Air-to-Air Missile
Enhanced Scoring with Kalman Smoothing

THESIS

Jonathon Gipson, Captain, USAF

AFIT/GE/ENG/12-18

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GE/ENG/12-18

Air-to-Air Missile
Enhanced Scoring with Kalman Smoothing

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Jonathon Gipson, B.S.E.E.
Captain, USAF

March 2012

AFIT/GE/ENG/12-18

Air-to-Air Missile Enhanced Scoring with Kalman Smoothing

Jonathon Gipson, B.S.E.E.
Captain, USAF

Approved:

// signed //	7 Mar 2012
Maj Kenneth A. Fisher, Ph.D. (Chairman)	date
// signed //	7 Mar 2012
Dr. Meir Pachter (Member)	date
// signed //	7 Mar 2012
LtCol Michael Stepaniak, Ph.D. (Member)	date

AFIT/GE/ENG/12-18

Abstract

The United States Air Force air-to-air Weapons System Evaluation Program (WSEP) targets unmanned aerial drones in hundreds of live-fire missile tests each year. The current QF-4 drone inventory is expected to be depleted by 2015. The QF-16 Full Scale Aerial Target FSAT contract has been awarded to convert usable early model F-16's into remote-controlled drones. The QF-16 will provide a highly maneuverable, realistic testing environment for 5th generation fighters. When a missile fails to destroy a target aircraft, a scoring system is useful in determining what caused the failed intercept. A correct estimate of a missile's flight path is critical for weapons test and evaluation to ensure accuracy. This research analyzes the use of Kalman smoothing techniques with Frequency-Modulated Continuous Wave (FMCW) radar sensors arranged on the QF-16 platform to satisfy these goals.

Estimating the trajectory of an air-to-air missile is difficult due to high dynamic capabilities, short time of flight, and advanced guidance systems. Kalman smoothers lend themselves to tasks such as post-flight trajectory estimation because they combine the utility of forward and backward-propagating Kalman filters. The combined result is optimal post-flight missile scoring.

Six Kalman smoothers (EKS, IEKS, SFRA EKS, UKS, IUKS, and SFRA UKS) are simulated. The performance assessment is based on multiple Monte Carlo comparisons among all algorithms with a variety of missile dynamics models and air-to-air engagment scenarios. Simulations are conducted utilizing varying levels of injected noise and sensor availability to provide a comprehensive analysis of potential performance benefits. This technical assessment provides the basis for recommendation of the Unscented Kalman Smoother (UKS) as the DoD/USAF standard for post-processing and scoring live-fire missile data.

Acknowledgements

First and foremost, I owe a large debt of gratitude to God, my Mom, Dad, Sister, and fiancée...

Jonathon Gipson

Table of Contents

List of Figures

Figure Page

Figure Page

Figure Page

List of Tables

List of Abbreviations

Abbreviation		Page
FSAT	Full Scale Aerial Target	iv
FMCW	Frequency-Modulated Continuous Wave	1
EKF	Extended Kalman Filter	2
UKF	Unscented Kalman Filter	2
EKS	Extended Kalman Smoother	2
IEKS	Iterated Extended Kalman Smoother	2
SFRA	Single Filter Reactive Adaptation	2
UKS	Unscented Kalman Smoother	2
IUKS	Iterated Unscented Kalman Smoother	2
COTS	Commercial-Off-The-Shelf	2
EKF	Extended Kalman Filter	9
NED	North-East-Down	29
PDOP	Positional Dilution of Precision	48
CV	constant velocity	50
CA	constant acceleration	50
CT	three-dimensional coordinated turn	50
GRDCS	Gulf Range Drone Control System	57
RSS	Root of Sum of Squares	65

AIR-TO-AIR MISSILE
ENHANCED SCORING WITH KALMAN SMOOTHING

I. Introduction

1.1 Motivation and Problem Description

A correct estimate of a missile's flight path is essential to USAF test and evaluation for ensuring accuracy and functionality. The United States Air Force air-to-air Weapons System Evaluation Program (WSEP) targets unmanned aerial drones in hundreds of live-fire missile tests each year [15]. The current QF-4 drone inventory is expected to be depleted by 2015. The QF-16 Full Scale Aerial Target (FSAT) contract has been awarded to convert usable early model F-16's into remote-controlled drones. The QF-16 will provide a highly-maneuverable, realistic testing environment for 5th generation fighters. The delivery of the first six QF-16's is currently scheduled for 2014. In order to accomplish their mission, WSEP requires a scoring system capable of estimating the trajectory of a missile relative to the drone aircraft. When a missile fails to destroy a target aircraft, this scoring system is useful in analyzing whether a missile suffered a guidance failure, decoyed on aircraft countermeasures, or lacked energy or maneuverability to complete the intercept. Additionally, many firing profiles are designed to be "near misses" for drone preservation. Missile flight path reconstruction near the drone allows evaluation of the missile's performance. This research analyzes the use of Kalman smoothing techniques coupled with Frequency-Modulated Continuous Wave (FMCW) radar sensors carefully arranged on the QF-16 platform to satisfy these goals.

1.2 Overview

The purpose of this research is to develop a series of enhanced Kalman smoothers to glean maximum benefit from post-flight live-fire missile test data for trajectory reconstruction and missile scoring. A secondary objective is to quantify the improvement of post-flight analysis versus real-time analysis. This research refines previous work to reconstruct the flight path of an air-to-air missile relative to a target (QF-16 drone) aircraft. The dynamics models, measurement models, engagement scenarios, and truth data are all used from previous work [11]. The Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) also previously developed form the performance baseline for this effort.

Two development paths were followed to analyze the performance potential of Kalman smoothing for post-flight trajectory estimate improvement. First-order post-flight approximation is performed by the Extended Kalman Smoother (EKS), Iterated Extended Kalman Smoother (IEKS), and Single Filter Reactive Adaptation (SFRA) EKS. Second-order approximation is handled by the Unscented Kalman Smoother (UKS), the Iterated Unscented Kalman Smoother (IUKS), and the Single Filter Reactive Adaptation (SFRA) UKS developed for this research. The real-time versions of the iterated and SFRA filters (IEKF, SFRA EKF, IUKF, SFRA UKF) are also analyzed for comparison with the original EKF and UKF.

1.3 Assumptions

This research builds upon existing work with the additional assumption that missile scoring will be performed post-flight. Accordingly, this Kalman smoother-based approach to missile scoring cannot be performed in real time. The RF sensor arrangement is designed to resemble the geometry provided by the QF-16 platform. The sensor characteristics are modeled after a commercial-off-the-shelf (COTS) RF sensor with a maximum range of 350 meters [11]. This research is limited in scope to assess the performance improvement of various post-flight Kalman smoothers over previously attempted real-time Kalman filters when estimating the trajectory of an inbound air-to-air missile.

1.4 Problem Approach

This thesis generates high-fidelity truth data which is noise-corrupted and passed to a Kalman smoother for reconstruction. The Kalman smoothers (EKS, IEKS, SFRA EKS, UKS, IUKS, and SFRA UKS) are simulated using *MATLAB*. Specifically, simulation tools *ProfGen* [8] and *Argos 3.0* [1] are used to process sample target and shooter trajectories into a high fidelity 'truth' missile trajectory. Noise is be added to truth data before it is sampled and used as representative sensor measurements. The Kalman smoothers are employed with the sensor measurements to reconstruct the true missile trajectory. The performance assessment is based on multiple Monte Carlo comparisons among all the Kalman smoothers with a variety of high fidelity missile models. Simulations are conducted utilizing varying levels of injected noise and random sensor occlusion to provide a comprehensive analysis of potential performance benefits. This technical assessment provides the basis for recommendation of the Unscented Kalman Smoother (UKS) as the DoD/USAF standard for post-processing and scoring live-fire missile data. See Chapter IV for the motivation and methodology for creating the various Kalman filters and smoothers.

Estimating the trajectory of an air-to-air missile is difficult due to high dynamic capabilities, short time of flight, and advanced guidance systems. Kalman smoothers lend themselves to tasks such as post-flight trajectory estimation because they optimally combine forward and backward-propagating Kalman filters. The combined result is optimal post-flight missile scoring.

1.5 Research Contributions

This research is focused on providing the best available post-flight estimation of an inbound missile's trajectory. This research answers the following questions based on analytics and discrete simulations:

- Why use a post-processing smoother instead of a real-time forward-only filter?

- How does accuracy and precision compare using different Kalman smoothing techniques?
- How does performance vary using different missile dynamics models and engagement geometries?
- What is the overall precision of this scoring system based on simulated performance of COTS sensors with Kalman smoothing applied?
- What is the optimal Kalman smoothing technique for post-processing live fire missile data?
- Using this optimal technique, how large is the expected mean position error of a simulated air-to-air missile trajectory?

1.6 Thesis Outline

A discussion of previous research in these subject areas is provided in Chapter III. The mathematical basis for the development of the six Kalman smoothers is available in Chapter II. The sensor suite physical layout, dynamics models, filter initialization, truth model, and air-to-air engagement scenarios are described in Chapter IV. The results, which provide a performance stratification for all of the Kalman Smoothers is available in Chapter V. Chapter VI provides a summary of the most important results as well as recommendations for future research in this area. Appendix A contains the results of all simulation profiles for each Kalman smoother variant.

II. Mathematical Background

2.1 Mathematical Notation

This thesis uses the following mathematical notation:

- **Scalars**: Scalars are denoted by lower or upper case non-bold characters (e.g., x or X)
- **Vectors**: Vectors are represented by lower case characters in bold font (e.g., $\boldsymbol{x}$)
- **Matrices**: Matrices are denoted by upper case characters in bold font or upper case script characters (e.g., $\boldsymbol{X}$ or $\mathscr{X}$)
- **Vector Transpose**: A vector transpose is indicated by a superscript Roman T (e.g., x^T)
- **Estimated Variables**: An estimated variable is designated by the use of a *hat* character (e.g., $\hat{x}$)
- **Reference Frame**: If a variable's reference frame is designated, it is annotated by a superscript character (i.e., $\boldsymbol{x}^n$ is the vector $\boldsymbol{x}$ in the n frame)
- **Direction Cosine Matrices**: A direction cosine matrix from frame i to frame n is represented by C_i^n
- **Discrete Time**: The subscript k is used to denote the k-th time step in a discrete time sequence (i.e., $\hat{\boldsymbol{x}}_k$ is an estimate of the vector $\boldsymbol{x}$ at time k)
- ***Apriori* Estimate**: An estimate of a system's navigation parameters prior to incorporating a measurement update is designated with a superscript minus (e.g., $\hat{\boldsymbol{x}}_{k|k-1}$)
- ***Aposteriori* Estimate**: An estimate of a system's navigation parameters after incorporating a measurement update is designated with a subscript indicating the estimate is predicated upon a filter estimate (e.g., $\hat{\boldsymbol{x}}_{k|k}$)
- ***Iterated* Estimate**: An estimate of a system's navigation parameters after incorporating a measurement update, is designated with a subscript indicating the estimate is predicated upon an observed measurement (e.g., $\hat{\boldsymbol{x}}_{k|k}^{+}$)

2.2 The Kalman Filter

The world is filled with systems that exhibit random behavior. If we can make simplifying assumptions to characterize the nature of this behavior, we can attempt to model these real-world systems. The Kalman Filter [6] is a commonly used recursive algorithm which can provide a statistically optimal estimate of stochastic systems. The KF has two components, a dynamics model and an observation model. The dynamics model describes the expected behavior of the system as well as system dynamics uncertainties. For example, an air-to-air missile's dynamics are partially governed by its velocity and achievable turn rate, yet disturbances such as wind introduce uncertainties. The KF uses this model to predict the changes in system states between measurement updates. Process noise is the stochastic component associated with the dynamics model. The second component is the observation model which provides a mathematical relationship between sensor measurements and the system states. This relationship can be used to update system states based on the sensor data. Measurement noise is the stochastic component associated with the observation model. The power of the KF lies in its ability to update the system state estimates by optimally weighting the measurement data with the predictions of the states based on the dynamics model.

There are several simplifying assumptions that are made about the stochastic nature of the system before we can ensure the KF produces an optimal estimate. The process and measurement noise components are assumed to be Gaussian processes. This means that all information regarding the stochastic components are captured by the first and second moments (expected value and covariance). The Gaussian noise sources are assumed to be zero-mean, additive, and uncorrelated in time. Using state space representation, the KF assumes a linear system dynamics model of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) + \boldsymbol{G}(t)\boldsymbol{w}(t) \tag{2.1}$$

which represents the system dynamics. The variable $\boldsymbol{x}(t)$ is a vector of system states, $\boldsymbol{u}(t)$ is an input vector and $\boldsymbol{w}(t)$ the process noise vector. The relationship between the inputs, process noise, and system state vectors is defined by the $\boldsymbol{F}(t)$, $\boldsymbol{B}(t)$ and $\boldsymbol{G}(t)$ matrices. The process noise covariance matrix is defined as

$$E[\boldsymbol{w}(t)\boldsymbol{w}^T(t)] \triangleq \boldsymbol{Q}. \tag{2.2}$$

The $\boldsymbol{Q}$ matrix contains the covariance and cross-covariance values of the process noise and can be increased to account for a low fidelity dynamics model [6].

In state space, the KF assumes a linear measurement model of the form

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{2.3}$$

where $\boldsymbol{z}_k$ is a vector of measurements, the matrix $\boldsymbol{H}_k$ relates the measurements to current states and $\boldsymbol{v}_k$ is a vector of zero-mean, additive, white, Gaussian measurement noise. The measurement noise covariance kernel is defined by

$$E[\boldsymbol{v}_k \boldsymbol{v}_j^T] = \boldsymbol{R}\delta_{kj} \tag{2.4}$$

where $\boldsymbol{R}$ is the measurement noise covariance. The $\boldsymbol{R}$ matrix contains the covariance and cross-covariance values of the measurement noise components and can be increased to account for low quality sensor data. The $\boldsymbol{Q}$ to $\boldsymbol{R}$ ratio is an indicator of how trustworthy the dynamics model is in relation to the measurement model. Values within $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be modified as tuning parameters to optimize KF performance [6].

The KF must be employed as a discrete-time algorithm to effectively handle digitally sampled data in Equation 2.3 as well as to implement Equation (2.1) on a computer. The dynamics model difference equation becomes

$$\boldsymbol{x}_k = \boldsymbol{\phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{B}_{k-1}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} \tag{2.5}$$

where the discrete white, Gaussian noise strength is $\boldsymbol{Q}_d$. The discrete input $\boldsymbol{u_k}$, is formed by sampling $\boldsymbol{u}$ at the current time step and assuming it remains constant over the propagation interval. This research relies on the Van Loan method to convert from a continuous-time differential equation to a discrete-time difference equation [3]. Beginning with the dynamics model, Equation (2.1), the matrix

$$\boldsymbol{A} = \begin{bmatrix} -\boldsymbol{F} & \boldsymbol{G}\boldsymbol{W}\boldsymbol{G}^T \\ \boldsymbol{0} & \boldsymbol{F}^T \end{bmatrix} \Delta t \tag{2.6}$$

is created. The sample time, Δt, is the time delay between propagation steps. The matrix

$$\boldsymbol{B} = e^{\boldsymbol{A}} = \begin{bmatrix} \dots & \boldsymbol{\phi}^{-1}\boldsymbol{Q}_d \\ \boldsymbol{0} & \boldsymbol{\phi}^T \end{bmatrix} \Delta t \tag{2.7}$$

contains the state transition matrix, $\boldsymbol{\phi}$. The discrete-time noise strength, $\boldsymbol{Q}_d$, can be calculated $\boldsymbol{B}$ using the Van Loan method in linear algebra.

The system inputs are assumed to be deterministic and all noise sources are assumed to be zero-mean, additive, white, and Gaussian. Since the state vector is a function of these two components, it has a Gaussian PDF and can be completely described by its expected value and covariance.

The discrete-time propagate equations are described by

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{\phi}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}_{k-1}\boldsymbol{u}_{k-1} \tag{2.8}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{\phi}_k \boldsymbol{P}_{k-1|k-1}\boldsymbol{\phi}_k^T + \boldsymbol{Q}_d \tag{2.9}$$

where $\hat{\boldsymbol{x}}_{k|k-1}$ is the *apriori* state estimate and $\boldsymbol{P}_{k|k-1}$ is the associated covariance. Measurement updates are calculated by

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k[\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}] \tag{2.10}$$

and

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T[\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k]^{-1}. \tag{2.11}$$

The update covariance matrix is given by

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{P}_{k|k-1} \tag{2.12}$$

where $\boldsymbol{K}_k$ defined in Equation (2.11) is the Kalman gain which optimally weights the results of the dynamics and observation models based on the current measurement update [6]. $\boldsymbol{P}_{k|k}$ is the covariance associated with the new state estimate. At this point, the KF returns to Equation (2.8) and the *aposteriori* state estimate, $\hat{\boldsymbol{x}}_{k|k}$, becomes the new *apriori* state estimate for next propagation step $\hat{\boldsymbol{x}}_{k+1|k}$.

2.2.1 Extended Kalman Filter . The Extended Kalman filter (EKF) is the nonlinear version of the Kalman Filter. The EKF uses the first-order Taylor series expansion (Jacobian) method to linearize the dynamics and/or observation model. This research uses linearized dynamics models and a nonlinear measurement model. Consequently, only the observation model is linearized. The assumption of zero-mean, additive, white Gaussian noise sources is unchanged from the conventional KF. Therefore, the mean and covariance completely characterize the state estimate.

Since its inception in 1965, the EKF has become the industry standard for theoretical non-linear state estimation [6]. The EKF relies on first-order Taylor series approximations of the input and output equations to propagate and update the error states of the system state variables. Therefore, one must assume error states are adequately modeled by 1^{st} order approximation. As one might predict, the EKF is relatively easy to implement. This is offset by poor tracking performance when the system dynamics or observation functions are not well-represented by a 1^{st} order approximation. The nonlinear equation

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(x, u) + \boldsymbol{G}\boldsymbol{w} \tag{2.13}$$

must be linearized by calculating the Jacobian

$$\boldsymbol{F} \triangleq \frac{\partial \boldsymbol{f}}{\delta \boldsymbol{x}}|_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{2.14}$$

This research focuses solely on linearized 3-D missile dynamics models (constant velocity, constant turn rate, and constant acceleration). As mentioned, the observation model is distinctly nonlinear. The general form of which is

$$\boldsymbol{z}_k = \boldsymbol{h}[\boldsymbol{x}_k] + \boldsymbol{v}_k \tag{2.15}$$

where $\boldsymbol{z}_k$ is the measurement, $\boldsymbol{h}[.]$ is a nonlinear observation function and $\boldsymbol{v}_k$ is the discrete-time additive white Gaussian noise component.

In order to linearize the observation function about the current state vector, a Jacobian matrix is calculated by performing a first-order Taylor expansion of each nonlinear function with respect to the current states. This Jacobian is evaluated at the current state estimate

$$\boldsymbol{H}_k = \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{x}}|_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{2.16}$$

Afterwards, a nominal (predicted) measurement is generated based on the current state estimate

$$\hat{\boldsymbol{z}}_k = \boldsymbol{h}[\hat{\boldsymbol{x}}_{k|k-1}] \tag{2.17}$$

The final step is defining an error state, $\delta\boldsymbol{z}$, which is the difference between the measurement and the nominal (predicted) measurement

$$\delta\boldsymbol{z}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k \tag{2.18}$$

Described in continuous time, the linear CV dynamics model is [2]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{w}(t) \tag{2.19}$$

where

$$\boldsymbol{F} = \left[\begin{array}{c|c} \boldsymbol{0}_{3x3} & \boldsymbol{I}_{3x3} \\ \hline \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \end{array}\right] \tag{2.20}$$

$$\boldsymbol{G} = \left[\begin{array}{c} \boldsymbol{0}_{3x3} \\ \hline \boldsymbol{I}_{3x3} \end{array}\right] \tag{2.21}$$

and the magnitude of the noise vector $\boldsymbol{w}(t)$ is defined by

$$E[\boldsymbol{w}(t)\boldsymbol{w}(t+\tau)] = \boldsymbol{Q} = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{bmatrix} \tag{2.22}$$

where q becomes a tuning parameter describing the uncertainty associated with the model.

Instead of modeling the acceleration components as zero-mean, white, Gaussian variables, the CA model uses three additional states to propagate acceleration components. The CA model can be described as

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & v_x & v_y & v_z & a_x & a_y & a_z \end{bmatrix}^T. \tag{2.23}$$

The $\boldsymbol{F}$ and $\boldsymbol{G}$ matrices can be described as [2]

$$\boldsymbol{F} = \left[\begin{array}{c|c|c} \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} & \mathbf{0}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{array}\right] \tag{2.24}$$

$$\boldsymbol{G} = \left[\begin{array}{c} \mathbf{0}_{6x3} \\ \hline \boldsymbol{I}_{3x3} \end{array}\right]. \tag{2.25}$$

The dynamic noise strength matrix, $\boldsymbol{Q}$, is in the same form as Equation (2.22).

The CT model contains exactly the same nine navigation states as the CA model. The main difference is the velocity navigation states are propagated according to an assumed constant turn rate, $\boldsymbol{\omega}$. The CT model is described mathematically as

$$\boldsymbol{\omega} = \frac{|\boldsymbol{v} \times \boldsymbol{a}|}{|\boldsymbol{v}|^2} \tag{2.26}$$

The acceleration states are propagated according to

$$\dot{\boldsymbol{a}}(t) = -\omega^2 \boldsymbol{v}(t) + \boldsymbol{w}(t). \tag{2.27}$$

The continuous time linear dynamics matrices are

$$\boldsymbol{F} = \left[\begin{array}{c|c|c} \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} & \mathbf{0}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} \\ \hline \mathbf{0}_{3x3} & A & \mathbf{0}_{3x3} \end{array}\right] \tag{2.28}$$

$$\boldsymbol{A} = \begin{bmatrix} -\omega^2 & 0 & 0 \\ 0 & -\omega^2 & 0 \\ 0 & 0 & -\omega^2 \end{bmatrix} \tag{2.29}$$

$$\boldsymbol{G} = \left[\begin{array}{c} \boldsymbol{0}_{6x3} \\ \hline \boldsymbol{I}_{3x3} \end{array}\right] \tag{2.30}$$

where $\boldsymbol{v}(t)$ is the 3D velocity vector and $\boldsymbol{\omega}(t)$ is a corresponding vector of independent, zero-mean, white Gaussian noise sources. All three linearized dynamics models (CV, CA, CT) are functions of time which allows them to account for nonlinear effects.

Since all three dynamics models are linearized, the propagate equations are unchanged from the conventional KF. The post-update state estimate is calculated with

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \delta \boldsymbol{z}_k. \tag{2.31}$$

The EKF functions very similarly to the conventional KF with the exception of the Jacobian linearization of the observation function, $\boldsymbol{h}_k$, in the measurement equations.

2.2.2 Iterated Extended Kalman Filter. The IEKF functions very similarly to the EKF with the exception of multiple iterations of Equations (2.32) through (2.33) until $\boldsymbol{x}_k^{n+1} - \boldsymbol{x}_k^n$ is sufficiently small. The goal of the IEKF is to relinearize $\boldsymbol{h}[\cdot]$ to improve estimation quality. The purpose of iteration is converge on a better state estimate before propagating to the next time step. As expected, this process requires heavy computation but should yield results similar to a 2^{nd} order Gauss filter [7].

1. The IEKF performs the first iteration of each time step, t_k, the same as the simple linearized EKF Equations (2.8) through (2.10). The IEKF performs iteration from $n = 0, 1 \ldots N-1$ on

$$\boldsymbol{K}_k^n = \boldsymbol{P}_{k|k-1}[\boldsymbol{H}_k^n]^T[\boldsymbol{H}_k^n \boldsymbol{P}_{k|k-1}[\boldsymbol{H}_k^n]^T + \boldsymbol{R}_k]^{-1} \tag{2.32}$$

and

$$\hat{\boldsymbol{x}}_k^{n+1} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k^n[\boldsymbol{z}_k - \boldsymbol{h}_k^n - \boldsymbol{H}_k^n[\hat{\boldsymbol{x}}_{k|k-1} - \hat{\boldsymbol{x}}_k^n]] \tag{2.33}$$

where the initial $\hat{\boldsymbol{x}}_k^n|_{n=0}$ is the first *aposteriori* estimate, $\hat{\boldsymbol{x}}_{k|k}$ and the initial $\hat{\boldsymbol{H}}_k^n|_{n=0}$ is the first *aposteriori* $\boldsymbol{H}[\cdot\,]$ matrix, $\hat{\boldsymbol{H}}_{k|k}$ and.

2. Once the newly iterated *aposteriori* state estimate, $\hat{\boldsymbol{x}}_k^{n+1}$ is obtained, it is compared to the previous iteration, $\hat{\boldsymbol{x}}_k^n$ by

$$\boldsymbol{x}_k^{n+1} - \boldsymbol{x}_k^n < \boldsymbol{C}_I \tag{2.34}$$

where $\boldsymbol{C}_I$ is a user-defined threshold.

3. If Equation (2.34) is not satisfied, the filter stays on the current time step t_k and reevaluates $\boldsymbol{h}[\cdot\,]$ to achieve a better $\hat{\boldsymbol{x}}_k^{n+1}$.

$$\hat{\boldsymbol{x}}_k^{n+1} \rightarrow \hat{\boldsymbol{x}}_k^n \tag{2.35}$$

4. Once Equation (2.34) is satisfied or the maximum number of N iterations is met, the result becomes the iterated *aposteriori* state estimate, $\hat{\boldsymbol{x}}_k^+$, and the IEKF moves on to the next time step, t_{k+1} and the process starts again.

$$\hat{\boldsymbol{x}}_k^{n+1} \rightarrow \hat{\boldsymbol{x}}_k^+ \tag{2.36}$$

$$t_k \rightarrow t_{k+1} \tag{2.37}$$

Note that $\hat{\boldsymbol{x}}_k^+$ is the final state estimate resulting from the iteration process.

2.2.3 Single Filter Reactive Adaptation Extended Kalman Filter. Although conceived independently, the SFRA EKF is quite similar to an algorithm introduced by Blackman and Popoli [2]. The SFRA EKF does not guarantee optimality but is able to detect abrupt maneuvers through residual monitoring. It functions very similarly to the EKF except it iterates through Equations (2.11) through (2.10) using a re-linearized $\boldsymbol{h}[.]$ from Equation (4.25). In fact, the first iteration of the SFRA EKF is identical to the EKF. After the first iteration, the algorithm does not repeat propagation Equations (2.8) and (2.9). The SFRA EKF monitors the mean of the filter residuals which are compared to a threshold value set by the filter designer.

The SFRA EKF performs the first iteration of each time step, t_k identically to linearized EKF equations (2.8) through (2.10). Once the first measurement residuals are obtained,

1. Test the 2-norm of i residuals *(i = 1,2,...,M)* versus a threshold based on the norm of the square root of the corresponding variance $\boldsymbol{S}_k$ multipled by a tuning parameter $\boldsymbol{C}_R$,

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k \tag{2.38}$$

$$||\boldsymbol{\gamma}_{[1:M]}|| < \boldsymbol{C}_R ||\sqrt{\boldsymbol{S}_k}|| \tag{2.39}$$

   where $\sqrt{\boldsymbol{S}_k}$ is obtained using the Cholesky decomposition.

2. If Equation (2.39) is not satisfied, do not proceed past the current time step, t_k. Instead, shift the *aposteriori* estimate, $\boldsymbol{x}^n_{k|k}$, towards the measurement.

3. Re-run the EKF algorithm, skipping the EKF propagation steps.

4. Relinearize $\boldsymbol{h}[\cdot\,]$ using the new *aposteriori* estimate, $\boldsymbol{x}^n_{k|k}$, using

$$\boldsymbol{H}^n_{k|k} = \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{x}}|_{\hat{\boldsymbol{x}}^n_{k|k}}. \tag{2.40}$$

   Afterwards, a new measurement prediction is generated based on the current state estimate

$$\hat{\boldsymbol{z}}^n_{k|k} = \boldsymbol{h}[\hat{\boldsymbol{x}}^n_{k|k}]. \tag{2.41}$$

Observations are re-gated based on $\hat{\boldsymbol{x}}^n_{k|k}$ which forms a new measurement $\boldsymbol{z}^n_k$. A new Kalman gain is calculated and new residuals are obtained, γ^n_i, which will attempt to satisfy Equation (2.39),

$$\boldsymbol{K}^n_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^{nT}_{k|k-1}[\boldsymbol{H}^n_{k|k-1}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^{nT}_k + \boldsymbol{R}_k]^{-}1 \tag{2.42}$$

$$\boldsymbol{\gamma}^n = \boldsymbol{z}^n_k - \hat{\boldsymbol{z}}^n_k. \tag{2.43}$$

Next, the measurement update is performed. The newly formed Kalman gain from Equation (2.42) determines how much emphasis will be placed on the dynamics model versus the measurement. The measurement update is

$$\hat{\boldsymbol{x}}^{n+1}_{k|k} = \hat{\boldsymbol{x}}^n_{k|k} + \boldsymbol{K}^n_k[\mathbf{z}^n_k - \boldsymbol{h}[\hat{\boldsymbol{x}}^n_{k|k}]] \tag{2.44}$$

5. If Equation (2.39) is not satisfied, the SFRA EKF iterates again until the filter has moved the state estimate $\hat{\boldsymbol{x}}^n_{k|k}$ sufficiently towards the measurement cluster. Once Equation (2.39) is satisfied or the maximum number of N iterations is met, the result becomes the iterated *aposteriori* state estimate, $\hat{\boldsymbol{x}}^+_{k|k}$, and the SFRA EKF moves on to the next time step, t_{k+1} and the process starts again.

$$\boldsymbol{x}^{n+1}_{k|k} \rightarrow \boldsymbol{x}^+_{k|k} \tag{2.45}$$

$$t_k \rightarrow t_{k+1} \tag{2.46}$$

The SFRA EKF moves the state estimate towards the measurement in small steps until the residuals are small enough to continue. This effectively reduces the Q/R ratio (placing more emphasis on the centroid of the measurement cluster) when the measurement centroid is far from the predicted measurement. This achieves a

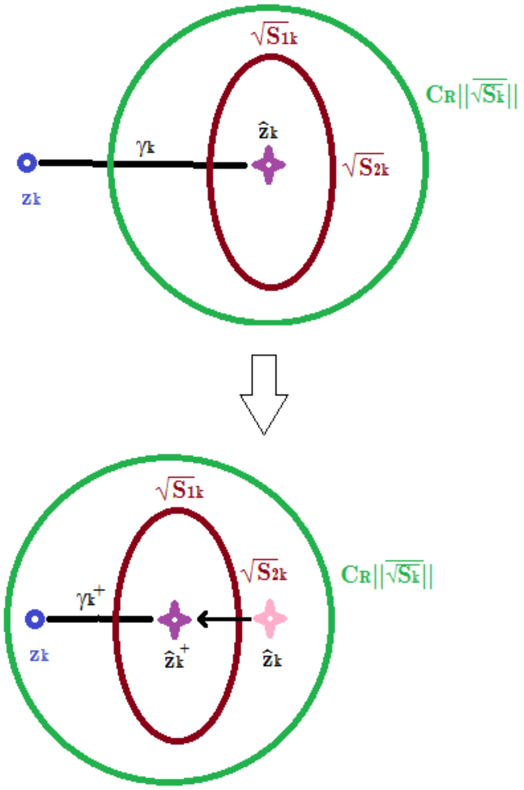


Figure 2.1: SFRA Filter with Residual Monitoring

basic form of maneuver detection. The downside is that the filter does not perform well when $\boldsymbol{Q}$ or $\boldsymbol{R}$ are mismatched. Figure 2.1 shows the nominal measurement moving towards the actual measurement based on residual monitoring.

2.2.4 Synopsis of EKF versus UKF. The UKF is a relatively new approach to system state estimation. This new version of the Kalman filter has several key advantages over the tried-and-true EKF. The UKF relies on a novel nonlinear transform called the unscented transformation. The basic idea behind the unscented transformation is that "it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation" [4].

The EKF works well when systems are nearly linear over the update interval because it relies on successive first-order approximations of the system. The UKF works well even with nonlinear systems because it does not directly linearize the state estimate. Instead, sample points (called “sigma” points) are created to parameterize the mean and covariance values of the state variables. These sigma points are transformed using the unscented transform to new parameterized terms that are propagated and updated using a modified Kalman filter.

Like the EKF, the UKF requires that the noise sources be Gaussian. The UKF differs because it does not attempt to linearize the system. The UKF approximates a Gaussian pdf up to 2^{nd}-order terms and is arguably easier to implement. Typically, the EKF’s higher instability requires that process noise covariance be artificially increased to account for the linearization errors made by the filter.

2.2.5 Unscented Kalman Filter .

The UKF is a relatively new development and is not limited by the linear approximation issues of the EKF. The probability distribution of the state variables is still assumed to be Gaussian, but is specified using a set of carefully chosen sample points [4]. These “sigma” points do a much better job describing the true mean and covariance of the distribution. The basis of the UKF is the unscented transform [4]. This is used to perform a nonlinear transform of the sigma points into measurement space without losing all higher-order terms (as with the EKF). In fact, the UKF can capture the mean and covariance of the states accurately to the 3rd order (Taylor series expansion). These sigma points are propagated and updated along with the state estimate. The UKF does an excellent job handling nonlinearities with Gaussian-distributed variables but is slightly more difficult to implement than the EKF.

The estimated state and covariance are augmented with the mean and covariance of the process noise. A set of $2L+1$ sigma points is derived from the augmented state and covariance where L is the number of states [4]. These sigma points can be described as

$$\chi_0 = \bar{x} \tag{2.47}$$

$$\chi_i = \bar{x} + \left(\sqrt{(L+\lambda)P_x}\right)_i \; i = 1 \ldots L \tag{2.48}$$

$$\chi_i = \bar{x} - \left(\sqrt{(L+\lambda)P_x}\right)_{i-L} \; i = L+1 \ldots 2L \tag{2.49}$$

where the tuning parameter, λ was chosen to be commensurate with the recommendation of Julier and Uhlmann: $\lambda = \alpha^2(\kappa + L) - L = 0$. $\left(\sqrt{(L+\lambda)P_x}\right)_i$ is the i^{th} column of the matrix square root. This is calculated using the Cholesky decomposition.

The weighting values for the mean and covariance are [4]

$$w_0^{(m)} = \frac{\lambda}{L+\lambda} \tag{2.50}$$

$$w_0^{(c)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \tag{2.51}$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(L+\lambda)} \tag{2.52}$$

where superscripts m and c denote mean and covariance respectively. The tuning parameters α and κ control the spread of the sigma points. For this research, tuning values are $\alpha = 10^{-3}$, $\kappa = 0$. β is related to the distribution of x. A value of $\beta = 2$ is used because the state vector PDF is assumed to be Gaussian.

The UKF propagate equations are expressed as [4]

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2n_a} \boldsymbol{w}_i^{(m)} \boldsymbol{X}_{i,k|k-1}^{x}, \text{ where } \boldsymbol{X}_{i,k|k-1}^{x} = f[\boldsymbol{X}_{i,k-1|k-1}^{x}, \mathbf{u}_{k-1}, \boldsymbol{X}_{i,k-1|k-1}^{w}] \tag{2.53}$$

$$\boldsymbol{P}_{k|k-1}^{xx} = \sum_{i=0}^{2n_a} \boldsymbol{w}_i^{(c)} [\boldsymbol{X}_{i,k|k-1}^{x} - \hat{\mathbf{x}}_{i,k|k-1}][\boldsymbol{X}_{i,k|k-1}^{x} - \hat{\mathbf{x}}_{i,k|k-1}]^T \tag{2.54}$$

The propagation step uses the process model to predict the state vector at the next time step. The measurement prediction, $\hat{\boldsymbol{z}}_{k|k-1}$, and the residual covariance, $\boldsymbol{P}^{zz}$ are calculated according using the nonlinear observation function

$$\hat{\boldsymbol{z}}_{k|k-1} = \sum_{i=0}^{2n_a} \boldsymbol{w}_i^{(m)} \boldsymbol{Z}_{i,k|k-1}^{x}, \text{ where } \boldsymbol{Z}_{i,k|k-1}^{x} = \boldsymbol{h}[\boldsymbol{X}_{i,k|k-1}^{x}, \boldsymbol{X}_{i,k|k-1}^{r}] \tag{2.55}$$

and

$$\boldsymbol{P}_{k|k-1}^{zz} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} [\boldsymbol{Z}_{i,k|k-1}^{x} - \hat{\mathbf{z}}_{i,k|k-1}][\boldsymbol{Z}_{i,k|k-1}^{x} - \hat{\mathbf{z}}_{i,k|k-1}]^T \tag{2.56}$$

The cross correlation of the measurement and state vector, $\boldsymbol{P}^{xz}$, is calculated using

$$\boldsymbol{P}_{k|k-1}^{xz} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} [\boldsymbol{X}_{i,k|k-1}^{x} - \hat{\mathbf{x}}_{i,k|k-1}][\boldsymbol{Z}_{i,k|k-1}^{x} - \hat{\mathbf{z}}_{i,k|k-1}]^T \tag{2.57}$$

The Kalman gain, K, is now expressed in terms of residual covariance, P^{zz} and measurement-state cross correlation P^{xz} according to

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}^{xz}[\boldsymbol{P}_{k|k-1}^{zz}]^{-1} \tag{2.58}$$

The last step is the measurement update. The newly formed Kalman gain from Equation (2.58) determines how much emphasis will be placed on the dynamics model versus the measurement. The measurement update is

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k[\mathbf{z}_k - \boldsymbol{h}[\hat{x}_{k|k-1}]] \tag{2.59}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{P}_{k|k-1}^{zz} \boldsymbol{K}_k^T \tag{2.60}$$

2.2.6 Iterated Unscented Kalman Filter . The IUKF functions similarly to the UKF with the exception of multiple iterations of Equations (2.61) through (2.62) until $\boldsymbol{x}_k^{n+1} - \boldsymbol{x}_k^{n}$ is sufficiently small. The goal of the IUKF is to iteratively reevaluate,

$\boldsymbol{h}[.]$ to improve estimation quality. It should be noted that this implementation is unstable because the regeneration of sigma points about the iterated $h[.]$ does not guarantee convergence. Often, the algorithm will reach the maximum number of iterations without converging on a satisfactory state estimate. This explanation is included for the sake of completeness.

1. The IUKF performs the first iteration of each time step, t_k, the same as the basic UKF Equations (2.47) through (2.60). The IUKF performs iteration $n = 1, 2 \ldots N$ on

   $$\hat{\boldsymbol{z}}^n_{k|k-1} = \sum_{i=0}^{2n_a} \boldsymbol{w}_i^{(m)} \boldsymbol{Z}^{xn}_{i,k|k-1}, \text{ where } \boldsymbol{Z}^{xn}_{i,k|k-1} = \boldsymbol{h}[\boldsymbol{X}^{xn}_{i,k|k-1}, \boldsymbol{X}^{rn}_{i,k|k-1}] \tag{2.61}$$

   and

   $$\hat{\boldsymbol{x}}^{n+1}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k[\mathbf{z}_k - \boldsymbol{h}[\hat{x}^n_k]] \tag{2.62}$$

   where the initial $\hat{\boldsymbol{x}}^n_k$ is the first *aposteriori* estimate, $\hat{\boldsymbol{x}}_{k|k}$.

2. Once the newly iterated *aposteriori* state estimate $\hat{\boldsymbol{x}}^{n+1}_k$ is obtained, it is compared to the previous iteration, $\hat{\boldsymbol{x}}^n_k$ by

   $$\hat{\boldsymbol{x}}^{n+1}_k - \hat{\boldsymbol{x}}^n_k < \boldsymbol{C}_I \tag{2.63}$$

   where $\boldsymbol{C}_I$ is a user-define threshold.

3. If Equation (2.63) is not satisfied, the filter stays on the current time step t_k and reevaluates $\boldsymbol{h}[\cdot]$ to achieve a better $\hat{\boldsymbol{x}}^{n+1}_k$.

   $$\hat{\boldsymbol{x}}^{n+1}_k \rightarrow \hat{\boldsymbol{x}}^n_k \tag{2.64}$$

   $$\boldsymbol{Z}^{xn}_{i,k|k-1} = \boldsymbol{h}[\boldsymbol{X}^{xn}_{i,k|k-1}, \boldsymbol{X}^{rn}_{i,k|k-1}] \tag{2.65}$$

The filter iterates on $\boldsymbol{h}[\cdot]$ until Equation (2.63) is satisfied. Afterwards, the IUKF moves on to the next time step, t_{k+1} and the process starts again. Once Equation (2.63) is satisfied or the maximum number of N iterations is met, the result becomes the iterated *aposteriori* state estimate, $\hat{\boldsymbol{x}}^{+}_{k|k}$, and the IUKF moves on to the next time step, t_{k+1} and the process starts again.

$$\hat{\boldsymbol{x}}^{n+1}_{k|k} \rightarrow \hat{\boldsymbol{x}}^{+}_{k|k}, t_k \rightarrow t_{k+1} \tag{2.66}$$

The IUKF algorithm is suboptimal and also unstable because the regeneration of sigma points and transformation through the iterated measurement function does not always converge. *It is not recommended to use this algorithm.* These drawbacks are explained further in Section IV.

2.2.7 Single Filter Reactive Adaptation Unscented Kalman Filter . The SFRA UKF does not guarantee optimality but adds the flexibility of maneuver detection using residual monitoring with the accuracy of the Unscented Transform. The SFRA UKF functions very similarly to the UKF with the exception of multiple iterations of Equations (2.47) through (2.60) without repeating propagation Equations (2.53) and (2.54). In fact, the first iteration of the SFRA UKF is identical to the UKF. The SFRA UKF monitors the mean of the filter residuals which are compared to a threshold value set by the filter designer.

The SFRA UKF performs the first iteration of each time step, t_k identically to UKF Equations (2.47) through (2.60). Once the first measurement residuals are obtained,

1. Test the 2-norm of i residuals *(i = 1,2,...,M)* versus a threshold based on the norm of the square root of the corresponding variance $\boldsymbol{S_k}$ multipled by a tuning parameter $\boldsymbol{C}_R$,

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k \tag{2.67}$$

$$||\boldsymbol{\gamma}_{[1:M]}|| < \boldsymbol{C}_R||\sqrt{\boldsymbol{S}_k}|| \tag{2.68}$$

where $\sqrt{\boldsymbol{S}_k}$ is obtained using the Cholesky decomposition.

2. If Equation (2.68) is not satisfied, do not proceed past the current time step, t_k. Instead, shift the *aposteriori* estimate, $\boldsymbol{x}^n_{k|k}$, towards the measurement.

3. Re-run the algorithm, skipping the UKF propagation steps. Generate $2L+1$ new sigma points about the new *apriori* estimate, $\boldsymbol{x}^n_{k|k-1}$
4. Recompute the measurement process using the new *apriori* estimate, $\boldsymbol{x}^n_{k|k-1}$

$$\hat{\boldsymbol{z}}^n_{k|k-1} = \sum_{i=0}^{2n_a} \boldsymbol{w}_i^{(m)} \boldsymbol{Z}^{xn}_{i,k|k-1}, \text{ where } \boldsymbol{Z}^{xn}_{i,k|k-1} = \boldsymbol{h}[\boldsymbol{X}^{xn}_{i,k|k-1}, \boldsymbol{X}^{rn}_{i,k|k-1}] \tag{2.69}$$

$$\boldsymbol{P}^{zzn}_{k|k-1} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} [\boldsymbol{Z}^{xn}_{i,k|k-1} - \hat{\mathbf{z}}^n_{i,k|k-1}][\boldsymbol{Z}^{xn}_{i,k|k-1} - \hat{\mathbf{z}}^n_{i,k|k-1}]^T \tag{2.70}$$

5. Recalculate the cross-correlation of the measurement and state vector, $\boldsymbol{P}^{xz}$, using

$$\boldsymbol{P}^{xzn}_{k|k-1} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} [\boldsymbol{X}^{xn}_{i,k|k-1} - \hat{\mathbf{x}}^n_{i,k|k-1}][\boldsymbol{Z}^{xn}_{i,k|k-1} - \hat{\mathbf{z}}^n_{i,k|k-1}]^T \tag{2.71}$$

6. The new Kalman gain, $\boldsymbol{K}^n$, is now expressed in terms of residual covariance, $\boldsymbol{P}^{zz}$ and measurement-state cross correlation P^{xz} according to

$$\boldsymbol{K}^n_k = \boldsymbol{P}^{xzn}{}_{k|k-1}[\boldsymbol{P}^{zzn}{}_{k|k-1}]^{-1} \tag{2.72}$$

7. Next, the measurement update is performed. The newly formed Kalman gain from Equation (2.72) determines how much emphasis will be placed on the dynamics model versus the measurement. The measurement update is described

according to

$$\hat{\boldsymbol{x}}_{k|k}^{n+1} = \hat{\boldsymbol{x}}_{k|k}^{n} + \boldsymbol{K}_k^n[\mathbf{z}_k^n - \boldsymbol{h}[\hat{\boldsymbol{x}}_{k|k}^n]] \tag{2.73}$$

8. The residuals $\boldsymbol{\gamma}$ are then calculated with the goal of satisfying Equation (2.68). If not satisfied, the filter iterates again until the filter has shifted the state estimate $\hat{\boldsymbol{x}}_{k|k^n}$ within a short distance of the measurement cluster centroid. Once Equation (2.68) is satisfied or the maximum number of N iterations is met, the result becomes the iterated *aposteriori* state estimate, $\hat{\boldsymbol{x}}_{k|k}^{+}$, and the SFRA UKF moves on to the next time step, t_{k+1} and the process starts again.

$$\boldsymbol{x}_{k|k}^{n+1} \rightarrow \boldsymbol{x}_{k|k}^{+} \tag{2.74}$$

$$t_k \rightarrow t_{k+1} \tag{2.75}$$

The SFRA UKF moves the state estimate towards the measurement in small steps until the residuals are small enough to continue. This effectively reduces the Q/R ratio (placing more emphasis on the centroid of the measurement cluster) when the measurement centroid is sufficiently far from the predicted measurement. This achieves a basic form of maneuver detection. Figure 2.1 shows the nominal measurement moving towards the actual measurement based on residual monitoring. The downside of the SFRA algorithm is reduced accuracy compared to the standard UKF when $\boldsymbol{Q}$ or $\boldsymbol{R}$ are mismatched.

2.3 Kalman Smoother Development

A smoother is a combination of two Kalman filters. One propagates forward in time and the other propagates backwards. In general, this results in a better trajectory estimate because the smoother limits state covariance propagation between measurements [7]. The smoother errors are typically less than both the forward and backward filter. The main limitation of this approach is that it cannot be performed in real time.

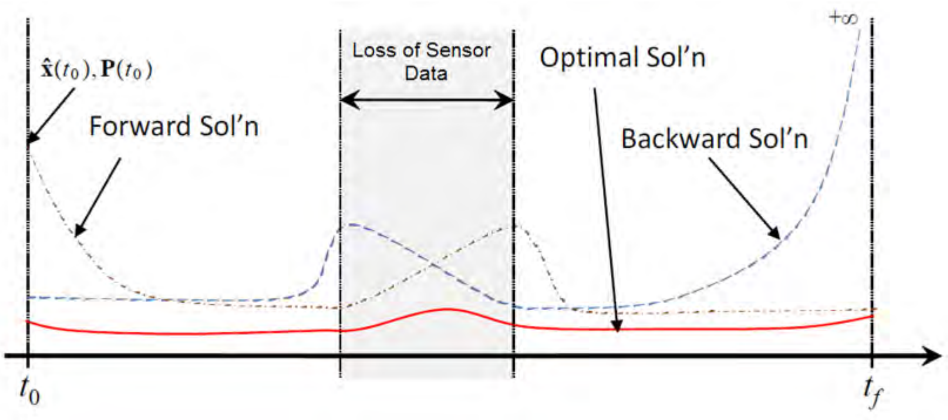


Figure 2.2: Example of a Fixed Interval Smoother

There are several types of optimal smoothers, not limited to the following: fixed interval, fixed point, and fixed lag. A fixed interval smoother provides the optimal estimate of $\hat{x}_{k|n}(k < n)$ using measurements from a fixed interval defined by z_0 to z_n. A fixed point smoother is similar to the fixed interval, but provides an estimate of $\hat{\boldsymbol{x}}_n$ using measurements from an interval defined from $\boldsymbol{z}_i$ to $\boldsymbol{z}_n$, where i is a fixed value. The fixed lag smoother provides an optimal estimate of $\hat{\boldsymbol{x}}_{k-N|k}$ based on N previous steps. Since the live fire missile data is being analyzed post-test, the fixed interval smoother will be used to take advantage of all available measurements, $\boldsymbol{z}_0$ to $\boldsymbol{z}_f$. Figure 2.2 shows an example of a fixed interval smoother.

2.3.1 Extended Kalman Smoother. The Extended Kalman Smoother (EKS) is a variation which combines two Extended Kalman Filters (EKF). This type of smoother was chosen because the EKF is well-understood and provides a meaningful baseline for comparison. The EKS optimally combines the results of a forward and a backward EKF. The forward EKF estimates mean and covariance ($\hat{\boldsymbol{x}}^f_{k|k}, \boldsymbol{P}^f_{k|k}$) given all $\boldsymbol{z}_i$ for $i \leq k$ and the backward EKF estimates ($\hat{\boldsymbol{x}}^b_{k|k+1}, \boldsymbol{P}^b_{k|k+1}$) given all $\boldsymbol{z}_i$ for $i > k$. The conversion from forward EKF to backward EKF dynamics model requires only an inversion of input and output data coupled with using the reverse of the dynamics model

$$\dot{\boldsymbol{x}}_{-t} = -\boldsymbol{f}[\boldsymbol{x}_{-t}, \boldsymbol{u}_{-t}, \boldsymbol{w}_{-t}] \tag{2.76}$$

which results in the transpose of the state transition matrix

$$\boldsymbol{\phi}_{-k} = \boldsymbol{\phi}^T_k. \tag{2.77}$$

The measurement model is also inverted according to

$$\boldsymbol{y}_{-t} = \boldsymbol{h}[\boldsymbol{x}_{-t}, \boldsymbol{r}_{-t}]. \tag{2.78}$$

After running two EKF passes (forward and backward) through the data, the EKS is formed by optimally combining the results of each filter. The combination of the backward and forward filters is a smoothed mean and covariance ($\hat{\boldsymbol{x}}^s_k, \boldsymbol{P}^s_k$) given by [7]

$$[\boldsymbol{P}^s_k]^{-1} = [\boldsymbol{P}^f_{k|k}]^{-1} + [\boldsymbol{P}^b_{k|k+1}]^{-1} \tag{2.79}$$

$$\hat{\boldsymbol{x}}^s_k = \boldsymbol{P}^s_k\Big[[\boldsymbol{P}^f_{k|k}]^{-1}\hat{\boldsymbol{x}}^f_{k|k} + [\boldsymbol{P}^b_{k|k+1}]^{-1}\hat{\boldsymbol{x}}^b_{k|k+1}\Big]. \tag{2.80}$$

2.3.2 Iterated Extended Kalman Smoother. The Iterated Extended Kalman Smoother (IEKS) is formed by combining a forward and a reverse Iterated Extended Kalman Filter (IEKF). The backwards IEKF dynamics model is obtained by calculating the transpose of the transition matrix according to Equation 2.77. The backwards measurement model is inverted according to Equation (2.78). The process of combining to the two is identical to the EKS according to Equations (2.79) through (2.80). Mathematically, it appears the IEKS will give marginal improvement over the EKS.

2.3.3 Single Filter Reactive Adaptation Extended Kalman Smoother. The Single Filter Reactive Adaptation Extended Kalman Smoother (SFRA EKS) is formed by combining a forward and a reverse SFRA Extended Kalman Filter (SFRA EKF). The backwards SFRA EKF dynamics model is obtained by calculating the transpose of the transition matrix according to Equation (2.77). The backwards measurement model is inverted according to Equation (2.78). The process of combining to the two is identical to the EKS according to Equations (2.79) through (2.80).

2.3.4 Unscented Kalman Smoother. The Unscented Kalman Smoother (UKS) is a variation which combines two Unscented Kalman Filters (UKF). This type of smoother was chosen because the UKF can provide a quick and accurate (2^{nd}-order) system approximation. The UKS optimally combines the results of a forward and a backward UKF. The forward UKF estimates mean and covariance ($\hat{\boldsymbol{x}}^f_{k|k}, \boldsymbol{P}^f_{k|k}$) given all $\boldsymbol{z}_i$ for $i \leq k$ and the backward UKF estimates ($\hat{\boldsymbol{x}}^b_{k|k+1}, \boldsymbol{P}^b_{k|k+1}$) given all $\boldsymbol{z}_i$ for $i > k$. The conversion from forward UKF to backward UKF dynamics model requires only an inversion of input and output data coupled with using the reverse of the dynamics model [7]

$$\dot{\boldsymbol{x}}_{-t} = -\boldsymbol{f}[\boldsymbol{x}_{-t}, \boldsymbol{u}_{-t}, \boldsymbol{w}_{-t}] \tag{2.81}$$

$$\boldsymbol{y}_{-t} = \boldsymbol{h}[\boldsymbol{x}_{-t}, \boldsymbol{r}_{-t}] \tag{2.82}$$

.

After running two UKF passes (forward and backward) through the data, the UKS is formed by optimally combining the results of each filter. The combination of the backward and forward filters is a smoothed mean and covariance $(\hat{\boldsymbol{x}}_k^s, \boldsymbol{P}_k^s)$ given by [13]

$$[\boldsymbol{P}_k^s]^{-1} = [\boldsymbol{P}_{k|k}^f]^{-1} + [\boldsymbol{P}_{k|k+1}^b]^{-1} \tag{2.83}$$

$$\hat{\boldsymbol{x}}_k^s = \boldsymbol{P}_k^s[[\boldsymbol{P}_{k|k}^f]^{-1}\hat{\boldsymbol{x}}_{k|k}^f + [\boldsymbol{P}_{k|k+1}^b]^{-1}\hat{\boldsymbol{x}}_{k|k+1}^b]. \tag{2.84}$$

2.3.5 Iterated Unscented Kalman Smoother. The Iterated Unscented Kalman Smoother (IUKS) is formed by combining a forward and a reverse Iterated Unscented Kalman Filter (IUKF). The backwards IUKF dynamics model is obtained by inverting the dynamics model according to Equation (3.8). The backwards measurement model is inverted according to Equation (3.9). The IUKS is formed by combining a forward and a reverse IUKF according to Equations (3.10) and (3.11). This algorithm is suboptimal and also unstable because the regeneration of sigma points about the iterated measurement function does not always converge. This explanation is included for completeness and it is not recommended to use this algorithm! These drawbacks are explained further in Section IV.

2.3.6 Single Filter Reactive Adaptation Unscented Kalman Smoother. The Single Filter Reactive Adaptation Unscented Kalman Smoother (SFRA UKS) is formed by combining a forward and a backward SFRA Unscented Kalman Filter (SFRA UKF). The backwards SFRA UKF dynamics model is obtained by inverting the dynamics model according to Equation (3.8). The backwards measurement model is inverted according to Equation (3.9). The SFRA UKS is formed by combining a forward and a reverse SFRA UKF according to Equations (3.10) and (3.11).

2.4 Measurement Environment

This section contains the methodology for taking measurements using the proposed FMCW sensor array located on the QF-16. The sensor locations are fixed in relation to the aircraft fuselage and are expressed in the body frame. These aircraft body frame coordinates are transformed to an Earth-fixed North-East-Down (NED) navigation frame so that observation calculations can be made. The aircraft and missile trajectories are expressed in the navigation frame. Accordingly, the Kalman filter maintains missile trajectory state estimates in this frame. The details of these reference frames are discussed in Section 2.4.1. Range and velocity calculations are based on multilateration described in Section 2.4.3 and trilateration in Section 2.4.4. Techniques used for measurement gating and data association are discussed in Section 2.4.5.

2.4.1 Reference Frames. The following reference frames are used in this research [14]:

- Body frame (b-frame)
- Vehicle-fixed navigation frame (n'-frame)
- Earth-fixed navigation frame (n-frame)
- Earth frame (e-frame)
- Earth-fixed inertial frame (i-frame)

Figure 2.3 depicts the relationship between the Earth-rotating frame (e-frame) and the Earth-fixed inertial fame (i-frame). The i-frame provides an approximation of a truly inertial reference frame where Newton's laws of motion are valid. The origin is co-located with the center of the Earth and the axes are non-rotating with respect to fixed stars. The x and y axes form the equatorial plane and the z-axis is co-located with the Earth's polar axis. Since this model ignores the Earth's revolution around the Sun, it is not truly inertial. The e-frame differs from i-frame only in that the x and y axes rotate along with the Earth.

The Eath-fixed navigation frame (n-frame) is used for navigation with respect to a geographic point on the Earth. As seen in Figure 2.4, the x, y, and z-axis point in the North, East, and Down (NED) directions with respect to a particular point on the surface of the globe. The down direction is defined by the local gravity vector. The vehicle-fixed navigation frame (n'-frame) is projected onto the Earth with the origin typically chosen to be a fixed point with respect the vehicle.

The origin of the body frame (b-frame) is located at a fixed point with respect to the body of a vehicle. When defining a b-frame, the vehicle's center of gravity (C.G.) is typically chosen as the origin.

2.4.2 Coordinate Transformations. Coordinate transformation are used as a convenient means of transforming a vector between two reference frames. The multitude of navigation frames would be relatively useless without a way to move between them. Direction cosine matrices (DCM) are used in this research to express a vector in different coordinate frames. DCM's can be described by

$$\boldsymbol{r}^b = \boldsymbol{C}_a^b \boldsymbol{r}^a \tag{2.85}$$

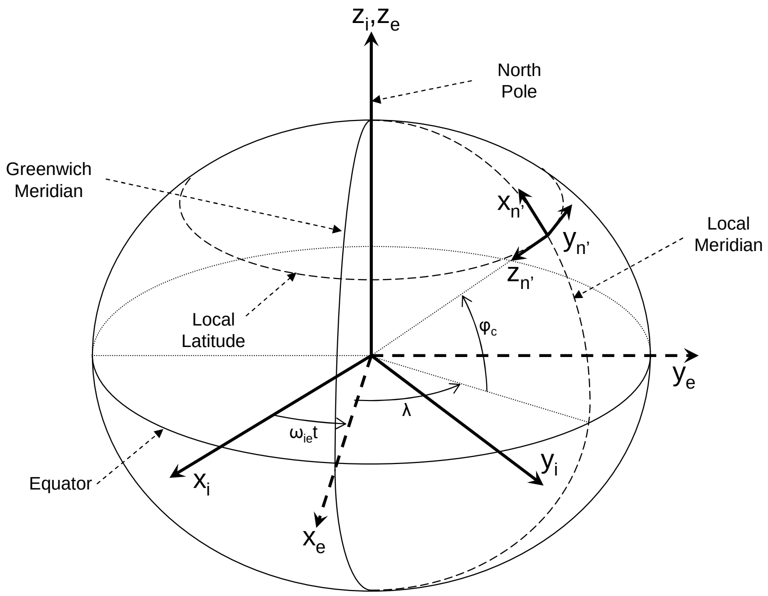


Figure 2.3: Earth-Fixed Reference Frames

where $\boldsymbol{r}^a$ is a vector expressed in an arbitrary frame a, $\boldsymbol{r}^b$ is the same vector expressed in frame b and $\boldsymbol{C}_a^b$ is the DCM.

Euler angles provide a method for deriving the DCM to transform from one-coordinate system to another by performing a series of three rotations about different axes [14]. Rotations of ϕ about the x-axis, θ about the y-axis, and ψ about the z-axis are expressed mathematically by the DCMs

$$\boldsymbol{C}_1 = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.86}$$

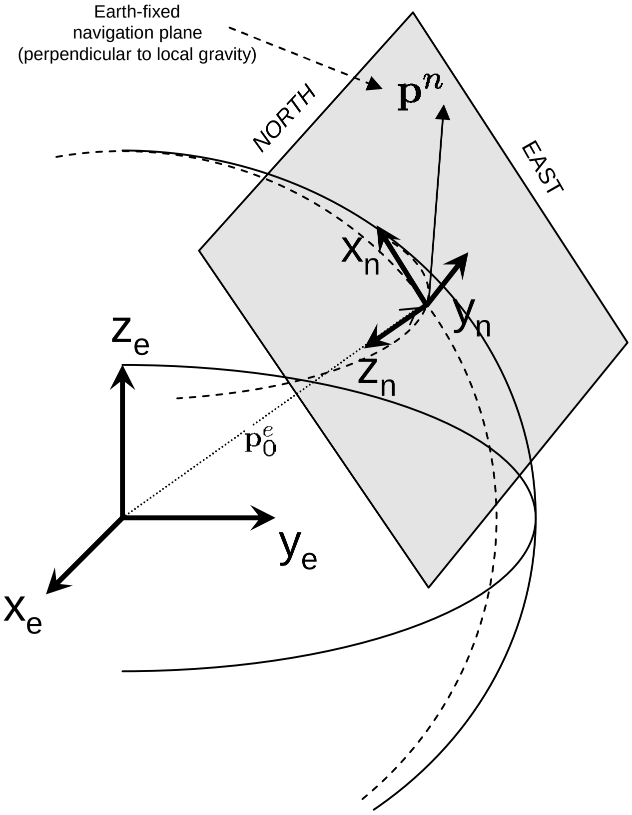


Figure 2.4: Earth-Fixed Navigation Reference Frame

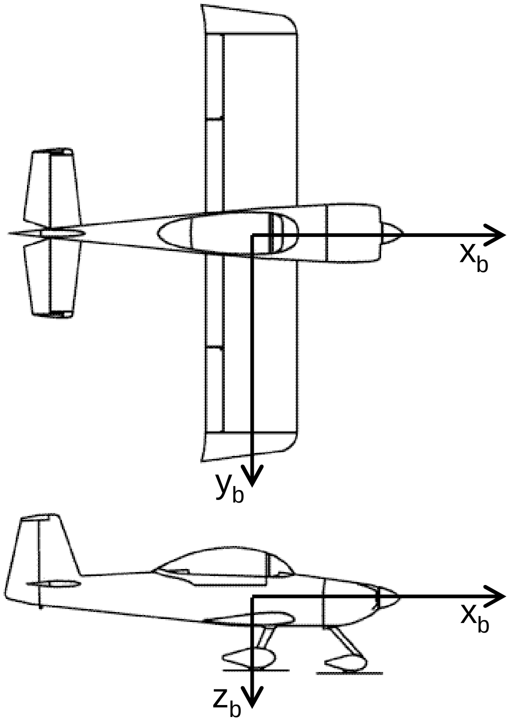


Figure 2.5: Aircraft Body Reference Frame

$$\boldsymbol{C}_2 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.87}$$

$$\boldsymbol{C}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{2.88}$$

When performing a transformation from a navigation frame to an aircraft body frame, the angle ψ represents the angle between the nose of the aircraft and north. Similarly, the angles θ and ϕ represent the pitch and roll of the aircraft, respectively.. The product of these DCMs yields a transformation from the reference frame to the body frame according to

$$\boldsymbol{C}_n^b = \boldsymbol{C}_3 \boldsymbol{C}_2 \boldsymbol{C}_1 \tag{2.89}$$

Using this DCM a vector $\boldsymbol{r}^b$, defined in the body axes is transformed into the navigation reference frame by

$$\boldsymbol{r}^n = \boldsymbol{C}_b^n \boldsymbol{r}^b \tag{2.90}$$

Reversing transformation direction can be performed by using the transpose of the original DCM $\boldsymbol{C}_n^b$ (i.e., $\boldsymbol{C}_b^n = (\boldsymbol{C}_n^b)^T$).

