Utilizing Hidden Markov Models to Classify Maneuvers and Improve Estimates of an Unmanned Aerial Vehicle During High Dynamic Flight

by

Amy Strong

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 7, 2021

Keywords: Hidden Markov Model, Unmanned Aerial Vehicle, Extended Kalman Filter

Approved by

Scott Martin, Chair, Assistant Research Professor of Mechanical Engineering
David Bevly, Bill and Lana McNair Professor of Mechanical Engineering
Bo Liu, Assistant Professor of Computer Science and Software Engineering

Abstract

Unmanned Aerial Vehicles (UAVs) are an increasing presence around the world; however, they can pose a threat to secure facilities. Many UAV mitigation techniques require accurate knowledge of UAV states to successfully intercept an adversarial UAV, but access to UAV on-board sensors may not be possible. One potential solution to this problem is to estimate UAV states using only radar measurements. This scenario is examined in simulation and with real world data. A discrete Extended Kalman Filter (EKF) with a constant acceleration dynamic model provides a baseline estimation performance of simulated UAV maneuvers and is shown to have consistent error in state estimates during high dynamic maneuvers. The simulated UAV maneuvers are then modelled as Hidden Markov Models (HMMs). HMMs are utilized to perform real time classification of maneuvers and to provide acceleration and jerk estimates of the UAV through the use of a Gaussian Mixture Regression. HMM classification of simulated maneuvers results in high accuracy classification during UAV flight. The HMM acceleration and jerk estimates are then incorporated into a state estimation framework as inputs to the filter's dynamic model. This new system is known as the EKF+HMM. When estimating high dynamic maneuvers, the EKF+HMM performs better than the baseline EKF, while performing at similar levels when estimating low dynamic maneuvers. HMM classification and the EKF+HMM are also tested on a real world data set of maneuvers performed by a Tarot X8 Octacopter. HMMs were trained for each maneuver, using experimental data or simulated data. HMM classification was successful using both types of HMMs, although models trained with experimental data performed better. The EKF+HMM was also tested on the real-world data set and performed worse than the EKF when using simulation data trained HMMs and at the same level as the EKF when using HMMs trained with experimental data.

Acknowledgments

I am so grateful to have had the opportunity to work and learn in the GAVLAB. I had the freedom to explore in my work, but also the guidance and support to help me learn and to prepare myself for the future. Thank you to Dr. Martin and Dr. Bevly for being amazing advisors.

Thank you to Dan Kamrath, who helped me with all of my experimental data collection. Thank you to Jacob Pryor who was always ready to answer any of my questions. Thank you to all the wonderful GAVLABers who I could not imagine grad school without: Matt Boler and Anderson Givhan, who are always up for a lively debate, Drew Jennings, who had total dedication to his Halloween costume, Will Bryan, an excellent friend and ping-pong player, Jake Ward, who is always down for Friday night trivia, John David Sprunger, resident coding and beard expert, and Scott Burchfield and Josh Wood, who live upstairs. Board game nights with you all were some of the best parts of my time in the GAVLAB.

Thank you to Brooke and Anna; our coffee dates, craft nights, and debates keep me sane. I'm so fortunate to have such smart, interesting, and empathetic friends. Thank you to Nikki. I could not have survived undergrad or grad school without you. I'm so glad I decided to go to Atlanta that day. Thank you to Brendan, who has constantly supported throughout this process and who I could not have done it without.

Thank you to my dad, who always has the best invention ideas for me to put my engineering degree to use. Your words of encouragement throughout this process have been invaluable. Thank you to my mom, for always encouraging my curiosity and love of learning – from letting me play with pipettes to telling me about your days at the lab, I've never doubted my place in the world of science and engineering because of you.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The presence of unmanned aerial vehicles (UAVs) is increasing throughout the United States. The Federal Aviation Administration has registered over 800,000 UAVs as of March 2021 [2]. While UAVs have many exciting commercial and recreational, they can also pose a threat to secure facilities. UAVs have been used to smuggle drugs across the border [3] and to surveil nuclear power plants without authorization [4]. Currently, countermeasures are being explored in the event of an adversarial UAV, which include nets, projectiles, and laser beams [5]. However, UAVs are highly dynamic and capable of performing evasive maneuvers that may make them unpredictable and difficult to track. Accurate state estimates of a UAV during these high dynamic maneuvers can provide invaluable information when attempting to enact countermeasures.

State estimation of a UAV is possible with on-board or off-board sensors. Typically, the problem is evaluated from the perspective of using on-board sensors. Kingston and Beard estimated position and attitude of a UAV through GPS and IMU measurements and a cascaded Extended Kalman Filter (EKF) [6]. Marantos et al. utilized an Adaptive Complementary Filter with GPS, IMU, barometer, and altimeter measurements to estimate UAV attitude, position, and velocity [7]. However, in some scenarios, such as an adversary UAV, on-board sensors are inaccessible. Instead sensors such as cameras, radar, or LiDAR must be used. For example, Hoffman et al. used Micro-Doppler radar measurements and an EKF to estimate the position and velocity of the UAV [8]. This thesis aims to implement an EKF that utilizes radar measurements and a constant acceleration dynamic model to estimate the position, velocity, and acceleration of a maneuvering UAV.

UAV maneuvers can also be modelled through data driven approaches, in which specific models are built for known UAV maneuvers. These models can be used to classify maneuvers as well as provide additional information about them. Martin et al. describes a trajectory as "a sequence of observed vectors in some measurement space" [9]. Many methodologies have been used to represent, recognize, and classify trajectories. When performing human gesture recognition, Martin et al. modelled gestures as "temporal trajectories of parameters" and used a multi-dimensional histogram to perform gesture recognition. Rao et al. modelled different human actions as a series of dynamic instances and intervals [10]. The dynamic instances were detected using the spatio-temporal curvature of the human action's trajectory. Human gesture recognition has also been performed using dynamic time warping to align and compare gestures [11].

Hidden Markov Models (HMMs) remain a popular method of modelling and classifying trajectories because "they offer dynamic time warping, a training algorithm and clear Bayesian semantics" [12]. HMMs have been utilized to classify time series movements. For example, HMMs were sued to classify American Sign Language (ASL) through camera data [13]. The hands of the person signing were modelled as 2D ellipses that changed in position, eccentricity, and orientation throughout the trajectory of the sign. Similarly, Brand et al. created coupled HMMs to model interacting processes and used these coupled HMMs to classify two-handed tai-chi gestures [12].

Bashir et al. (2005) and Bashir et al. (2007) aimed to represent and classify trajectories in a Euclidean space using not only position of the object, but also velocity [14] [15]. In Bashir et al. (2005), trajectories were modelled using Principle Component Analysis (PCA) and Gaussian Mixture Models (GMMs) [14]. However, in Bashir et al. (2007), it was found that trajectories modelled with PCA and HMMs described the temporal relationship between variables more accurately when compared to trajectories modelled with PCA and GMMs [15]. This thesis aims to model known UAV maneuvers with HMMs using the position, velocity, and acceleration states of the UAV. These models will also be used for trajectory generation, which will provide additional estimates to the EKF.

Trajectory generation is often explored in robotics applications – specifically, programming by demonstration, which uses examples of a task being performed (opening a door, pouring water in a cup, etc.) to generate a desired trajectory for a robot. The concept is also referred to as trajectory learning, which conceptually "consider[s] the demonstrations as the observations of one unique intended trajectory" [16].

Abbeel et al. worked to learn helicopter trajectories. Helicopter maneuvers were performed by a pilot, and then a trajectory that maximized the likelihood of the maneuver data and prior knowledge was generated through a Kalman smoother [17]. Similarly, Choi et al. learned the desired trajectory of a multirotor UAV using the Expectation-Maximization algorithm on the demonstrated trajectory [16]. Another method for trajectory generation is using Gaussian processes to model different states of a trajectory and then performing a Gaussian process regression to create the trajectory [18].

The works regarding trajectory generation most relevant to this thesis are [19], [20], and [21], as they use HMMs and GMMs for trajectory modelling and generation. Calinon and Billard (2005) decomposed gestures from a human demonstrator with PCA and then modelled the gestures using HMMs [19]. A trajectory is generated from an HMM by finding the HMM state sequence, using the mean values of each state's emission probability function to get a set of points, interpolating, and re-projecting from PCA to the original features. Calinon et al. (2006) again decomposed gestures with PCA. The decomposed gestures were then modelled as HMMs with GMMs used as the emission probability function. The HMMs classified different gestures and generated trajectories through a Gaussian Mixture Regression (GMR). The GMR was performed using time, t, as the input variable. Finally, Calinon et al. (2007) represented gestures purely as GMMs, where the state vector included time and the PCA decomposition of the trajectory's variables [21]. A GMR was performed about time, and once re-projected, the output was used as a robot desired trajectory.

