MANEUVERING INTRUDER PASSIVE RANGING FOR DETECT-AND-AVOID

By
JORDAN DAUGHERTY
Bachelor of Science in Mechanical Engineering
Oklahoma State University
Stillwater, Oklahoma, USA
2014

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2018

MANEUVERING INTRUDER PASSIVE RANGING FOR DETECT-AND-AVOID

Thesis Approved:

Dr. He Bai

Thesis Advisor

Dr. Jamey Jacob

Committee Member

Dr. Rushikesh Kamalapurkar

Committee Member

Name: Jordan Daugherty

Date of Degree: May, 2018

Title of Study: MANEUVERING INTRUDER PASSIVE RANGING FOR DETECT-AND-AVOID

Major Field: MECHANICAL ENGINEERING

Abstract: For the last two decades, unmanned aircraft systems (UAS) has seen much interest from both the civilian and military sector. As civilian applications expand, the issue of safety becomes more apparent. One major technical challenge currently facing UAS operations is properly sharing the national airspace with conventional aircraft. For safety purposes, it is necessary that UAS be able to properly detect intruding aircraft, including manned and unmanned aircraft, and avoid them. This requirement has been termed Detect-and-Avoid (DAA). We investigate the orbiting intruder passive ranging problem, where an ownship aircraft is moving with a constant velocity and the intruding aircraft is conducting an orbiting maneuver. We assume that the ownship measures the bearing angles to the intruder aircraft. We approach the problem utilizing a filter bank algorithm parameterized with respect to the range, the heading of the intruder, and the angular velocity. We test the performance of the filter bank algorithm using two different system models. The first system model comprises of the relative position in Cartesian coordinates and velocities in polar coordinates. The second system model is the modified polar coordinates. We conduct Monte Carlo simulations and utilize the root mean square error over time to determine the best parameterization of the filter algorithm for both system models. The results show that the system model in Cartesian coordinates performs better when estimating the range while the modified polar coordinates achieves better estimates for the heading of the intruder. We find that the filter in the modified polar coordinates exhibits more divergent behavior than the system in Cartesian coordinates. After an investigation of the orbiting intruder problem, we investigate the maneuvering intruder problem. Often the intruders trajectory will follow segments of straight legs and orbits legs. We introduce a way to integrate our filter bank algorithm onto a Interacting multiple models framework. We utilize a constant velocity model on a single EKF. We implement a mixing strategy, where the IMM mixing stage will mix at a certain rate. We conduct a simulation study to identify the effects of varying mixing rate and the values of the model transition probability. We find that the model transition probability has the largest effect on performance. Finally, we show preliminary results of our algorithms performance on flight test data.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

Introduction

1.1 Unmanned Aerial Systems (UAS)

Unmanned Aerial Systems (UAS) are defined as a system, whose components include the air vehicles and associated equipment that do not carry a human operator, but instead fly autonomously or are remotely piloted. UAS must be considered in a systems context which includes the command, control and communications (C3) system, and personnel necessary to control the unmanned aircraft [5]. The components of a UAS can include the unmanned aircraft, the ground control system, and the command and control link. The unmanned aircraft's flight can be controlled remotely by a pilot on the ground via a ground control station or autonomously controlled using a pre-planned flight plan and autopilot system. Unmanned aircraft platforms are equipped with a flight computer, sensors, and actuators for navigation and control purposes, and a payload that is mission dependent.

Classification of Unmanned Aircraft

Since the inception of UAS, many different types of platforms have been designed. These different designs are mission dependent, such as for surveillance and reconnaissance, or the delivery of a payload. Based on platform characteristics, unmanned aircraft can be categorized as either fixed-wing, rotary-wing, blimps, or flapping wings. UAS can be further classified by their weight or flight endurance, range or altitude according to Tables 1.1 and 1.2.

Table 1.1: Classification by Weight [1]

Category	Gross Weight (kg)
Super heavy	≥ 2000
Heavy	200-2000
Medium	50-200
Light	5-50
Micro	≤ 5

Table 1.2: Classification of UASs according to endurance, range, and altitude [1]

Category	Endurance (h)	Flying range (km)	Altitude(km)
Long/High	$\geq$24	$\geq$400	$\geq$10
Medium	5-24	100-400	1-10
Low	$\leq$5	$\leq$100	$\leq$1

1.2 Integration into the National Airspace (NAS)

For the last two decades, unmanned aircraft systems (UAS) has seen much interest from both the civilian and military sector. This can be attributed to the utility that UAS have over conventional aircraft due to platform size and operating cost. Furthermore, UAS provides a solution to tasks coined by a term called the three D's: dirty, dangerous, and dull. Tasks that may cause inherent risks to a pilot operating a conventional aircraft can be avoided by a pilot remotely operating a UAS. Additionally, missions with long operation requirements can be conducted by multiple pilots from ground control stations. Some examples of how UAS have opened up opportunities in the civilian sector include surveying farmland, precision agriculture, monitoring natural disasters, and transporting goods[5].

As civilian applications expand, the issue of safety becomes more apparent. One major technical challenge currently facing UAS operations is properly sharing the national airspace with conventional aircraft [6]. Due to the varying sizes of UAS platforms, pilots in conventional aircraft can have issues identifying small UAS (sUAS). For safety purposes, it is necessary that UAS be able to properly detect intruding aircraft, including manned and unmanned aircraft, and avoid them. This requirement has been termed Detect-and-Avoid (DAA).

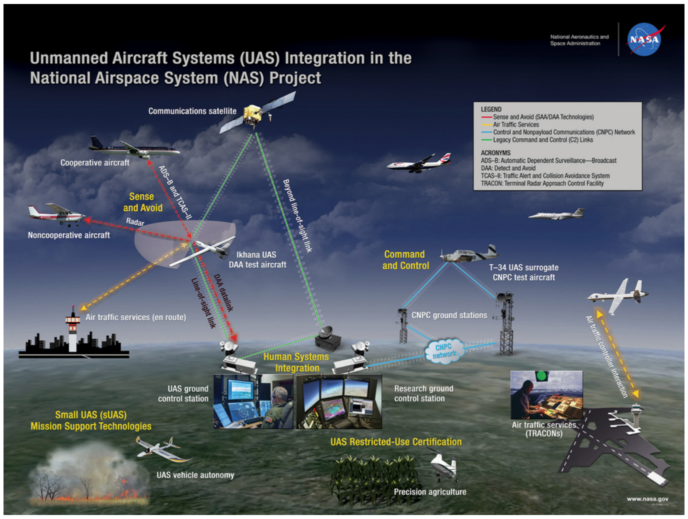

National Aeronautics and Space Administration
NASA
Unmanned Aircraft Systems (UAS) Integration in the National Airspace System (NAS) Project
LEGEND
Sense and Avoid (SAA/DAA Technologies)
Air Traffic Services
Control and Nonpayload Communications (CNPC) Network
Legacy Command and Control (C2) Links
ACRONYMS
ADS–B: Automatic Dependent Surveillance—Broadcast
DAA: Detect and Avoid
TCAS–II: Traffic Alert and Collision Avoidance System
TRACON: Terminal Radar Approach Control Facility
Communications satellite
Cooperative aircraft
ADS–B and TCAS–II
Sense and Avoid
Radar
Noncooperative aircraft
Ikhana UAS DAA test aircraft
Beyond line-of-sight link
Line-of-sight link
DAA datalink
Command and Control
T–34 UAS surrogate CNPC test aircraft
CNPC ground stations
CNPC network
Air traffic controller interaction
Air traffic services (en route)
Human Systems Integration
UAS ground control station
Research ground control station
Air traffic services (TRACONs)
Small UAS (sUAS) Mission Support Technologies
UAS vehicle autonomy
UAS Restricted-Use Certification
Precision agriculture
www.nasa.gov

Figure 1.1: An illustration describing integration of UAS into the national airspace from [2]

1.3 Detect-and-Avoid (DAA)

The detect and avoid model can be described as a continuous feedback loop composed of the following four processes: sensing hardware, decision mechanism, path planner, and flight controller. The sensing hardware gathers information on the ownship and intruder vehicles throughout the flight. A decision mechanism determines whether the current flight path is still satisfactory or needs to be adjusted. In the case the path must be adjusted, the path planner will determine a trajectory that the ownship vehicle can follow to avoid collision with the intruder vehicle. The flight controller recieves the path planner's adjustments and actuates the platform to the desired trajectory. Figure 1.2 illustrates this continuous feedback loop process.

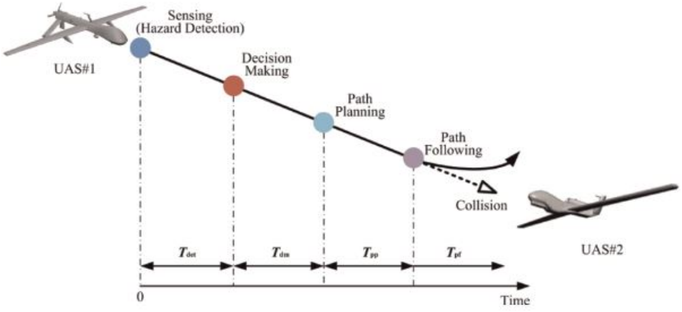
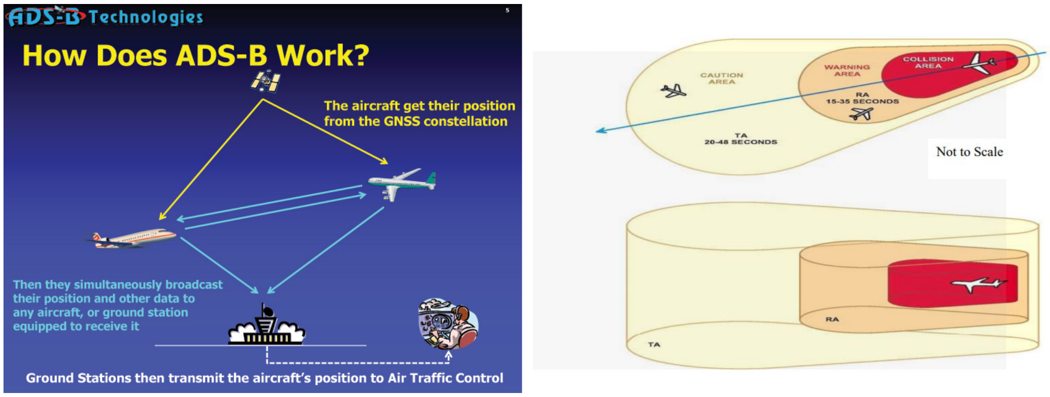


Figure 1.2: An illustration of the DAA model from [1]

Detect and avoid can be broadly categorized into two different approaches: Cooperative and Non-Cooperative. Cooperative approaches utilizes two-way communication between aircraft and ground control stations(GCS). Non cooperative approaches assume no communication between aircraft or GCS.

Two examples of cooperative approaches include the Automatic Dependent Surveillance-Broadcast (ADS-B) and the Traffic Collision Avoidance System (TCAS)[7, 8, 9]. ADS-B are used by an aircraft to broadcast its position and velocity information determined by onboard instruments to other aircraft, while also receiving other aircrafts position and velocity information. Potential hazards for the ADS-B can include GPS receiver or signal transmission malfunctions [10]. TCAS monitor the airspace around the ownship aircraft utilizing an active transponder to warn of potential collisions with other aircraft equipped with TCAS. Currently issues regarding the usage of TCAS on UAS are primarily due to the cost and payload size [9]. Figure 1.3 provides illustrations of the two cooperative approaches, ADS-B and TCAS. One of the primary weaknesses of cooperative approaches is that they are ineffective in the tracking ground-based obstacles in an urban environment. It is also expected that some manned aircraft may not be equipped with ADS-B or TCAS transponders.


ADS-B Technologies
How Does ADS-B Work?
The aircraft get their position from the GNSS constellation
Then they simultaneously broadcast their position and other data to any aircraft, or ground station equipped to receive it
Ground Stations then transmit the aircraft's position to Air Traffic Control
CAUTION AREA
WARNING AREA
COLLISION AREA
RA 15-35 SECONDS
TA 20-48 SECONDS
Not to Scale
RA
TA

(a) (b)

Figure 1.3: An illustration of two prominent Cooperative Approaches being used, (a) the Automatic Dependent Surveillance-Broadcast (ADS-B) from [2] and (b) the Traffic Collision Avoidance System (TCAS) from [7]

Noncooperative approaches cover this weakness.

Noncooperative approaches currently under wide consideration are systems that utilize a combination of passive and active Forward Looking Sensors (FLS) [11]. Active sensors can include Synthetic Aperture Radar (SAR) and Laser/Light Detection and Ranging (LIDAR). Passive sensors include electro-optical (EO), infrared(IR), and acoustic systems. Active sensors such as SAR and LIDAR are able to obtain range information on objects, however, they are typically expensive and the requirements may be costly in terms of size, weight, and power (SWAP). Passive sensors, such as EO and IR systems, provide an alternative to active sensors. Passive sensors lack direct range measurements, but are able to obtain azimuth and elevation angles of objects. Figure 1.4 provides images of various sensors used for noncooperative approaches.

1.4 Summary of Paper

In this paper, we address range estimation using passive sensors, which is typically called 'Passive Ranging.' Transponders from cooperative approaches, such as TCAS and ADS-B, provide excellent results, however can be expensive in terms of SWAP.

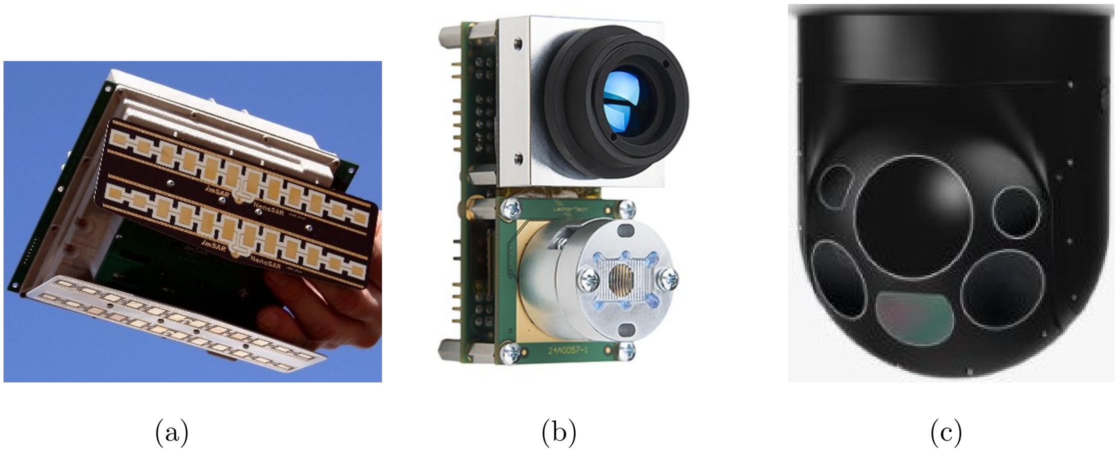


Figure 1.4: An illustration of sensors used in Noncooperative approaches: (a) the Synthetic Aperture Radar (SAR) from [12],(b) the Laser/light Detection and Ranging (LIDAR) from [13], and (c) the Electro-Optical (EO) sensor from [14]

Additionally, the transponders provide no purpose if the intruding aircraft is not equipped with a transponder as well. Active sensors from noncooperative approaches, such as SAR and LIDAR provide range information of targets, but are also expensive in terms of SWAP. Depending on the UAS platform, the power requirements for implementation of active sensors may be high. Passive sensors, such as EO and IR sensors, are cheap in terms of SWAP.

Most of the passive ranging references that we have found address the scenario where the intruder aircraft moves with a constant velocity. In this paper, we focus on the orbiting intruder passive ranging problem, where an ownship UAS is moving with a constant velocity and the intruder aircraft is orbiting with a constant angular velocity and speed. The orbiting intruder problem is of interest when a UAS encounters a fixed-wing aircraft (manned or unmanned) conducts a mission in a bounded region, such as tracking a ground target and monitoring a field. In this case, the fixed-wing aircraft exhibits an orbiting pattern.

We approach the orbiting intruder ranging problem using a filter bank algorithm motivated by the RPEKF [15]. Our filter bank algorithm differs from the RPEKF because in addition to the range, we also parameterize the heading and angular velocity states. We utilize the filter bank algorithm for two different system models, the Cartesian Position, Polar Velocity (CPPV) and the Modified Polar Coordinates (MPC) [16, 17]. In the CPPV model, we represent the relative position between the two aircraft and the velocity of the intruder aircraft in Cartesian and polar coordinates, respectively.

To examine the performance of the filter bank algorithm, we conduct Monte-Carlo simulations to determine the best parameterization of the filter in the CPPV model. The simulation results show that for the CPPV model, the most important state to parameterize is the range, followed by the heading, and then the angular velocity of the intruder. For the filter in the MPC model, we determine that the most important state to parameterize is the range, followed by the angular velocity, and then heading. We also report a comparison of the performance of the filters in the CPPV and MPC models, which to our knowledge is missing in the literature for the orbiting intruder passive ranging problem. Our results show that the CPPV model achieves better estimates for the range while the MPC achieves better estimates for the heading of the intruder. The filter in the MPC results in more divergence cases than in the CPPV model. In addition, we test the robustness of the filter with respect to ownship velocity noise (due to GPS/inertial measurement unit (IMU) fusion errors) and bearing angle biases (due to inaccuracies in the sensor calibration).

After examining the filter bank algorithm's performance for the orbiting intruder problem, we then turn to look at the Interacting Multiple Models (IMM) algorithm to address the maneuvering intruder ranging problem. For the IMM we utilize two different models to track the maneuvering intruder: a constant velocity (CV) model, and a coordinated turn (CT) model. We utilize our filter bank algorithm for the CT model and a single EKF for the CV model, both in CPPV coordinates. We conduct Monte-Carlo simulations to identify the importance and the effects of the frequency of the IMM mixing stage and the values for the model transition probability matrix. We find that lowering δ, the off-diagonal values of the model transition probability matrix, has more of an effect on performance than the lowering the frequency of the IMM mixing stage.

1.5 Organization

This paper is organized as follows. In Chapter 2, we conduct a literature review over relevent knowledge to ensure the reader has a common ground to understand the content. We cover the the Bearing Only tracking problem, the extended Kalman filter, the Range-Parameterized extended Kalman filter, and the Interacting Multiple Models. In Chapter 3 we formulate the orbiting intruder problem, propose our filter bank algorithm, show how we implement the algorithm in CPPV and MPC coordinates, and provide an Interacting Multiple Models framework for the filter bank algorithm. Chapter 4, the simulation studies, can be largely split into two parts. First, we investigate the orbiting intruder problem, where we utilize the filter bank algorithm. We go into different studies on the filter's performance, such as investigating how to parameterize the filter bank, what are the effects of the coordinate system, and their robustness with respect to velocity noise and measurement bias. The latter half of Chapter 4 covers the maneuvering intruder problem. Utilizing the IMM algorithm, we go on to investigate the effects of varying the mixing rate of the different model's filters and the values of the model transition probability matrix. After this we look at the IMM's performance on different maneuvering intruder encounters. In Chapter 5, we describe our collaboration with UtopiaCompression, and the results of our filter on the real flight test data we were given for a better understanding of performance on real data sets. Conclusions and future work are discussed in Chapter 6. An appendix section is included at the end of this thesis for additional calculations.

CHAPTER 2

Literature Review

2.1 Bearing Only Tracking

The Bearings-Only Tracking (BOT) problem is the estimation of a target's position of unknown kinematic parameters, such as position, speed, and heading, of a target by utilizing noise corrupted bearing angle measurements collected by a moving ownship over time [18]. The BOT problem has been of interest to areas such as sonar in marine applications, or radar in aerial applications.

The difficulty of this problem can be attributed to the nonlinearity of the system and/or the measurement equation. In addition, the lack of range information in the measurement equation leads to a partially observable system. It is well known that if an ownship aircraft and an intruder aircraft both move with a constant velocity, the ownship platform must maneuver and generate higher-order motion to yield state observability of the intruding platform [19, 18].

Typical estimators for solving passive ranging include the extended Kalman Filter (EKF) [20]. When implementing an EKF, it can easily diverge without a good initialization [21]. One classic approach for mitigating the issue of divergence is the utilization of a bank of EKFs, where each filter is initialized with a different initial estimate for the range [22]. It is commonly referred to as the range-parameterized EKF (RPEKF) in passive ranging. Another estimator for the passive ranging problem is particle filter [23]. In recent years, the particle flow filter has been proposed for passive ranging applications, which utilizes the homotopy function to propagate the particles [24, 25].

Other than changing the filter algorithm, research has also been conducted by utilzing different coordinate systems. Two popular system models that are used for passive ranging include Cartesian coordinates and polar coordinates. Studies have been conducted to test for the best model [17, 16]. In [16], the author introduced the Modified Polar Coordinates (MPC). It was claimed that the MPC exhibit good performance because it decouples the observable and nonobservable components of the state vector, and reduces covariance matrix ill-conditioning. In [17], two models to estimate the motion for an orbiting intruder are described, one in Cartesian coordinates for both position and velocity, and the other in Cartesian coordinates for position and polar coordinates for the velocity. The latter model was proved to exhibit better performance.

Often times, during passive ranging, the intruder aircraft will not follow just a single mode of maneuver. With this in mind, research has been conducted to ensure adequate performance for estimation of an intruding aircraft in different modes of maneuvering. One method that has seen use is the Interacting Multiple Models (IMM) algorithm. The standard IMM algorithm is composed of a filter for each model, a model probability evaluator, a state estimate mixer at the input of the filters, and a state estimate combiner at the output of the filters [3].

2.2 Range-Parameterized Extended Kalman Filter

The Range-Parameterized Extended Kalman Filter (RPEKF) is a bank of Extended Kalman Filters (EKF) that are parametrized geometrically with respect to user-defined range assumptions [22, 15]. Suppose that we are given an upper bound and lower bound of the initial range of the intruder $[r_{min}, r_{max}]$. We can divide this interval into N_r subintervals. The subintervals are distributed geometrically according to:

$$r^{(i)} = \frac{r_{min}}{2}\left(\rho^{i} + \rho^{i-1}\right), \rho = \left(\frac{r_{max}}{r_{min}}\right)^{\frac{1}{N_r}} \tag{2.1}$$

$$\sigma_r^{(i)} = C_r r^{(i)}, C_R = \frac{2(\rho - 1)}{\sqrt{12}(\rho + 1)}, \tag{2.2}$$

where $i = 1, \cdots, N_r$ represents the index of a specific subinterval, $r^{(i)}$ is the ith range estimate in the interval, $\sigma_r^{(i)}$ is the ith standard deviation for range, and C_R is the coefficient of variation. For each EKF, all states other than the range within the estimates and covariances are constant for every filter in the filter bank when initialized.