2.4.3 Multilateration . Sensor range measurements can be combined to produce an estimate of missile position via trilateration. Figure 2.6 [11] shows a 2D example where three separate sensors, P1 - P3, are used to estimate the position of a target at point B. A single range measurement from P1 constrains the target's position to a circle with radius r1 centered at P1. If two range measurements are obtained at the same time, the target's position can be constrained to the intersection of two circles, with radii of r1 and r2 uniquely centered at P1 and P2, respectively. These potential target locations are labeled points A and B. If three range measurements are obtained simultaneously, the target's location becomes unambiguous and is constrained to point B. If any sensor happens to be co-linear with another sensor with respect to the target, its range measurement will not help uniquely identify the target position.

This work uses simulated range measurements to reconstruct a missile's trajectory in 3D space. In this case, a single range measurement specifies a sphere of potential positions. A second simultaneous range measurement from a uniquely located sensor will constrain the target position to the intersection of two spheres (i.e., a circle). A range measurement from third sensor will further constrain the target position to an intersection of two circles (i.e., a point). Similar to the 2D example, a fourth sensor is required to uniquely identify the target location in 3D.

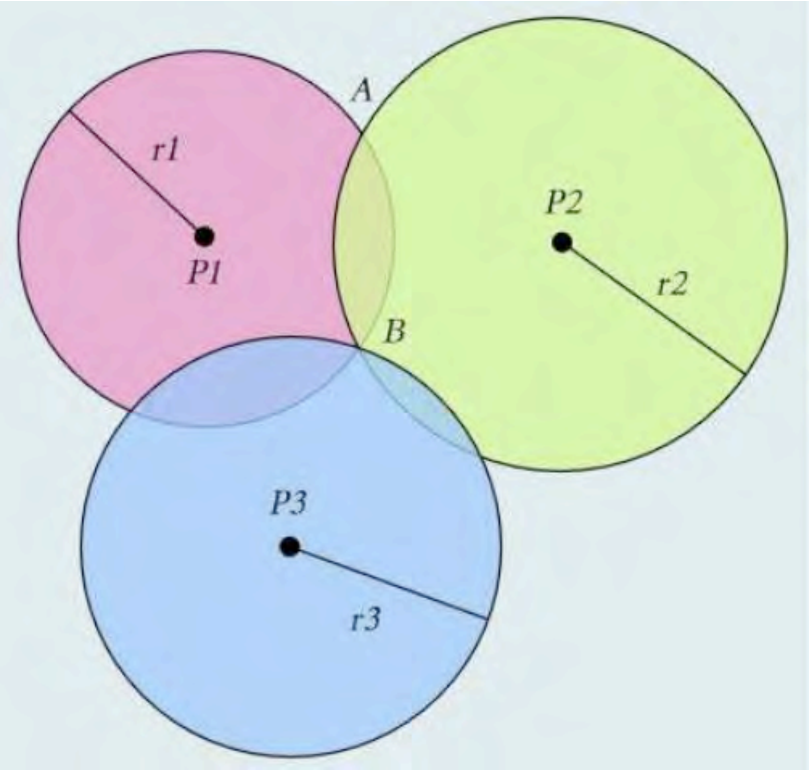


Figure 2.6: 2D Trilateration

In the real world, no sensor is able to obtain perfect measurements. If perfect measurements were available, there would be no utility in using more than the minimum number of measurements required to uniquely determine a target's position. Since sensors exhibit random noise, they are modeled as such in this research. Since no measurement is perfect, it is beneficial to incorporate measurements from more than the minimum number of sensors. This process assumes each sensor measurement is unbiased and applies a least-squares error estimation to calculate the target location [11].

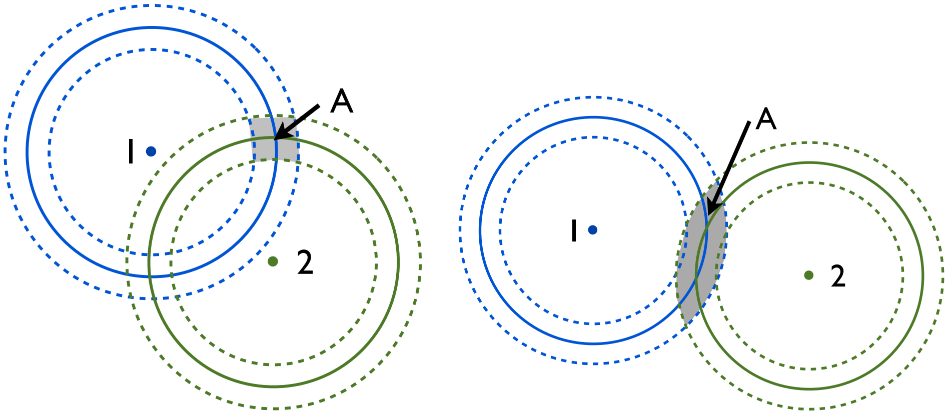
Poor sensor geometry can cause large errors when performing position calculations using imperfect measurements. The ideal 2D sensor geometry is angular separation of 90 degrees in the sensor plane. In general, optimal sensor geometry can be described by

$$\theta = \frac{360}{N_{sens}}^{\circ} \tag{2.91}$$

where N_{sens} is the number of sensors.

Figure 2.7 shows how poor sensor geometry can produce large position estimation errors in 2D [11]. The solid circles indicate the range measurements and the dashed circles describe the associated measurement uncertainty. The shaded areas indicate the size and shape of the regions likely containing the target's true position. Figure 7(a) shows optimal geometry with a minimized shaded "uncertainty" region. The suboptimal case shown in Figure 7(b) provides a visual representation of increased position uncertainty based solely on poor sensor geometry.

2.4.4 Velocity Vector Calculations using Range-Rate Measurements . Range-rate measurements from different sensors can be used to calculate a target's velocity vector in a manner very similar to multilateration. Figure 2.8 is a 2D example of this process [9]. The two range-rate sensors, $S1$ and $S2$, can provide magnitude measurements of the target's radial velocity (i.e., speed). The target's measured velocity vectors are shown by v_1 and v_2. Assuming the positions of the target and sensors are known, these vectors can be combined to form a velocity vector estimate, v. Sensor



(a) Optimal Geometry (b) Poor Geometry

Figure 2.7: Impact of Sensor Geometry on Precision of Position Calculation

geometry requirements and limitations are the same as those in multilateration. A minimum of three sensors is required in 3D to reconstruct the target velocity vector. No two sensors may be co-linear in LOS to the target and all three cannot be co-planar with the target.

2.4.5 Measurement Gating and Data Association. For the sake of completion, a synopsis of the gating and data association process is included. The measurement gating and data association process is virtually unchanged from the existing strategy completed in previous work [11]. The sensor measurements are created using additive, white Gaussian noise and also introducing false "clutter" measurements to the *Argos*-generated missile trajectory discussed in Section 4.3.

The goal of measurement gating is to eliminate sensor observations far away from the predicted trajectory. This research uses a two-stage gating process to efficiently remove unlikely observations. As described by [2], square gating uses the maximum eigenvalue of the residual covariance $\boldsymbol{S}_k$ which is scaled by a sizing factor γ and can be expressed as

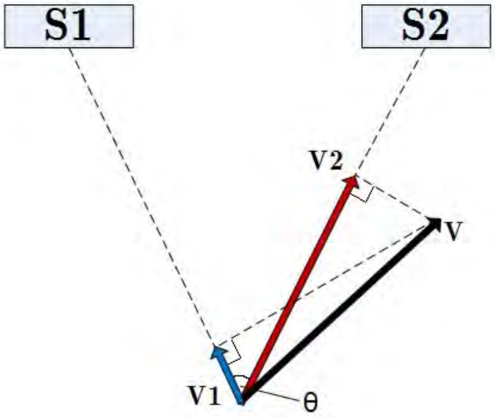


Figure 2.8: Calculation of 2D Velocity from Speed Measurements

$$e\,max = \sqrt{\mathsf{max}(\mathsf{eig}(\gamma \boldsymbol{S}_k))} \tag{2.92}$$

The gating is sized such that clutter measurements are effectively eliminated. According to [2] and [11], choosing $\gamma = 9.2$ provides a 99 percent chance that the true observation is within the gate [2]. The residual covariance is calculated from the *apriori* state covariance, $\boldsymbol{P}_k^-$, and the measurement model parameters using the formula [2]

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k \tag{2.93}$$

After computing $e\,max$, each measurement, $\boldsymbol{z}_j$, is compared to the expected target measurement, $\hat{\boldsymbol{z}}$, using the formula [2]

$$\hat{\boldsymbol{z}} - e\,max \leq \boldsymbol{z}_j \leq \hat{\boldsymbol{z}} + e\,max \tag{2.94}$$

The subscript j refers to the j^{th} measurement from the sensor. Every measurement outside this region is eliminated as a possible candidate for updating the target.

The first stage is relatively coarse because it uses a worst-case, one-dimensional approximation based on the residual covariance. The second stage refines the measurement observations using ellipsoidal gating. The residual norm, d_j^2, of each measurement is compared to the gate size. If the residual norm is larger than the gate size, $\gamma < d_j^2$, the measurement is eliminated from inclusion as a track update. If multiple observations occur with residual norms within the gate size, the one with the smallest residual norm is chosen to update the track.

Data association is a complicated topic with many theorized suboptimal and optimal approaches. Much has been written about solving the data association problem using both hard and soft data assignments. Hard assignment algorithms typically allow for only one possible data assignment for each measurement. Soft assignment algorithms are typically recursive and assign probabilities to each data assignment hypothesis. The scope of this research is limited to a simple hard data assignment algorithm known as global nearest neighbor (GNN). The closest observation is assessed by comparing the residual norm, d_j^2, for each observation which survives the gating process. The residual norm is defined as

$$\boldsymbol{d}_j^2 = \boldsymbol{\gamma}_k^T \boldsymbol{S}_k^- 1 \boldsymbol{\gamma}_k. \tag{2.95}$$

where $\boldsymbol{\gamma}_k$ is the difference between the actual and predicted measurements.

$$\gamma_k = z_k - \hat{z}_k. \tag{2.96}$$

2.5 Summary

This chapter covered the mathematical basis required to build and implement six Kalman smoothers (EKS, IEKS, SFRA EKS, UKS, IUKS, SFRA UKS). The UKF provides a convenient way to gain 2^{nd} order estimation capability. The EKF works well when systems are nearly linear over the update interval because it relies on successive first-order approximations of the system. The UKF works well even with nonlinear systems because it does not directly linearize the state estimate. In both cases, the benefits of coupling a dynamics model with measurement data are impressive. These benefits are maximized in post-processing by the application of a fixed-interval smoother. A smoother combines forward and backward KF's to generate an optimal state estimate. These algorithms provide trade offs between computational load and estimation performance. After establishing the basis for these Kalman smoothers, a basic measurement environment with gating and data association was defined. Chapter III discusses previous research on this topic and provides insight for the motivation to develop these Kalman smoothers.

III. Past Research

3.1 Overview

Missile tracking is a rich topic which lends itself to the application of Kalman filters and data smoothing. This literature review begins with the introduction of the UKF by Julier and Uhlmann [4] for system state estimation. This transform presents advantages over direct linearization of system equations and forms the basis of this work. Teixeira et al present a method for reconstructing the flight path of a sailplane aircraft using a UKF and UKS with the overall goal of estimating sensor biases [13]. The smoother implementation was used as a model for UKS implementation in this thesis. Roumeliotis et al present a Kalman smoother based localization algorithm for a mobile robot using periodic sensor updates with a loosely-coupled INS [10]. Sweeney presents a novel architecture for the generation and reconstruction of air-to-air missile trajectories using Kalman filters [11]. His work forms the baseline effort for this research.

3.2 A New Extension of the Kalman Filter to Nonlinear Systems

Julier and Uhlmann introduced the Unscented Kalman filter (UKF) as a new technique for system state estimation [4]. This new version of the Kalman filter has several key advantages over the tried-and-true Extended Kalman filter (EKF). The UKF relies on a novel nonlinear transform called the unscented transformation. The basic idea behind the unscented transformation is that "it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation" [4]. The EKF works well when systems are nearly linear over the update interval because it relies on successive first-order approximations of the system. The UKF does not directly linearize the system. Instead, sample points (called sigma points) are created to parameterize the mean and covariance values of the state variables. These sigma points are transformed using the unscented transform to new parameterized terms that are propagated and updated using a modified Kalman filter. Unlike the EKF, the UKF does not require that the noise sources be Gaussian and also does not attempt to linearize the system. The authors argue that UKF performance is equivalent to that of a 2^{nd}-order Gauss filter and is easier to implement. The authors show that the less stable EKF requires that the process noise covariance be artificially increased to account for the linearization errors made by the filter.

The authors argue that the Jacobian matrix calculation required in the EKF for a first-order linear approximation is difficult to implement in most systems. The other cited drawback of the EKF is the instability caused by a poorly executed linear approximation of the system. A pictorial of the unscented transformation is shown in Figure 3.1.

Since the distribution of $\mathbf{x}$ is approximated to the second order and transformed without linearization, the distribution of $\mathbf{y}$ is also known to this level of accuracy. The UKF requires no linearization of the system. In comparison, the EKF linearizes the system and assumes all noise sources are Gaussian. Since the UKF does not approximate the system, many higher order effects are preserved by the estimator. The authors show how the UKF can be made to perform just as well as a 2^{nd}-order Gauss filter without the need to calculate Jacobians or Hessians. Sigma points can be generated using

$$\mathscr{X}_0 = \hat{\boldsymbol{x}}_k^- \tag{3.1}$$

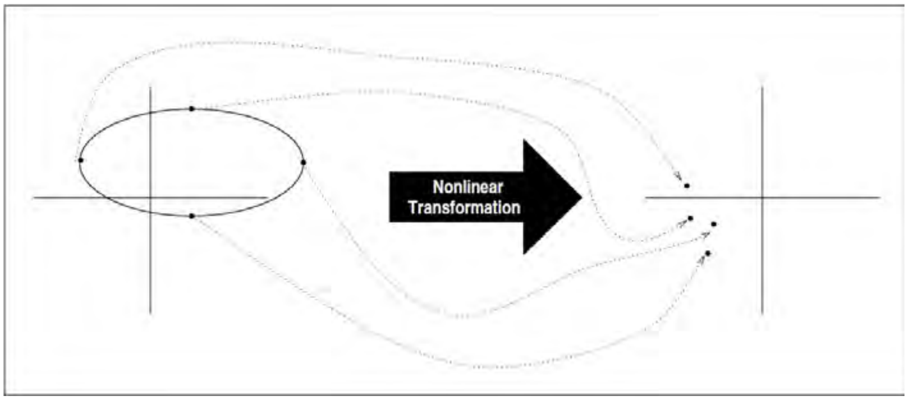


Figure 3.1: Principle of the Unscented Transform [4]

$$\mathscr{X}_i = \hat{\boldsymbol{x}}_k^- + \sqrt[C]{(\lambda + L)\boldsymbol{P}_{x_k}^-}|_i \quad i = 1, 2...L \tag{3.2}$$

$$\mathscr{X}_{i+L} = \hat{\boldsymbol{x}}_k^- - \sqrt[C]{(\lambda + L)\boldsymbol{P}_{x_k}^-}|_i \quad i = 1, 2...L \tag{3.3}$$

where L refers to the number of states and the subscript i refers to the column number. The variable λ is a scaling parameter defined as

$$\lambda = \alpha^2(L + \kappa) - L \tag{3.4}$$

The α term changes the spread of the sigma points and κ is a secondary tuning parameter which is set to zero. After calculating the sigma points they are grouped into a matrix such that each sigma point is a column of the matrix. The complete set of sigma points is

$$\mathscr{X}_{Lx(2L+1)} = \begin{bmatrix} \mathscr{X}_0 & \mathscr{X}_1 & \cdots & \mathscr{X}_{2L} \end{bmatrix} \tag{3.5}$$

Next, the sigma points are transformed through the nonlinear observation function shown mathematically by

$$\mathscr{Z}_k|_i = h[\mathscr{X}_k|_i] \;\; \forall \;\; i \in [0, 2L] \tag{3.6}$$

The Kalman filter algorithm is modified to propagate and update the new sigma points. The major difference between the UKF and EKF is the transformed sigma points are stored and propagated in the estimator. In the EKF, only the mean and covariance of the system state are propagated.

Julier and Uhlmann show how the UKF outperforms the EKF in the relatively simple system described by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix}, \nabla f = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \tag{3.7}$$

The basic idea is that a sonar system with good ranging accuracy ($\sigma_r = 2$cm) and poor bearing accuracy ($\sigma_\theta = 15$deg) is used to determine the 2-dimensional Cartesian location of a target. Since the bearing information is not very accurate, the local assumption of linearity is violated in the EKF. The only way to maintain a stable estimate of the state is to artificially increase the process noise covariance to account for the linearization errors made by the EKF.

Figure 3.2 shows the mean and standard deviation ellipses for the actual and linearized form of the transformation. The true mean is at "x" and the uncertainty ellipse is solid. Linearization calculated the mean at "o" and the uncertainty ellipse is dashed [4]. The UKF approach is summarized with the ability to predict the system state more accurately and with less difficulty. The paper ends with a description of a companion document which contains the derivation of an unscented transform with

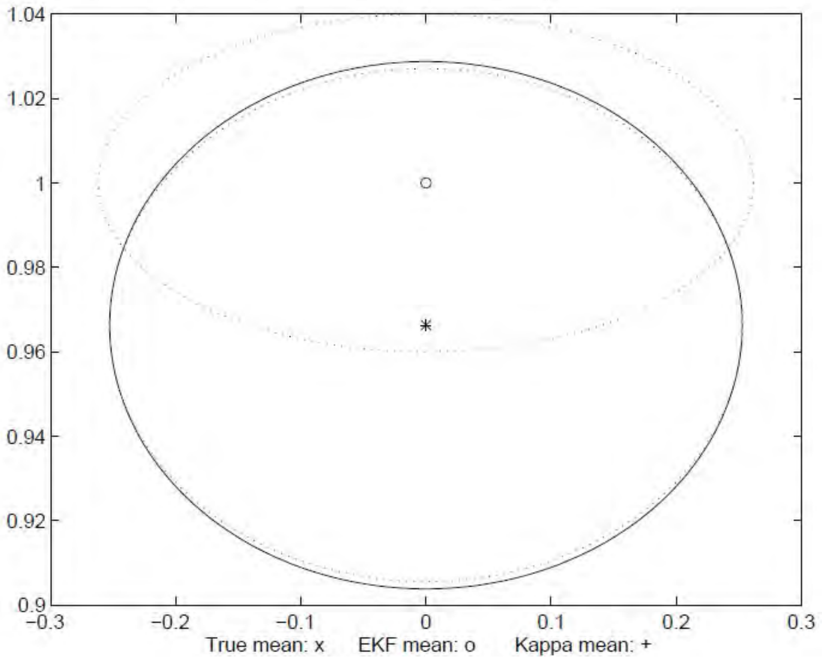


Figure 3.2: Visualization of Unscented Transform Measurement Model [4]

more sigma points. This modified UKF is lauded to match up to 4th-order moments of the state variables. The Unscented Trasnform forms the core of the UKF and UKS algorithms. It provides a convenient way to acheive an accurate approximation of system state variables.

3.3 Flight Path Reconstruction Using The Unscented Kalman Filter Algorithm.

Teixeira, Torres, Andrade de Oliveira, and Aguirre propose the use of an Unscented Kalman Filter (UKF) and derivated Unscented Kalman Smoother (UKS) to reconstruct the flight path of a sailplane aircraft during a series of flight tests [13]. The UKS is possible only when estimation if performed offline (i.e., cannot occur in real-time). One of the important benefits of reconstructing the flight path accurately based on all sensor data available is that individual sensor biases and sensitivity mismatches could be calculated. An important goal of the research is to calculate the biases associated to each accelerometer and gyro in the sailplane's Inertial Measurment Unit (IMU).

The authors begin with a general approach to state vector recursive estimation. Mention is made of possibly using a first-order estimator like the Extended Kalman Filter (EKF). The reasoning for using a second-order estimator such as the UKF was based on the author's intuition that it should be easier to approximate a Gaussian distribution than an arbitrary nonlinear function [4]. Instead of linearizing the model equations, this algorithm propagates a small representative group of deterministically chosen points (actually vectors) named sigma points: $\boldsymbol{X}_i, i = 0, 1, ..., 2L,$. The dimension of the augmented state vector L, which by construction includes the mean and covariance information of the state estimate at time $k-1$, with $t = kT_s$ where k denotes discrete time and T_s is the sampling period, in order to numerically calculate the prior state estimate $\hat{\boldsymbol{x}}_{k|k-1}$ and its covariance matrix $\boldsymbol{P}_{k|k-1}$ by their propagation through the discrete counterpart of the nonlinear equations [13].

The recursive UKF algorithm provides estimates of the state vector based on all past measurements including the present. Since the authors had the option of offline estimation to reconstruct the flight path of the sailplane, they employed a UKS. Basically, the UKS optimally combines the results of a forward and a backward UKF. The forward UKF estimates mean and covariance ($\hat{\boldsymbol{x}}^f_{k|k}, \boldsymbol{P}^f_{k|k}$) given all past and present data and the backward UKF estimates ($\hat{\boldsymbol{x}}^b_{k|k+1}, \boldsymbol{P}^b_{k|k+1}$) given all future data. The conversion from forward UKF to backward UKF dynamics model requires only an inversion of input and output data coupled with using the reverse of the dynamics model:

$$\dot{\boldsymbol{x}}_{-t} = -\boldsymbol{f}[\boldsymbol{x}_{-t}, \boldsymbol{u}_{-t}, \boldsymbol{w}_{-t}] \tag{3.8}$$

$$\boldsymbol{y}_{-t} = \boldsymbol{h}[x_{-t}, r_{-t}] \tag{3.9}$$

After running two UKF passes (forward and backward) through the data, the UKS if formed by optimally combining the results of each filter. The combination of the backward and forward filters is a smoothed mean and covariance ($\hat{\boldsymbol{x}}^s_k, \boldsymbol{P}^s_k$) given by

$$[\boldsymbol{P}^s_k]^{-1} = [\boldsymbol{P}^f_{k|k}]^{-1} + [\boldsymbol{P}^b_{k|k+1}]^{-1} \tag{3.10}$$

and

$$\hat{\boldsymbol{x}}^s_k = \boldsymbol{P}^s_k[[\boldsymbol{P}^f_{k|k}]^{-1}\hat{\boldsymbol{x}}^f_{k|k} + [\boldsymbol{P}^b_{k|k+1}]^{-1}\hat{\boldsymbol{x}}^b_{k|k+1}]. \tag{3.11}$$

A battery of simulations was performed in MATLAB using a high-fidelity model of a DHC-2 Beaver aircraft model and additive white Gaussian noise corrupted measurements. Two types of trajectories were examined, smooth and very rapid. The UKS used the state space model to incorporate IMU data ($T_s = 0.10$s) with GPS updates ($T_{GPS} = 1.0$s) post-flight to better reconstruct the flight path and thus obtain a better estimate of IMU biases. It was noted the smooth flight trajectory is preferred. This is expected because a smooth flight trajectory should make it easier to distinguish large errors introduced by accelerometer and gyro biases [13]. The rapid flight path made it difficult for the algorithm to distinguish between abrupt control inputs and IMU biases. The simulation was conducted using differing signal-to-noise ratios (SNR) for inputs and outputs. The results for the smooth trajectory simulation showed that the UKF was capable of converging on reasonable values for IMU biases. These results were further refined by using a post-flight UKS. Typically, the UKS provided at least a 50% reduction in state estimation error compared to the UKF for each of the 9 states.

The authors implemented a UKS to reconstruct the flight path of a sailplane and showed impressive improvements in accuracy over real-time data processing using emperical data from flight tests. These promising results partially motivated the pursuit of a similar approach to air-to-air missile flight path reconstruction. This particular smoother implementation was used as a model for the missile scoring algorithm presented in this work.

3.4 Air-to-Air Missile Vector Scoring Using COTS Sensors

Maj Nicholas Sweeney presents an architecture for air-to-air missile scoring using COTS radar sensors [12]. Sweeney's work provides the basis for this thesis. The Frequency Modulated Continuous Wave (FMCW) radar sensors are carefully located on the QF-16 platform. Seven antennas are located on the aircraft as follows: one directional antenna on the top and bottom of the nose section, one directional antenna on the top and bottom of each wingtip and an omnidirectional antenna on the aircraft tail. Sweeney predicts that this configuration may perform poorly for missile trajectories that approach in-plane with a wings level aircraft but any trajectories from above or below will likely have excellent sensor visibility.

The two software tools *Profgen* and *Argos 3.0* are used to generate truth trajectories for both the QF-16 drone and the inbound missile. *Profgen*, developed by

AFRL, is used to generate the QF-16 trajectory based on defensive maneuvers to defeat an inbound missile. The output of *Profgen* is used to provide target information for use in *Argos*. Argos is a 6DOF (Six Degree of Freedom) air-to-air missile simulation tool developed through collaboration between the National Air and Space Intelligence Center (NASIC) and the Air Force Research Laboratories (AFRL). The true missile position is not known by the Kalman filter, but it is used create noise corrupted sensor measurements.

An extended Kalman filter (EKF) is used to predict the missile's path based on AWG noise corrupted range and range-rate measurements. The QF-16 drone equipped with sensors which measure range and range-rate of an incoming air-to-air missile [12]. It is assumed that the sensor locations on the platform are known. The filter computed missile position improves near the intercept point using an EKF based upon a constant velocity (CV) model of the missile dynamics. In Major Sweeney's thesis, a UKF and PF are also implemented [11].

Radar clutter is simulated through the inclusion of random clutter measurements in the noise-corrupted measurements provided to the Kalman filter. Ellipsoidal gating and Global Nearest Neighbor (GNN) data association are used to isolate accurate measurements. These methods of gating and data association are utilized in this thesis.

There are three engagement scenarios simulated in Sweeney's work. These three scenarios are used in this thesis to provide a meaningful performance baseline. Scenario 1 is a non-maneuvering QF-16 drone being attacked from below. Scenario 2 is a tail-aspect attack against the drone performing a 9G break-turn. Scenario 3 is also tail-aspect, but the simulated drone performs a 7G vertical pull-up maneuver. The tuning process focused on scenario 2 due to its high dynamic properties. This method of tuning was also used in this thesis.

Sweeney recommends implementing a nonlinear filter such as the unscented Kalman filter. The implementation of a fixed-interval smoother is also recommended since the current CONOPS allow for data processing post-flight. Sweeney's recommendations for future research are the motivation for the UKS developed for this thesis. The *Profgen* and *Argos* truth generation system, measurement environment, and gating scheme is used in this thesis. This work forms a basis for the exploration and implementation of fixed-interval EKS, IEKS, SFRA EKS, UKS, IUKS, and SFRA UKS analysis.

IV. Methodology

As mentioned in Chapter I, this research introduces a post-flight missile-scoring system which uses FMCW RF sensors carefully located on a QF-16 drone to measure the range and range-rate of an inbound missile with the goal of estimating the missile's trajectory. As described in Chapter II, the accuracy of position and velocity calculations is highly dependent on sensor geometry and the associated Positional Dilution of Precision (PDOP). The sensor locations on the QF-16 target platform are located to minimize PDOP. The Kalman smoother-based scoring system described in Chapter II is applicable for post-flight processing, and due to inherent limitations, cannot be employed in real time.

Six different Kalman smoothers are employed for estimating the missile's trajectory based on kinematic measurements from the RF sensor suite: EKS, IEKS, SFRA EKS, UKS, IUKS, and SFRA UKS. As described in Chapter II, each of these smoothers is a combination of forward and backward filters. For each simulation, one of three dynamics models (discussed in Section 4.2) is paired with one of three air-to-air engagement profiles (discussed in Section 4.5). In all cases, the forward filter is initialized with a hand-off state estimate and covariance before processing through all available data (discussed further in Section 4.4). The backward filter is initialized using a simulated GRDCS value before propagating backwards through time and performing its own estimate of the missile's trajectory. Using the mathematics described in Chapter II, the results from both filters are combined to produce a smoothed result.

This chapter begins with a discussion of the RF sensors and their location on the QF-16 drone platform in Section 4.1. The three missile dynamics models used in this research are described in Section 4.2. The initialization process for the forward and backward Kalman filters is explained in Section 4.4. The truth model used to assess the performance of each smoother is described in Section 4.3. Section 4.5 describes three basic air-to-air engagement scenarios: Non-manuevering, 9G descending break turn, and 7G vertical pull-up. These scenarios form the basis for judging smoother tracking performance.

4.1 Aircraft Sensor Configuration

An existing automotive-grade FMCW sensor is simulated for use in this research. The specifications of this sensors are [11]:

- Maximum Range: 350 meters
- 1-σ range resolution: 0.01 × range
- Range-rate resolution: 0.25 meters/second
- Transmit Pattern: Continuous, Spherical
- Measurements: Range and Range-rate only

This research assumes the QF-16 platform is the basis for the sensor suite. It is assumed there are a total of seven sensors located on the platform to provide uninterrupted visibility of the inbound missile [11]. The approximate specifications of the QF-16 sensor platform are:

- Length: 16 meters
- Wingspan: 10 meters
- Number of Sensors: 7
- Sensor Locations:
  - 2 directional in nose section (top/bottom)
  - 2 directional left wingtip (top/bottom)
  - 2 directional right wingtip (top/bottom)
  - 1 omnidirectional antenna on tail.
- Optimized for inbound missile trajectories above/below aircraft.

Table 4.1 lists the location of each sensor in the QF-16 body frame with the origin at the geometric center of the QF-16. Figure 4.1 shows the radar sensor layout on the top and bottom of the QF-16.

Table 4.1: Radar Sensor Locations w.r.t Center of Aircraft

Sensor Number	Location	x (m)	y (m)	z (m)
1	Nose, Top	8	0	-0.5
2	Nose, Bottom	8	0	0.5
3	Left Wing, Top	0	-5	0
4	Left Wing, Bottom	0	-5	0.1
5	Right Wing, Top	0	5	0
6	Right Wing, Bottom	0	5	0.1
7	Tail (Omni directional)	-8	0	-1

4.2 System Models

As discussed previously, a KF is composed of a dynamics model and an observation model. A Kalman Smoother is the optimal combination of two KF's, one propagating forward in time and one propagating backward. Chapter II describes the mathematical relationship between these components. This research compares performance using three linear dynamics models. Recall from Equations (2.79) and (2.80) that the dynamics model for a backward KF is the inverse of the forward KF. The forward and backward KF's used in the development of the six Kalman Smoothers are based on three linear dynamics models: constant velocity (CV), constant acceleration (CA), and a 3D coordinated turn (CT). Accordingly, only the forward dynamics models will be discussed in Sections 4.2.1 through 4.2.3.

4.2.1 Constant Velocity. The CV model uses only six navigation states to describe the position and velocity of the inbound missile. The state vector associated with this model is [2]

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix}^T . \tag{4.1}$$

Described in continuous time, the linear CV dynamics model is [2]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{w}(t) \tag{4.2}$$

where

$$\boldsymbol{F} = \left[\begin{array}{c|c} \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{array}\right] \tag{4.3}$$

$$\boldsymbol{G} = \left[\begin{array}{c} \mathbf{0}_{3x3} \\ \hline \boldsymbol{I}_{3x3} \end{array}\right] \tag{4.4}$$

and the covariance kernel of the AWG noise vector $\boldsymbol{w}(t)$ is defined by

$$E[\boldsymbol{w}(t)\boldsymbol{w}(t+\tau)] = \boldsymbol{Q} = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{bmatrix} \delta(\tau) \tag{4.5}$$

where q becomes a tuning parameter describing the uncertainty associated with the model.

The CV model assumes constant velocity along each inertial axis while modeling acceleration along each axis as an independent, zero-mean, Gaussian, white noise. The entire missile track lasts less than 1 second, so frequent updates are required to produce a high fidelity reconstruction. Accordingly, a flat Earth assumption is made and the time steps, T, are seperated by 10msec. For this research, missile trajectory propagation is performed in a local-level navigation frame with an origin fixed on the surface of the Earth.

The difference equation resulting from converting to discrete time is

$$\boldsymbol{x}_k = \boldsymbol{\phi}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1} \tag{4.6}$$

The state transition matrix, $\boldsymbol{\phi}$, is

$$\boldsymbol{\phi} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.7}$$

Using the Van Loan [3] method to solve for the noise strength matrix, $\boldsymbol{Q}_d$, produces

$$\boldsymbol{Q}_d = \begin{bmatrix} \frac{T^3}{3} & 0 & 0 & \frac{T^2}{2} & 0 & 0 \\ 0 & \frac{T^3}{3} & 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^3}{3} & 0 & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & 0 & T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 & 0 & T & 0 \\ 0 & 0 & \frac{T^2}{2} & 0 & 0 & T \end{bmatrix} q. \tag{4.8}$$

where $T = 10$msec.

4.2.2 Constant Acceleration. Instead of modeling the acceleration components as zero-mean, white, Gaussian variables, the CA model uses three additional states to propagate acceleration components. The resulting state vector is [2]

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & v_x & v_y & v_z & a_x & a_y & a_z \end{bmatrix}^T. \tag{4.9}$$

The $\boldsymbol{F}$ and $\boldsymbol{G}$ matrices can be described as [2]

$$\boldsymbol{F} = \left[\begin{array}{c|c|c} \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} & \mathbf{0}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{array}\right] \tag{4.10}$$

$$\boldsymbol{G} = \left[\begin{array}{c} \mathbf{0}_{6x3} \\ \hline \boldsymbol{I}_{3x3} \end{array}\right]. \tag{4.11}$$

The dynamic noise strength matrix, $\boldsymbol{Q}$, is in the same form as Equation (4.5).

The derivative of acceleration, known commonly as jerk, is modeled as a zero-mean, Gaussian, white noise [2].

The state transition matrix, $\boldsymbol{\phi}$, and discrete dynamic noise strength, $\boldsymbol{Q}_d$, are

$$\boldsymbol{\phi} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & \frac{T^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 & 0 & \frac{T^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

$$\boldsymbol{Q}_d = \begin{bmatrix} \frac{T^5}{20} & 0 & 0 & \frac{T^4}{8} & 0 & 0 & \frac{T^3}{6} & 0 & 0 \\ 0 & \frac{T^5}{20} & 0 & 0 & \frac{T^4}{8} & 0 & T & \frac{T^3}{6} & 0 \\ 0 & 0 & \frac{T^5}{20} & 0 & 0 & \frac{T^4}{8} & 0 & 0 & \frac{T^3}{6} \\ \frac{T^4}{8} & 0 & 0 & \frac{T^3}{3} & 0 & 0 & \frac{T^2}{2} & 0 & 0 \\ 0 & \frac{T^4}{8} & 0 & 0 & \frac{T^3}{3} & 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^4}{8} & 0 & 0 & \frac{T^3}{3} & 0 & 0 & \frac{T^2}{2} \\ \frac{T^3}{6} & 0 & 0 & \frac{T^2}{2} & 0 & 0 & T & 0 & 0 \\ 0 & \frac{T^3}{6} & 0 & 0 & \frac{T^2}{2} & 0 & 0 & T & 0 \\ 0 & 0 & \frac{T^3}{6} & 0 & 0 & \frac{T^2}{2} & 0 & 0 & T \end{bmatrix} q \tag{4.13}$$

4.2.3 Three-dimensional Coordinated Turn. The CT model contains exactly the same nine navigation states as the CA model. The main difference is the velocity navigation states are propagated according to an assumed constant turn rate, ω.

$$\omega = \frac{|\boldsymbol{v} \times \boldsymbol{a}|}{|\boldsymbol{v}|^2} \tag{4.14}$$

The acceleration states are propagated according to

$$\dot{\boldsymbol{a}}(t) = -\omega^2 \boldsymbol{v}(t) + \boldsymbol{w}(t). \tag{4.15}$$

The continuous time linear dynamics matrices are [2]

$$\boldsymbol{F} = \left[\begin{array}{c|c|c} \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} & \mathbf{0}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \boldsymbol{I}_{3x3} \\ \hline \mathbf{0}_{3x3} & A & \mathbf{0}_{3x3} \end{array}\right] \tag{4.16}$$

$$\boldsymbol{A} = \begin{bmatrix} -\omega^2 & 0 & 0 \\ 0 & -\omega^2 & 0 \\ 0 & 0 & -\omega^2 \end{bmatrix} \tag{4.17}$$

$$\boldsymbol{G} = \left[\begin{array}{c} \mathbf{0}_{6x3} \\ \hline \boldsymbol{I}_{3x3} \end{array}\right] \tag{4.18}$$

where $\boldsymbol{v(t)}$ is the 3D velocity vector and $\boldsymbol{\omega(t)}$ is a corresponding vector of independent, zero-mean, white Gaussian noise sources.

The CT state transition matrix is a function of ω as shown by [2]

$$\boldsymbol{\phi}(\omega) = \left[\begin{array}{c|c|c} \boldsymbol{A(\omega)} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \hline \mathbf{0}_{3x3} & \boldsymbol{A(\omega)} & \mathbf{0}_{3x3} \\ \hline \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \boldsymbol{A(\omega)} \end{array}\right] \tag{4.19}$$

where $\boldsymbol{A}(\omega)$ is defined as

$$\boldsymbol{A(\omega)} = \begin{bmatrix} 1 & \frac{\sin \omega T}{\omega} & \frac{1-\cos \omega T}{\omega^2} \\ 0 & \cos \omega T & \frac{\sin \omega T}{\omega} \\ 0 & -\omega \sin \omega T & \cos \omega T \end{bmatrix}. \tag{4.20}$$

The closed form solution for $\boldsymbol{Q}_d$ is rather complex and is not included for this reason.

4.2.4 Observation Model .

The mathematical relationship between range measurements and navigation position states is nonlinear. Accordingly, the measurement model is nonlinear with independent, additive, white, Gaussian noise. The range measurement from sensor i is related to the position states according to

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \tag{4.21}$$

The coordinates of sensor i, $[x_i \; y_i \; z_i]$, are fixed in relation to the body frame but must be converted to the same reference frame as the missile using DCM's as discussed in Chapter II. The state vector of the inbound missile is expressed in the local level Earth-fixed navigation frame (n-frame).

Similarly, the relationship between range-rate measurements and navigation velocity states is also highly nonlinear. Radar range-rate measurements from sensor i are defined by

$$v_i = -\frac{(v_x - v_{x_i})(x - x_i) + (v_y - v_{y_i})(y - y_i) + (v_z - v_{z_i})(z - z_i)}{\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}} \tag{4.22}$$

This research focuses solely on linearized 3-D missile dynamics models (constant velocity, constant turn rate, and constant acceleration). As mentioned, the observation model is distinctly nonlinear. The general form of which is

$$\boldsymbol{z}_k = \boldsymbol{h}[\boldsymbol{x}_k] + \boldsymbol{v}_k \tag{4.23}$$

where $\boldsymbol{z}_k$ is a vector of measurements, the matrix H_k relates the measurements to current states and $\boldsymbol{v}_k$ is a vector of zero-mean, additive, white, Gaussian measurement noise. The measurement noise covariance kernel is defined by

$$E[\boldsymbol{v}_k \boldsymbol{v}_j^T] = \boldsymbol{R}\delta_{kj} \tag{4.24}$$

In the EKS-based algorithms, the observation function is linearized about the current state vector using a Jacobian matrix which is calculated by performing a first-order Taylor expansion of each nonlinear function with respect to the current states. This Jacobian is evaluated at the current state estimate

$$\boldsymbol{H}_k = \frac{\delta \boldsymbol{h}}{\delta \boldsymbol{x}}|_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{4.25}$$

The UKS-based algorithms do not directly linearize the state vector and are able to preserve some of the nonlinearities present through sigma point propagation.

4.3 Truth Model

The truth data for the QF-16 drone flight trajectory is generated using *ProfGen*. This simulation tool converts user-specified aircraft dynamic capabilities into a software flight model. The user can use this model to produce representative evasive maneuvers such as break turns, jinks, rolls, etc. The QF-16 is modeled as a point-mass body which means lift, drag, and thrust calculations are approximated. *ProfGen* is sufficient for providing an approximate kinematic trajectory for the QF-16 platform in the WGS-84 ECEF reference frame. This trajectory is converted using DCM's in *MATLAB* to a local-level Earth-centered North-East-Down (NED) navigation frame before used as an input to *Argos 3.0*.

Argos 3.0 is a 6-DOF missile simulation tool developed by the National Air and Space Intelligence Center (NASIC) in cooperation with the Air Force Research Laboratories (AFRL). For this research, the missile model used in *Argos 3.0* is an unclassified short-range missile operating in a flat-Earth environment. The software operates in local-level NED navigation frame. The output of this tool is a high fidelity missile intercept trajectory in response to the QF-16 trajectory generated by *ProfGen*. Figure 4.2 shows an overview of the truth generation and flight path reconstruction process.

Radar measurement noise and clutter are simulated in *MATLAB* for individual sensors. During each measurement update, individual sensors return observations based upon the noise-corrupted version of the true range and range-rate as well as clutter measurements. The observations are generated by adding random noise to the range and range-rate measurements. This noise is a zero-mean, Gaussian distribution with variance defined by the sensor performance as described in Section 4.1. The number of clutter observations are chosen from a uniform distribution of integers over the interval [0,3]. The actual range values of these clutter observations are chosen from a random variable which is uniformly distributed over the sensor's entire detection range.

4.4 Target Initialization

The forward Kalman filters are initialized using missile position and velocity data from an external tracking source. Most USAF missile test ranges are equipped with a system which monitors missile position for safety. For example, the Eglin Gulf Test range uses the Gulf Range Drone Control System (GRDCS) [5] to provide real-time missile positioning. This system provides position accuracy with $1-\sigma$ values of 15 meters in the x and y-axis (North/East) and 45 meters in the z-axis (Down). The system updates missile position at 20 Hz and velocity is calculated based on the change in position between time steps. The initial forward filter missile state vector $\hat{x}_0^f$ is set equal to the GRDCS hand-off according to

$$\hat{\boldsymbol{x}}_0^f = \hat{\boldsymbol{x}}_{final}^{GRDCS}. \tag{4.26}$$

Random position and velocity error are added to each axis from a zero-mean normal distribution with the following standard deviations: $\sigma^f_{x0} = 15$, $\sigma^f_{y0} = 15$, $\sigma^f_{z0} = 45$, $\sigma^f_{v_x} = 10$, $\sigma^f_{v_y} = 10$ and $\sigma^f_{v_z} = 10$ meters. The initial forward state covariance matrix, P^f_0, is established using these GRDCS parameters via

$$\boldsymbol{P}^f_0 = \begin{bmatrix} {\sigma^f_{x_0}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & {\sigma^f_{y_0}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & {\sigma^f_{z_0}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & {\sigma^f_{v_{x_0}}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & {\sigma^f_{v_{y0}}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & {\sigma^f_{v_{z0}}}^2 \end{bmatrix} \tag{4.27}$$

The backwards Kalman filter is initialized using a simulated GRDCS hand-off taken from the end of the missile's flight path. The initial backward state covariance matrix, P^b_0, is established using

$$\boldsymbol{P}^b_0 = \begin{bmatrix} {\sigma^b_{x_0}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & {\sigma^b_{y_0}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & {\sigma^b_{z_0}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & {\sigma^b_{v_{x_0}}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & {\sigma^b_{v_{y0}}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & {\sigma^b_{v_{z0}}}^2 \end{bmatrix} \tag{4.28}$$

4.5 Engagement Scenarios

The flight profile for Scenario 1 is shown in Figure 4.3. The target aircraft is flying straight and level with a Northbound heading at an altitude of 5000 meters and maintains this attitude for the duration of the scenario. The shooter's initial position is 1.6 kilometers in front of the target, at an altitude of 500 meters with a 70 degree pitch-up attitude. The entire scenario lasts about 8 seconds. The missile enters within the max range of the drone's sensors at approximately T = 7.5 seconds.

Figure 4.3 also shows the forward and smoother estimates of the missile trajectory performed by the UKF and UKS algorithms, respectively.

Figure 4.4 shows the flight profile for Scenario 2. The target aircraft begins the scenario flying wings-level with a northbound heading at an altitude of 5000 meters. Immediately, the target is aware of the missile and rolls to perform a 90 degree 9G break turn. This maneuver is a common defensive tactic used against a missile observed within visual range. It allows the targeted aircraft to change direction quickly without bleeding off all its energy. The missile impact occurs as the target approaches an easterly heading at 4300 meters altitude. The shooter's initial position is 4.8 kilometers in trail with northbound heading at an altitude of 5000 meters. The entire simulation lasts about 8 seconds and the missile enters within the max range of the drone's sensors at approximately T = 7.6 seconds.

The flight profile for Scenario 3 is shown in Figure 4.5. The target begins the scenario flying wings-level with a northbound heading at an altitude of 5000 meters. The missile is launched at the outset of the simulation from a position two miles in trail with the same attitude and altitude as the target aircraft. The target is aware of the missile and immediately performs a 7G vertical pull at the beginning of the simulation. The entire simulation lasts about 8 seconds and the missile enters within the max range of the drone's sensors at approximately T = 7.6 seconds.

4.6 Summary

Simulation development begins with a defensive flight trajectory for the QF-16 generated in *ProfGen* software developed by AFRL. Initial shooter position and attitude are combined with this trajectory in *Argos 3.0* to generate a missile flight path. Simulated RF sensors, located on the QF-16, are used to generate range and range-rate measurements which are corrupted by AWG noise and clutter measurements in *MATLAB*. The forward Kalman filter is initialized based on a missile trajectory hand-off from GRDCS. After the forward filter process the flight path, the final state estimate and covariance of the missile trajectory are used to initialize the backward filter. The observation model processes these corrupted measurements and attempts to reconstruct the missile flight path using one of three dynamics models: CV, CA, CT. In addition to the three dynamics models, there are three basic air-to-air engagement scenarios: Non-manuevering, 9G horizontal break turn, and 7G vertical pull-up. These dynamics models and scenarios form the basis for judging flight path reconstruction performance.

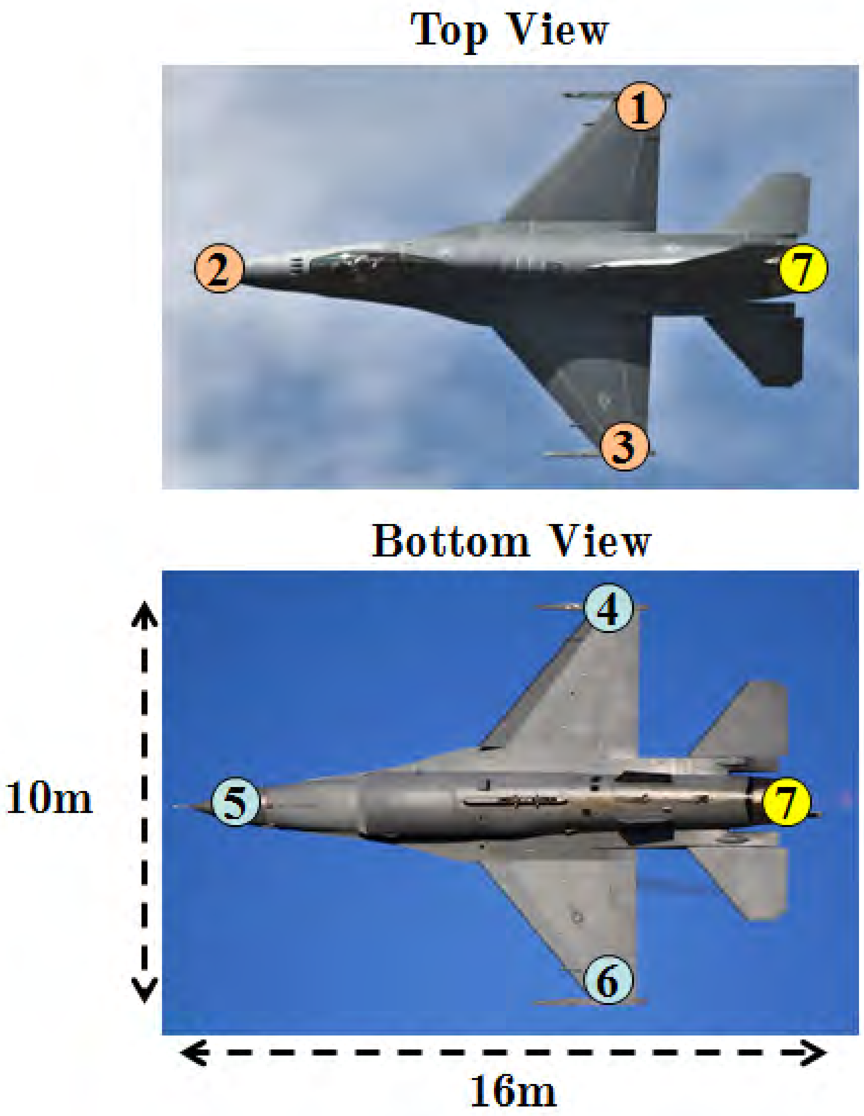


Figure 4.1: FMCW Radar Sensor Layout on QF-16

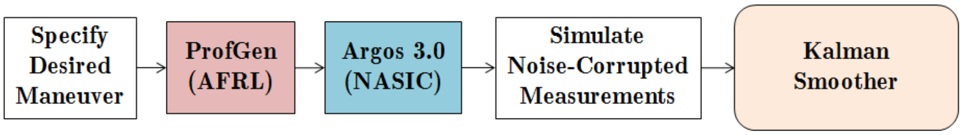


Figure 4.2: Overview of Truth Generation and Reconstruction

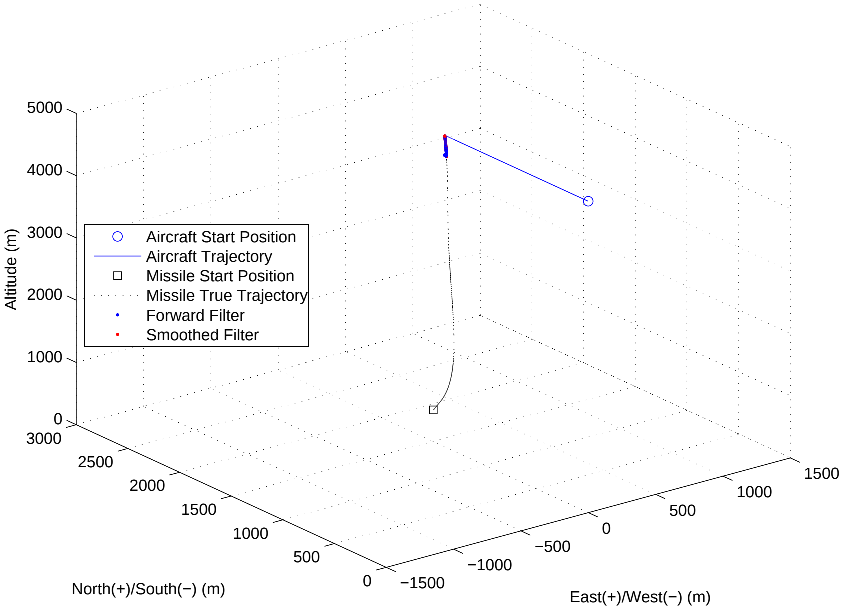


Figure 4.3: Scenario 1: Target Aircraft Non-maneuvering

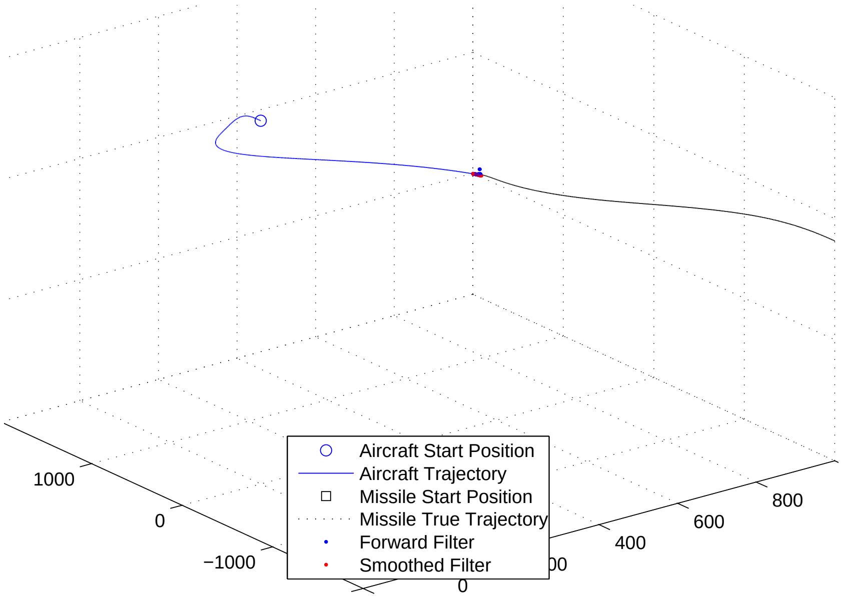


Figure 4.4: Scenario 2: Target Aircraft Performing a 9G Horizontal Break-Turn

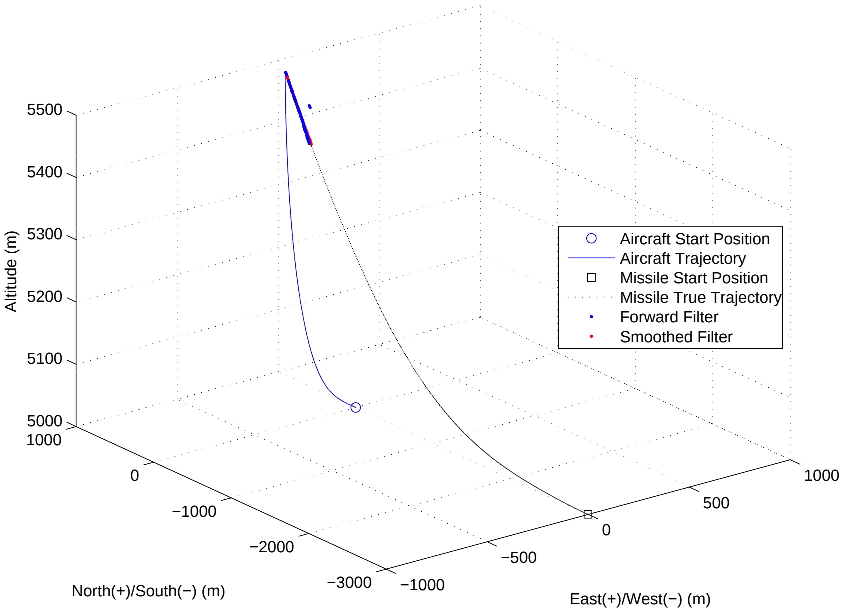


Figure 4.5: Scenario 3: Target Aircraft Performing a Vertical Climb

V. Experimental Results

5.1 Simulations

The goal of this research is to summarize the performance of six Kalman smoothers and recommend the best algorithm for DoD/USAF use in missile vector scoring and trajectory reconstruction. These smoothers are examined using three dynamics models, each in three different air-to-air engagement scenarios as mentioned in Chapter IV. This equates to a minimum of 54 simulation combinations. Sensor noise and target initialization errors are generated for each engagement scenario and saved with the requirement that each of the six smoothers uses the same noise realizations.

All simulations are based on 100-run Monte Carlo analyses for which position and velocity mean error and standard deviation are plotted for each state. The Monte Carlo Mean 3-D RSS position error $\pm 1\sigma$ is plotted to show expected position error throughout the missile fly-out. Since nearly all the forward-only filters and smoothers were able to provide centimeter-level accuracy at the point of impact, the main performance metric used is the expected value of the Monte Carlo Mean 3-D RSS position error. The reasoning for using this metric is discussed in Section 5.2. Section 5.3 explains the systematic manner with which filter tuning for all smoothers was established. A synopsis of the performance results from the six smoothers is available in Section 5.4. Appendix A provides a comprehensive catalog of simulation results for every smoother/scenario/dynamics model combination. As one will see, the results between the SFRA UKS and standard UKS are incredibly close. In order to differentiate between the two, three advanced stability tests are created. Discussed further in Section 5.5, these tests simulate random sensor dropout and artificially increase sensor noise to analyze filter stability. This section concludes with an analysis of these stability tests and establishes which algorithm performs the best in this research.

5.2 Performance Metric Selection

The goal of selecting a performance metric is to capture many components of several complicated processes in an easily-digestible value. Stochastic processes are often analyzed using Monte Carlo simulations to capture stochastic performance when ergodicity is not guaranteed. Given these two constraints, a logical first choice is Monte Carlo mean position error at point of impact. The point of impact calculations were calculated and observed to not exceed 20 cm for any given Kalman Smoother/Dynamics Model/Flight Scenario combination.

Miss distance alone does not adequately evaluate the performance of an air-to-air missile. The main purpose of missile flight path reconstruction is to capture true missile performance. This includes the missile's flight vector, turn rate, closing velocity, and miss distance. It is difficult to convey these attributes in a single metric.

It was apparent that the metric selected should be able to convey the accuracy of the entire missile trajectory estimate. In doing so, the new metric should be able to partially capture the convergence time and overall stability of the algorithm. Thus, the decision was made to use the expected value of Monte Carlo 3D Mean Root of Sum of Squares (RSS) Position Error. This metric describes the expected value of 3D position error for Monte Carlo simulations over the entire missile trajectory. The method for calculating this metric is shown below.

For each time step k of each simulation run, position error for each NED axis is calculated according to

$$N_{err_k} = \hat{x}^N_{k|k} - x^N_{t_k} \tag{5.1}$$

$$E_{err_k} = \hat{x}^E_{k|k} - x^E_{t_k} \tag{5.2}$$

$$D_{err_k} = \hat{x}^D_{k|k} - x^D_{t_k} \tag{5.3}$$

where $\hat{x}_{k|k}$ is the aposteriori state estimate and x_{t_k} is the true missile position at time step k . At this point, the 3D RSS position error is calculated for each time step according to

$$RSS_k = \sqrt{N_{err_k}^2 + E_{err_k}^2 + D_{err_k}^2}. \tag{5.4}$$

After obtaining RSS_k for each time step of a single run, a 100-run Monte Carlo simulation is performed. The arithmetic mean of the 3D RSS position error is calculated for each time step k across 100 runs. The Monte Carlo standard deviation is also calculated. The results are

$$E[RSS_k] = \frac{1}{100} \sum_{i=1}^{100} RSS_k. \tag{5.5}$$

$$\sigma_{RSS_k} = \sqrt{\sum_{i=1}^{100} \frac{(RSS_k - E[RSS_k])^2}{100}}. \tag{5.6}$$

For example, Figure 5.1 shows $E[RSS_k] \pm \sigma_{RSS_k}$ for the forward, backward, and smoother outputs. This is a sample output using the UKS algorithm with the CT dynamics model and 7G Vertical trajectory.

The performance metric, (expected value of Monte Carlo Mean 3D RSS position error) is then calculated (i.e., calculating the arithmetic mean of $E[RSS_k]$ for all time steps $k = 1, 2...k_{final}$)

$$PM = \frac{1}{k_{final}} \sum_{k=1}^{k_{final}} E[RSS_k]. \tag{5.7}$$

Using the example provided in Figure 5.1, PM_{UKF} is 4.549 meters and PM_{UKS} is 2.119 meters. Results for all algorithms are available in Tables 5.2- 5.4.

5.3 Filter Tuning

To maintain consistency and applicability of results, filter tuning was not changed significantly from the baseline established in previous research. Since the backward-propagation filters are derived from the forward filter dynamics models and use the

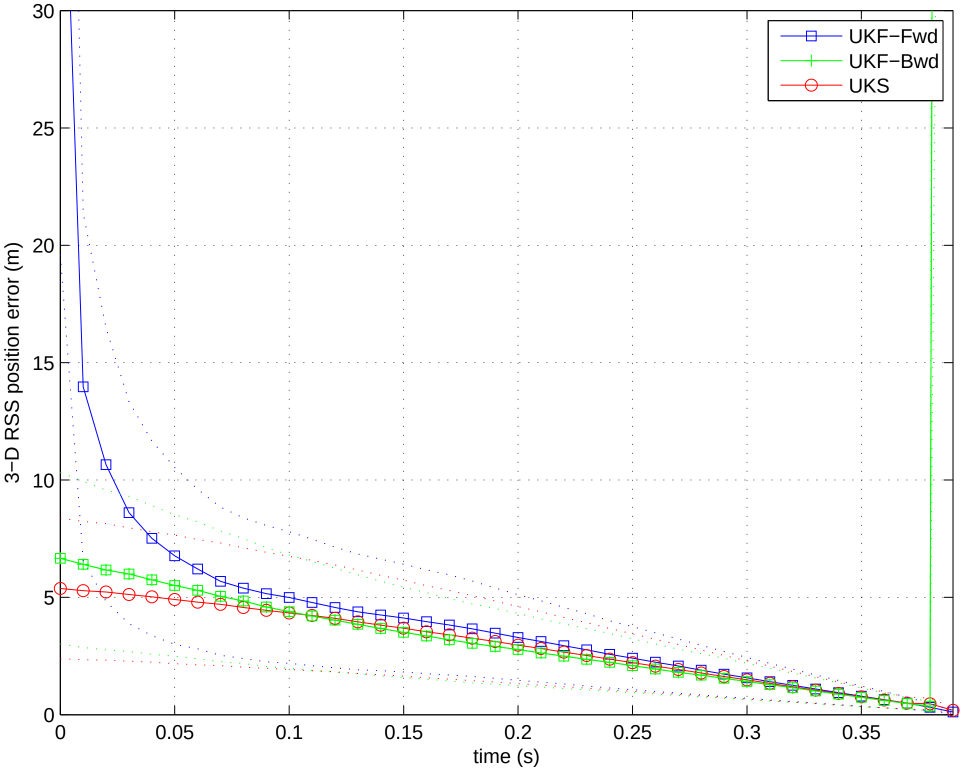


Figure 5.1: Monte Carlo Mean 3D RSS Position Error (100 Runs)

same measurement models, backward filter tuning was also unchanged. Tuning for both the EKF and UKF was established using the CA dynamics model with the 9G break turn (Scenario 2). The CA model was chosen for because it provides dynamics fidelity between that of the CV and CT models. The 9G break turn scenario was chosen because of the high level of missile dynamics. The purpose of using systematic tuning is to provide a comparable baseline for all smoothers and to aide in assessing stability. Only two parameters were varied while tuning for Scenario 2: dynamic noise strength, $\boldsymbol{Q}$, and the magnitude values of the measurement noise strength matrix, $\boldsymbol{R}$.

The Monte Carlo mean position and velocity errors ± 1σ were plotted for each combination to observe KF stability. Figure 5.2 shows an example for the UKS using the CA dynamics model in the 9G break turn scenario. The mean error and associated standard deviation is plotted for the position and velocity states (x, y, z, v_x, v_y, v_z). The turn rate ω and acceleration states could also be used for tuning if applicable to the dynamics model. A properly tuned filter will produce mean error values of nearly 0 and should be bounded within ± 1σ at least 68.29% of the time. The UKS in Figure 5.2 appears to be performing well and is stable.

Multiple trials were performed using varying dynamics and measurement noise values. Both the EKS and UKS tuning parameters were obtained based on systematic analyses of the performance metric established in Equation (5.7). The values obtained using this tuning process for the CV dynamics model in scenario 2 are $q = 1,000$, $R_{range} = 10$, and $R_{velocity} = 20$. A sample discretized Q_d generated using the Van Loan method is

$$\boldsymbol{Q}_d = \begin{bmatrix} 0.0003 & 0 & 0 & 0.0500 & 0 & 0 \\ 0 & 0.0003 & 0 & 0 & 0.0500 & 0 \\ 0 & 0 & 0.0003 & 0 & 0 & 0.0500 \\ 0.0500 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0.0500 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0.0500 & 0 & 0 & 0.1 \end{bmatrix}. \tag{5.8}$$

After establishing the tuning baseline for the CV model in scenario 2, values were obtained using the same method for the CA and CT dynamics models. For these models, the measurement noise remained the same and only the dynamics noise q was changed. Table 5.1 lists the tuning values used for each dynamics model.

