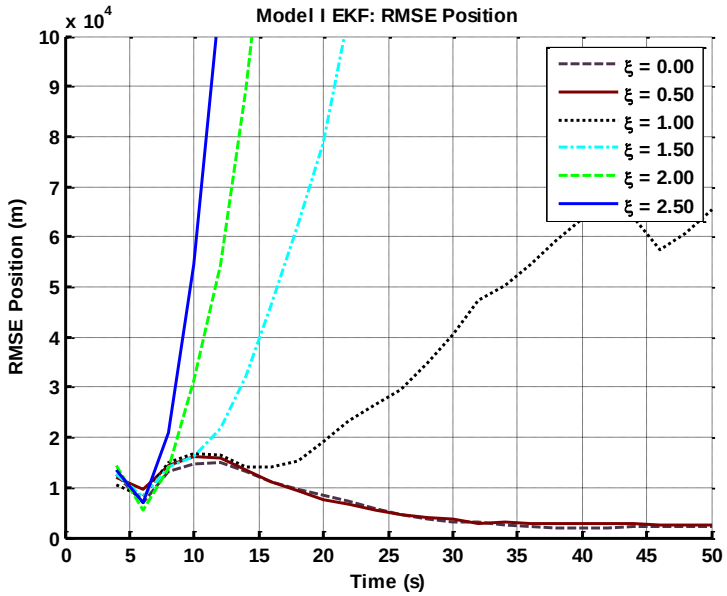

Figure 16: RMSE Position Test 2 EKF Model I

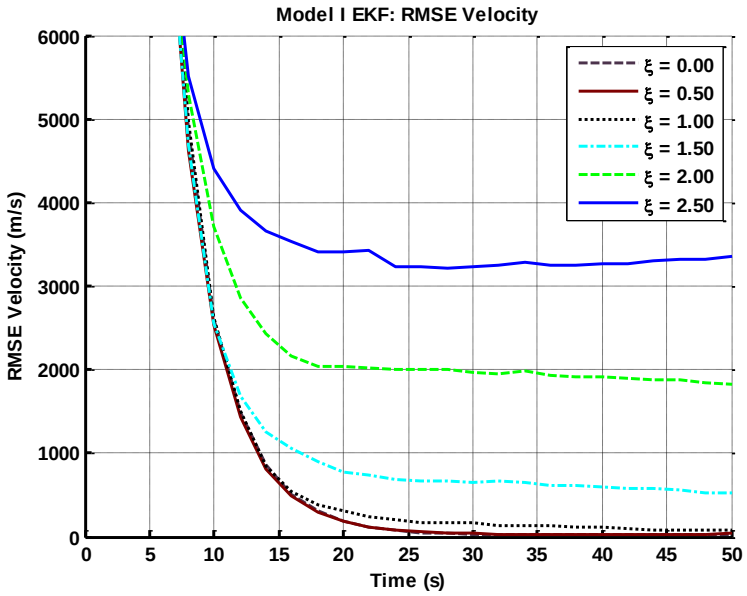

Figure 17: RMSE Velocity Test 2 EKF Model I

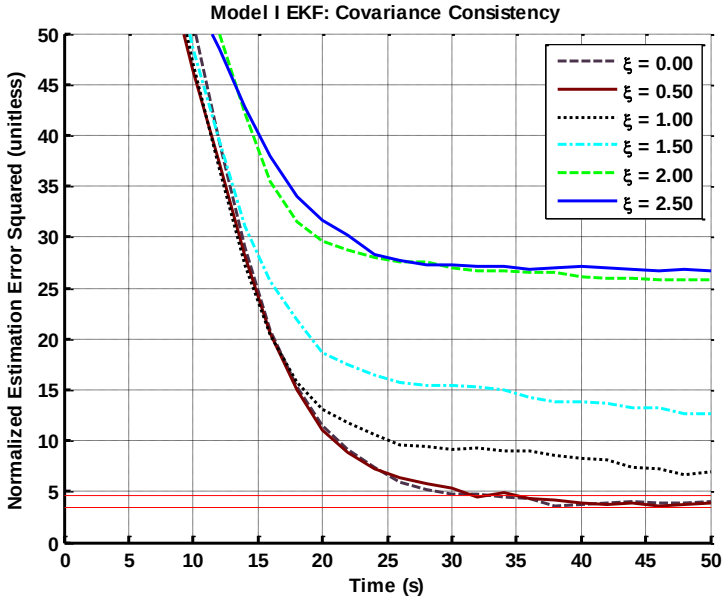

Figure 18: Covariance Consistency Test 2 EKF Model I

8.2.1.2. Model II

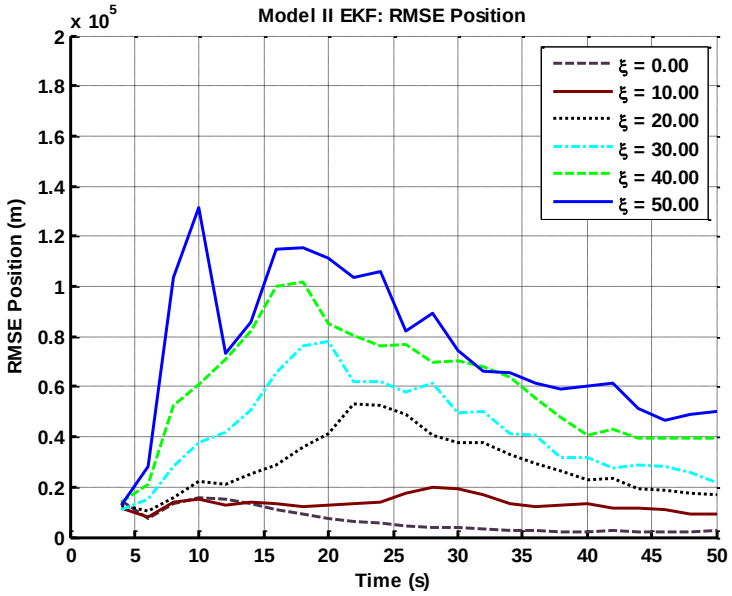

Figure 19: RMSE Position Test 2 EKF Model II

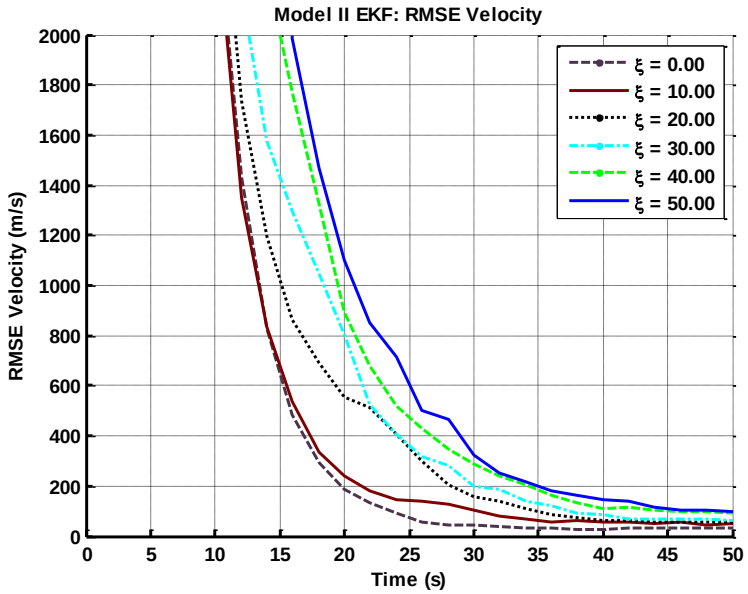

Figure 20: RMSE Velocity Test 2 EKF Model II

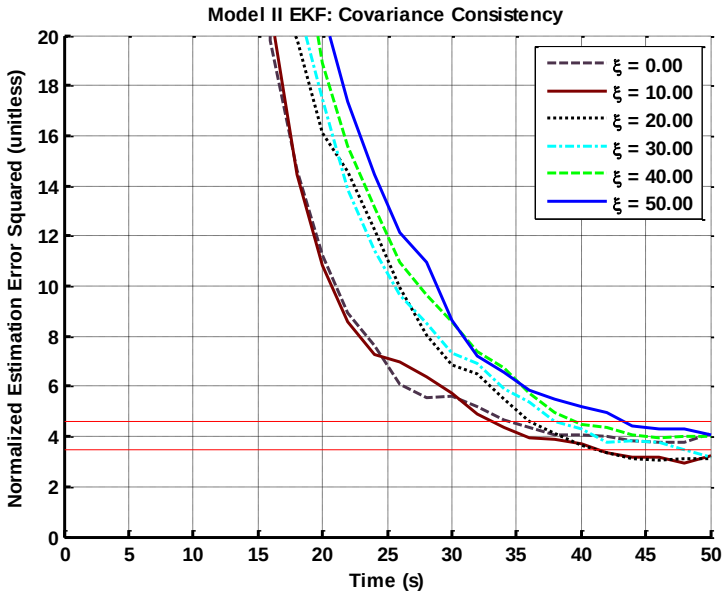

Figure 21: Covariance Consistency Test 2 EKF Model II