Each filter is associated with a weight, $W_k^{(i)}$, at time step k. The weights are initialized uniformly and summed to one. The weight of each EKF in the bank is updated using Bayes rule:

$$W_k^{(i)} = cW_{k-1}^{(i)} p(z_k|i) \tag{2.3}$$

$$\sum\nolimits_{i=1}^{N_r} W_k^{(i)} = 1, \tag{2.4}$$

where c is a normalizing constant, and $p(z_k|i)$, the likelihood for the ith EKF, is given by:

$$p(z_k|i) \propto \frac{1}{\sqrt{det(S_k^{(i)})}} exp\left(-\frac{1}{2}(z_k - h(\hat{x}_{k|k-1}^{(i)}))^T {S_k^{(i)}}^{-1}(z_k - h(\hat{x}_{k|k-1}^{(i)}))\right). \tag{2.5}$$

The combined estimate and covariance are calculated as:

$$\hat{x}_{k|k} = \sum\nolimits_{i=1}^{N_t} W_k^{(i)} \hat{x}_{k|k}^{(i)} \tag{2.6}$$

$$P_{k|k} = \sum\nolimits_{i=1}^{N_t} W_k^{(i)} (P_{k|k}^{(i)} + (\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k})(\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k})^T). \tag{2.7}$$

2.2.1 The Extended Kalman Filter

The Kalman filter is a recursive estimation algorithm that utilizes a series of measurements and inputs, each containing their own respective statistical noises, to produce an estimate of unknown state variables. The Kalman filter assumes that the dynamic and measurement equations, f_{k-1} and h_k are linear, perturbed by white zero-mean gaussian noises, v_k, and w_k, with known covariance matrices Q_{k-1} and R_k[26, 27]. This assumption, however, becomes an issue when either model is not represented by a linear function. The extended Kalman filter(EKF) was the solution to this problem, linearizing both the system and observation models about an operating point.

Let us suppose that the true state at time step $k-1$, x_{k-1}, can be mapped to the state at time step k, x_k given the input u_k, and process noise v_k given the dynamic function f_{k-1}, and that the current sensor measurement, z_k, is observed by the measurement function h_k:

$$\begin{aligned} x_k &= f(x_{k-1}, u_k) + v_k \\ z_k &= h(x_k) + w_k \qquad . \end{aligned}$$

The extended Kalman filter can be broken into two stages of calculations, the prediction stage and the update stage. The equations for each stage are as follows

Prediction Stage

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \tag{2.8}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{2.9}$$

Update Stage

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}) \tag{2.10}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{2.11}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{2.12}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{2.13}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{2.14}$$

Where the dynamic and observation matrices, F_k and H_k, are the Jacobians:

$$F_k = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1|k-1},u_k} \tag{2.15}$$

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k-1|k-1}} \quad , \tag{2.16}$$

which are linearized about the prior estimate at time step $k-1$, $\hat{x}_{k-1|k-1}$.

Let us suppose that the filter is given a well-conditioned state estimate. Over time, the measured values from the sensor will act as a feedback that modify the prediction of the filter's dynamics model. This modification is conducted by the Kalman Gain, K_k and the innovation, $\tilde{y}_k$. If the filter's dynamics model are accurate, then the innovation, which is the residual of the observation of the filter and the measurement of the sensor, will be low and the the Kalman gain will play little effect on the filter's estimate. However, when the innovation becomes large, this can mean two things: either the dynamics model is inaccurate, or the measurement is too noisy.

2.3 Interacting Multiple Models

Often times, the use of one dynamics model may not be adequate enough to describe the motion of a maneuvering intruder. For this reason, the Interacting Multiple models algorithm has become one of the de facto solutions for tracking applications [3]. Figure 2.1 depicts the four steps of the Interacting Multiple Model (IMM) algorithm: Mixing state estimates, model updates, model probabilities update, and combining state estimates.

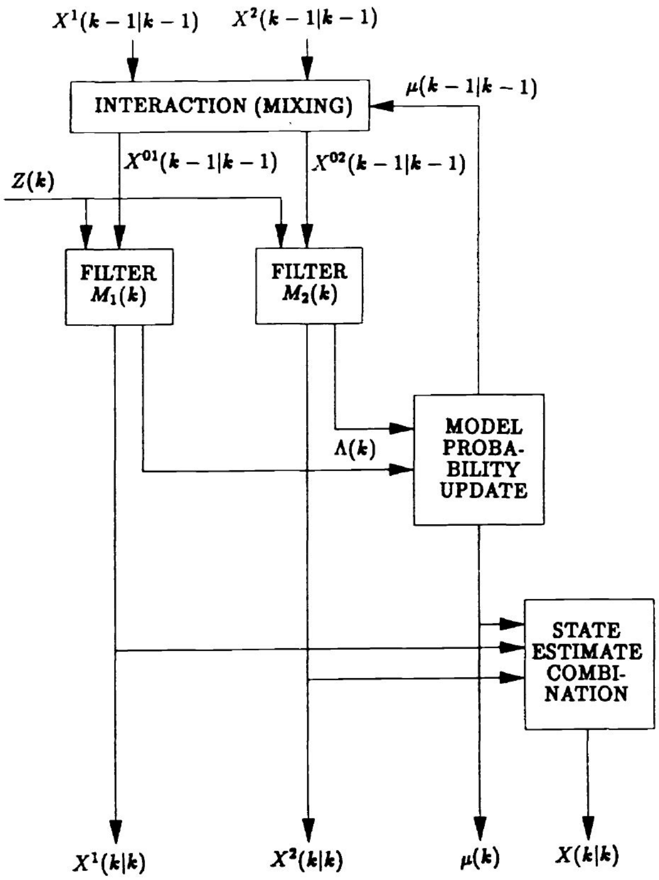


Figure 2.1: A Diagram that describes how the IMM algorithm works from [3]

Suppose a linear system can be represented with Markovian switching coefficients as

$$X_{k+1} = F_k(\theta_k)X_k + G_k(\theta_k)w_k \tag{2.17}$$

$$Y_k = H_k(\theta_k)X_k + v_k \tag{2.18}$$

where X_k is the system state, θ_k is a finite state Markov chain with values in $1, \dots, N$ according to the probability transition p_{ij}, that transitions model i to model j.

At time step k, during the mixing stage, model i mixes its prior state estimate with the prior state estimate at time step $k-1$ of model j according their respective model probabilities to produce a mixed state estimate $X^{0j}_{k-1|k-1}$. After this mixing stage, each model's respective filter conducts a filtering cycle of prediction and update stage. The model likelihood for each model is then calculated and used to update the model probabilities. The mixed state estimate output can from the weighted summation of each model's filter estimates.

Mixing of State Estimates

Relevant terms:

- $X^j_{k-1|k-1}$ a priori state estimate
- $P^j_{k-1|k-1}$ a priori error covariance
- $\mu_{k-1}(j)$ associated model probability
- $M_k(j)$ model j at time k
- p_{ij} model transition probability for switching from model i to model j
- $\bar{c}_j$ normalization constant

The mixed state estimate for $M_k(j)$ is computed as

$$X^{0j}_{k-1|k-1} = \sum X^i_{k-1|k-1}\mu_{k-1|k-1}(i|j), \tag{2.19}$$

and

$$\mu_{k-1|k-1}(i|j) = \frac{1}{\bar{c}_j} p_{ij}\mu_{k-1}(i) \qquad \bar{c}_j = \sum_{i=1}^{N} p_{ij}\mu_{k-1}(i). \tag{2.20}$$

The mixed state covariance for $M_k(j)$ is computed as

$$P^{0j}_{k-1|k-1} = \sum_{i=1}^{N} [P^i_{k-1|k-1} + \left(X^i_{k-1|k-1} - X^{0j}_{k-1|k-1}\right)\left(X^i_{k-1|k-1} - X^{0j}_{k-1|k-1}\right)^T]\mu_{k-1|k-1}(i|j) \tag{2.21}$$

Update Models

Each model's filter is updated using the standard Kalman Filter update equations 2.10-2.14.

Model Likelihood

Likelihood of $M_k(j)$ is computed with filter residuals $\tilde{Z}_k^j$, the filter residuals S_k^j, and gaussian assumptions. The likelihood of $M_k(j)$ given by

$$\Lambda_k^j = \frac{1}{\sqrt{|2\pi S_k^j|}} exp\big[-0.5(\tilde{Z}_k^j)^T(S_k^j)\tilde{Z}_k^j\big] \tag{2.22}$$

Model Probability Update

The model probabilities are updated as

$$\mu_k(j) = \frac{1}{c}\Lambda_k(j)\bar{c}_j \qquad \text{where} \qquad c = \sum_{i=1}^{N} \Lambda_k(i)\bar{c}_i \tag{2.23}$$

State Estimate Combination

The state estimate and error covariances are found as

$$X_{k|k} = \sum_{i=1}^{N} X_{k|k}^i \mu_k(i) \quad P_{k|k} = \sum_{i=1}^{N} \mu_k(i)\big[P_{k|k}^i + (X_{k|k}^i - X_{k|k})(X_{k|k}^i - X_{k|k})^T\big] \tag{2.24}$$

CHAPTER 3

Proposed Algorithms

3.1 Problem Formulation

For the orbiting intruder ranging problem, an ownship vehicle moves with a constant velocity and tracks an intruder vehicle that orbits with a constant angular velocity and a constant linear speed. We let (x_o, y_o) be the position of the ownship in the Cartesian coordinates. Similarly, let (v_{ox}, v_{oy}) be its velocity in the Cartesian coordinates. The ownship's motion model is given by the constant velocity model:

$$\begin{bmatrix} \dot{x_o} \\ \dot{v_{ox}} \\ \dot{y_o} \\ \dot{v_{oy}} \end{bmatrix} = \begin{bmatrix} v_{ox} \\ 0 \\ v_{oy} \\ 0 \end{bmatrix}. \tag{3.1}$$

For the intruder, we let (x_i, y_i) be its position in the x, y directions, v its linear speed, θ its heading, and ω its angular velocity. The intruder's motion model is given by:

$$\begin{bmatrix} \dot{x_i} \\ \dot{y_i} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} vcos(\theta) \\ vsin(\theta) \\ 0 \\ \omega \\ 0 \end{bmatrix}. \tag{3.2}$$

We assume that $\omega \neq 0$, which means that the intruder is orbiting. We further assume that the ownship measures the bearing (azimuth) angles of the intruder, β, defined as:

$$\beta = arctan\left(\frac{y_i - y_o}{x_i - x_o}\right) \cdot \tag{3.3}$$

Our objective is to employ the bearing measurements and the ownship information, such as v_{ox}, v_{oy}, to estimate the intruder's states. Towards this end, we define $r_x = x_i - x_o$ and $r_y = y_i - y_o$, and write the relative motion between the vehicles as:

$$\begin{bmatrix} \dot{r_x} \\ \dot{r_y} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} vcos(\theta) - v_{ox} \\ vsin(\theta) - v_{oy} \\ 0 \\ \omega \\ 0 \end{bmatrix} . \tag{3.4}$$

The states in (3.4) include the relative positions in Cartesian coordinates, polar velocity, and angular velocity. Therefore we call it a Cartesian Position Polar Velocity (CPPV) model. An illustration of the orbiting intruder problem is shown in Figure 3.1.

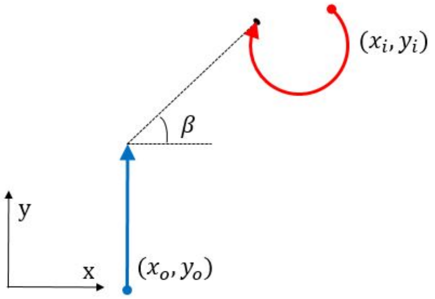


Figure 3.1: An illustration of the orbiting intruder problem, where the ownship (x_o, y_o) moves on a straight path while taking bearing angle measurements, β, of the intruding vehicle (x_i, y_i) over time.

We define the state vector X as:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \\ v \\ \theta \\ \omega \end{bmatrix}. \tag{3.5}$$

Because the measurements are taken at discrete time steps, we discretize (3.4) using the linearized discretization approach [17] and obtain the dynamics as:

$$x_k = \Phi_{k|k-1}(x_{k-1}) - U_k \tag{3.6}$$

where $\Phi_{k|k-1}$ is the state transition matrix given by

$$\Phi_{k|k-1}(x_{k-1}) = \begin{bmatrix} r_x + \frac{2v}{\omega} sin(\frac{\omega T}{2}) cos(\theta + \frac{\omega T}{2}) \\ r_y - \frac{2v}{\omega} sin(\frac{\omega T}{2}) sin(\theta + \frac{\omega T}{2}) \\ v \\ \theta + \omega T \\ \omega \end{bmatrix}, \tag{3.7}$$

U_k is the input vector given by

$$U_k = \begin{bmatrix} v_{ox}T \\ v_{oy}T \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.8}$$

and T is the sampling time. Note that the input vector contains the ownship information. The measurement equation becomes:

$$z = h_1(X) + \epsilon_\beta = arctan\left(\frac{x_2}{x_1}\right) + \epsilon_\beta, \tag{3.9}$$

where $h_1(X)$ is the measurement function and ϵ_β is zero mean Gaussian noise with a standard deviation of σ_β.

Model in the Modified Polar Coordinates

We can also write (3.4) in the Modified Polar Coordinates (MPC). The state variables in the MPC are the bearing rate, the range rate over range, the bearing, the inverse range, and the angular velocity. Let $r = \sqrt{r_x^2 + r_y^2}$ and define the state variables Y as

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} \dot{\beta} \\ \dot{r}/r \\ \beta \\ 1/r \\ \omega \end{bmatrix}. \tag{3.10}$$

The dynamic model of Y is:

$$\dot{Y} = \begin{bmatrix} -2y_1y_2 + v_{ox}y_4y_5cos(y_3) + v_{oy}y_4y_5sin(y_3) + y_2y_5 \\ y_1^2 - y_2^2 + v_{ox}y_4y_5sin(y_3) - v_{oy}y_4y_5cos(y_3) - y_1y_5 \\ y_1 \\ -y_2y_4 \\ 0 \end{bmatrix} \tag{3.11}$$

$$= f(Y, v_{ox}, v_{oy}), \tag{3.12}$$

and the measurement model for MPC is:

$$z = h_2(Y) + \epsilon_\beta = y_3 + \epsilon_\beta. \tag{3.13}$$

Note that for the MPC model, the dynamics (3.12) is highly nonlinear while the measurement model (3.13) is linear. A derivation of the MPC coordinates can be found in A.

3.2 Filter Bank Algorithm

To estimate the states in (3.7) and (3.9), or (3.12) and (3.13), we now design an EKF bank algorithm. Suppose that there are N_t filters in the EKF bank. Each EKF in the bank is initialized with a different initial state estimate, propagated through the prediction stage, and updated in the update stage using the measurement at time step k.

In particular, for filter $i = 1, \cdots, N_t$, the update stage is defined as:

$$S_k^{(i)} = H_k^{(i)} P_{k|k-1}^{(i)} {(H_k^{(i)})}^T + R \tag{3.14}$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} {(H_k^{(i)})}^T (S_k^{(i)})^{-1} \tag{3.15}$$

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k^{(i)} (z_k - h(\hat{x}_{k|k-1}^{(i)}) \tag{3.16}$$

$$P_{k|k}^{(i)} = (I - K_k^{(i)} H_k^{(i)}) P_{k|k-1}^{(i)}, \tag{3.17}$$

where R is the measurement noise, $H_k^{(i)}$ is the Jacobian of the measurement function $h(\cdot)$ (for either (3.9) or (3.13) depending on the system model), $P_{k|k-1}^{(i)}$ is the predicted covariance estimate, $P_{k|k}^{(i)}$ is the updated covariance estimate, $\hat{x}_{k|k-1}^{(i)}$ is the predicted state estimate, and $\hat{x}_{k|k}^{(i)}$ is the updated state estimate.

Each filter is associated with a weight, $W_k^{(i)}$, at time step k. The weights are initialized uniformly and summed to one. The weight of each EKF in the bank is updated using Bayes rule:

$$W_k^{(i)} = c W_{k-1}^{(i)} p(z_k|i) \tag{3.18}$$

$$\textstyle\sum_{i=1}^{N_t} W_k^{(i)} = 1, \tag{3.19}$$

where c is a normalizing constant, and $p(z_k|i)$, the likelihood for the ith EKF, is given by:

$$p(z_k|i) \propto \frac{1}{\sqrt{det(S_k^{(i)})}} exp\left(-\frac{1}{2}(z_k - h(\hat{x}_{k|k-1}^{(i)}))^T {S_k^{(i)}}^{-1} (z_k - h(\hat{x}_{k|k-1}^{(i)}))\right). \tag{3.20}$$

The combined estimate and covariance are calculated as:

$$\hat{x}_{k|k} = \textstyle\sum_{i=1}^{N_t} W_k^{(i)} \hat{x}_{k|k}^{(i)} \tag{3.21}$$

$$P_{k|k} = \textstyle\sum_{i=1}^{N_t} W_k^{(i)} (P_{k|k}^{(i)} + (\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k})(\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k})^T). \tag{3.22}$$

Initialization of Filter Bank Algorithm - CPPV model

For the CPPV implementation, we parameterize the EKFs by their initial range, angular velocity, and heading estimates. Suppose that we are given an upper bound and lower bound of the initial range of the intruder $[r_{min}, r_{max}]$. We divide this interval into N_r subintervals. The subintervals are distributed geometrically according to [22]:

$$r^{(j)} = \frac{r_{min}}{2}\left(\rho^{j} + \rho^{j-1}\right), \rho = \left(\frac{r_{max}}{r_{min}}\right)^{\frac{1}{N_r}} \tag{3.23}$$

$$\sigma_r^{(j)} = C_r r^{(j)}, C_R = \frac{2(\rho - 1)}{\sqrt{12}(\rho + 1)}, \tag{3.24}$$

where $j = 1, \cdots, N_r$ represents the index of a specific subinterval, $r^{(j)}$ is the jth range estimate in the interval, $\sigma_r^{(j)}$ is the jth standard deviation for range, and C_R is the coefficient of variation.

Similarly, for the angular velocity and the heading, we divide their initial state and variances into uniform intervals. We divide the angular velocity into N_ω subintervals and the heading into N_θ subintervals according to predefined limits $[\omega_{min}, \omega_{max}]$ and $[\theta_{min}, \theta_{max}]$, respectively, as:

$$\omega^{(k)} = \omega_{min} + (k - 1)\frac{\omega_{max} - \omega_{min}}{N_\omega - 1} \tag{3.25}$$

$$\sigma_\omega = \frac{N_\omega}{2}\frac{\omega_{max} - \omega_{min}}{N_\omega - 1} \tag{3.26}$$

$$\theta^{(\ell)} = \theta_{min} + (\ell - 1)\frac{\theta_{max} + \theta_{min}}{N_\theta} \tag{3.27}$$

$$\sigma_\theta = \frac{1}{2}\frac{\theta_{max} + \theta_{min}}{N_\theta}, \tag{3.28}$$

where $k = 1, \cdots, N_\omega$ and $\ell = 1, \cdots, N_\theta$ represent the index of a specific subinterval for the angular velocity and heading parameterization respectively. The kth initial estimate for the angular velocity is $\omega^{(k)}$, and the ℓth initial estimate for the heading is $\theta^{(\ell)}$. The standard deviation for the angular velocity and heading, σ_ω and σ_θ, are constant for each of their respective subintervals. In the special case where $N_\theta = 1$

and $N_\omega = 1$, we set:

$$\omega^{(1)} = \frac{\omega_{max} + \omega_{min}}{2} \tag{3.29}$$

$$\sigma_\omega = \frac{\omega_{max} + \omega_{min}}{2} \tag{3.30}$$

$$\theta^{(1)} = 0 \tag{3.31}$$

$$\sigma_\theta = \sqrt{\frac{\pi^2}{3}}. \tag{3.32}$$

To initialize filter i, where $i = 1 \cdots N_t$, we make use of $r^{(j)}$, $\omega^{(k)}$, and $\theta^{(\ell)}$ defined in (3.23), (3.25), and (3.27), respectively. Suppose

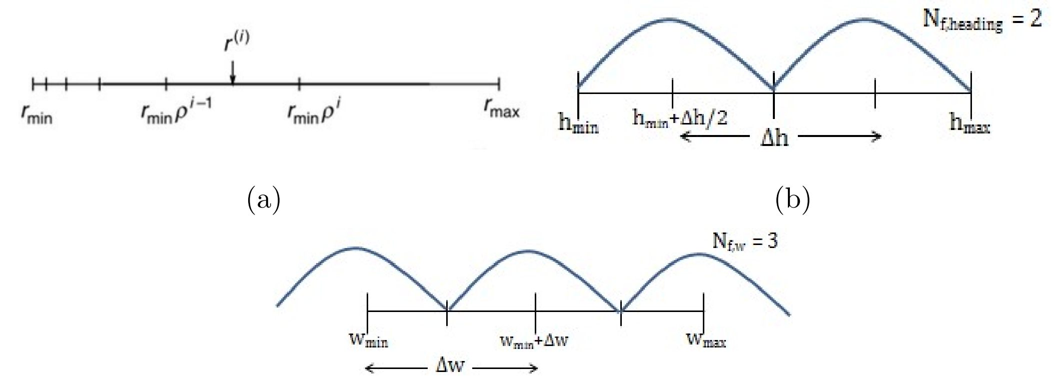
$$i = N_r N_\omega(\ell - 1) + N_r(k - 1) + j, \tag{3.33}$$

for some $j \in \{1, \cdots, N_r\}$, $k \in \{1, \cdots, N_\theta\}$, and $\ell \in \{1, \cdots, N_\omega\}$. Then the initial value of the state vector for the ith EKF is:

$$x_0^{(i)} = \begin{bmatrix} r^{(j)}cos(\beta_0) \\ r^{(j)}sin(\beta_0) \\ v_0 \\ \theta^{(k)} \\ \omega^{(\ell)} \end{bmatrix}, \tag{3.34}$$

where β_0 is the first measured bearing angle, and v_0 is the ownship velocity, where no prior knowledge of the intruder's velocity is available.

Figure 3.2 visualizes how we parameterize with respect to the range, heading, and angular velocity states.



(a)

(b)

(c)

Figure 3.2: An illustration that represents the parameterization for the range(a), heading(b), and angular velocity(c) states of the Filter bank algorithm. Illustration (a) is from [22]

The initial error covariance is given by,

$$P_{0,LOS}^{(i)} = \begin{bmatrix} \left(\sigma_r^{(i)}\right)^2 & 0 \\ 0 & \left(r^{(i)}\sigma_{\beta_0}^{(i)}\right)^2 \end{bmatrix}, R_{rot} = \begin{bmatrix} cos\beta_0 & sin\beta_0 \\ -sin\beta_0 & cos\beta_0 \end{bmatrix},$$

$$P_{0,Local}^{(i)} = R_{rot} P_{0,LOS}^{(i)} R_{rot}$$

$$P_0^{(i)} = diag\left(P_{0,Local}^{(i)}, (\sigma_v)^2, (\sigma_\omega)^2, (\sigma_\theta)^2\right), \tag{3.35}$$

where the position variance $P_{0,LOS}^{(i)}$ is aligned to the local coordinate system from the line of sight (LOS), and σ_ω and σ_θ are defined in (3.26) and (3.28), respectively. We approximate σ_v^2, the variance of the intruder velocity as:

$$\sigma_v^2 = \frac{1}{12}(v_{max} - v_{min})^2, \tag{3.36}$$

where v_{max} and v_{min} are limits that we predefine based on airspeeds of typical aircraft.