Table 5.1: Kalman Smoother Tuning Parameters

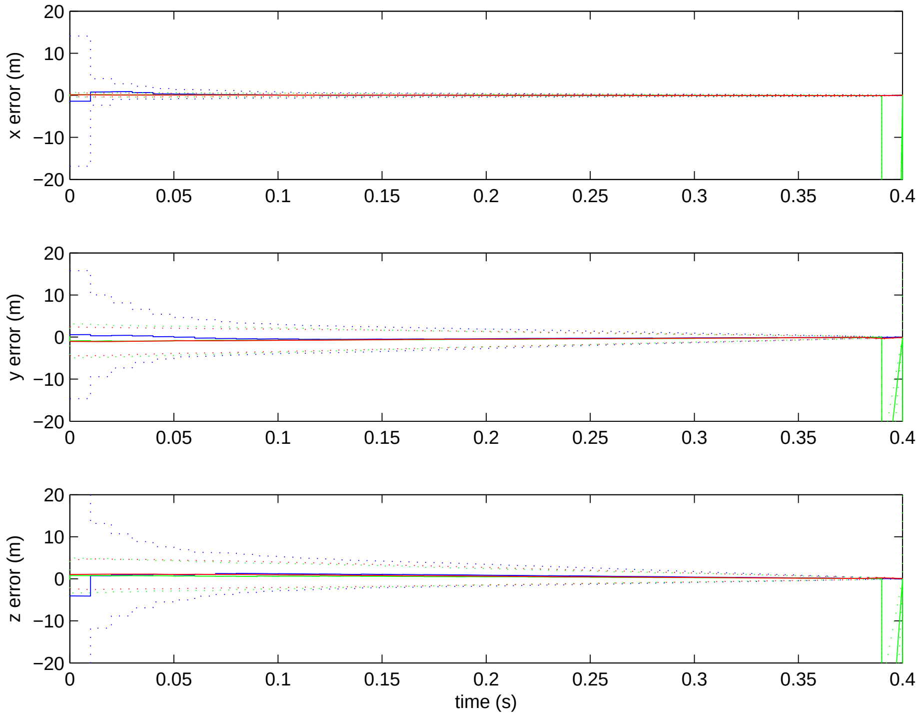
Filter-Dynamics Model	q	r_{range}	$r_{velocity}$
CV	1,000	10	2
CA	800,000	10	2
CT	800,000	10	2

5.3.1 Tuning for Iterative Filters . Recall from Chapter II, there are user-defined thresholds which must be established for each iterative filter. In these cases, a systematic method of tuning was used to establish a balance between filter processing time and performance. The user-defined threshold value for the IEKF from

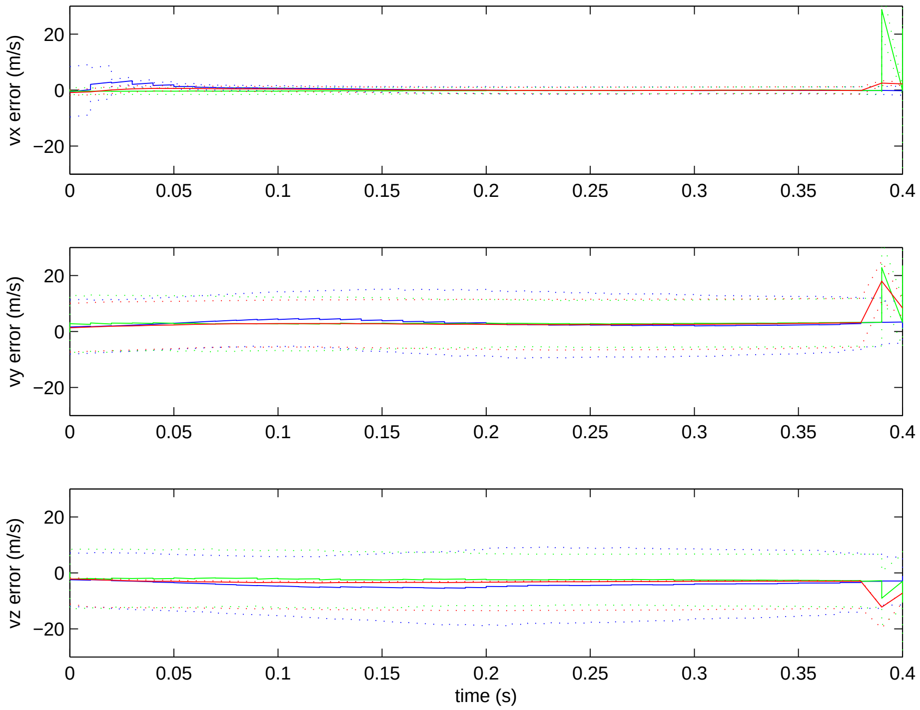
Equation (2.34) was established by continually reducing the iterative threshold C_I until the performance metric from Equation (5.7) converged to a stable value within $10cm$. For this thesis, the threshold for both the forward and backwards IEKF is $C_I = 0.1$. Recall that iterations continue until the difference between the estimates is smaller than C_I or a maximum number of iterations has been reached. A maximum iteration count of 250 was selected based on monitoring the total number of iterations performed at each time step and selecting a number sufficiently large to make the possibility of converging to a value within $10cm$ likely.

The unstable IUKF from Section 2.2.6 also requires an iterative user-defined threshold. The values for both the forward and backwards IUKF is 0.1, obtained by observing when the position error metric converges to within $10cm$. The iterations continue until the difference between the estimates is smaller than the threshold or a maximum number of iterations has been reached. A maximum iteration count of 1000 was selected based on monitoring the total number of iterations performed at each time step and selecting a number sufficiently large to increase the probability of convergence. At this point in IUKS development, it was discovered that the IUKS will not necessarily converge on a better estimate as iterations increase. New sigma points are generated and transformed through $\boldsymbol{h}[\cdot]$ at each iteration. This process does not necessarily converge no matter the choice of C_I.

5.3.2 Tuning for SFRA Filters . Recall from Sections 2.2.3 and 2.2.7 the SFRA EKF and SFRA UKF require user-defined thresholds for comparison with measurement residuals. A threshold value of $C_R = 15$ was chosen for this research also based on position error metric convergence analysis to $10cm$. If the mean of the residuals exceeds this threshold, the filter returns to Equation (2.47) and generates new sigma points. These sigma points are sent directly to Equation (2.69) to be converted into measurement space. From this point, measurements are re-gated based on the new measurement prediction and new residuals are formed. If the mean of the new residuals still exceeds the threshold, the filter iterates again. Iteration ceases if the mean of the residuals is below the threshold or if the maximum number of iterations is reached.



(a) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(b) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure 5.2: UKS Mean Monte Carlo Position and Velocity Errors for CA dynamics model (Scenario 2) Blue: Forward KF, Green: Backward KF, Red: Smoother

5.4 Performance Results

Before interpreting the results from Tables 5.2-5.4, it is important to remember that this research uses both linearized and nonlinear fixed-interval smoothers to approximate non-linear effects. Although the three dynamics models are all linear, the measurement model has 2^{nd} order nonlinearities. Therefore it is expected that the nonlinear smoothers should outperform their linearized counterparts. This research presents a myriad of Kalman Filters, smoothers, dynamics models, and engagement scenarios to provide a comprehensive analysis. The goal of this research is to compare various Kalman smoothers against known baselines and determine which algorithm is best-suited for missile vector scoring and trajectory reconstruction.

The results for all Kalman algorithms, organized by engagement and scenario dynamics model are listed in Tables 5.2- 5.4.

Table 5.2: Scenario 1: Non-Maneuvering, Arithmetic Mean of Monte Carlo Mean 3D RSS Position Error

Algorithm	Scenario 1, CV	Scenario 1, CA	Scenario 1, CT
UKS	3.104 m	3.343 m	3.344 m
UKF	4.835 m	4.999 m	4.999 m
SFRA UKS	2.921 m	3.844 m	4.558 m
SFRA UKF	4.657 m	4.812 m	4.812 m
IUKS	6.991 m	3.605 m	3.605 m
IUKF	8.397 m	5.879 m	5.879 m
EKS	2.918 m	3.819 m	3.819 m
EKF	5.118 m	5.958 m	5.958 m
SFRA EKS	3.015 m	3.892 m	3.891 m
SFRA EKF	5.346 m	6.159 m	6.159 m
IEKS	2.865 m	3.670 m	3.670 m
IEKF	5.106 m	5.823 m	5.822 m

Table 5.3: Scenario 2: 9G Descending Break Turn, Arithmetic Mean of Monte Carlo Mean 3D RSS Position Error

Algorithm	Scenario 2, CV	Scenario 2, CA	Scenario 2, CT
UKS	2.855 m	2.477 m	2.477 m
UKF	4.573 m	4.584 m	4.584 m
SFRA UKS	2.278 m	2.953 m	2.743 m
SFRA UKF	4.617 m	4.673 m	4.673 m
IUKS	5.160 m	4.923 m	5.305 m
IUKF	6.482 m	6.011 m	6.744 m
EKS	4.008 m	7.199 m	7.096 m
EKF	8.844 m	12.207 m	12.204 m
SFRA EKS	3.057 m	5.271 m	5.261 m
SFRA EKF	7.622 m	10.956 m	10.953 m
IEKS	3.884 m	6.904 m	6.875 m
IEKF	8.539 m	11.472 m	11.470 m

Table 5.4: Scenario 3: 7G Vertical Maneuver, Arithmetic Mean of Monte Carlo Mean 3D RSS Position Error

Algorithm	Scenario 3, CV	Scenario 3, CA	Scenario 3, CT
UKS	2.978 m	2.961 m	2.961 m
UKF	4.644 m	4.715 m	4.715 m
SFRA UKS	2.989 m	3.464 m	4.330 m
SFRA UKF	4.452 m	4.471 m	4.471 m
IUKS	2.875 m	7.941 m	5.642 m
IUKF	6.125 m	8.934 m	7.448 m
EKS	4.017 m	6.331 m	6.329 m
EKF	7.188 m	9.791 m	9.789 m
SFRA EKS	3.312 m	5.028 m	5.027 m
SFRA EKF	5.996 m	8.068 m	8.067 m
IEKS	3.992 m	7.106 m	7.070 m
IEKF	7.206 m	10.299 m	10.294 m

Perhaps more easily digestible, Figure 5.3 sums all the position error performance metrics for each algorithm. From this perspective it is easy to see that the implementation of a fixed-interval smoother increases accuracy appreciably. In fact, it decreases expected overall position error by an average of 43.1%.

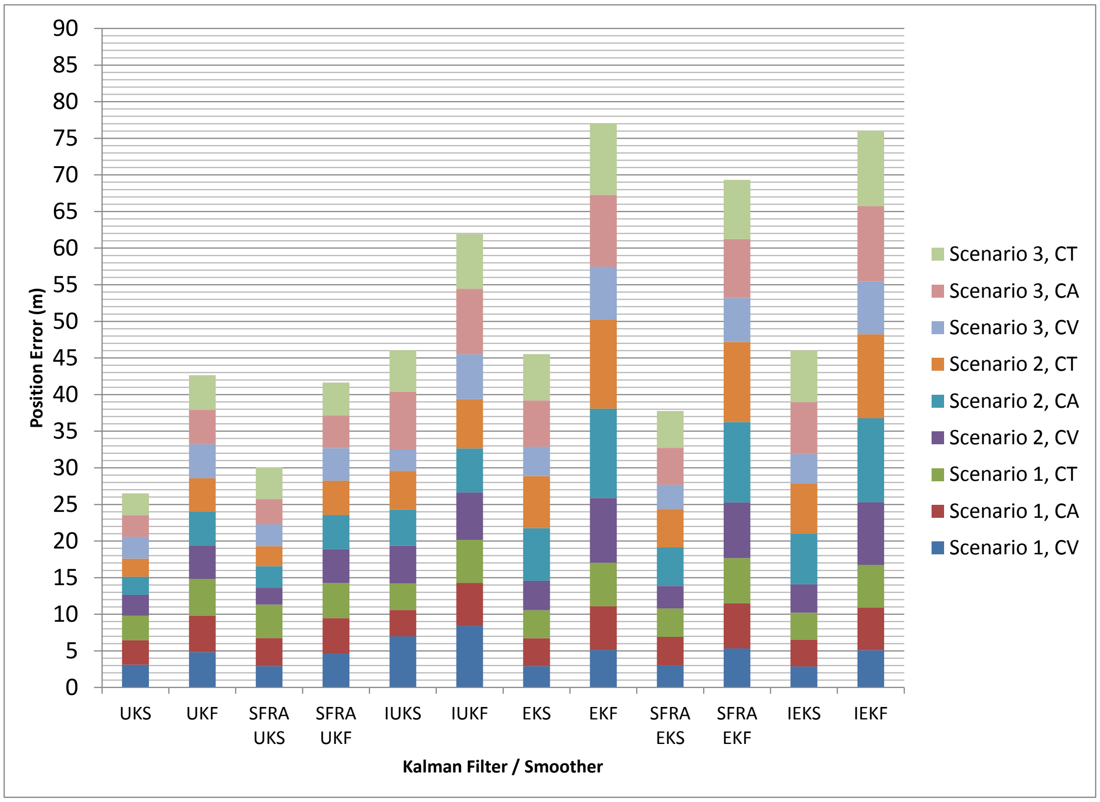


Figure 5.3: Accumulated Performance Metrics for All Algorithms

Figure 5.4 shows another view by eliminating the results from the forward-only KF's. The three nonlinear smoothers are presented with their linearized counterparts. When examining the linearized smoothers, the SFRA EKS provides a small 4.7% overall reduction in position error over the EKS. If we examine the results from the non-maneuvering engagment (scenario 1), the EKS is on par with SFRA EKS. As one might expect, the maneuver detection provided by the SFRA EKS enabled higher accuracy estimates during the highly dynamic scenarios (2-3). The IEKS provides a small 2.3% position error reduction when compared to the standard EKS algorithm.

This is expected because the relinearization of $\boldsymbol{h}[.]$ should reduce some of the error associated with linear approximation, but the implementation of a smoother should provide most of the gains. This shows there is small performance margin to gain in both re-linearization of $\boldsymbol{h}[.]$ and in maneuver detection.

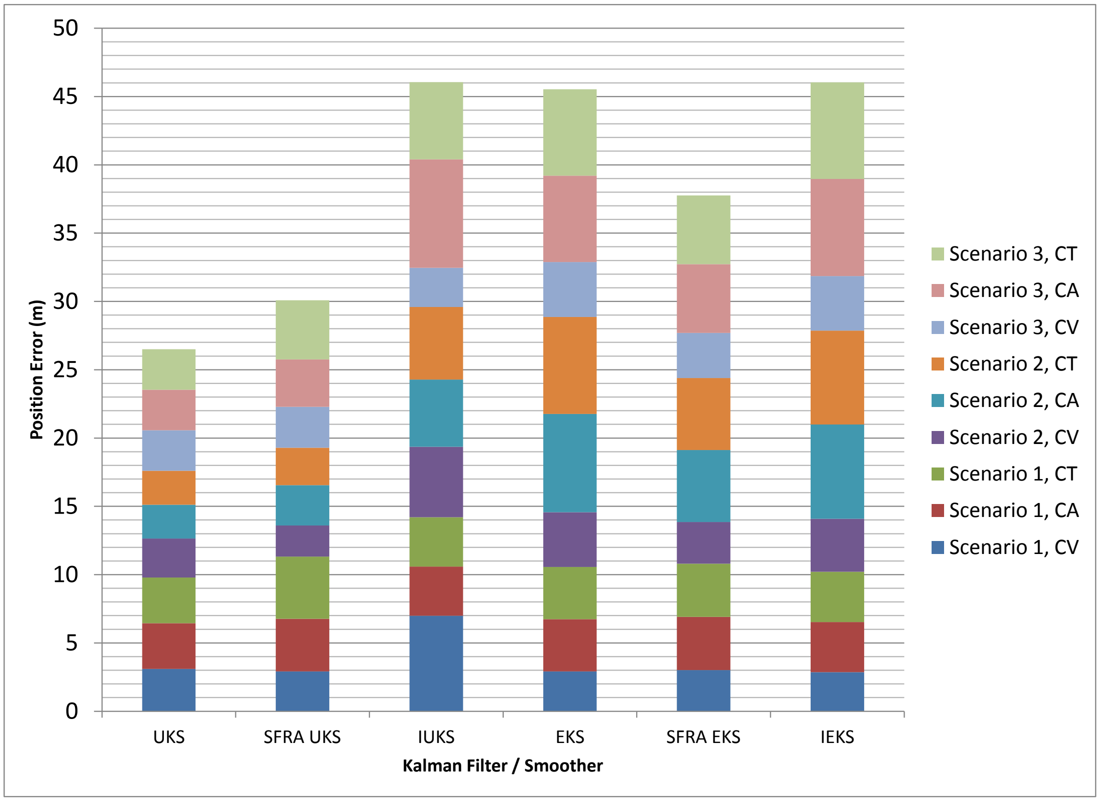


Figure 5.4: Performance Metrics for Kalman Smoothers

The IUKS results are included only for the sake of completeness. It has been established that the IUKS algorithm will not necessarily converge to a stable estimate. The performance reduction can be attributed to the fact that 2^{nd} order nonlinearities of the measurement function $\boldsymbol{h}[\cdot]$ should be adequately captured by the standard UKS. The iterative transformation of sigma points in the IUKS nets no increase in performance. When examining the nonlinear Kalman smoothers versus their linearized counterparts, this research indicates a 30.9% reduction in expected position

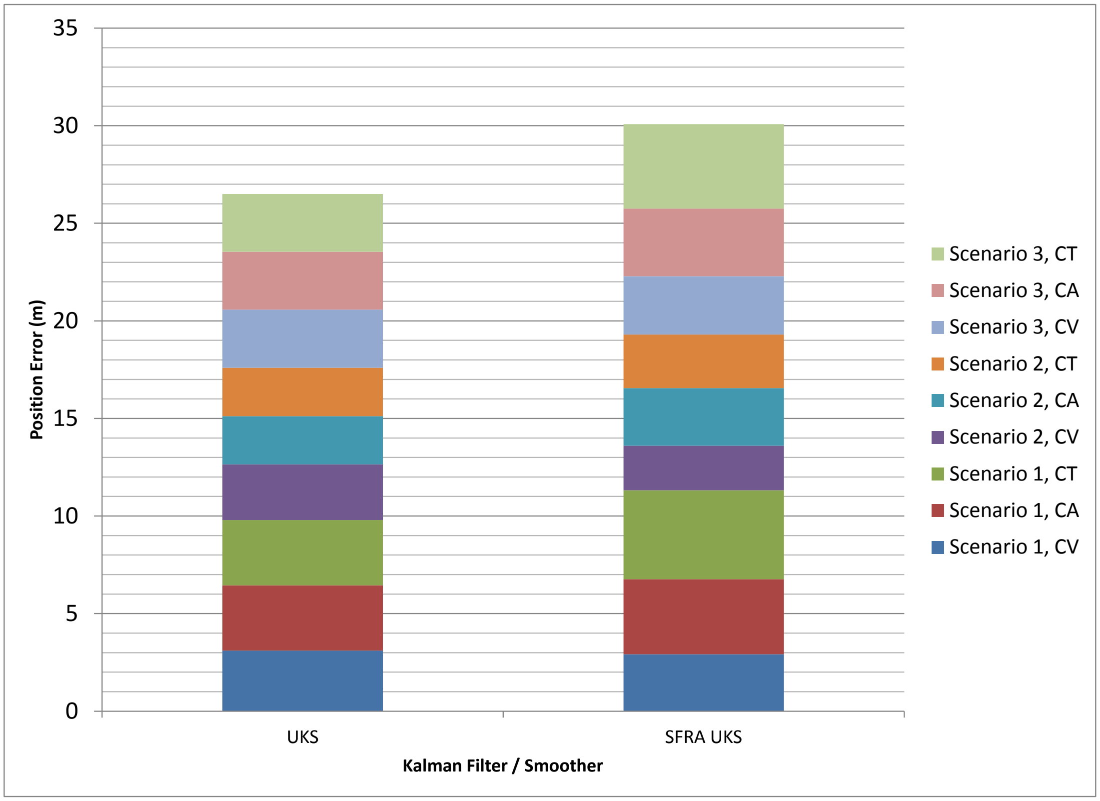


Figure 5.5: Performance Metrics for UKS and SFRA UKS

error (excluding the non-optimal IUKS). The results of this work indicate that the UKS or SFRA UKS should be used.

From Figure 5.5, it is still difficult to ascertain the differences between the UKS and SFRA UKS. The SFRA UKS performs on par with the UKS with the exception of the non-maneuvering scenario. The UKS provides an 11.5% reduction in overall position error. The only minor performance difference improvement for the SFRA UKS seems to lie within the two highly-dynamic engagement scenarios. The basic maneuver detection of the SFRA UKS may provide an incremental performance benefit. Since the results of this comparison are close, it is imperative to analyze these two smoothers from another perspective. The next section compares the UKS versus the SFRA UKS in a series of advanced stability tests to determine an overall winner.

5.5 Performance Tests

Since the UKS and SFRA UKS perform nearly identically across the battery of dynamics models and engagement scenarios, a new series of tests needs to be developed. The first series of tests simulates random sensor dropout. This test should help pinpoint any weaknesses in algorithm stability. Sensor noise and all other parameters are unchanged from the original configuration. For time step t_k, a random number of sensors is available. The PDF of this random sensor dropout is a uniform distribution. A 100-run Monte Carlo simulation is performed across all dynamics models and engagement scenarios. Figures 5.6 and 5.7 show $E[RSS_k]$ for both 100 runs and for 1 run. The realization of sensor availability for the single run is also included.

Interestingly, the UKS and SFRA UKS perform on par with each other, with a slight 4.1% performance margin in favor of the SFRA UKS. The overall expected position error for each smoother has effectively doubled from the original case with all available sensors. Sensor dropout causes an increase the time it takes to converge on the missile trajectory, but does not seem to affect overall stability. Since this test is inconclusive, more tests must be conducted.

The weakness of the SFRA UKS lies in its inability to perform well when real world dynamics and sensor performance do not match the model. This is because the SFRA algorithm monitors residuals and moves the state estimate towards the measurement until the residual threshold is achieved. The second test involves increasing the position sensor noise by a factor of 10 from 0.01 to 0.10 x range. No tuning was performed to either smoother before running through the familiar gamut of dynamics models and scenarios. The results of this test are shown in Figure 5.9.

The UKS shows an overall 10.6% decrease in expected position error. The basic maneuver detection in the SFRA UKS relies on residual monitoring and is more likely to mistake bad measurements for abrupt changes in trajectory. This results in less accuracy, especially when the dynamics and observation models are mismatched. For the sake of completeness, the results for the same test with $\boldsymbol{R}$ properly compensated

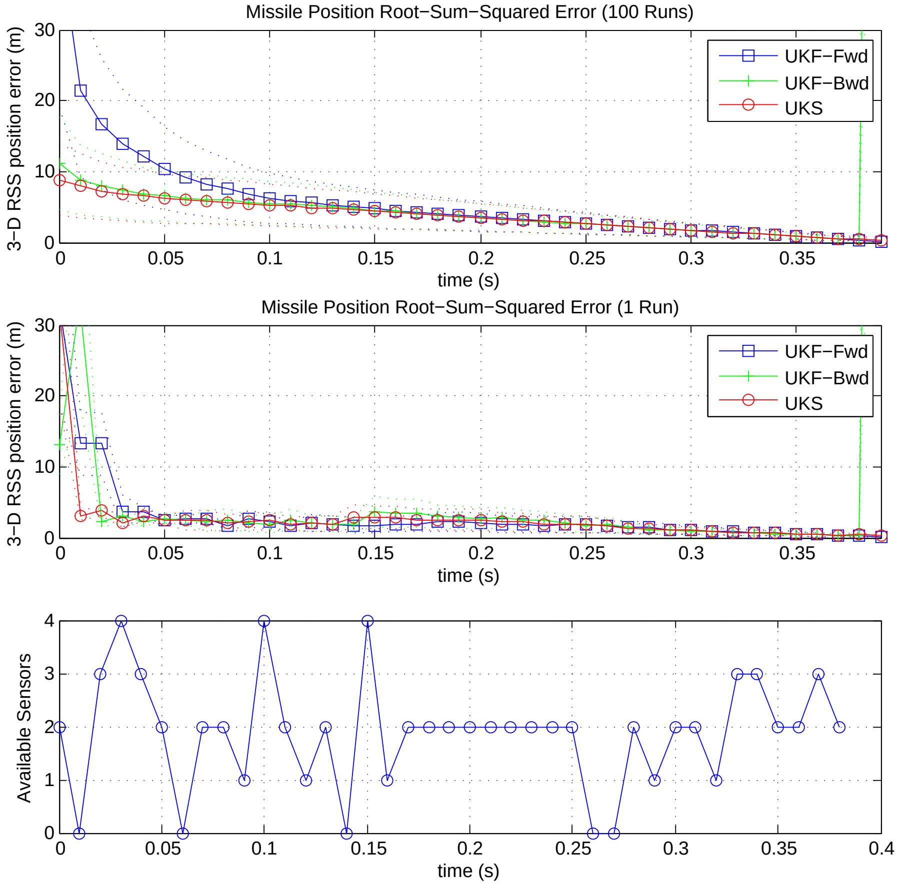


Figure 5.6: UKS Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with CA Dynamics Model (Scenario 3)

are shown in Figure 5.10. When the filters are properly tuned, the SFRA UKS and UKS perform on par once again. From these results, the UKS is recommended for use with post-flight live-fire missile test data for trajectory reconstruction and missile scoring.

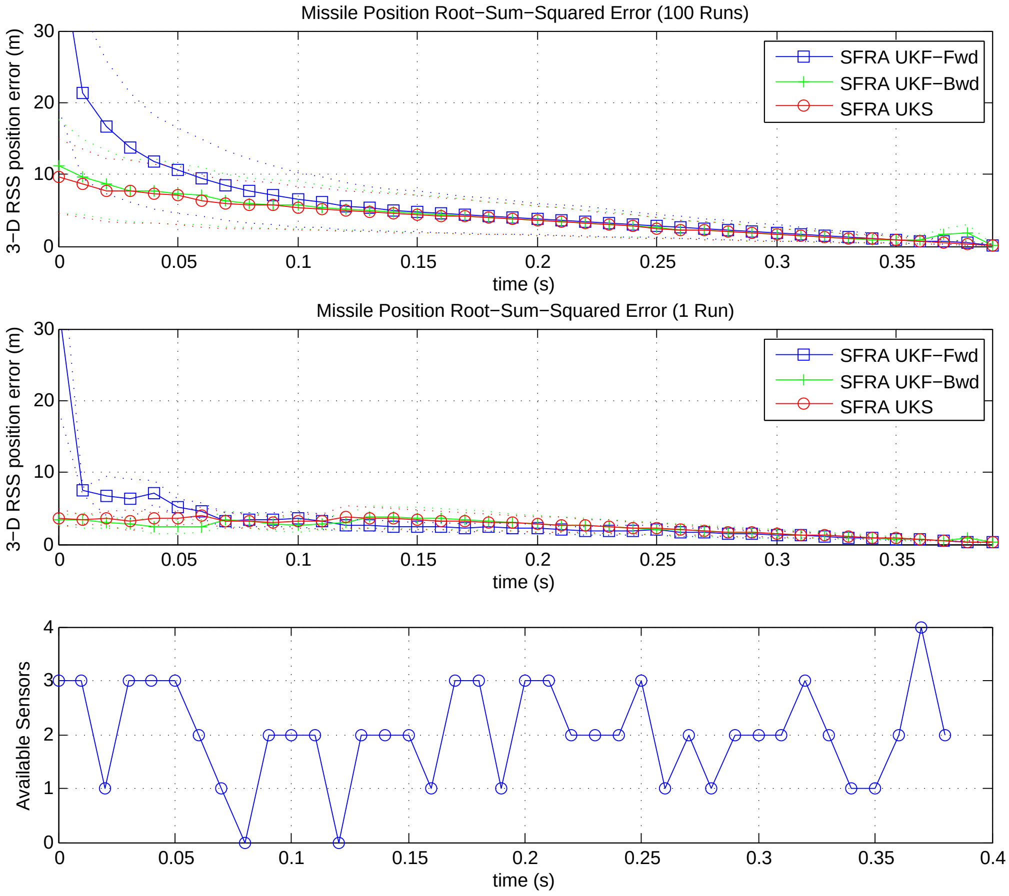


Figure 5.7: SFRA UKS Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with CA Dynamics Model (Scenario 3)

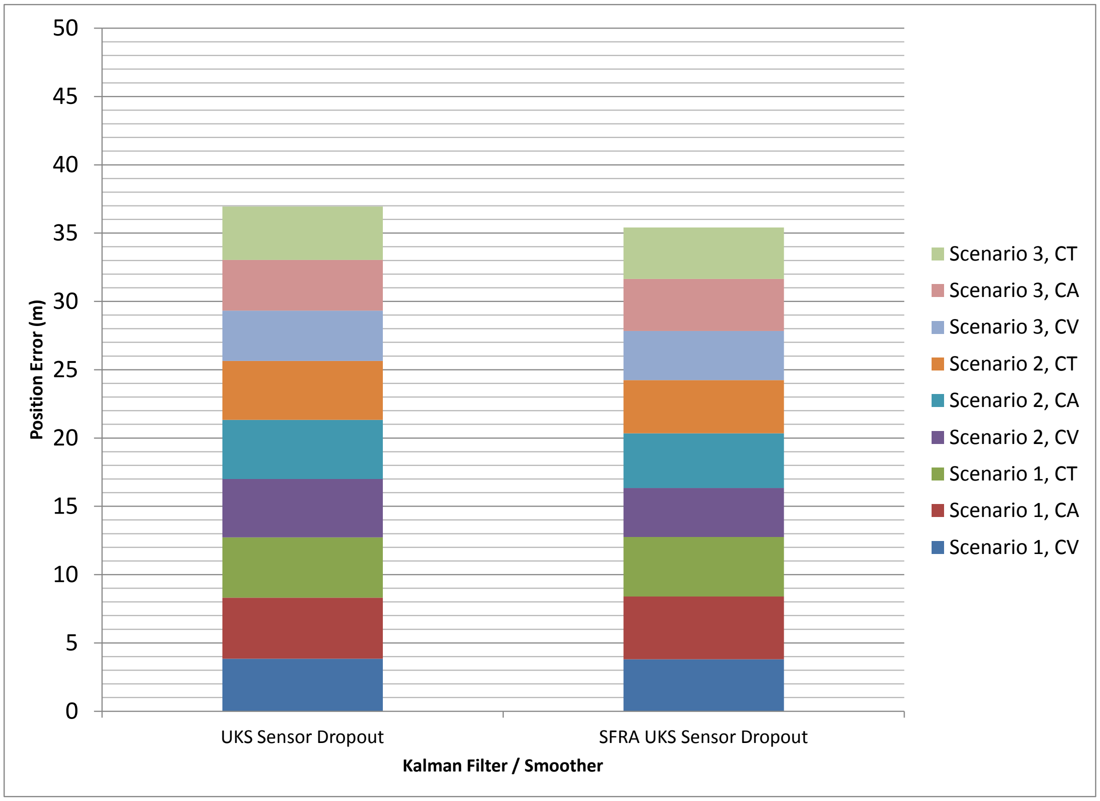


Figure 5.8: Accumulated Performance Metrics for UKS and SFRA UKS with Random Sensor Dropout

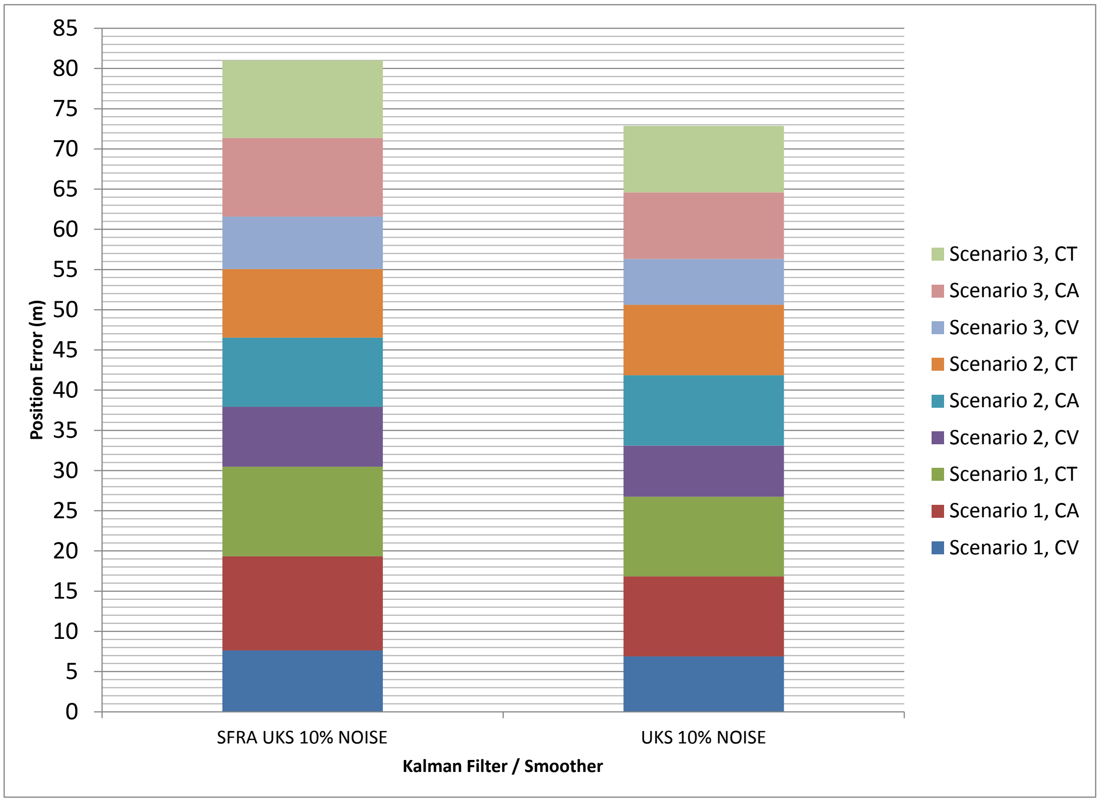


Figure 5.9: Accumulated Performance Metrics for UKS and SFRA UKS with 10% Noise - Untuned

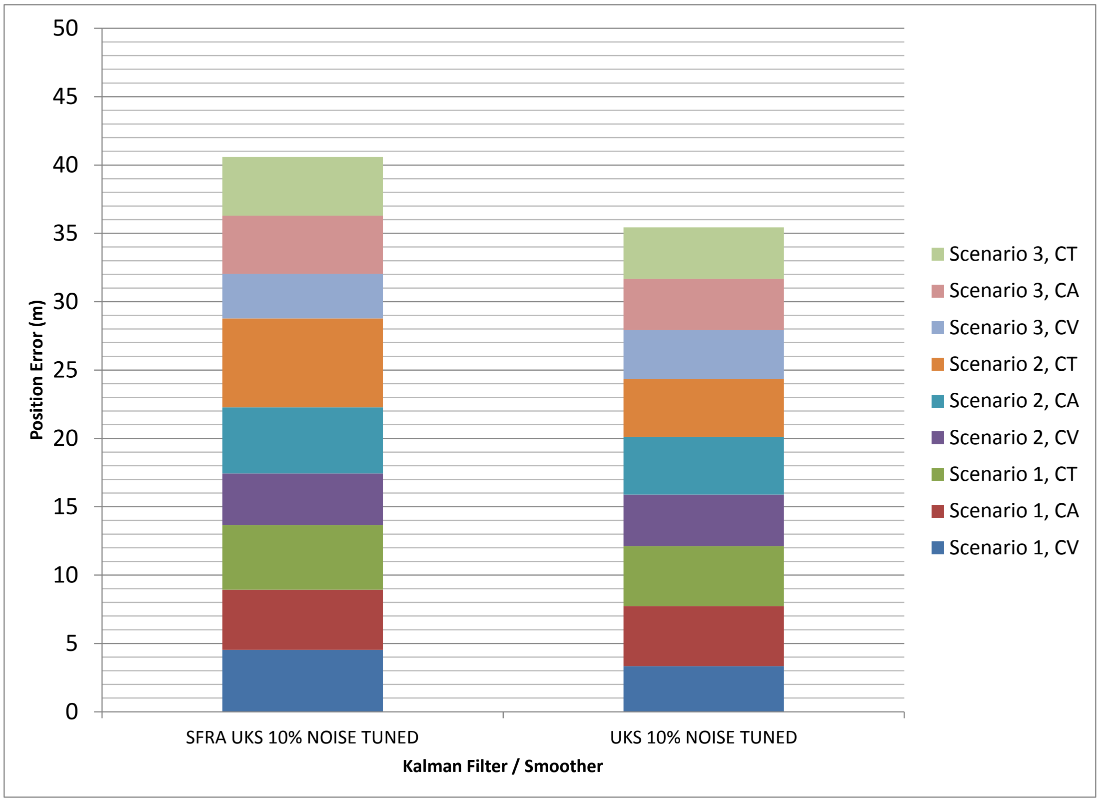


Figure 5.10: Accumulated Performance Metrics for UKS and SFRA UKS with 10% Noise - Tuned

VI. Conclusions and Recommendations

6.1 Summary of Results

This research uses both linearized and nonlinear fixed-interval smoothers to approximate non-linear effects. Although the three dynamics models are all linear, the measurement model has 2^{nd} order nonlinearities. Therefore it is expected that the nonlinear smoothers should outperform their linearized counterparts. This research has presented a myriad of Kalman filters, smoothers, dynamics models, and engagement scenarios to provide a comprehensive analysis.

The results of the UKS, SFRA UKS, EKS, SFRA EKS, and IEKS are shown in Figure 6.1. When examining the linear smoothers, the SFRA EKS provides a small 4.7% overall reduction in position error over the EKS. If we examine the results from the non-maneuvering engagment (scenario 1), the EKS is on par with SFRA EKS. As one might expect, the maneuver detection provided by the SFRA EKS enabled higher accuracy estimates during the highly dynamic scenarios (2-3). The IEKS provides an overall 9.5% position error reduction when compared to the standard EKS algorithm. When examining the nonlinear Kalman smoothers versus their linear counterparts, this research indicates a 56.9% reduction in expected position error (excluding the non-optimal IUKS).

A series of performance tests were conducted using UKS and SFRA UKS to investigate ability to handle mismatches between actual and simulated dynamics/sensor models. After increasing the sensor noise by a factor of 10, the UKS shows an overall 15.6% decrease in expected position error. The basic maneuver detection in the SFRA UKS relies on residual monitoring and is more likely to mistake bad measurements for abrupt changes in trajectory. This results in less accuracy, especially when the smoother is not tuned properly. Due to its versatility and superb stability, the Unscented Kalman Smoother (UKS) is recommended as the DoD/USAF standard for post-processing and scoring live-fire missile data.

6.2 Future Work

Future pursuits in the realm of missile vector scoring and trajectory reconstruction could include the development and simulation of a fixed-interval particle-based smoother. Considerations to keep in mind would be creative particle regeneration and the use of more advanced system models. It is hoped that the UKS algorithm is used with real-world missile test data and validated against the current USAF/DoD performance standard. There is much to be gleaned from Kalman smoother-based post-flight missile trajectory reconstruction.

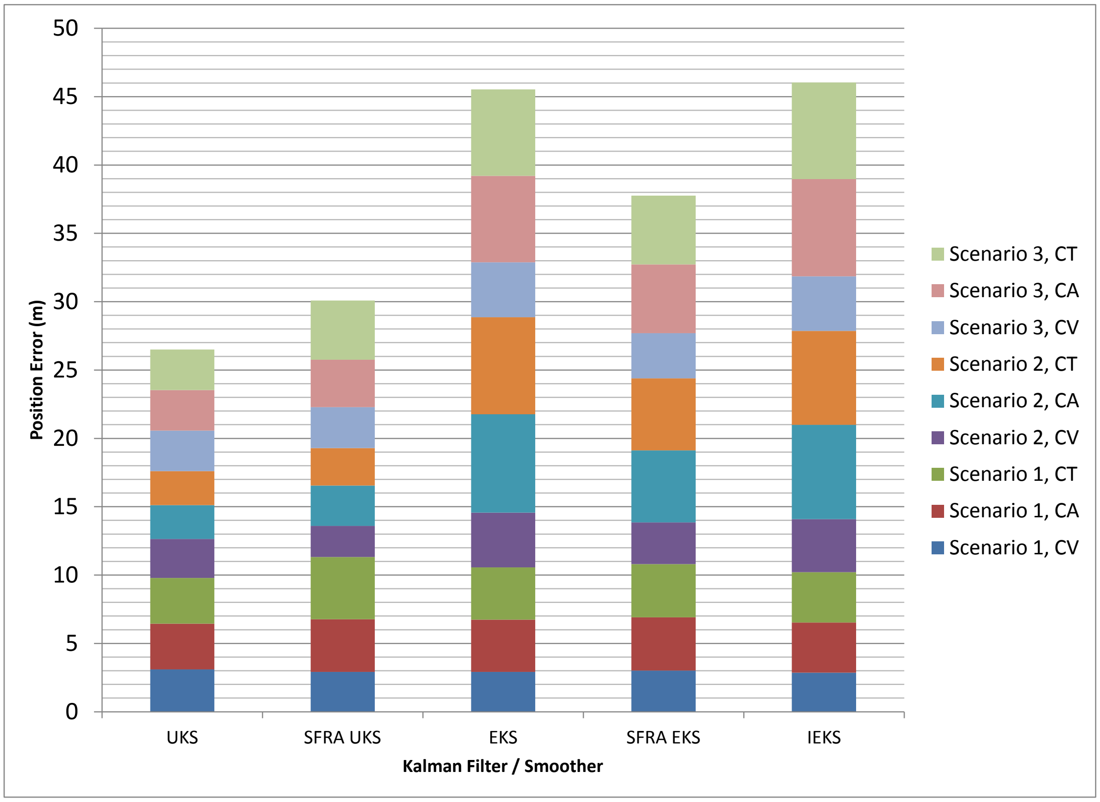


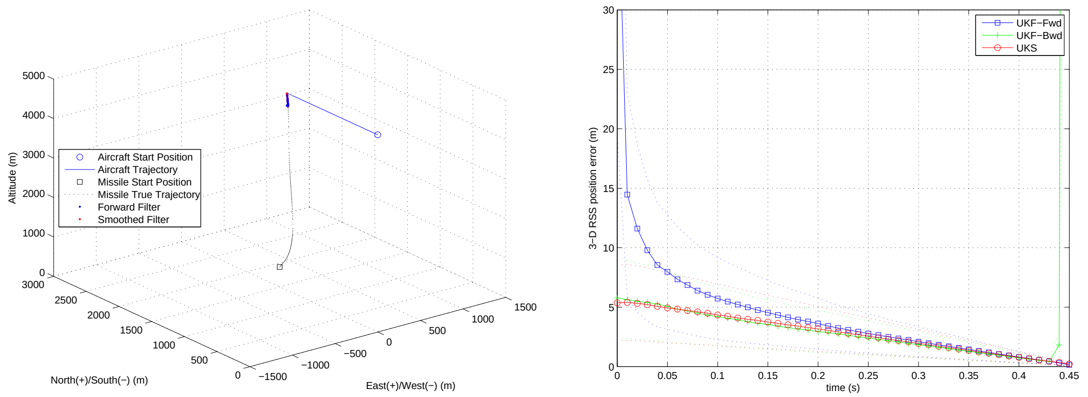
Figure 6.1: Accumulated Performance Metrics for Kalman Smoothers excluding IUKS

Appendix A. Simulation Results

A.1 Introduction

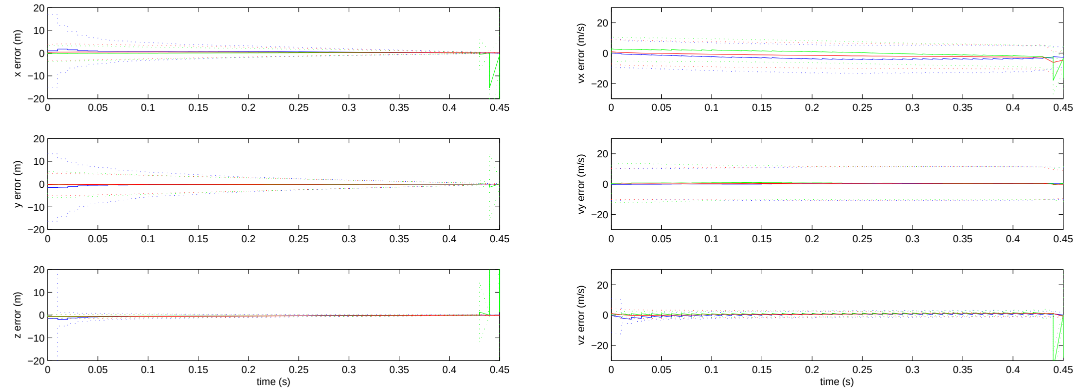
Appendix A contains the results of all 126 simulations performed for this research. The *a* plot (top left) shows 3D aircraft and missile trajectories. The *b* plot (top right) shows the 100-run monte carlo mean RSS position error. The *c* and *d* plots show 100-run monte carlo mean error and error standard deviation for missile position and velocity states, respectively. Sections A.2, A.3, and A.4 contain the UKS, SFRA UKS, and IUKS results. Sections A.5, A.6, and A.7 contain the EKS, SFRA EKS, and IEKS results. Sections A.8, and A.9, contain the UKS and SFRA UKS results with tuning for increased 10% sensor noise. Sections A.10, and A.11, contain the UKS and SFRA UKS results for increased 10% sensor noise without tuning. Sections A.12, and A.13, contain the UKS and SFRA UKS results with random sensor dropout.

A.2 Unscented Kalman Smoother Simulations



(a) 3D Aircraft and Missile Trajectory

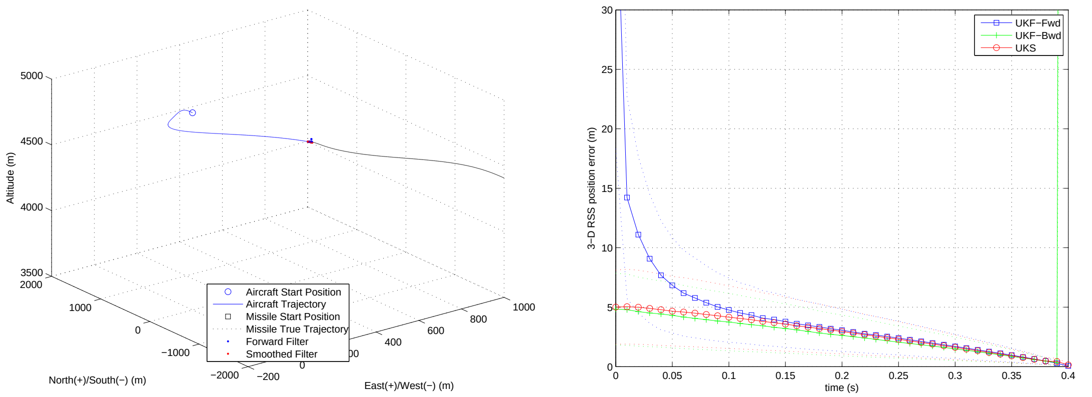
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

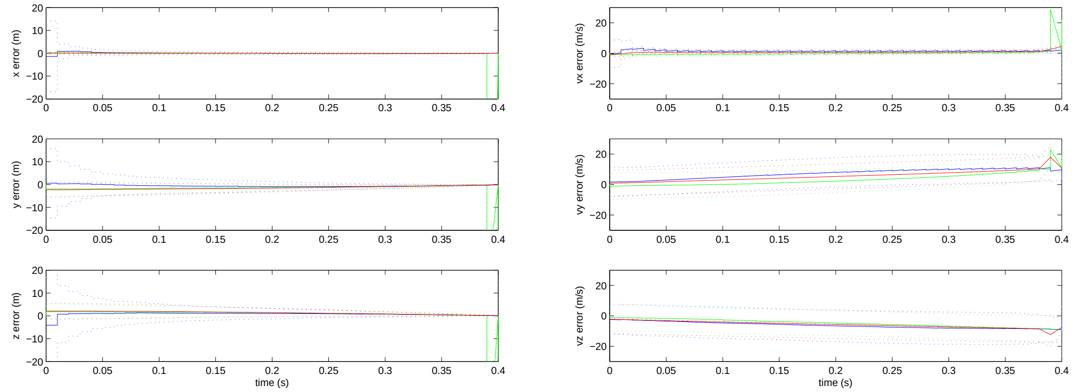
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.1: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

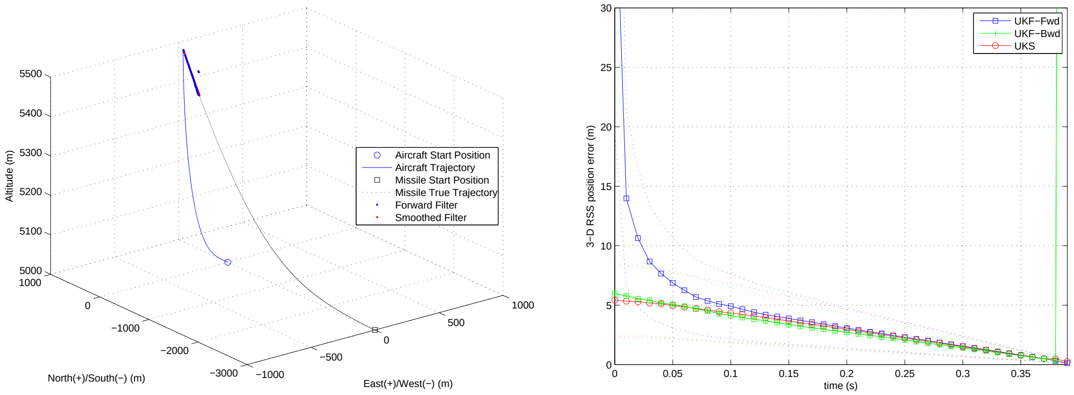
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

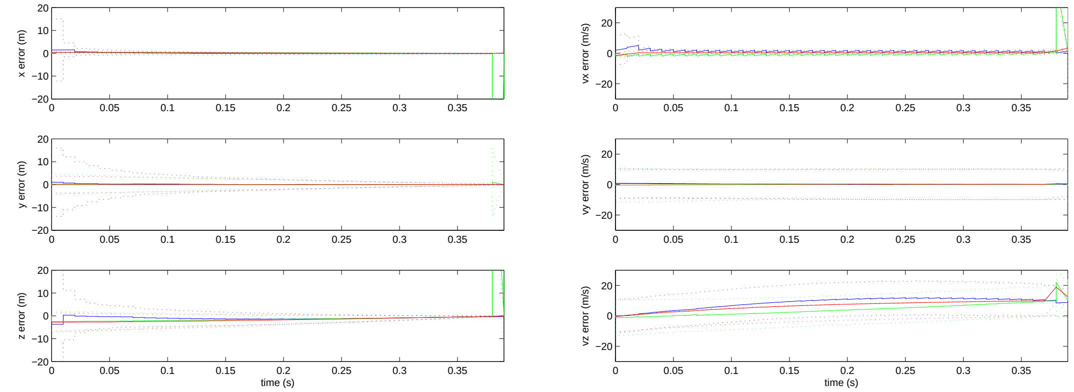
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.2: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

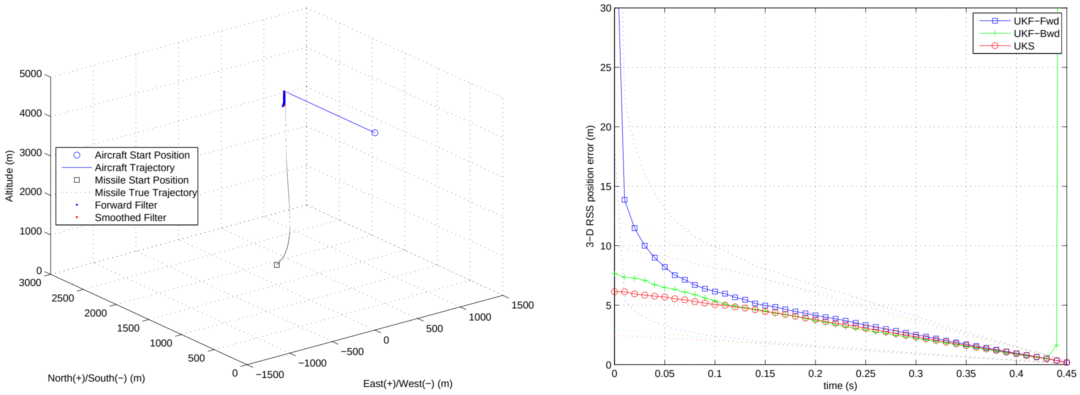
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

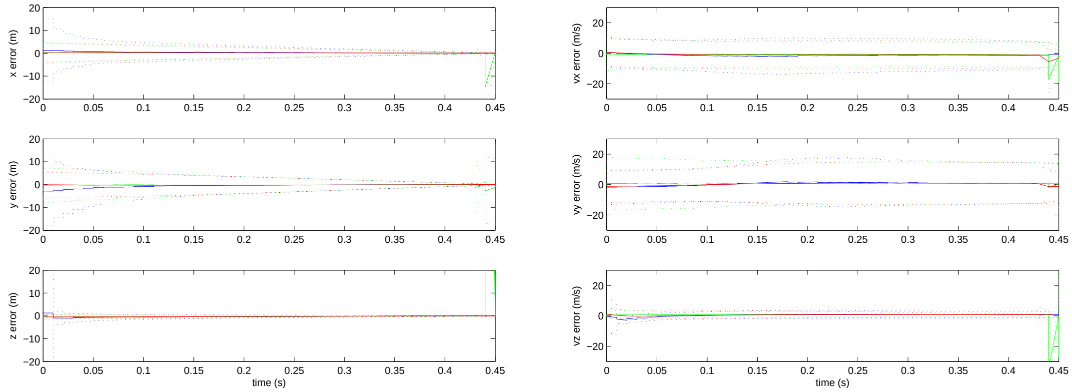
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.3: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

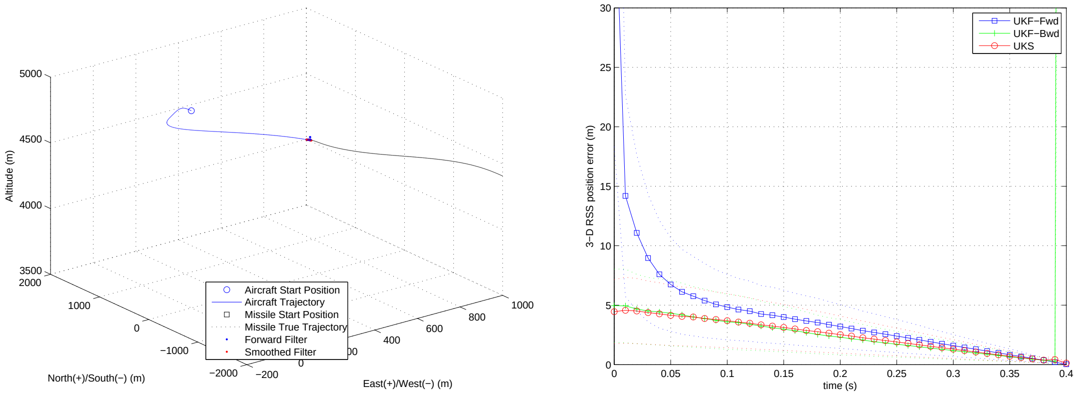
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

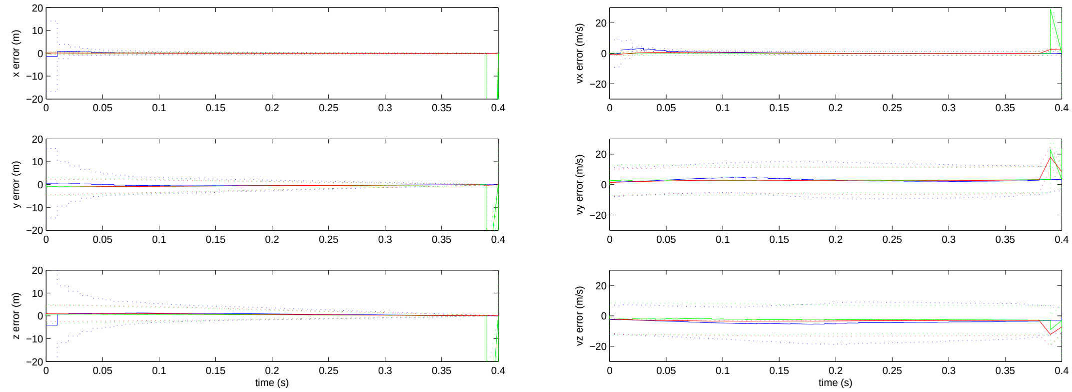
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.4: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

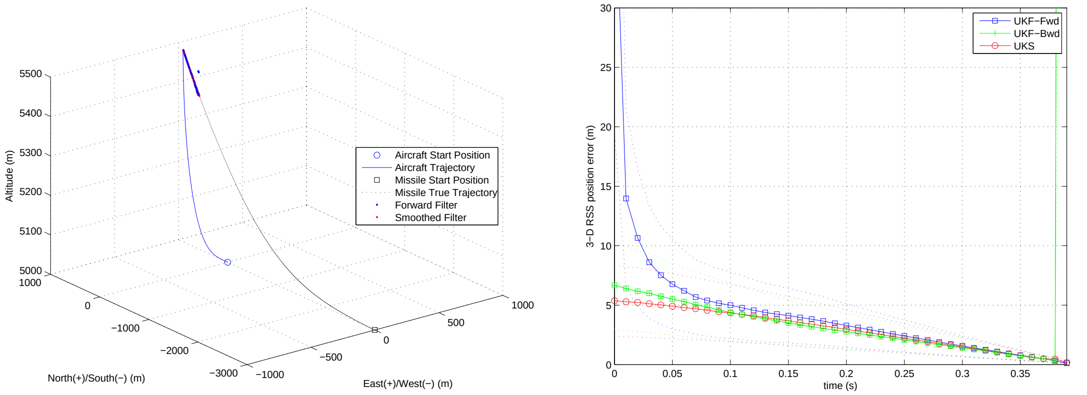
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

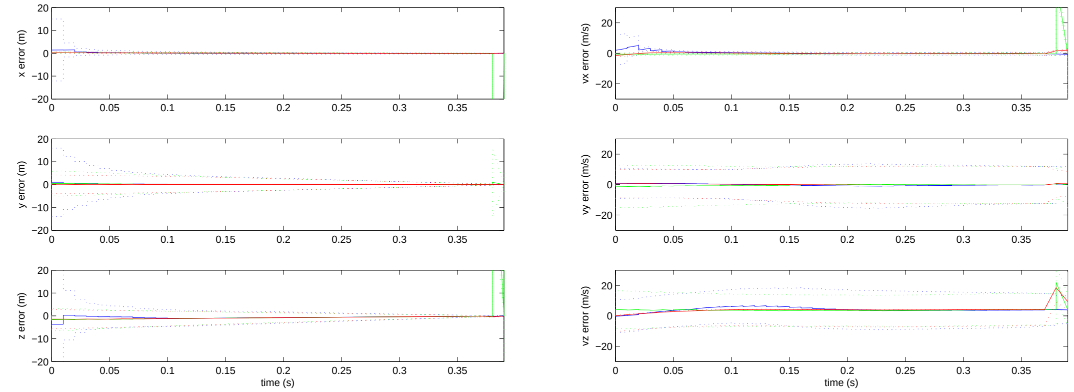
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.5: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

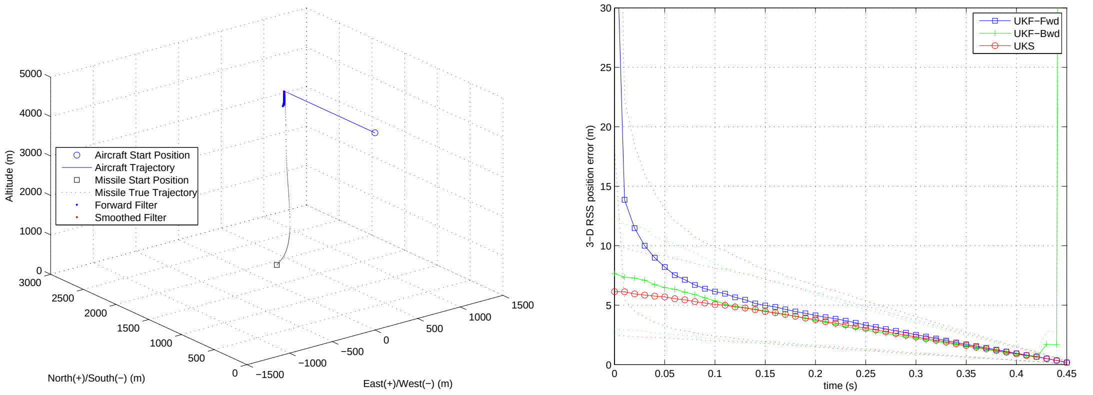
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

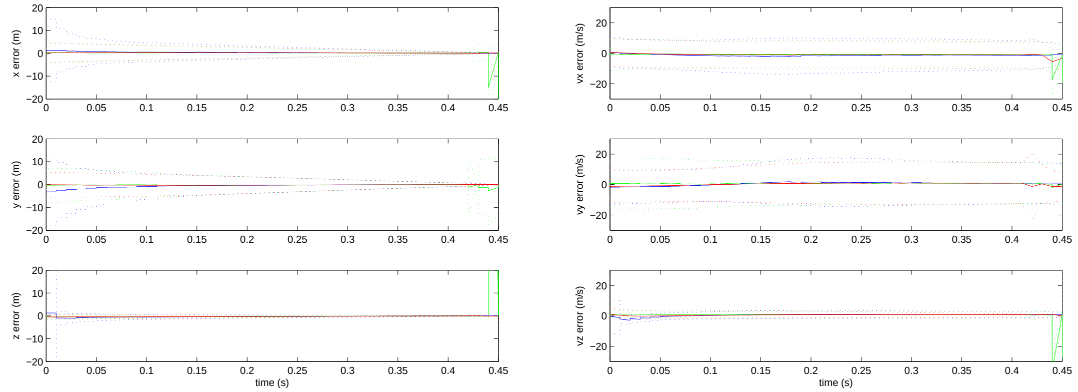
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.6: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

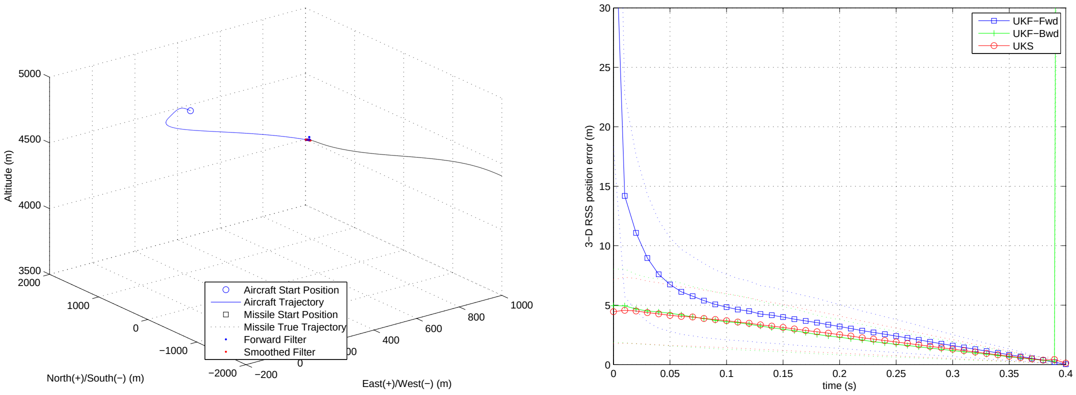
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