8.2.1.3. Model III

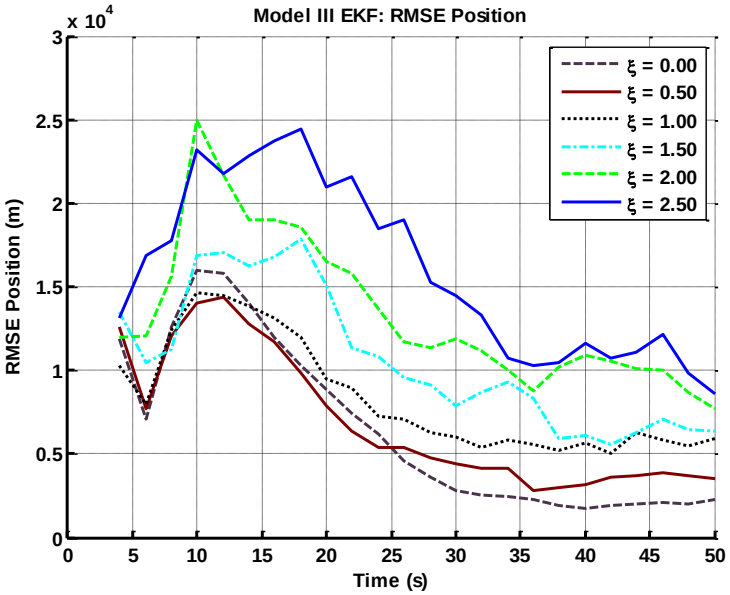

Figure 22: RMSE Position EKF Test 2 Model III

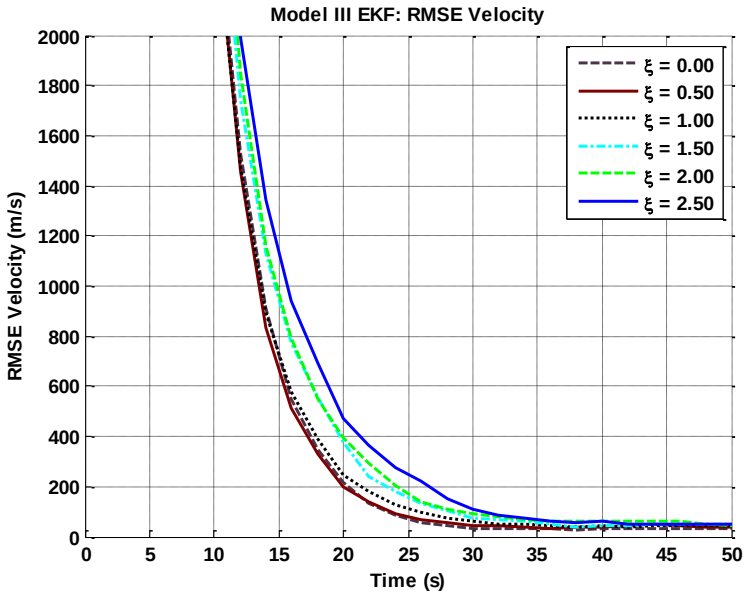

Figure 23: RMSE Velocity EKF Test 2 Model III

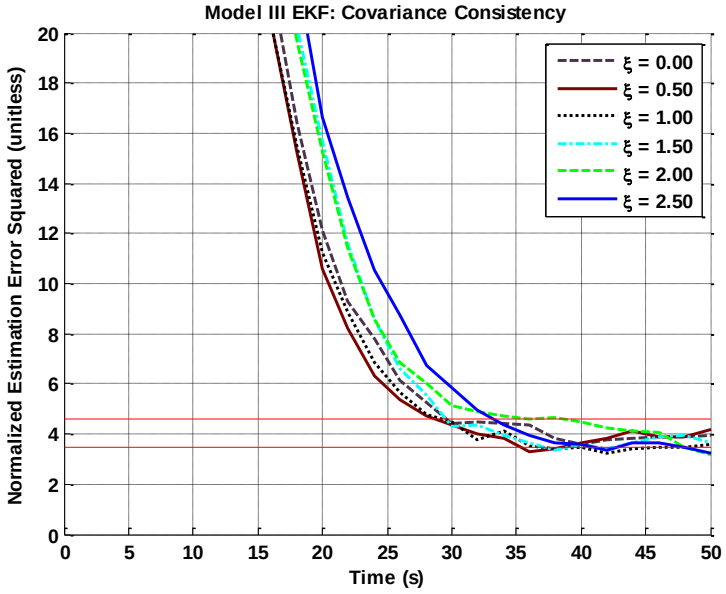

Figure 24: Covariance Consistency EKF Test 2 Model III

8.2.2. Gaussian Particle Filter

J-Divergence I/II/III	Percentage of Unused Updates		
	Model I	Model II	Model III
0/0/0	0	0	0

1/10/0.2	60.06	47.58	0.29
2/20/0.4	83.46	63.17	32.68
3/30/0.6	89.60	70.03	48.88
4/40/0.8	93.125	76.01	57.79
5/50/1.0	94.5	78.94	64.04