This thesis aims to estimate the linear position, velocity, and acceleration of a UAV throughout dynamic maneuvers. An EKF that uses radar measurements and a constant acceleration dynamic model will be used to estimate UAV states; however, EKF estimates can be subpar during highly dynamic maneuvers. Past information about a maneuver (either commands or real world data) can be used to create an HMM, which can be used for classification and trajectory generation. This thesis will create an EKF+HMM, a system that utilizes maneuver classification information and estimates generated from an HMM to improve UAV state estimation during a maneuver.

1.1 Research Contributions

The contributions of this thesis are listed below.

- Modelled UAV maneuvers as HMMs. Created a classification scheme that used HMMs to provide real-time classification of maneuvers throughout estimation.
- Created HMM estimates of UAV acceleration and jerk using trajectory generation techniques.
- Combined and EKF and HMM to create an EKF+HMM, which utilizes HMM estimates to supplement original EKF estimates. Applied this method to the estimation of UAV maneuvers and compared it to baseline EKF performance.
- Applied HMM classification and EKF+HMM to both simulated and real world data of UAV maneuvers.

1.2 Thesis Outline

The remaining chapters of the thesis are organized as follows. Chapter 2 discusses UAV modelling, simulation, and maneuver generation, as well as basic EKF estimation of the UAV. Chapter 3 provides background on HMMs, discusses the process of modelling maneuvers as

HMMs, and then evaluates the classification and trajectory generation accuracy of these HMMs. Chapter 4 discusses the formulation of the EKF+HMM and compares results of the EKF and EKF+HMM when estimating simulated maneuvers. Real world execution of maneuvers with an octacopter is discussed in Chapter 5. Both real world data and simulated data are used for HMM creation. Both HMMs are used to classify real world maneuvers and to compare the EKF and EKF+HMM. Finally, conclusions and future work are presented in Chapter 6.

Chapter 2

Unmanned Aerial Vehicle Modelling, Simulation, and Estimation

Simulation provides a controlled environment to implement and test estimation algorithms. Through simulation, true states and disturbances are known perfectly, so error in state estimation can be fully evaluated. As such, much of this thesis relies on an Unmanned Aerial Vehicle (UAV) simulation as a way to generate test maneuvers and evaluate estimation algorithms. This chapter provides the basis for such simulations.

The scenario examined in this thesis involves estimating the states of a UAV without access to any on-board sensors and potentially without knowledge of the UAV beyond the maneuvers it could potentially perform. One method to estimate UAV motion in this situation is the use of an Extended Kalman Filter (EKF) and radar measurements to estimate linear position, velocity, and acceleration of the UAV. This method provides a point of comparison for the EKF+HMM described later.

The following sections will first discuss the UAV model and all of its components, such as coordinate systems and equations of motion. Then, the UAV maneuvers used in this thesis for EKF, HMM, and EKF+HMM testing are described and simulated. Finally, the EKF is explained and used to estimate UAV states during maneuvers to provide baseline filter performance.

2.1 UAV Model

2.1.1 Coordinate Frames

The two coordinate frames used for the UAV model are the body frame and the North East Down (NED) frame. Figure 2.1 shows the body frame, where the x_B-axis extends from

UAV center of mass to its top right rotor, the y_B-axis extends from center of mass to the bottom right rotor and the z_B-axis points down.

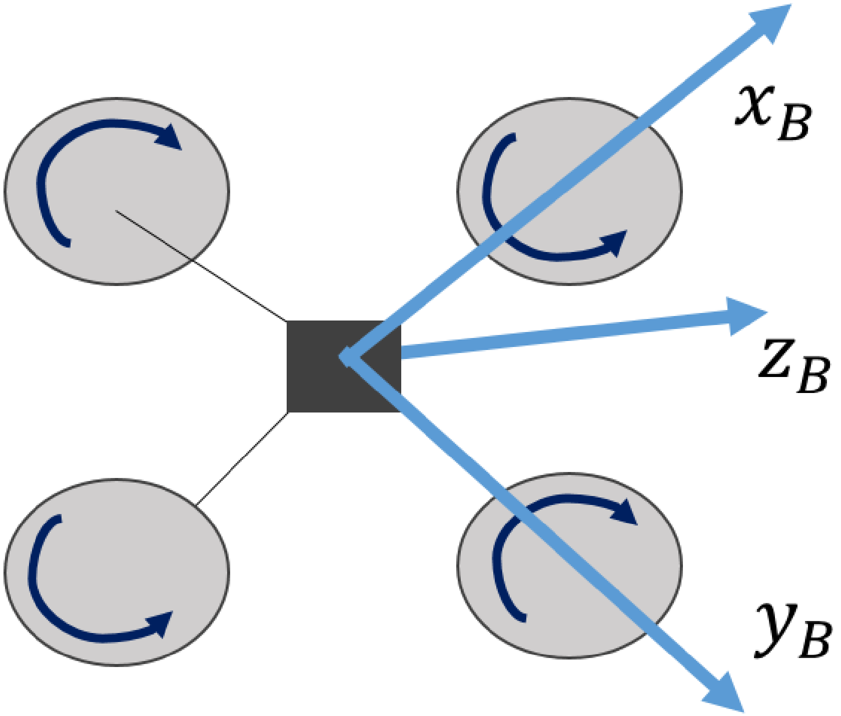


Figure 2.1: Body axes of a UAV

The NED frame and body frame are related through the Euler angles,

$$\boldsymbol{\eta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \tag{2.1}$$

where ϕ is roll, θ is pitch, and ψ is yaw.

A 1-2-3 Euler Angle sequence, also known as the Tait-Bryan angles, is used to rotate coordinates from the NED frame $(x - y - z)$ to the body frame $(x_B - y_B - z_B)$ [22]. The first rotation is about the x-axis,

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}, \tag{2.2}$$

the second rotation is about the y'-axis,

$$R_2(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{2.3}$$

and the third rotation is about the z''-axis,

$$R_3(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

The full rotation matrix is written as

$$R_{123}(\phi, \theta, \psi) = R_1(\phi)R_2(\theta)R_3(\psi). \tag{2.5}$$

When rotating from body frame back into NED frame, the transpose of the original rotation matrix is used,

$$R_{321}(\phi, \theta, \psi) = R_{123}(\phi, \theta, \psi)^T. \tag{2.6}$$

2.1.2 Quadcopter Model Description

The quadcopter states are position ($\mathbf{r}$), velocity ($\mathbf{v}$), and acceleration ($\mathbf{r}$), which are further expanded to

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \qquad \mathbf{a} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}. \tag{2.7}$$

Each state is specified to be in the NED coordinate frame. The parameters of the quadcopter, such as mass, moment of inertia, and lift constant are stated in [23].

The inputs to the UAV are

$$\mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \tag{2.8}$$

where U_1 is the thrust, U_2 is the torque inducing roll, U_3 is torque inducing pitch, and U_4 is torque inducing yaw. The torques are generated by creating a difference in the angular velocities of the UAV rotors [23]. Figure 2.2 shows the process of inducing roll (left), pitch (middle), and yaw (right). The bold arrows in the figure represent a higher rotor angular velocity, while a thin arrow represents a lower rotor angular velocity. U_2, U_3, and U_4 are used to reach commanded Euler angles. Because the UAV has four inputs, it can control four out of six degrees of freedom. In this thesis, $\mathbf{U}$ is used to control z, ϕ, θ, and ψ.

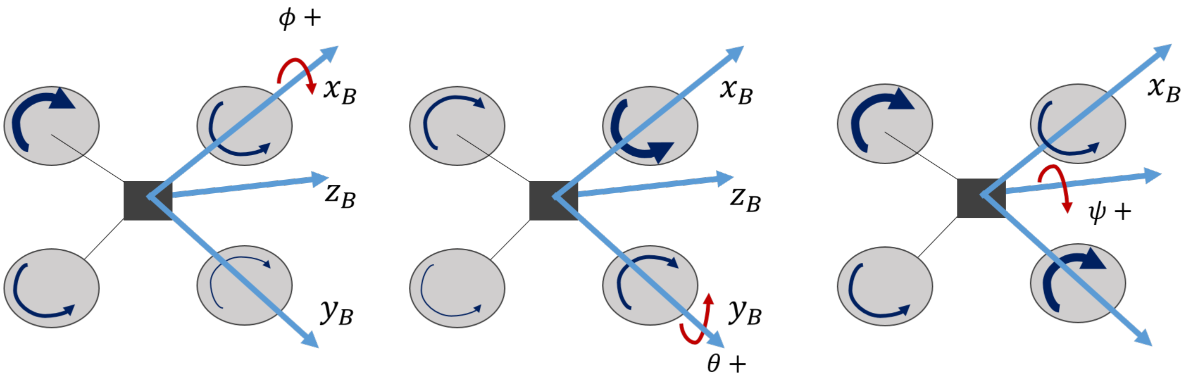


Figure 2.2: Changes in rotor angular velocities can create torques that induce roll (left), pitch (middle) or yaw (left) moments.