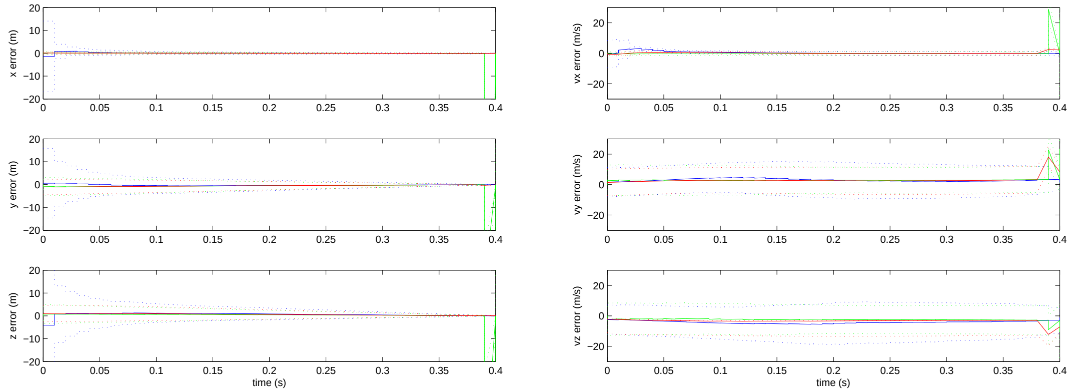
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.7: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

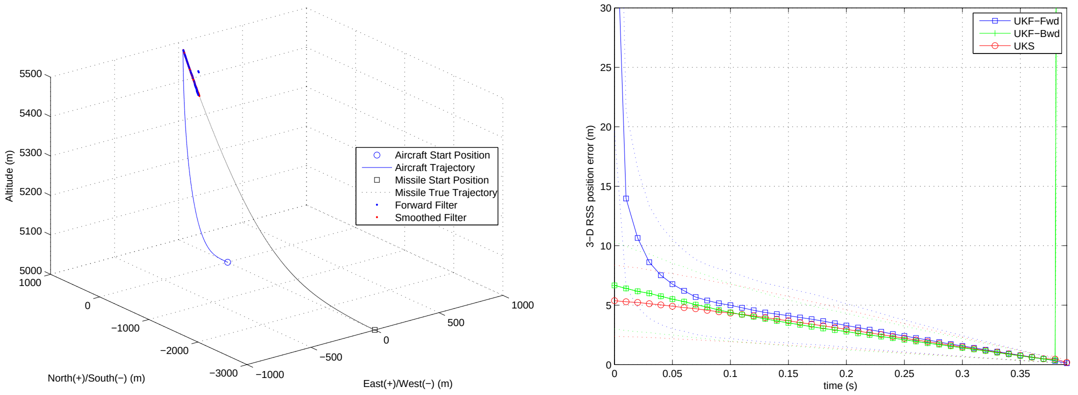
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

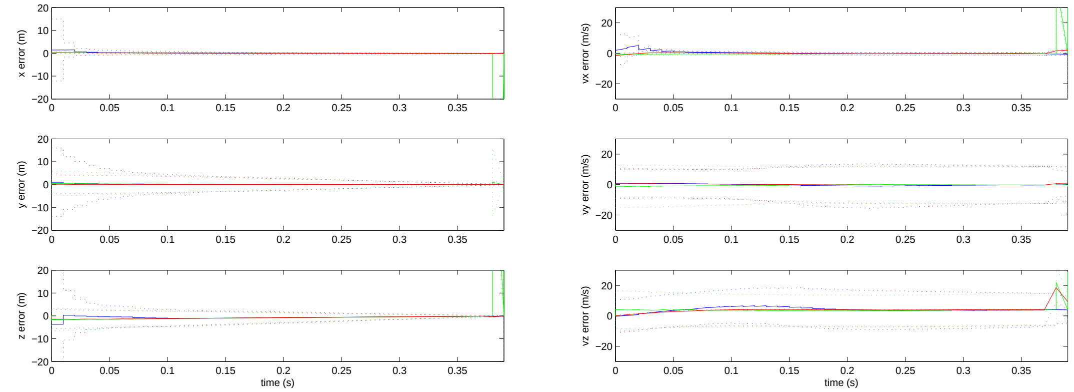
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.8: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

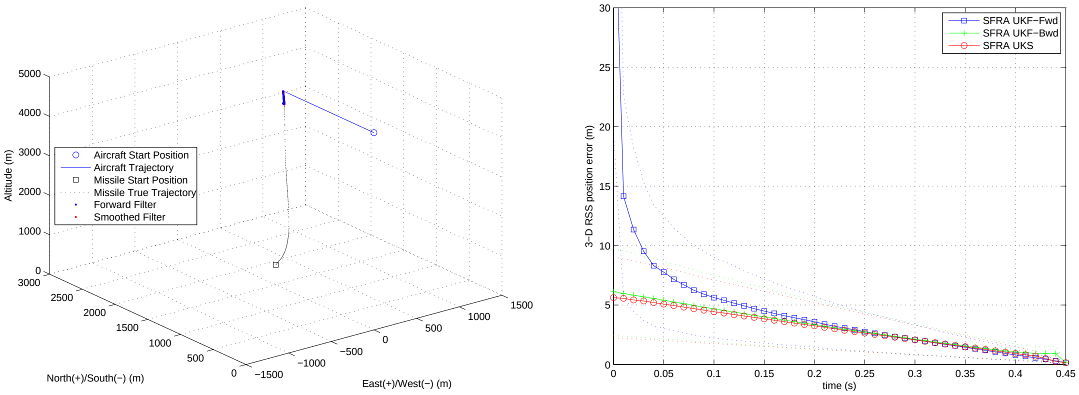


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

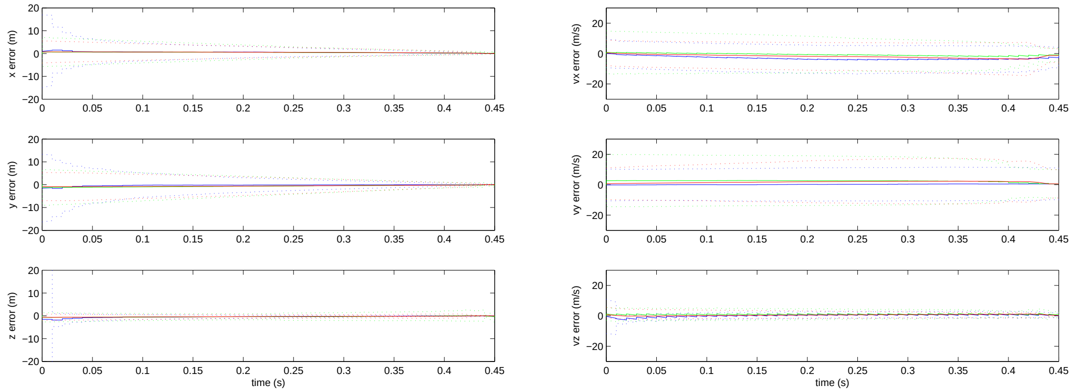
Figure A.9: Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.3 SFRA Unscented Kalman Smoother Simulations



(a) 3D Aircraft and Missile Trajectory

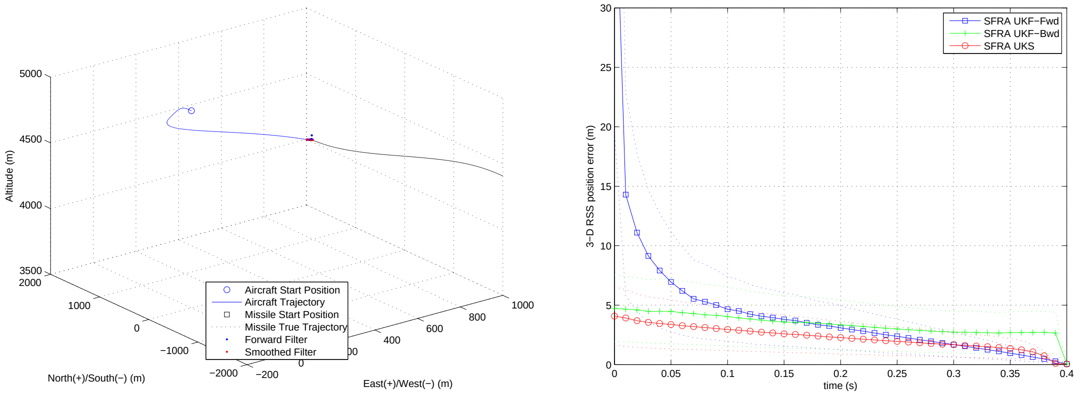
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

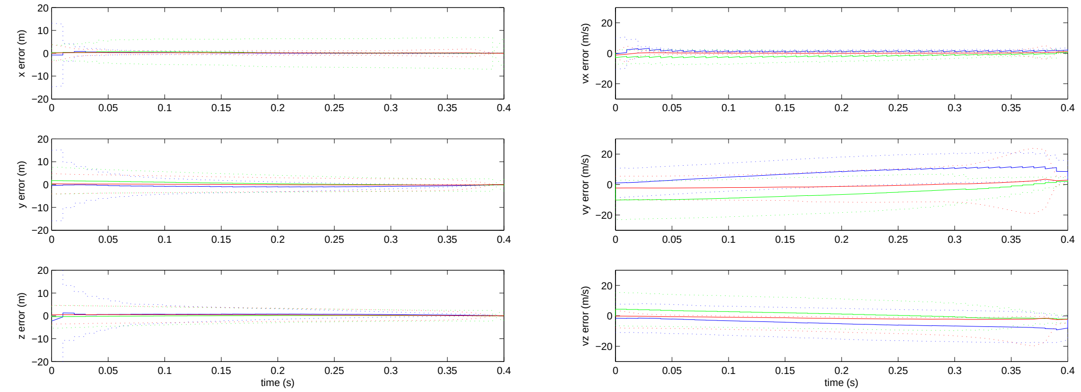
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.10: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

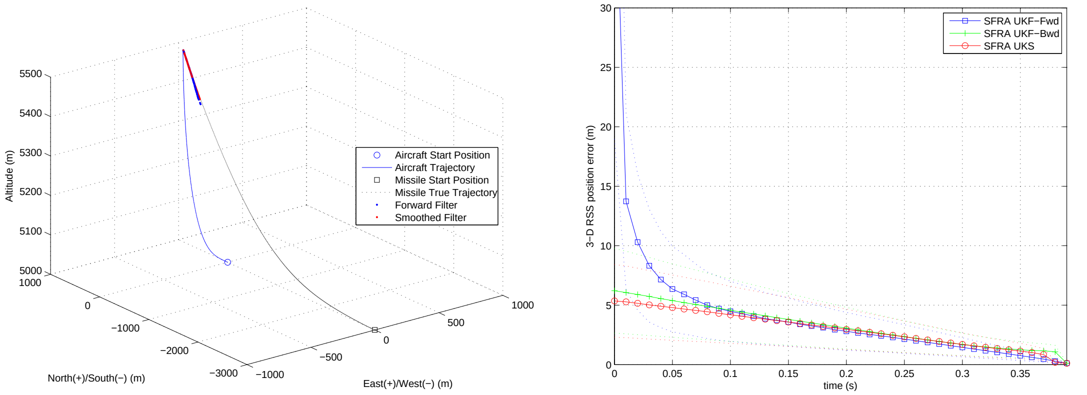
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

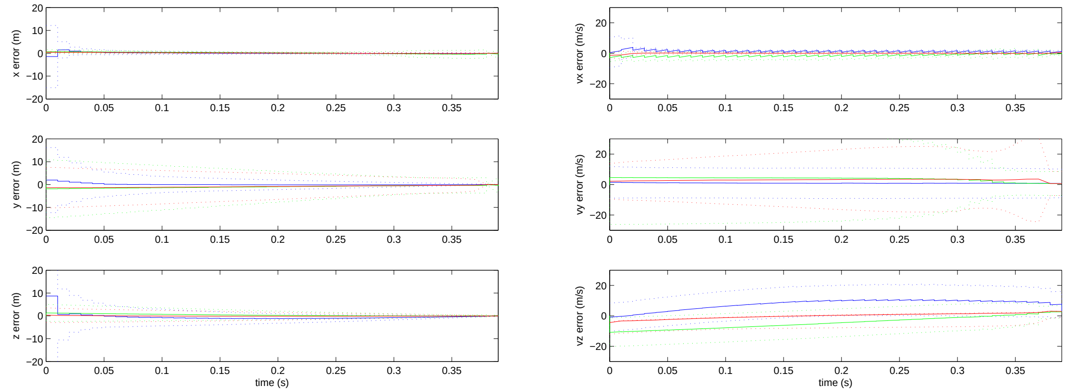
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.11: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

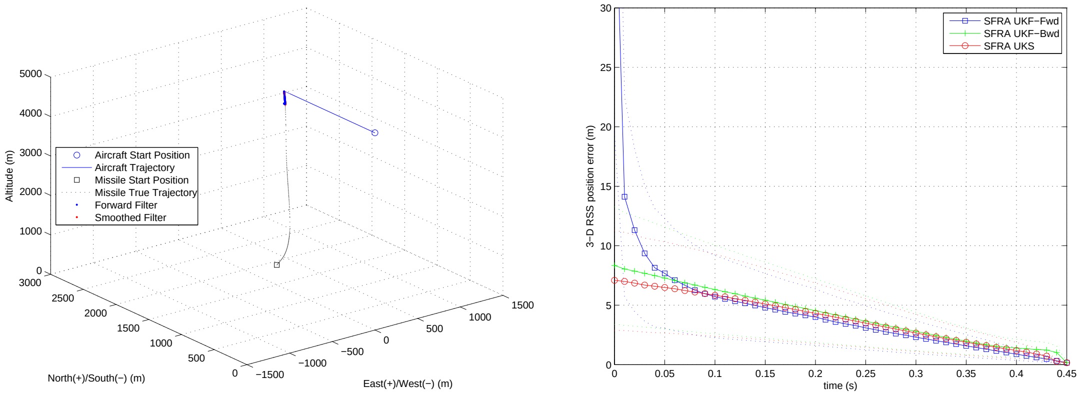
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

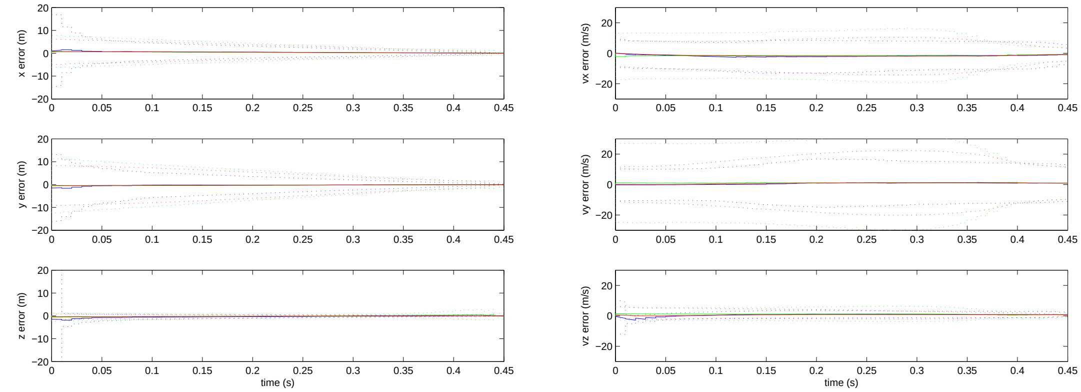
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.12: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

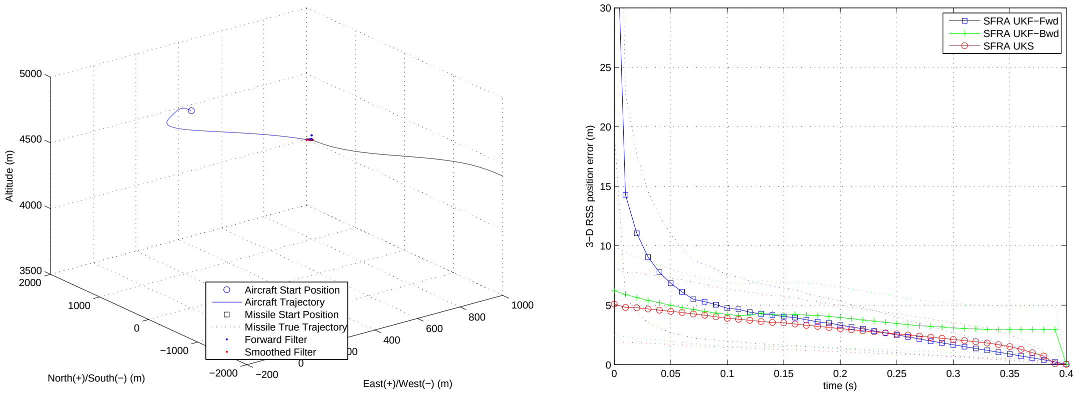
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

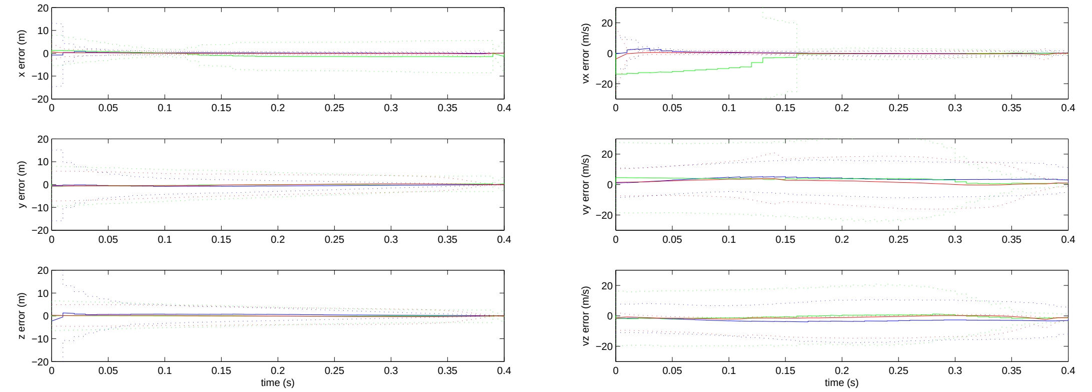
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.13: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

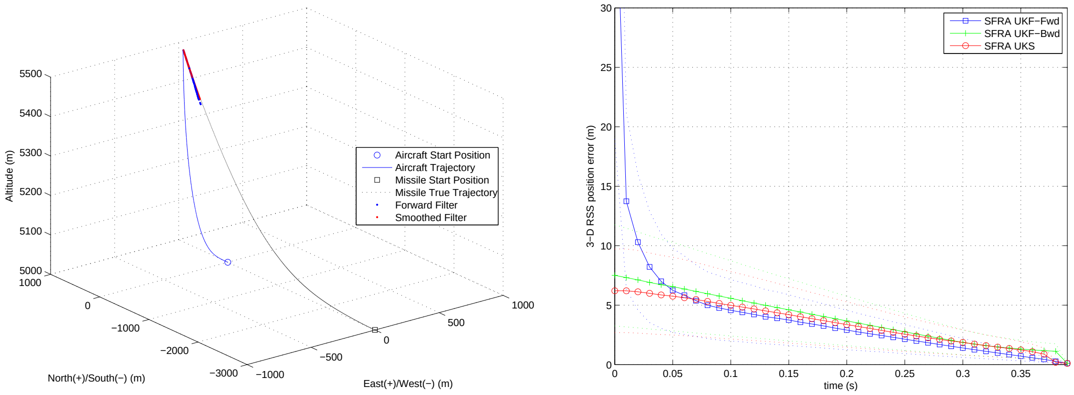
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

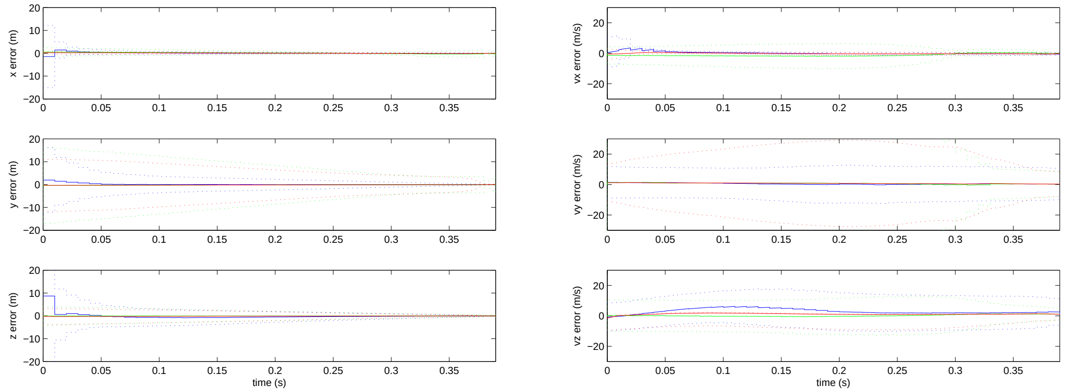
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.14: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

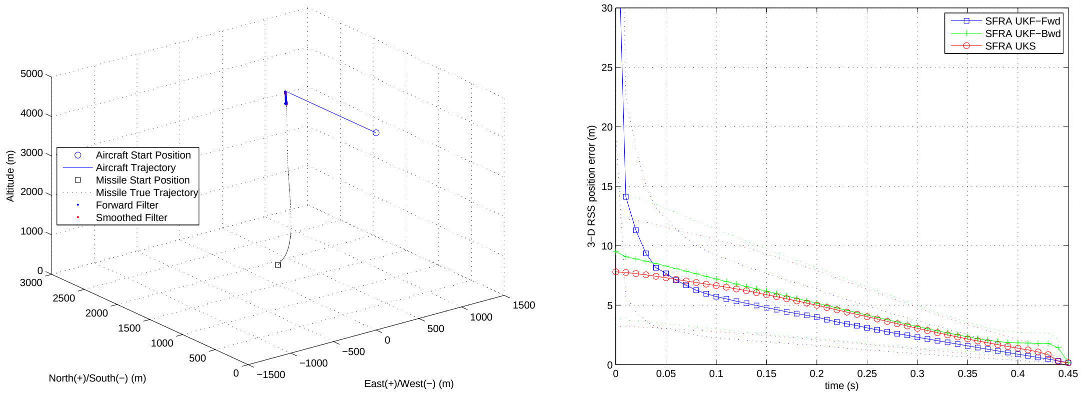
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

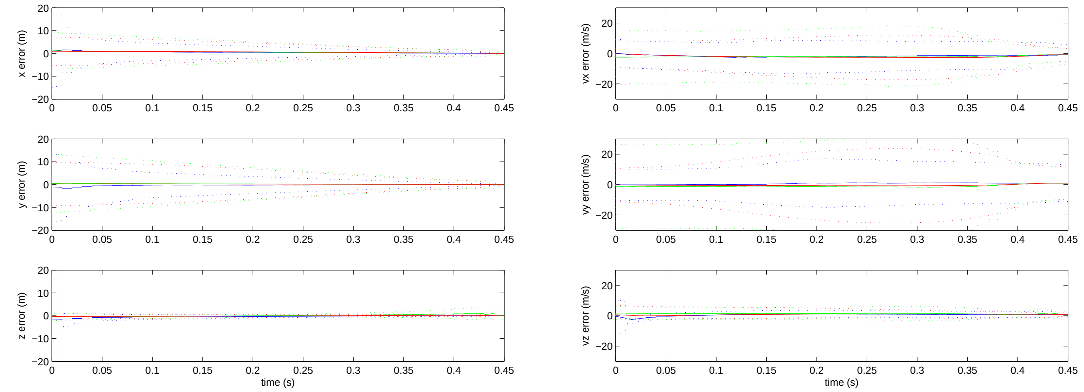
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.15: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

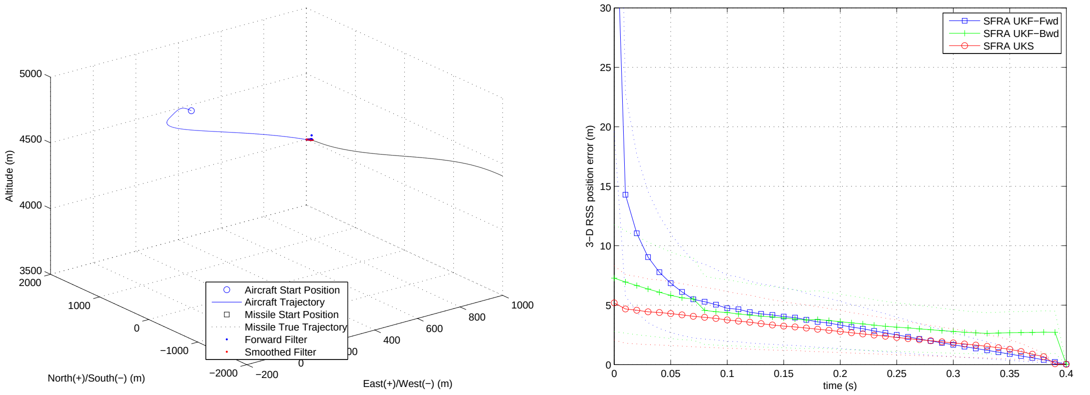
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

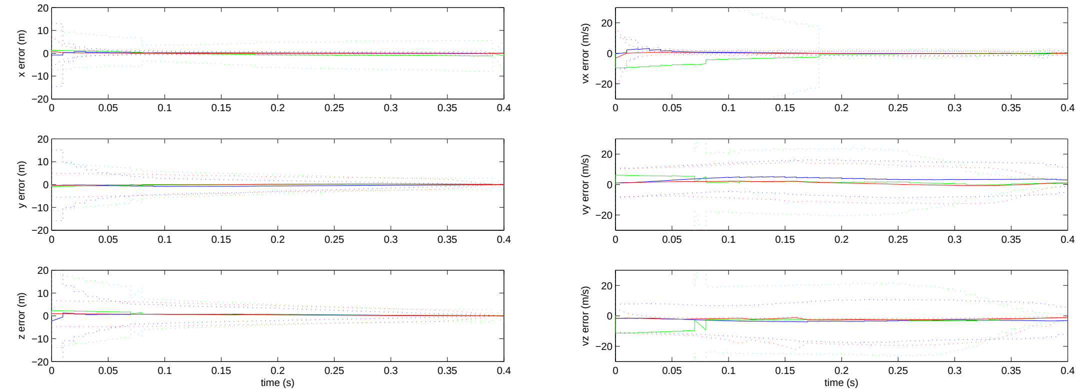
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.16: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

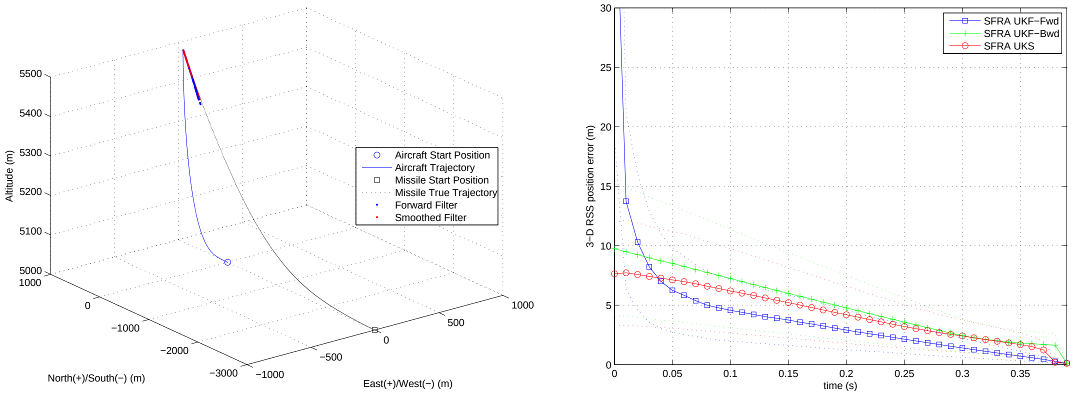
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

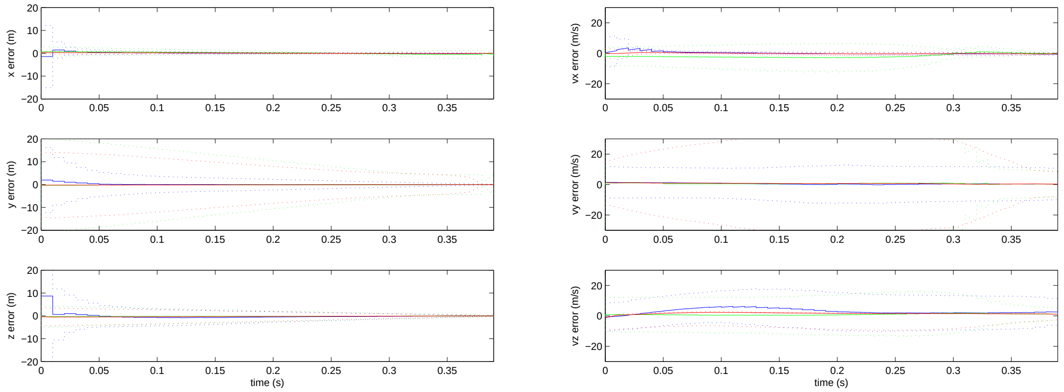
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.17: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

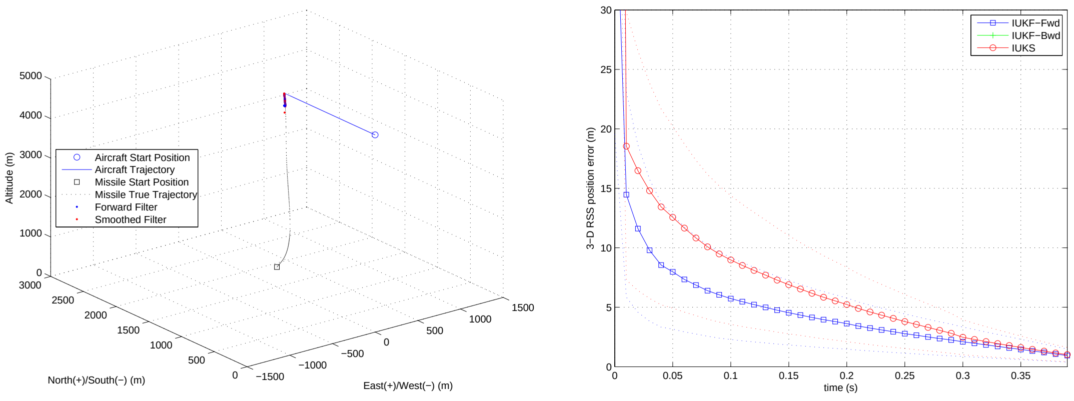


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

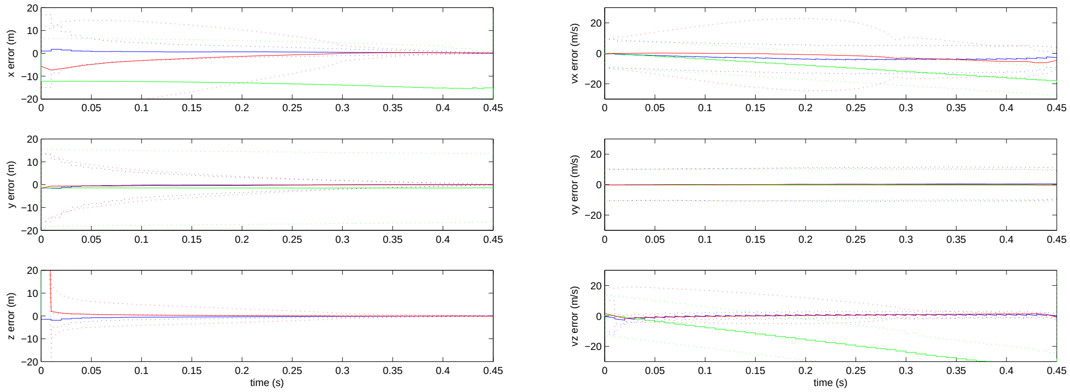
Figure A.18: SFRA Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.4 Iterated Unscented Kalman Smoother Simulations



(a) 3D Aircraft and Missile Trajectory

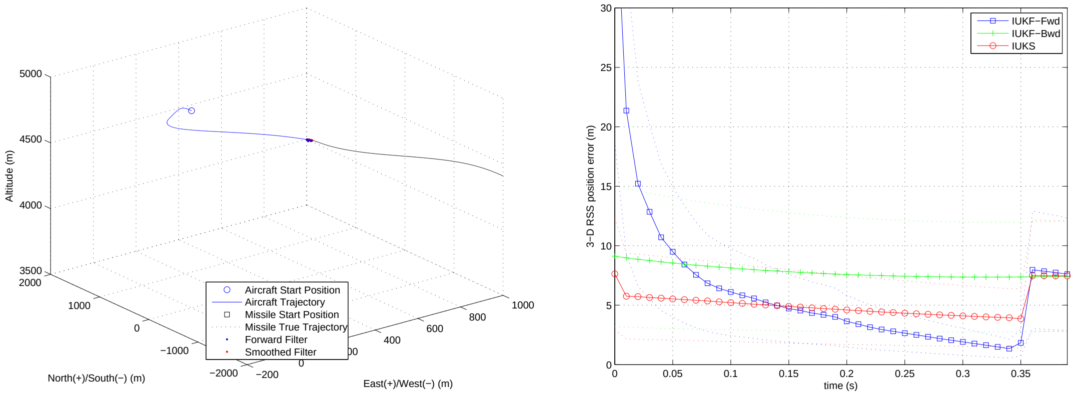
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

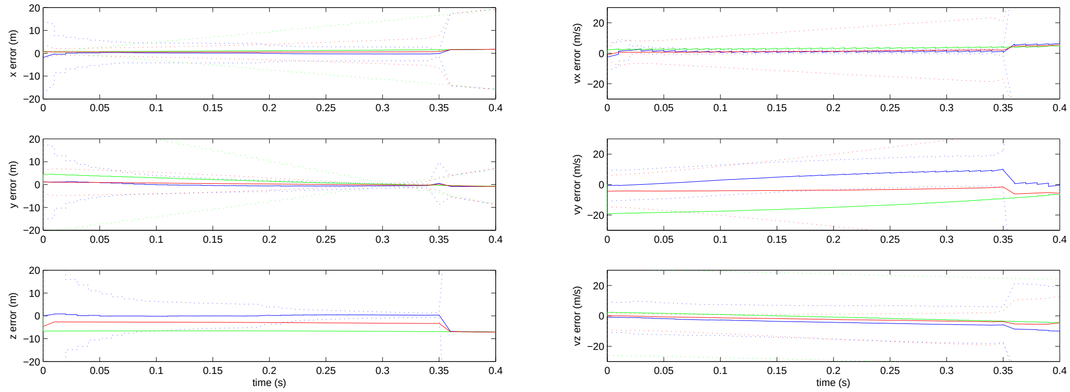
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.19: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

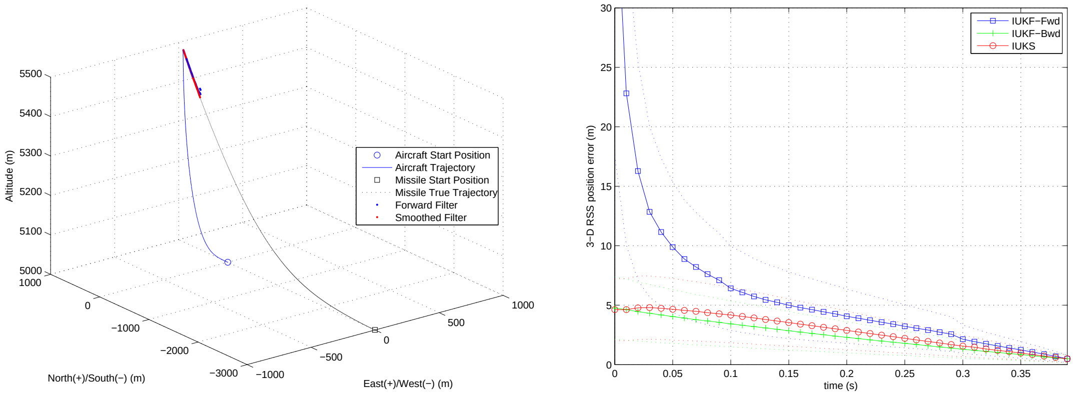
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

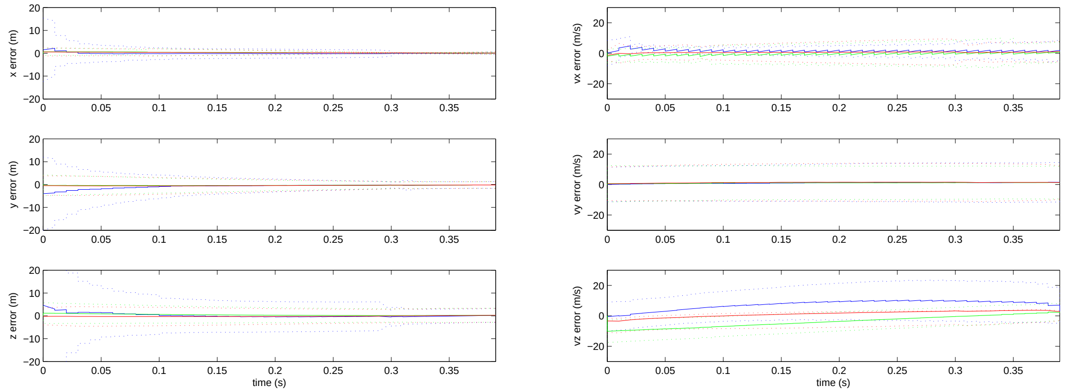
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.20: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

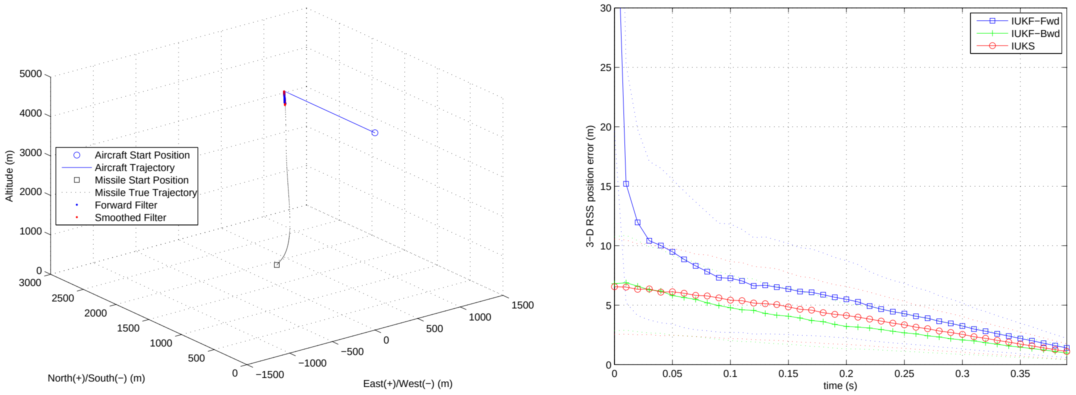
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

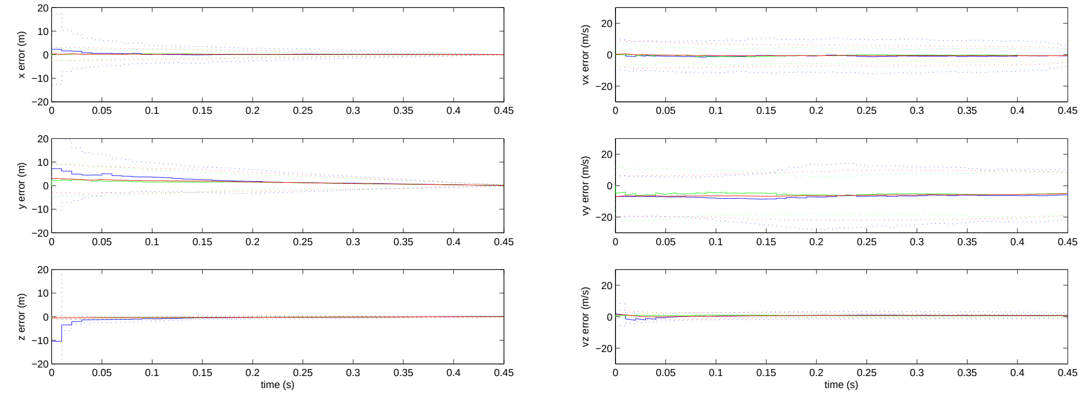
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.21: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

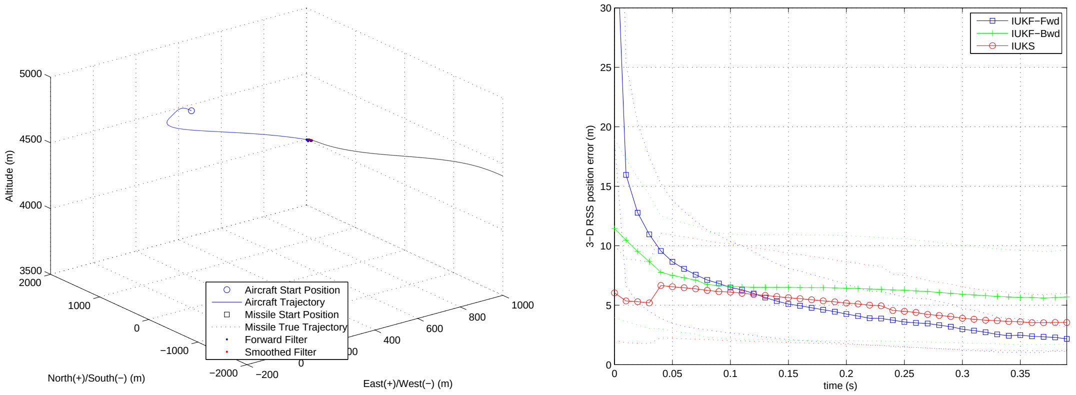
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

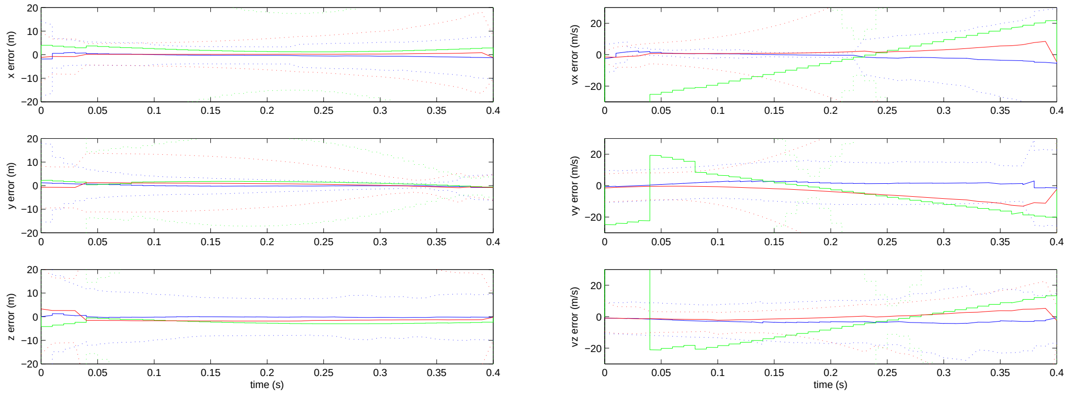
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.22: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

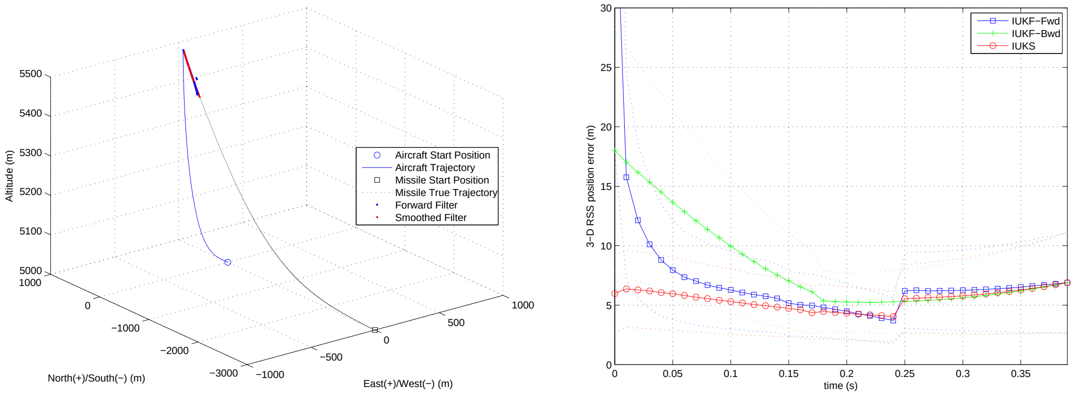
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

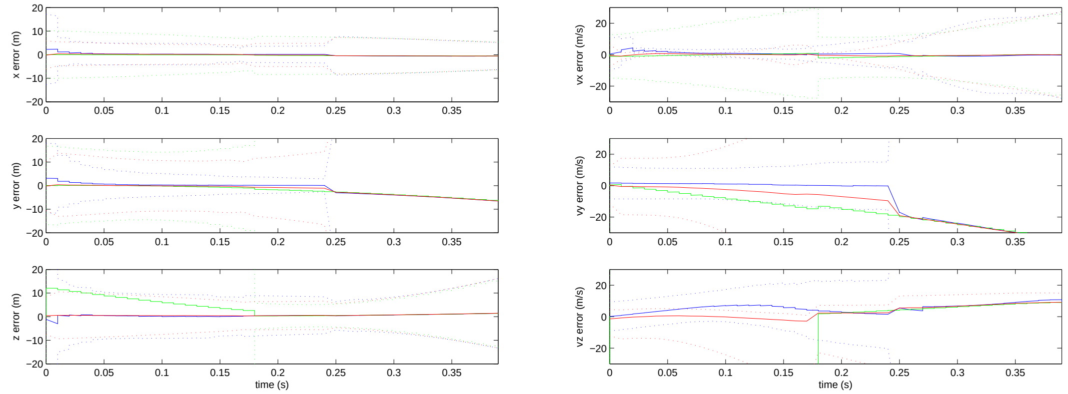
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.23: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

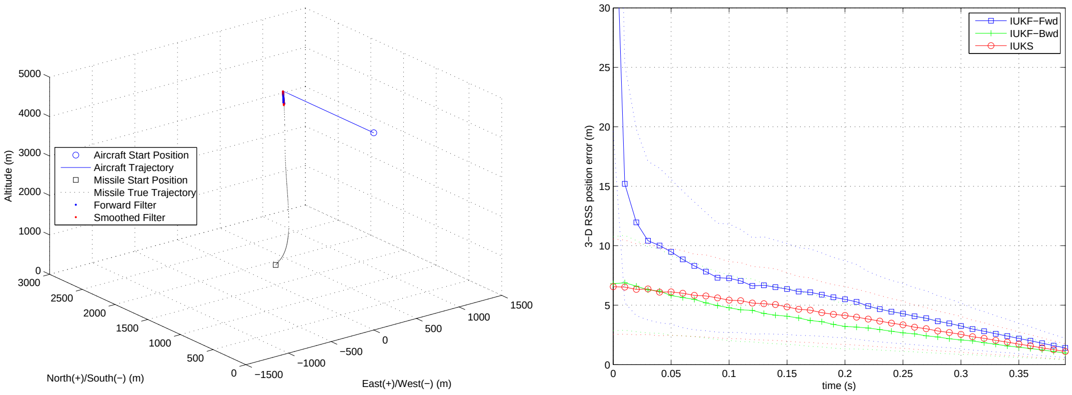
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

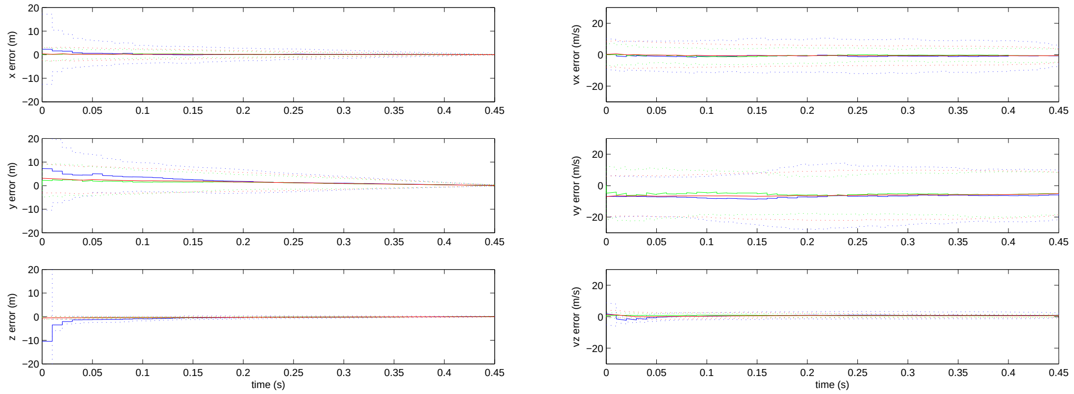
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.24: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

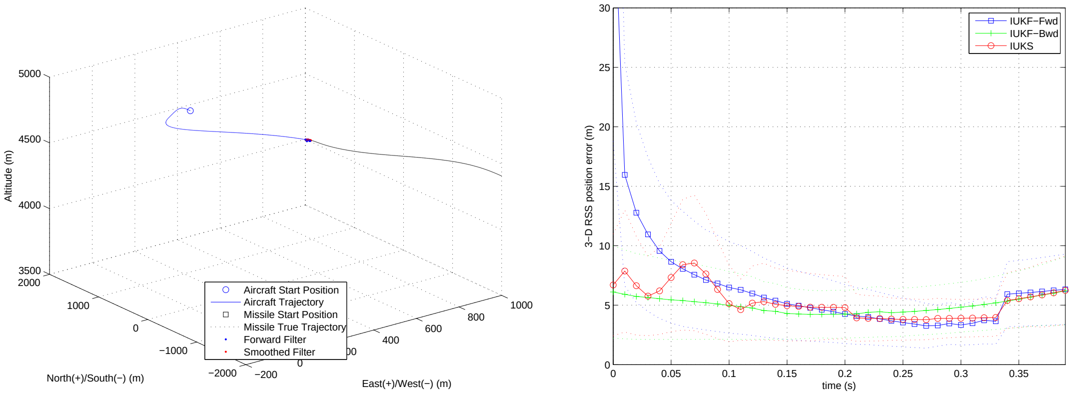
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

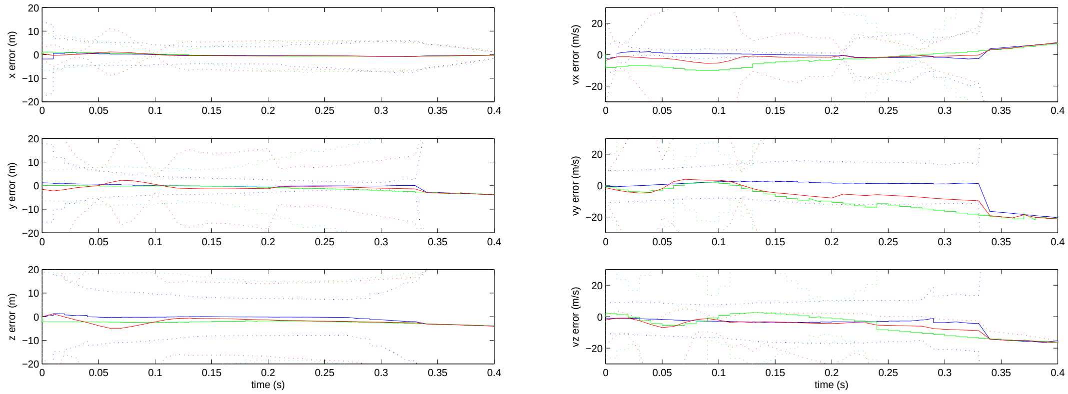
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.25: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

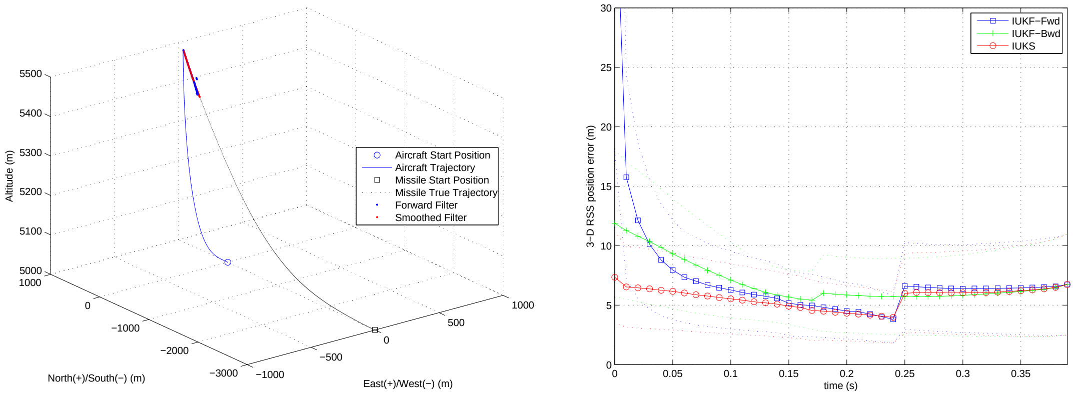
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

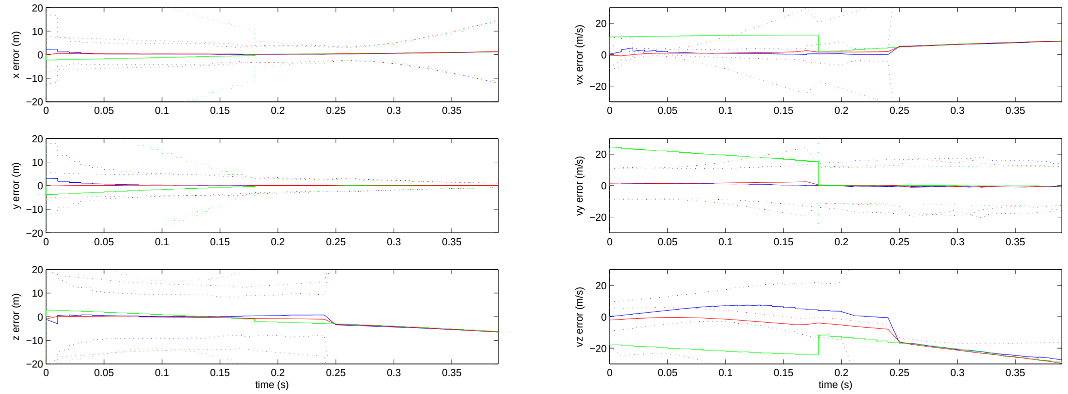
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.26: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

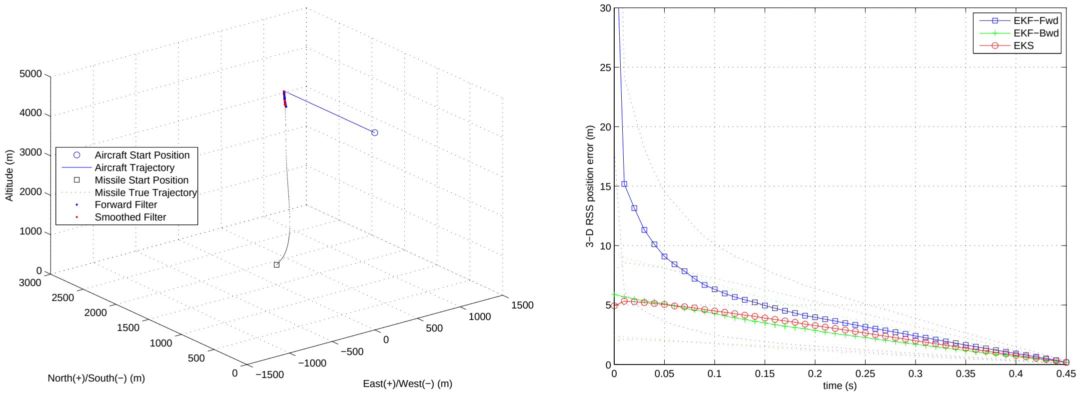


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

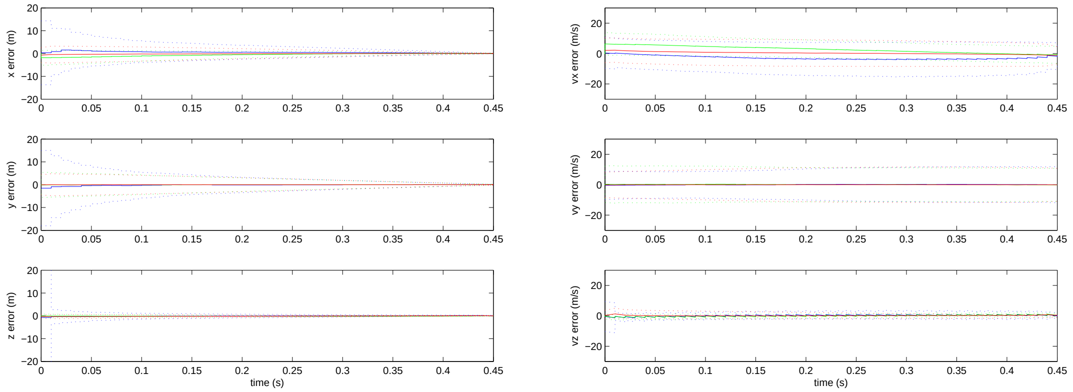
Figure A.27: Iterated Unscented Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.5 Extended Kalman Smoother Simulations



(a) 3D Aircraft and Missile Trajectory

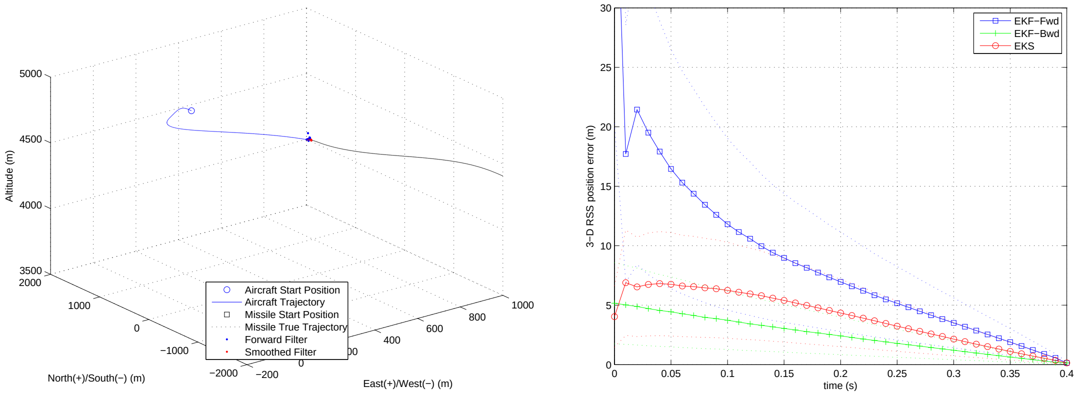
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

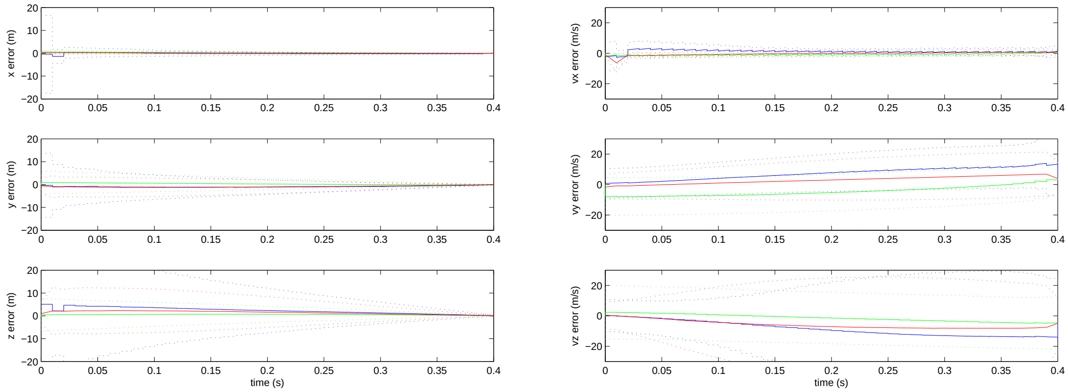
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.28: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

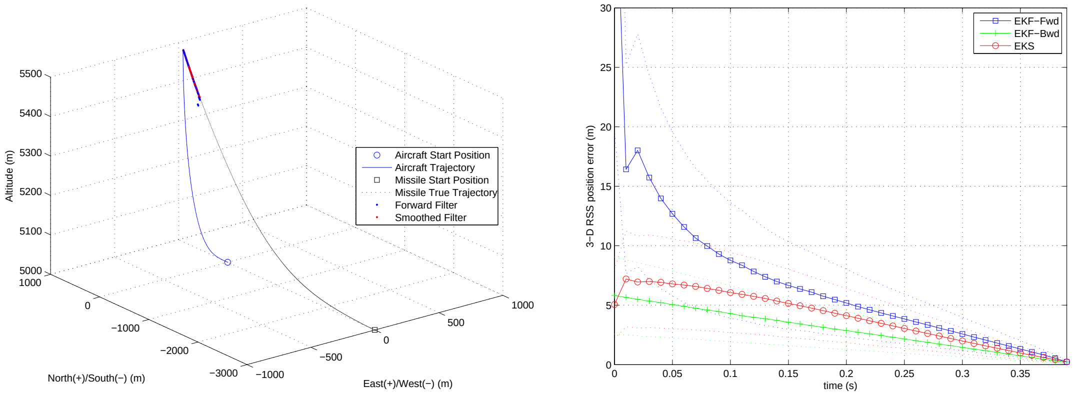
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

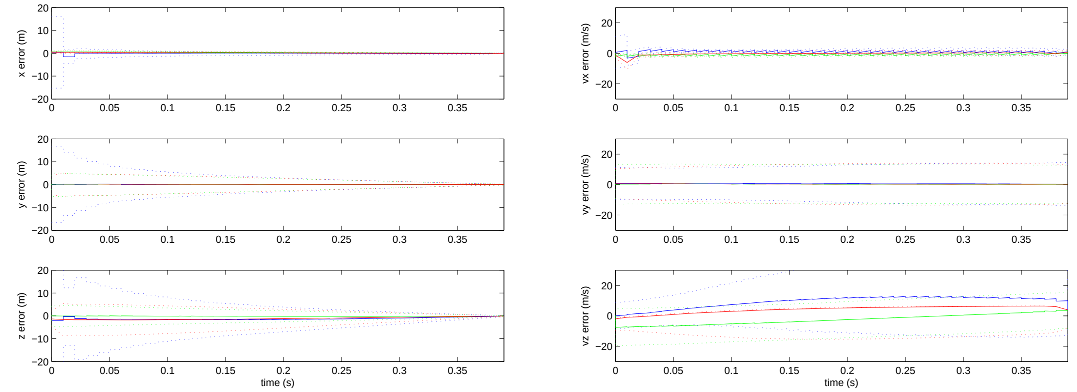
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.29: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