Table 8: Test 2 GPF J-Divergence Unused Updates Chart

8.2.2.1. Model I

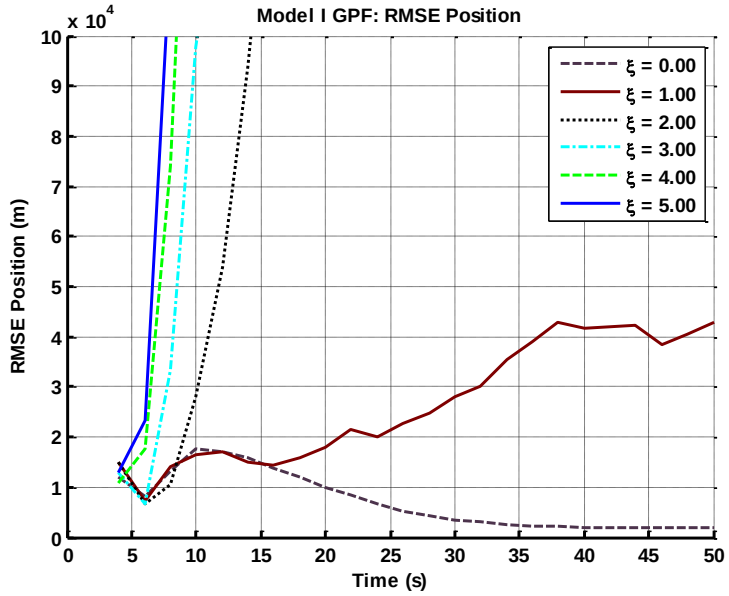

Figure 25: RMSE Position GPF Test 2 Model I

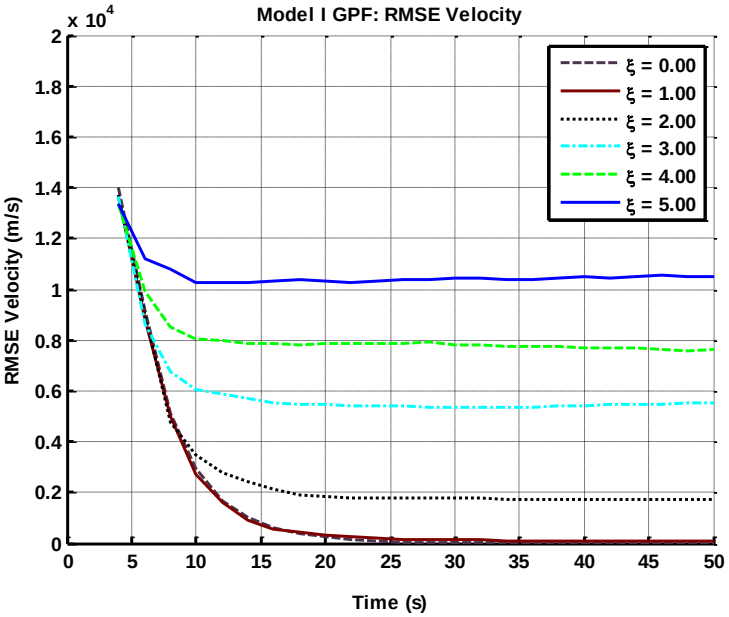

Figure 26: RMSE Velocity GPF Test 2 Model I

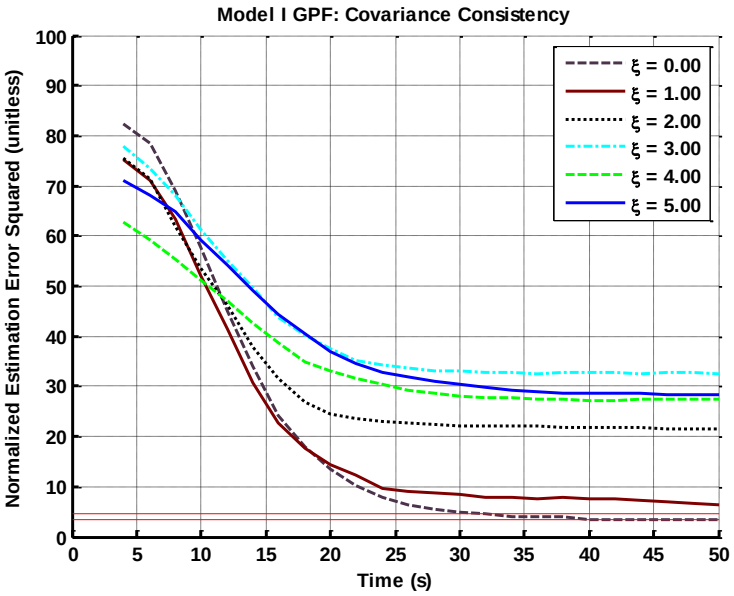

Figure 27: Covariance Consistency GPF Test 2 Model I

8.2.2.2. Model II

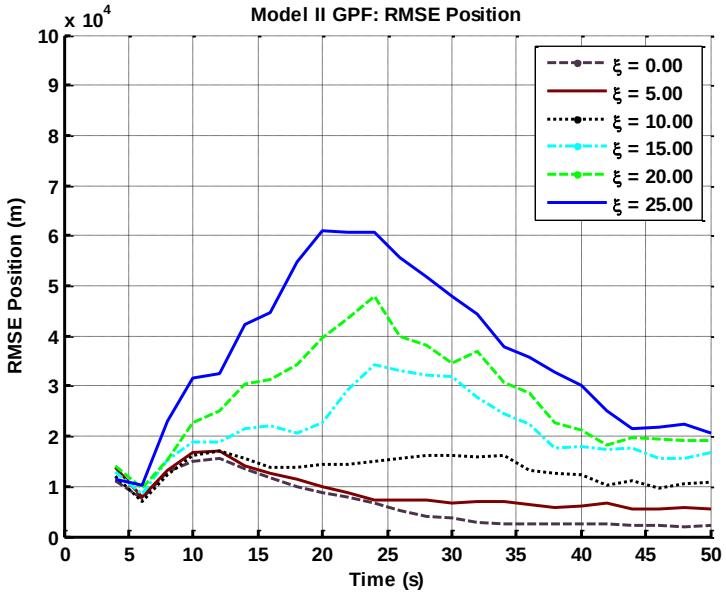

Figure 28: RMSE Position GPF Test 2 Model II

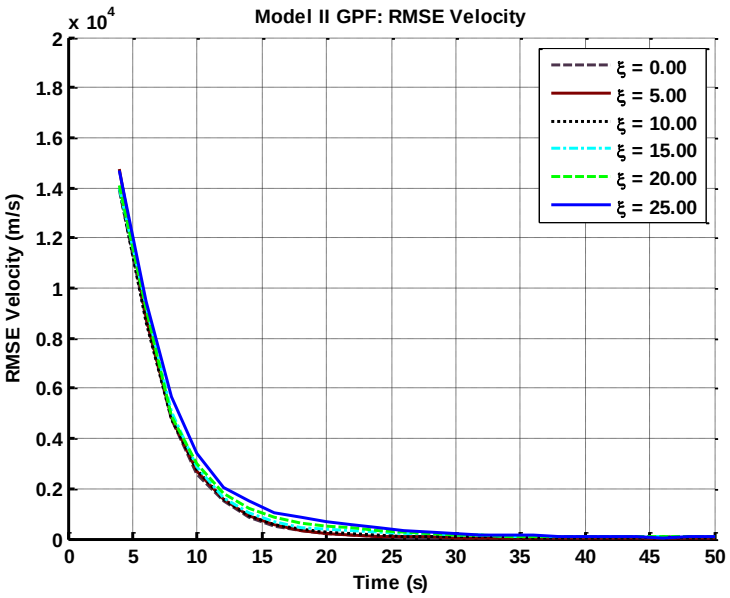

Figure 29: RMSE Velocity GPF Test 2 Model II

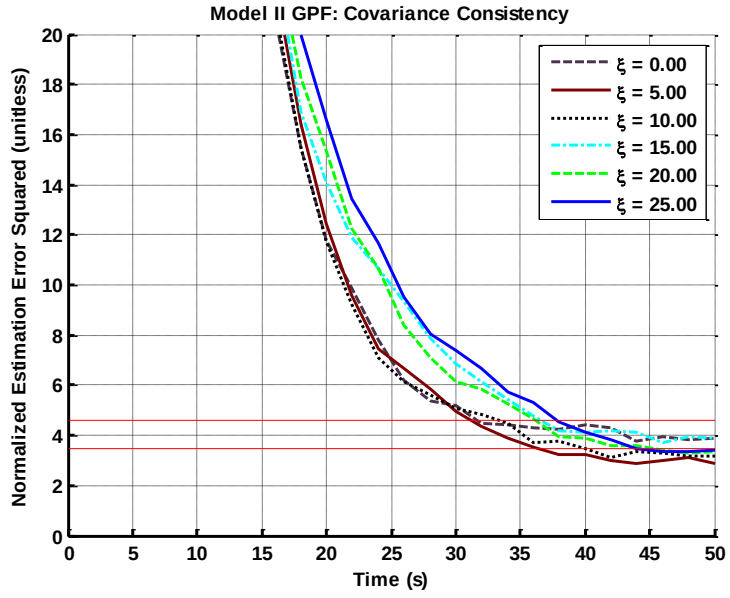