Initialization of Filter Bank Algorithm - MPC model

For the implementation in the MPC, we parameterize the EKFs in the same fashion as the CPPV implementation. We do this to ensure that the filter bank algorithm for both models are initialized with the same initial conditions. We map the initial state for each EKF from the CPPV model to the MPC model using a transformation $G(\cdot)$ given by

$$G(X, v_{ox}, v_{oy}) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} \frac{x_1(x_3 sin(x_4) - v_{oy}) - x_2(x_3 cos(x_4) - v_{ox})}{x_1^2 + x_2^2} \\ \frac{x_1(x_3 cos(x_4) - v_{ox}) + x_2(x_3 sin(x_4) - v_{oy})}{x_1^2 + x_2^2} \\ tan^{-1}\left(\frac{x_2}{x_1}\right) \\ \frac{1}{\sqrt{x_1^2 + x_2^2}} \\ x_5 \end{bmatrix}. \tag{3.37}$$

For each filter, the initial value of the state vector is:

$$y_0^{(i)} = G(x_0^{(i)}, v_{ox}, v_{oy}). \tag{3.38}$$

The initial state covariance matrix is defined as:

$$P_0^{(i)} = diag[\sigma^2_{\dot{\beta}}, \sigma^2_{\frac{\dot{r}}{r}}, \sigma^2_{\beta}, \sigma^2_{\frac{1}{r}}, \sigma^2_{\omega}]. \tag{3.39}$$

Because the standard deviations are different for the CPPV and MPC models, we ensure that they are made as equivalent as possible [15]. We achieve this by:

$$\sigma_{\dot{\beta}} = \frac{\sigma_v}{r^{(j)}}, \sigma_{\frac{\dot{r}}{r}} = \frac{\sigma_v}{r^{(j)}}, \sigma_{\frac{1}{r}} = \frac{\sigma_r}{{r^{(j)}}^2}. \tag{3.40}$$

We numerically integrate (3.12) to predict the estimate of the system's state and covariance matrix at the next time step. We make use of the Euler method for the integration over time [28]. We use a time step of 0.001 sec for this integration. The update step is kept the same as the discrete-time EKF. A derivation of the mapping function $G(\cdot)$ and a derivation of the relationship of the variance between the range and inverse range covariance can be found in Appendix B and E respectively.

3.3 IMM Algorithm

The algorithms in Section 3.2 are designed to estimate an orbiting trajectory of an intruder. However, the intruder's trajectory could consist of straight legs and orbiting legs. Therefore, it is important to integrate the algorithm in Section 3.2 with an estimator that tracks the intruder during the straight leg and examine the overall tracking performance. Such an integration is typically performed using IMM algorithms.

For our IMM, we utilize two different models to describe the intruder's motion: a constant velocity (CV) model and coordinated turn (CT) model. We utilize the filter bank algorithm for the CT model as described in Section 3.2, and a single EKF for the CV model. Note that designing an estimator for the CV model is not the focus of the thesis because extensive research has been conducted for passive ranging with the CV model [15, 22, 29]. Therefore, we use a single EKF as an example to illustrate how the filter bank algorithm is integrated in the IMM framework. More specifically, we utilize the CPPV coordinates in our studies. To simplify notations, we denote the index for CV and CT as 1 and 2, respectively. The standard IMM mixing stage is modified for the filter bank algorithm.

In addition to this modification, we implement a mixing strategy to this as well. The IMM mixing rate, $T_d(hz)$, is used to mix the CT and CV model estimate. At every T_d, the IMM mixing stage is used. When the filter is not at T_d, the CT and CV models do not mix their estimates and their filters are updated separately.

Mixing of State Estimates

Relevant terms

- W_i associated weight for filter i in filter bank
- $\bar{X}^2$ weighted mean state estimate for CT model
- $\bar{P}^2$ weighted error covariance for CT model
- $X^{2,n}_{k-1|k-1}$ state estimate of filter n for CT model
- $P^{2,n}_{k-1|k-1}$ error covariance of filter n for CT model

We modify equations 2.19 and 2.21 as follows, where for filter n of N_t filters in the filter bank algorithm,

$$X^{01}_{k-1|k-1} = X^{1}_{k-1|k-1}\mu_{k-1|k-1}(1|1) + \bar{X}^{2}_{k-1|k-1}\mu_{k-1|k-1}(2|1) \tag{3.41}$$

$$X^{02,n}_{k-1|k-1} = X^{1}_{k-1|k-1}\mu_{k-1|k-1}(1|2) + X^{2,n}_{k-1|k-1}\mu_{k-1|k-1}(2|2), \tag{3.42}$$

and

$$\begin{aligned} P^{01}_{k-1|k-1} = {} & \mu_{k-1|k-1}(1|1)\left[P^{1}_{k-1|k-1} + \left(X^{1}_{k-1|k-1} - X^{01}_{k-1|k-1}\right)\left(X^{1}_{k-1|k-1} - X^{01}_{k-1|k-1}\right)^{T}\right] \\ & + \mu_{k-1|k-1}(2|1)\left[P^{1}_{k-1|k-1} + \left(\bar{X}^{2}_{k-1|k-1} - X^{01}_{k-1|k-1}\right)\left(\bar{X}^{2}_{k-1|k-1} - X^{01}_{k-1|k-1}\right)^{T}\right] \end{aligned} \tag{3.43}$$

$$\begin{aligned} P^{02,n}_{k-1|k-1} = {} & \mu_{k-1|k-1}(1|2)\left[P^{2,n}_{k-1|k-1} + \left(X^{1}_{k-1|k-1} - X^{02,n}_{k-1|k-1}\right)\left(X^{1}_{k-1|k-1} - X^{02,n}_{k-1|k-1}\right)^{T}\right] \\ & + \mu_{k-1|k-1}(2|2)\left[P^{2,n}_{k-1|k-1} + \left(\bar{X}^{2,n}_{k-1|k-1} - X^{02,n}_{k-1|k-1}\right)\left(\bar{X}^{2,n}_{k-1|k-1} - X^{02,n}_{k-1|k-1}\right)^{T}\right] \end{aligned} \tag{3.44}$$

The model transition probability matrix p_{ij} for two models is

$$p_{ij} = \begin{bmatrix} 1-\delta & \delta \\ \delta & 1-\delta \end{bmatrix} \quad , \tag{3.45}$$

where δ is a small off-diagonal user-defined value.

CV model in CPPV Coordinates

The CV model utilizes a single EKF. The state vector in the CV model is $X^{01} = [r_x\ r_y\ v\ \theta]$. The state transition matrix, $\Phi^{CV}_{k|k-1}$ is given by

$$\Phi^{CV}_{k|k-1} = \begin{bmatrix} r_x + vTcos(\theta) \\ r_y + vTsin(\theta) \\ v \\ \theta \end{bmatrix}, \tag{3.46}$$

and the Jacobian for this dynamics model is

$$F_k = \begin{bmatrix} 1 & 0 & Tcos(\theta) & -vTsin(\theta) \\ 0 & 1 & Tsin(\theta) & vTcos(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \tag{3.47}$$

Equations 3.46 - 3.47 are used in the standard EKF equations 2.8 - 2.14 to estimate the state vector.

CHAPTER 4

Simulation Studies

4.1 Orbiting Intruder Studies

In this section, we present Monte-Carlo simulation results of the orbiting intruder ranging problem in Section 3.1 using the filter bank algorithm described in Section 3.2. In particular, we conduct two studies. In the first one, we investigate the important states to parameterize for the CPPV and MPC models. In the second study, we examine the robustness of the filters with respect to noise in the ownship velocity.

For each study, we conduct 500 simulations to evaluate the performance of the filter in the CPPV and MPC models. Our metric of performance is the root mean square error (RMSE). To illustrate how the RMSE is calculated, we define the relative percentage error at time k for the ith simulation as:

$$e^i(k) = \frac{|\bar{r^i}(k) - r(k)|}{r(k)}, \tag{4.1}$$

where $\bar{r}$ represents the estimates of a particular state, e.g., the range, and r represents the truth corresponding to that state. The average error of N simulations at each time step is given by:

$$RMSE = \left(\frac{\sum_{i=1}^{N} e^i(1)}{N}, \ldots, \frac{\sum_{i=1}^{N} e^i(k)}{N} \right). \tag{4.2}$$

We calculate the RMSE for the range (r), linear intruder velocity (v), and the angular velocity (ω). For the heading variable (θ), we calculate the absolute difference between the estimate and the true value. The absolute difference at time k for the

ith simulation can be defined as:

$$d^i(k) = |c\left(\bar{\theta^i}(k) - \theta(k)\right)|, \tag{4.3}$$

where $c(\cdot)$ is a function that maps its argument to the [-π, π) region.

For each simulation, the intruder/ownship geometry is different. The intruder conducts a coordinated turn motion at a constant angular velocity and a constant linear velocity. In each simulation, values for the angular velocity and linear velocity are constant and sampled randomly between 3 and 8 deg/s and between 60 and 80 m/s, respectively. The intruder orbit direction is random, and can be clockwise or counter-clockwise. The initial heading of the intruder orbit is randomly sampled between 0 and 360 deg. The ownship travels at a constant linear velocity of 40 m/s. The initial range between the intruder and ownship is randomly sampled between 1.8 and 6.5 km. The angle between the ownship and the center of the intruder's orbit is allowed to vary from 0 and 180 deg. For an encounter, the ownship vehicle takes bearing measurements of the intruder vehicle at a sampling rate of $T = 0.05$ sec for a total encounter period of 120 sec. Figure 4.1 shows a few possible trajectories for intruder encounters. There is no field of view constraint for the scenarios. The process noise for the CPPV model is set as:

$$Q_{CPPV} = Cov(w_k) = diag\left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, T^2\sigma_{\dot{v}}^2, \begin{bmatrix} T^3\sigma_{\dot{\omega}}^2/3 & T^3\sigma_{\dot{\omega}}^2/2 \\ T^2\sigma_{\dot{\omega}}^2/2 & T^2\sigma_{\dot{\omega}}^2 \end{bmatrix}\right),$$

and the process noise for the MPC model is set as:

$$Q_{MPC} = Cov(w_k) = q \cdot I_5.$$

We set the algorithm parameters as in Table 4.1.

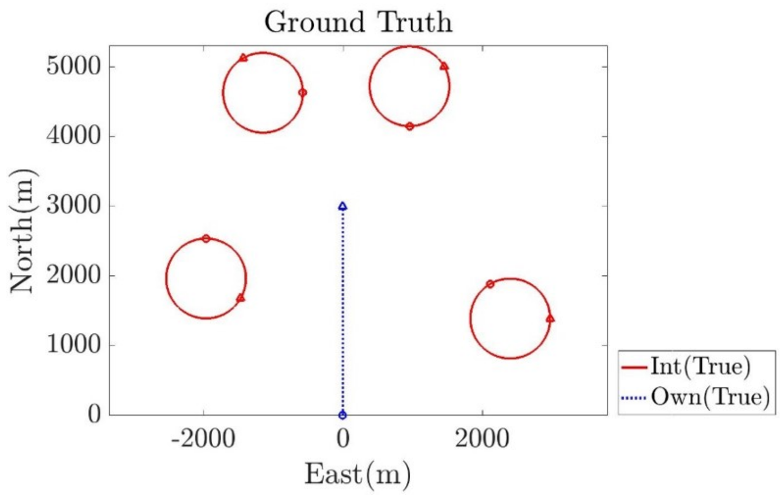


Figure 4.1: Four sample scenarios of the orbiting intruder encounter that can be generated during Monte Carlo Simulations. The ownship moves on the same linear trajectory for each generated scenario.

Table 4.1: Algorithm Parameters

$r_{max} = 10km$	$r_{min} = 1km$	$\omega_{max} = 8deg/s$	$\omega_{min} = -8deg/s$
$\theta_{max} = 2\pi$	$\theta_{min} = 0$	$v_{initial} = 40m/s$	$\sigma_v^2 = 40^2/12$
$\sigma_{\dot{v}}^2 = 1e-3$	$\sigma_{\dot{\omega}}^2 = 5e-6$	$q = 1e-12$	$R = 0.2deg$

4.1.1 Study 1: Effects of Parameterization

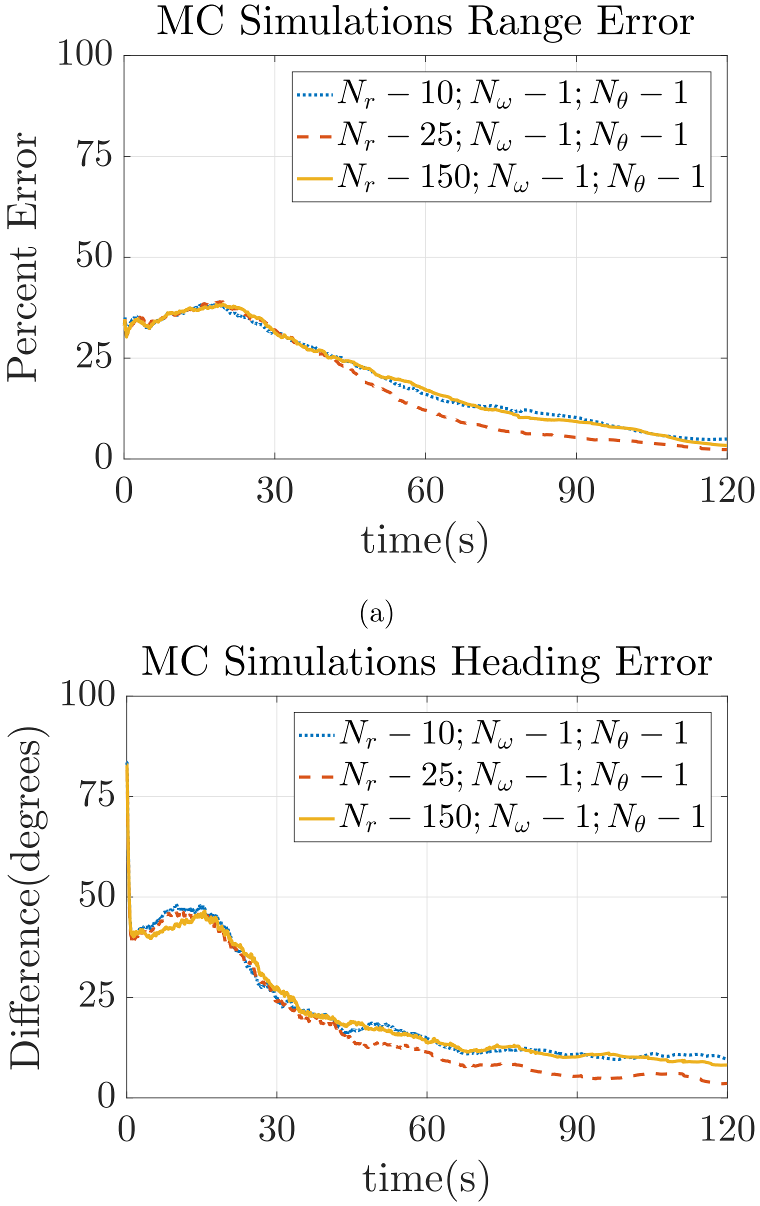
For this study we parameterize the filter in CPPV under different combinations listed in Table 4.2. For each combination, we limit the total number of filters N_t to be less than 150.

Table 4.2: Study 1 Cases

Study	Model	Parameterization			Figure
		Nr	N_ω	N_θ	
Case 1	CPPV	10	1	1	Fig 4.2
		25	1	1	
		150	1	1	
Case 2	CPPV	25	1	1	Fig 4.3
		25	3	1	
		25	6	1	
Case 3	CPPV	25	1	1	Fig 4.4
		25	1	3	
		25	1	6	
Case 4	CPPV	25	1	6	Fig 4.5
		25	2	3	
		15	2	5	
		16	3	3	
Case 5	CPPV	25	1	6	Fig 4.6
	MPC	15	5	2	

Case 1: Varying Range Intervals

For the first case, we test the effects of increasing the number of intervals for the range, N_r. The heading and angular velocity intervals are kept as 1, $N_\theta = 1$, N_ω =1 for each variation. We test three range intervals, N_r = 10, 25, and 150. The estimation performance corresponding to these three variations are shown in Figure 4.2. For the three variations, we observe that the second case, where N_r = 25, performs best for all state estimates. Our explanation of this behavior is that when the number of range subintervals is too large, the resulting range variance for each EKF in turn decreases. When the filter has too much belief in an erroneous range estimate, the filter update stages will then take a longer amount of time to converge.



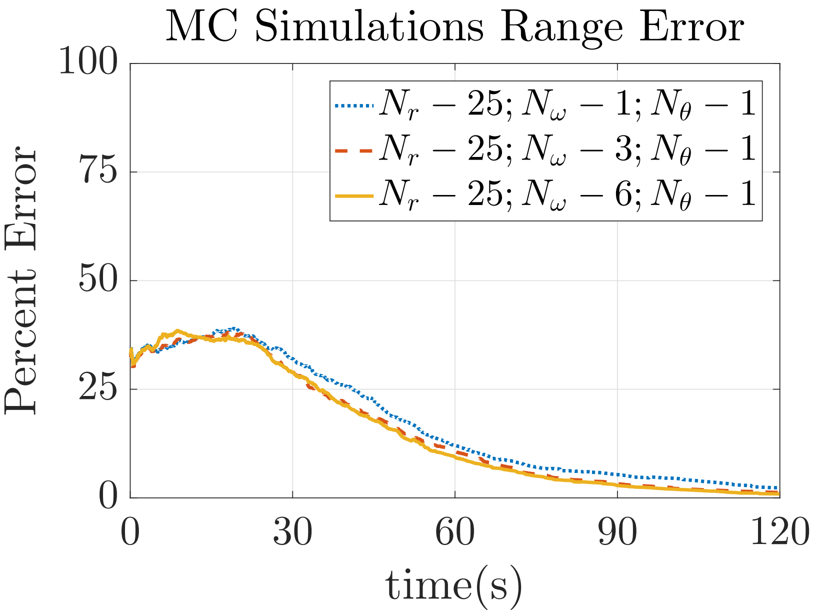
(a)

(b)

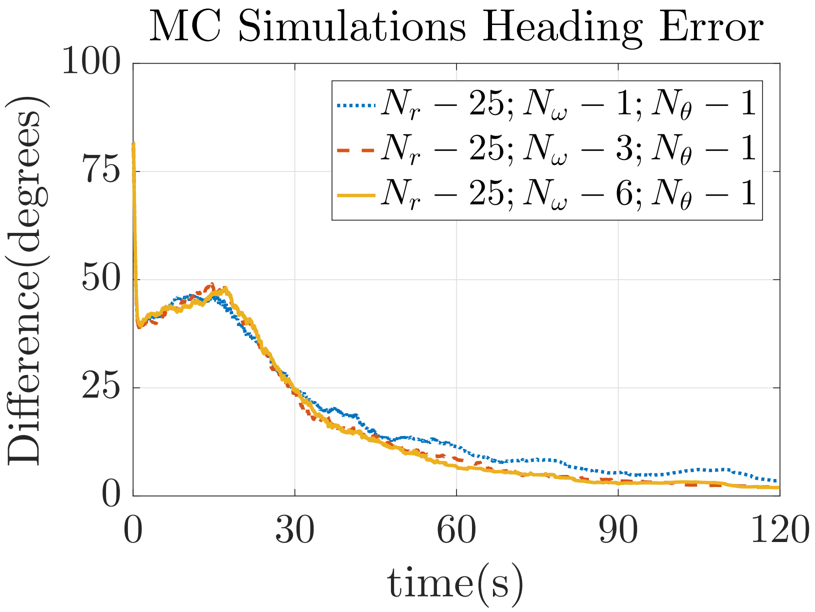
Figure 4.2: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the CPPV system model. We plot three parameter variations with a focus on different range interval sizes.

Case 2: Varying Angular Velocity Intervals

For the second case, we compare the effects of parameterizing the angular velocity together with the range. We maintain the number of intervals for the range and the heading as $N_r = 25$ and $N_\theta = 1$, respectively. The initial angular velocity is parameterized for three variations: $N_\omega = 1, 3$, and 6, respectively. The simulation results are shown in Figure 4.3. The cases where $N_\omega = 3$ and $N_\omega = 6$ perform better than $N_\omega = 1$. The performance of $N_\omega = 3$ and $N_\omega = 6$ is similar except that there is a slight increase in percentage error for the range estimates when $N_\omega = 6$.



(a)

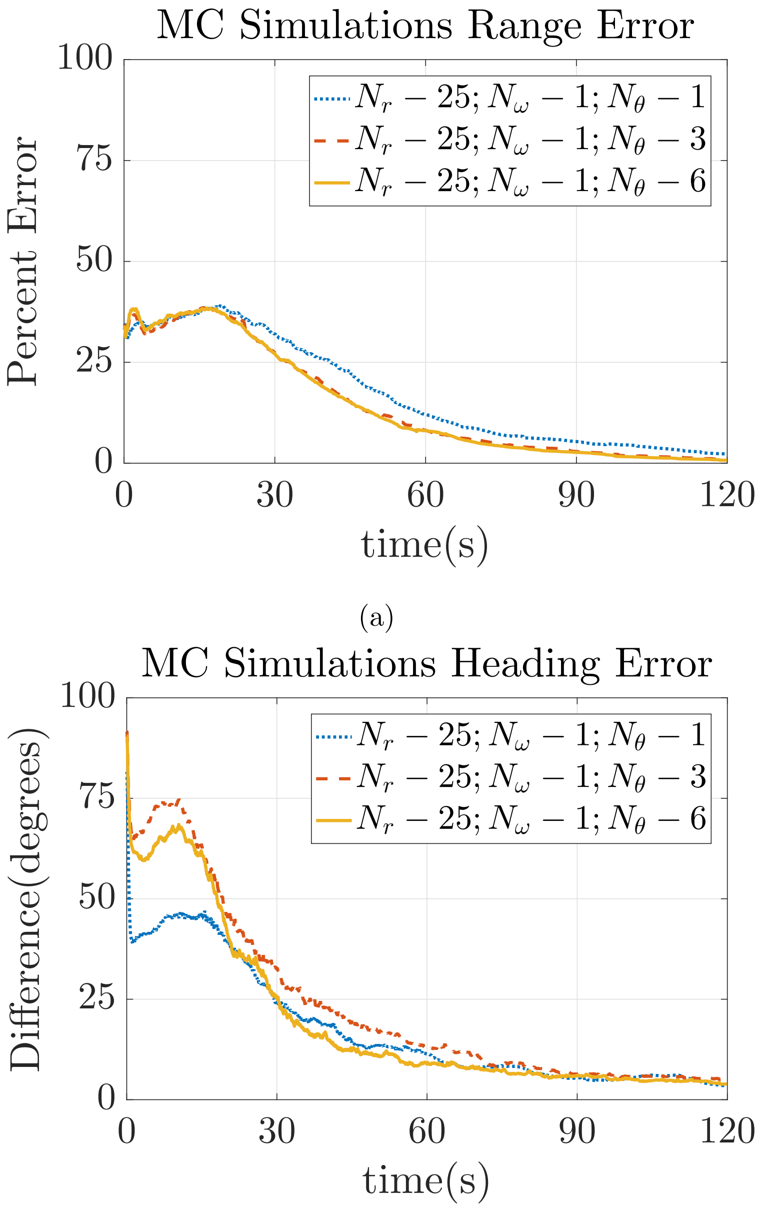


(b)

Figure 4.3: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the CPPV system model. We plot three variations with a focus on different angular velocity interval sizes.