2.1.3 Equations of Motion

The forces acting on the UAV are force of gravity (F), air drag (D), and thrust of the UAV [23] [24]. More intricate models that include aspects of flight such as blade flapping or effects of angle of attack can be found in [25][26]. The Newton equation for forces acting on the UAV is

$$\sum \mathrm{F} = m\ddot{\mathbf{x}} = -\mathrm{D} + \mathrm{F}_B + \mathrm{T}. \tag{2.9}$$

Force of gravity is represented as,

$$\mathrm{F} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \tag{2.10}$$

where m is the mass of the UAV and g is gravity. The force of air resistance acts opposite the direction of inertial velocity and is directly proportional to it [24], as seen below,

$$\mathrm{D} = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \mathbf{v}. \tag{2.11}$$

Values for air resistance constants are found in [24]. Wind is not modelled in the equations of motion. It is assumed to be part a disturbance acting on the system.

The final force acting on the UAV is thrust, which is exerted in the body frame. Thrust can be found by summing the square of each rotor's angular velocity and multiplying it by the lift constant:

$$\mathrm{T}_B = \begin{bmatrix} 0 \\ 0 \\ k\ \sum_{i=1}^{4} \omega_i^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix}. \tag{2.12}$$

Thrust is converted to the NED frame through the use of a rotation matrix found in Equation (2.6),

$$\mathrm{T} = R_{321}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} = \begin{bmatrix} \sin(\phi)\sin(\psi) + \cos(\phi) cos(\psi)\sin(\theta) \\ \cos(\phi)\sin(\psi)\sin(\theta) - \cos(\psi)\sin(\phi) \\ \cos(\phi)\cos(\theta) \end{bmatrix} U_1. \tag{2.13}$$

Note that the Euler angles affect the directions in which thrust is enacted.

2.2 UAV Maneuver Simulation

In order to create high dynamic UAV maneuvers, a commanded thrust (T_B) and commanded Euler angles (ϕ, θ, and ψ) were created. Euler angles were limited to a magnitude of 50 degrees. Velocity was limited to a magnitude of 40 m/s and acceleration to a magnitude of 10 m/s^2 to remain within the dynamic constraints of a quadrotor [27].

The controller of the UAV was assumed to be perfect and therefore controlled to desired values without delay or error. Any effects of the controller were assumed to be negligible. Figure 2.3 shows the five UAV maneuvers simulated. Each maneuver is twenty-five seconds total and was created to be dynamically distinct from the others. Maneuvers 1, 2, 3, and 4 are later utilized to create HMMs, while Maneuver 5 is utilized as an "unknown" test trajectory.

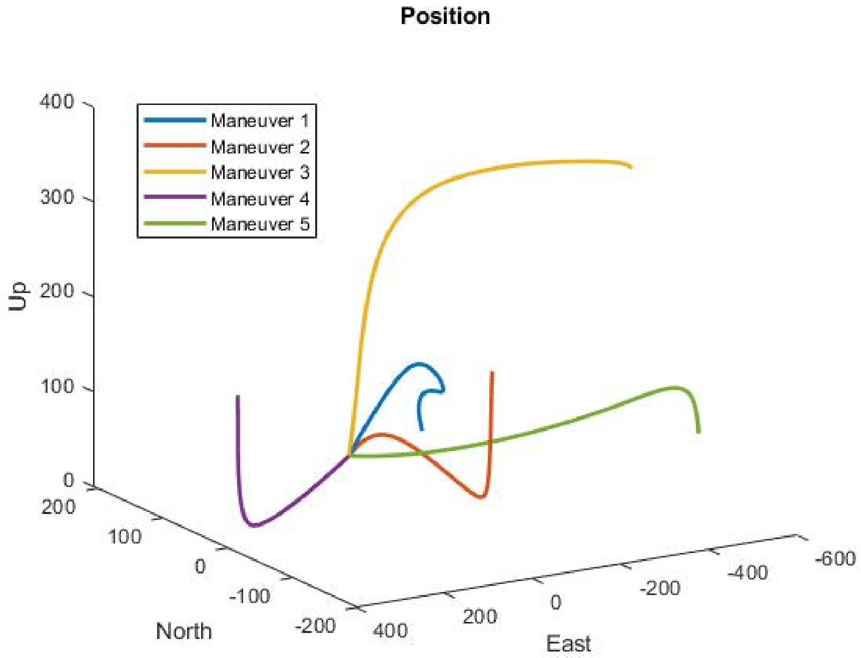


Figure 2.3: Simulated position of the five UAV dynamic maneuvers used within this thesis.

These five maneuvers do not represent all motions that a UAV could perform. Instead, they are a test case that provide examples of maneuvers that could be used for interception avoidance or to move towards a target. For real world implementation, common UAV maneuvers in relevant scenarios (urban flight, interception avoidance, trespassing into a sports stadium, etc.) would be observed. Then, training data for these maneuvers could be developed in simulation.

To create a data set for training HMMs, each maneuver was simulated 2300 times. The North and East start and end position for the UAV changed each time the maneuver was simulated, as well as the direction of flight. Additionally, process noise was added to the acceleration of the UAV to simulate disturbances in the environment, such as wind or blade flapping. Throughout the creation of the 2300 trajectory data set, the maneuver commands remain constant. The only changes to the maneuver are its position relative to an observer and the disturbances (or process noise) throughout flight. The following sections show an example flight of each maneuver with no additional process noise.

2.2.1 Maneuver 1

Figure 2.4 shows the position of the UAV while executing Maneuver 1. It begins at a a height of approximately 10 m and ends at a height of 55 m. Figure 2.5 shows the NED position, velocity, and acceleration of the UAV throughout the maneuver. The accelerations are oscillatory and at times rapidly change direction, which creates a large jerk.

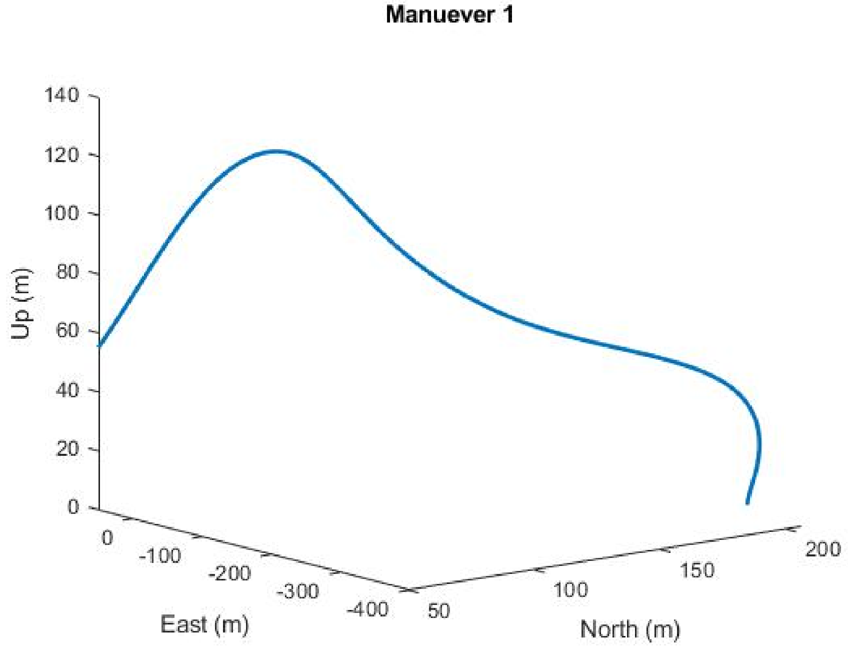
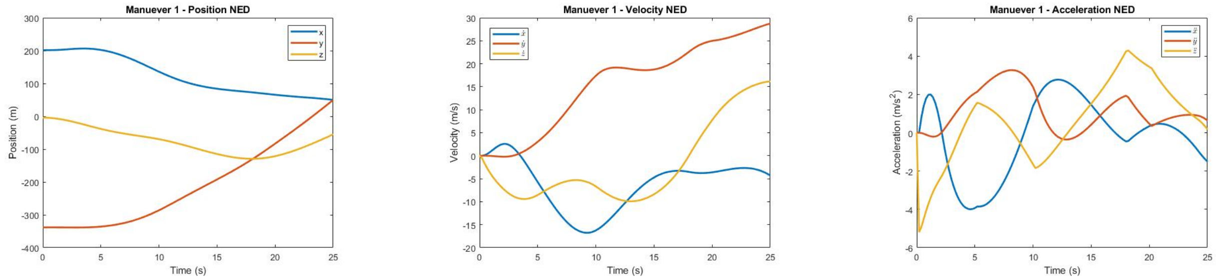


Figure 2.4: UAV position throughout Maneuver 1.



(a) UAV position over time (NED).

(b) UAV velocity over time (NED).

(c) UAV acceleration over time (NED).