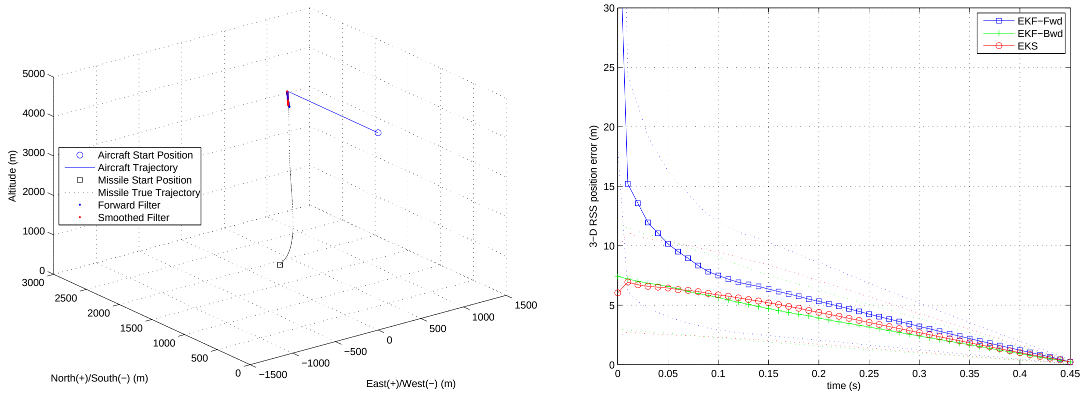
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

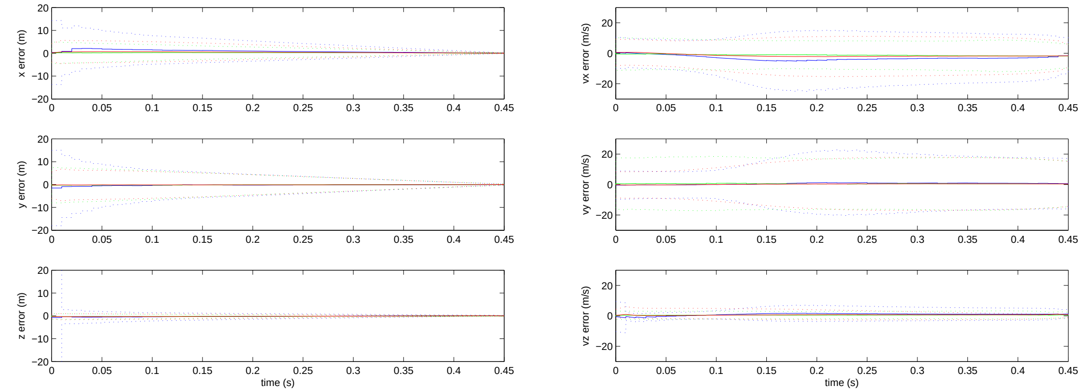
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.30: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

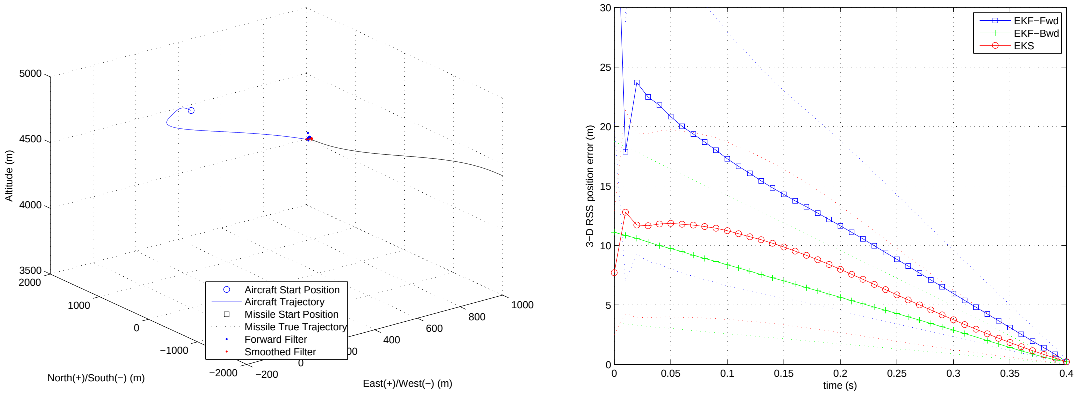
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

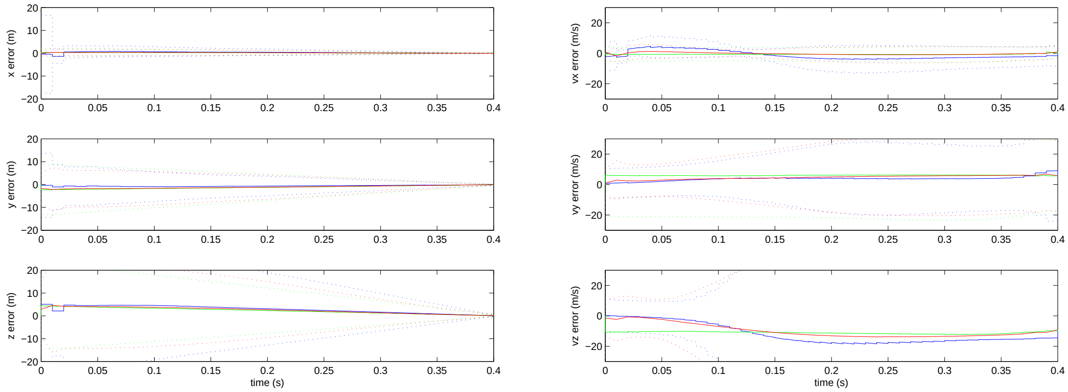
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.31: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

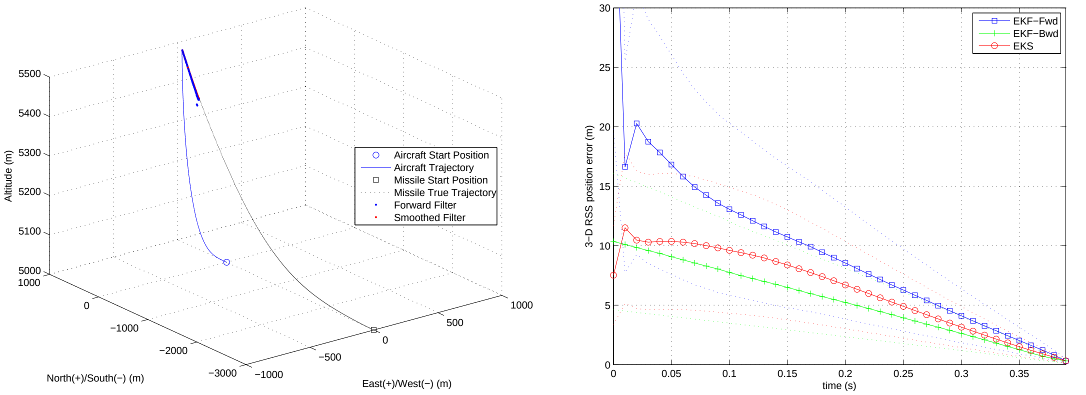
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

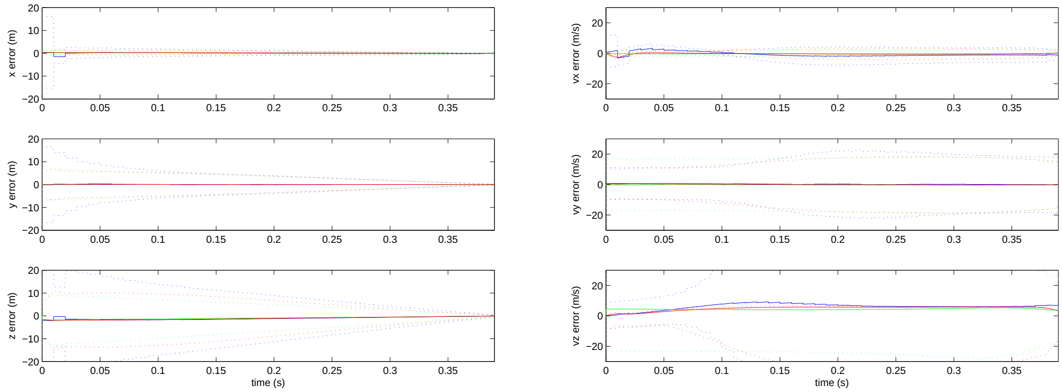
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.32: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

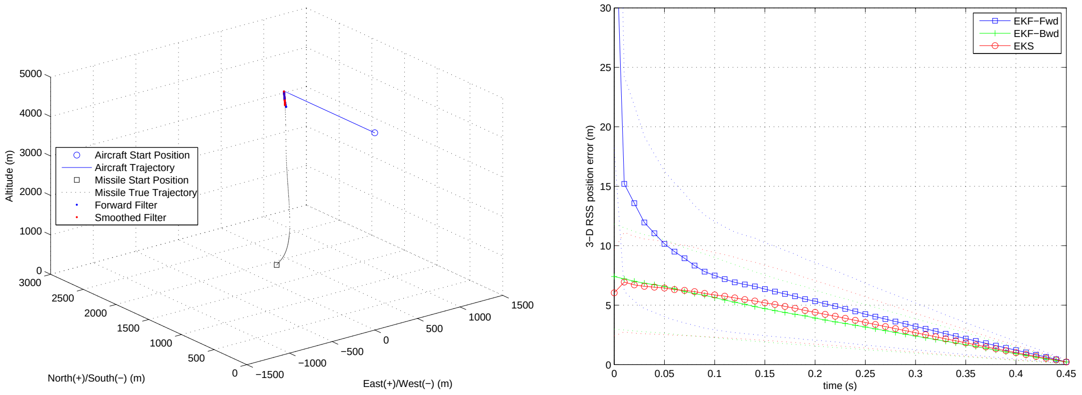
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

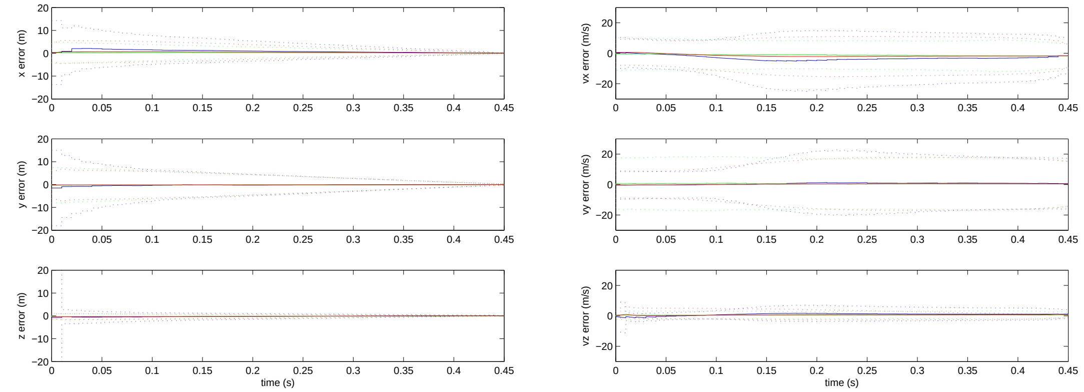
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.33: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

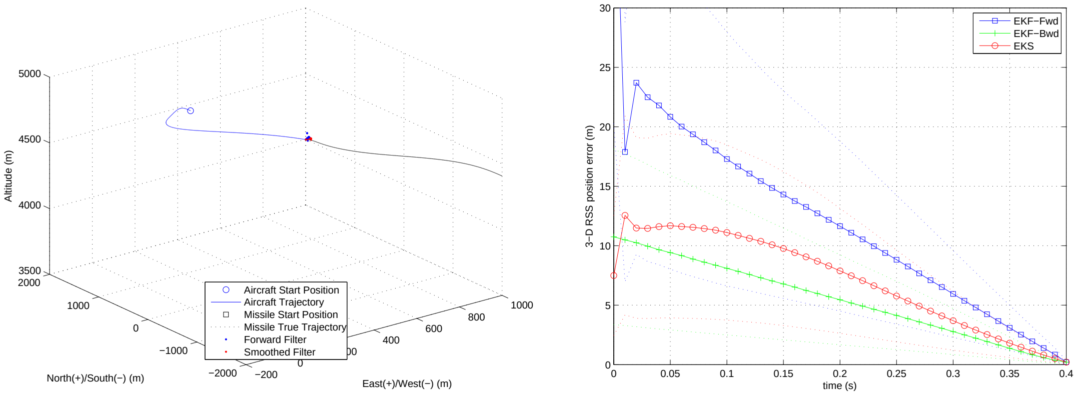
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

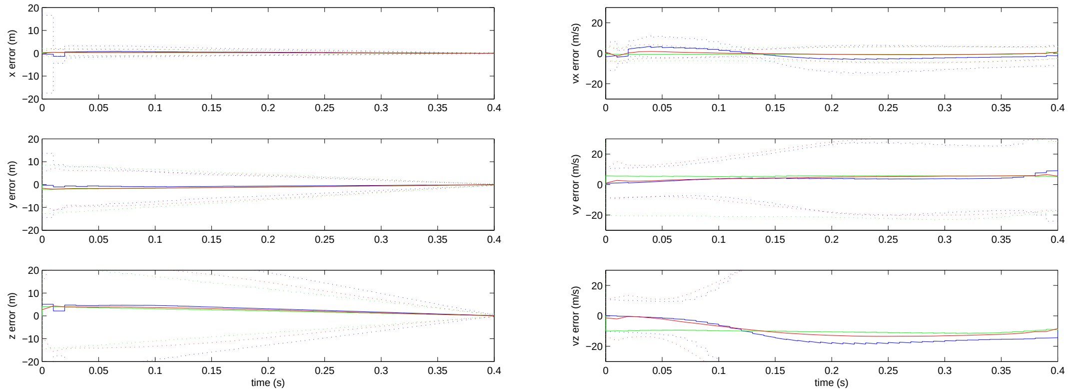
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.34: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

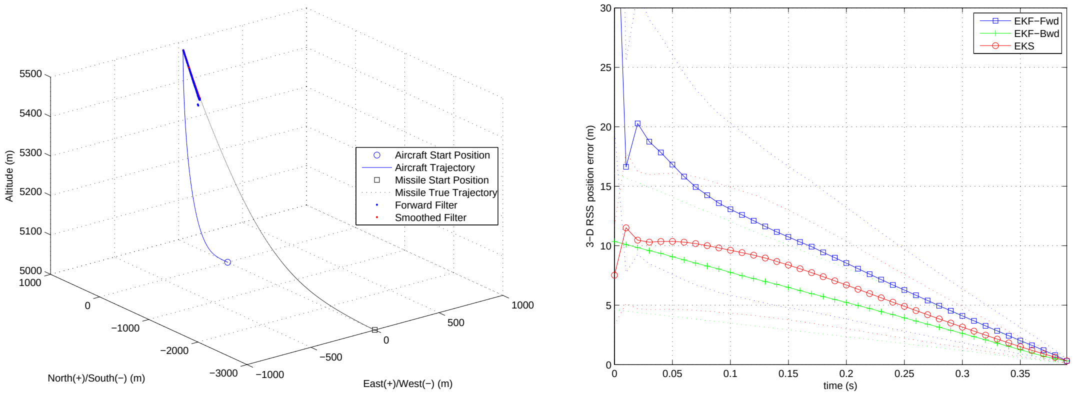
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

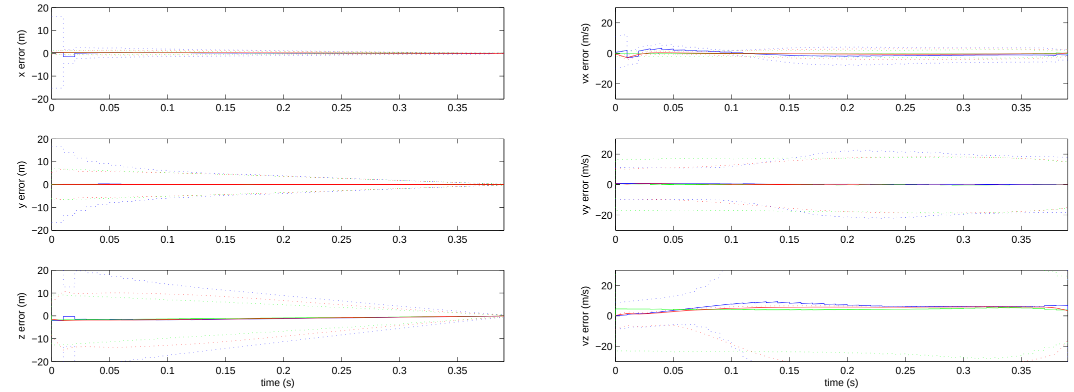
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.35: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

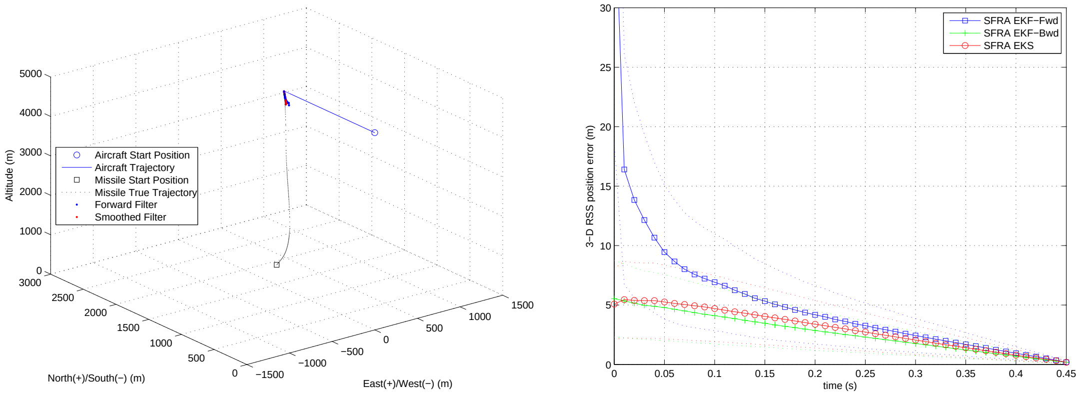


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

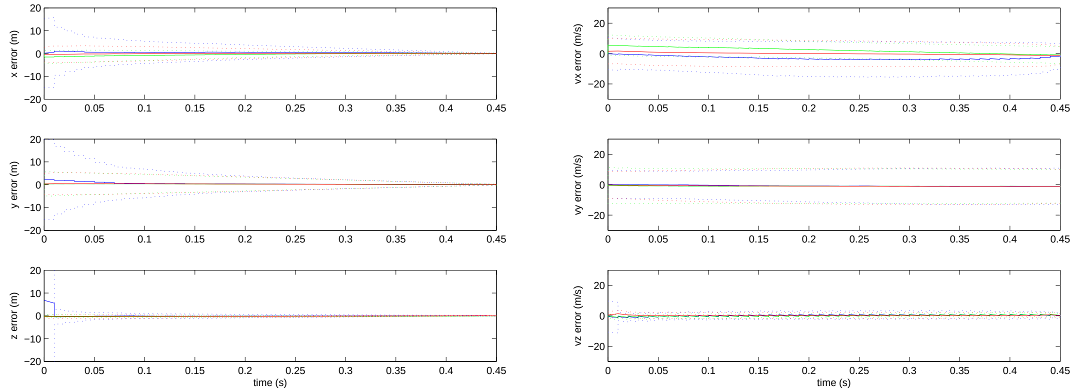
Figure A.36: Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.6 SFRA Extended Kalman Smoother Simulations



(a) 3D Aircraft and Missile Trajectory

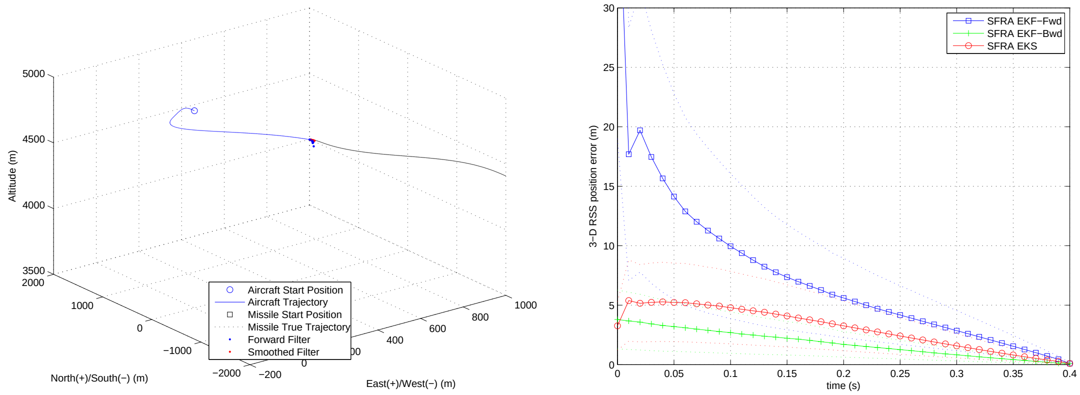
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

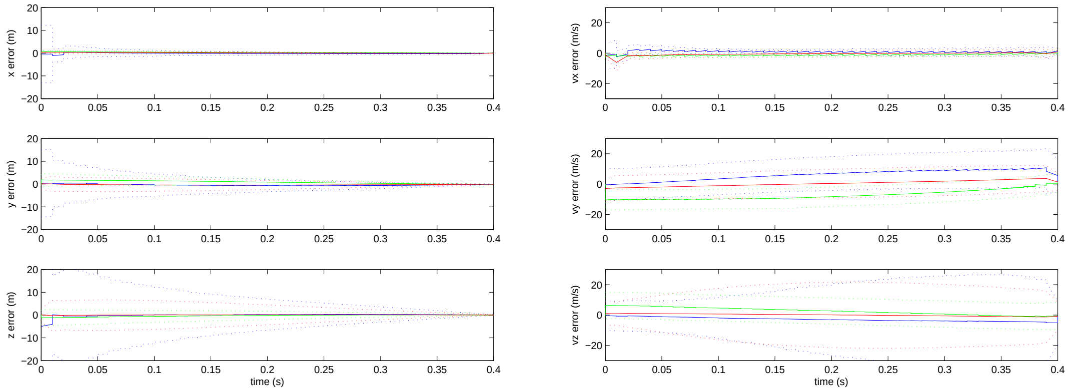
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.37: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

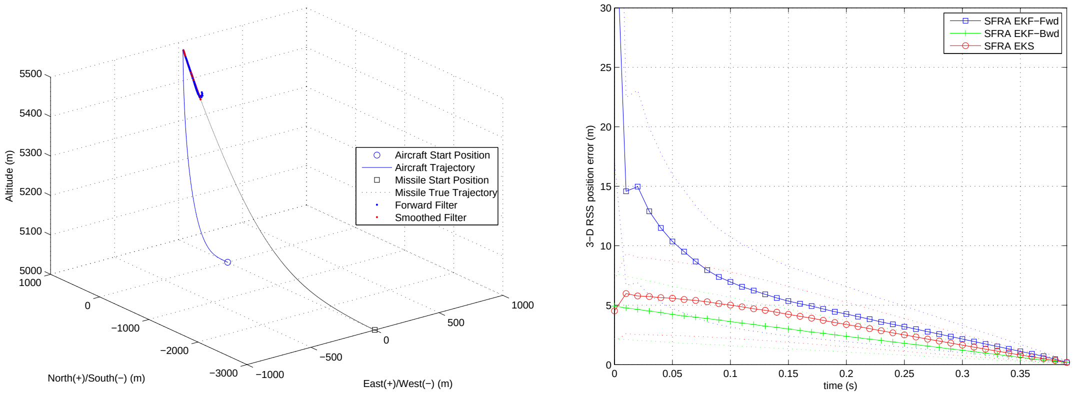
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

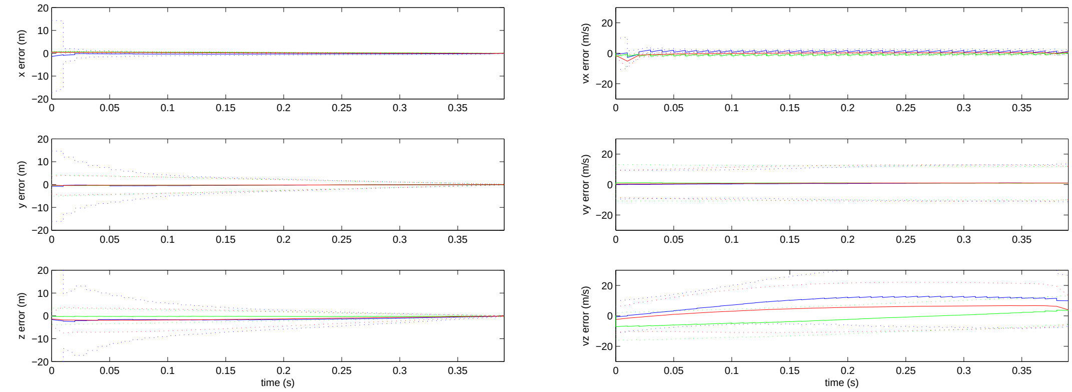
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.38: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

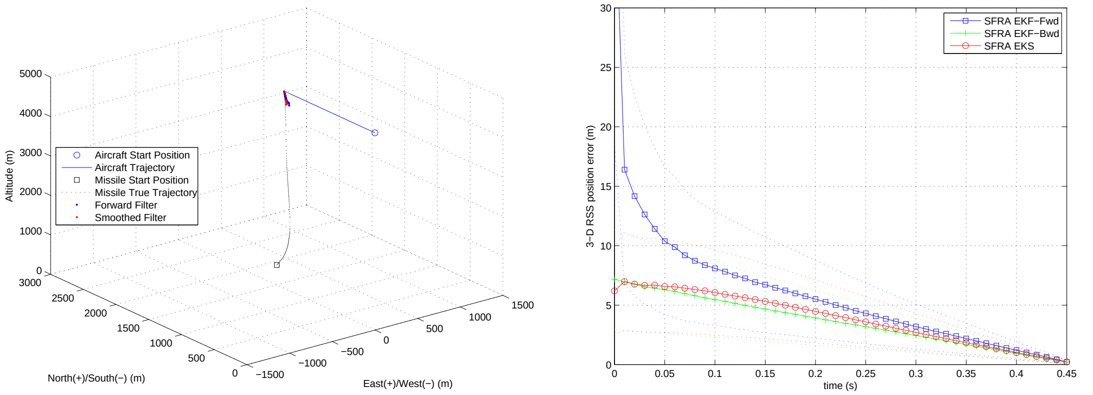
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

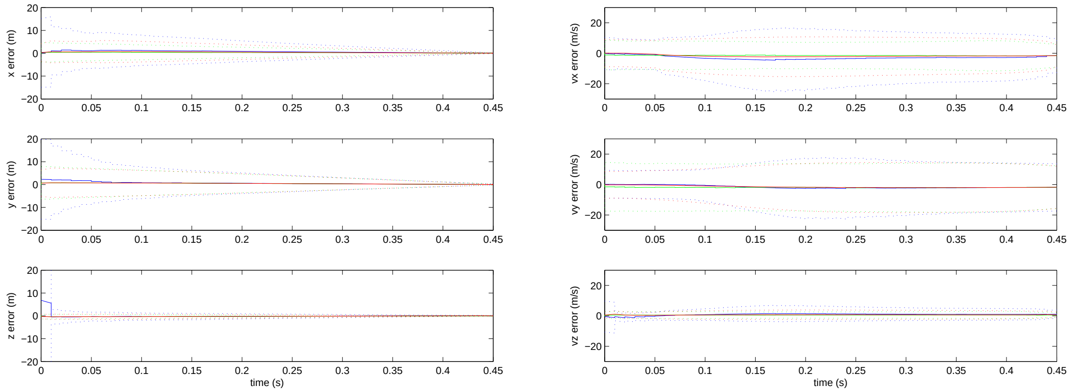
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.39: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

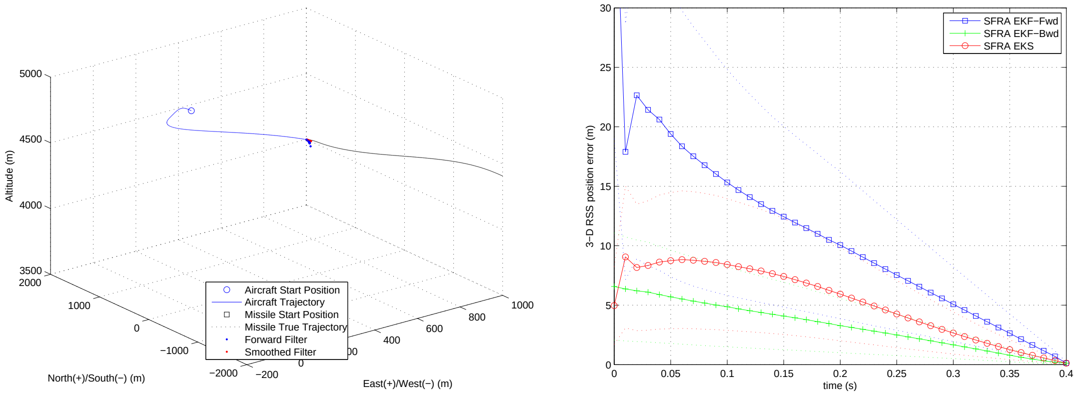
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

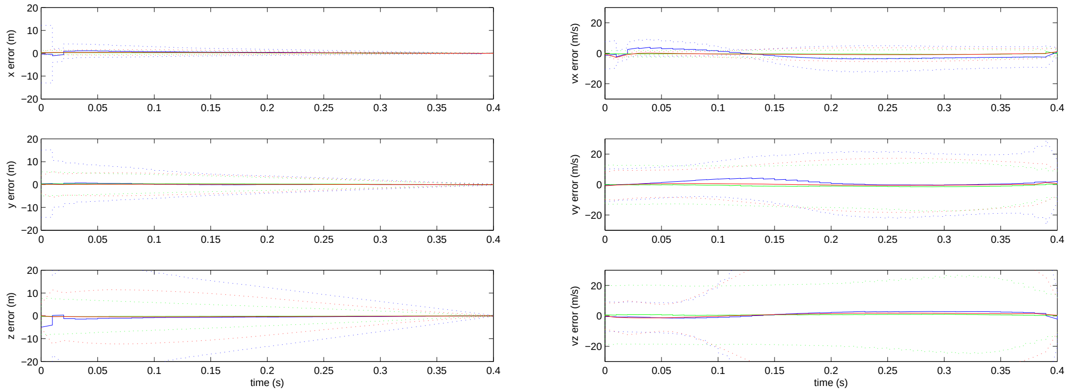
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.40: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

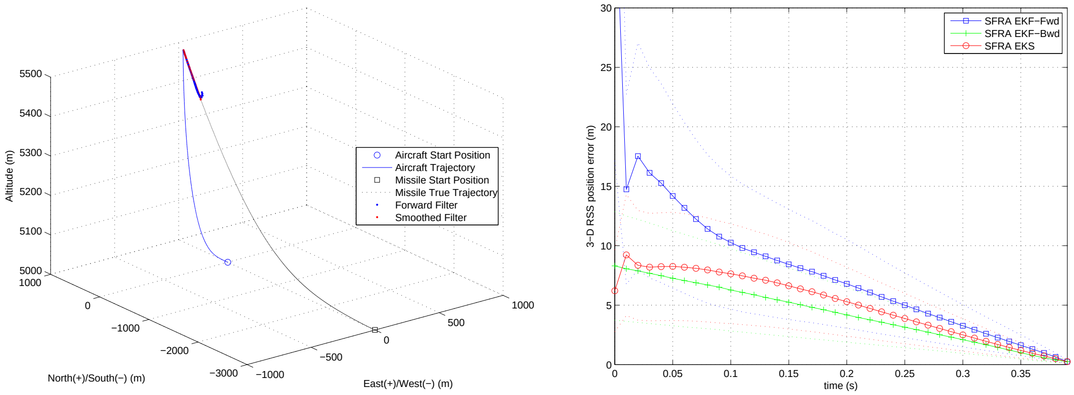
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

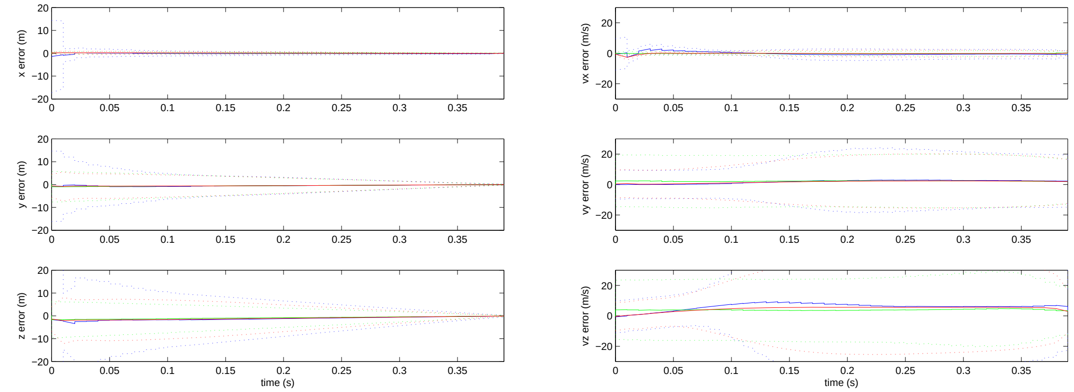
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.41: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

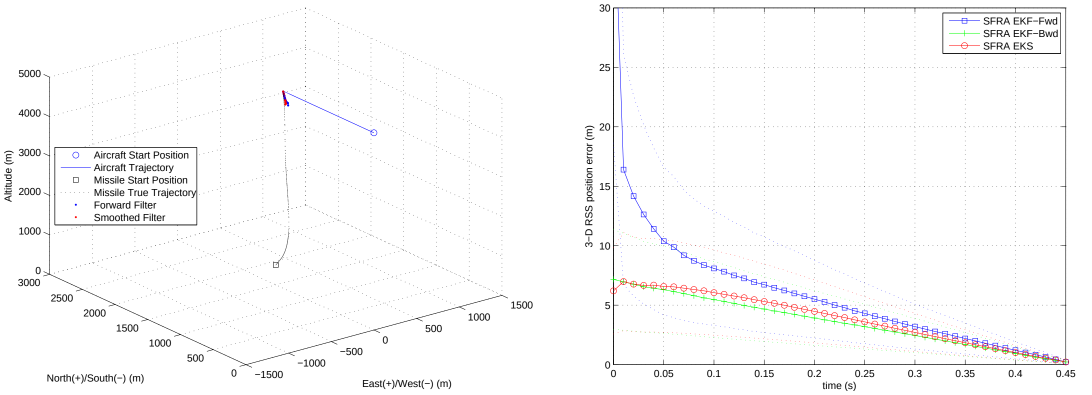
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

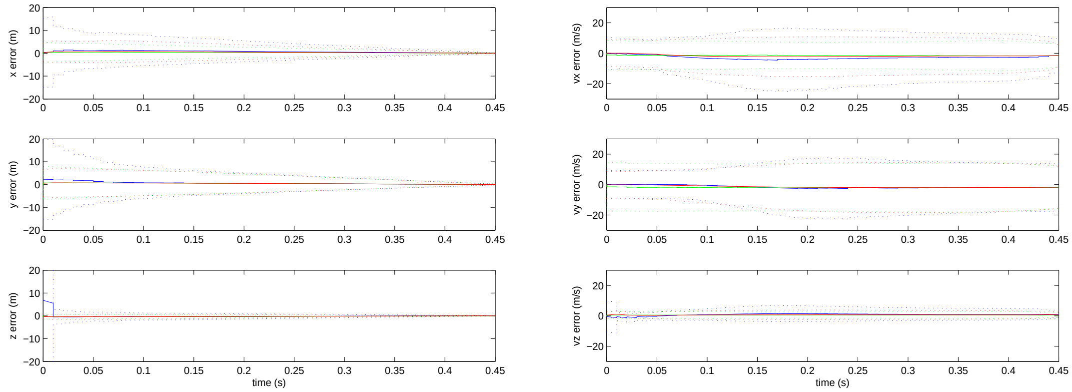
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.42: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

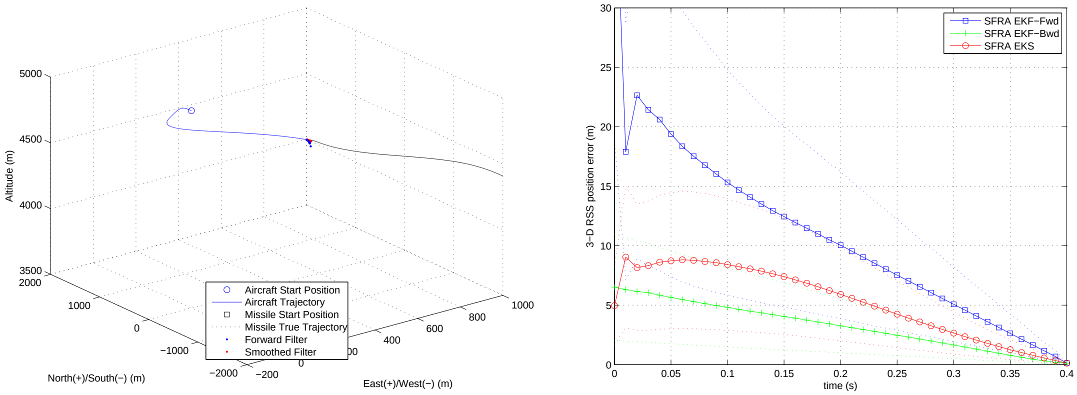
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

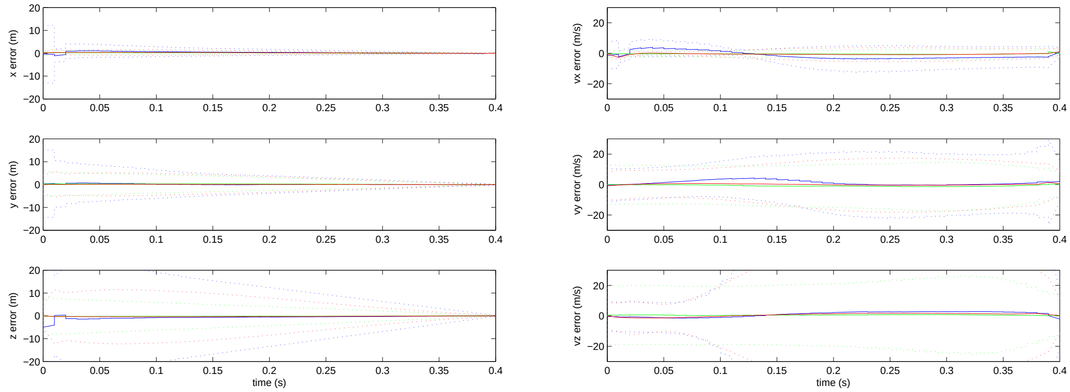
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.43: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

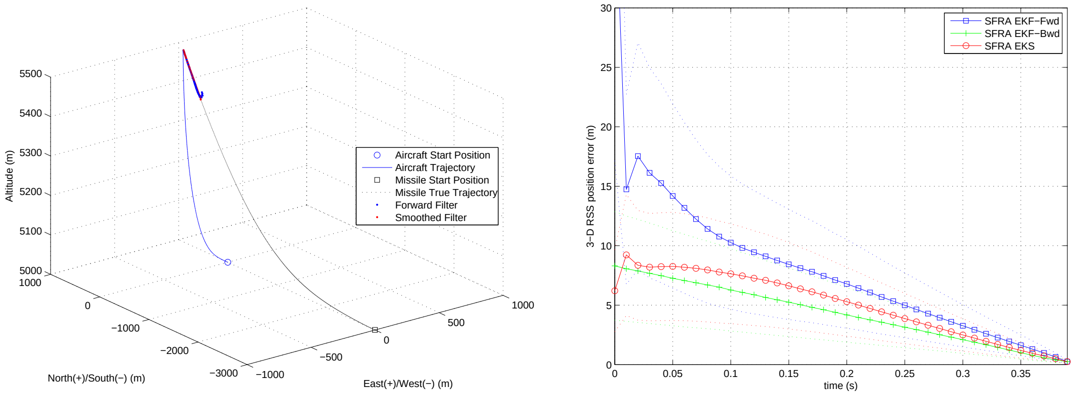
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

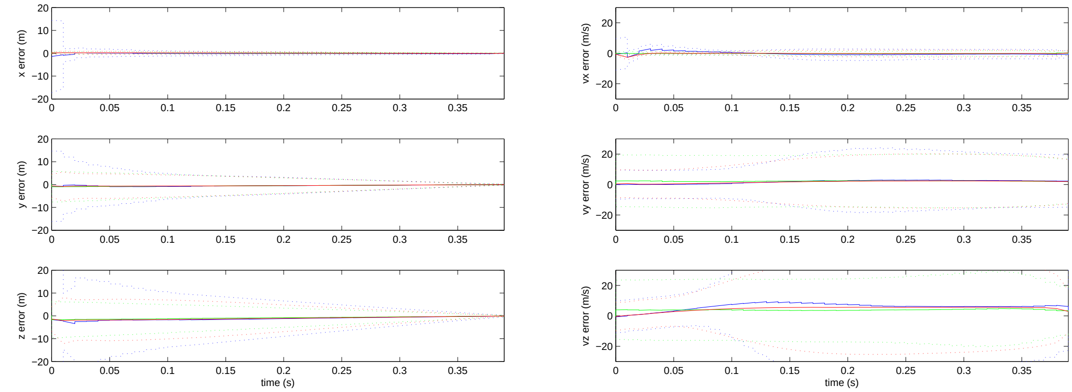
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.44: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

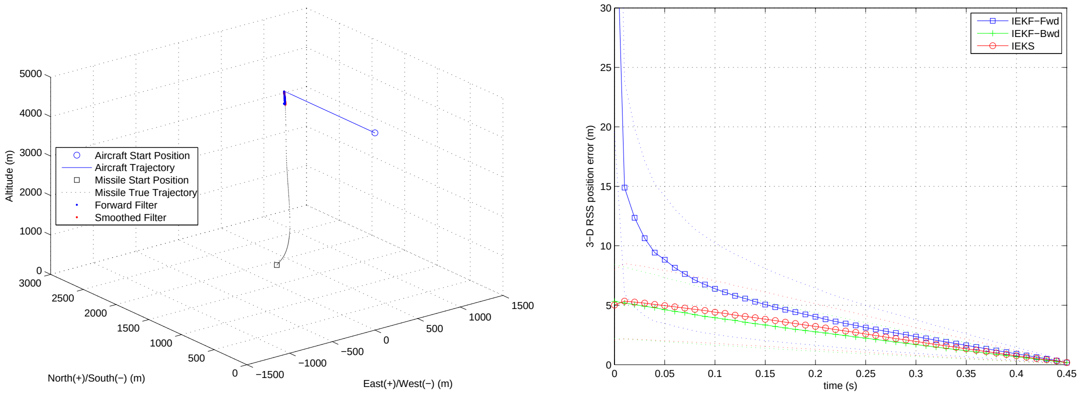


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

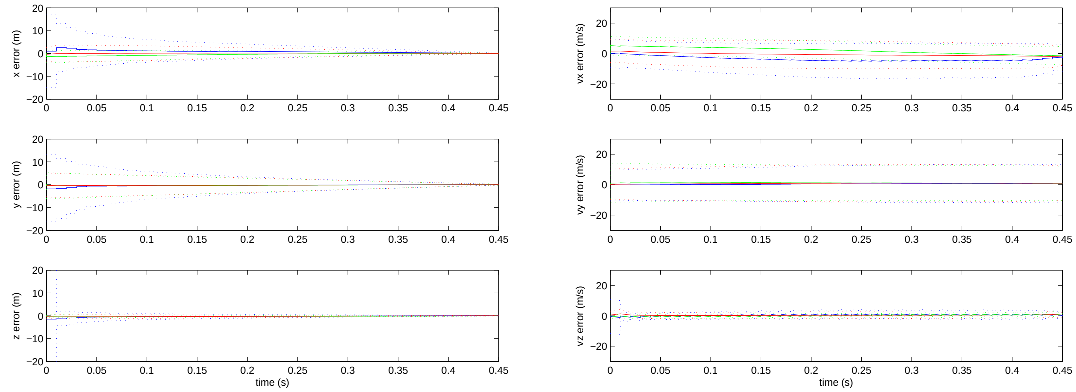
Figure A.45: SFRA Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.7 Iterated Extended Kalman Smoother Simulations



(a) 3D Aircraft and Missile Trajectory

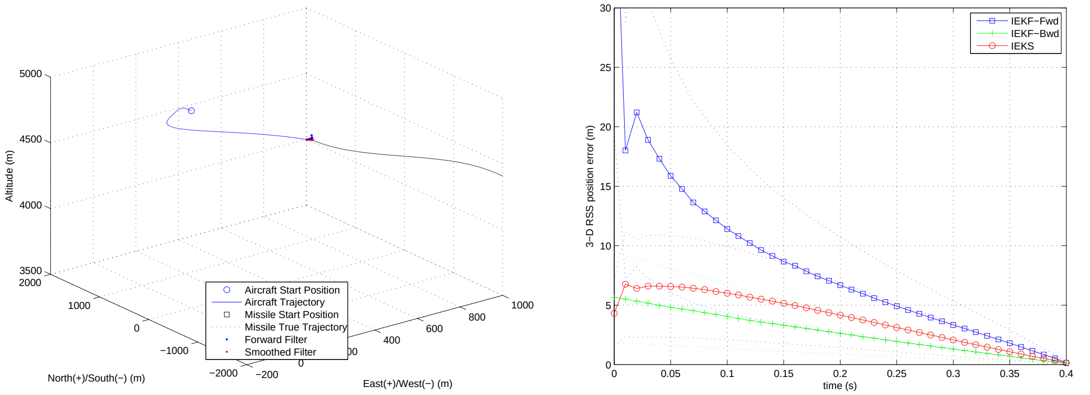
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

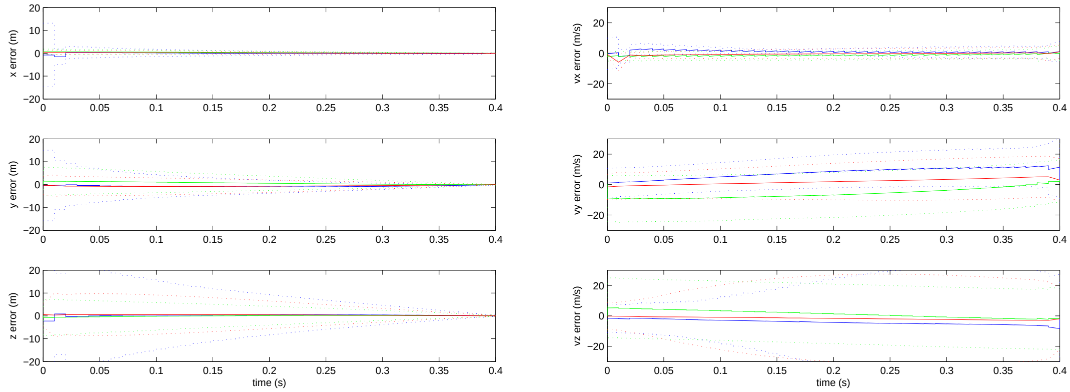
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.46: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

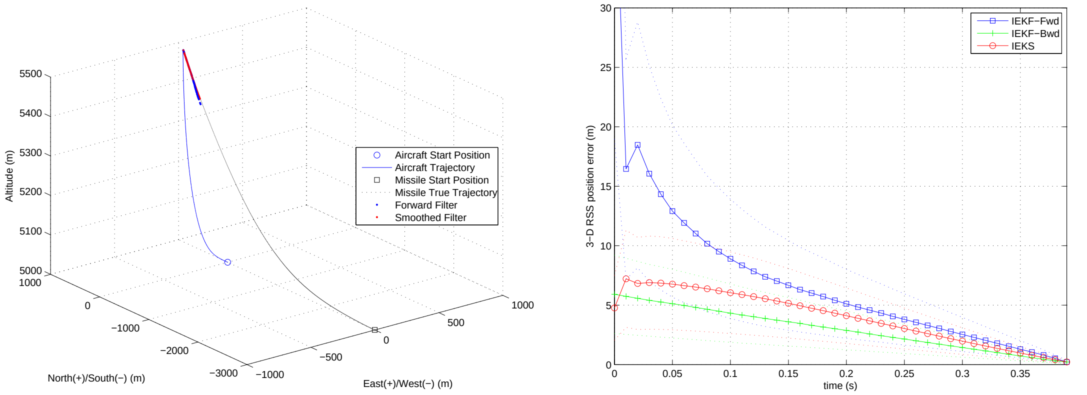
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

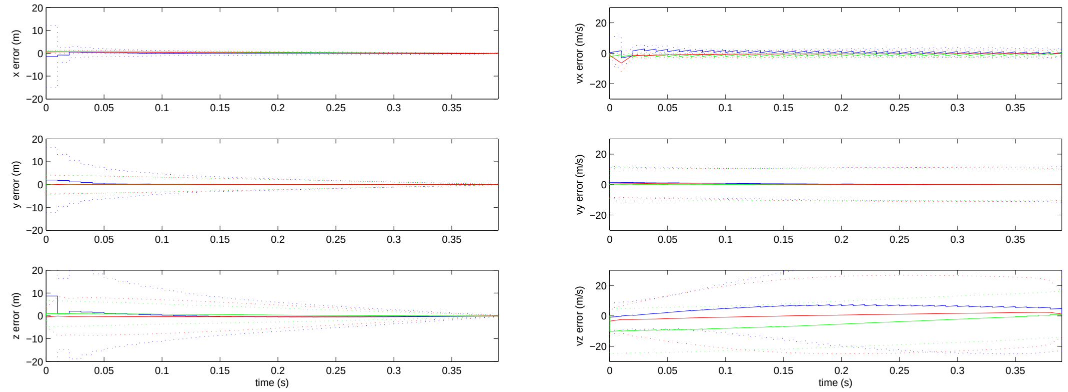
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.47: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

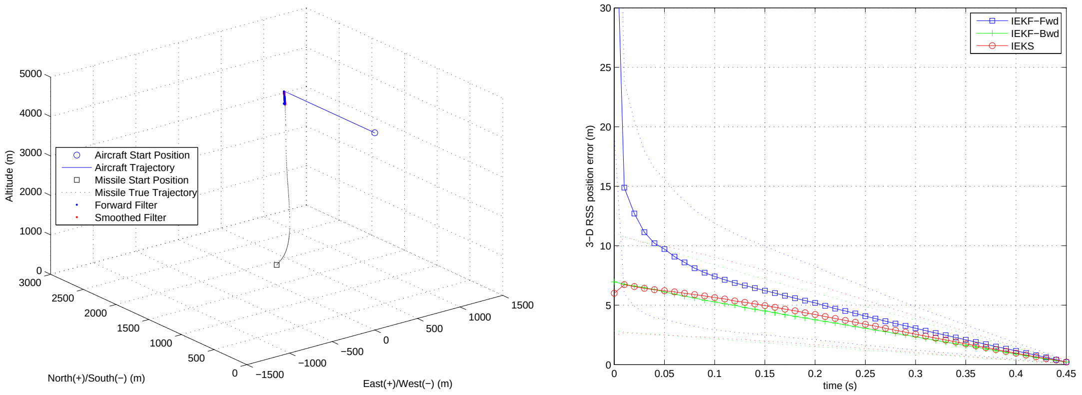
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

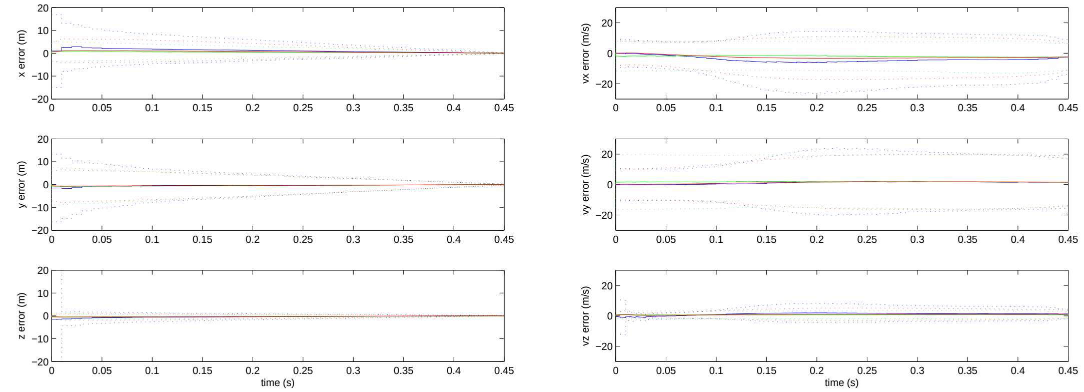
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.48: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

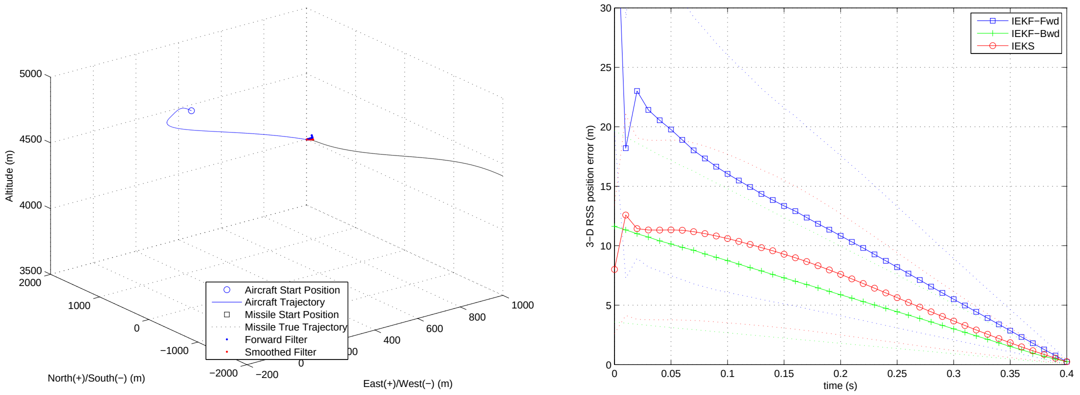
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

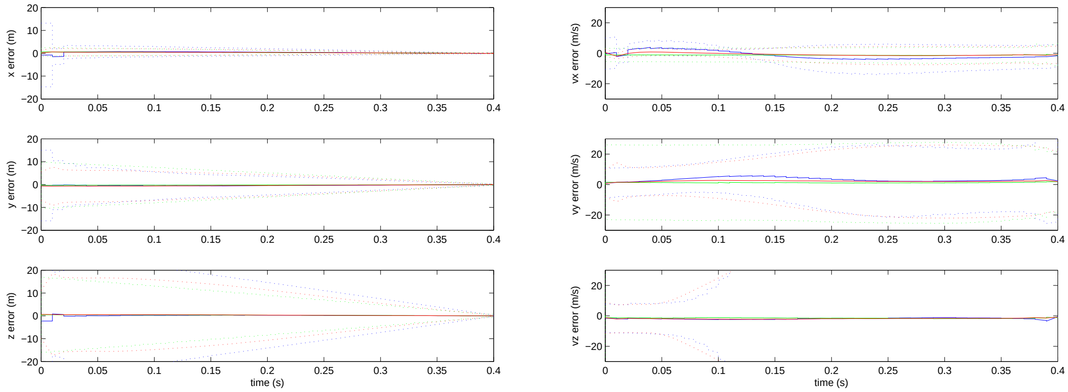
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.49: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

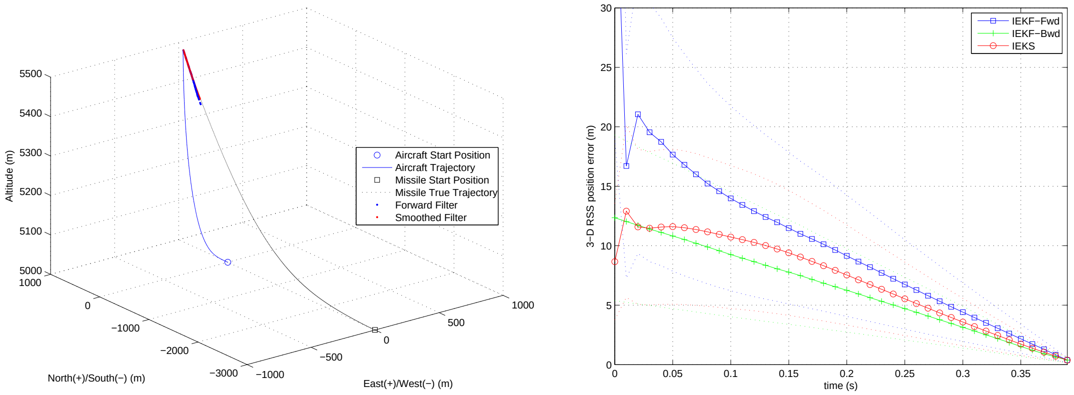
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

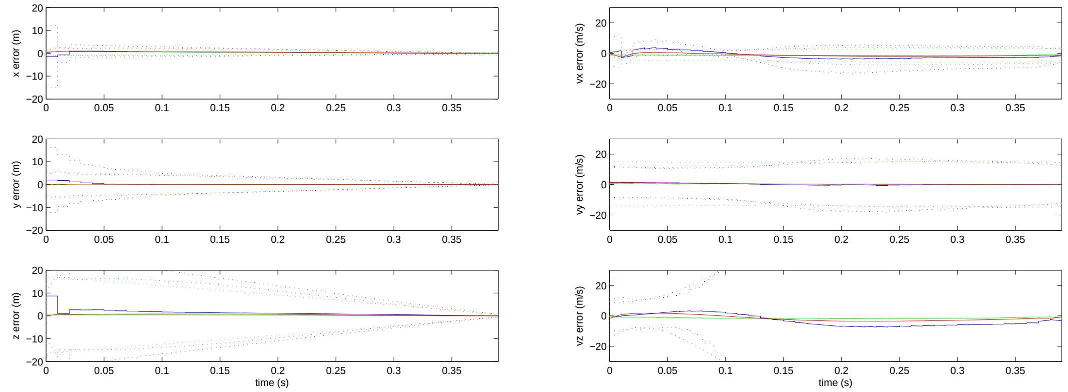
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.50: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

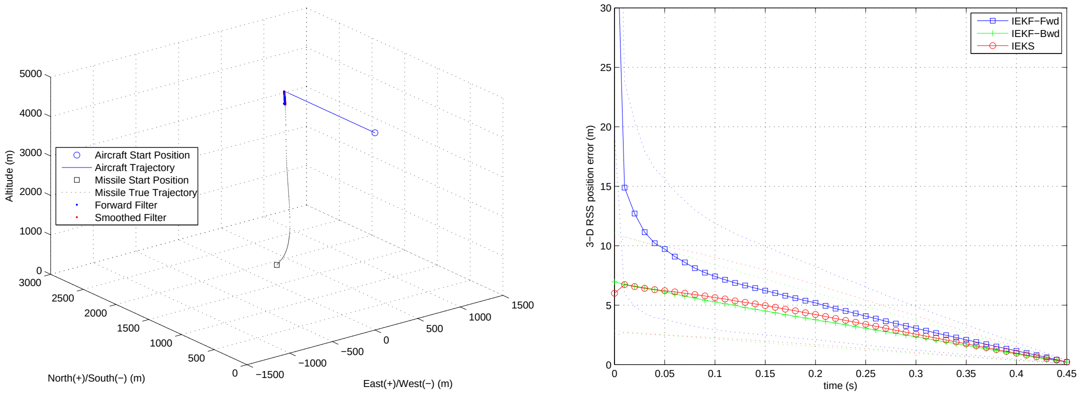
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

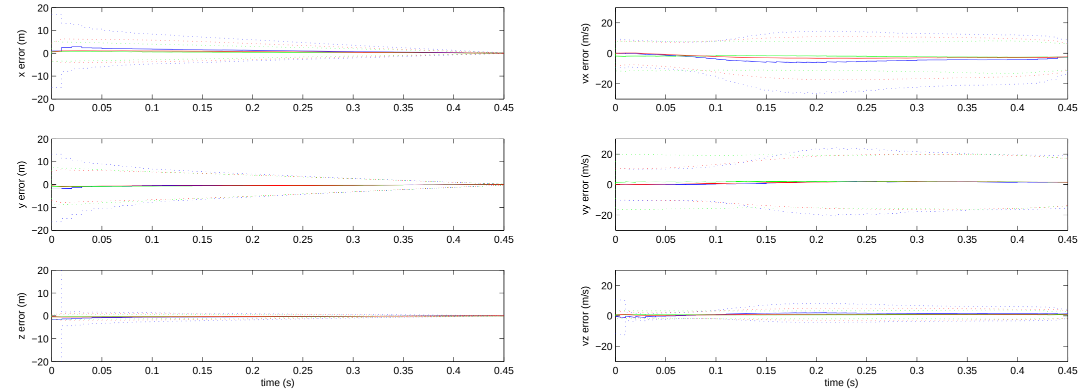
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.51: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

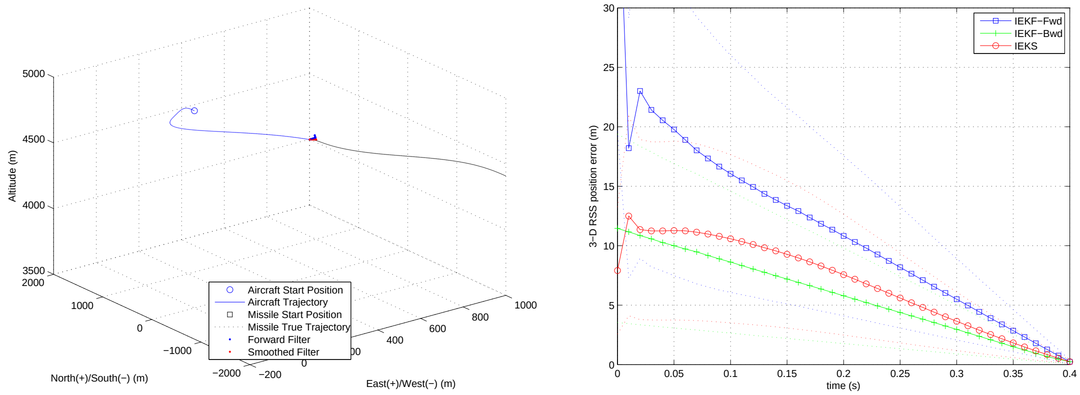
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

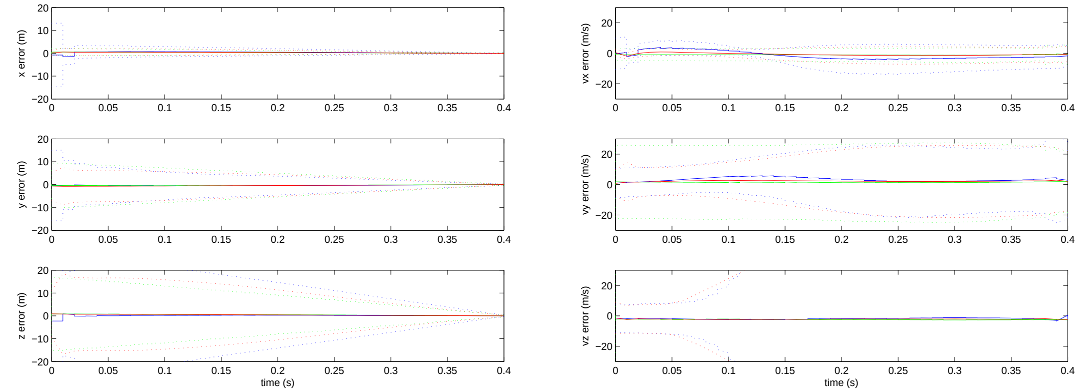
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.52: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