Case 3: Varying Heading Intervals

Similarly, for the third case, we compare the effects of parameterizing the heading with the range. We maintain the number of intervals for the range and the angular velocity as $N_r = 25$ and $N_\omega = 1$, respectively. The heading states are tested for three interval variations, $N_\theta = 1$, 3, and 6, respectively, in Figure 4.4. The second and third variation, $N_\theta = 3$ and $N_\theta = 6$ both perform similarly for the range estimates. The third variation, however, performs best for the heading estimate. It reaches a steady error of 10 percent at $t = 60$s while the second variation reaches a 10-percent error at $t = 70$s.



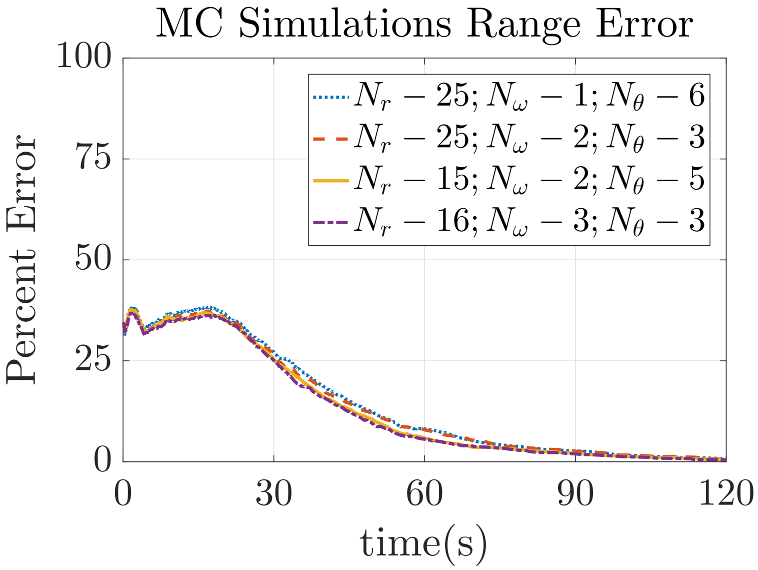
(a)

(b)

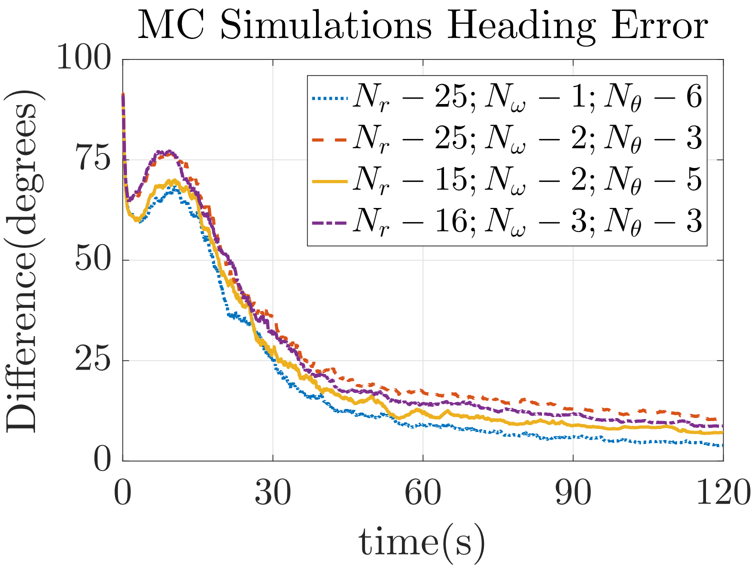
Figure 4.4: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the CPPV system model. We plot three parameter variations with a focus on different heading interval sizes.

Case 4: Varying All Intervals

For the fourth case, we examine parametrizing the range, heading, and angular velocity. The parameterization combinations can be seen in Figure 4.5. The fourth variation, where $N_r = 16$, $N_\omega = 3$, and $N_\theta = 3$ performs the best when estimating the range. This is not the case for the heading estimates, where the best parametrization is the first variation, $N_r = 25$, $N_\omega = 1$, and $N_\theta = 6$. This performance appears to be due to the number of heading intervals being highest. Overall, we consider that the parameterization $N_r = 25$, $N_\omega = 1$, and $N_\theta = 6$ yields the best performance.



(a)

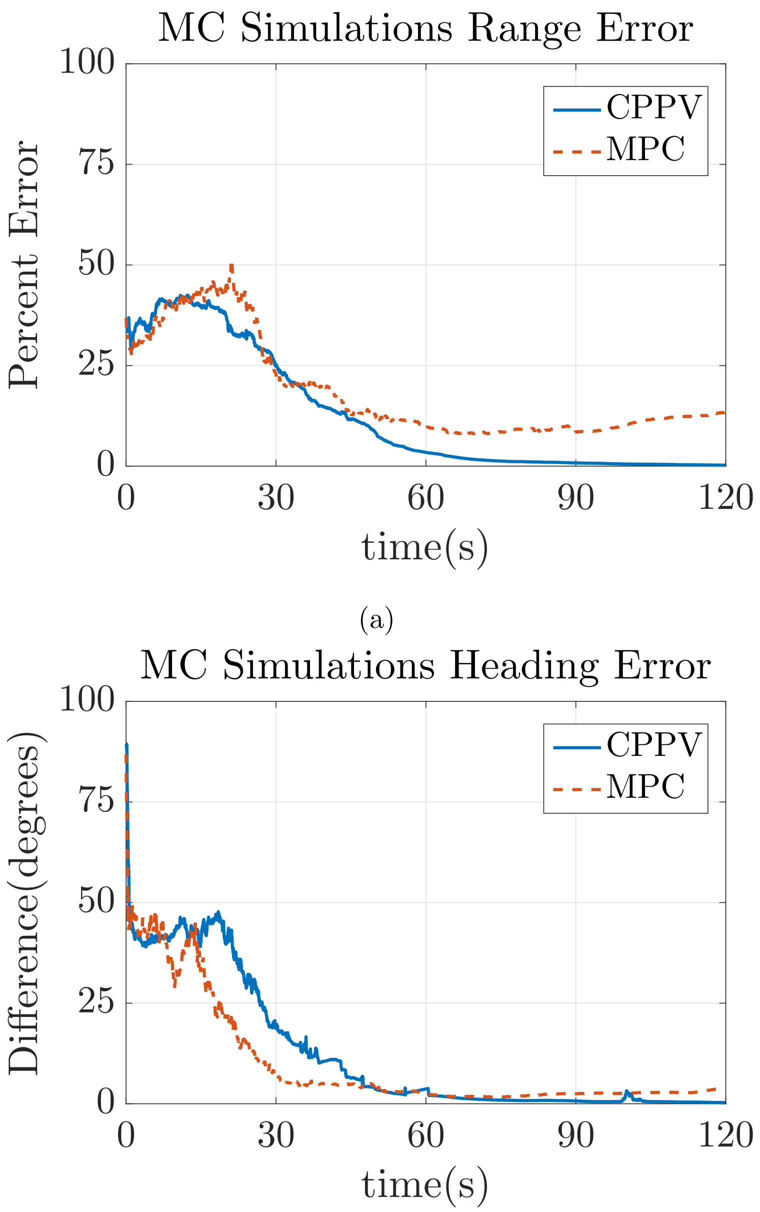


(b)

Figure 4.5: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the CPPV system model. We plot parameter variations that vary for all three states.

Case 5: Comparison of CPPV and MPC

Lastly, we compare the performance of the CPPV model to the MPC model in Figure 4.6. It should be noted that we perform the same type of parameterization study for the MPC model. In favor of space, we do not include the results of the study. From that study, we conclude that the parameterization priority should be given to the range and angular velocity. The parameterization chosen for the CPPV model was $N_r = 25$, $N_\omega = 1$, and $N_\theta = 6$, whereas the parameterization for the MPC model was $N_r = 15$, $N_\omega = 5$, and $N_\theta = 2$.



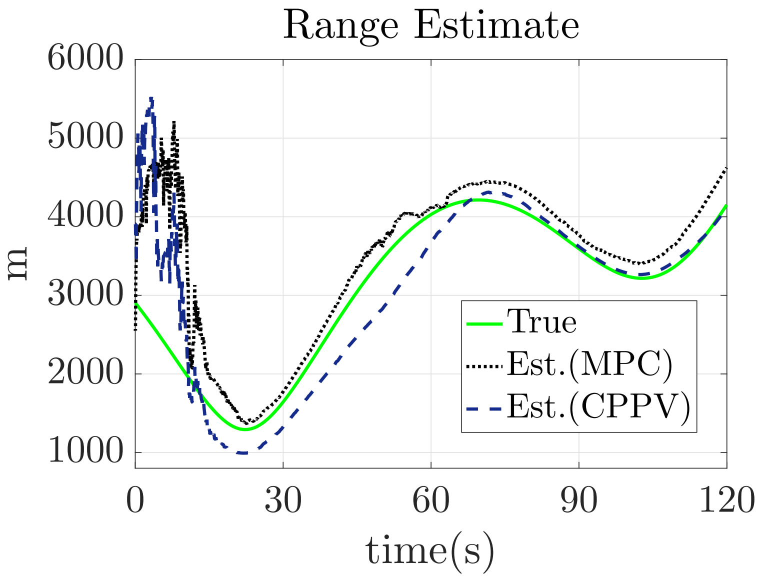
(a)

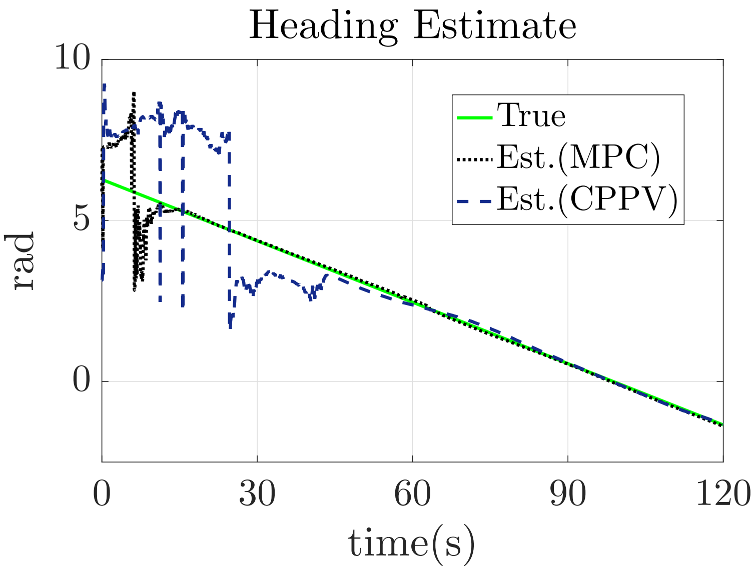
(b)

Figure 4.6: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations are plotted for the filter algorithm in CPPV and MPC system models. The best parameterization combination for each respective model are used.

From the simulations, we observe that there are more divergent scenarios for the MPC model. Those divergent scenarios are removed from the data. To ensure a fair comparison, for the CPPV model we remove the same number of scenarios, which have worse range estimation performance than others. For this reason, the same data set used in Figure 4.5 does not appear the same as in Figure 4.6. Figure 4.6 shows that the CPPV model is outperformed by the MPC model when estimating the heading state. The MPC model achieves a faster convergence for the heading estimate at $t = 30$ sec, however, the range estimates it produces do not reach the accuracy of the CPPV model estimates.

Figure 4.7 shows a single run from the simulations where the range estimate for the filter in CPPV is converging while the range estimate for the MPC model appears to drift from the truth. However, the heading estimate from the MPC converges faster than the CPPV model.



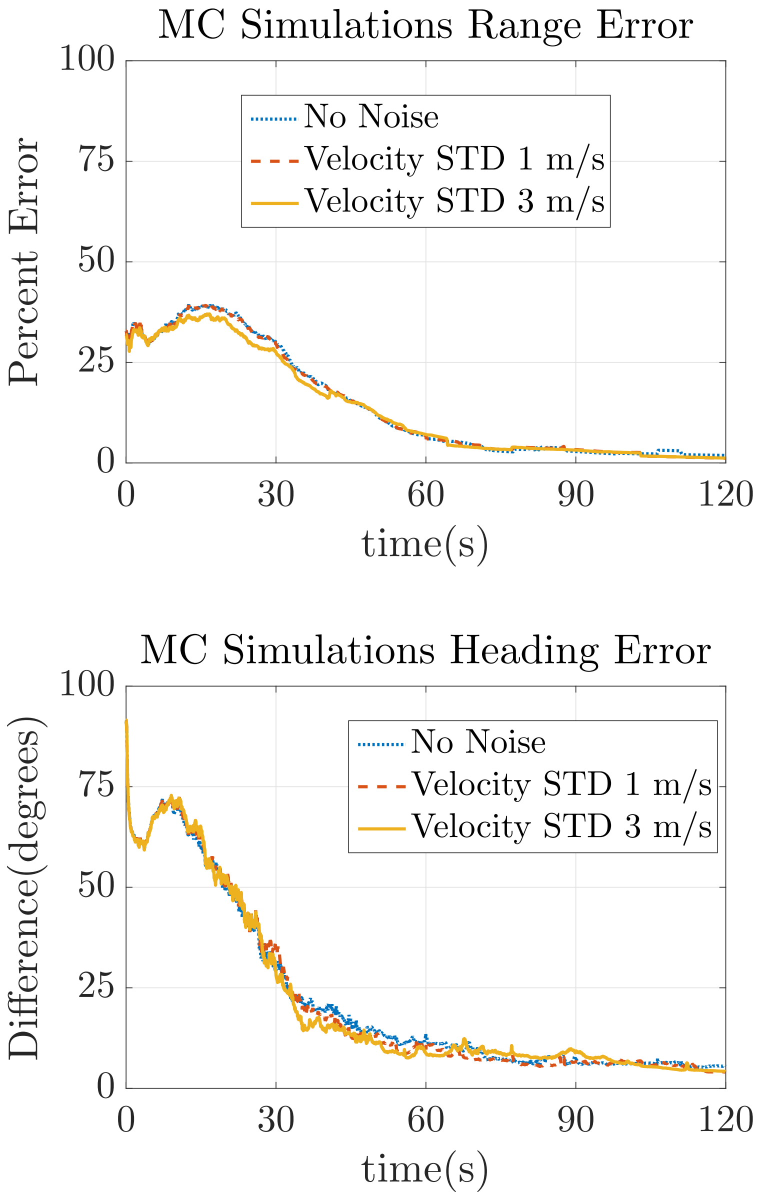


(a)

Figure 4.7: A comparison of the filter in CPPV and MPC models for one simulation chosen from the 500 Monte Carlo Simulations of Fig 4.6 . The range (a) and heading (b) state estimates are compared against the truth.

4.1.2 Study 2: Robustness to Velocity Noise

For this study, we examine the robustness of the filter algorithm for both the CPPV and MPC models. We test this by adding gaussian noise to the ownship velocity. For each model, we use the same parameterization used in Figure 4.6. We make a comparison for three variations for the ownship velocity's noise standard deviation (STD): a noiseless case, 1 m/s, and 3 m/s in Figure 4.8. We conclude that a small amount of ownship velocity noise has little effect on the estimation performance. We have also performed a test by adding a measurement bias of 0.2 deg and found that there are negligible effects for either system model.

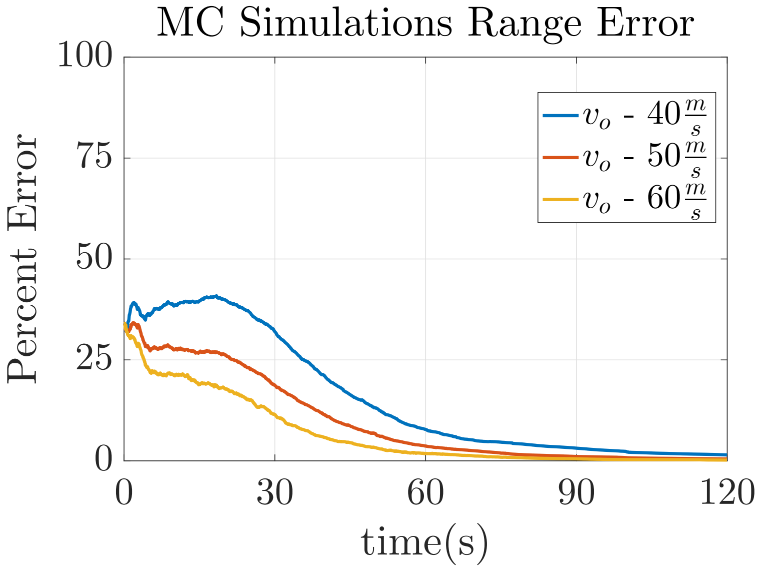


(a)

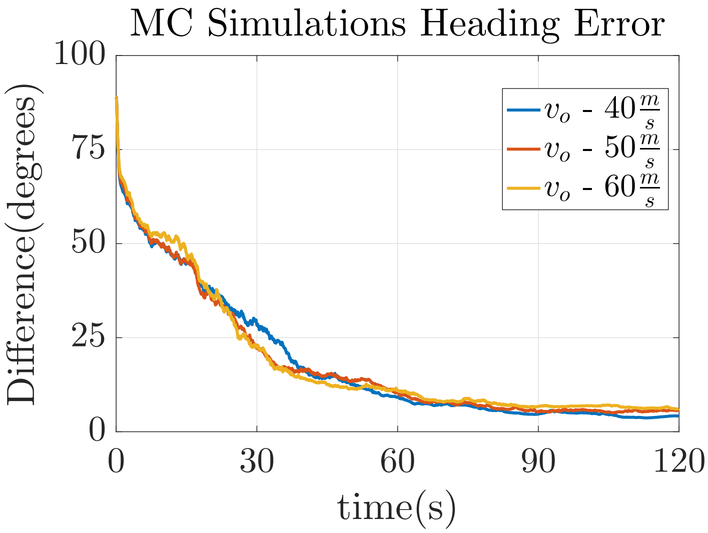
Figure 4.8: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations are plotted for the filter algorithm in the CPPV system model. Three variations of ownship velocity noise are shown.

4.1.3 Study 3: Effects of Ownship Velocity

In this study, we conduct an investigation on the effects of the ownship speed on the filter bank algorithm's performance. It is well known topic that to obtain range information of an intruder system, the ownship must conduct a higher-motion maneuver to yield observability [19, 18]. We utilize 3 variations of ownship speed,v_o, 40, 50, and 60 m/s in Figure 4.9 for 500 simulation runs. The filter parameterization utilized here is $N_r = 25$, $N_\omega = 1$, and $N_\theta = 6$. The results of this study shows that a higher ownship velocity yields better performance in range estimation. The heading estimates for all three cases exhibit similar performance.



(a)

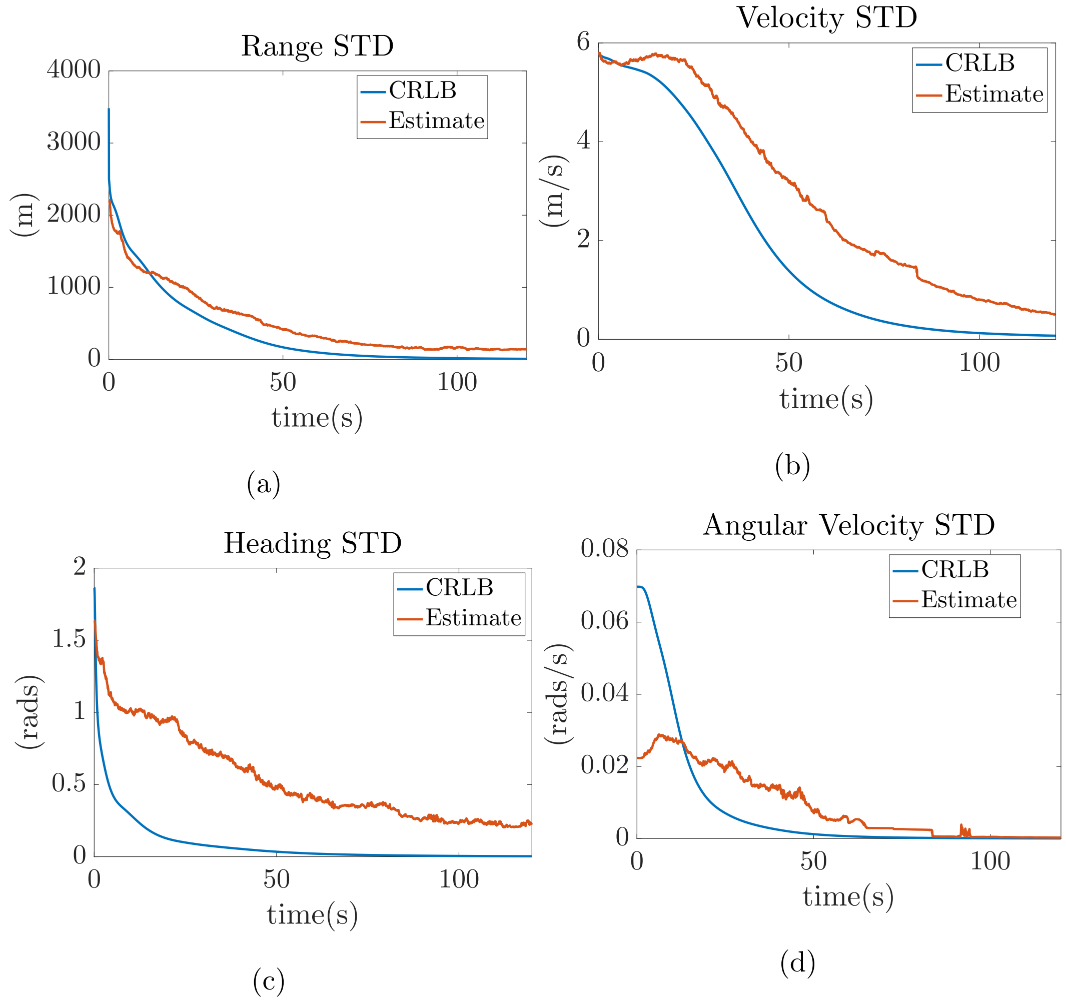


(b)

Figure 4.9: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations are plotted for the filter algorithm in the CPPV system model. Three variations of ownship speed are shown.