Figure 2.5: The nine states of the UAV throughout Maneuver 1.

2.2.2 Maneuver 2

The position of the UAV while performing Maneuver 2 is shown in Figure 2.6. The UAV begins at a height of 180 m and descends to 55 m. The maneuver causes the UAV to circle the target position before reaching it. Figure 2.7 shows the position, velocity, and acceleration of the UAV throughout Maneuver 2. The majority of the dynamics take place in the last fifteen seconds of the maneuver. Before this point, only the vertical position of the UAV is rapidly changing.

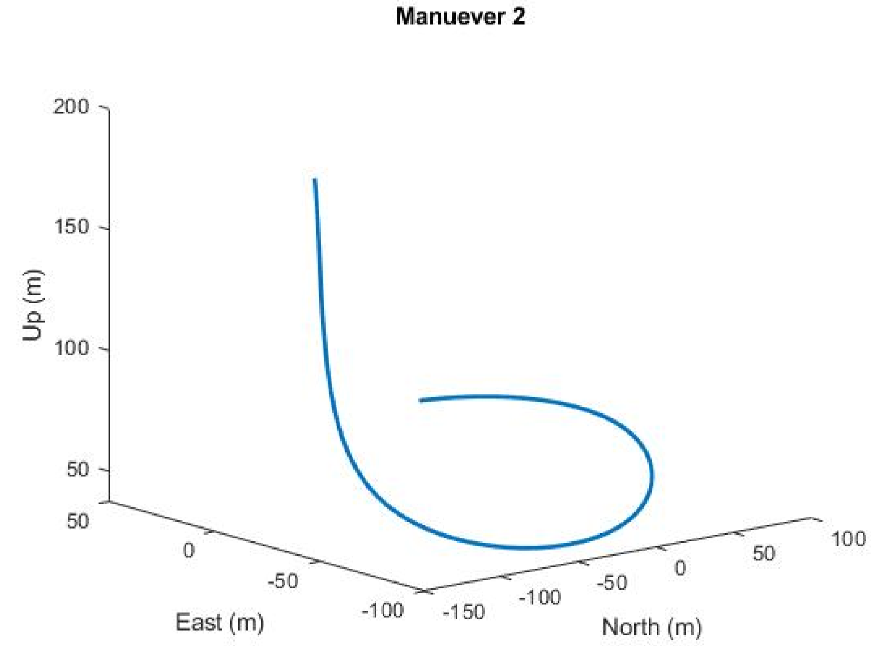
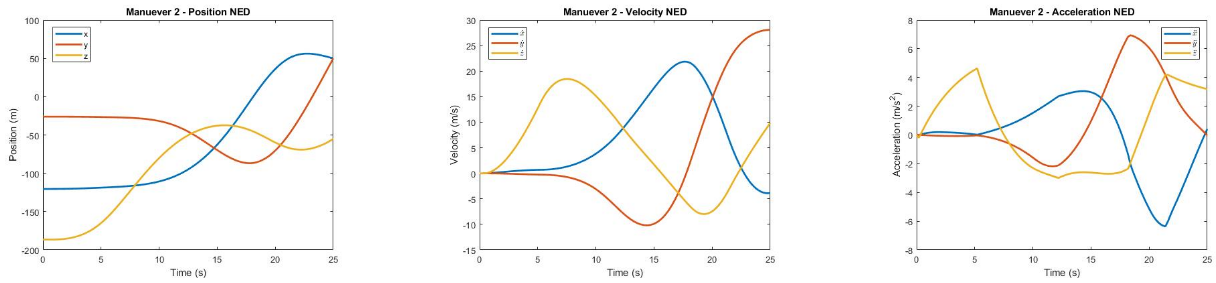


Figure 2.6: UAV position throughout Maneuver 2.



(a) UAV position over time (NED).

(b) UAV velocity over time (NED).

(c) UAV acceleration over time (NED).

Figure 2.7: The nine states of the UAV throughout Maneuver 2.

2.2.3 Maneuver 3

Figure 2.8 shows the position of the UAV throughout Maneuver 3. Trhought this trajectory, the UAV descends towards its target. The NED position, velocity, and acceleration of the UAV throughout maneuver is displayed in Figure 2.9. While the UAV experiences high acceleration in the y-direction (East) during the trajectory, the x and z accelerations are much lower in comparison.

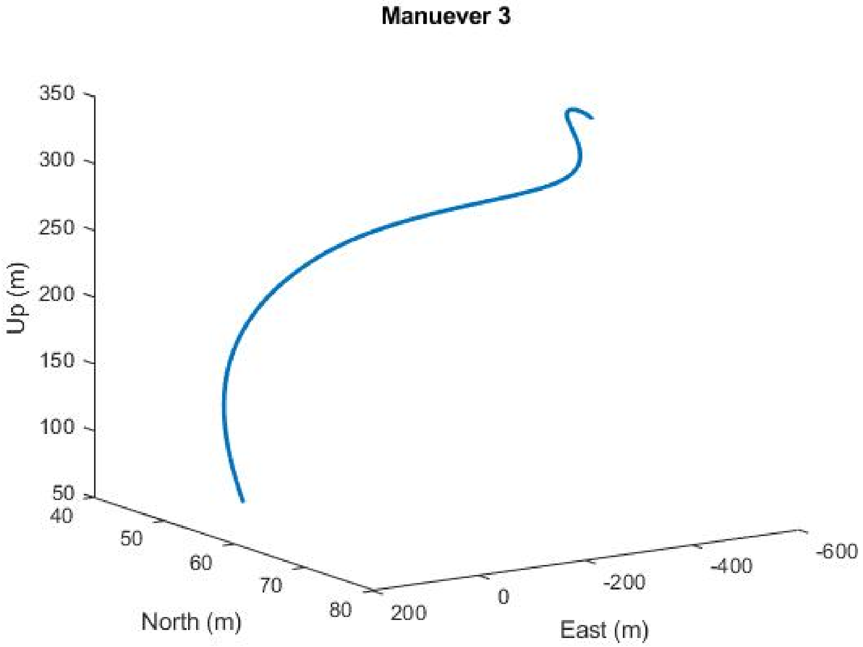
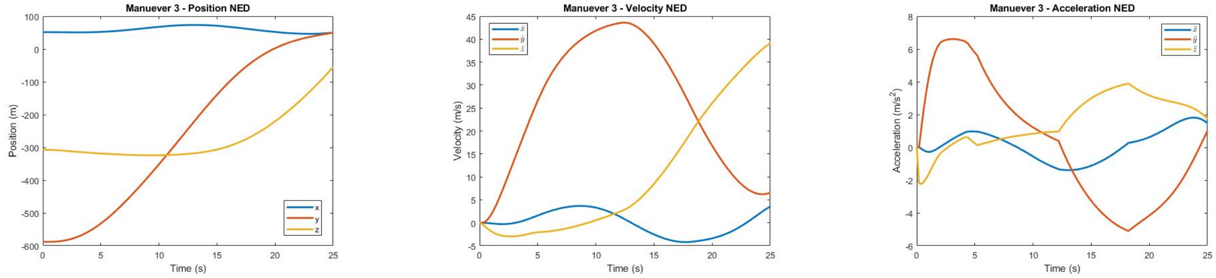


Figure 2.8: UAV position throughout Maneuver 3.



(a) UAV position over time (NED).

(b) UAV velocity over time (NED).

(c) UAV acceleration over time (NED).

Figure 2.9: The nine states of the UAV throughout Maneuver 3.

2.2.4 Maneuver 4

Figure 2.10 depicts the position of the UAV throughout Maneuver 4, while Figure 2.11 displays the position, velocity, and acceleration of the UAV in the NED frame. This maneuver is the least dynamic of the five, and primarily has high acceleration in the z direction. Note that Maneuver 4 has similarities to the z-acceleration of Maneuver 2, which becomes relevant when classifying the maneuvers.