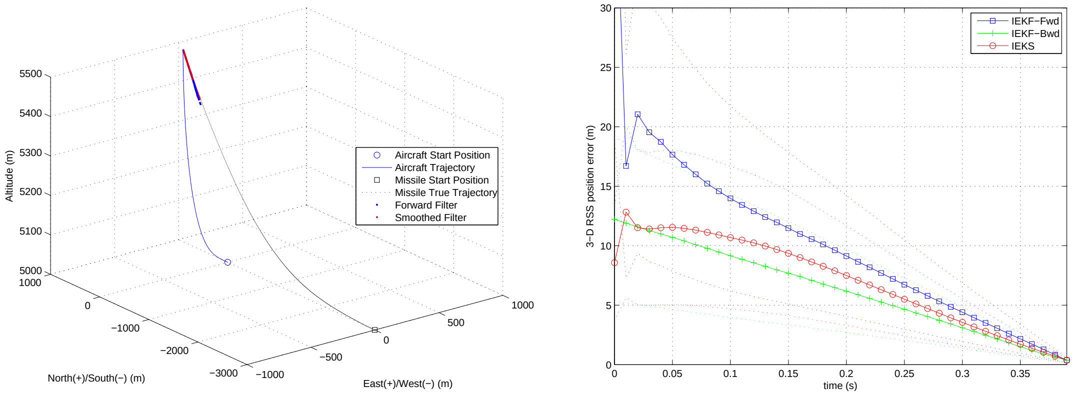
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

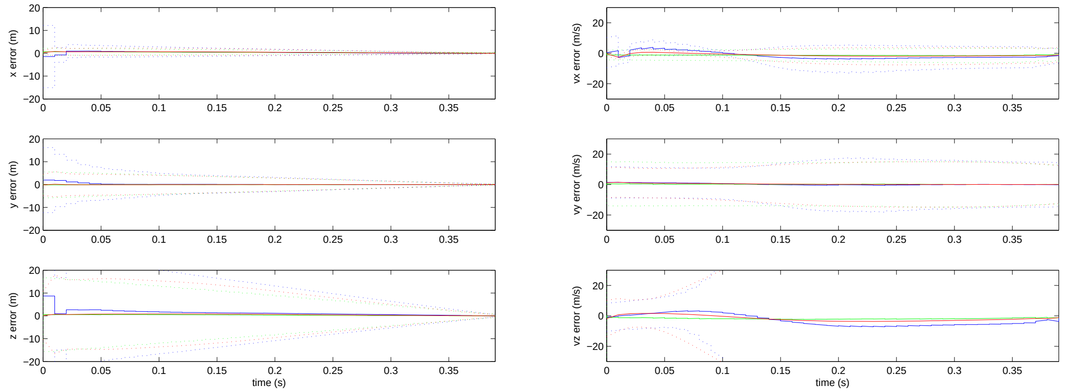
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.53: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

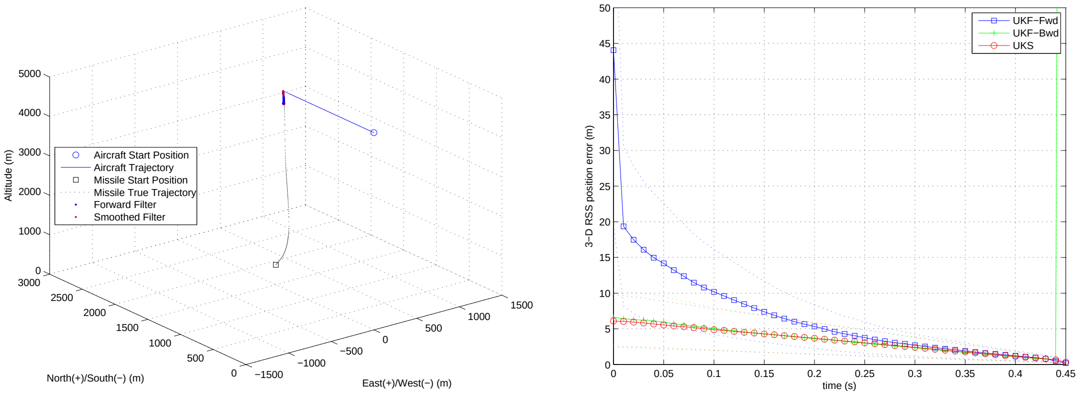


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

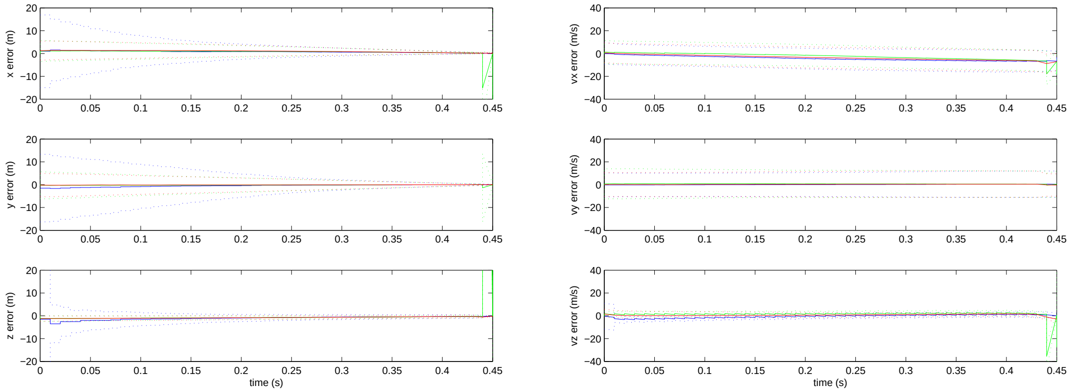
Figure A.54: Iterated Extended Kalman Smoother Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.8 Unscented Kalman Smoother Simulations with 10% *Sensor Noise and Tuning*



(a) 3D Aircraft and Missile Trajectory

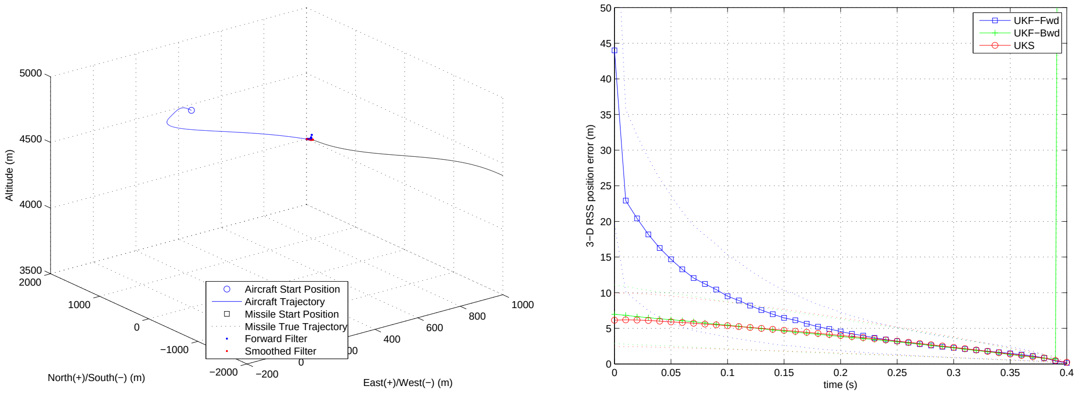
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

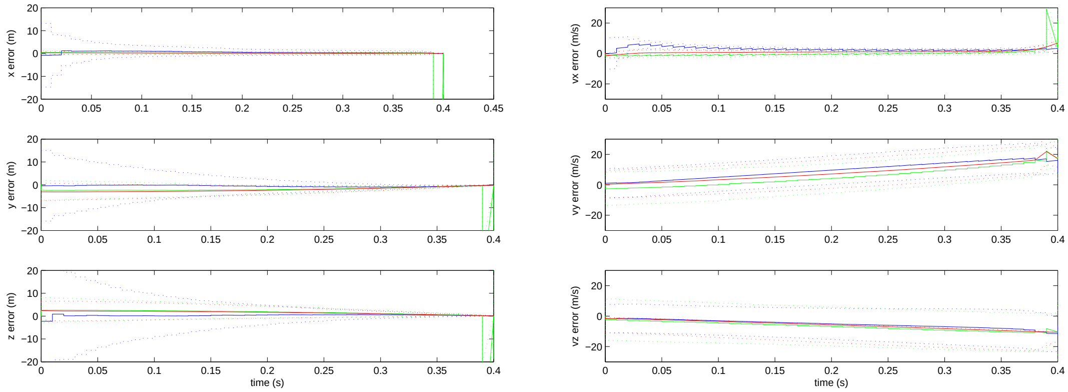
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.55: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

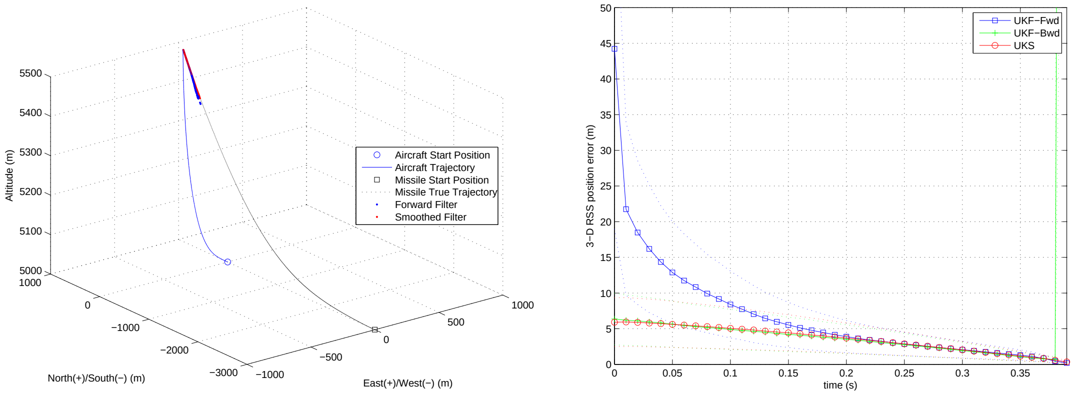
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

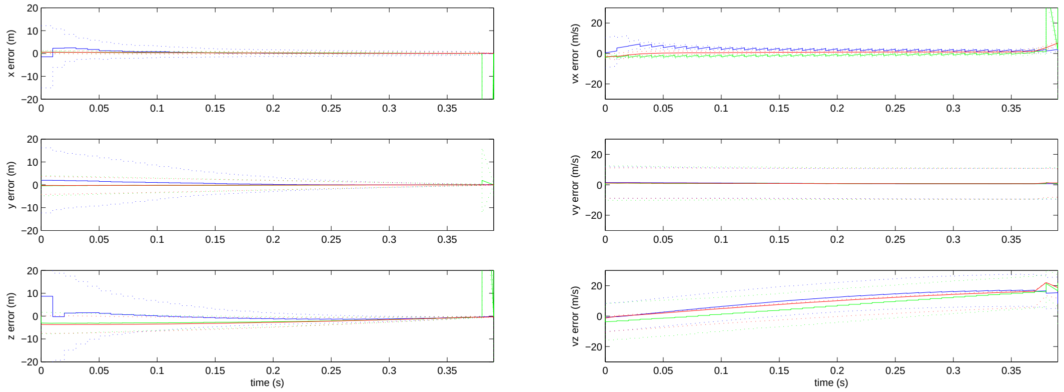
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.56: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

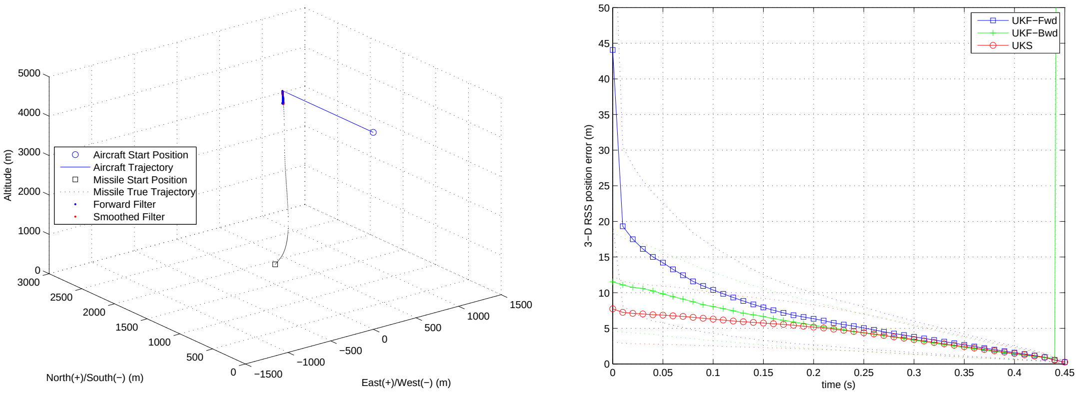
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

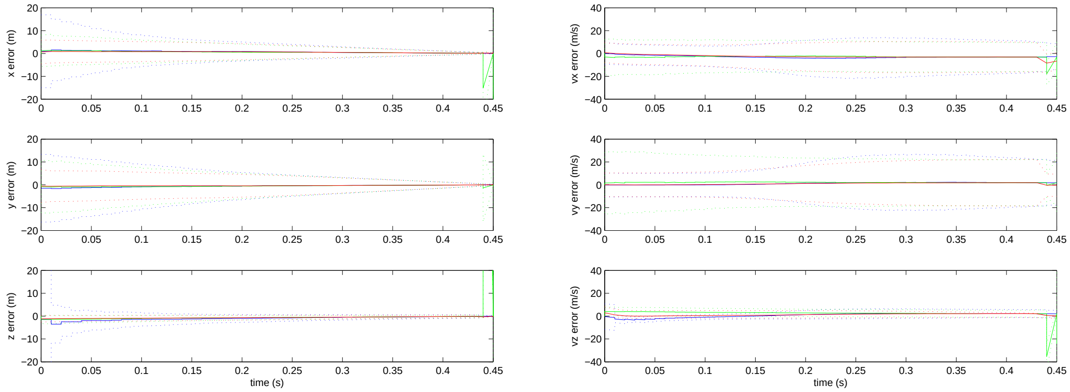
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.57: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

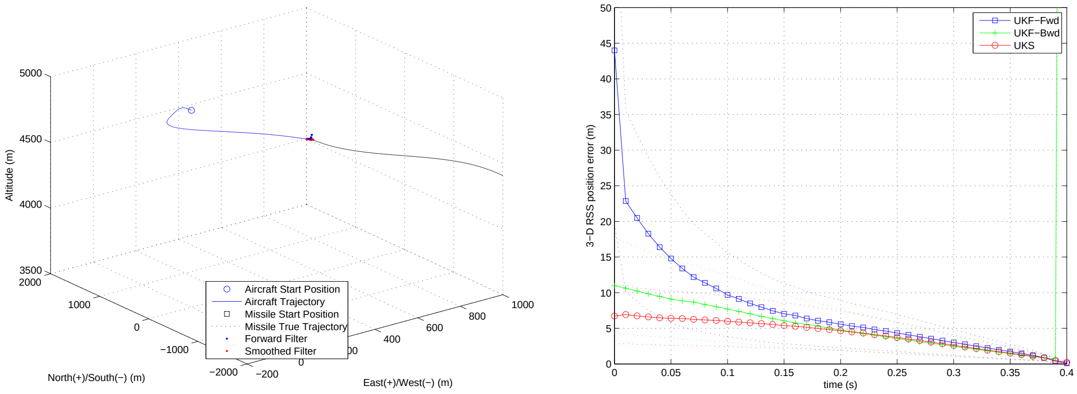
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

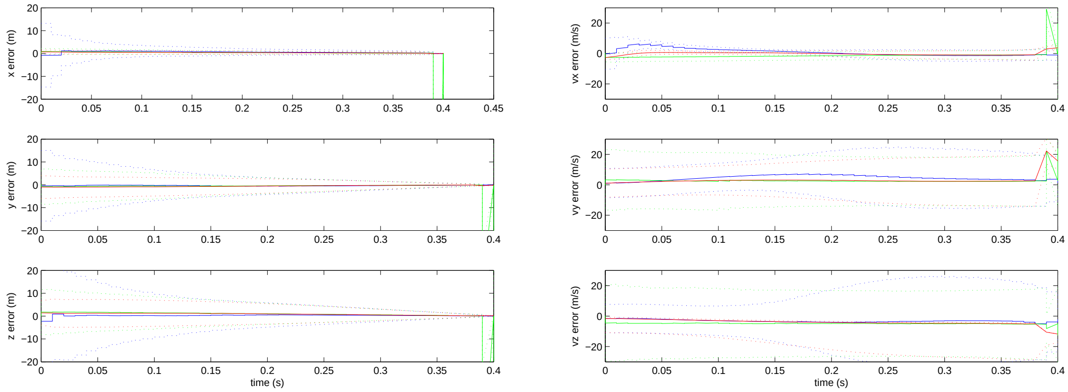
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.58: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

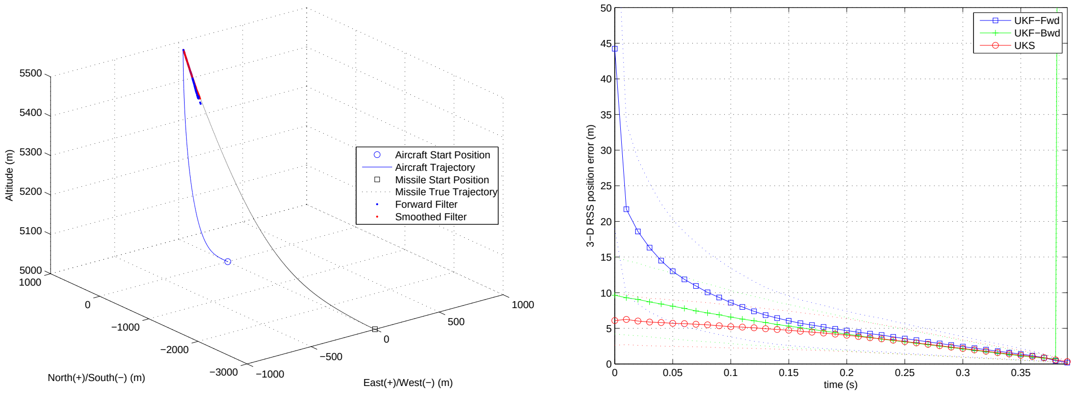
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

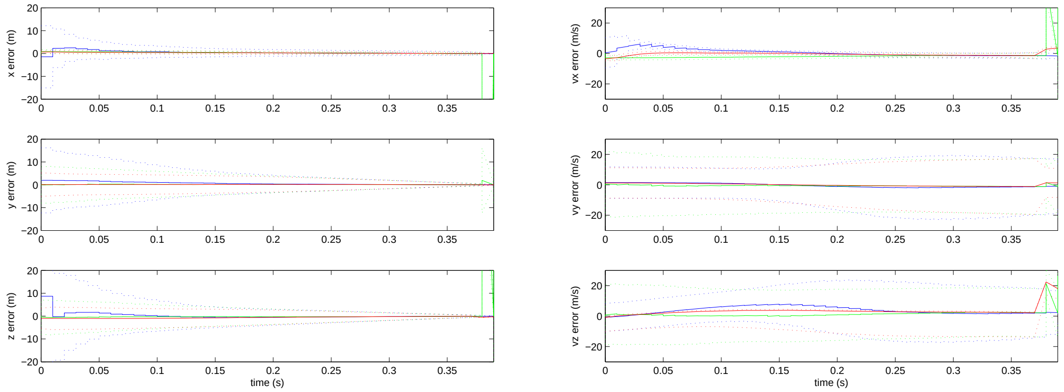
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.59: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

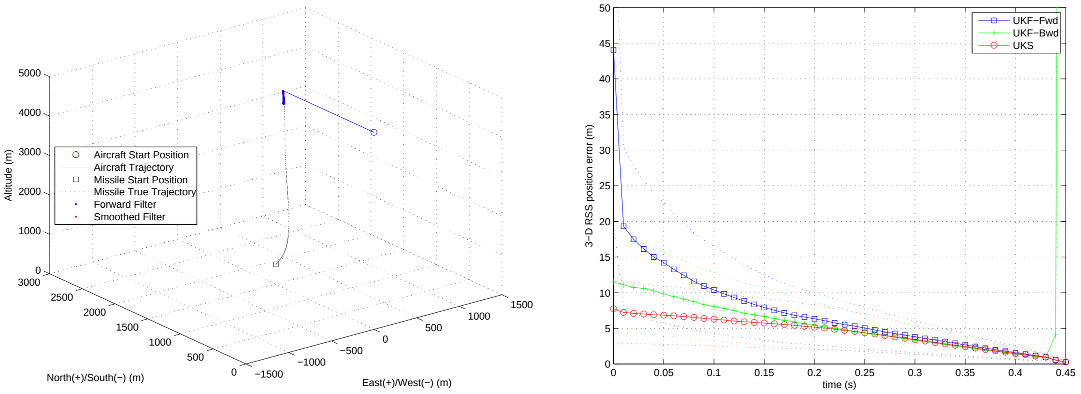
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

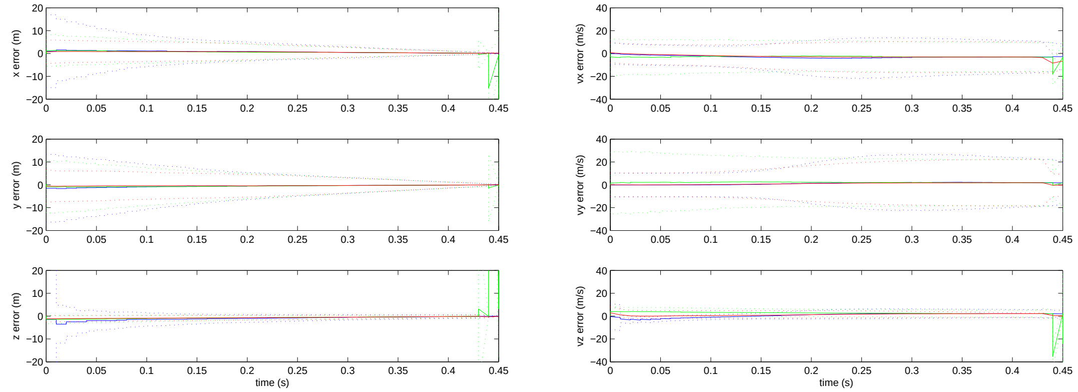
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.60: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

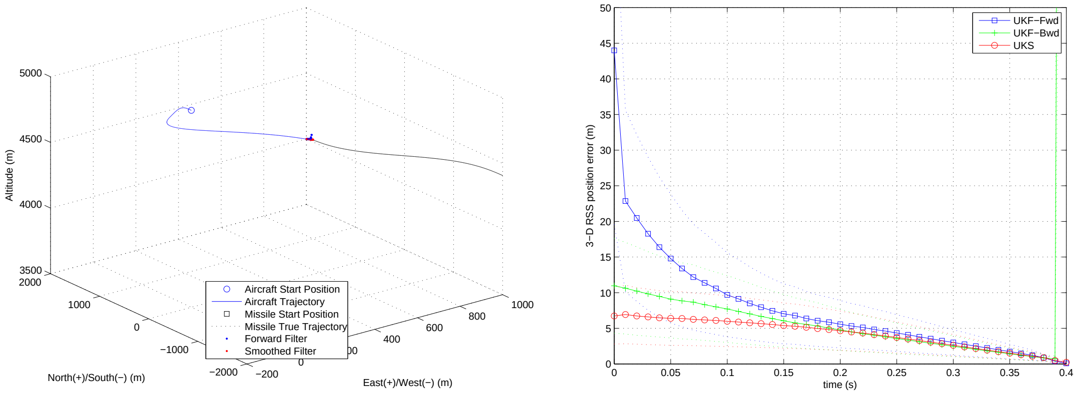
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

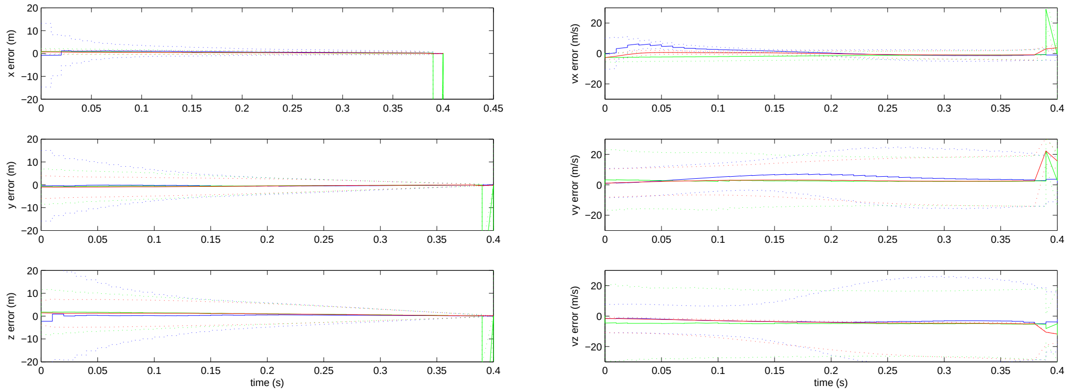
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.61: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

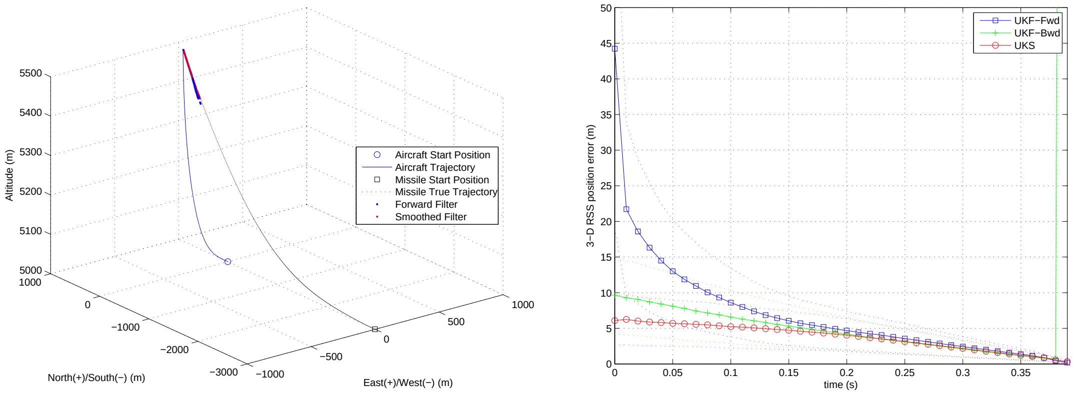
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

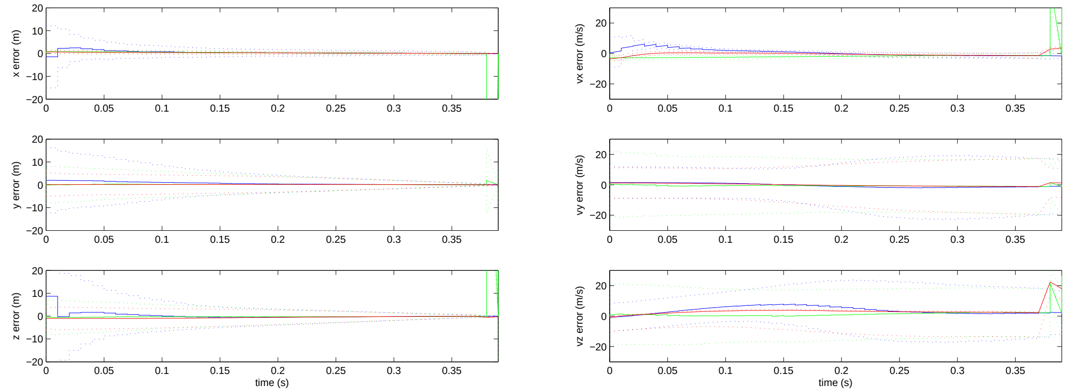
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.62: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

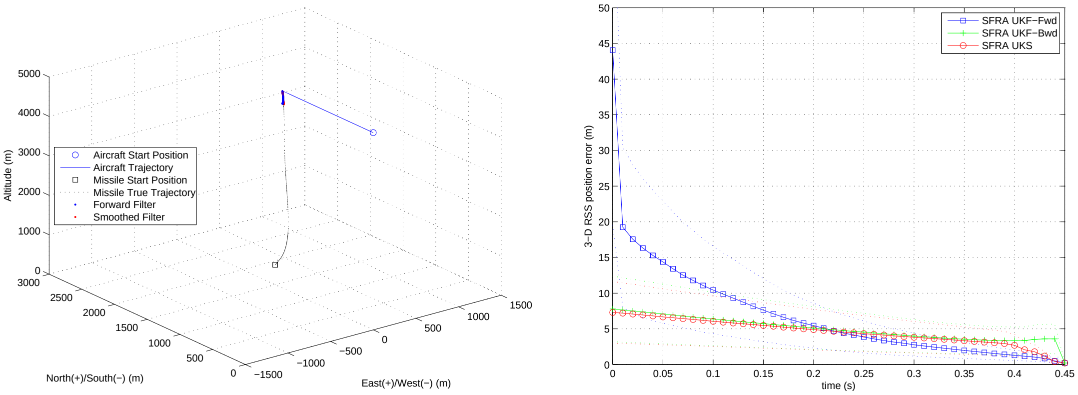


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

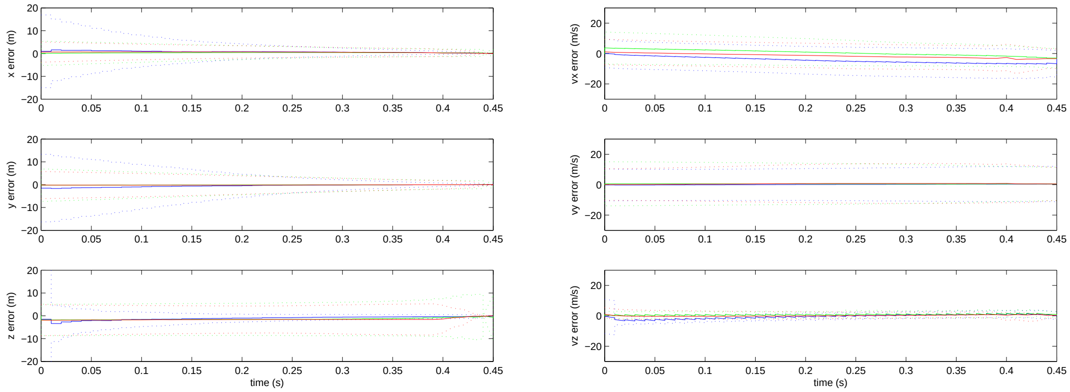
Figure A.63: Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.9 SFRA Unscented Kalman Smoother Simulations with 10% *Sensor Noise and Tuning*



(a) 3D Aircraft and Missile Trajectory

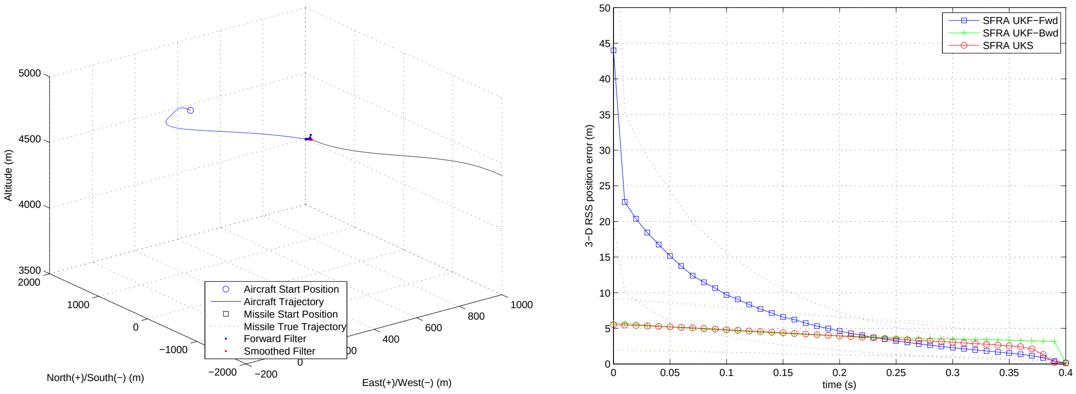
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

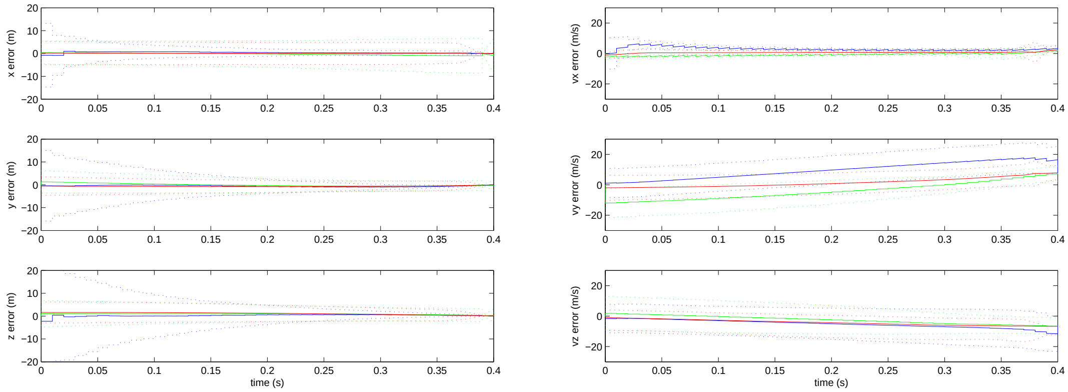
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.64: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

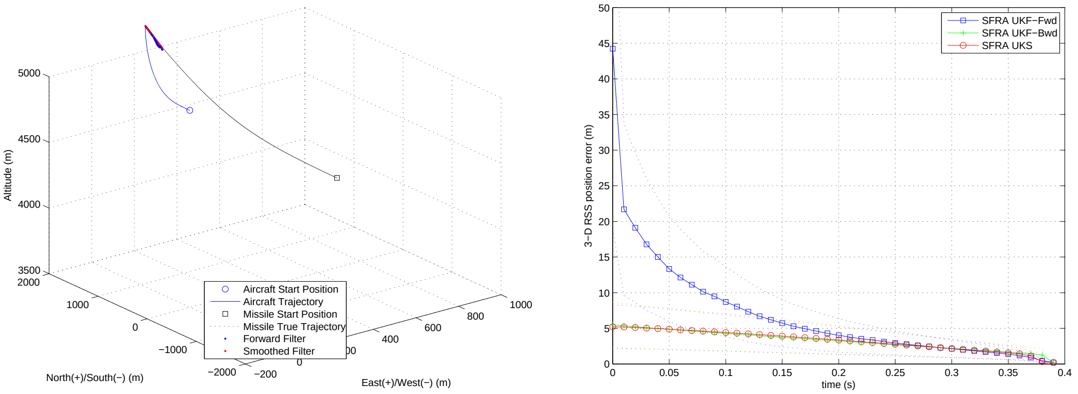
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

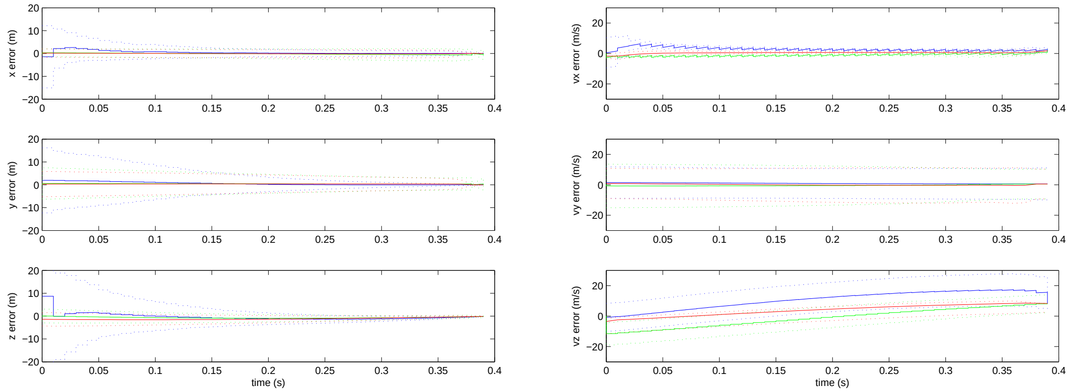
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.65: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

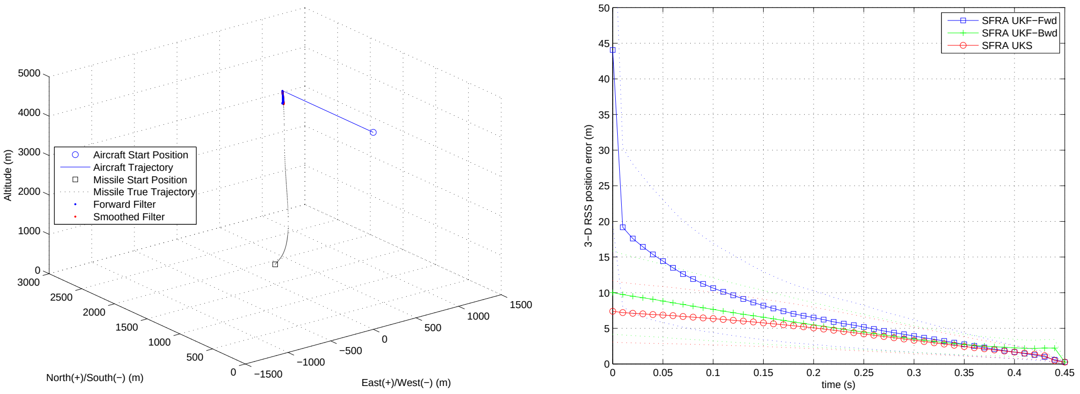
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

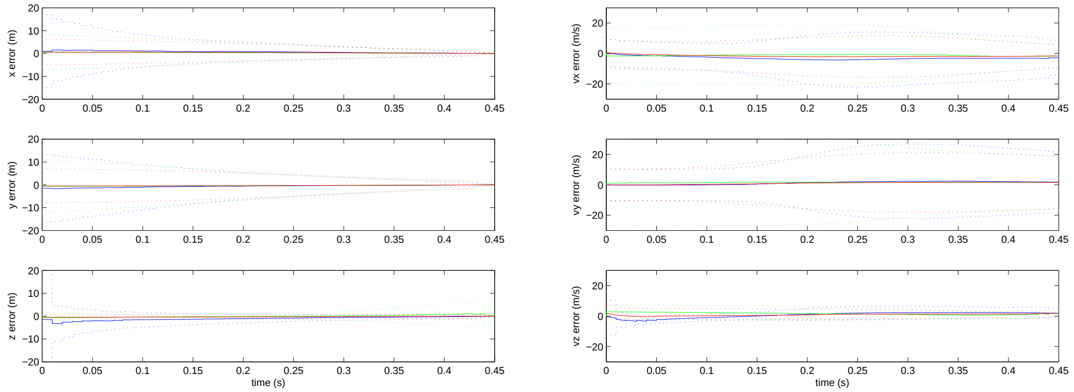
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.66: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

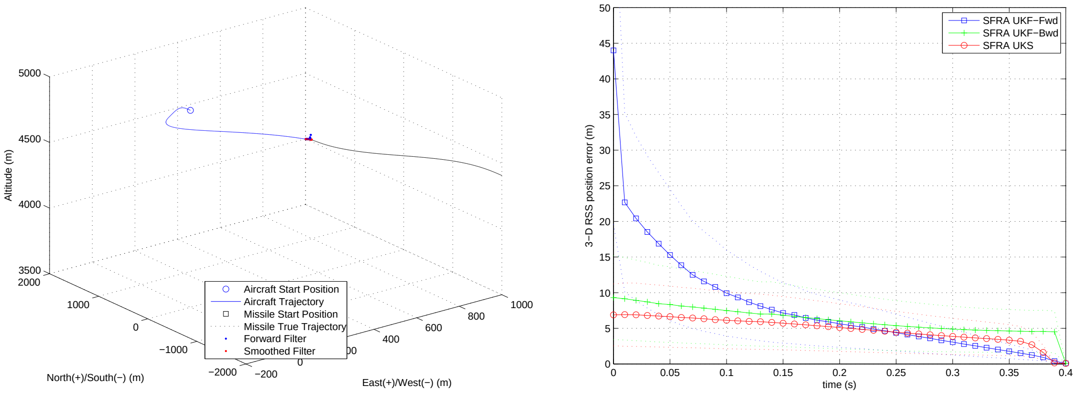
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

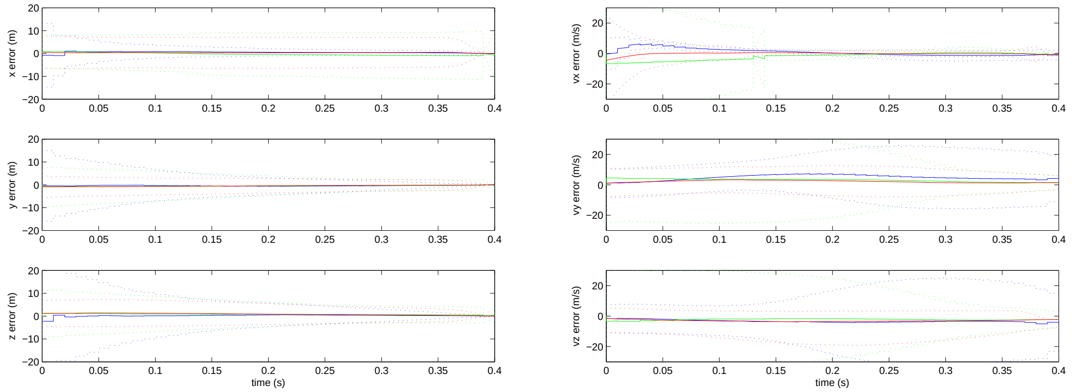
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.67: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

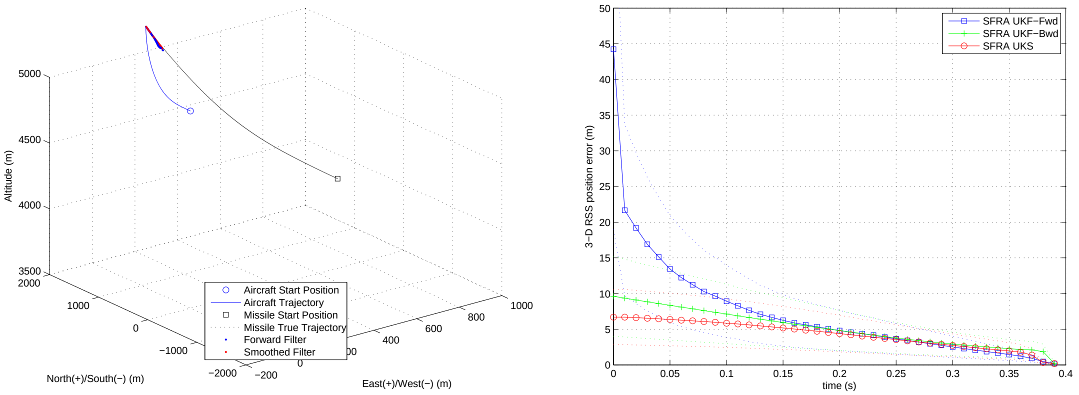
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

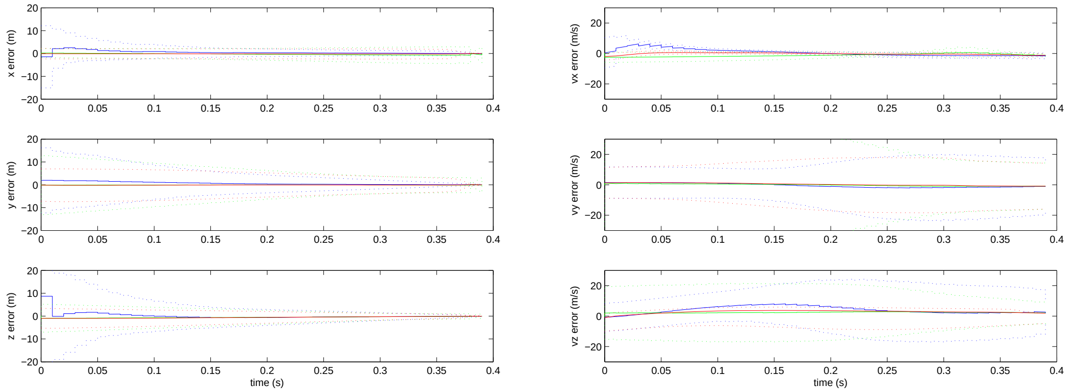
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.68: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

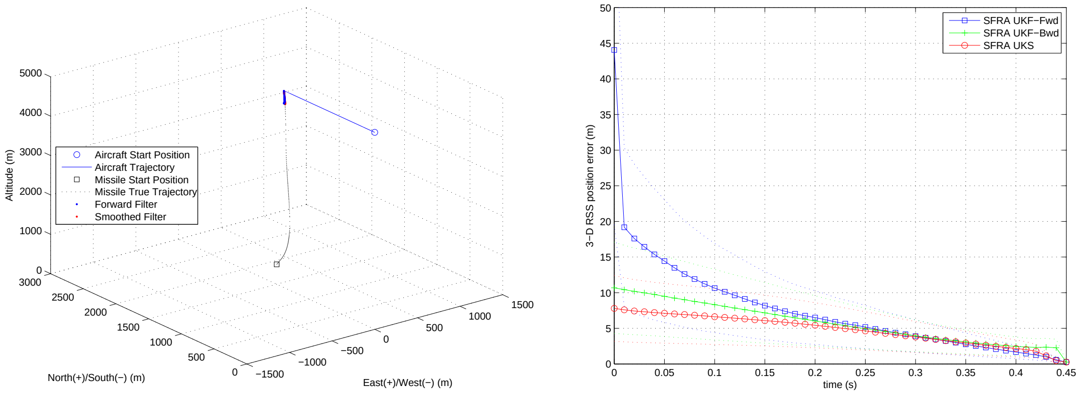
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

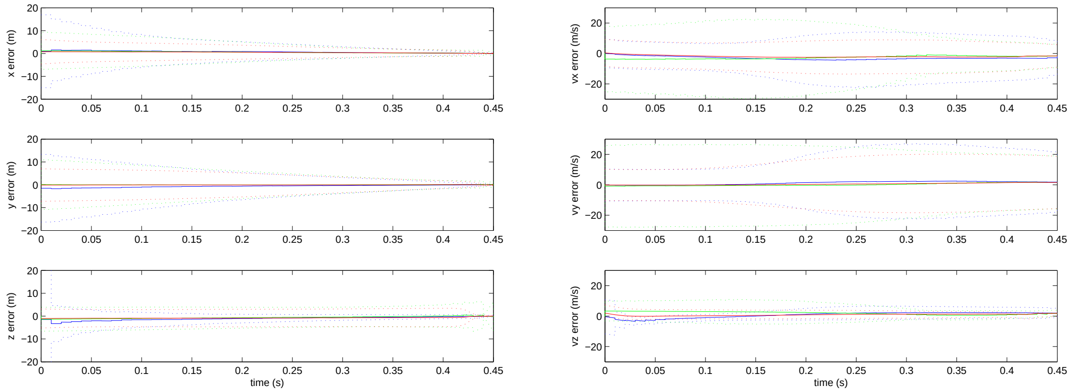
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.69: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

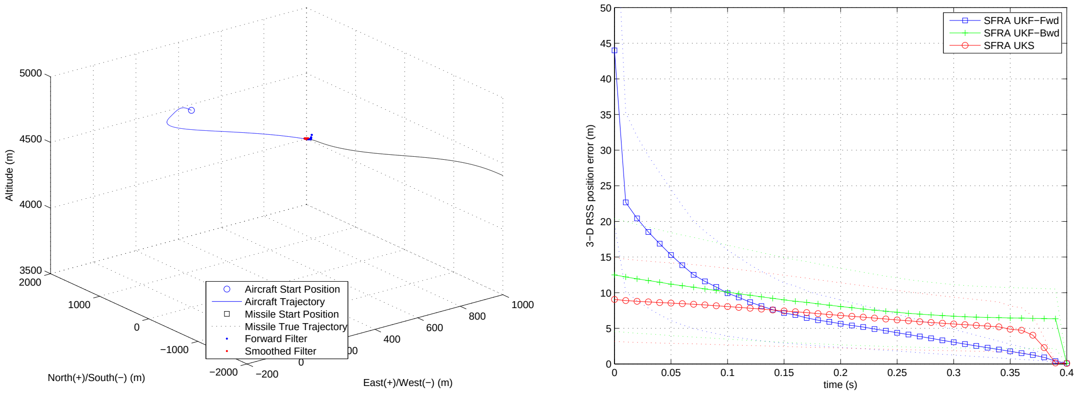
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

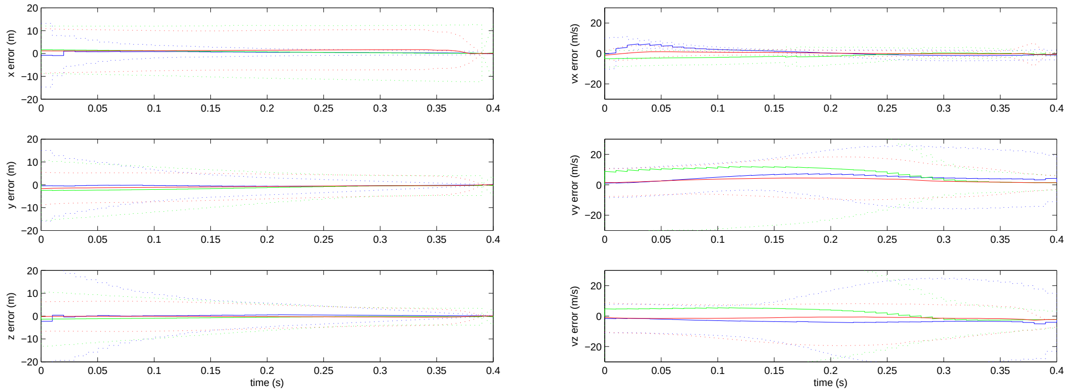
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.70: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

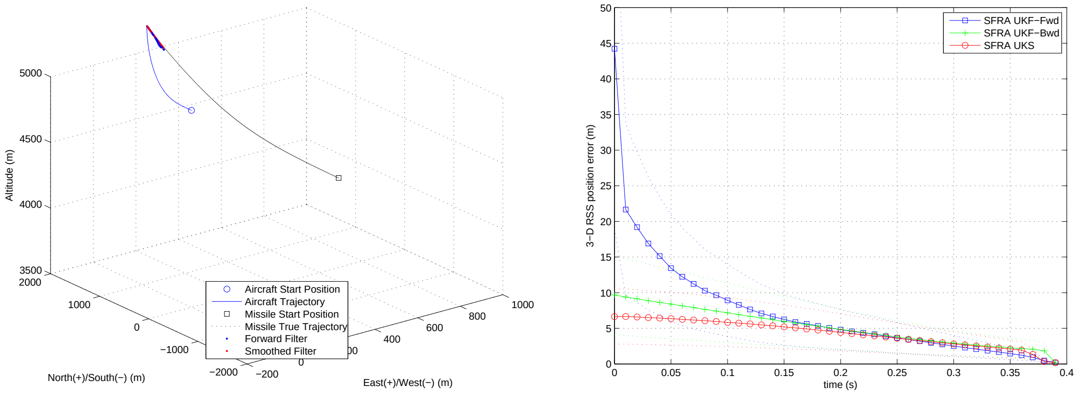
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

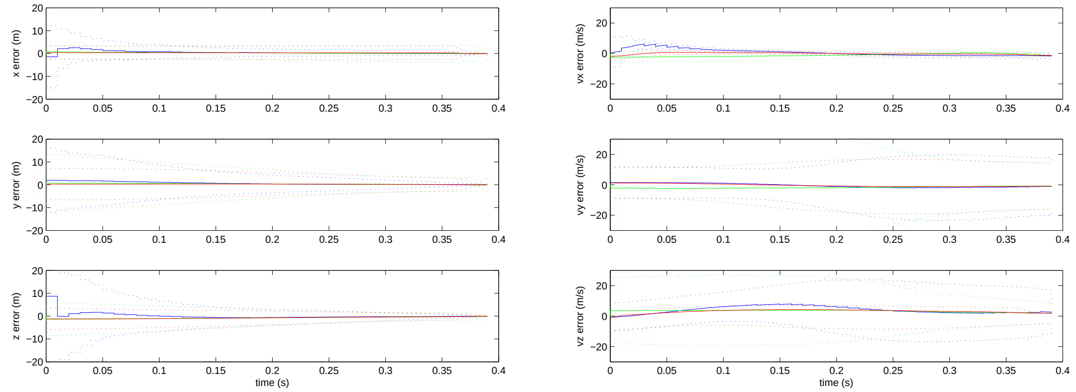
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.71: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

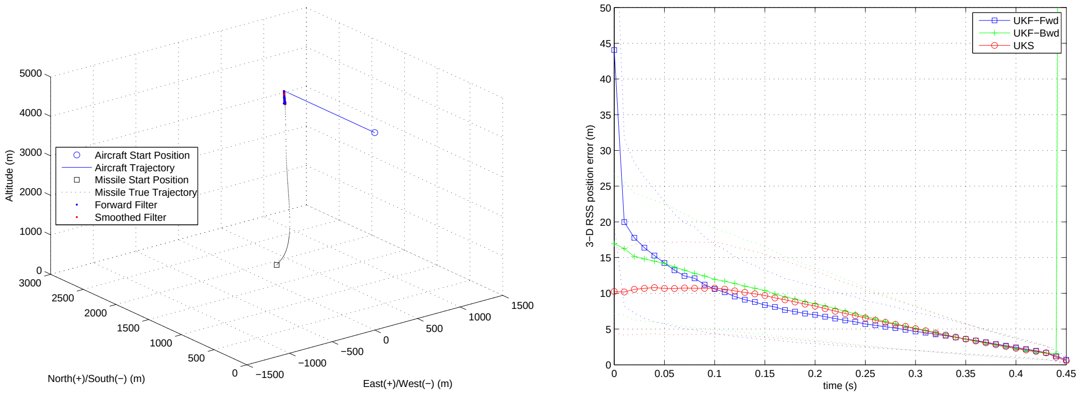


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

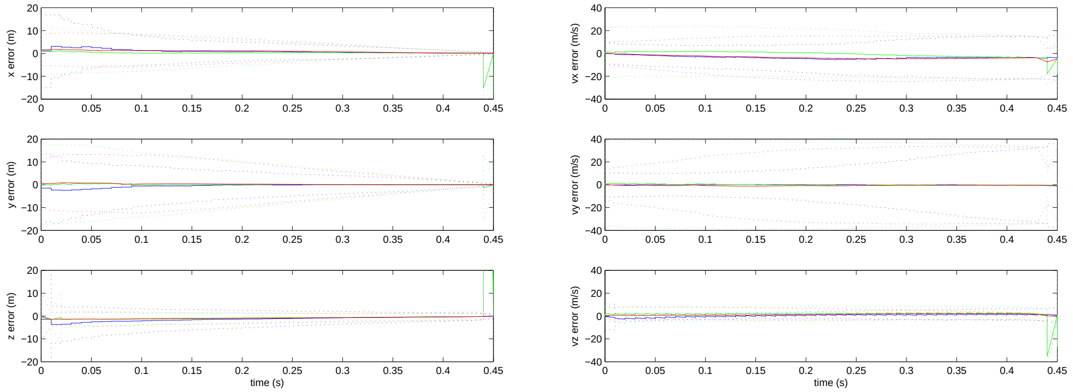
Figure A.72: SFRA Unscented Kalman Smoother 10% Sensor Noise Tuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.10 Unscented Kalman Smoother Simulations with 10% *Sensor Noise Untuned*



(a) 3D Aircraft and Missile Trajectory

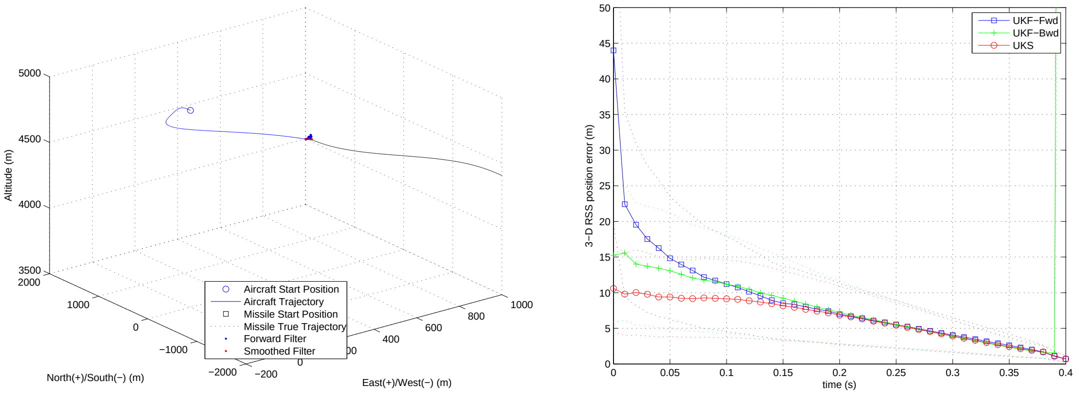
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

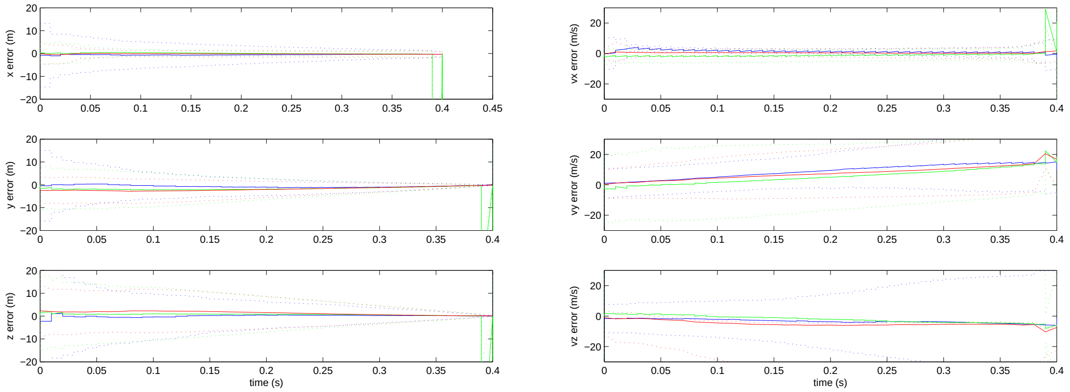
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.73: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

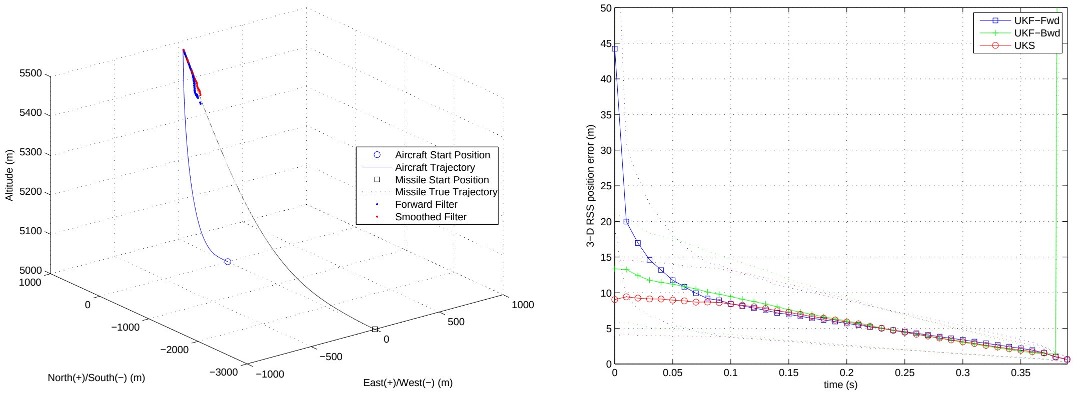
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

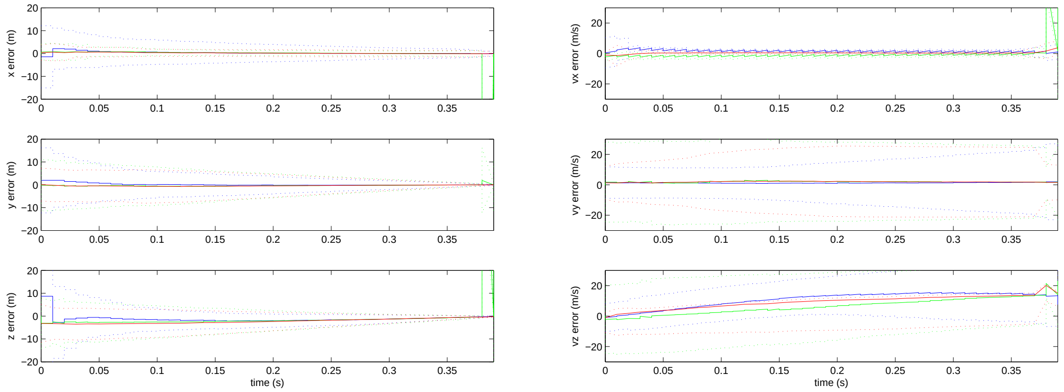
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.74: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

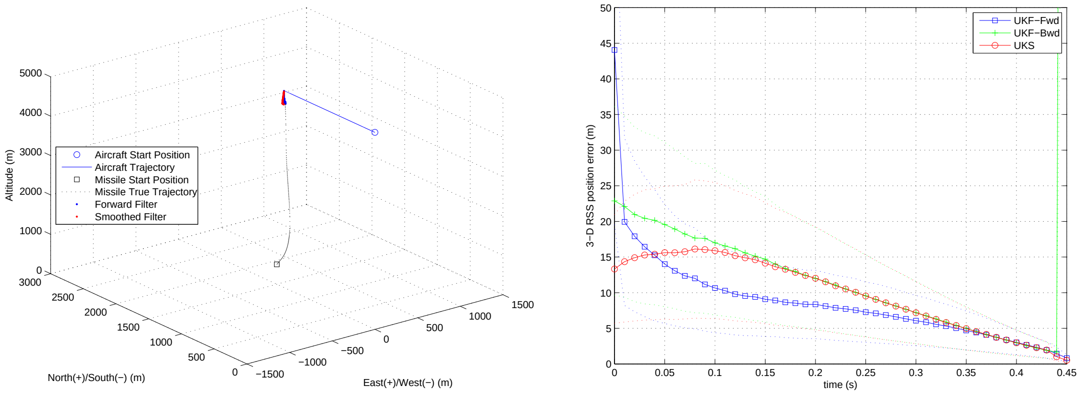
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

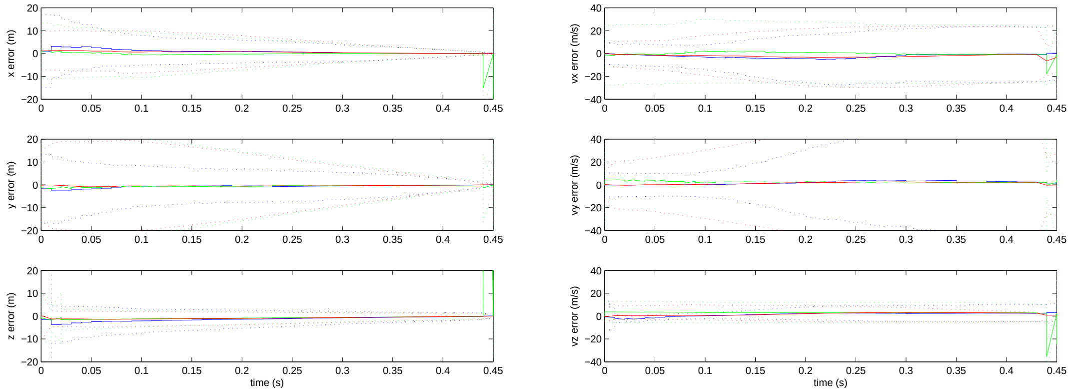
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.75: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