4.1.4 Study 4: Cramer-Rao Lower Bound Analysis

To determine the most optimal performance of the filter, we pursue utlizing a different performance metric, the Cramer-Rao Lower Bound (CRLB). We describe our implementation of the CRLB in Appendix D. 500 Monte Carlo Simulations are randomly generated utilizing the algorithm parameters cited previously in Table 4.1. However, each simulations shares the same initial bearing angle. The results of the CPPV model for the range, velocity, heading, and angular velocity are shown in figure 4.10. From inspection, it appears that the range, velocity, and angular velocity standard deviations correlate closely to that of the CRLB. This appears to be less of the case for the heading.



(a)

(b)

(c)

(d)

Figure 4.10: The Cramer Rao Lower Bound is compared to 500 randomly generated runs for the range (a), velocity (b), heading (c), and angular velocity (d).

4.2 Maneuvering Intruder Studies

In this section, we present our simulation results for the maneuvering intruder ranging problem. We utilize an implementation of the IMM algorithm as described in Section 3.3. We focus on investigating how to improve the performance of this algorithm by varying two parameters, the IMM mixing stage rate, T_d, and the model transition probability value, δ. For the CT mode, our filter bank algorithm is parameterized according to the best parameterization found in our previous study, $N_r = 25$,$N_\omega = 1$,and $N_\theta = 6$. The CV mode's single EKF is initialized using the weighted average state estimate and covariance of the filter bank. The IMM algorithm utilized the CPPV coordinates for this study. For each study, we conduct 500 simulations. Our metric of performance here is the RMSE, eq. 4.1-4.3.

Scenarios are generated similarly to the orbiting intruder studies in Section 4.1. For each simulation, the intruder/ownship geometry is different. The difference is that the intruder vehicle is in a coordinated turn for 120 sec, then transitions to travel at a constant linear velocity for 70 sec. Figure 4.11 shows a few possible trajectories for intruder encounters. The parameters utilized in this study follow Table 4.1.

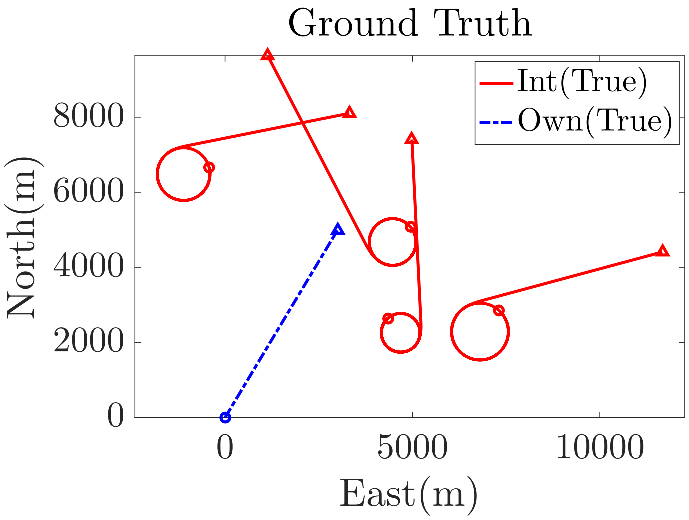
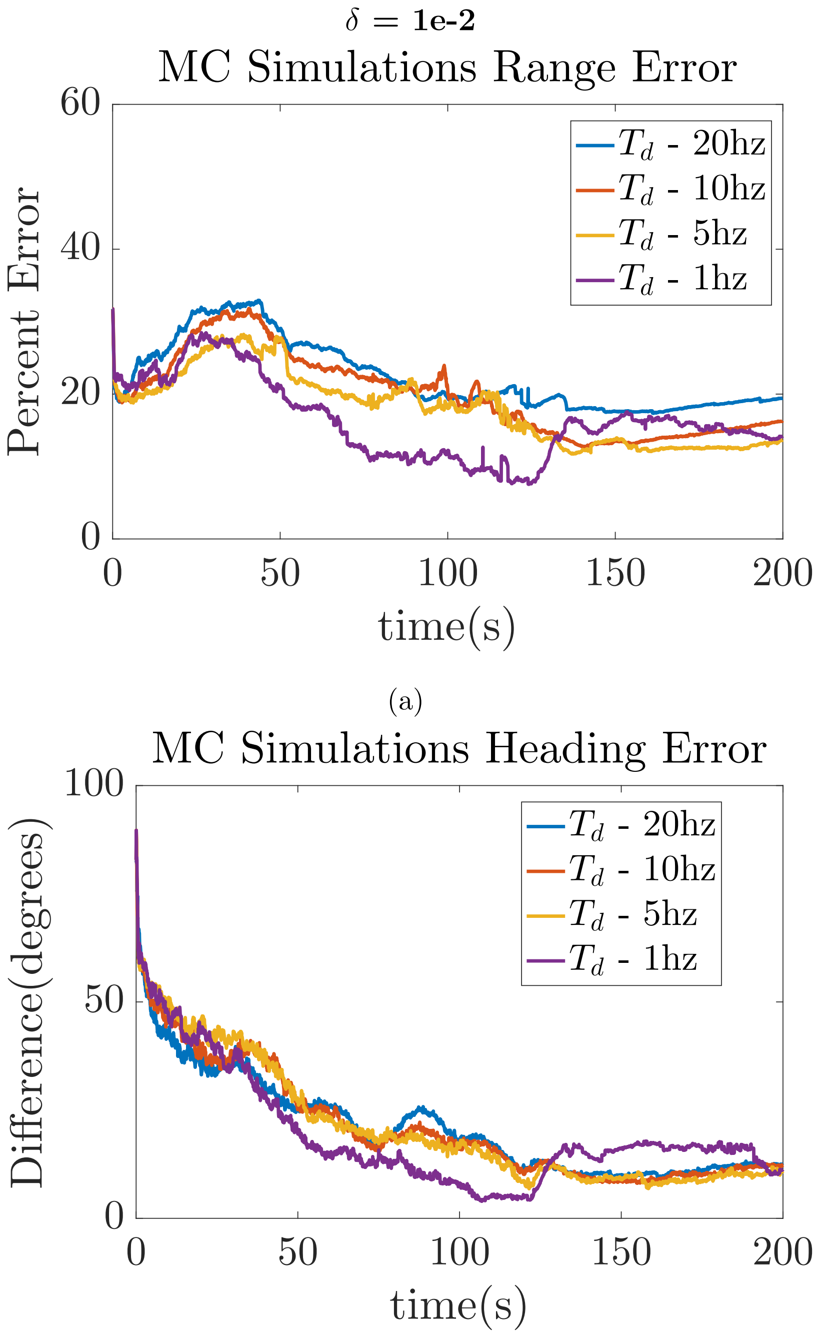


Figure 4.11: Four sample scenarios of the maneuvering intruder encounter that can be generated during Monte Carlo Simulations. The ownship moves on the same linear trajectory for each generated scenario. The maneuvering intruder orbits for 120 sec, then transitions to a constant linear velocity for 70 sec.

4.2.1 Study 1: Varying the IMM Mixing rate, T_d

In this study, we examine how varying the IMM mixing rate, T_d, effects performance. Four different IMM mixing rates are investigated, 20, 10, 5, and 1 hz. Figures 4.12 - 4.14 show this comparison for three different values for δ, the off diagonal value for the model transition probability matrix. In Figure 4.12, where δ=1e-2, it can be seen that 1 hz has the best range performance until t=120 sec, where an increase in error can be found. The IMM mix rate 20 hz performed worst for the entire simulation time. In Figures 4.13 and 4.14, where δ=1e-3 and δ=1e-4 respectively, all variations performed better than those with δ=1e-2. For these figures, an IMM mix rate of 20 hz had the best performance. For the time period until t = 120 sec, it can be seen that IMM mix rates greater than 1 hz had very similar performance. Increasing the IMM mix rate proved to have better results for when the intruder vehicle's motion transitions from the CT mode to the CV mode.

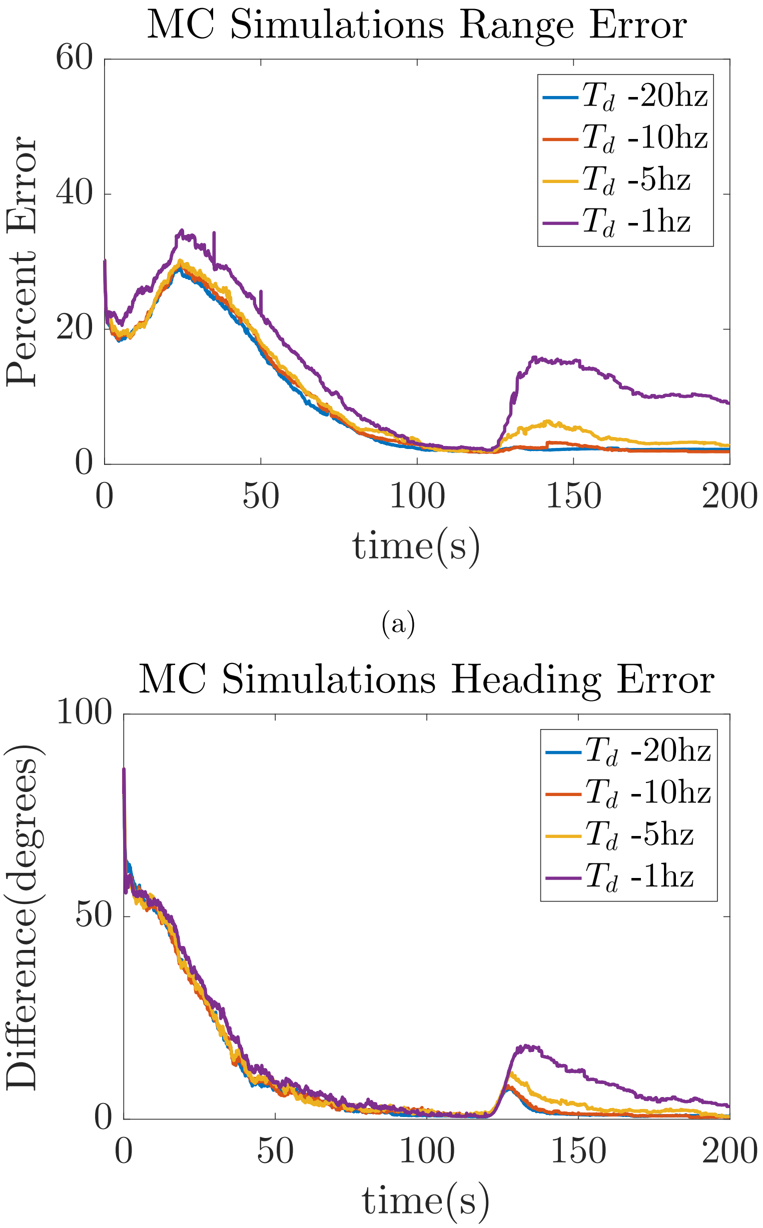


(a)

(b)

Figure 4.12: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the IMM algorithm, where δ = 1e-2

δ = 1e-3

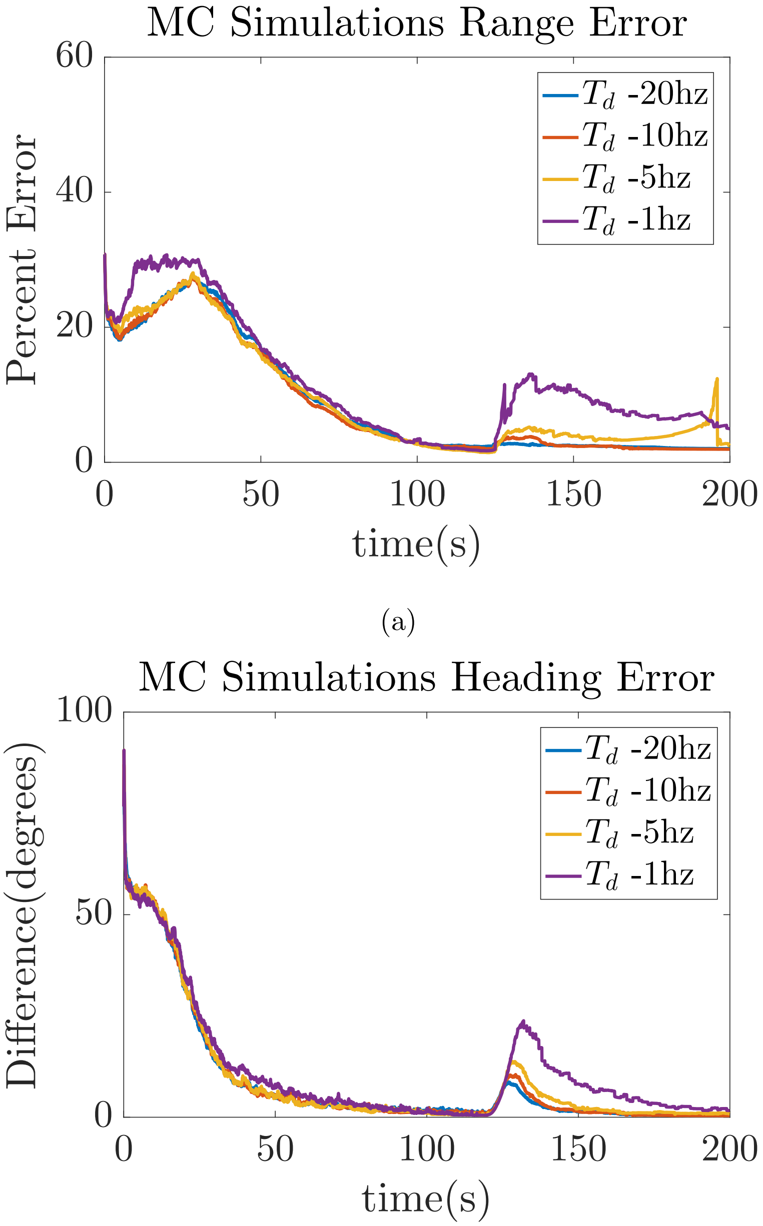


(a)

(b)

Figure 4.13: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the IMM algorithm, where δ = 1e-3

δ = 1e-4



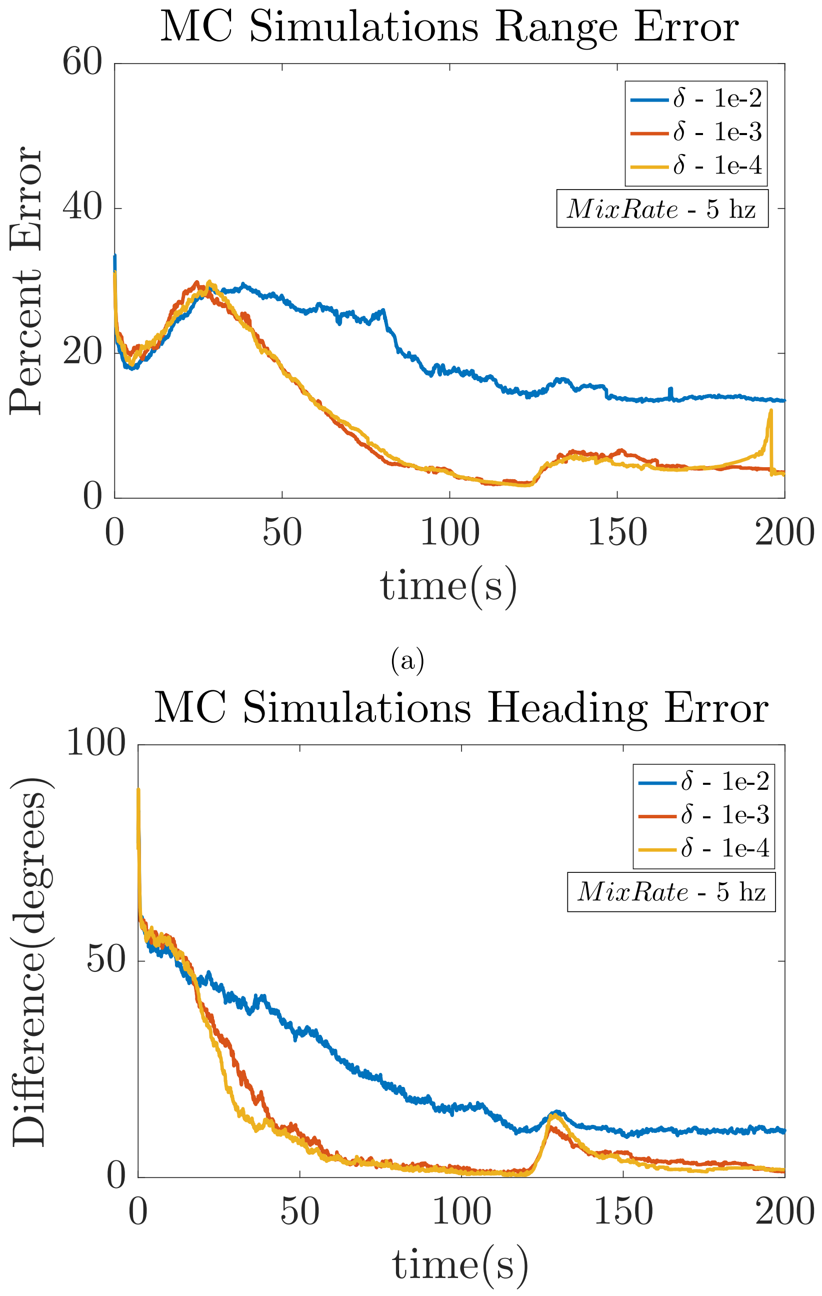
(a)

(b)

Figure 4.14: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the CPPV system model, where δ = 1e-4.

4.2.2 Study 2: Varying the off-diagonal value of the model transition matrix p_{ij}, δ

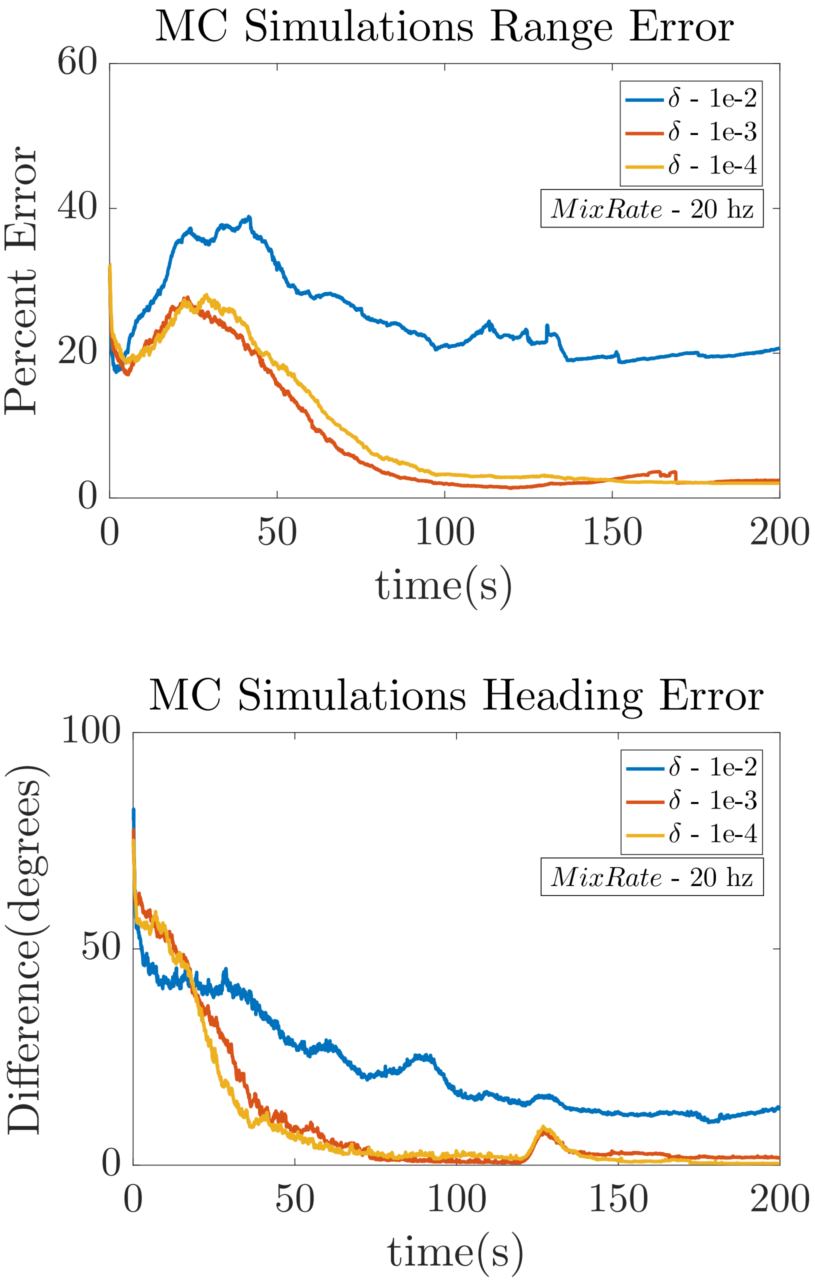
In this study, we keep the IMM mix rate, T_d, constant, and look at varying the off-diagonal value of model probability transition matrix p_{ij}, δ. Figures 4.15 and 4.16 shows similar results to the previous study, in that a δ value of 1e-2 gave poor performance. In Figure 4.15, a δ value of 1e-4 had similar a faster convergence for the heading state than a δ value of 1e-3. In Figure 4.16, δ = 1e-3 had better range results from t=0sec to t=150sec than δ = 1e-4. The heading error for δ = 1e-4 converged faster at t=25sec.



(a)

(b)

Figure 4.15: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the CPPV system model, where the mixing rate is held at a constant 5hz and three variations of δ are compared.