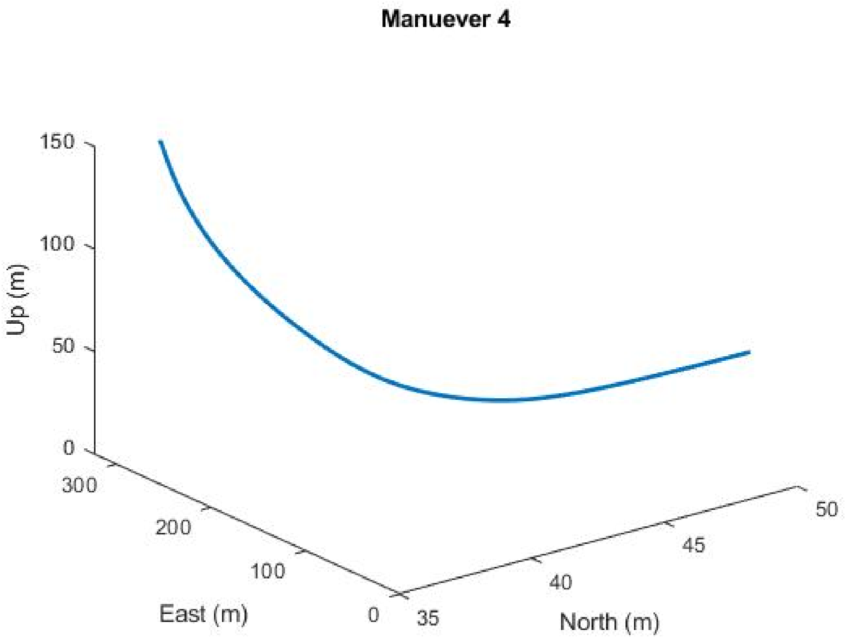
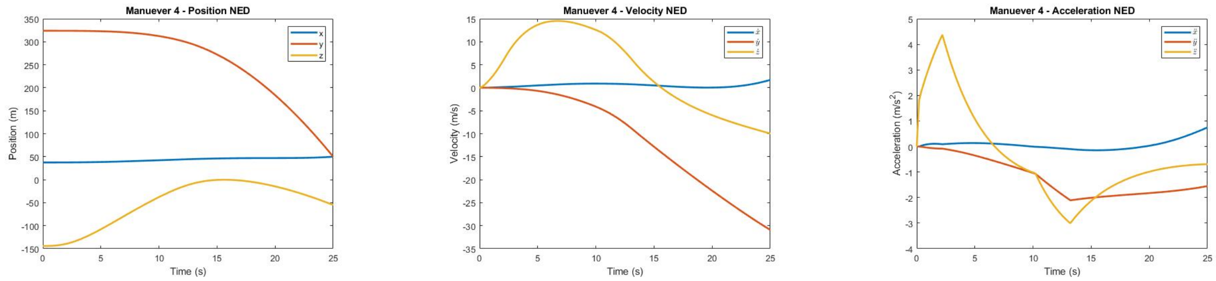


Figure 2.10: UAV position throughout Maneuver 4.



(a) UAV position over time (NED).

(b) UAV velocity over time (NED).

(c) UAV acceleration over time (NED).

Figure 2.11: The nine states of the UAV throughout Maneuver 4.

2.2.5 Maneuver 5

The final maneuver is Maneuver 5, shown in Figures 2.12 and 2.13. This maneuver has an ascension portion of flight for this first 10 seconds of the trajectory before descending towards the target position. It contains large changes in acceleration for all acceleration directions.

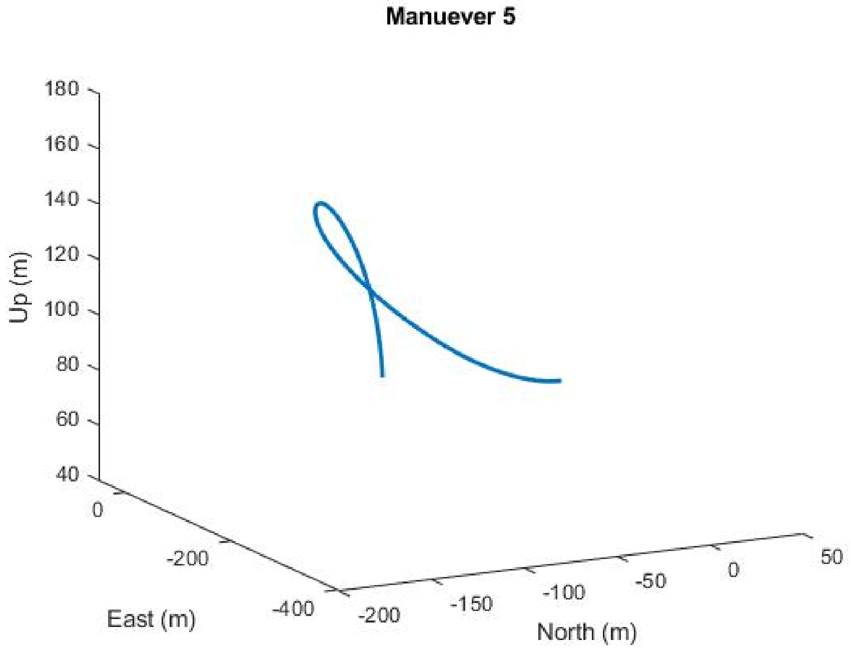
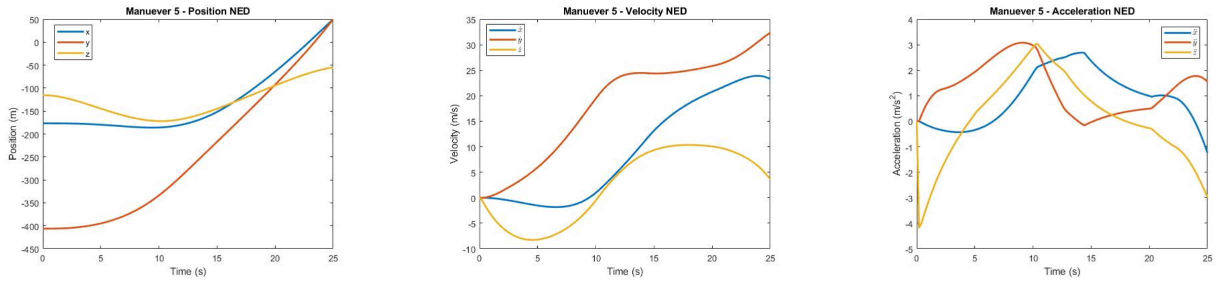


Figure 2.12: UAV position throughout Maneuver 5.



(a) UAV position over time (NED).

(b) UAV velocity over time (NED).

(c) UAV acceleration over time (NED).

Figure 2.13: The nine states of the UAV throughout Maneuver 5.

2.3 UAV State Estimation

A nine-state discrete EKF was used to estimate the NED frame position ($\hat{\mathbf{r}}$), velocity ($\hat{\mathbf{v}}$), and acceleration ($\hat{\mathbf{a}}$) of the UAV,

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{r}} \\ \hat{\mathbf{v}} \\ \hat{\mathbf{a}} \end{bmatrix} . \tag{2.14}$$

A block diagram in Figure 2.14 shows the structure of the EKF. The filter estimates vehicle states by first propagating the states forward based on a dynamic model in the Dynamic Update stage. As radar measurements are input to the system, they are used to correct the states through a Measurement Update.

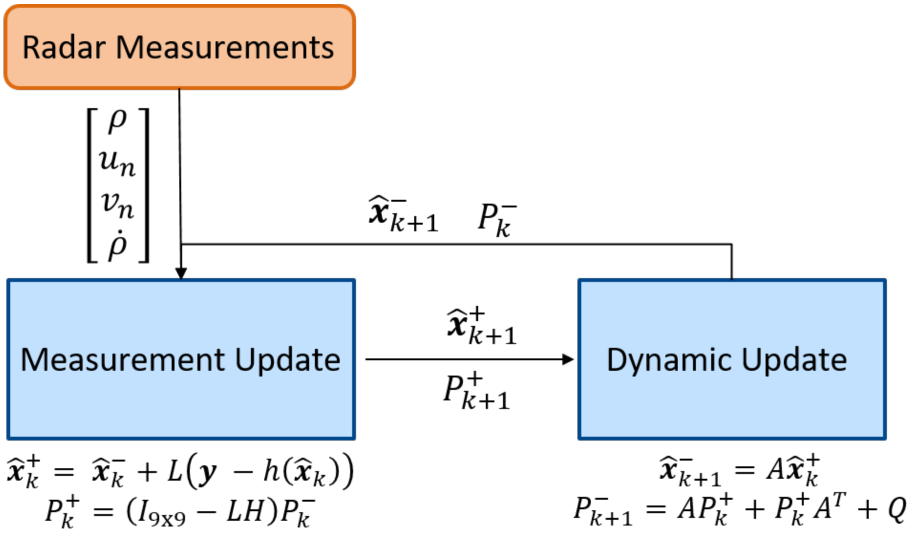

Radar Measurements
$\begin{bmatrix} \rho \\ u_n \\ v_n \\ \dot{\rho} \end{bmatrix}$
$\hat{\boldsymbol{x}}^-_{k+1}$ P^-_k
Measurement Update
$\hat{\boldsymbol{x}}^+_{k+1}$
P^+_{k+1}
Dynamic Update
$\hat{\boldsymbol{x}}^+_k = \hat{\boldsymbol{x}}^-_k + L(\boldsymbol{y} - h(\hat{\boldsymbol{x}}_k))$
$P^+_k = (I_{9x9} - LH)P^-_k$
$\hat{\boldsymbol{x}}^-_{k+1} = A\hat{\boldsymbol{x}}^+_k$
$P^-_{k+1} = AP^+_k + P^+_k A^T + Q$

Figure 2.14: The Extended Kalman Filter utilized to estimate UAV states.