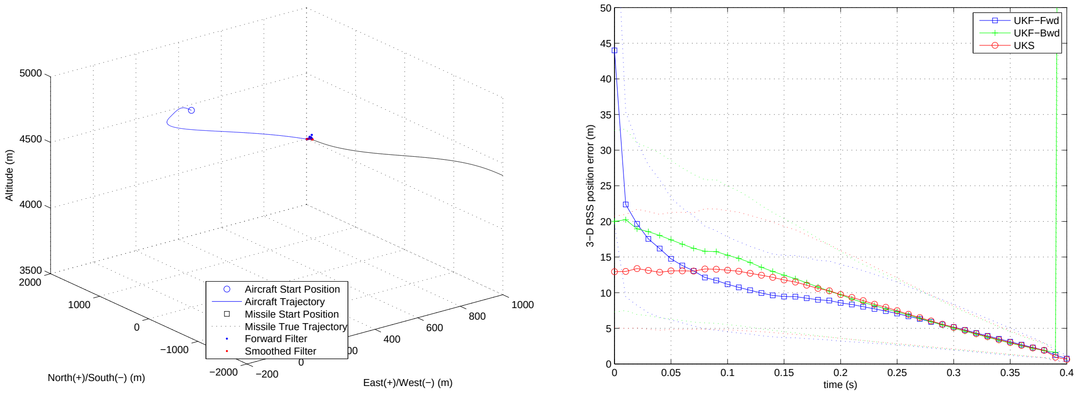
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

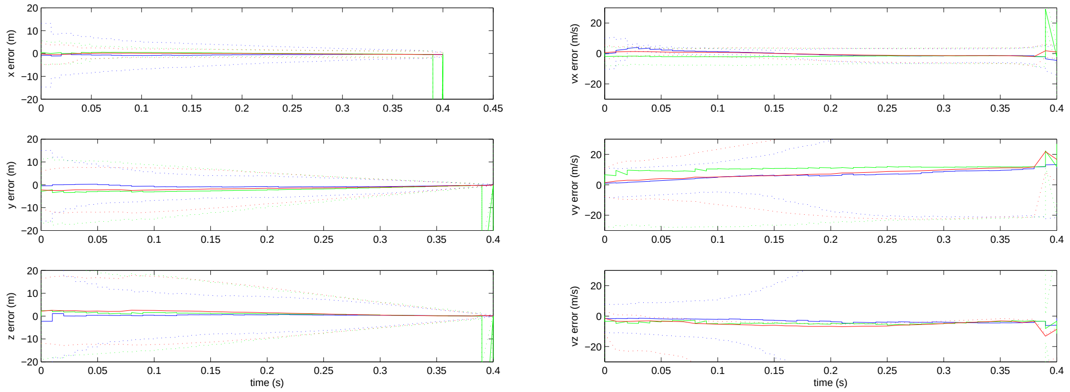
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.76: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

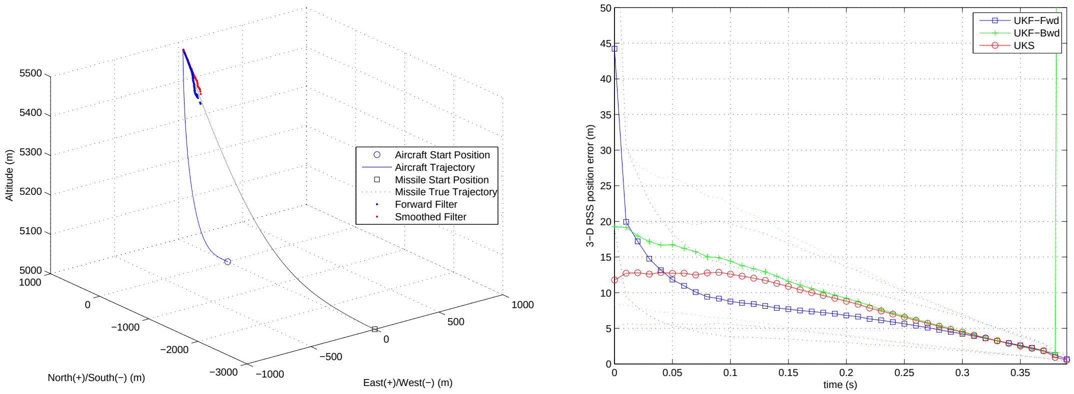
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

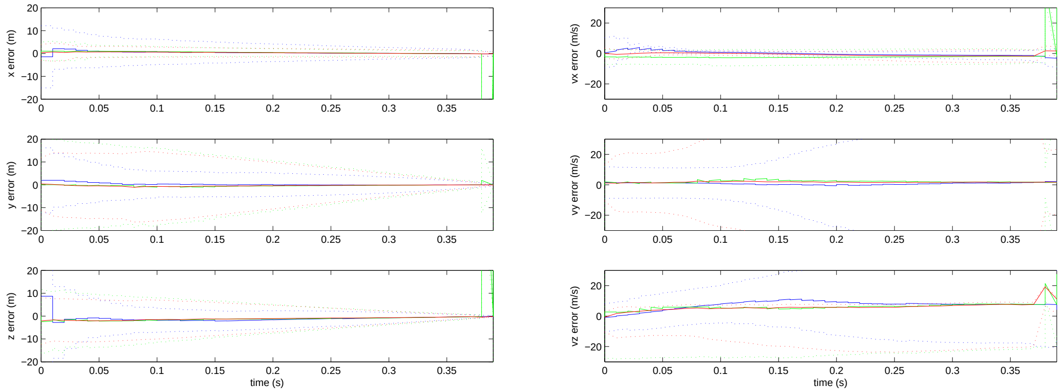
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.77: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

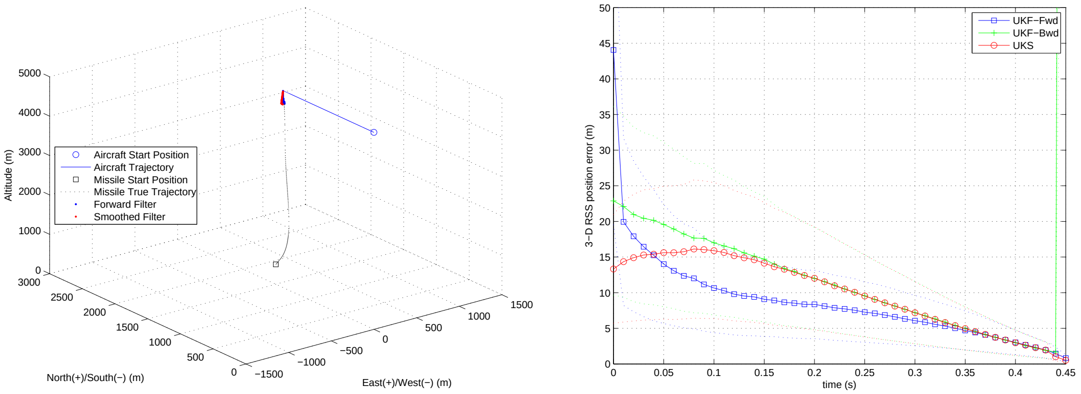
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

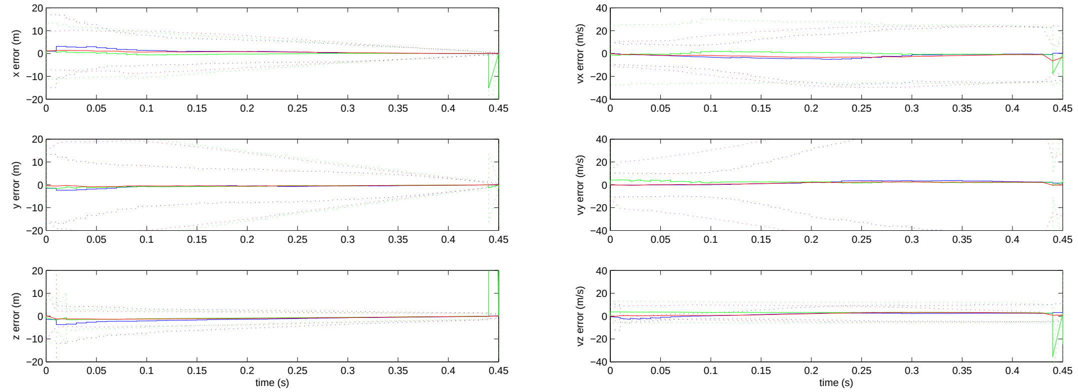
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.78: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

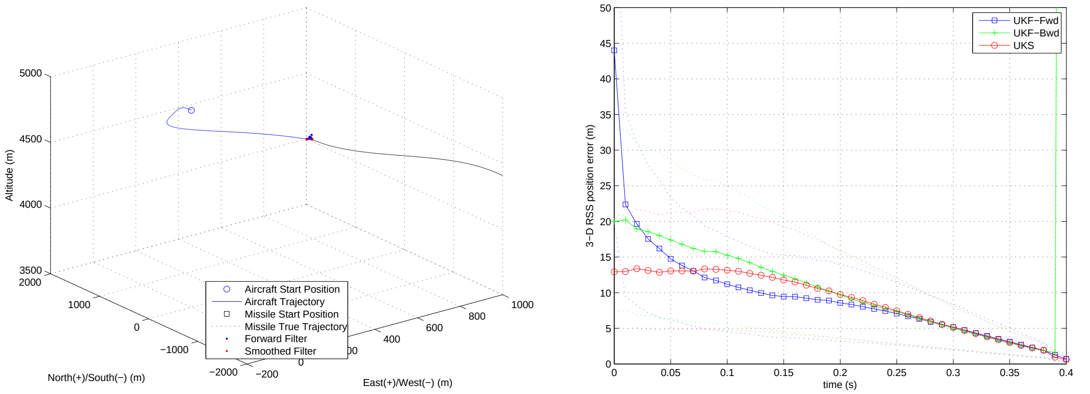
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

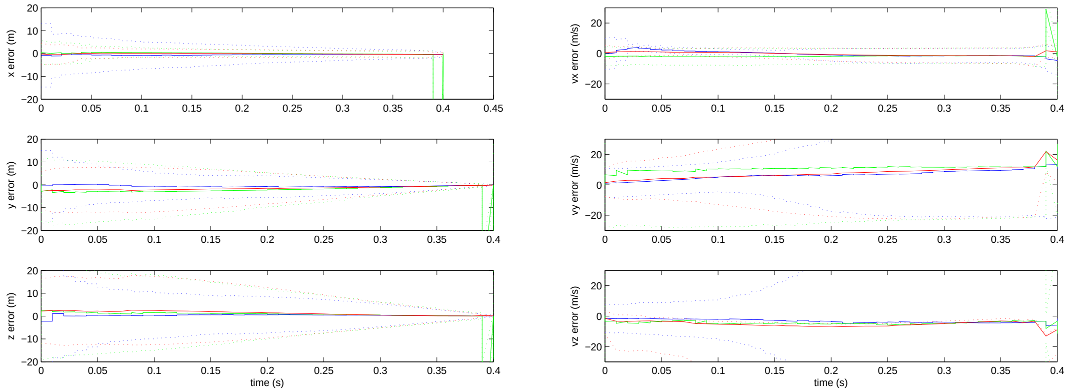
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.79: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

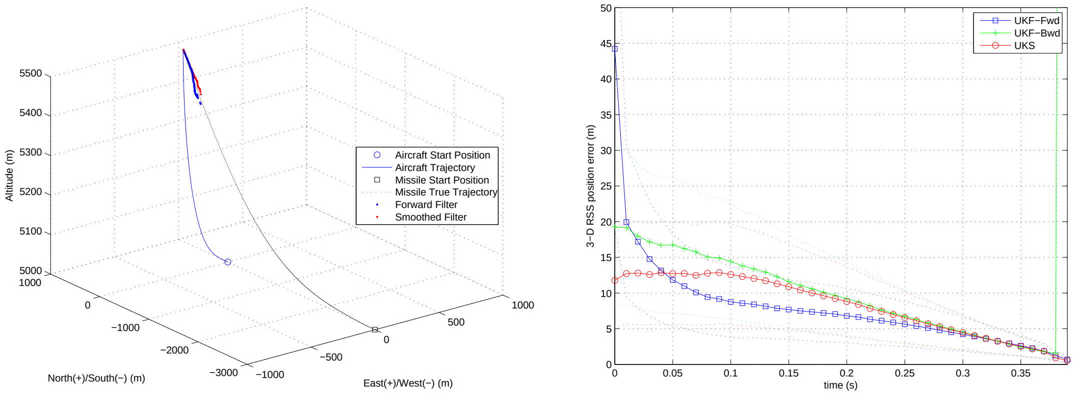
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

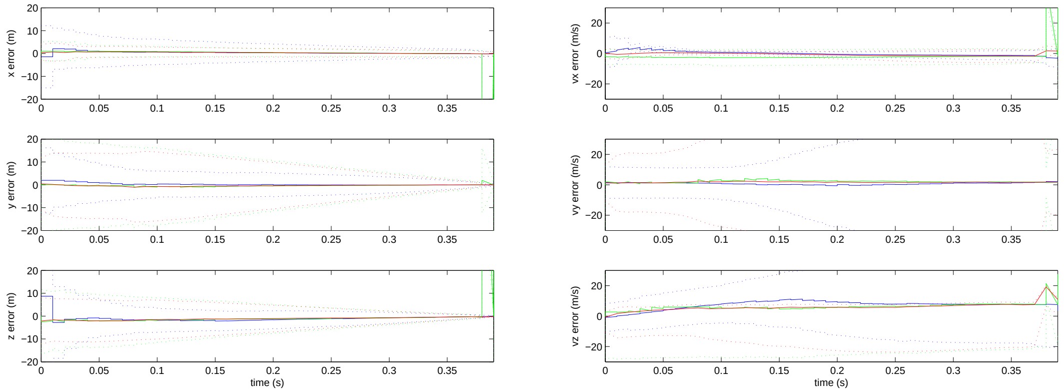
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.80: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

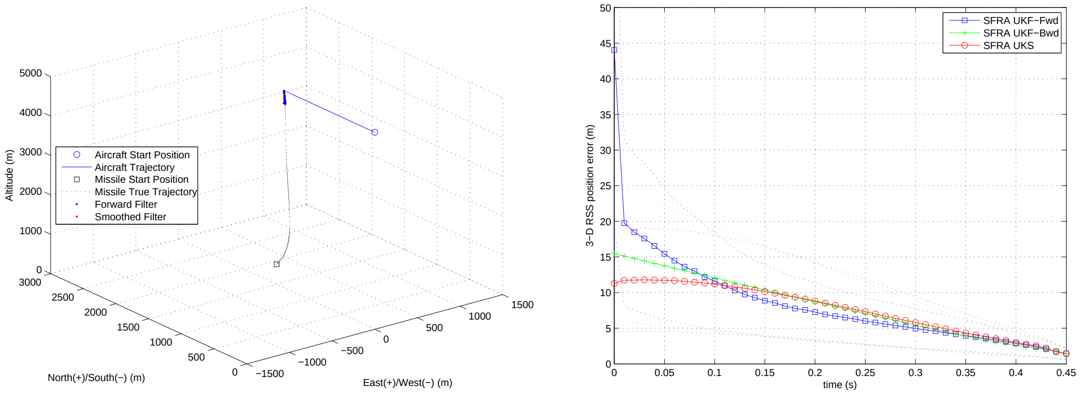


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

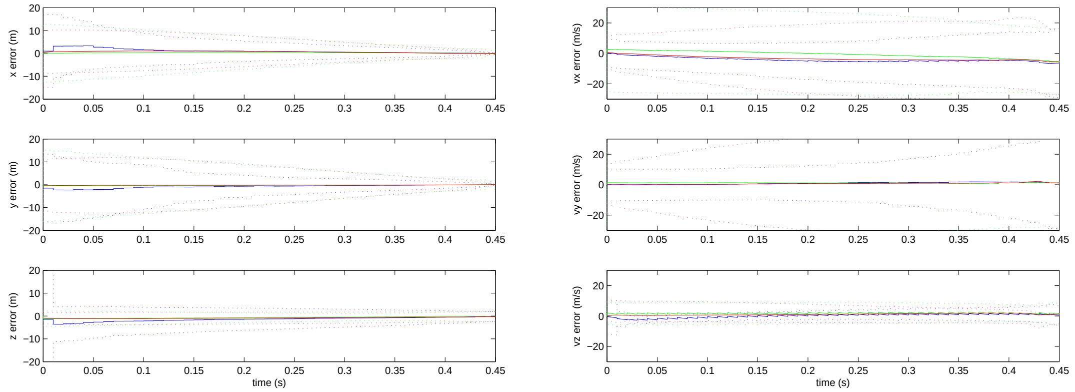
Figure A.81: Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.11 *SFRA Unscented Kalman Smoother Simulations with* 10% *Sensor Noise Untuned*



(a) 3D Aircraft and Missile Trajectory

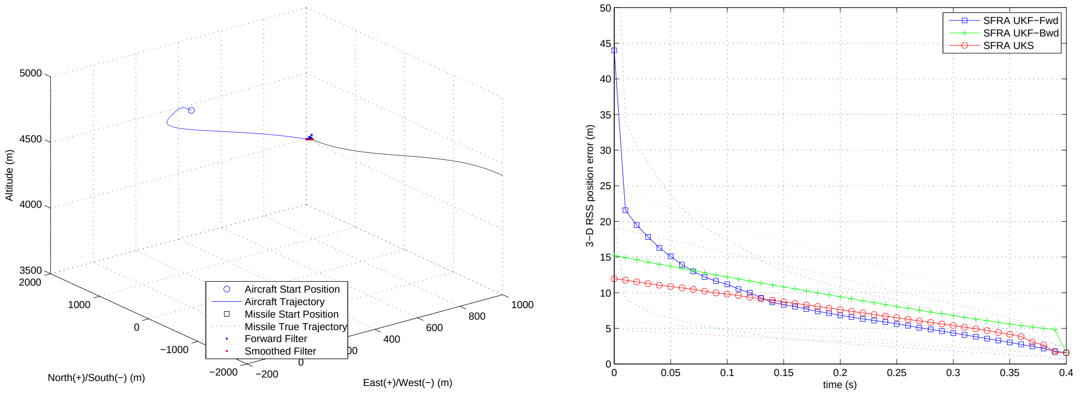
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



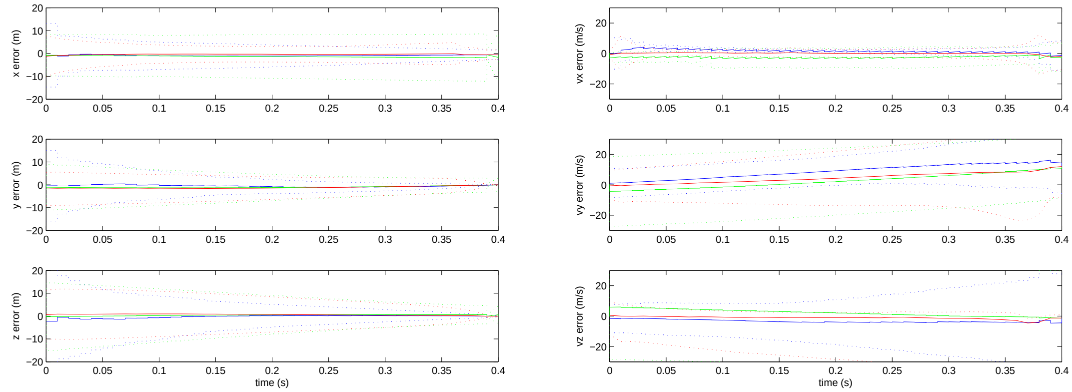
(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.82: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)

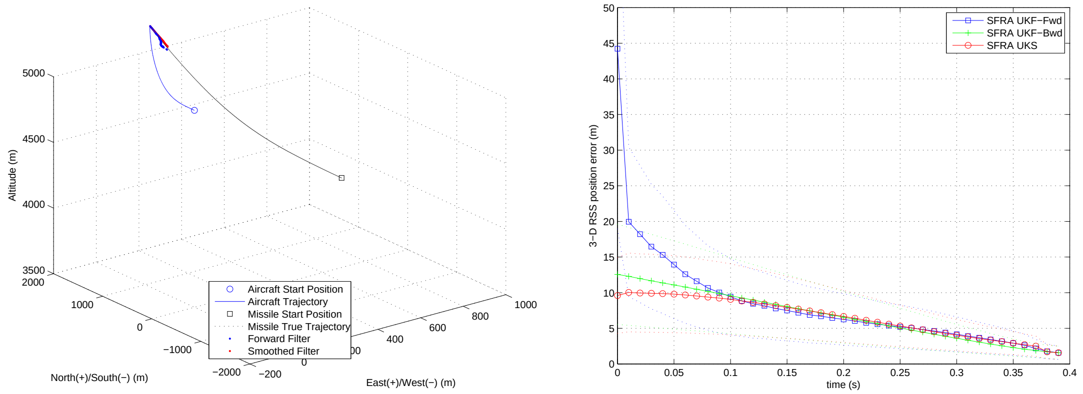


(a) 3D Aircraft and Missile Trajectory (b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



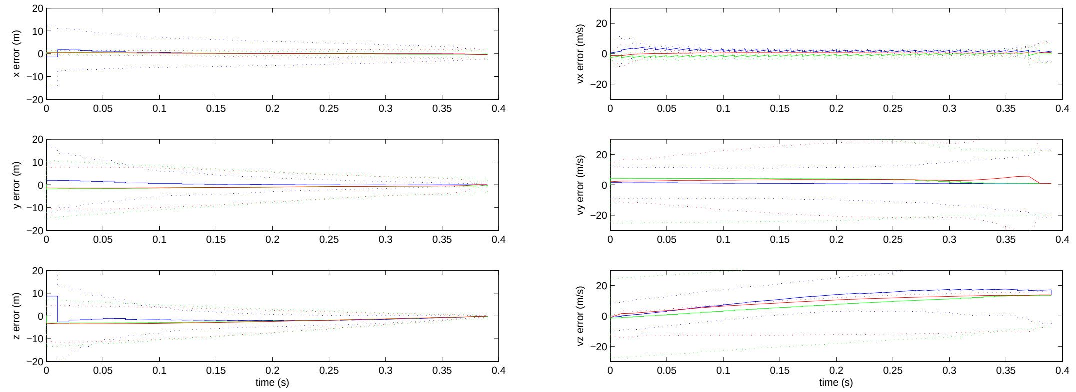
(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs) (d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.83: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

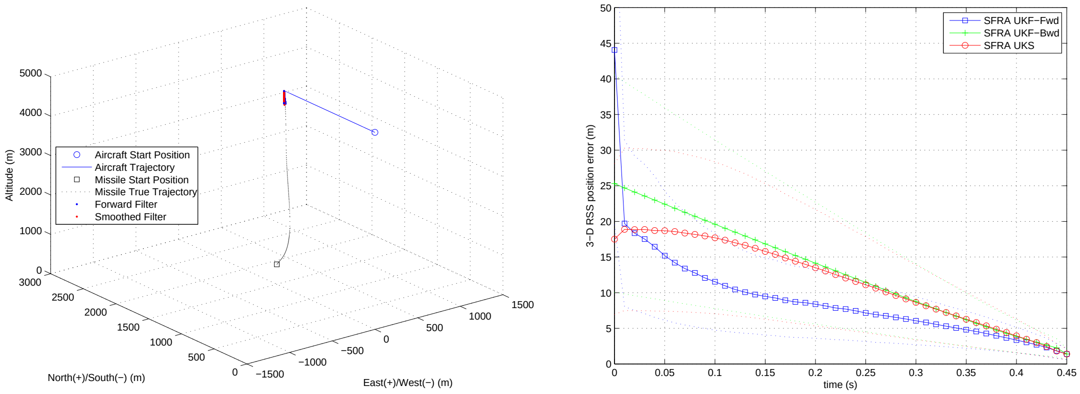
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

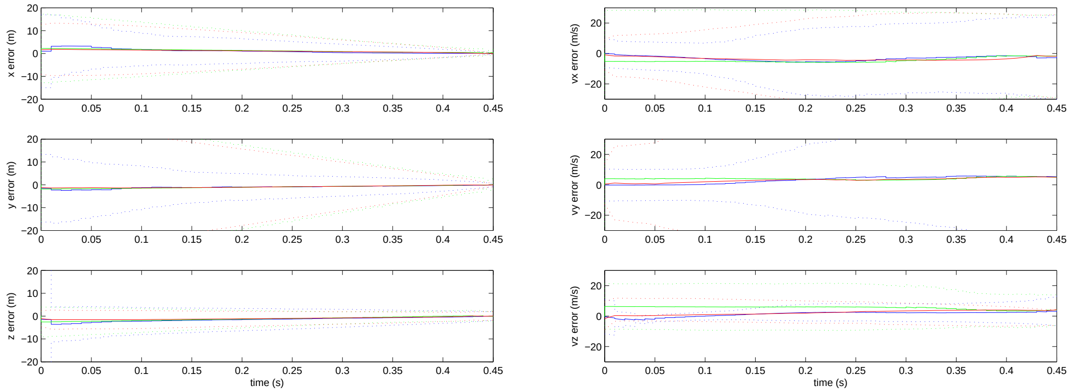
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.84: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

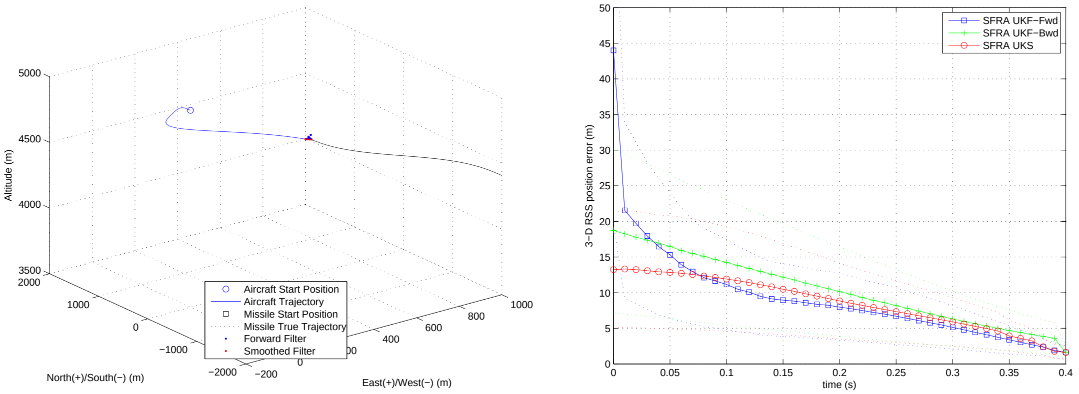
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

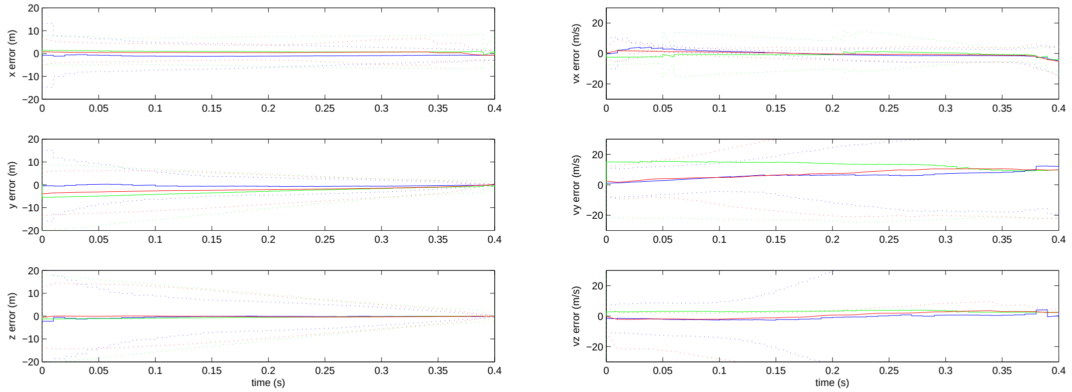
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.85: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

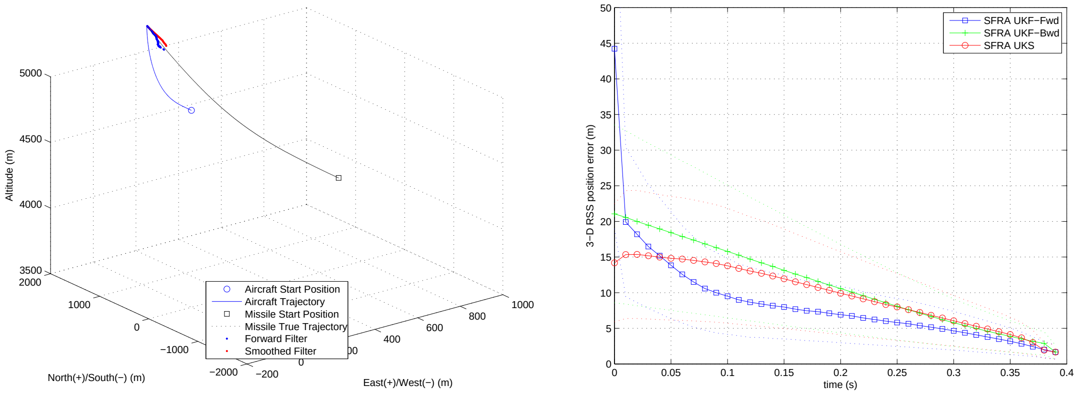
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

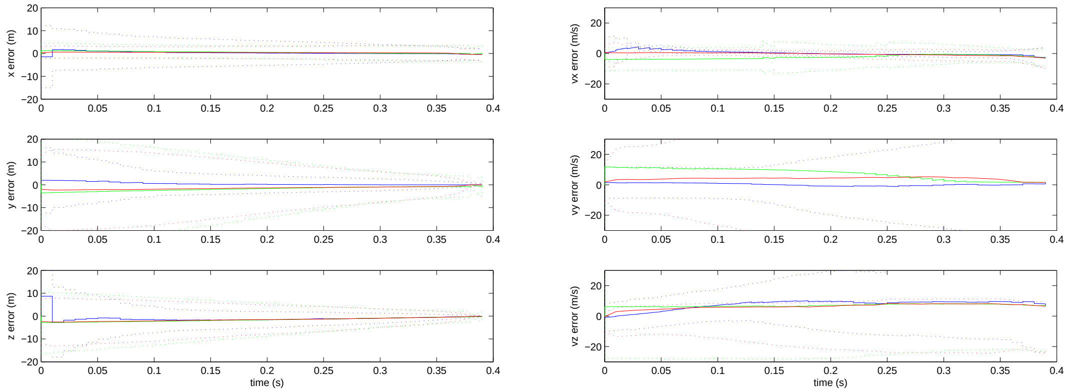
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.86: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

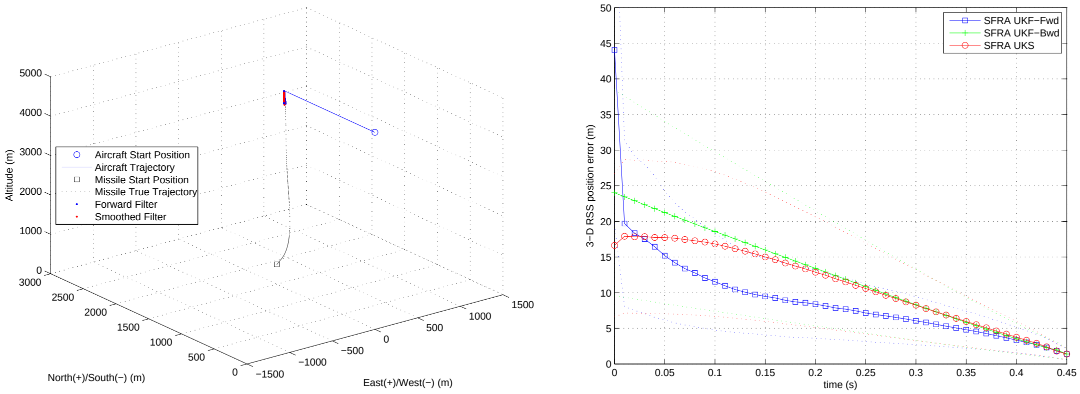
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

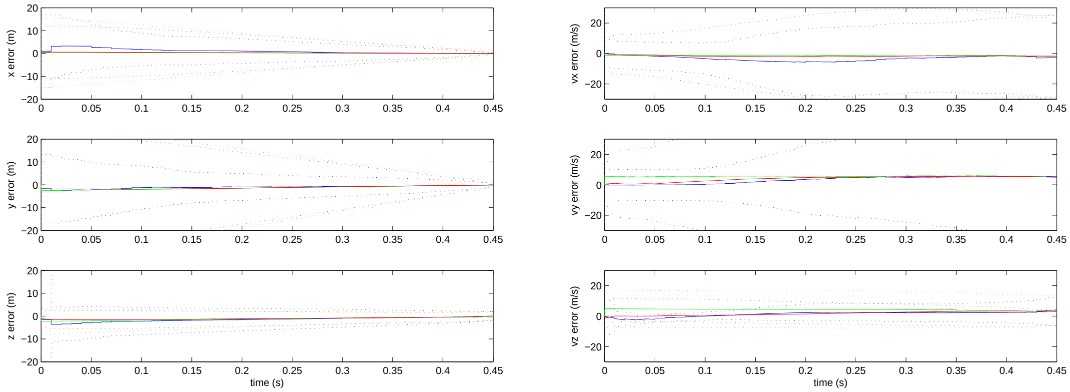
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.87: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

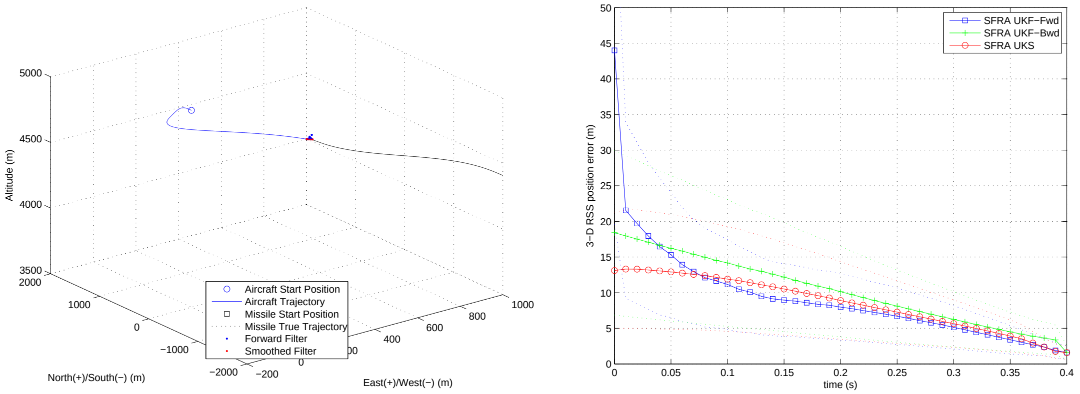
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

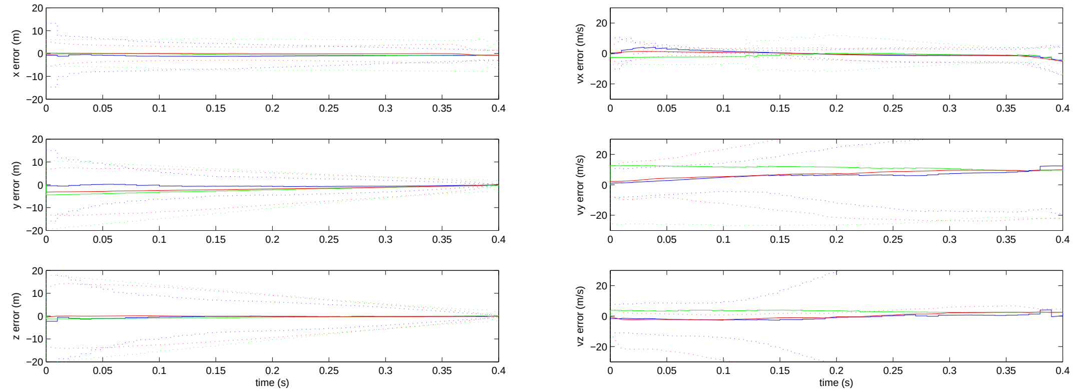
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.88: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

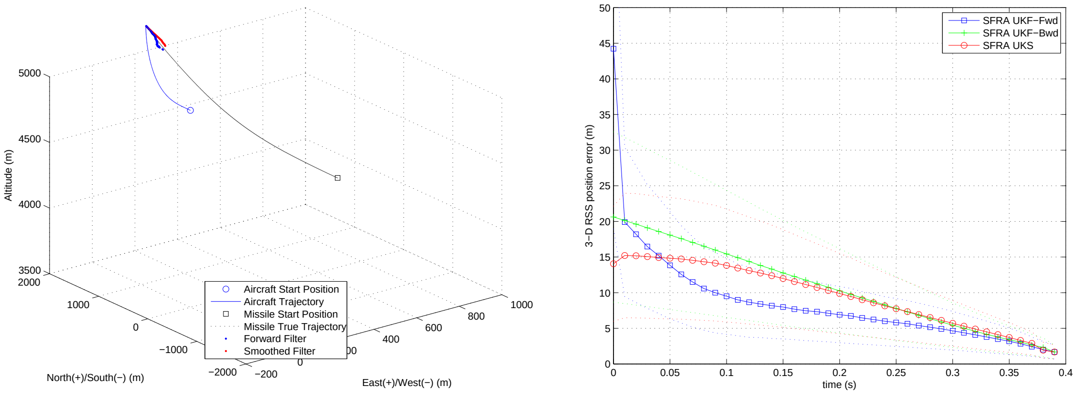
(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

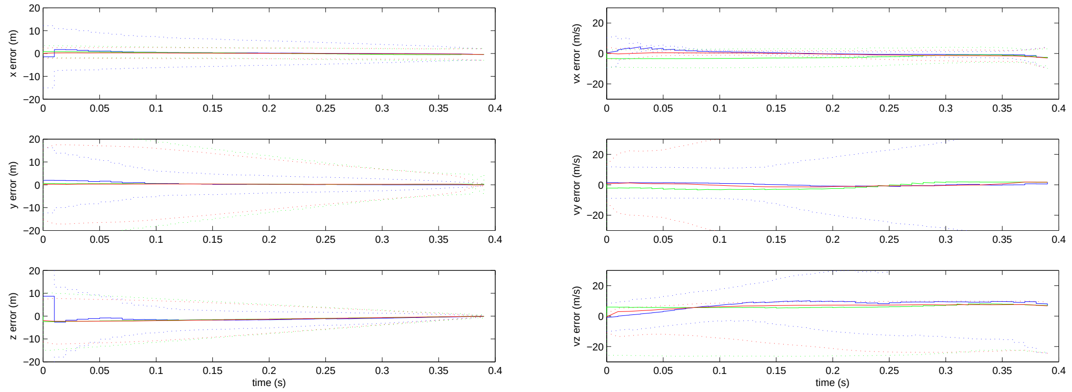
(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.89: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Root-Sum-Squared Error in Missile Position Estimate (100 Runs)

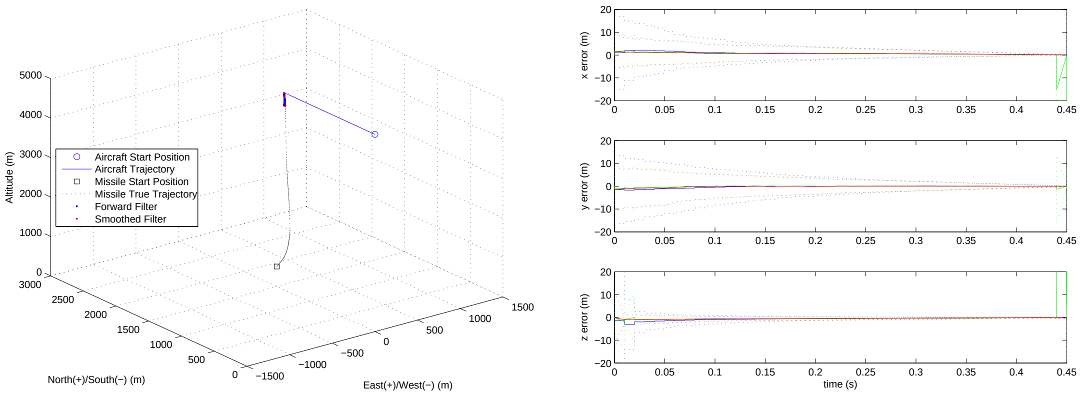


(c) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)

(d) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

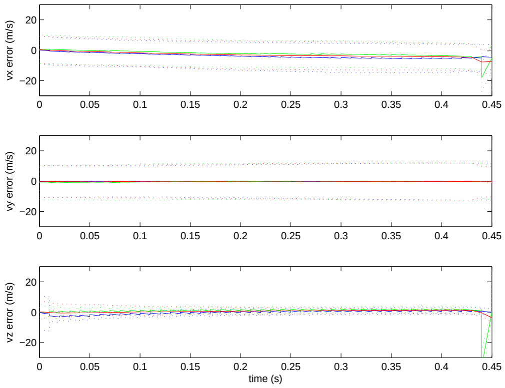
Figure A.90: SFRA Unscented Kalman Smoother 10% Sensor Noise Untuned Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

A.12 Unscented Kalman Smoother Simulations with Sensor Dropout



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.91: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)

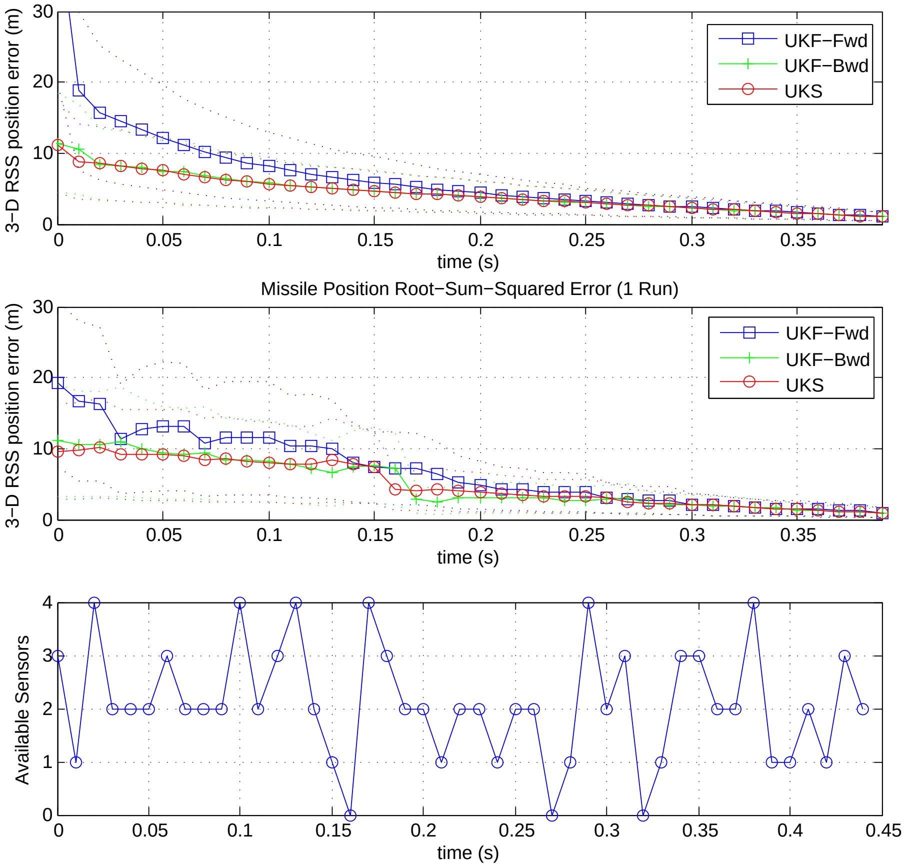
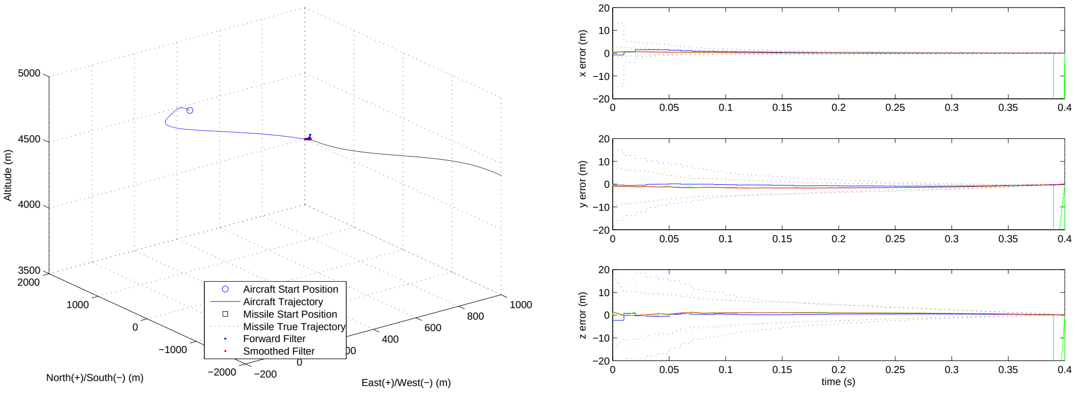
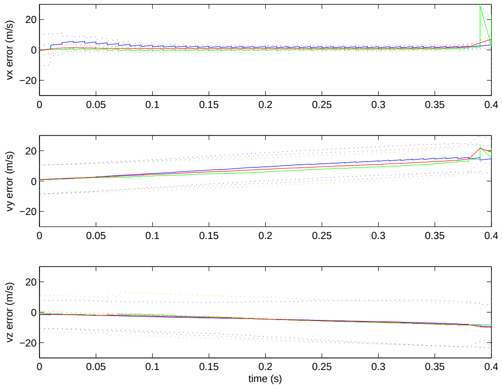


Figure A.92: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.93: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)

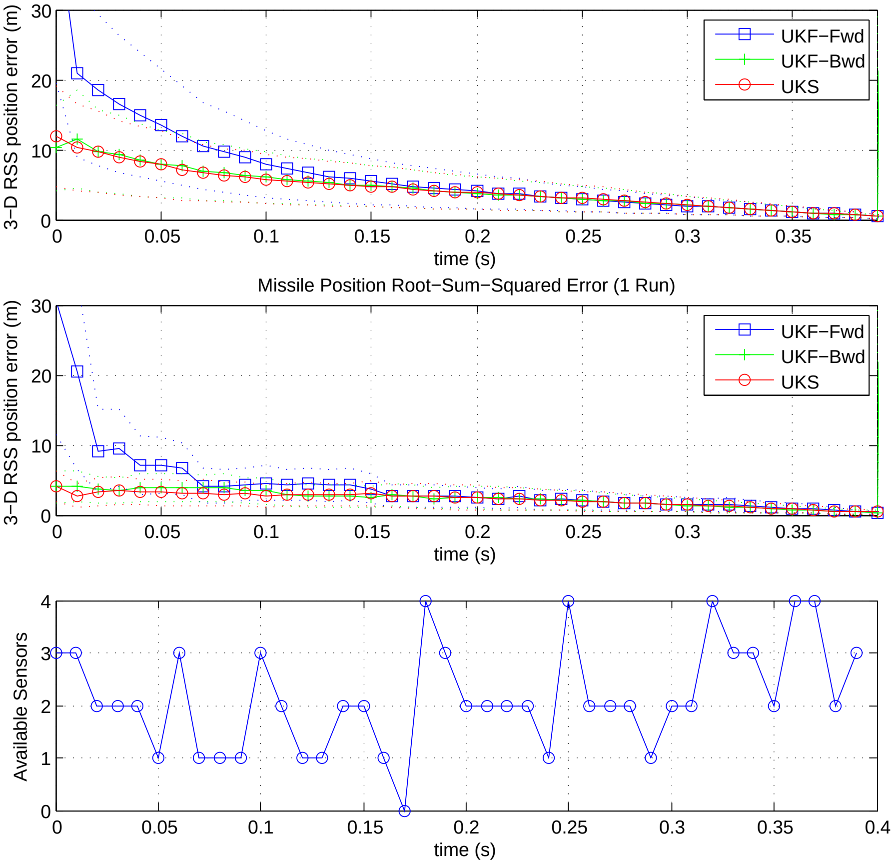
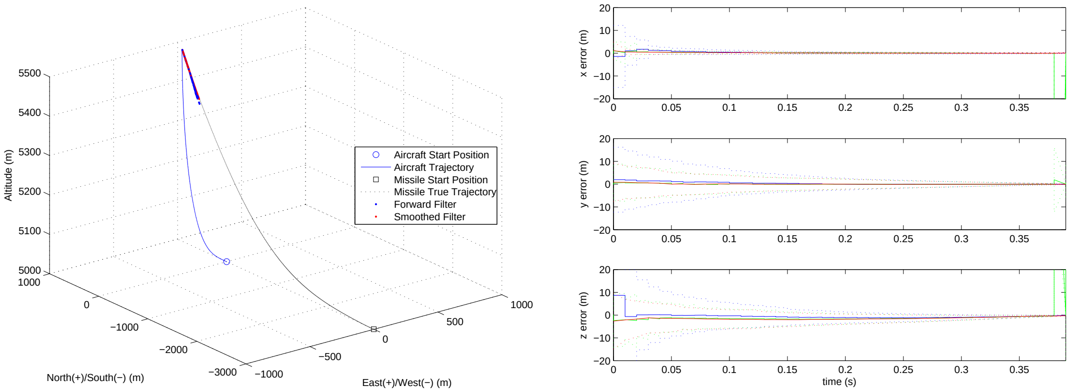
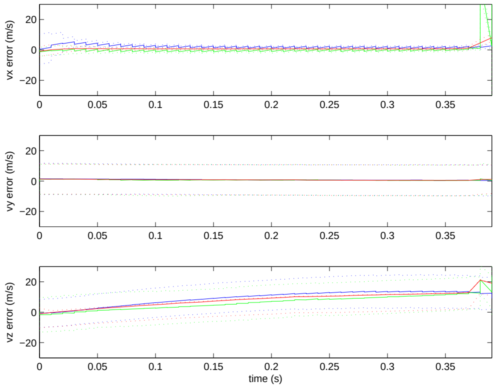

Missile Position Root−Sum−Squared Error (1 Run)

Figure A.94: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.95: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)

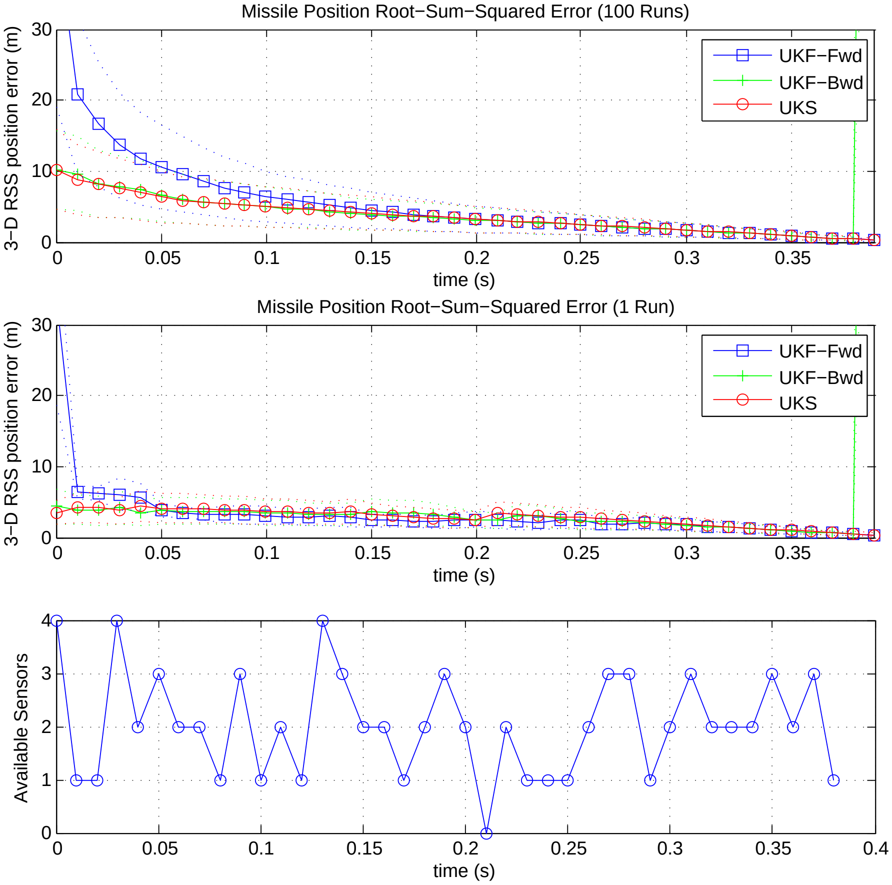
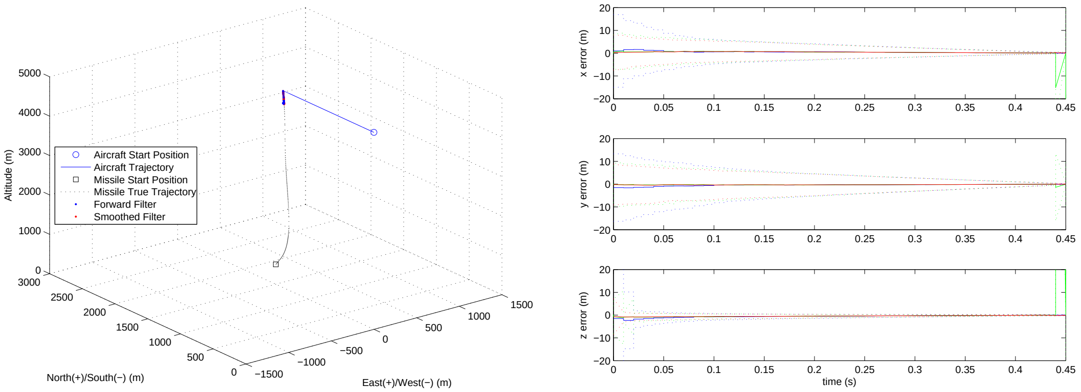
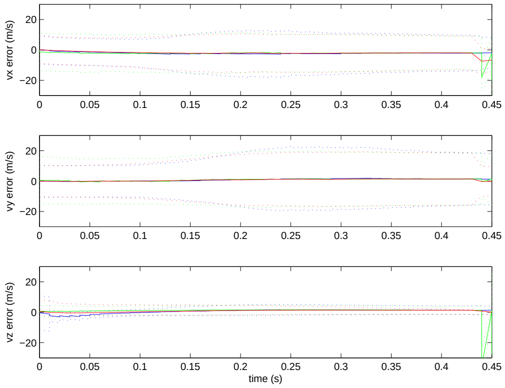


Figure A.96: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.97: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)

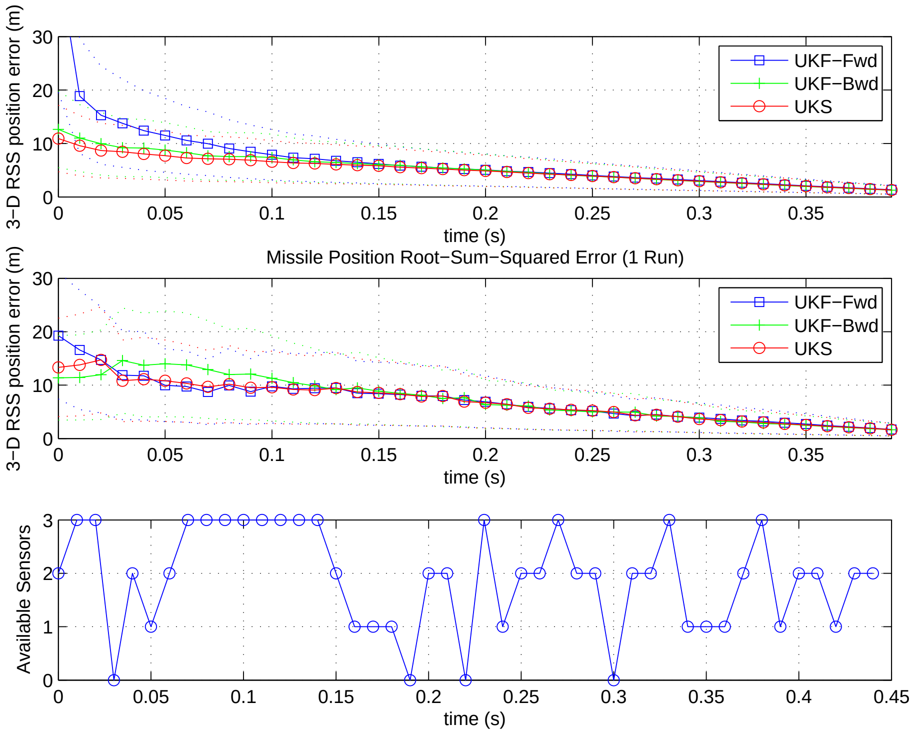
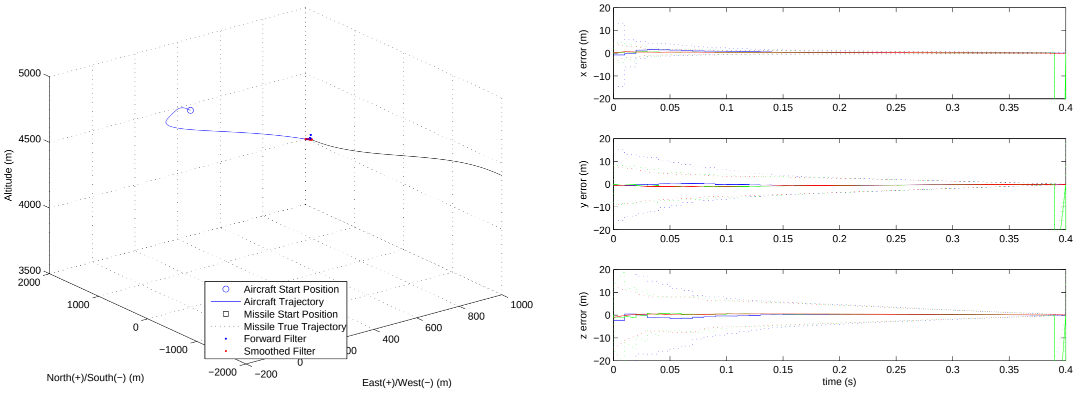
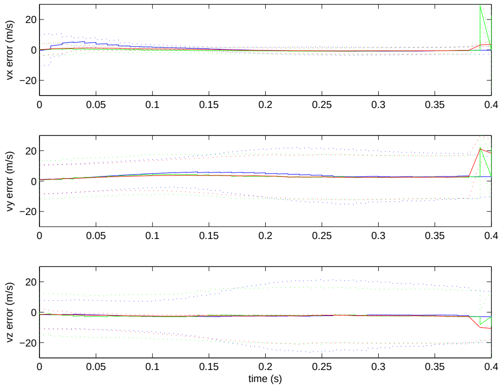


Figure A.98: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.99: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)

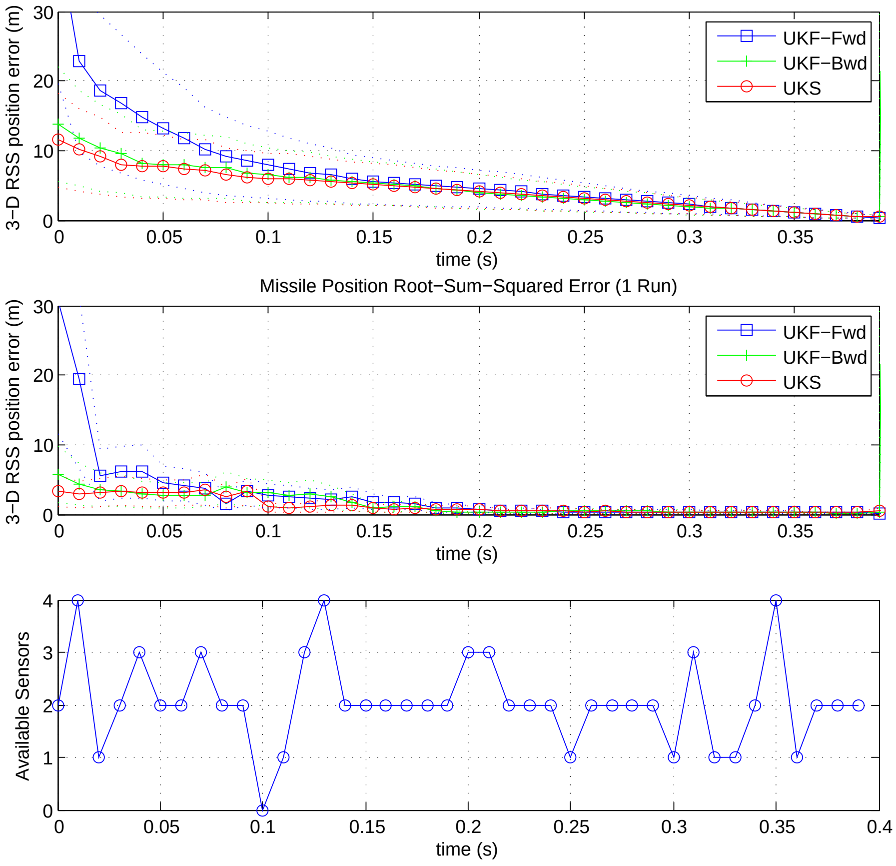
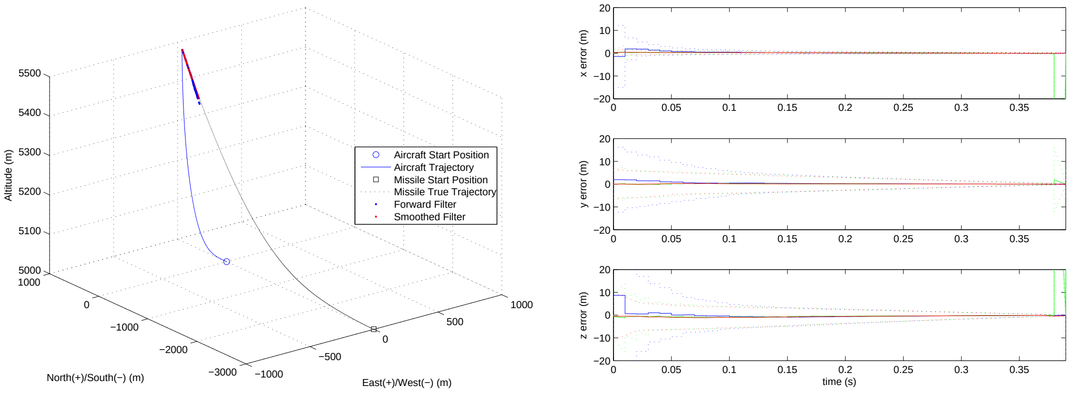
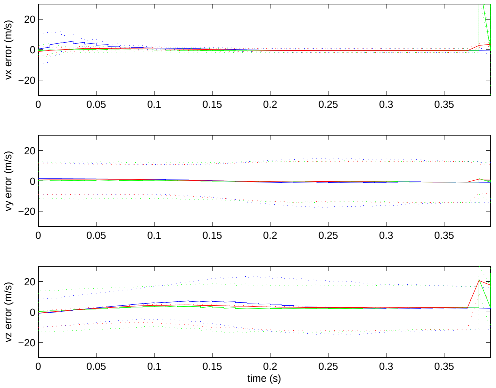


Figure A.100: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.101: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)