(a)

Figure 4.16: The RMSE over time for the range (a) and the absolute difference over time for the heading (b) state estimates using 500 Monte Carlo simulations for the CPPV system model, where the mixing rate is held at a constant 20hz and three variations of δ are compared

4.2.3 Study 3: Performance on Different Encounters

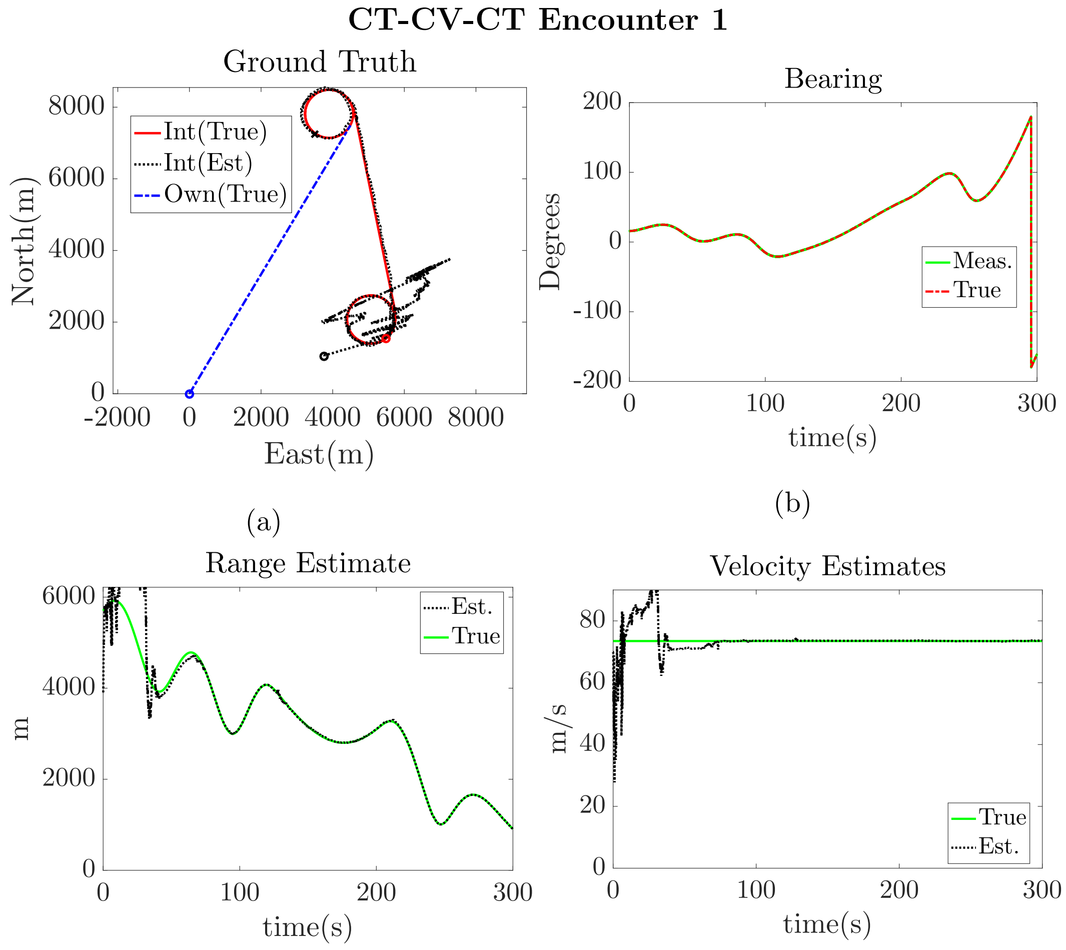
In this section, we show how the IMM filter performs for maneuvering intruder scenarios other than the one we utilized for the Monte Carlo simulations, CT-CV. We look at the following scenarios, CT-CV-CT, CV-CT, and CV-CT-CV. The purpose of this study is to determine the IMM's ability to adapt when the intruder transitions to a different maneuver. For all of the following scenarios, the filter bank parametrization is $N_r = 25$,$N_\omega = 1$,and $N_\theta = 6$, the IMM mix rate is 20 hz, and δ= 1e-4. For the CV-CT and CV-CT-CV scenarios, the CV filter was state estimates were set as the truth, with the covariance set given low values to ensure the IMM placed more probability for the CV mode initially. The estimates plotted are the maximum likelihood results. All noise assumptions are the same as found in Table 4.1, except for the angular velocity noise in the CV-CT and CV-CT-CV scenarios, which had a value of $\sigma^2_{\dot{\omega}}$=1e-4.

Figure 4.17 shows the plots for the CT-CV-CT scenario, where the intruder starts in an orbit, maneuvers to a constant velocity, then returns to an orbit. The estimates are fairly accurate for all states. In Figure 4.17g there is a spike in the mode probability at t=75sec. This spike represents the IMM placing higher likelihood for both model's estimates.

Figure 4.18 shows the plots for another CT-CV-CT scenario, but where the angular velocity is different for the two orbits. In Figure 4.18f, The angular velocity estimate that converged from the first orbit takes a long time to converge, coming closer to the true state at t=260sec. This slow convergence causes error in the mode probability until the angular velocity estimate gets closer to the true value.

Figure 4.19 covers the results for a CV-CT scenario. Here the EKF has been initialized near the true state and given fairly small covariance values to ensure the IMM has more belief in the CV model's filter than the CT model's filter. The angular velocity estimate is fairly noisy after convergence at t=90sec, which is due to the higher angular velocity noise we introduce for this scenario. Given the angular velocity noise of σ_ω^2=5e-6, the filter was unable to converge at t=90sec, only getting close to the true angular velocity near the end of the simulation.

Figure 4.20 shows our results for a CV-CT-CV scenario. In this scenario, the intruder is following a straight trajectory, follows an orbit for half of a circle, then returns to a straight trajectory. Here the more interesting plot is of the angular velocity estimate. The CV model's filter is able to converge at t=80sec fairly quickly. This plot shows that the IMM algorithm could have problems tracking maneuvers that end too quickly.

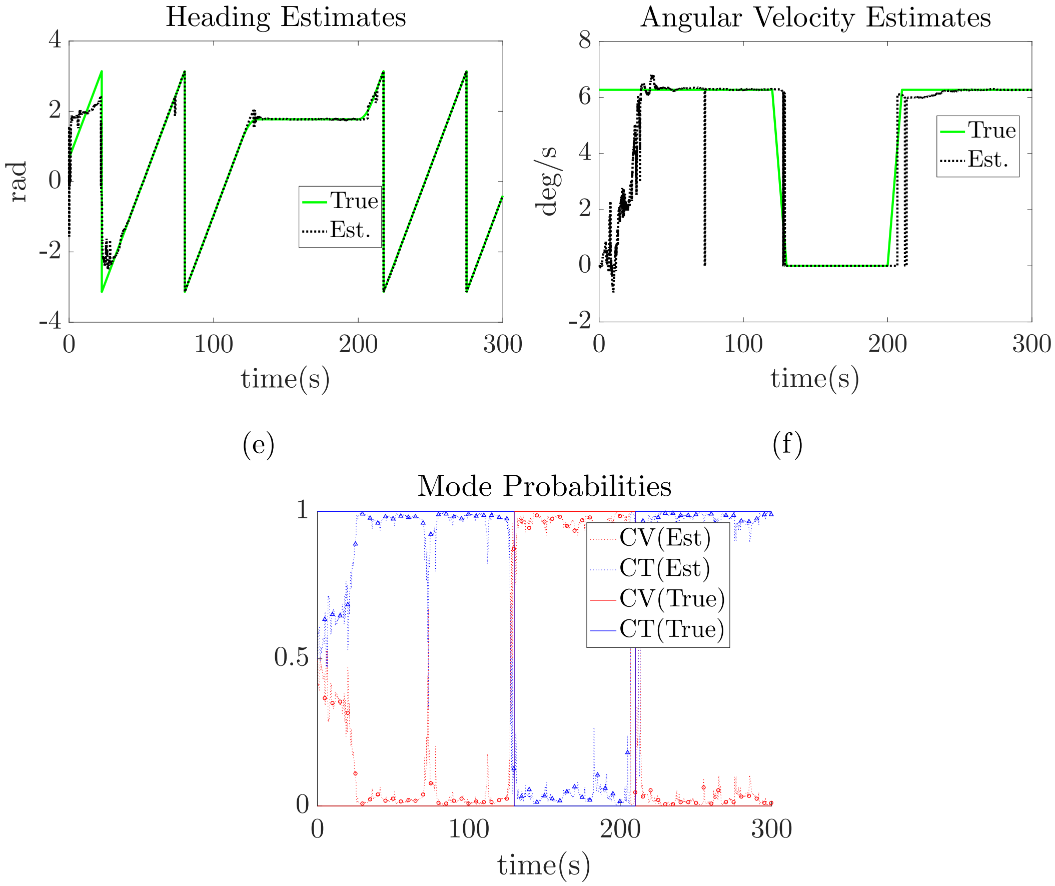


(a)

(b)

(c)

(d)



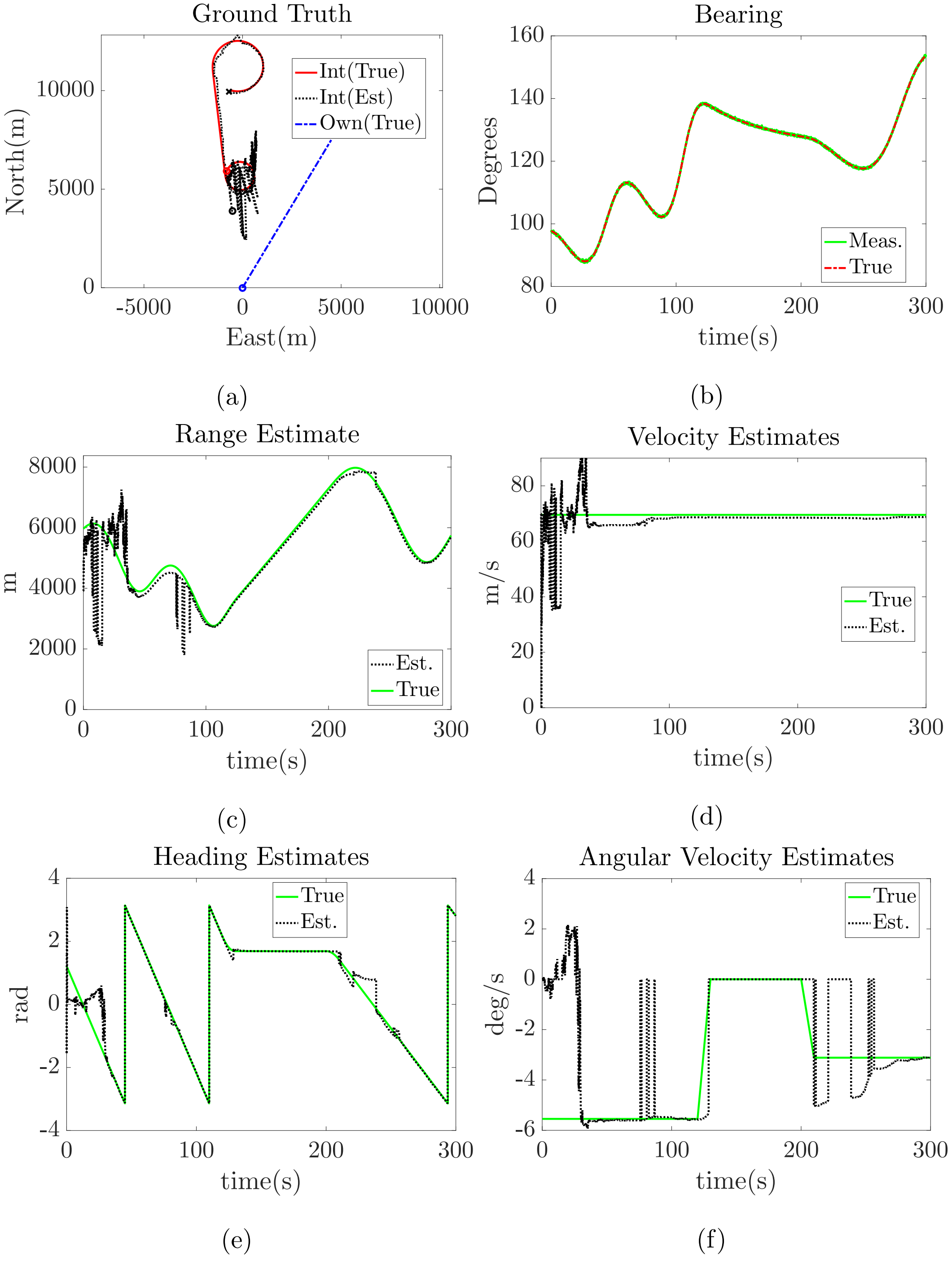
(e)

(f)

(g)

Figure 4.17: The performance of the IMM algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for an encounter where the intruder conducts an orbit, followed by a constant velocity, and then returning to an orbit. The mode probability over time (g) is shown as well.

CT-CV-CT Encounter 2



(a)

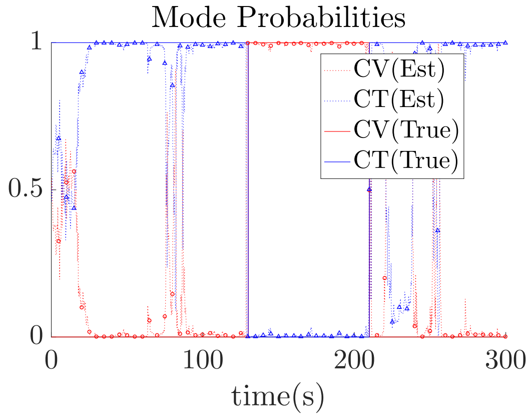
(b)

(c)

(d)

(e)

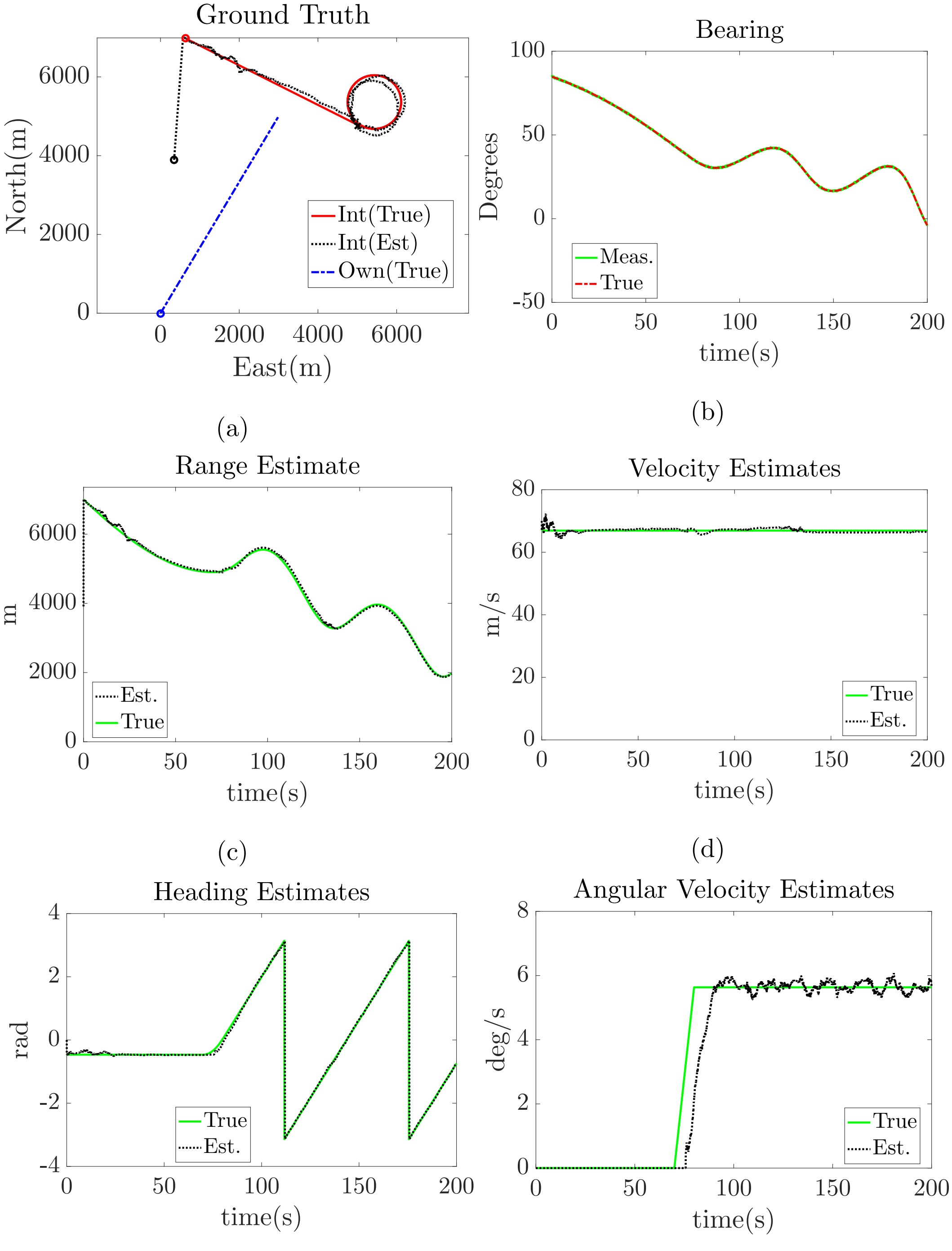
(f)



(g)

Figure 4.18: The performance of the IMM algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for an encounter where the intruder conducts an orbit, followed by a constant velocity, and then returning to an orbit. The mode probability over time (g) is shown as well.

CV-CT Encounter



(a)

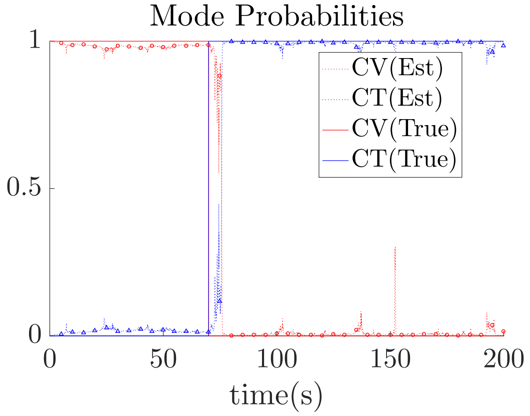
(b)

(c)

(d)

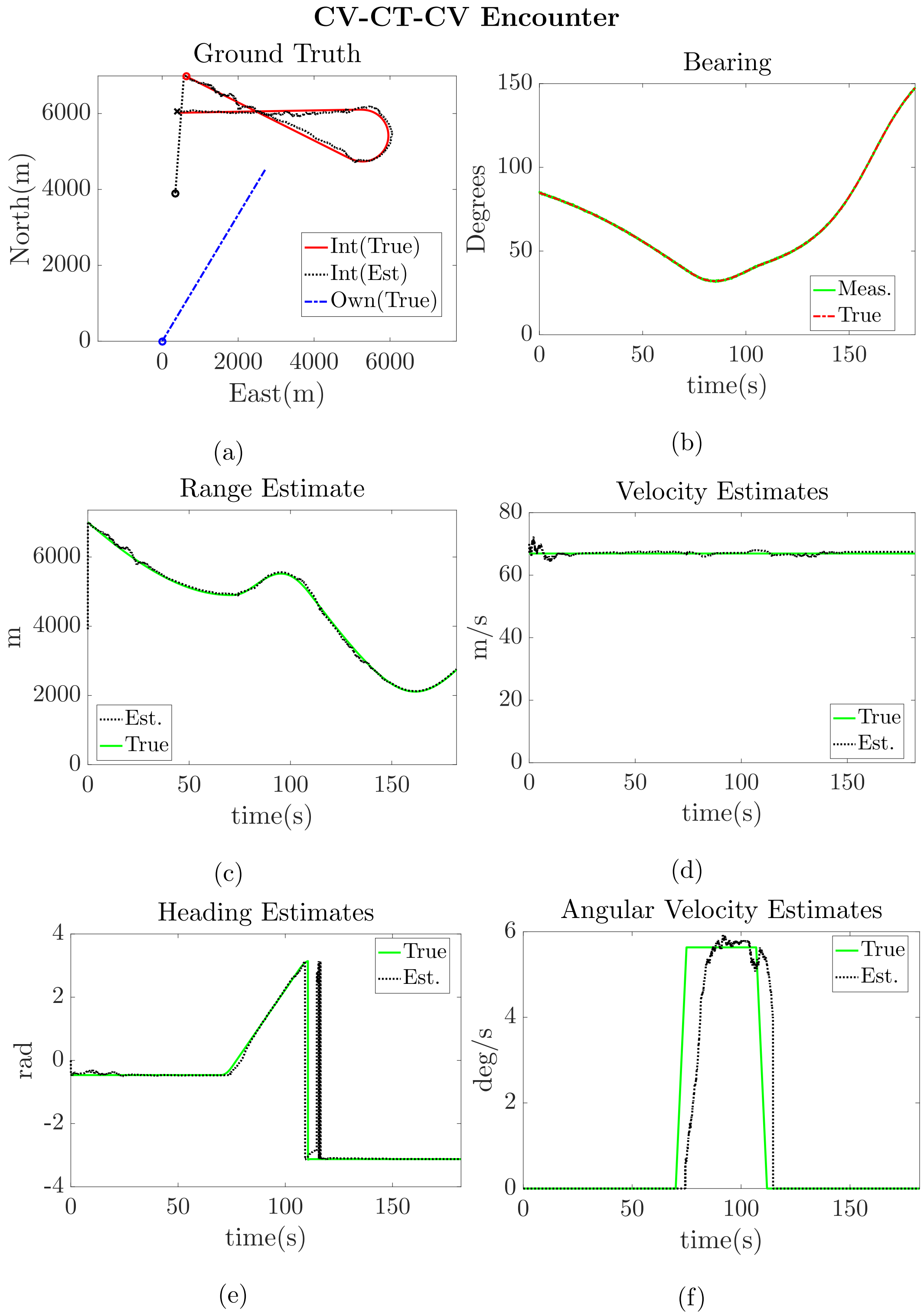
(e)

(f)



(g)

Figure 4.19: The performance of the IMM algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for an encounter where the intruder moves at a constant velocity, and then transitions to an orbit. The mode probability over time (g) is shown as well.


CV-CT-CV Encounter
Ground Truth
North(m)
East(m)
6000
4000
2000
0
0
2000
4000
6000
Int(True)
Int(Est)
Own(True)
Bearing
Degrees
time(s)
150
100
50
0
0
50
100
150
Meas.
True
Range Estimate
m
time(s)
6000
4000
2000
0
0
50
100
150
Est.
True
Velocity Estimates
m/s
time(s)
80
60
40
20
0
0
50
100
150
True
Est.
Heading Estimates
rad
time(s)
4
2
0
-2
-4
0
50
100
150
True
Est.
Angular Velocity Estimates
deg/s
time(s)
6
4
2
0
0
50
100
150
True
Est.

(a)

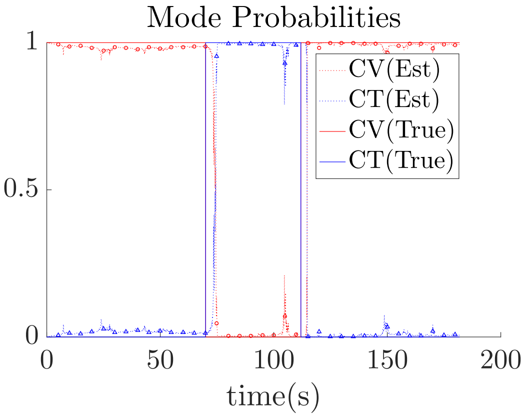
(b)

(c)

(d)

(e)

(f)



(g)

Figure 4.20: The performance of the IMM algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for an encounter where the intruder moves at a constant velocity, transitions to an orbit, then returns to move at a constant velocity. The mode probability over time (g) is shown as well.