Figure 30: Covariance Consistency GPF Test 2 Model II

8.2.2.3. Model III

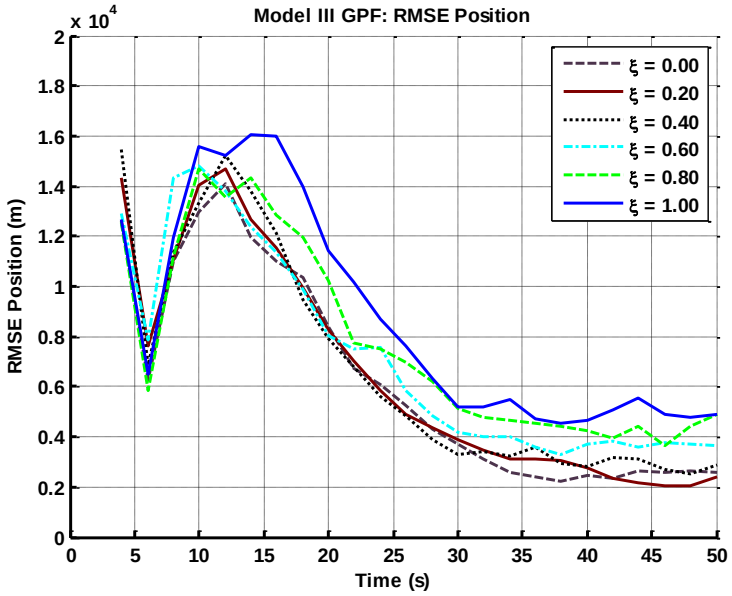

Figure 31: RMSE Position GPF Test 2 Model III

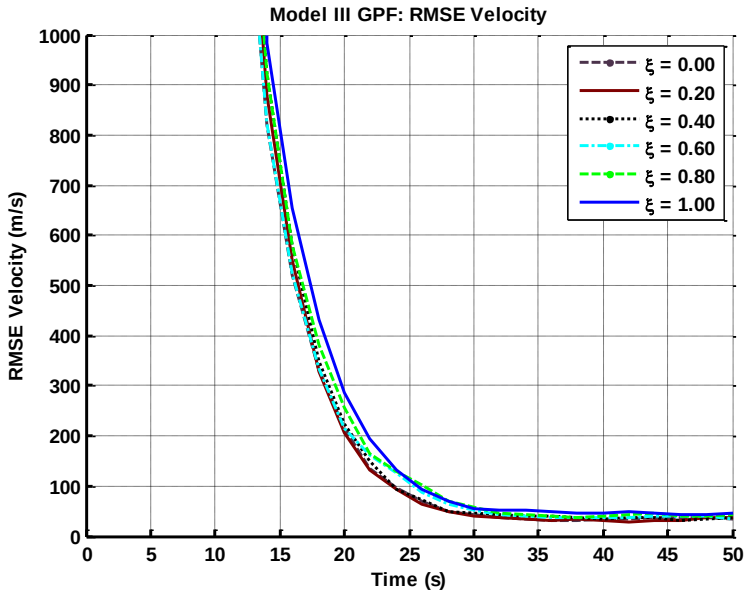

Figure 32: RMSE Velocity GPF Test 2 Model III

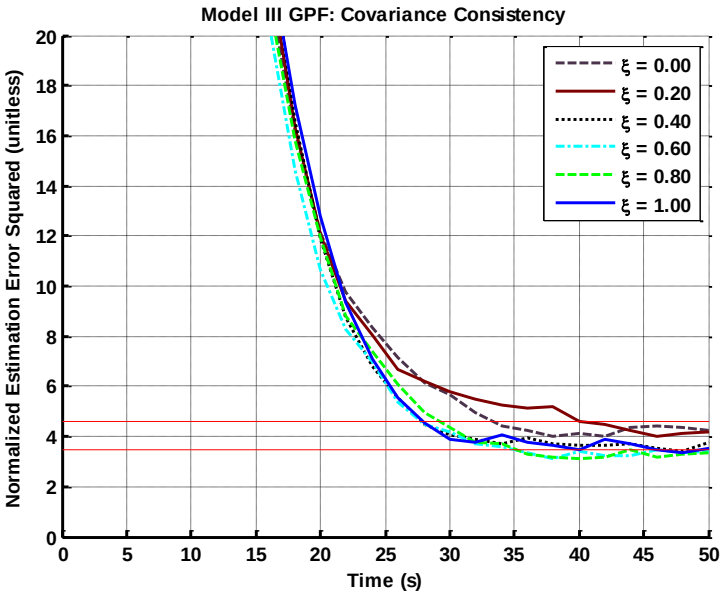

Figure 33: Covariance Consistency GPF Test 2 Model III

Test 3

8.2.3. Extended Kalman filter

J-Divergence I/II/III	Percentage of Unused Updates		
	Model I	Model II	Model III
0/0/0	0	0	0
2/500/2	52.96	39.17	44.36
4/1000/4	73.99	55.08	61.00
6/1500/6	78.19	64.5	66.25
8/2000/8	83.01	66.13	69.54
10/2500/10	84.75	70.37	72.13