2.3.1 Extended Kalman Filter Dynamic Update

The discrete state dynamic update propagates estimates forward in time and can be represented generally as

$$\hat{\mathbf{x}}^-_{k+1} = A\hat{\mathbf{x}}^+_k + B\hat{\mathbf{u}}_k + B_w\mathbf{w}_k, \tag{2.15}$$

where A is the 9x9 state matrix (or dynamic model), $\hat{\mathbf{x}}$ is the estimated state vector, B is an input matrix, and $\hat{\mathbf{u}}_k$ is the input vector for known deterministic inputs. B_w is also an input matrix for w_k, an unknown stochastic input. This application has no known deterministic inputs to the system, so both B and $\hat{\mathbf{u}}_k$ are 0. Jerk is the unknown stochastic input. It is modelled as zero-mean Gaussian noise and is input into the system through B_w, as seen in Equation (2.16).

A constant acceleration model is used as the dynamic model of the UAV, as it is not specific to the vehicle and therefore is generalizable to different UAV types or situations in which the UAV is unknown. The EKF dynamic update with a constant acceleration model and noisy jerk is represented as

$$\begin{bmatrix} \hat{\mathbf{r}}_{k+1} \\ \hat{\mathbf{v}}_{k+1} \\ \hat{\mathbf{a}}_{k+1} \end{bmatrix} = A \begin{bmatrix} \hat{\mathbf{r}}_{k} \\ \hat{\mathbf{v}}_{k} \\ \hat{\mathbf{a}}_{k} \end{bmatrix} B_w \mathbf{w}_k = \begin{bmatrix} 1 & \delta t & \frac{\delta t^2}{2} \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{r}}_{k} \\ \hat{\mathbf{v}}_{k} \\ \hat{\mathbf{a}}_{k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{w}_k. \tag{2.16}$$

After the estimates are propagated forward in time, the nine-state covariance matrix is updated to reflect any uncertainty created from the dynamic update using the equation,

$$P_{k+1}^{-} = AP_k^{+} + P_k^{+}A + Q, \tag{2.17}$$

where A is the same state matrix used in the dynamic update, P is the covariance matrix and Q is the process noise of the system. The process noise of the UAV was tuned using Bryon's trick.

2.3.2 Radar Measurement Model

Throughout its motion, the UAV is measured by a radar. The radar is stationary at position, $\mathbf{R}_n$, and measures the range, direction cosines, and range rate of the UAV at 5 Hz

[28]. The range distance, ρ_d, of the UAV is found using the equations,

$$\boldsymbol{\rho} = \mathbf{r}_k - \mathbf{R}_n, \qquad \rho_d = |\boldsymbol{\rho}|. \tag{2.18}$$

The final range measurement is a combination of range and range rate,

$$\rho = \rho_d + \tau_{RD}\dot{\rho}, \tag{2.19}$$

where range rate is defined as

$$\dot{\rho} = \frac{\mathbf{v}_k^T \boldsymbol{\rho}}{\rho} \tag{2.20}$$

and τ_{RD} is a the range doppler coupling time [29]. The direction cosines (u and v) are found using equations,

$$u = \frac{\boldsymbol{\rho}(1)}{\rho}, \qquad v = \frac{\boldsymbol{\rho}(2)}{\rho}, \tag{2.21}$$

as seen in [30].

The total radar measurement is defined as

$$\mathbf{y} = h(\mathbf{x}) + \nu = \begin{bmatrix} \rho + \tau_{RD}\dot{\rho} & u & v & \dot{\rho} \end{bmatrix}^T + \nu, \tag{2.22}$$

where ν is zero-mean white measurement noise. The standard deviation of the noise is defined using σ_ρ, $\sigma_{\dot{\rho}}$, σ_u, and σ_v. These values were found using a Monte Carlo simulation in [28] and can be seen in Table 2.1.

Table 2.1: Radar Statistical Parameters

Parameter	Distribution function	Standard Deviation
Radar Range Measurement Error	Gaussian Distribution	$\sigma_\rho = 1.5$ m
Radar Angle Measurement Error	Gaussian Distribution	σ_u and $\sigma_v = 45$ μrad
Radar Range Rate Measurement Error	Gaussian Distribution	$\sigma_{\dot{\rho}} = 0.2$ m/s

2.3.3 Extended Kalman Filter Measurement Update

The radar measurement vector, $\mathbf{y}$, updates the EKF estimates after they have been propagated by the dynamic model. First, the expected radar measurements are found,

$$\hat{\mathbf{y}} = h(\hat{\mathbf{x}}^-). \tag{2.23}$$

They are calculated with Equations (2.18)-(2.21) using estimated states ($\hat{\mathbf{x}}$) rather than true states ($\mathbf{x}$). The actual and expected measurement are then compare, and $\hat{\mathbf{x}}$ is updated based on the difference between the two,

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + L(\mathbf{y} - \hat{\mathbf{y}}). \tag{2.24}$$

The factor that determines how the error between actual and expected measurements changes $\hat{\mathbf{x}}_k^-$ is the Kalman gain, (L), which is defined as

$$L = P_k^- H^T (H P_k^- H' + R_n)^{-1}. \tag{2.25}$$

H is the linearized measurement model, P_k^- is the state covariance matrix after dynamic update, and R_n is the measurement noise matrix, which is defined as

$$R_n = E\left[\nu\nu^T\right]. \tag{2.26}$$

The linearized measurement matrix, H, also updates the state covariance matrix using the equation,

$$P_k^+ = (I_{9x9} - LH)P_k^-, \tag{2.27}$$

where I_{9x9} is an identity matrix.

The linearized measurement model, H, is found by taking the partial derivative of $h(\hat{\mathbf{x}})$ with respect to $\hat{\mathbf{x}}$, which is described in [28] and is defined as

$$H = \frac{\delta h(\hat{\mathbf{x}}_k)}{\delta \mathbf{x}} = \begin{bmatrix} \frac{\delta\rho}{\delta(x_k)} & \frac{\delta\rho}{\delta(y_k)} & \frac{\delta\rho}{\delta(z_k)} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\delta u}{\delta(x_k)} & \frac{\delta u}{\delta(y_k)} & \frac{\delta u}{\delta(z_k)} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\delta v}{\delta(x_k)} & \frac{\delta v}{\delta(y_k)} & \frac{\delta v}{\delta(z_k)} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\delta\dot{\rho}}{\delta(x_k)} & \frac{\delta\dot{\rho}}{\delta(y_k)} & \frac{\delta\dot{\rho}}{\delta(z_k)} & \frac{\delta\dot{\rho}}{\delta(\dot{x}_k)} & \frac{\delta\dot{\rho}}{\delta(\dot{y}_k)} & \frac{\delta\dot{\rho}}{\delta(\dot{z}_k)} & 0 & 0 & 0 \end{bmatrix}. \tag{2.28}$$

Both range and direction cosine measurements are calculated using only position states, and therefore the derivative of the measurements are only taken with respect to x_k, y_k, and z_k. The partial derivative of ρ with respect to each position state is defined as

$$\frac{\delta\rho}{\delta(x_k, y_k, z_k)} = \frac{1}{\rho}\boldsymbol{\rho}^T. \tag{2.29}$$

The partial derivative of the direction cosines with respect to x_k, y_k, and z_k are defined as

$$\frac{\delta u}{\delta(x_k, y_k, z_k)} = \frac{1}{\rho^3}\begin{bmatrix} 1 - \boldsymbol{\rho}(1)^2 & -\boldsymbol{\rho}(1)\boldsymbol{\rho}(2) & -\boldsymbol{\rho}(1)\boldsymbol{\rho}(3) \end{bmatrix} \tag{2.30}$$

and

$$\frac{\delta v}{\delta(x_k, y_k, z_k)} = \frac{1}{\rho^3}\begin{bmatrix} -\boldsymbol{\rho}(2)\boldsymbol{\rho}(1) & 1 - \boldsymbol{\rho}(2)^2 & -\boldsymbol{\rho}(2)\boldsymbol{\rho}(3) \end{bmatrix}. \tag{2.31}$$

Both position and velocity states are used in finding range rate measurements.The partial derivative of $\dot{\rho}$ with respect to x_k, y_k, and z_k is

$$\frac{\delta\dot{\rho}}{\delta(x_k, y_k, z_k)} = \frac{1}{\rho}\mathbf{v}_k^T(I_{3x3} - \frac{\boldsymbol{\rho}\boldsymbol{\rho}^T}{\rho^2}), \tag{2.32}$$

and the partial derivative with respect to $\dot{x}_k$, $\dot{y}_k$, and $\dot{z}_k$ is

$$\frac{\delta\dot{\rho}}{\delta(\dot{x}_k, \dot{y}_k, \dot{z}_k)} = \frac{\boldsymbol{\rho}^T}{\rho}. \tag{2.33}$$

2.4 Estimation of UAV States During Maneuvers

To analyze the capability of the basic EKF during dynamic UAV maneuvers, a Monte Carlo simulation was performed for Maneuvers 1, 2, 3, and 4. (More basic results of the EKF when estimating the states of constant velocity, acceleration, and jerk models can be found in Appendix A.) The maneuvers were simulated with 15 seconds of low dynamic movement before the maneuver began and 20 to 25 seconds of low dynamic movement after the maneuver ends. The low dynamic movement is created through a constant commanded thrust and pitch angle, and aims to show the convergence of the EKF before and after a maneuver. In total, each trajectory utilized for a Monte Carlo simulation is for 65 to 70 seconds.