CHAPTER 5

Flight Test Data Results

We utilized some of the flight test data that they collected to test our algorithm's performance on real data sets. One of UtopiaCompression's research goals is the maturation of a robust vision-based tracker. They conducted flight tests where a maneuvering GA aircraft Cessna 172G, as shown in Figure (5.1), would be tracked by a Group 3 FoxCar UAS as shown in Figure (5.2-5.3). More details on their flight test details can be found in [4]. In the following section, results of the filter's performance are shown.

Figure 5.1: A GA aircraft Cessna 172G was utilized for the collection of flight test data for algorithm by UtopiaCompression from [4]

5.1 Results

The Filter bank algorithm tested using real flight test data for multiple data sets. For all flight test (FT) scenarios, the ownship velocity is at a near constant velocity of

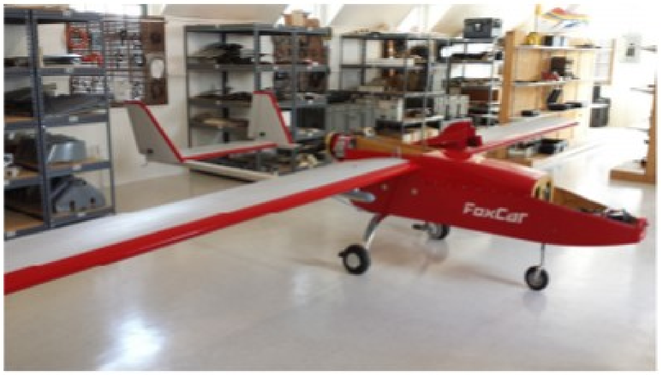


Figure 5.2: A Group 3 FoxCar UAS was utilized for collecting flight test data for algorithm by UtopiaCompression from [4]

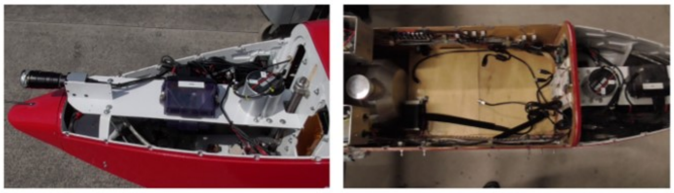


Figure 5.3: Overview of sensor payload mounted to UAS for collecting flight test data for algorithm by UtopiaCompression from [4]

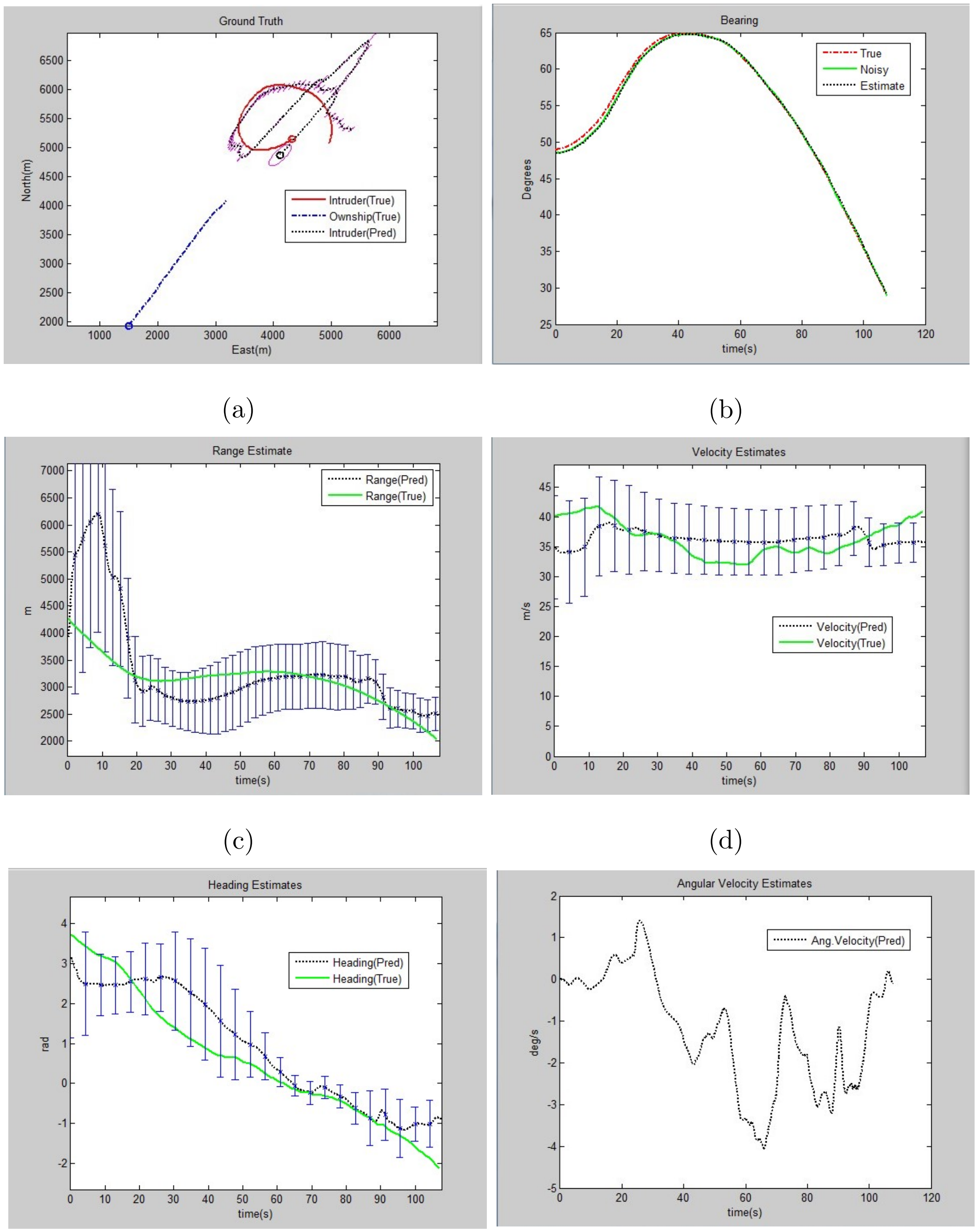
25m/s, with the intruder velocity at around 45m/s. For these plots, the vertical lines on the estimates represents a 1σ standard deviation. For FT 20 and 29, we conducted testing using the bearing angle data obtained from UtopiaCompression's computer vision-based algorithm. For FT 20 and 29, we found a bias of roughly 0.4° in the sensor data, which led to the underestimation of some measurements. Additionally, there are noisy measurements found in the sinusoidal peaks of the data. An outlier rejection was implemented to avoid noisy measurements. A standard filter parameter could not be made that maximizes performance for both scenarios. The primary parameter that was difficult to tune was the velocity variance. In order to reduce the range error, for FT20, a lower variance was required, whereas for FT29, a higher variance was required. To account for the non-constant angular velocity, the process noise was increased. The criteria for convergence was based on when the standard deviation of the heading reached a value lower than 30°. Using this, the convergence time for FT20 and FT29 were respectively 58s and 112s.

For FT 10 and 50, we had to simulate the bearing measurements due to errors during the online capture of data during these flight tests. We added a gaussian noise of 0.2° to the bearing measurements. For optimal performance, we used different parameterizations for each of these flight tests.

Table 5.1: FT 20 and 29 Parameters

$N_r = 10$		$N_\omega = 4$	$N_h = 3$
$r_{max} = 10km$	$r_{min} = 1km$	$\omega_{max} = 8deg/s$	$\omega_{min} = -8deg/s$
$\theta_{max} = 2\pi$	$\theta_{min} = 0$	$v_{initial} = 50m/s$	$\sigma_v^2 = 50^2/12$
$\sigma_v^2 = 0.5$		$\sigma_\omega^2 = 2.5e - 2$	$R = 0.1581deg$

Flight Test 20



(a)

(b)

(c)

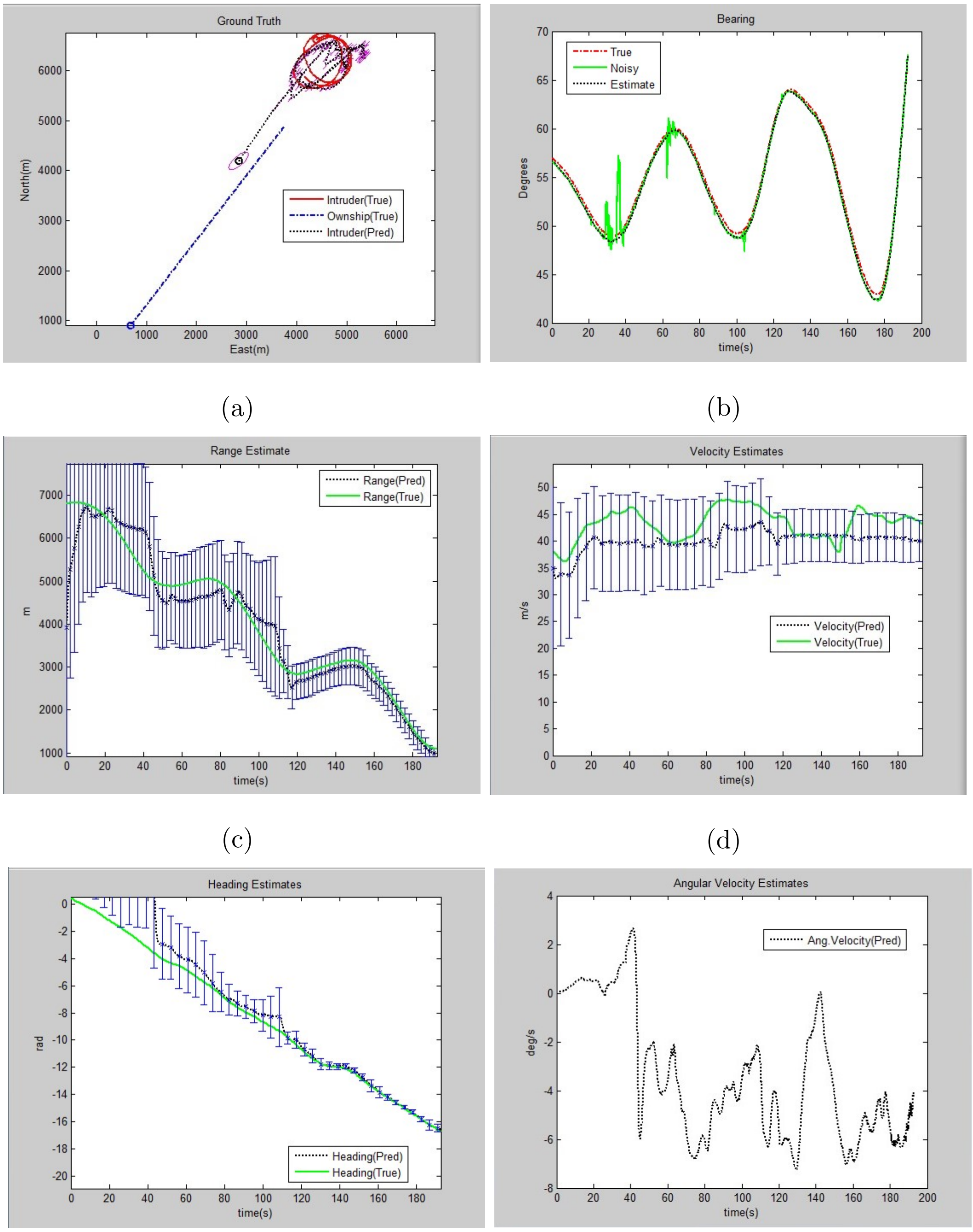
(d)

(e)

(f)

Figure 5.4: The performance of the filter bank algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for flight test 20 using the parameters from table 5.1.

Flight Test 29



(a)

(b)

(c)

(d)

(e)

(f)

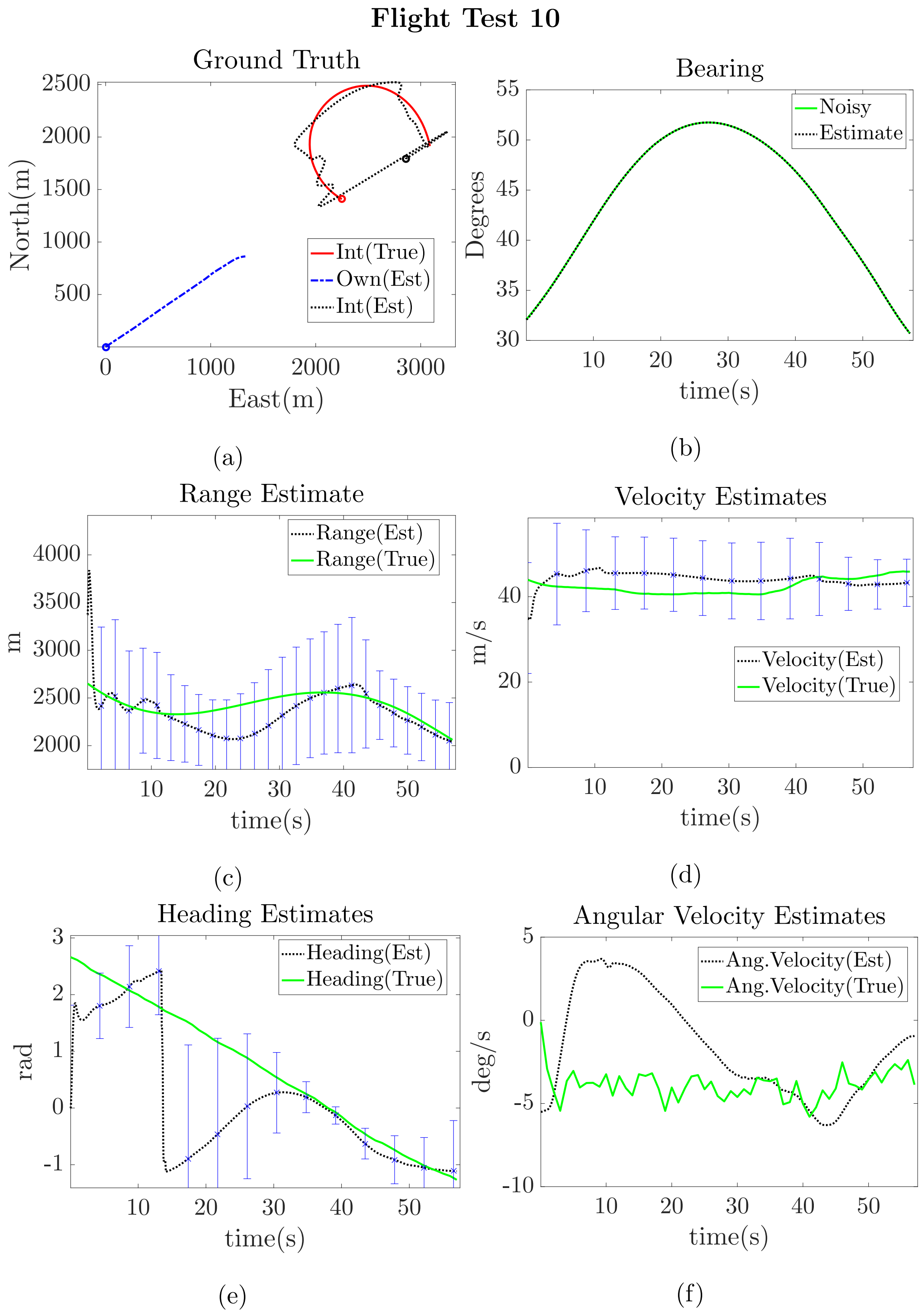
Figure 5.5: The performance of the filter bank algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for flight test 29 using the parameters from table 5.1.

Table 5.2: FT 10 Parameters

$N_r = 15$		$N_\omega = 4$	$N_h = 1$
$r_{max} = 8km$	$r_{min} = 1km$	$\omega_{max} = -10deg/s$	$\omega_{min} = -1deg/s$
$\theta_{max} = 2\pi$	$\theta_{min} = 0$	$v_{initial} = 35m/s$	$\sigma_v^2 = 45^2/12$
$\sigma_{\dot{v}}^2 = 5$		$\sigma_{\dot{\omega}}^2 = 5e - 2$	$R = 0.2deg$

Table 5.3: FT 50 Parameters

$N_r = 15$		$N_\omega = 4$	$N_h = 3$
$r_{max} = 12km$	$r_{min} = 1km$	$\omega_{max} = 10deg/s$	$\omega_{min} = 1deg/s$
$\theta_{max} = 2\pi$	$\theta_{min} = 0$	$v_{initial} = 35m/s$	$\sigma_v^2 = 45^2/12$
$\sigma_{\dot{v}}^2 = 1$		$\sigma_{\dot{\omega}}^2 = 5e - 2$	$R = 0.2deg$



(a)

(b)

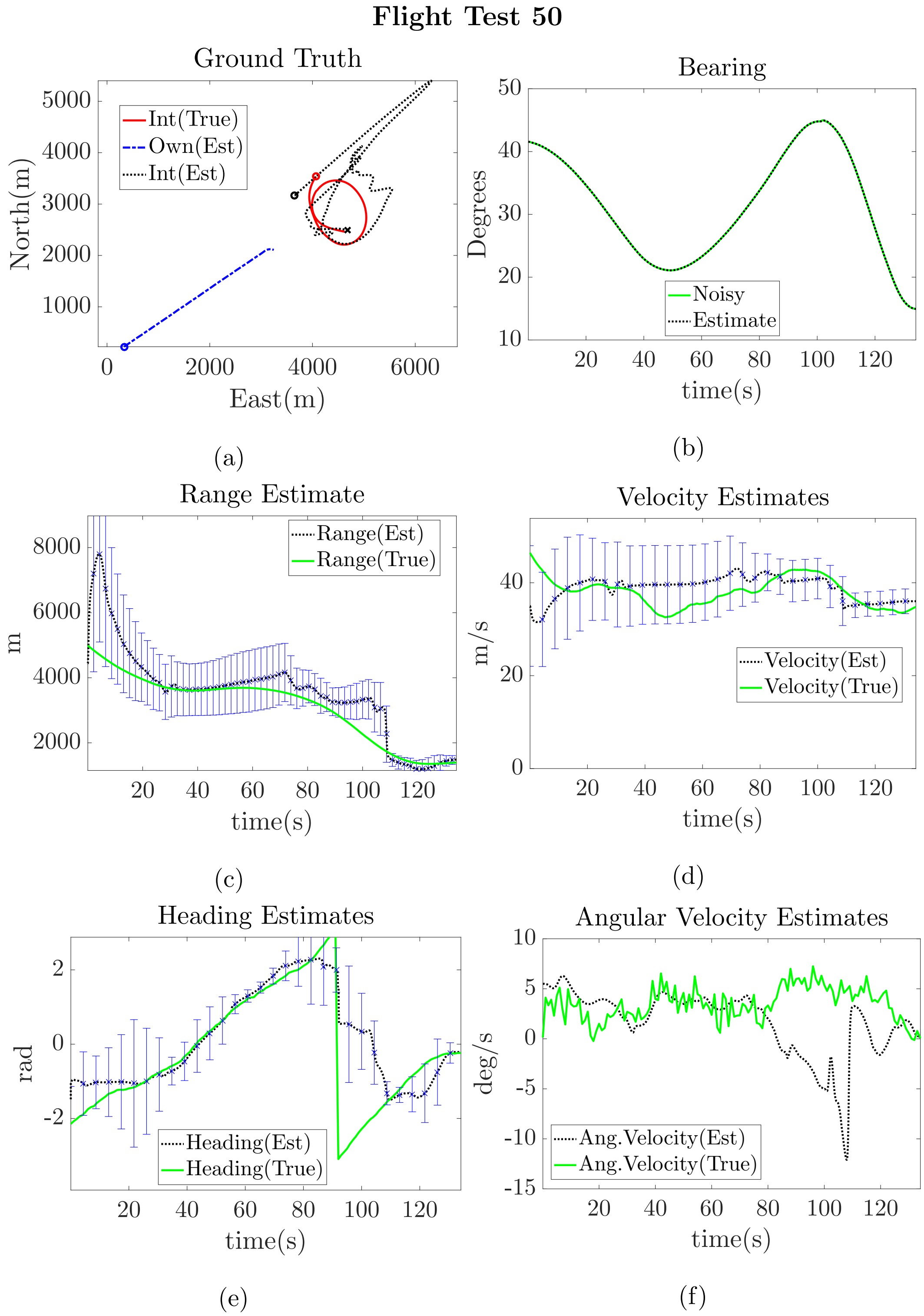
(c)

(d)

(e)

(f)

Figure 5.6: The performance of the filter bank algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for flight test 10 using the parameters from table 5.2.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.7: The performance of the filter bank algorithm is shown for the range(c), velocity(d), heading(e), and angular velocity (f) states for flight test 50 using the parameters from table 5.3.

CHAPTER 6

Conclusions

We conducted studies on the passive ranging problem of an orbiting intruder aircraft by an ownship aircraft moving on a straightline trajectory at a constant speed. We developed a filter bank algorithm, which was parameterized with respect to the range, heading, and angular velocity states. We conducted simulation studies on how to find the best way to optimize the filter bank algorithm. The results of this study were that the best performance was by increasing the number of intervals for the range and heading states. Increasing the number of intervals for the heading and angular velocity gave better results overall. Increasing the number of range intervals had a limit. With too many range intervals, the filter bank would believe the estimates more than the measurements, and lose accuracy when compared to lower range interval filter banks.

We also conducted a study comparing the filter bank algorithm's performance for the CPPV model and the MPC model. We found that the CPPV had less divergent scenarios, and had better estimates for the range states. The MPC model had better heading estimates than the CPPV model, and did converge faster. Both models exhibited robustness to ownship velocity noise.

We worked on developing the IMM algorithm for this problem as well. We utilized two modes, a constant velocity (CV) mode, and a coordinated turn (CT) mode. We adapted the filter bank aglorithm for the CT mode, and utilized a single EKF for the CV mode. We conducted a study on two parameters: the IMM mixing stage rate, and the model transition probability matrix. We find that the model transition probability plays a large effect on performance, and the off diagonal values must be set to fairly small values (δ <1e-2) to obtain good results.

Tests were conducted on real flight test data as well. We developed an outlier rejection to avoid measurements that were too noisy from the data. For each flight test, we focused on how to obtain the best estimates. To this effect, we used different filter bank parameterizations. We found promising results for some of the scenarios where the intruder did not maneuver too quickly.

6.1 Future Work

Given more time to work on this research, initial effort would be put into testing the IMM algorithm on the flight test data. Next would be the testing of different filters, such as the particle flow filter and the Unscented Kalman filter to compare the difference in their performance to what has been presented.