Table 9: Test 3 EKF J-Divergence Unused Updates Chart

8.2.3.1. Model I

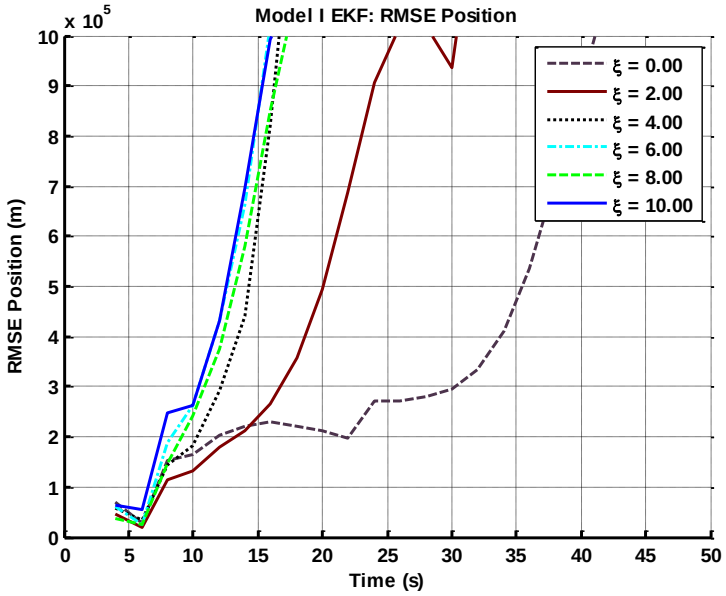

Figure 34: RMSE Position EKF Test 3 Model I

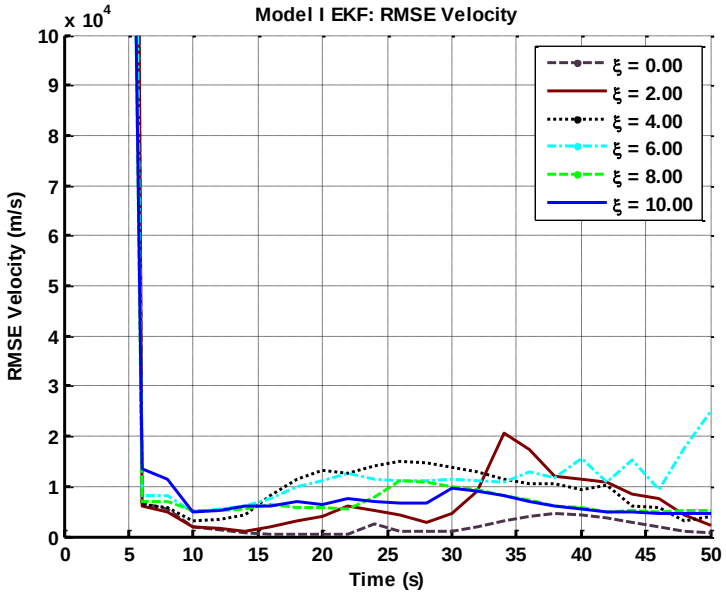

Figure 35: RMSE Velocity EKF Test 3 Model I

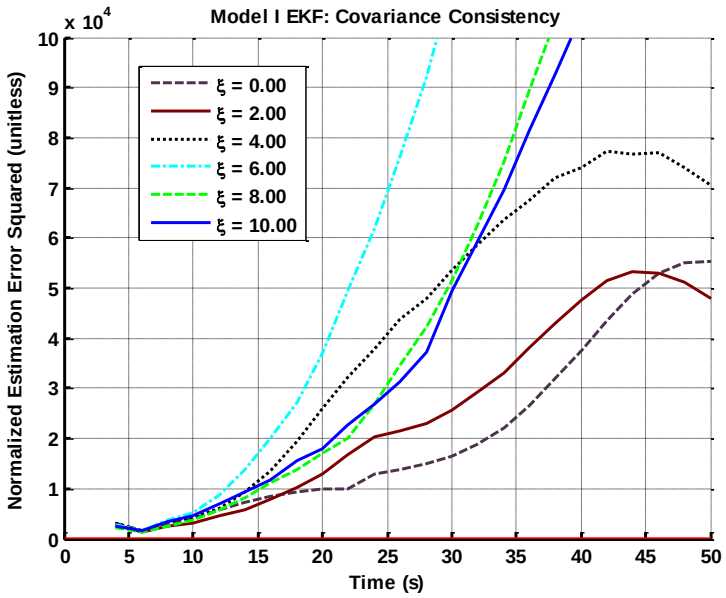

Figure 36: Covariance Consistency EKF Test 3 Model I

8.2.3.2. Model II

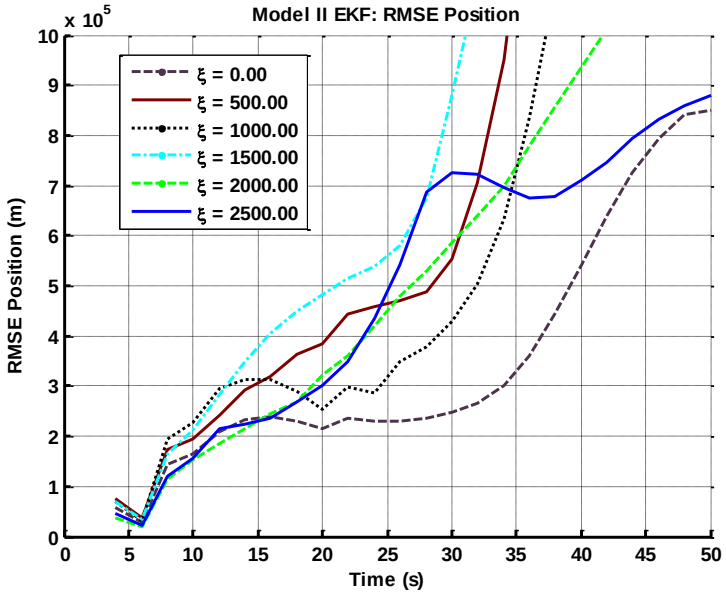

Figure 37: RMSE Position EKF Test 3 Model II

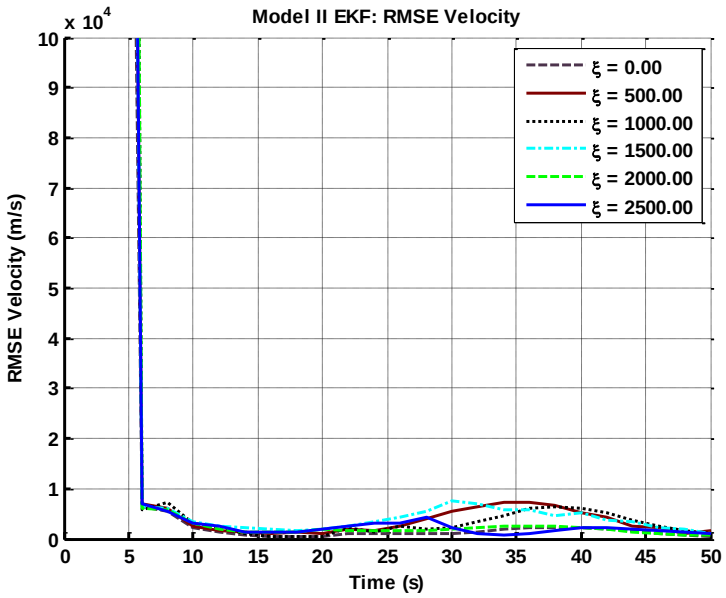

Figure 38: RMSE Velocity EKF Test 3 Model II

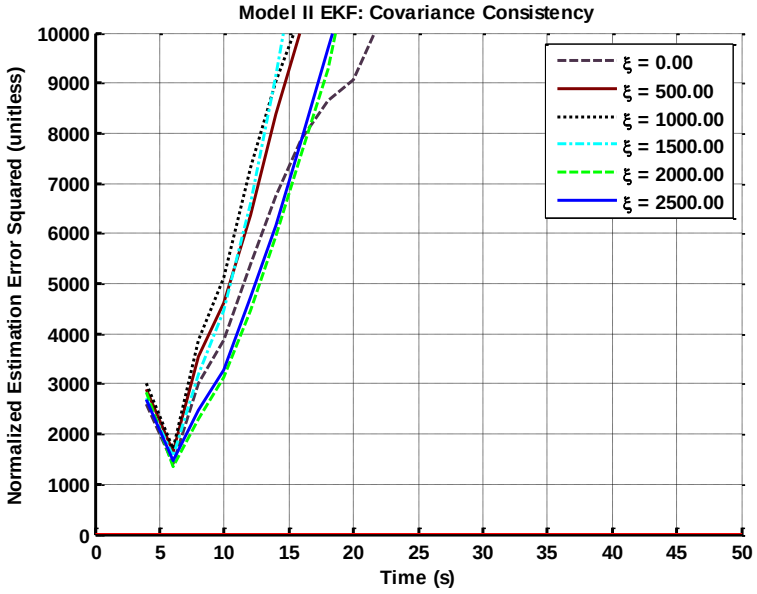

Figure 39: Covariance Consistency EKF Test 3 Model II

8.2.3.3. Model III

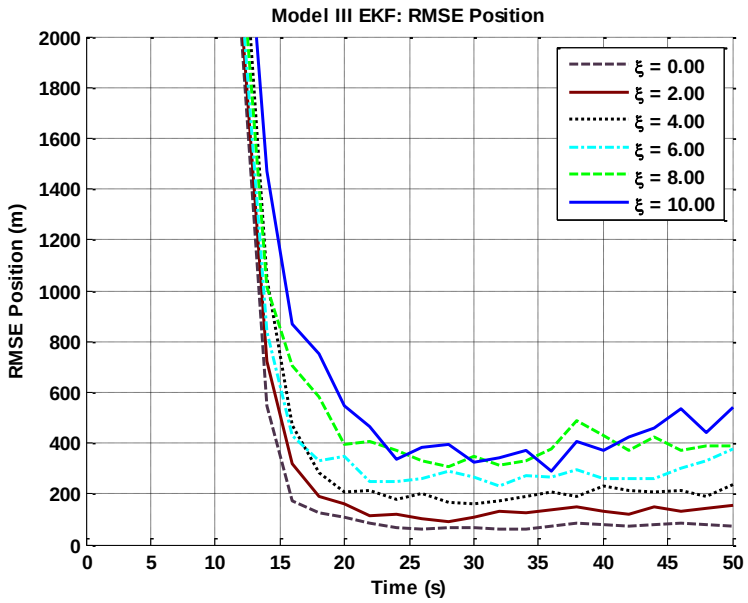

Figure 40: RMSE Position EKF Test 3 Model III