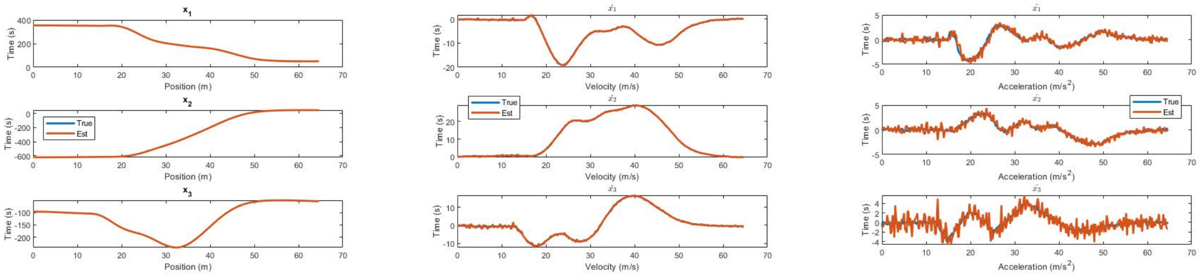
Each trajectory was simulated and estimated 1000 times for the Monte Carlo. The same start and end position of the UAV was used for each simulation. State estimates were initialized with a random offset,

$$\begin{bmatrix} \hat{\boldsymbol{r}}_1 \\ \hat{\boldsymbol{v}}_1 \\ \hat{\boldsymbol{a}}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{v}_1 \\ \boldsymbol{a}_1 \end{bmatrix} + \begin{bmatrix} 1 * randn \\ 0.1 * randn \\ 0.01 * randn \end{bmatrix} . \tag{2.34}$$

Radar measurements were also simulated, using a fixed radar position. The process noise matrix, Q, was tuned initially using Bryson's trick and then scaled. Many different scaling values were used to tune Q in an attempt to mitigate estimation lag during maneuvers while also not producing overwhelming noise on the EKF estimates. The following results show the process noise matrix that resulted in the best EKF performance when balancing these two needs.

2.4.1 Maneuver 1

An example run of the Monte Carlo for Maneuver 1 is shown in Figure 2.15. The position (left), velocity (center), and acceleration (right) of the UAV were estimated in the NED frame. Overall, the EKF was capable of estimating the states throughout the maneuver.



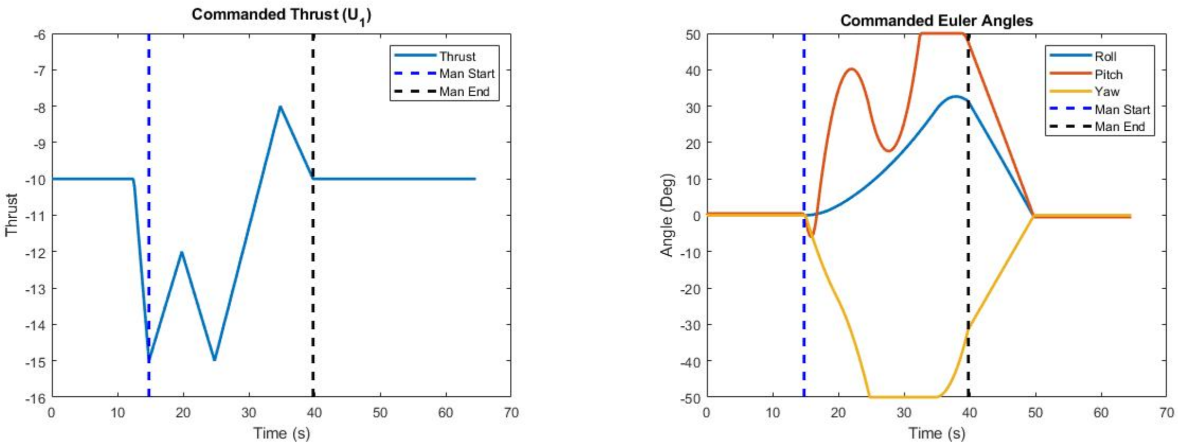
(a) An example of Maneuver 1 position estimates.

(b) An example of Maneuver 1 velocity estimates.

(c) An example of Maneuver 1 acceleration estimates.

Figure 2.15: Estimates of Maneuver 1 using basic EKF.

The commands used to create the total trajectory are shown in Figure 2.16. The blue dotted line indicates the start of the maneuver, while the black dotted line denotes the end. There is a transition period from 12 to 14.8 seconds from the low dynamic maneuver to the initial maneuver thrust, which prevents an immediate rapid increase in UAV acceleration when the maneuver starts. There is also a transition period at the end of the maneuver (from 40 to 50 seconds) in which the commanded angles are slowly decreased to their original values.



(a) Commanded UAV thrust

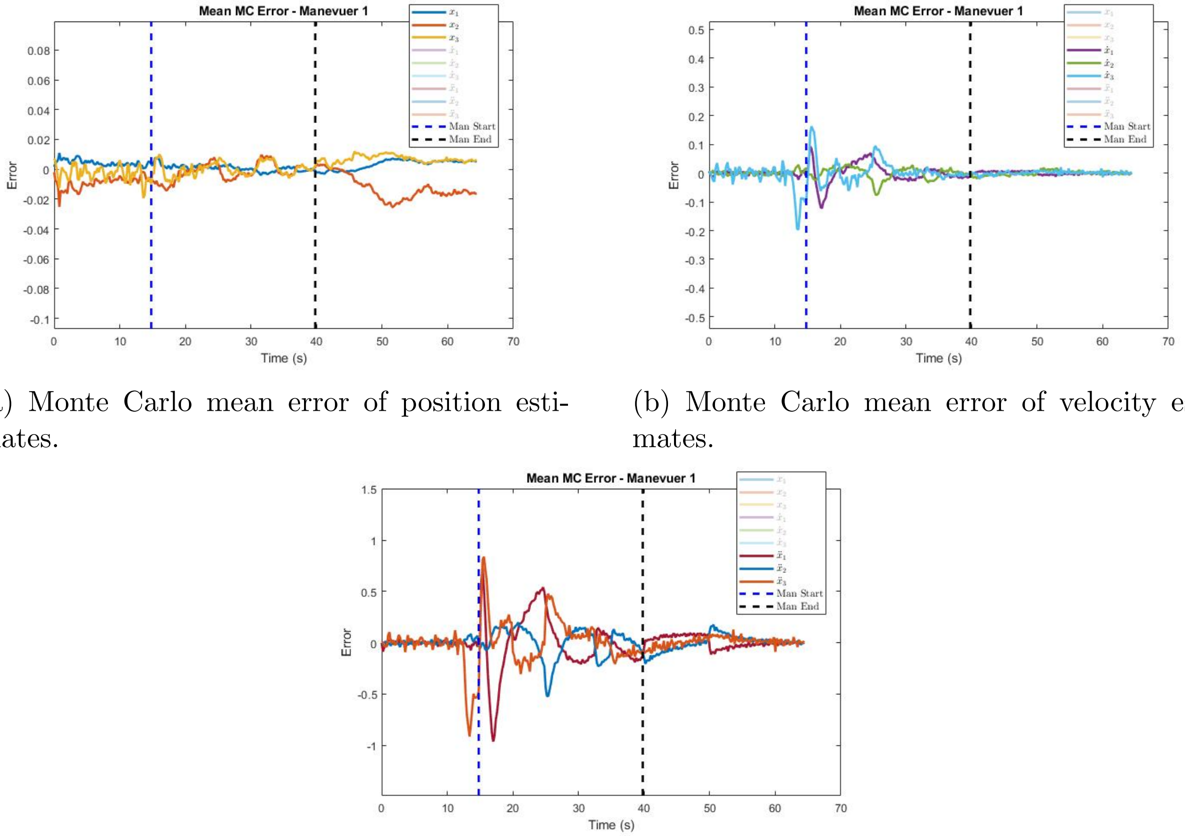
(b) Commanded UAV Euler angles.

Figure 2.16: UAV commands used to execute Maneuver 1.

The average errors across the Monte Carlo simulation for position, velocity, and acceleration are shown in Figures 2.17a, 2.17b, and 2.17c. The start and end time of the maneuver is displayed in each figure, where once again the blue dotted line indicates the maneuver start and the black dotted line indicated maneuver end.