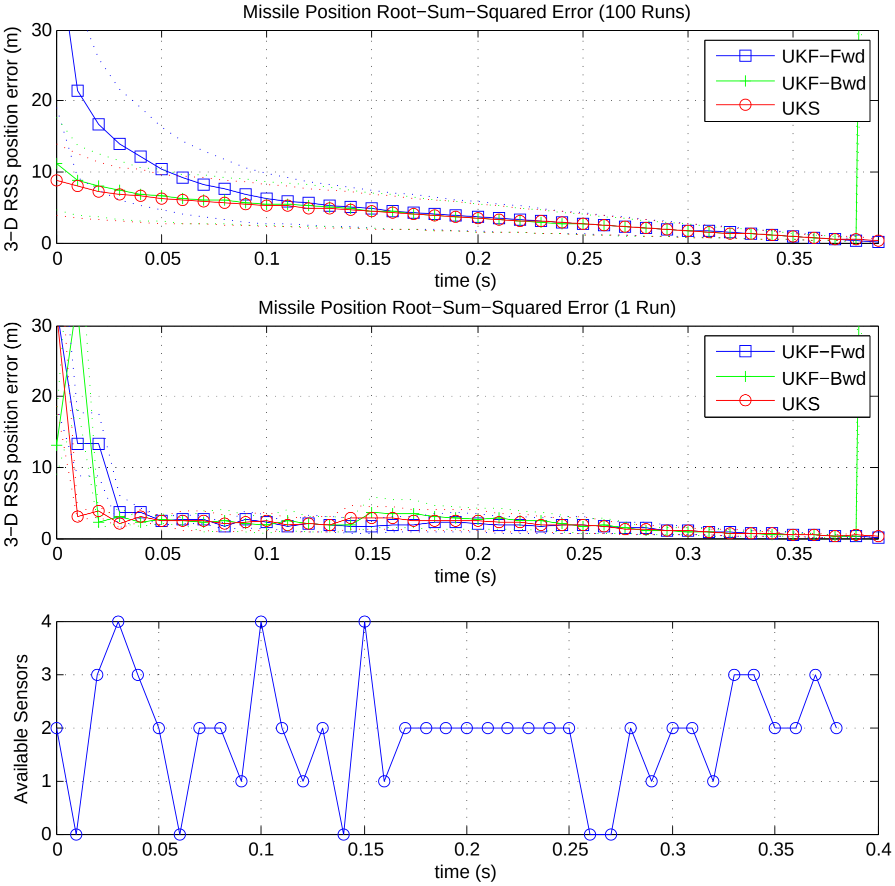
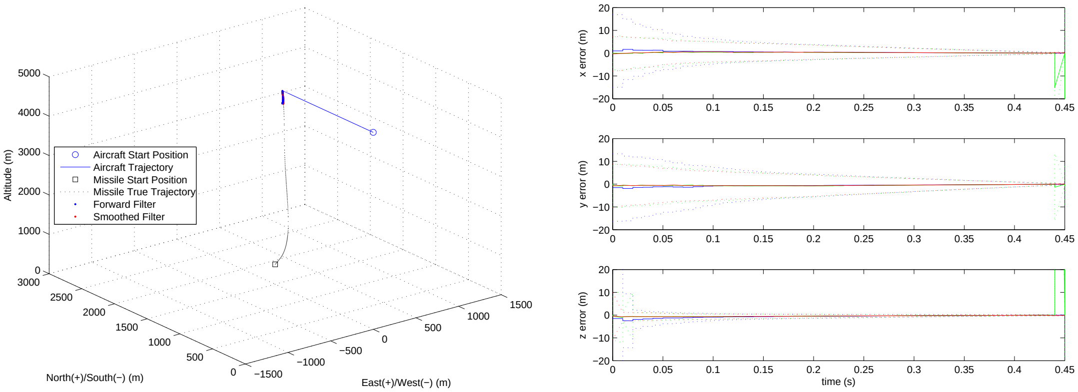
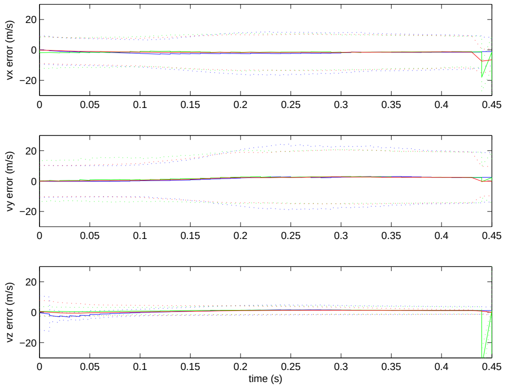


Figure A.102: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.103: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)

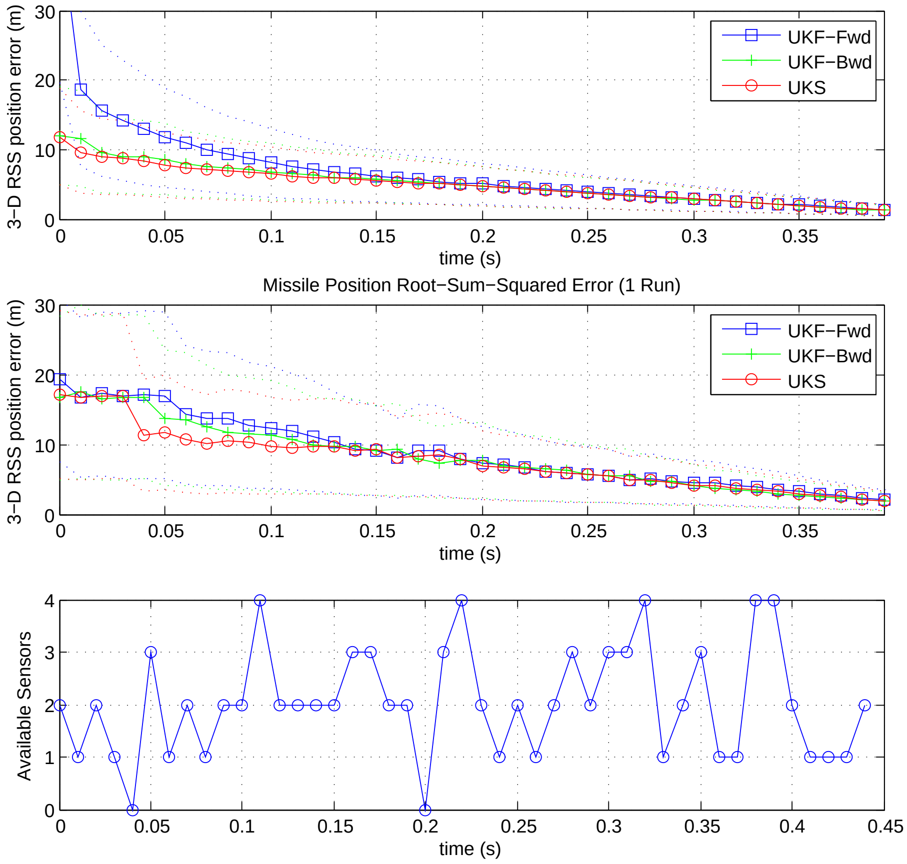
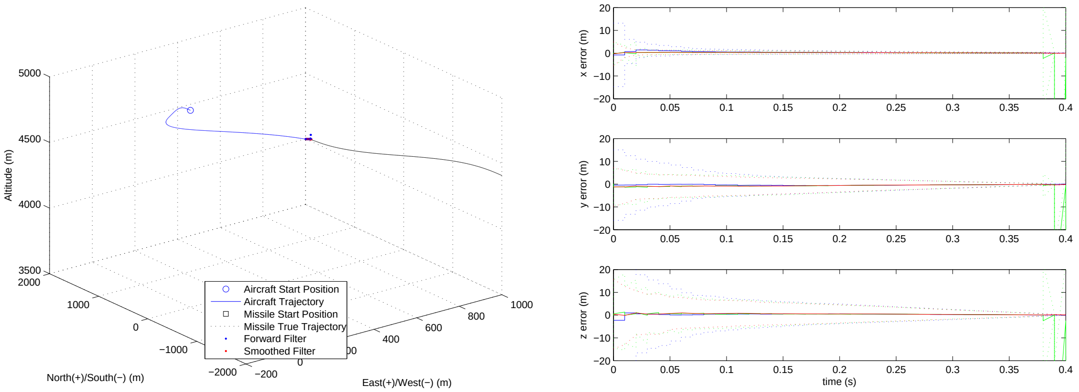
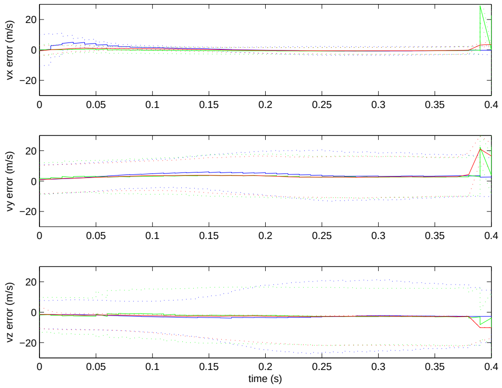


Figure A.104: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.105: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)

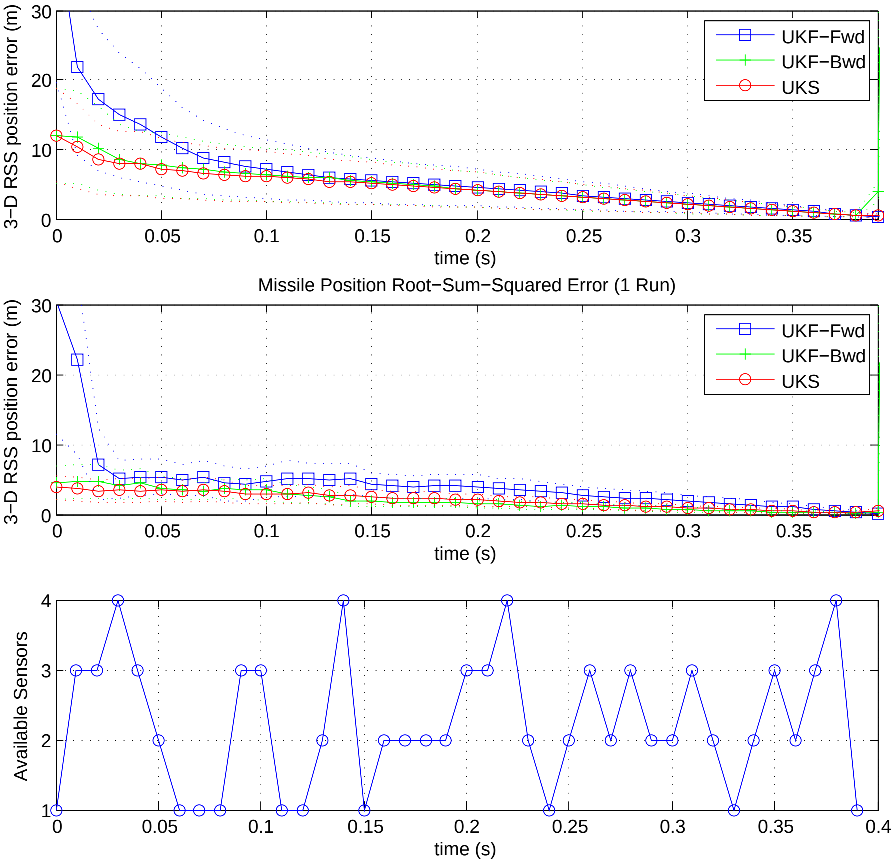
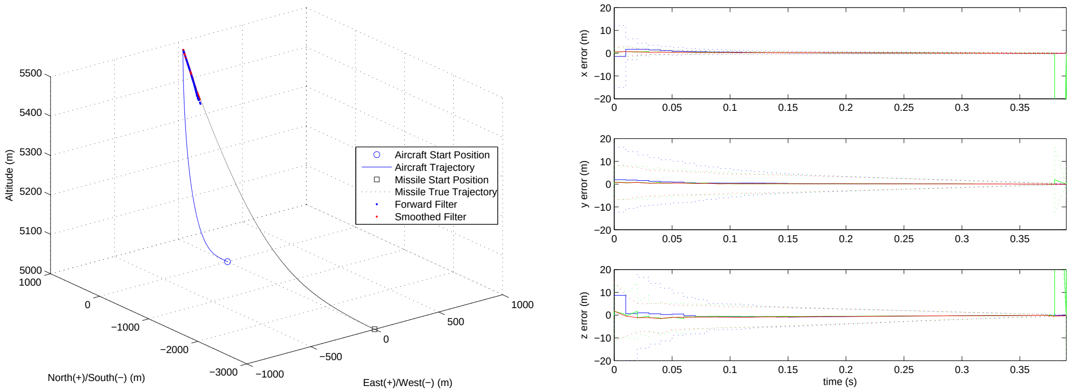
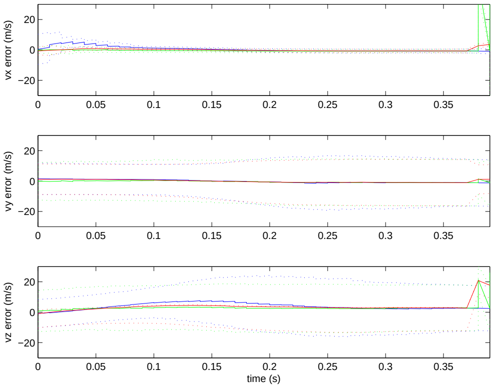


Figure A.106: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.107: Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

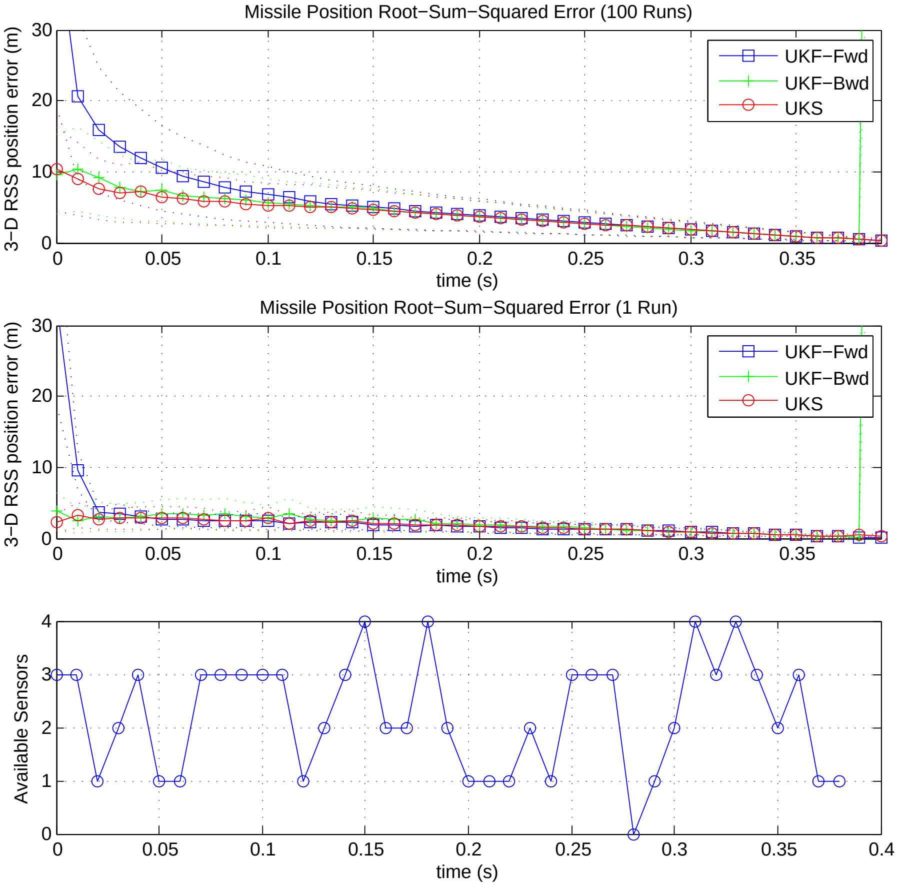
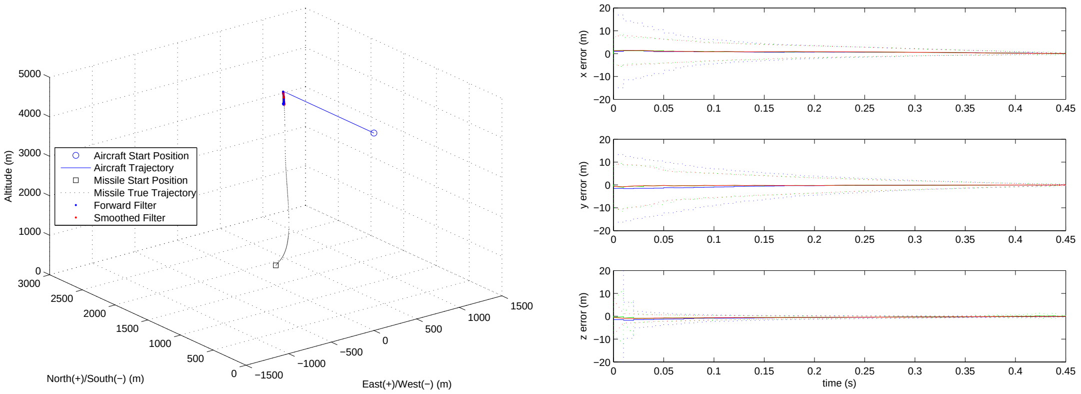


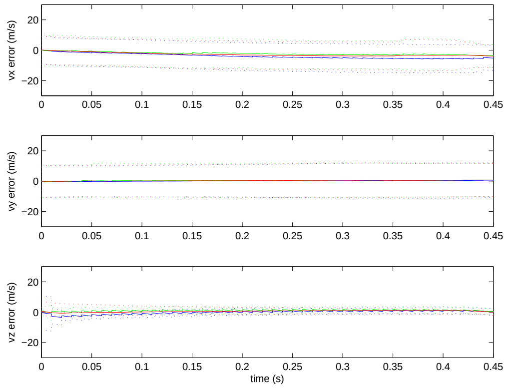
Figure A.108: Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Coordinated Turn Dynamics Model (Scenario 3)

A.13 SFRA Unscented Kalman Smoother Simulations with Sensor Dropout



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.109: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 1)

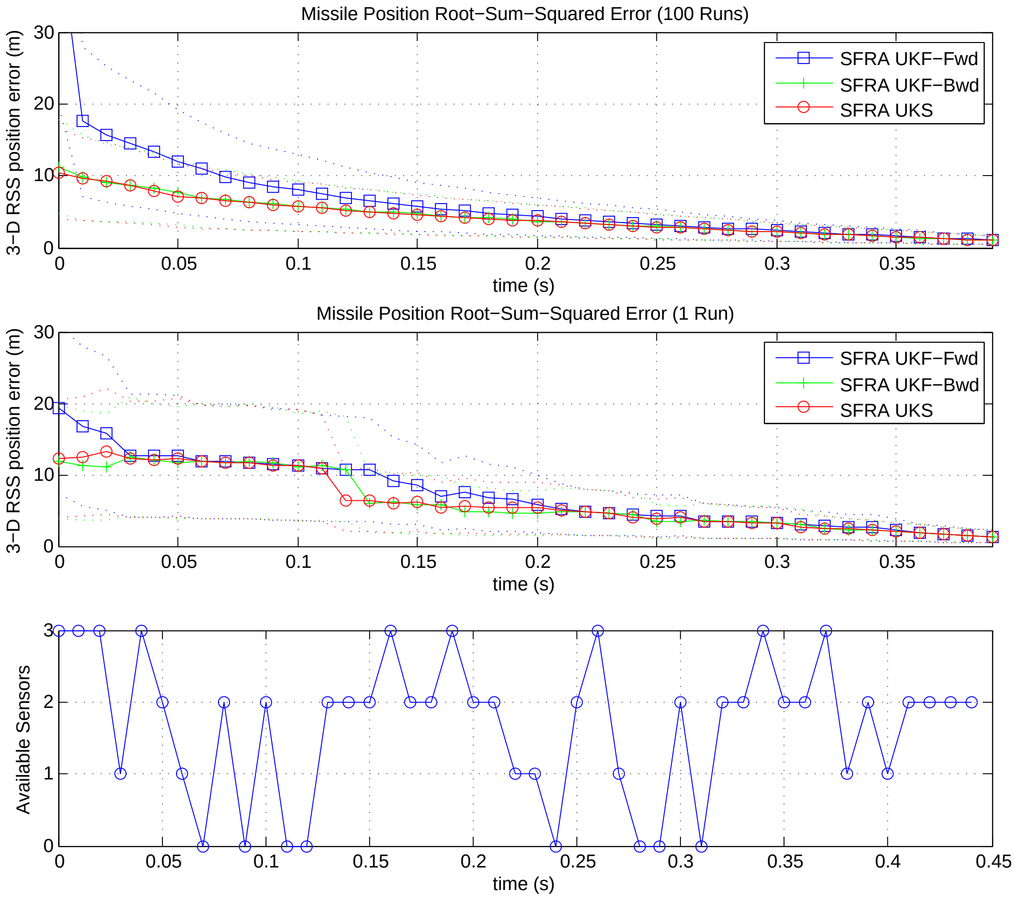
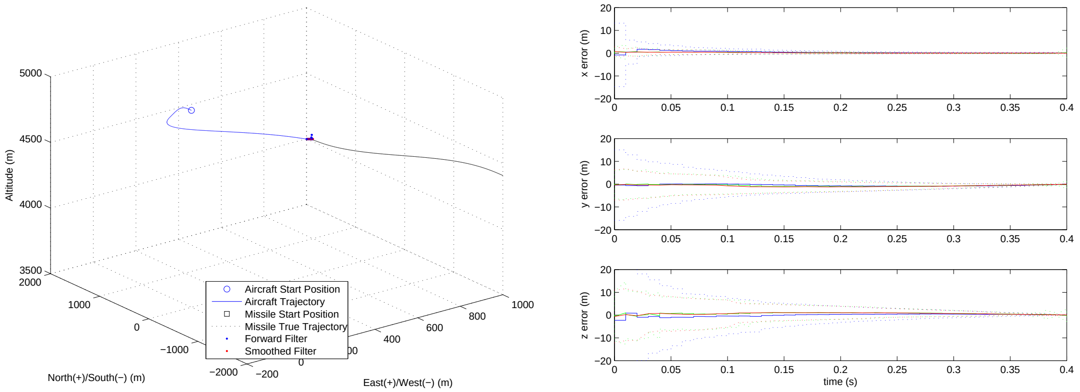
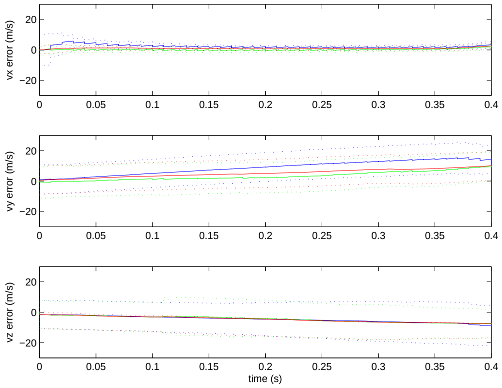


Figure A.110: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Continuous Velocity Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.111: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 2)

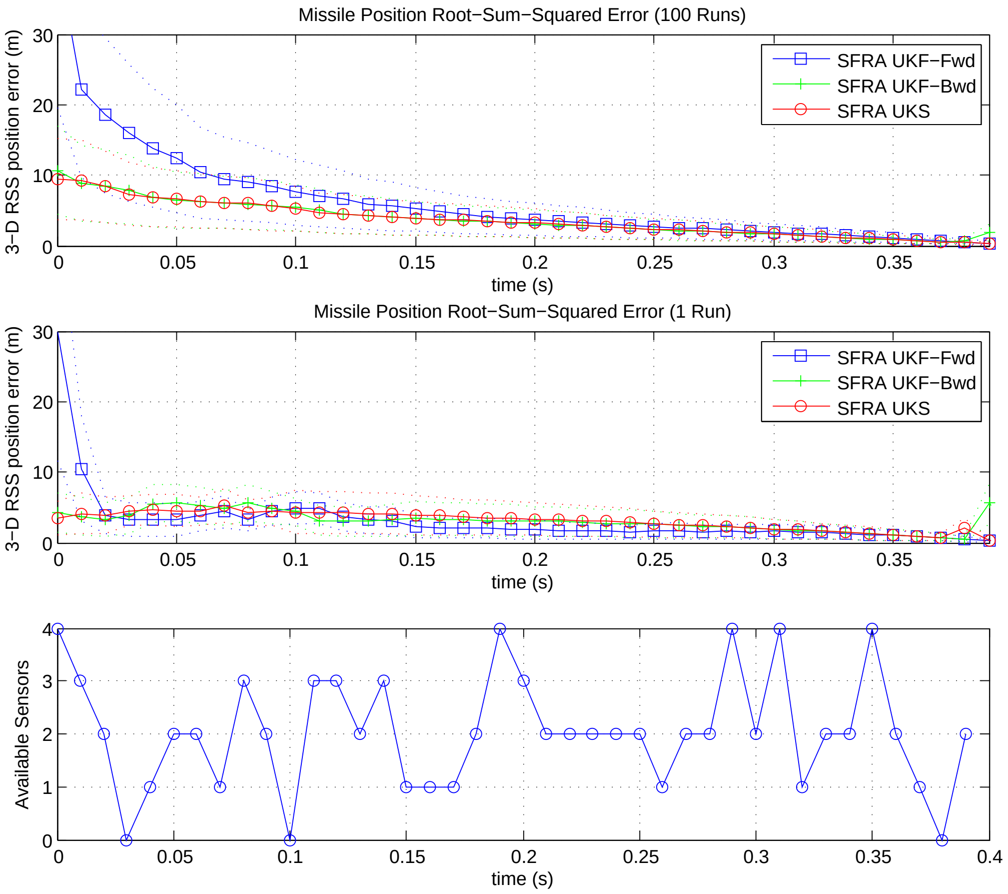
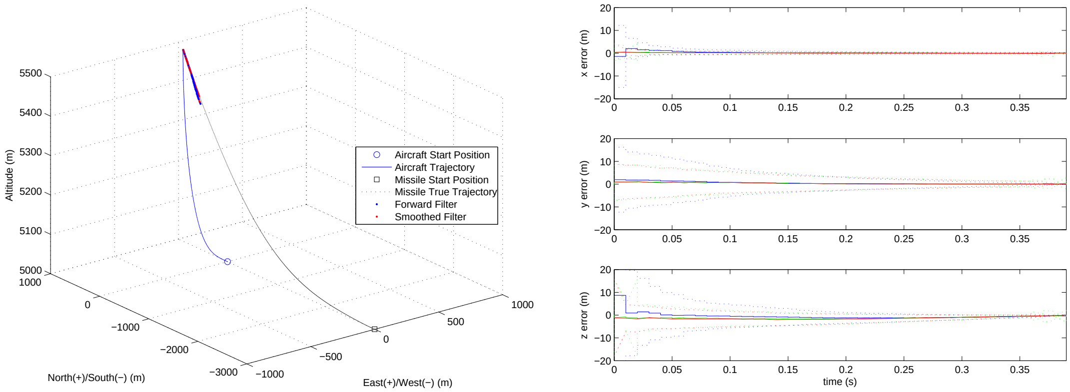
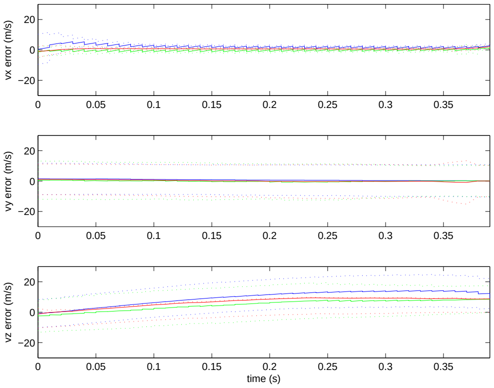


Figure A.112: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Continuous Velocity Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.113: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Continuous Velocity Dynamics Model (Scenario 3)

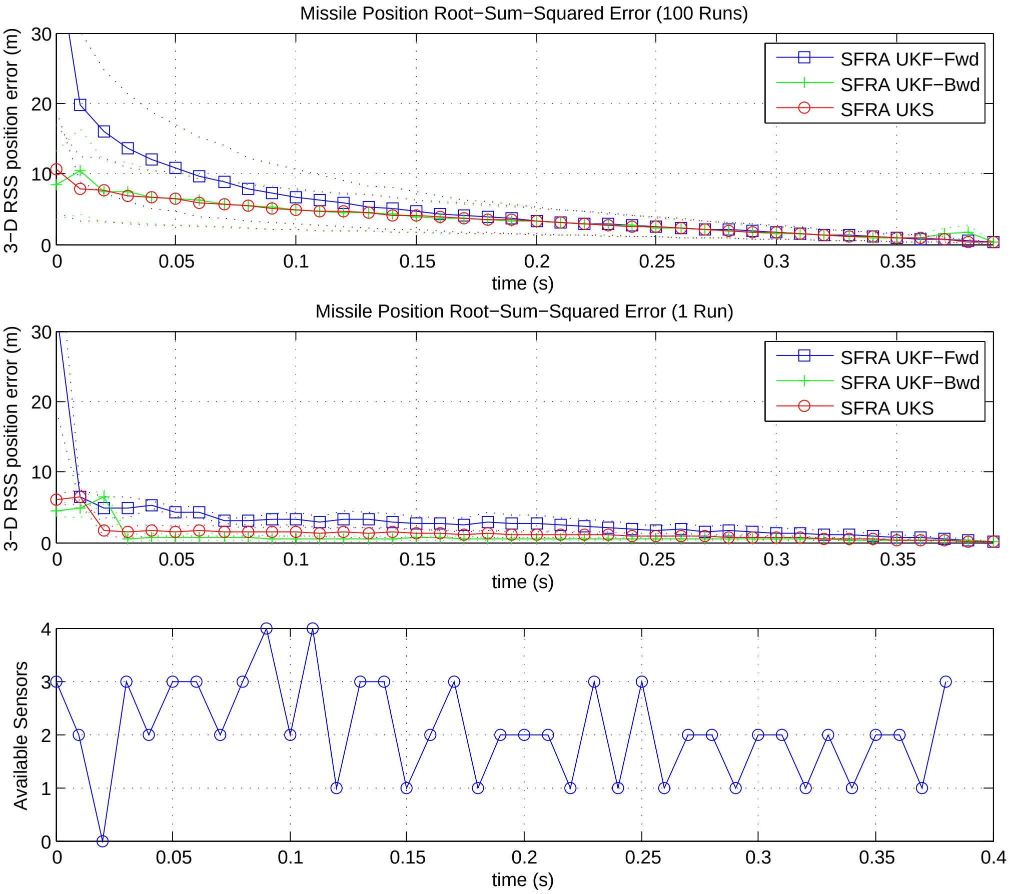
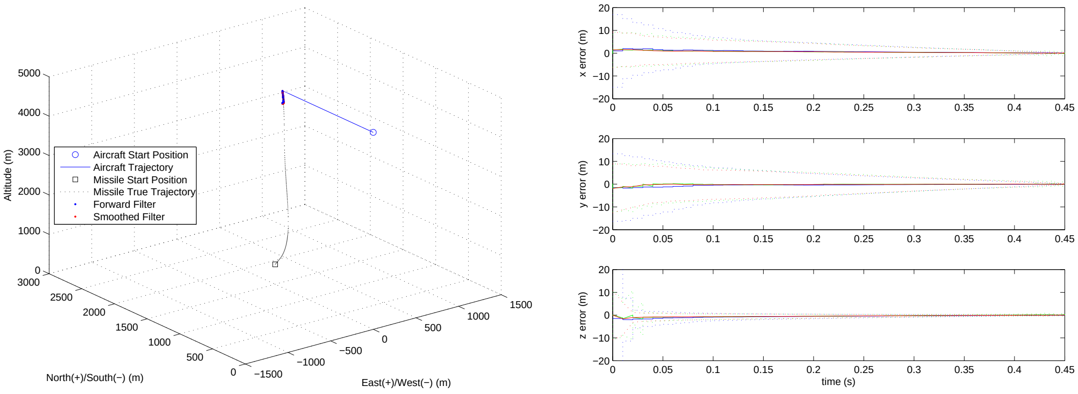
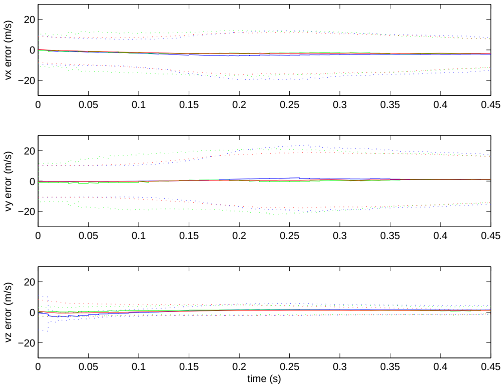


Figure A.114: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Continuous Velocity Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.115: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 1)

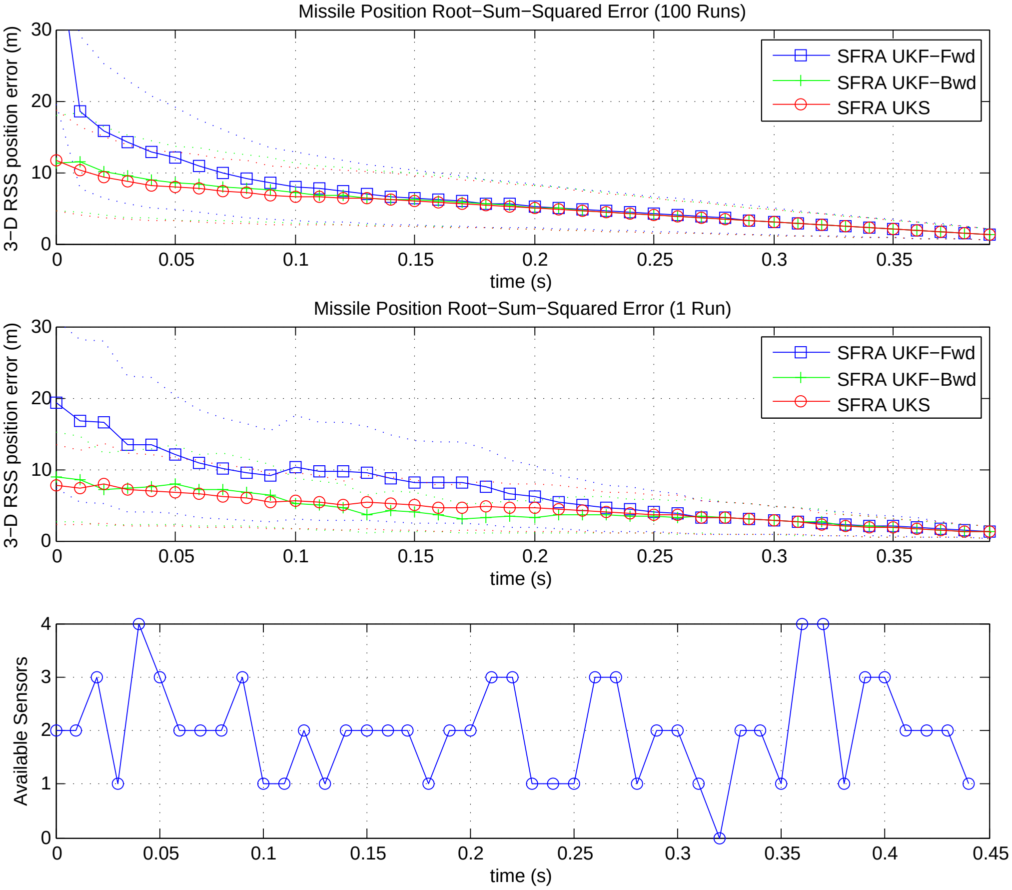
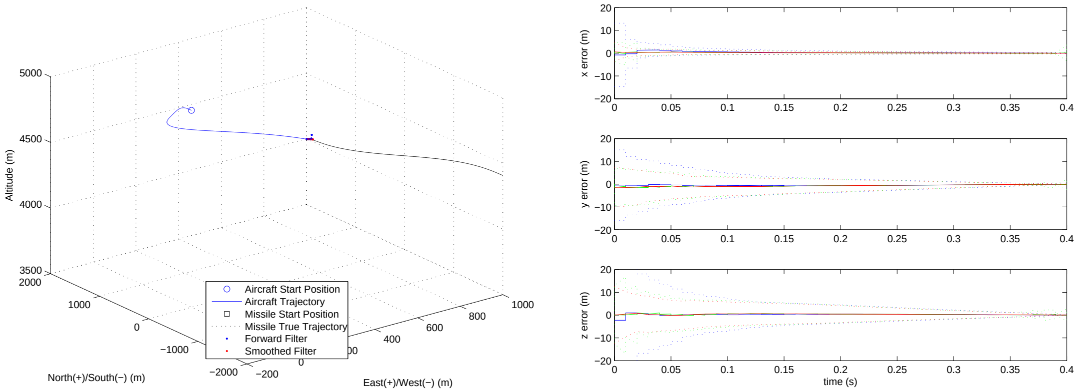
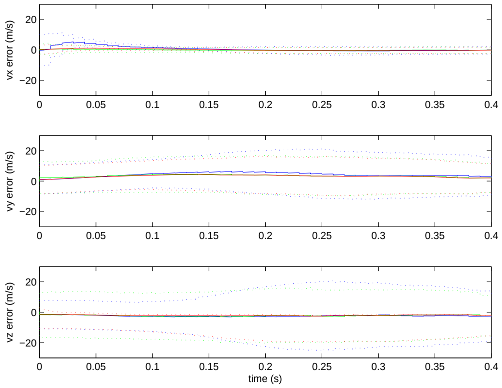


Figure A.116: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Constant Acceleration Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.117: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 2)

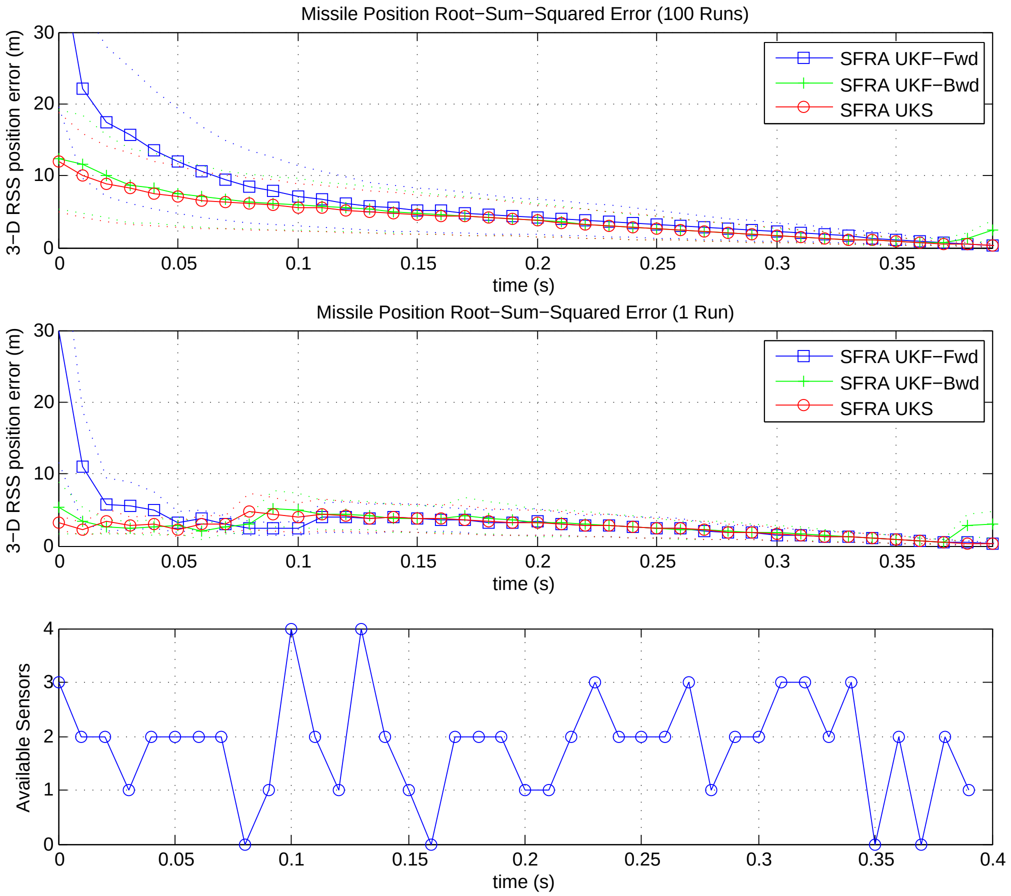
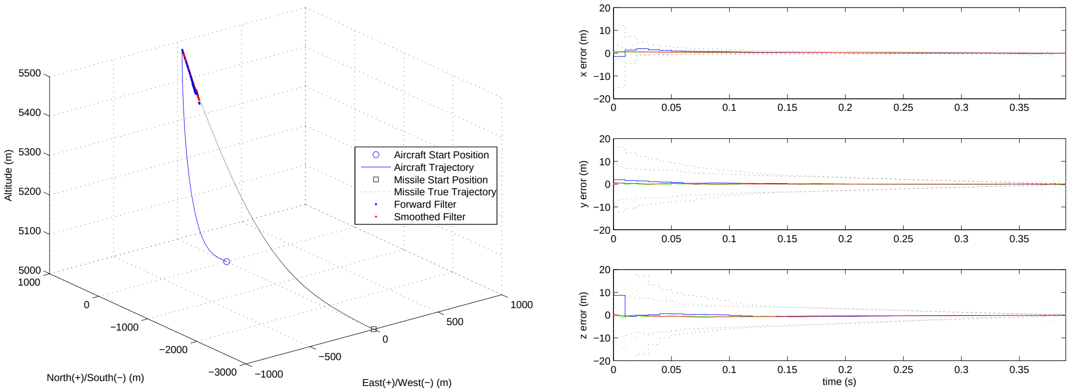
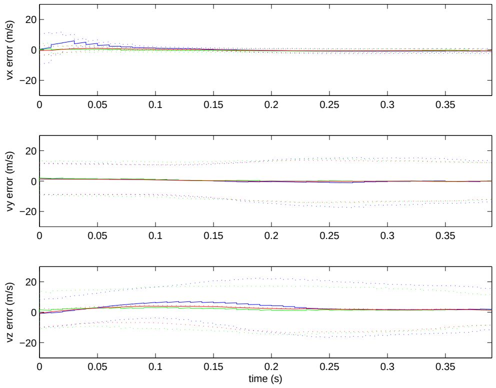


Figure A.118: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Constant Acceleration Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.119: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Constant Acceleration Dynamics Model (Scenario 3)

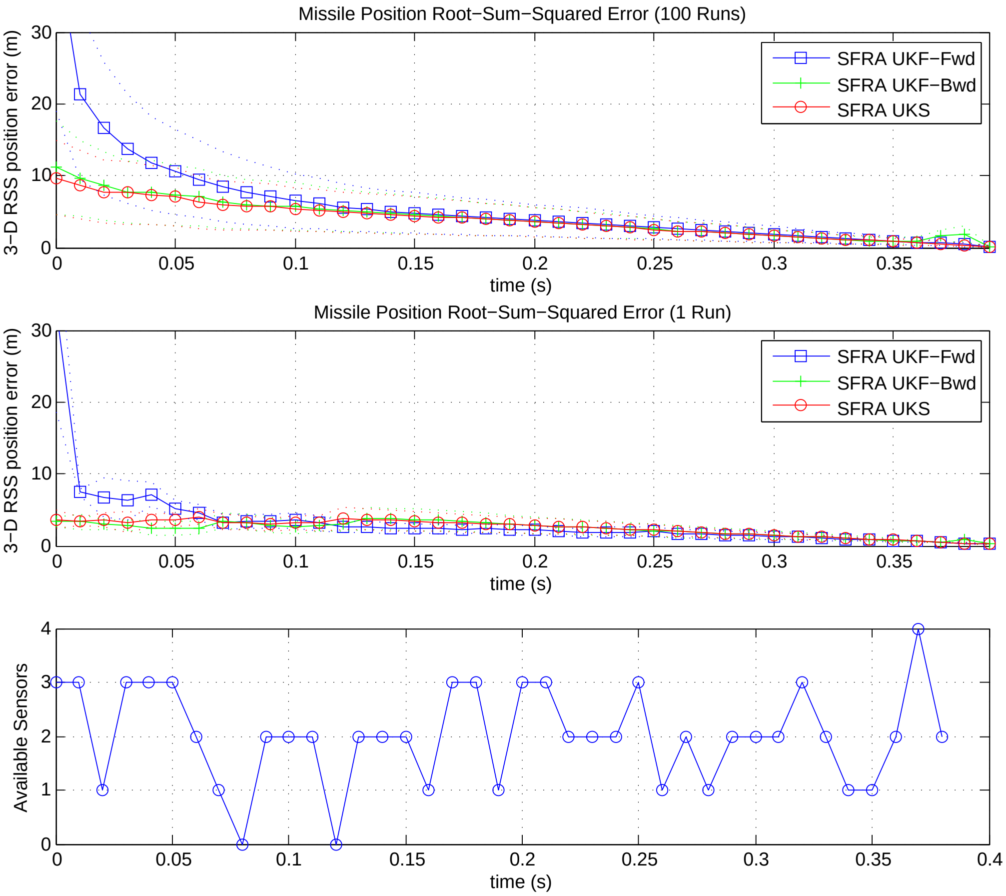
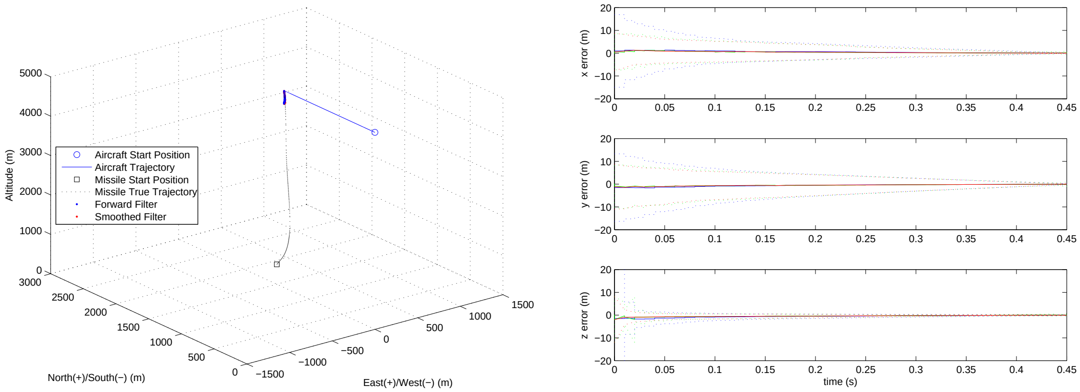
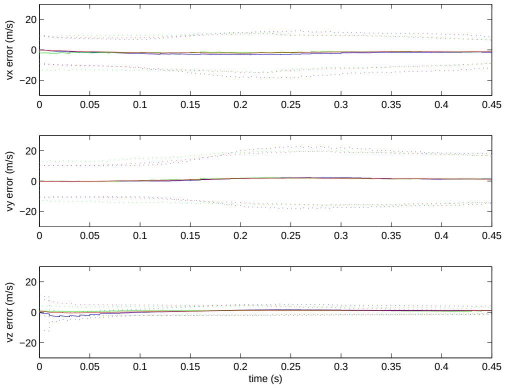


Figure A.120: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Constant Acceleration Dynamics Model (Scenario 3)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.121: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 1)

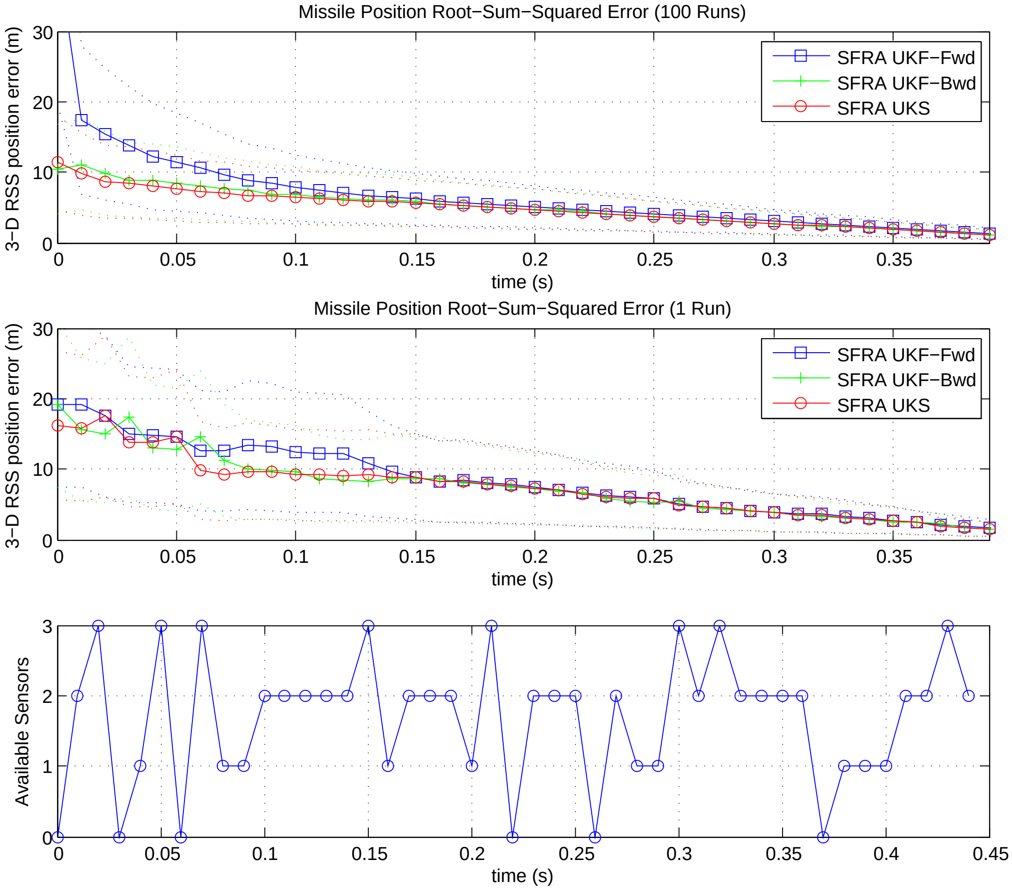
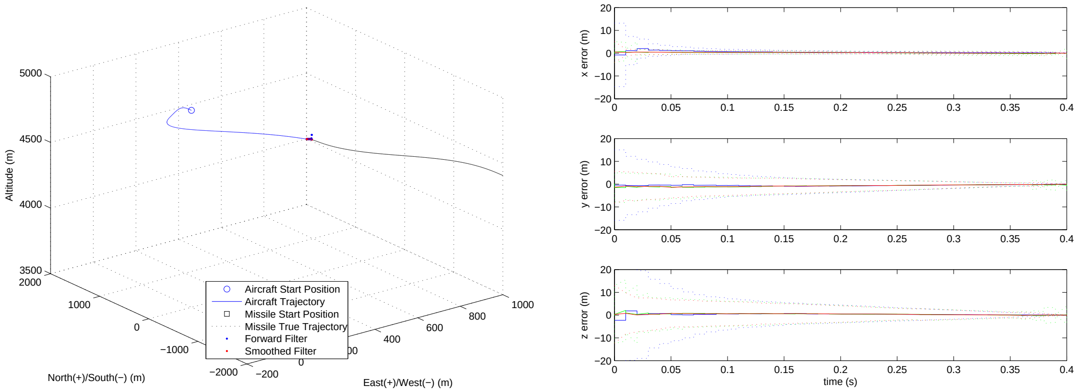
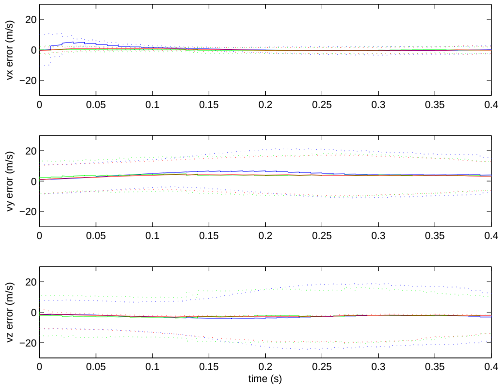


Figure A.122: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Coordinated Turn Dynamics Model (Scenario 1)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.123: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 2)

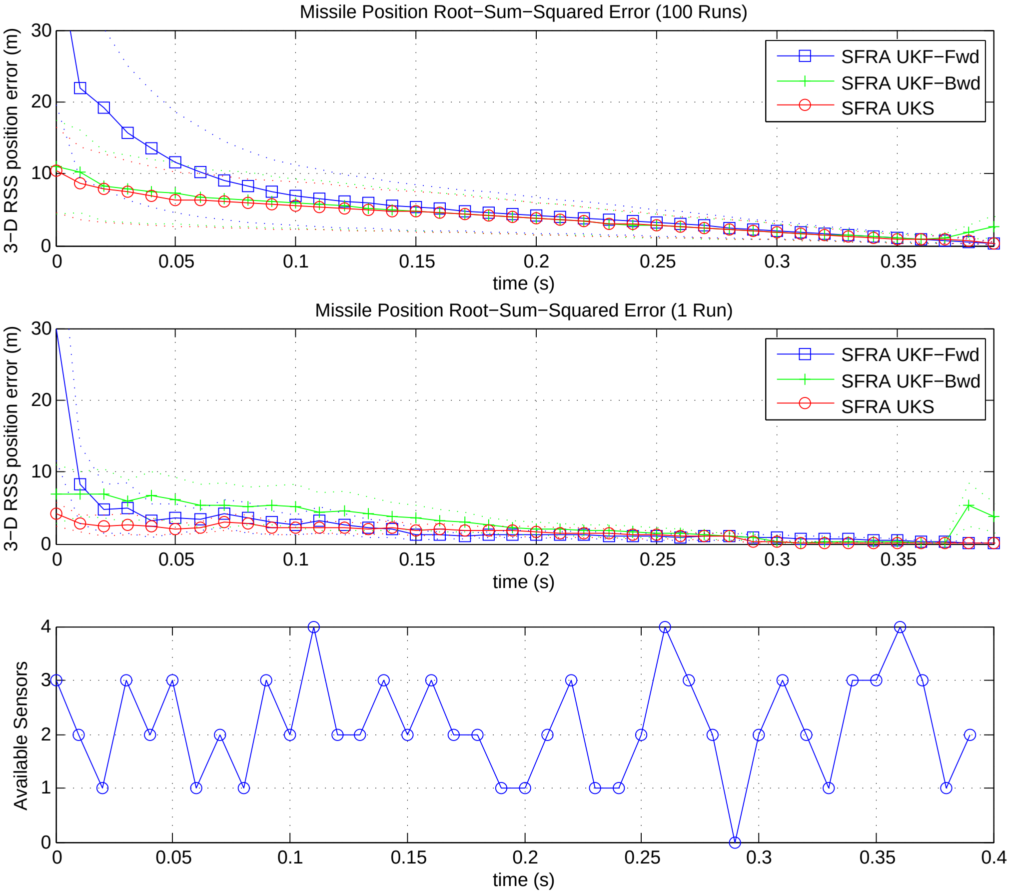
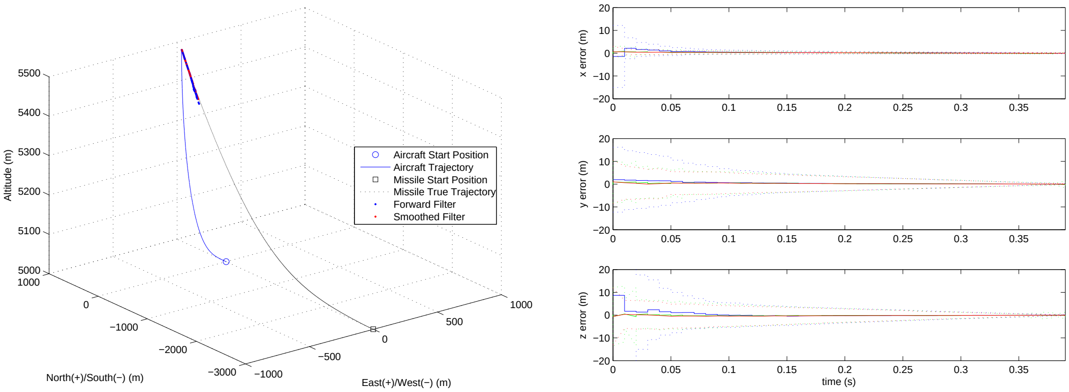
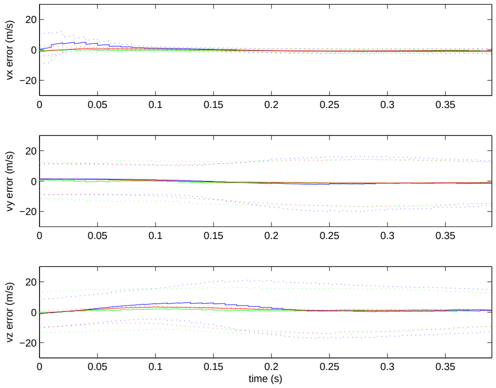


Figure A.124: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Coordinated Turn Dynamics Model (Scenario 2)



(a) 3D Aircraft and Missile Trajectory

(b) Mean Error and Error Standard Deviation of Missile Position States (100 Runs)



(c) Mean Error and Error Standard Deviation of Missile Velocity States (100 Runs)

Figure A.125: SFRA Unscented Kalman Smoother Random Sensor Dropout Performance in Air-to-Air Missile Scoring Application with Coordinated Turn Dynamics Model (Scenario 3)

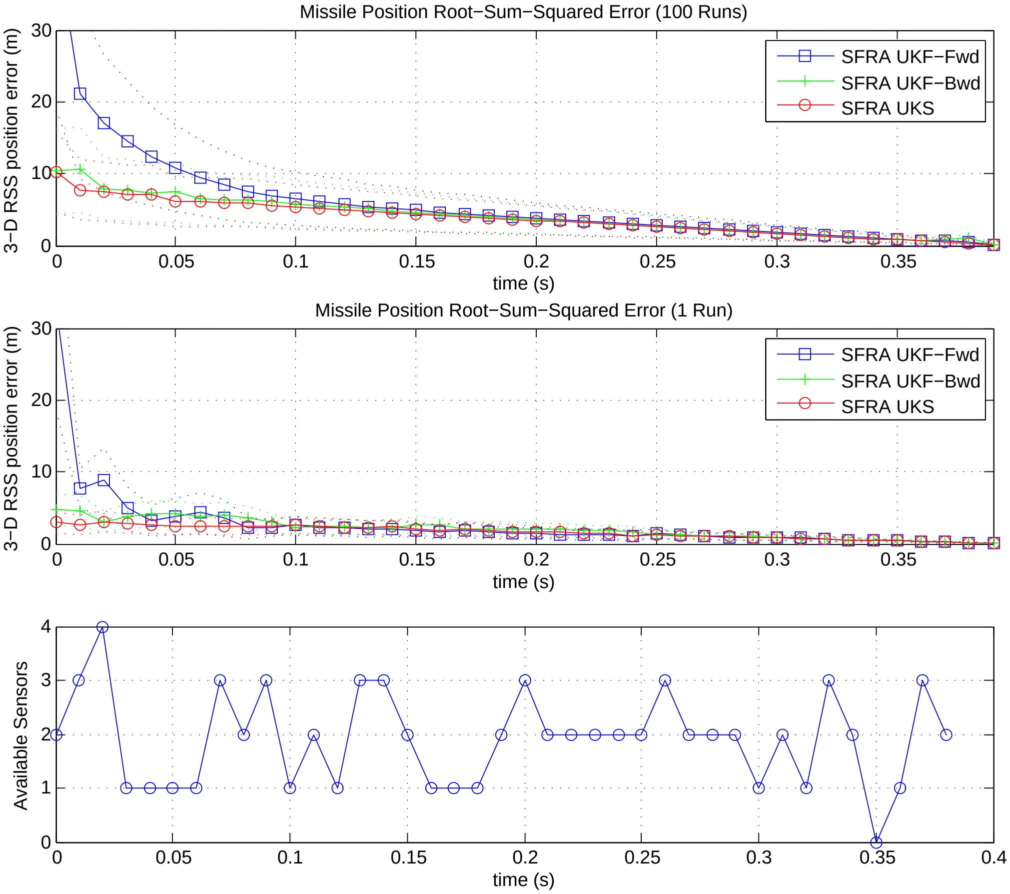


Figure A.126: SFRA Unscented Kalman Smoother Random Sensor Dropout Mean Root-Sum-Squared Error (100 Runs, 1 Run) in Missile Position Estimate and Sensor Availability with Coordinated Turn Dynamics Model (Scenario 3)

Bibliography

1. Barton, David A. *Paired Layering: The Argo Way of Simulation Construction.* National Air and Space Intelligence Center, December 2005.
2. Blackman, Samuel and Robert Popoli. *Modern Tracking Systems.* Artech House, 1999.
3. Brown, Robert G. and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons, 3rd edition, 1997.
4. Julier, Simon J. and Jeffrey K. Uhlmann. "A New Extension of the Kalman Filter to Nonlinear Systems". *The Proceedings of the American Control Conference*, Seattle, WA:1628–1632, 1995-1997.
5. Mahmood, Sultan, Jane Colee, Jim Luse, Dwight Payne, Joseph Taylor, and Samuel Burkett. "Gulf Range Target Control Using Rajpo GPS Equipment, Test Results and an Alternate Concept". *Position Location and Navigation Symposium*, 306–313. 1992.
6. Maybeck, Peter S. *Stochastic Models, Estimation and Control*, volume 1. Navtech Book and Software Store, 1994.
7. Maybeck, Peter S. *Stochastic Models, Estimation and Control*, volume 2. Navtech Book and Software Store, 1994.
8. Musick, Stanton H. *User's Guide for Profgen, A Trajectory Generator.* Air Force Research Laboratory, December 2004.
9. Pastorelli, A., G. Torricelli, M. Scabia, E. Biagi, and L. Masotti. "A Real-Time 2-D Vector Doppler System for Clinical Experimentation". *IEEE Transactions on Medical Imaging*, 27:1515–1524, 2008.
10. Roumeliotis, Stergios I. and George A. Bekey. "Smoother-based 3D Attitude Estimation for Mobile Robot Localiza- tion". *Department of Electrical Engineering*, Robotics Research Laboratories, University of Southern California. 1998.
11. Sweeney, Maj Nicholas. *Air-to-Air Missile Vector Scoring.* Master's thesis, Air Force Institute of Technology, 2012.
12. Sweeney, Nicholas and Kenneth Fisher. "Air-to-Air Missile Vector Scoring Using COTS Sensors". *JSDE/ION Joint Navigation Conf., Session A5: Missile/Projectile Applications.* 2011.
13. Teixeira, Bruno Otavio Soares, Leonardo Antonio Borges Torres, Paulo Henriques Iscold Andrade de Oliveira, and Luis Antonio Aguirre. "Flight Path Reconstruction Using the Unscented Kalman Filter Algorithm". *18th International Congress of Mechanical Engineering.* 2005.

14. Titterton, David H. and John L. Weston. *Strapdown Inertial Navigation Technology.* The Institution of Electrical Engineers, 2nd edition, 2004.
15. 53d Weapons Evaluation Group. "USAF Air-to-Air Weapon System Evaluation Program, Tyndall AFB, Fla." 2011.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*
22–03–2012

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From — To)*
Sept 2010 — Mar 2012

4. TITLE AND SUBTITLE
Air-to-Air Missile
Enhanced Scoring with Kalman Smoothing

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Gipson, Jonathon S., Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/12-18

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to Copyright protection in the United States.

14. ABSTRACT

Estimating the trajectory of an air-to-air missile provides many unique challenges. This is complicated by the high velocities and extremely high turn rates attained by the missile during its short Time-of-Flight (TOF) en route to the target. Kalman smoothers lend themselves to tasks such as post-flight trajectory estimation because they combine the utility of forward and backward-propagating Kalman filters. The combined result is maximum accuracy for post-flight missile scoring. Six Kalman smoothers (EKS, IEKS, SFRA EKS, UKS, IUKS, and SFRA UKS) are simulated. The performance assessment is based on multiple Monte Carlo comparisons among all algorithms with a variety of missile models and air-to-air engagment scenarios. This technical assessment provides the basis for recommendation of the Unscented Kalman Smoother (UKS) as the DoD/USAF standard for post-processing and scoring live-fire missile data.

15. SUBJECT TERMS
Kalman Smoother, UKS, EKS, IEKF, IEKS, UKF, EKF, Trajectory Reconstruction, Missile Scoring

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Maj Kenneth Fisher
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	UU	239	(937) 255–3636, ext 4677; kenneth.fisher@afit.edu