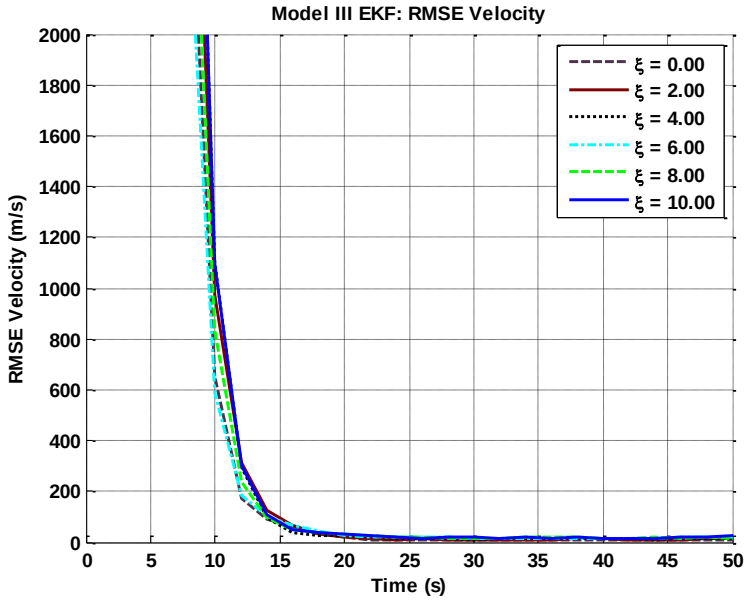

Figure 41: RMSE Velocity EKF Test 3 Model III

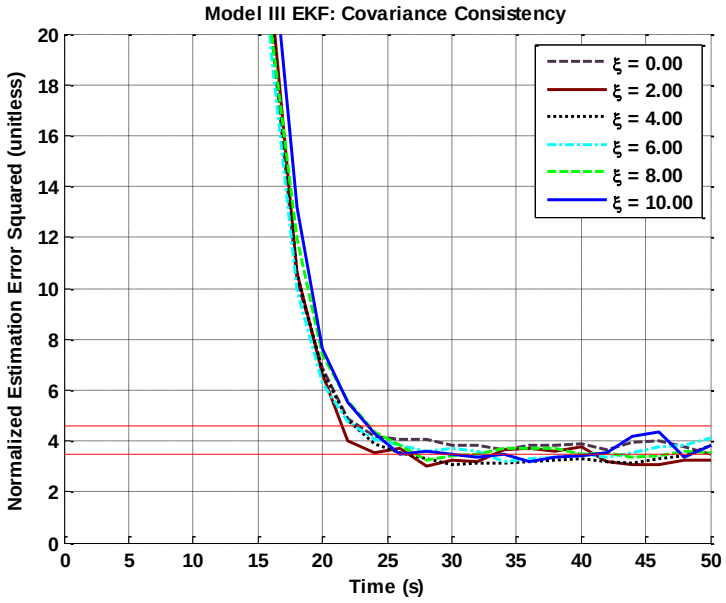

Figure 42: Covariance Consistency EKF Test 3 Model III

8.2.4. Gaussian Particle Filter

For this case, only 4 different threshold levels could be used. A problem occurs with the GPF when the divergence is great due to the degree of non-linearity in the problem. Since a random draw is required to run the particle filter between iterations, if the covariance is not properly defined over a few iterations the filter will quickly diverge. Even so, the plots are worth showing as explained in the Discussion Chapter.

J-Divergence I/II/III	Percentage of Unused Updates	
	Model I	Model II
0/0	0	0
1/500	0.69	71.64
2/1000	59.35	79.11
3/NA	76.3	NA