BIBLIOGRAPHY

[1] X. Yu and Y. Zhang, "Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects," *Progress in Aerospace Sciences*, vol. 74, pp. 152–166, 2015.

[2] Y. Gibbs, *Figure of Unmanned Aircraft Systems Integration into the National Airspace System from NASA Website*, 2017 (accessed Oct. 15, 2017). https://www.nasa.gov/centers/armstrong/news/FactSheets/FS-075-DFRC.html.

[3] G. A. Watson and W. D. Blair, "Imm algorithm for tracking targets that maneuver through coordinated turns," in *Aerospace Sensing*, pp. 236–247, International Society for Optics and Photonics, 1992.

[4] H. Bai, V. Venkataraman, S. Avadhanam, J. Newton, and J. Yadegar, "Developing a vision-based sense-and-avoid system," 2015.

[5] S. G. Gupta, M. M. Ghonge, and P. Jawandhiya, "Review of unmanned aircraft system (uas)," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 4, pp. pp–1646, 2013.

[6] S. B. Hottman, K. Hansen, and M. Berry, "Literature review on detect, sense, and avoid technology for unmanned aircraft systems," 2009.

[7] FAA, "Introduction to tcas ii version 7.1," 2011.

[8] L. R. Sahawneh, M. O. Duffield, R. W. Beard, and T. W. McLain, "Detect and avoid for small unmanned aircraft systems using ads-b," *Air Traffic Control Quarterly*, vol. 23, no. 2/3, pp. 203–240, 2015.

[9] M. C. Consiglio, J. P. Chamberlain, C. A. Munoz, and K. D. Hoffler, "Concepts of integration for uas operations in the nas," 2012.

[10] L. Purton, H. Abbass, and S. Alam, "Identification of ads-b system vulnerabilities and threats," in *Australian Transport Research Forum, Canberra*, pp. 1–16, 2010.

[11] S. Ramasamy, R. Sabatini, and A. Gardi, "Towards a unified approach to co-operative and non-cooperative rpas detect-and-avoid," in *Fourth Australasian Unmanned Systems Conference*, vol. 201, 2014.

[12] "Figure of synthetic aperture radar (sar) from barnard microsystems website."

[13] "Figure of laser/light detection and ranging (lidar) from leddartech website."

[14] "Figure of electro-optical system from flir website."

[15] N. Peach, "Bearings-only tracking using a set of range-parameterised extended Kalman filters," in *IEEE Proc. - Control Theory Appl.*, vol. 142, pp. 73–80, Jan. 1995.

[16] V. Aidala and S. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283–294, 1983.

[17] F. Gustafsson and A. Isaksson, "Best Choice of Coordinate System for Tracking Coordinated Turns," in *IEEE Proc. of the 35th International Conf. on Decision and Control*, Dec. 1996.

[18] Z. Xu, A. Sheng, and Z. Guo, “The analysis of the unobservable criteria on the single platform bearings-only target tracking system,” in *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, pp. 1008–1013, IEEE, 2008.

[19] S. C. Nardone and V. J. Aidala, “Observability criteria for bearings-only target motion analysis,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 2, pp. 162–166, 1981.

[20] R. Grover and P. Y. Hwang, “Introduction to random signals and applied kalman filtering,” *Willey, New York*, 1992.

[21] V. J. Aidala, “Kalman filter behavior in bearings-only tracking applications,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 29–39, 1979.

[22] F. Gustafsson and R. Karlsson, “Recursive Bayesian Estimation: Bearings-only Applications,” in *IEE Proc. Radar Sonar Navig. IEE Proceedings - Radar, Sonar and Navigation 152.5*, 2005.

[23] M. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, “Bearings-only tracking of manoeuvring targets using particle filters,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, p. 562960, 2004.

[24] K. L. Bell and L. D. Stone, “Implementation of the homotopy particle filter in the jpda and map-pf multi-target tracking algorithms,” in *Information Fusion (FUSION), 2014 17th International Conference on*, pp. 1–8, IEEE, 2014.

[25] Y. Li and M. Coates, “Particle filtering with invertible particle flow,” *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4102–4116, 2016.

[26] M. S. Grewal, “Kalman filtering,” in *International Encyclopedia of Statistical Science*, pp. 705–708, Springer, 2011.

[27] A. C. Ozelci, “Tracking and estimation algorithms for bearings only measurements,” 2013.

[28] J. C. Butcher, *Numerical methods for ordinary differential equations.* John Wiley & Sons, 2016.

[29] B. La Scala and M. Morelande, “An analysis of the single sensor bearings-only tracking problem,” in *Information Fusion, 2008 11th International Conference on*, pp. 1–6, IEEE, 2008.

[30] B. Ristic and M. S. Arulampalam, “Tracking a manoeuvring target using angle-only measurements: algorithms and performance,” *Signal processing*, vol. 83, no. 6, pp. 1223–1238, 2003.

APPENDIX A

Derivation of the Modified Polar Coordinates

For the derivation of the system dynamics in modified polar coordinates, It is necessary to derive functions from their cartesian equivalents. The following variables will be utilized for the state space variables:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} \dot{\beta} \\ \dot{r}/r \\ \beta \\ 1/r \\ \omega \end{bmatrix} \tag{A.1}$$

The variables are the bearing rate, range rate divided by range, bearing, reciprocal of range, and angular velocity respectively. Remember that for the range, $r = \sqrt{x^2 + y^2}$. Utilizing β with respect to the range yields its components the $x = rcos(\beta)$, and $x = rsin(\beta)$. When you take the dot derivative, you obtain: $\dot{r} = (x\dot{x} + y\dot{y})/r$. The following derivatives can be found as:

$$\frac{d}{dt}\left[\frac{1}{r}\right] = -\frac{\dot{r}}{r^2} = -\frac{1}{r}\frac{\dot{r}}{r} = -y_2 y_4 \tag{A.2a}$$

$$\frac{d}{dt}\left[w\right] = 0 \tag{A.2b}$$

$$\frac{d}{dt}\left[\beta\right] = \dot{\beta} = y_1 \tag{A.2c}$$

$$\frac{d}{dt}\left[\frac{\dot{r}}{r}\right] = \frac{\ddot{r}}{r} - \frac{\dot{r}^2}{r^2} \tag{A.2d}$$

$$\frac{d}{dt}\left[\dot{\beta}\right] = \ddot{\beta} \tag{A.2e}$$

To solve for A.2d and A.2e, it is necessary to find the relative velocities between the intruder and ownship in terms of cartesian coordinates. They are found as follows:

$$\begin{aligned} x = rcos(\beta) &\longrightarrow \dot{x} = \dot{r}cos(\beta) - rsin(\beta)\dot{\beta} \\ y = rsin(\beta) &\longrightarrow \dot{y} = \dot{r}sin(\beta) + rcos(\beta)\dot{\beta} \end{aligned} \tag{A.3}$$

Then $\ddot{r}$ is found as:

$$\begin{aligned} \ddot{r} &= \frac{\dot{x}^2 + \dot{y}^2 + x\ddot{x} + y\ddot{y}}{r} - \frac{\dot{r}^2}{r} \\ &= \frac{\dot{x}^2 + \dot{y}^2}{r} + \frac{x\ddot{x} + y\ddot{y}}{r} - \frac{\dot{r}^2}{r} \\ \frac{\dot{x}^2 + \dot{y}^2}{r} &= r\dot{\beta}^2 + \frac{\dot{r}^2}{r} \\ \ddot{r} &= r\dot{\beta}^2 + \ddot{x}cos(\beta) + \ddot{y}sin(\beta) \\ &= \frac{y_1^2}{y_4} + \ddot{x}cos(y_3) + \ddot{y}sin(y_3) \qquad (A.4) \end{aligned}$$

From this, the equation A.2d is found to be:

$$\frac{d}{dt}\left[\frac{\dot{r}}{r}\right] = y_1^2 - y_2^2 + y_4^2\left[\ddot{x}cos(y_3) + \ddot{y}sin(y_3)\right] \tag{A.5}$$

To find $\ddot{\beta}$, it is necessary to derive the relative acceleration in cartesian as follows:

$$\ddot{x} = \ddot{r}cos(\beta) - 2\dot{r}sin(\beta)\dot{\beta} - rcos(\beta)\dot{\beta}^2 - rsin(\beta)\ddot{\beta}$$

After a substitution of $\ddot{r}$ and $\dot{r}$, this can be rewritten as:

$$\begin{aligned} \ddot{\beta} &= \frac{1}{r}\left[\ddot{y} - \ddot{x}sin(\beta) - 2\dot{r}\dot{\beta}\right] \\ &= y_4\left[\ddot{y} - \ddot{x}sin(y_3)\right] - 2y_1y_2 \qquad (A.6) \end{aligned}$$

Now the only unknowns left to be solved in the derived functions are the relative accelerations. For this particular bearings-only tracking scenario, ownship maintains a constant velocity with no maneuver, whereas the target is conducting a maneuver. This leads to the relation:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \ddot{x}_t - \cancel{\ddot{x}_o} \\ \ddot{y}_t - \cancel{\ddot{y}_o} \end{bmatrix} \tag{A.7}$$

The target's acceleration can also be written as:

$$\begin{aligned}
\frac{d}{dt}\begin{bmatrix}\dot{x}_t \\ \dot{y}_t\end{bmatrix} &= \frac{d}{dt}\begin{bmatrix}vcos(\theta) \\ vsin(\theta)\end{bmatrix} \\
&= \begin{bmatrix}-vsin(\theta)\omega + \cancel{\dot{v}cos(\theta)} \\ vcos(\theta)\omega - \cancel{\dot{v}sin(\theta)}\end{bmatrix} = \begin{bmatrix}-\dot{y}_t\omega \\ \dot{x}_t\omega\end{bmatrix} \qquad \text{(A.8)}
\end{aligned}$$

Where θ is the target's heading. The target's velocity can be determined through the relative velocity:

$$\begin{aligned}
x &= x_t - x_o = rcos(\beta) \\
x_t &= x_o + rcos(\beta) \\
\dot{x}_t &= \dot{x}_o + \dot{r}cos(\beta) - rsin(\beta)\dot{\beta} \\
&= \dot{x}_o + \frac{y_2}{y_4}cos(y_3) - \frac{y_1}{y_4}sin(y_3) \qquad \text{(A.9)} \\
y &= y_t - y_o = rsin(\beta) \\
y_t &= y_o + rsin(\beta) \\
\dot{y}_t &= \dot{y}_o + \dot{r}sin(\beta) + rcos(\beta)\dot{\beta} \\
&= \dot{y}_o + \frac{y_2}{y_4}sin(y_3) + \frac{y_1}{y_4}cos(y_3) \qquad \text{(A.10)}
\end{aligned}$$

Plugging in equations A.9 and A.10 into A.8 yields:

$$\begin{aligned}
\ddot{x} &= -\left[\dot{y}_o y_5 + \frac{y_2 y_5}{y_4}sin(y_3) + \frac{y_1 y_5}{y_4}cos(y_3)\right] \\
\ddot{y} &= \dot{x}_o y_5 + \frac{y_2 y_5}{y_4}cos(y_3) - \frac{y_1 y_5}{y_4}sin(y_3)
\end{aligned} \qquad \text{(A.11)}$$

With this relationship solved, the system model is thus:

$$F_Y = \begin{bmatrix} -2y_1y_2 + \dot{x}_o y_4 y_5 cos(y_3) + \dot{y}_o y_4 y_5 sin(y_3) + y_2 y_5 \\ y_1^2 - y_2^2 + \dot{x}_o y_4 y_5 sin(y_3) - \dot{y}_o y_4 y_5 cos(y_3) - y_1 y_5 \\ y_1 \\ -y_2 y_4 \\ 0 \end{bmatrix} \qquad \text{(A.12)}$$

The Jacobian for the system model can then be found as:

$$\begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial y_5} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_5}{\partial y_1} & \cdots & \frac{\partial f_5}{\partial y_5} \end{bmatrix} = J_Y$$

$$\begin{aligned}
&\frac{\partial f_1}{\partial y_1} = -2y_2, \frac{\partial f_1}{\partial y_2} = -2y_1 + y_5 \\
&\frac{\partial f_1}{\partial y_3} = \dot{x}_o y_4 y_5 sin(y_3) + \dot{y}_o y_4 y_5 cos(y_3) \\
&\frac{\partial f_1}{\partial y_4} = \dot{x}_o y_5 cos(y_3) + \dot{y}_o y_5 sin(y_3) \\
&\frac{\partial f_1}{\partial y_5} = \dot{x}_o y_4 cos(y_3) + \dot{y}_o y_4 sin(y_3) + y_2 \\
&\frac{\partial f_2}{\partial y_1} = 2y_1 - y_5, \frac{\partial f_2}{\partial y_2} = -2y_2 \\
&\frac{\partial f_2}{\partial y_3} = \dot{x}_o y_4 y_5 cos(y_3) + \dot{y}_o y_4 y_5 sin(y_3) \\
&\frac{\partial f_2}{\partial y_4} = \dot{x}_o y_5 sin(y_3) - \dot{y}_o y_5 cos(y_3) \\
&\frac{\partial f_2}{\partial y_5} = \dot{x}_o y_4 sin(y_3) - \dot{y}_o y_4 cos(y_3) - y_1 \\
&\frac{\partial f_3}{\partial y_1} = 1, \frac{\partial f_3}{\partial y_2} = \frac{\partial f_3}{\partial y_3} = \frac{\partial f_3}{\partial y_4} = \frac{\partial f_3}{\partial y_5} = 0 \\
&\frac{\partial f_4}{\partial y_2} = -y_4, \frac{\partial f_4}{\partial y_1} = \frac{\partial f_4}{\partial y_3} = \frac{\partial f_4}{\partial y_4} = \frac{\partial f_4}{\partial y_5} = 0 \\
&\frac{\partial f_5}{\partial y_1} = \frac{\partial f_5}{\partial y_2} = \frac{\partial f_5}{\partial y_3} = \frac{\partial f_5}{\partial y_4} = \frac{\partial f_5}{\partial y_5} = 0
\end{aligned}$$

The measurement equation in MSC is :

$$z_k = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{A.13}$$

APPENDIX B

Relation Between Modified Polar Coordinates and Cartesian Coordinates

The conversion between the two coordinate systems is defined here. It is useful for the purpose of comparing their performance. The modified polar coordinates (MPC) system model is defined as previously from eq.A.1:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} \dot{\beta} \\ \dot{r}/r \\ \beta \\ 1/r \\ \omega \end{bmatrix} \tag{B.1}$$

Whereas for the Cartesian Position, Polar Velocity (CPPV) system is defined as:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \\ v \\ h \\ \omega \end{bmatrix} \tag{B.2}$$

The conversion function $g_{MPC \to CPPV}$ maps the state from MPC to CPPV coordinates is as follows:

$$
\begin{aligned}
x_1 &= r_x = rcos(\beta) = \frac{1}{y_4} cos(y_3) \\
x_2 &= r_y = rsin(\beta) = \frac{1}{y_4} sin(y_3) \\
x_3 &= v = \dot{r} = \sqrt{\begin{aligned} &\left(\frac{y_2}{y_4}\right)^2 + \left(\frac{y_1}{y_4}\right)^2 + \dot{x_o}^2 + \dot{y_o}^2 \\ &+ 2\frac{y_1}{y_4}\left(cos(y_3)\dot{y_o} - sin(y_3)\dot{x_o}\right) + 2\frac{y_2}{y_4}\left(cos(y_3)\dot{x}_o + sin(y_3)\dot{y_o}\right) \end{aligned}} \\
x_4 &= h = tan^{-1}\left(\frac{\frac{y_2}{y_4} sin(y_3) + \frac{y_1}{y_4} cos(y_3) + \dot{y}_o}{\frac{y_2}{y_4} cos(y_3) - \frac{y_1}{y_4} sin(y_3) + \dot{x}_o}\right) \\
x_5 &= \omega = y_5
\end{aligned}
\tag{B.3}
$$

Whereas the conversion function $g_{CPPV \to MPC}$ from CPPV to MPC systems is as follows:

$$
\begin{aligned}
y_1 &= \dot{\beta} = \frac{x_1(x_3 sin(x_4) - \dot{y_o}) - x_2(x_3 cos(x_4) - \dot{x_o})}{x_1^2 + x_2^2} \\
y_2 &= \frac{\dot{r}}{r} = \frac{x_1(x_3 cos(x_4) - \dot{x_o}) + x_2(x_3 sin(x_4) - \dot{y_o})}{x_1^2 + x_2^2} \\
y_3 &= \beta = tan^{-1}\left(\frac{x_2}{x_1}\right) \\
y_4 &= \frac{1}{r} = \frac{1}{\sqrt{x_1^2 + x_2^2}} \\
y_5 &= \omega = x_5
\end{aligned}
\tag{B.4}
$$

APPENDIX C

Jacobian of CPPV System Model

The Jacobians for the prediction and measurement models for the prediction step are as follows:

$$F_k = \begin{bmatrix} 1 & 0 & \frac{2}{w} sin(\frac{\omega T}{2}) cos(h + \frac{\omega T}{2} & -\frac{2v}{\omega} sin(\frac{\omega T}{2} sin(h + \frac{\omega T}{2}) & c_1 \\ 0 & 1 & \frac{2}{\omega} sin(h + \frac{\omega T}{2}) & \frac{2v}{\omega} sin(\frac{\omega T}{2} cos(h + \frac{\omega T}{2}) & c_2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.1}$$

$$c_1 = -\frac{2v}{\omega^2} sin(\frac{\omega T}{2}) cos(h + \frac{\omega T}{2}) + \frac{vT}{\omega} cos(h + \omega T)$$

$$c_2 = -\frac{2v}{\omega^2} sin(\frac{\omega T}{2}) sin(h + \frac{\omega T}{2}) + \frac{vT}{\omega} sin(h + \omega T)$$

$$H_k = \begin{bmatrix} -\frac{r_y}{r_x^2 + r_y^2} & \frac{r_x}{r_x^2 + r_y^2} & 0 & 0 & 0 \end{bmatrix} \tag{C.2}$$

APPENDIX D

Cramer-Rao Lower Bound

We approach solving the Cramer-Rao Lower bound as found in [30]. The Fisher information matrix can be initialized as:

$$J_0 = [P_0]^{-1}. \tag{D.1}$$

The information matrix can be recursively calculated (assuming an absence of process noise) using the following:

$$J_{k+1} = [F_k J_k^{-1} F_k^T]^{-1} + [H_{k+1}^T R_{k+1} H_{k+1}]^{-1}. \tag{D.2}$$

A comparison is made utilizing the error covariance matrix C_k, which should be bounded by J_k as:

$$C_k = E\{(\hat{x}_k - x_k)(\hat{x}_k - x_k)^T\} \geq J_k^{-1} \tag{D.3}$$

APPENDIX E

Derivation for Converting the Variance of Range to Inverse Range

The variance for the inverse range can be related to the variance of range in a probabilistic sense. For an arbitrary random variable X, the variance can be found from the following relation:

$$\sigma_x^2 = Var(X) = E\left[(X-\mu)^2\right] \tag{E.1}$$

In terms of the inverse range, this is:

$$\sigma^2_{\frac{1}{r}} = Var\left(\frac{1}{r}\right) = E\left[\left(\frac{1}{r} - \frac{1}{\bar{r}}\right)^2\right]$$

This can be formed into:

$$\begin{aligned} Var\left(\frac{1}{r}\right) &= E\left[\left(\frac{(\bar{r}-r)}{r\bar{r}}\right)^2\right] = \frac{1}{\bar{r}^2}E\left[\left(\frac{\bar{r}-r}{r}\right)^2\right] \\ &= \frac{1}{\bar{r}^2}E\left(\frac{\bar{r}}{r}-1\right)^2 = \frac{1}{\bar{r}^2}\left(E\left(\frac{\bar{r}^2}{r^2}\right) + E\left(-2\frac{\bar{r}}{r}\right) + 1\right) \end{aligned} \tag{E.2}$$

The variance can be approximated based on a Taylor series expansion, where $\delta = r - \bar{r}$ and $E(\delta)^2$ is to be found. The mean of delta is expected to be zero, $E(\delta) = 0$. This is for the case where there is little difference between the estimated range and the true range. The first term is found as:

$$\begin{aligned} E\left(\frac{\bar{r}}{r}\right)^2 &= E\left(\frac{\bar{r}}{\bar{r}+r-\bar{r}}\right)^2 = E\left(\frac{\bar{r}}{\bar{r}+\delta}\right)^2 = E\left(\frac{\bar{r}}{r} - \frac{\bar{r}}{(\bar{r}+\delta)^2}\Big|_{\delta=0}\delta + \dots\right)^2 \\ &= E\left(1 - \frac{1}{\bar{r}}\delta + \dots\right)^2 = E(1) - \cancel{2E\left(\frac{1}{\bar{r}}\delta\right)} + E\left(\frac{1}{\bar{r}^2}\delta^2\right) + h.o.t. \\ &= 1 + \frac{1}{\bar{r}^2}E(\delta^2) \end{aligned} \tag{E.3}$$

The second term is derived in the same way, yielding a value of -2. Combining the terms into (E.2) yields:

$$\begin{aligned} Var\left(\frac{1}{r}\right) &= \frac{1}{\bar{r}^2}\left(1+\frac{1}{\bar{r}^2}E(\delta^2)-2+1\right) = \frac{1}{\bar{r}^4}E(\delta^2) = \frac{1}{\bar{r}^4}E(r-\bar{r})^2 \\ \frac{1}{\bar{r}^2}\sigma_r &= \sigma_{\frac{1}{r}} \end{aligned} \tag{E.4}$$

Then, σ_β is the measurement noise standard deviation. The standard deviation for the other elements can be found similarly as [15]:

$$\sigma_{\dot{\beta}} = \frac{\sigma_v}{\bar{r}}, \sigma_{\frac{\dot{r}}{r}} = \frac{\sigma_v}{\bar{r}}, \sigma_{\frac{1}{r}} = \frac{\sigma_v}{\bar{r}^2},$$

APPENDIX F

Matlab Code

A copy of the code can be provided upon request by contacting the author at his e-mail:jordan.daugherty@okstate.edu

VITA

Jordan Daugherty

Candidate for the Degree of

Master of Science

Thesis: MANEUVERING INTRUDER PASSIVE RANGING FOR DETECT-AND-AVOID

Major Field: Mechanical Engineerng

Biographical:

Personal Data: Born in Joplin, Missouri, USA on August, 8th, 1991.

Education:

Received the B.S. degree from Oklahoma State University, Stillwater, Oklahoma, USA, 2014, in Mechanical Engineering.
Completed the requirements for the degree of Master of Engineering with a major in Mechanical Engineering at Oklahoma State University in April, 2018.

Publications:

Daugherty, Jordan, He Bai, and Sharath Avadhanam. ”Orbiting intruder passive ranging for small UAS Detect-and-Avoid.” Control Technology and Applications (CCTA), 2017 IEEE Conference on. IEEE, 2017.