Table 10: Test 3 GPF J-Divergence Unused Updates Chart

8.2.4.1. Model I

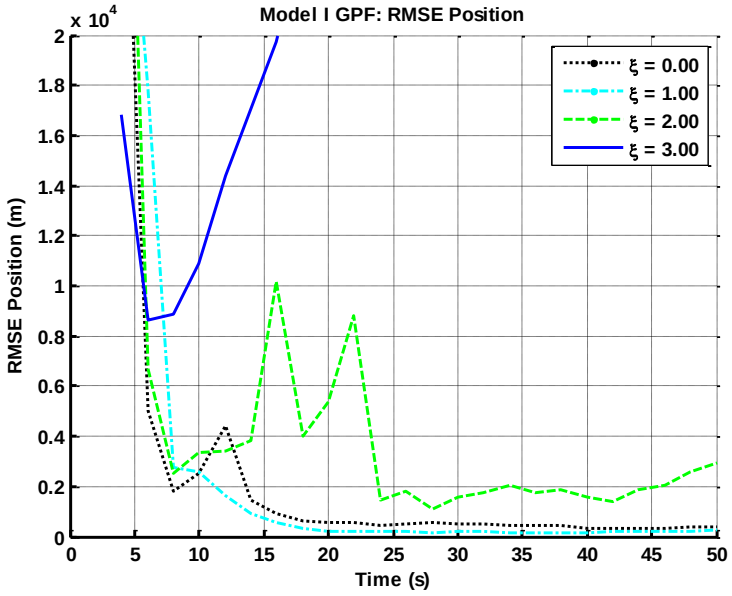

Figure 43: RMSE Position GPF Test 3 Model I

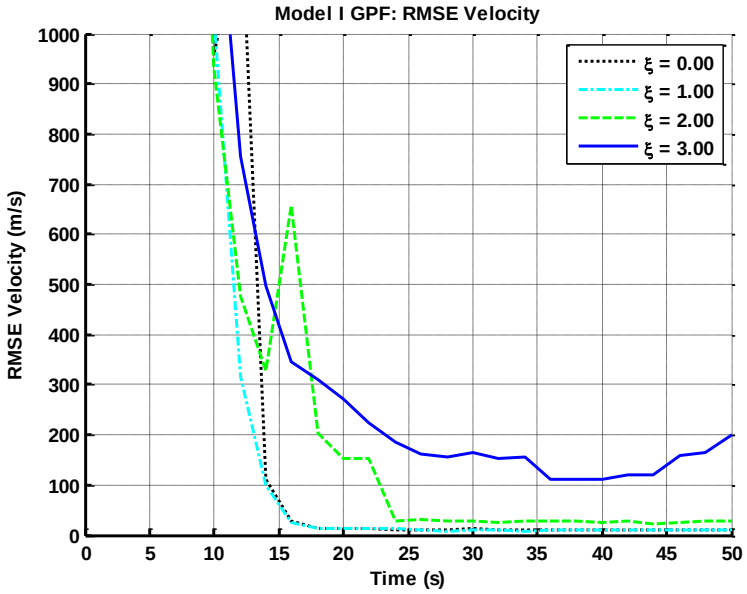

Figure 44: RMSE Velocity GPF Test 3 Model I

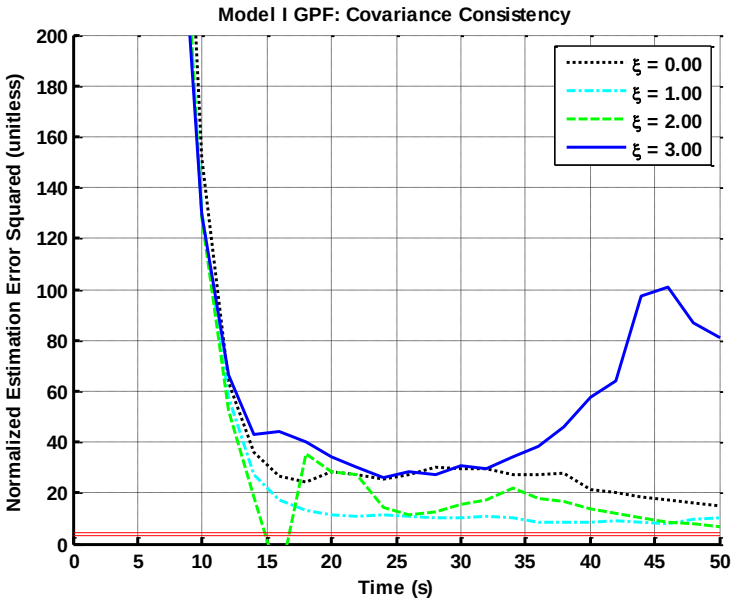

Figure 45: Covariance Consistency GPF Test 3 Model I

8.2.4.2. Model II

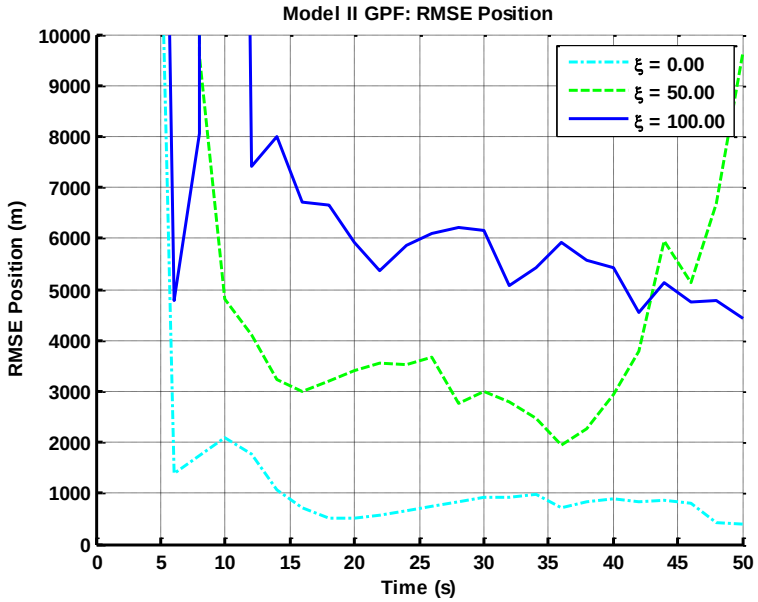

Figure 46: RMSE Position GPF Test 3 Model II

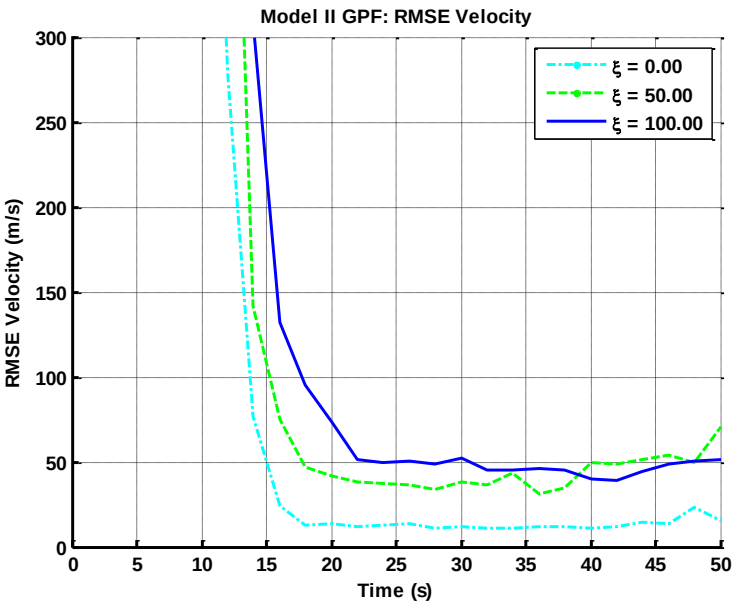

Figure 47: RMSE Velocity GPF Test 3 Model II

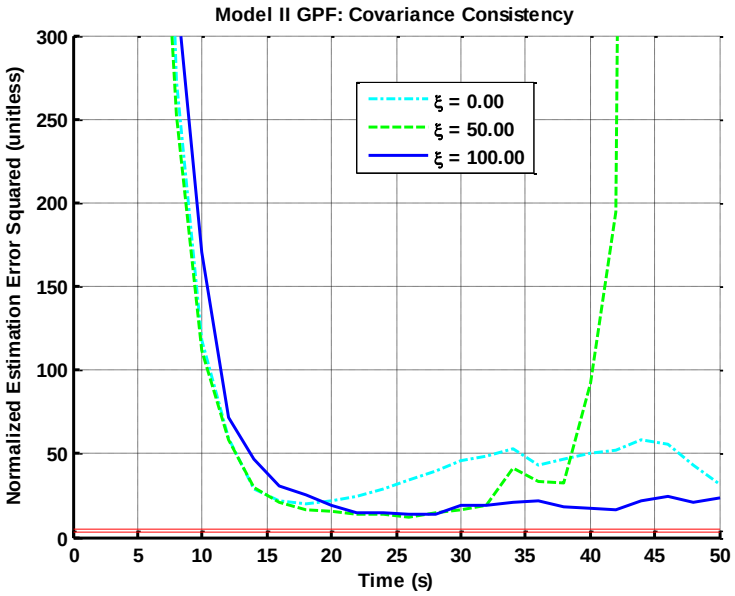

Figure 48: Covariance Consistency GPF Test 3 Model II

Chapter IX: Discussion

The following paragraphs include an interpretation of the results from Chapter VIII. Test 1 is a proof of concept experiment that shows that under unrealistic ideal conditions, censoring is viable. In this case, while the geometry of the problem remains the same, the sensor has an update rate of 100 Hz, which is above and beyond any of the values shown in Table 1. As the J-Divergence threshold is increased for all of the models, the RMSE for each case increases. Results from Model I show that as thresholding increase, the estimators begin to also diverge just as quickly. The global state estimate RMSE in Model II and Model III increase with a higher number of unused updates, but show convergence, unlike in Model I, due to the use of feedback. Feedback effectively allows for the fusion center to have control over when updates are provided to it. The covariance consistency curves for test 1 model 1, show an increase in the NEES when censoring is applied, forcing it to fall just outside of the [3.46, 4.57] 95% confidence interval. For models 2 and 3, the NEES values fall either inside or below the 95% confidence interval.

Test 2 provided a more realistic variation of the original scenario as the update rate was decreased to 0.5 Hz and also a GPF was used in addition to the EKF. The results are similar to that of Test 1, however the RMSE curves appear higher with a similar degree of censoring due to the decreased update rate. The shape of the RMSE curves in both the GPF and the EKF in Test 2 showed similar shapes to the corresponding curves in Test 1. The covariance consistency plots were also similar to those seen in Test 1, with both the GPF and EKF showing an increase in the NEES when censoring is applied in Model 1, and a convergence to values either within the NEES confidence interval or below it, for Models 2 and 3. As Test 2 is similar to Test 1, which follows a linear/Gaussian assumption, convergence in the NEES makes sense for the EKF. For the GPF,

since the local sensors send Gaussian state estimate approximations to the fusion center this also makes sense due to the linear/Gaussian assumption.

Test 3 is the same as Test 2, but is performed under highly non-linear conditions where the angle accuracy is low and the range accuracy is high. Model III for the EKF converged and also outperformed Model III for the EKF in test 2. Since the global state estimate replaces the local state estimates in Model III, the local sensors benefit from the spatial diversity accounted for in the fused estimate. In test 3, this reduces the effect of the non-linearity of the higher angle measurement variance on the state estimate update and allows the sensors to take advantage of the lower range measurement variance. In the GPF, not all of the cases could be run. This is due to the Gaussian approximation of the particle density that doesn't hold in the highly non-linear conditions. When a new measurement comes in that is far away from the particle cloud, the weights of the particles from the likelihood calculation all become zero, which when normalized effectively breaks the filter. Due to this phenomenon, no runs could be completed with Model III of the GPF. A single measurement generated in the tail of the measurement distribution cause the associated filter to break. The EKF is not sensitive to this, because the covariance will simply increase when the random measurements in the tail of the measurement distribution appear. In test 3 for Models I and II, since the local distributions have a greater covariance than the global distribution, the GPF is more robust than in Model III. This is because the measurements that appear in the tail of the measurement distribution, are more likely to appear closer to the cloud of particles generated from the local prior distribution than the global prior distribution, since the local prior distribution particle cloud is larger. In fact for Models I and II, the filters would not always diverge over the course of the 100 Monte-Carlo runs, and the in

these cases the GPF outperformed the EKF. It should be noted that on some random occasions, that the measurement still did appear in the tail of the measurement distribution, causing the filter to diverge. This is illustrated in Model II when the threshold was 50 and it began to diverge, whereas with a threshold of 100, no divergence was observed. All of these assertions are further proven by the shape of the covariance consistency plots. For the EKF in Models I and II, the NEES values diverge since the linear/Gaussian assumption does not hold, but in Model III, the NEES values converge within or just below the expected confidence interval because the fused state estimates allow the linear/Gaussian assumption to hold. For the GPF, both the NEES values for Models I and II show instabilities during convergence on the NEES values, likely due to the measurements that fall far away from the particle cloud and also the fact that the GPF does not hold onto the Gaussian assumption during its update phase. This shows that the EKF gains an advantage over the GPF due to its robustness in filtering measurements that appear in the tail, while the GPF is able to handle the non-linearity for Models I and II.

Chapter X: Conclusion

Three different tests for a distributed radar-tracking scenario have been performed using three different levels of feedback with varying levels of censoring, and two different estimators. Results from the simulation show that feedback can enable the fusion center to receive state estimate updates from the local sensors by indicating when the sensors have enough new information through the J-Divergence metric. This is illustrated by the convergence of the covariance consistency plots for Model II and Model III in Test 1, where the NEES values fall either inside or below the confidence interval regardless of the number of unused updates. Conversely when feedback is not applied as in Model I Test 1, the NEES values increase steadily with the number of unused updates. Censoring can be performed with both an Extended Kalman Filter and a Gaussian Particle Filter for highly non-linear cases as seen in Test 3, however the non-linearity can also cause the EKF to diverge and the Gaussian Particle Filter to break.

There are many potential ways in which this work can be improved. If the particle filter is upgraded to use non-Gaussian distributions, such as a Gaussian Mixture Model or Parzen Window, in local updating, fusion, and J-Divergence calculations, then this may better account for the measurements that appear in the tail of the measurement distribution that can break the GPF. In addition to stopping the GPF in Test 3 from randomly breaking, this will allow for the generation of plots for Model III. Also, the current tests require the J-Divergence to be modelled in advance in order to set a desired percentage of unused updates. It might be useful to derive an algorithm that provides a J-Divergence estimate for given a set covariance of the state estimate. This may not be an easy task, even with the Gaussian J-Divergence equation due to the complexity of the equation. Further tests could be performed using different types of filters such

as the unscented Kalman Filter, or the Daum-Huang Filter. Finally, a more complex network structure could be used rather than the simple tree structure that was provided in this thesis.

Bibliography

[1] R. A. Iltis and K. L. Anderson, "A Consistent Estimation Criterion for Multisensor Bearings-Only Tracking," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 32, no. 1, pp. 108-120, Jan 1996.

[2] R. Karlsson and F. Gustafsson, "Recursive Bayesian estimation: bearings-only applications," *IEE Proceedings Radar, Sonar, and Navigation,* vol. 152, no. 5, pp. 305-313, Oct 2005.

[3] A. Mohammai and A. Asif, "Distributed Consensus + Innovation Particle Filtering for Bearing/Range Tracking With Communication Constraints," *IEEE Transactions on Signal Processing,* vol. 63, no. 3, pp. 620-635, Feb 2015.

[4] G. Soares, A. Arnold-Bos, L. Jaulin and C. Maia, "Interval-Based Target Tracking Approach for Range-Only Multistatic Radar," *IEEE Transactions on Magnetics,* vol. 44, no. 6, pp. 1350-1353, Jun 2008.

[5] K. Zhou and S. Roumeliotis, "Optimal Motion Strategies for Range-Only Constrained Multisensor Target Tracking," *Robotics, IEEE Transactions on,* vol. 24, no. 5, pp. 1168-1185, Oct 2008.

[6] C. Y. Chong, S. Mori and K. C. Chang, "Distributed Multitarget Multisensor Tracking," in *Multitarget-Multi Senosr Tracking: Advanced Applications*, Artech House, 1990, pp. 247-295, Chapter 8.

[7] Y. Bar-Shalom, P. T. Willett and X. Tian, Tracking and Data Fusion A Handbook of Algorithms, Storrs: YBS Publishing, 2011.

[8] Y. Bar-Shalom, X. Li and T. Kirubarajan, Estimation with Applications to Tracking and Navigation Theory Algorithms and Software, New York: John Wiley & Sons, Inc., 2001.

[9] B. Ristic, S. Arulampalam and N. Gordon, Beyond the Kalman filter Particle Filters for Tracking Applications, Boston; London: Artech House, 2004.

[10] Y. Zheng, R. Niu and P. K. Varshney, "Sequential Bayesian Estimation with Censored Data for Multi-Sensor Systems," *IEEE Trans. Signal Process.,* vol. 62, no. 10, pp. 2626-2641, Apr 2014.

[11] E. Masazade, R. Niu and P. K. Varshney, "Dynamic Bit Allocation for Object Tracking in Wireless Sensor Networks," *IEEE Transactions on Signal Processing,* vol. 60, no. 10, pp. 5048-5063, Jun 2012.

[12] C. Rago, P. Willett and Y. Bar-Shalom, "Censoring Sensors: A Low-Communication-Rate Scheme for Distributed Detection," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 32, no. 2, pp. 554-568, Apr 1996.

[13] M. Cetin, Lei Chen, J. W. Fisher, A. T. Ihler, R. L. Moses, M. J. Wainwright and A. S. Willsky, "Distributed Fusion in Sensor Networks," *Signal Processing Magazine, IEEE,* vol. 23, no. 4, pp. 42-55, July 2006.

[14] K. S. Miller and D. M. Leskiw, "Nonlinear Estimation With Radar Observations," *IEEE Transactions on Aerospace and Electronic Systems,* Vols. AES-19, no. 2, pp. 192-200, Mar 1982.

[15] F. Daum, "Nonlinear filters: Beyond the Kalman filter," *IEEE Aerospace and Electronic Systems Magazine,* vol. 20, no. 8, pp. 57-69, Aug. 2005.

[16] F. Daum, "Exact finite-dimensional nonlinear filters," *IEEE Transactions on Automatic Control,* vol. 31, no. 7, pp. 616-622, Jul 1986.

[17] J. H. Kotecha and P. M. Djuric, "Gaussian Particle Filtering," *IEEE Transactions on Signal Processing,* vol. 51, no. 10, pp. 2592-2601, Oct. 2003.

[18] D. K. Barton, Radar System Analysis and Modeling, Norwood: Artech House, Inc., 2005.

[19] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 29, no. 3, pp. 1015-1022, Jul. 1993.

[20] C.-Y. Chong, S. Mori, W. H. Barker and K.-C. Chang, "Architectures and algorithms for track association and fusion," *IEEE Aerospace and Electronic Systems Magazine,* vol. 15, no. 1, pp. 5-13, Jan 2000.

[21] J. C. Principe, Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives, Springer Science+Business Media, 2010.

[22] W. Zhu, J. Tianzi and X. Li, "Local Region Based Medical Image Segmentation Using J-Divergence Measures," in *IEEE Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, Sep 2005.

[23] G. Lefebvre, R. Steele and A. C. Vandal, "A path sampling identity for computing the Kullback-Leibler and J divergences," *Computational Statistics & Data Analysis,* vol. 54, no. 7, pp. 1719-1731, Jul 2010.

[24] S. J. Roberts and W. D. Penny, "Variational Bayes for Generalized Autoregressive Models," *IEEE Transactions on Signal Processing,* vol. 50, no. 9, pp. 2245-2257, Sept 2002.

[25] R. Niu, P. K. Varshney, M. Alford, A. Bubalo, E. Jones and M. Scalzo, "Curvature Nonlinearity Measure and Filter Divergence Detector for Nonlinear Tracking Problems," in *11th International Conference on Information Fusion*, Cologne, June 2008.

Vita

Armond S. Conte II received his Bachelor of Science degree in Biomedical Engineering from Virginia Commonwealth University in 2008.