Before the maneuver begins, acceleration and velocity estimates have zero-mean white noise error. There is a slight elevation in estimation error in the two seconds prior to maneuver start. This is consistent with the thrust transition period seen in Figure 2.16a. Position estimates have some error throughout the 15 seconds leading up to the maneuver, but the errors are quite low and clearly converging towards 0.

At the beginning of the maneuver, both velocity and acceleration have consistent errors that are indicative of a lag in the estimation. Even the position estimation has a small consistent error throughout the maneuver (particularly with x_2 and x_3). The errors lessen towards the end of the maneuver as the accelerations decrease – except for the error of x_2. After the maneuver is complete, the velocity estimates return to zero-mean white noise. The acceleration estimates do not converge towards 0 until the the commanded angle transition period (see Figure 2.16b) is complete. The error of the position estimates begin to converge to non-zero values. Because radar measurements do not give complete observability, this is likely caused by a less than optimal radar position relative to the UAV.



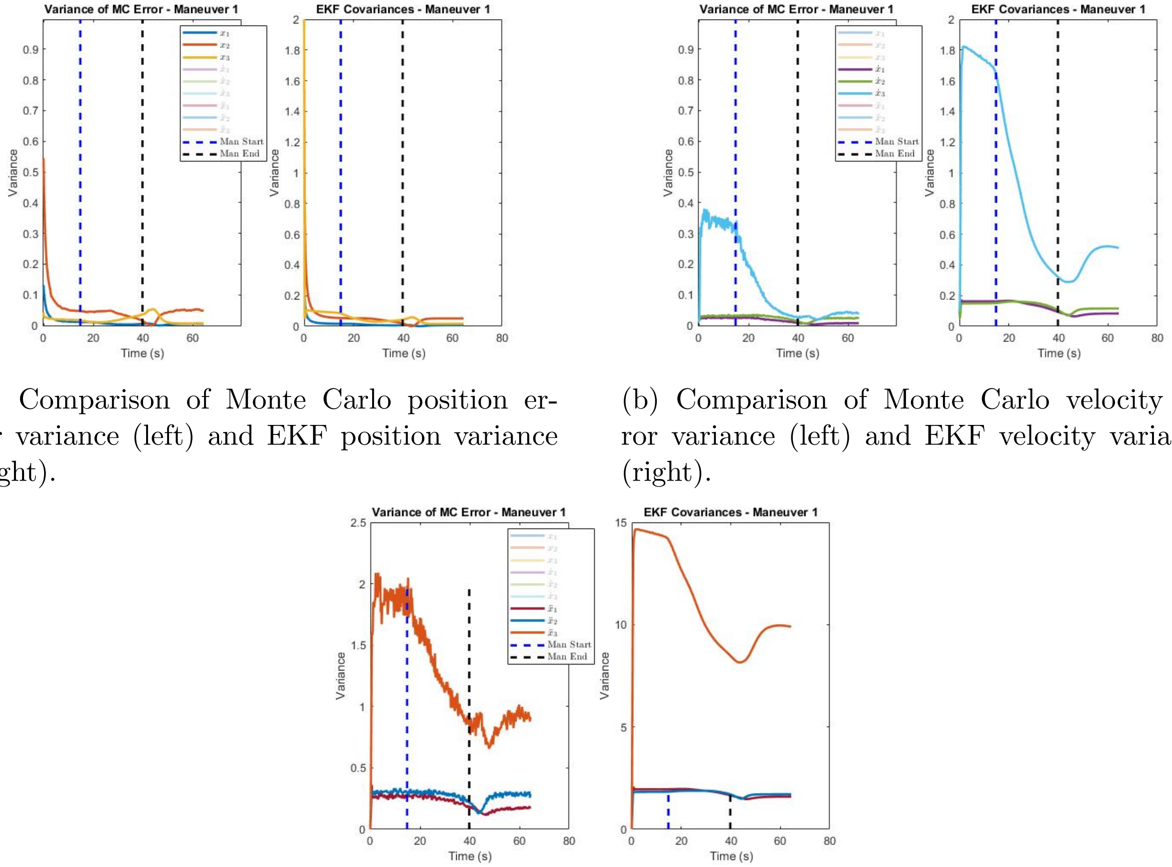
(a) Monte Carlo mean error of position estimates.

(b) Monte Carlo mean error of velocity estimates.

(c) Monte Carlo mean error of acceleration estimates.

Figure 2.17: Monte Carlo mean error when using basic EKF to estimate UAV states during Maneuver 1.

The error variance of each state estimate during the Monte Carlo simulation (left) is compared to the EKF variance (right) in Figure 2.18. The variance of the EKF matches the error variances well. EKF variances are often scaled up when compared to error variances, but this is to be expected as the process noise of the EKF was tuned to a high value.



(a) Comparison of Monte Carlo position error variance (left) and EKF position variance (right).

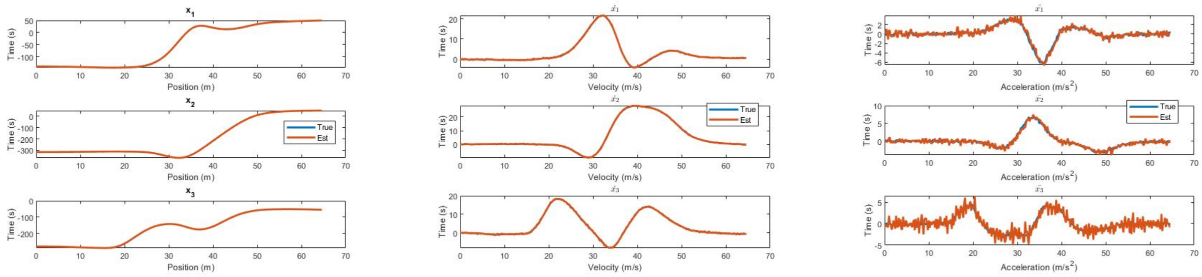
(b) Comparison of Monte Carlo velocity error variance (left) and EKF velocity variance (right).

(c) Comparison of Monte Carlo acceleration error variance (left) and EKF acceleration variance (right).

Figure 2.18: Comparison of Maneuver 1 Monte Carlo error variance to basic EKF error variance.

2.4.2 Maneuver 2

Figure 2.19 shows an example run of the EKF as it estimates UAV states throughout Maneuver 2. Although noisy at times, the EKF is generally capable of tracking the true states.



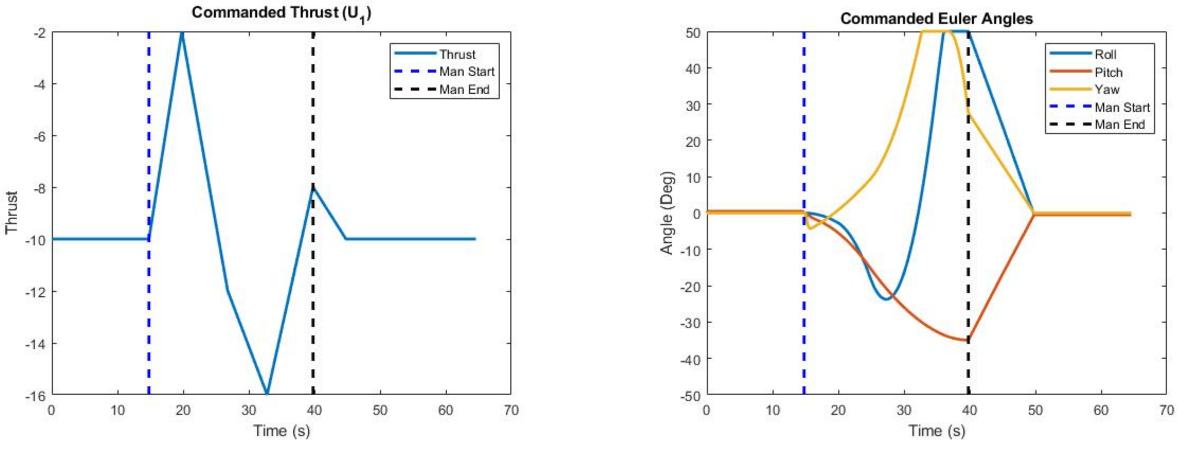
(a) An example of Maneuver 2 position estimates.

(b) An example of Maneuver 2 velocity estimates.

(c) An example of Maneuver 2 acceleration estimates.

Figure 2.19: Estimates of Maneuver 2 using EKF.

The commands used to create the entirety of the trajectory are shown in Figure 2.20, where the blue and black dotted lines represent the start and end of the trajectory respectively. Unlike Maneuver 1, Maneuver 2 does not require a transition period before the start of the maneuver as the initial thrust matches the desired initial thrust of the maneuver. However, Maneuver 2 still requires a transition period at the end of the maneuver in which both the commanded thrust and Euler angles move to their final values that create low dynamic motion.



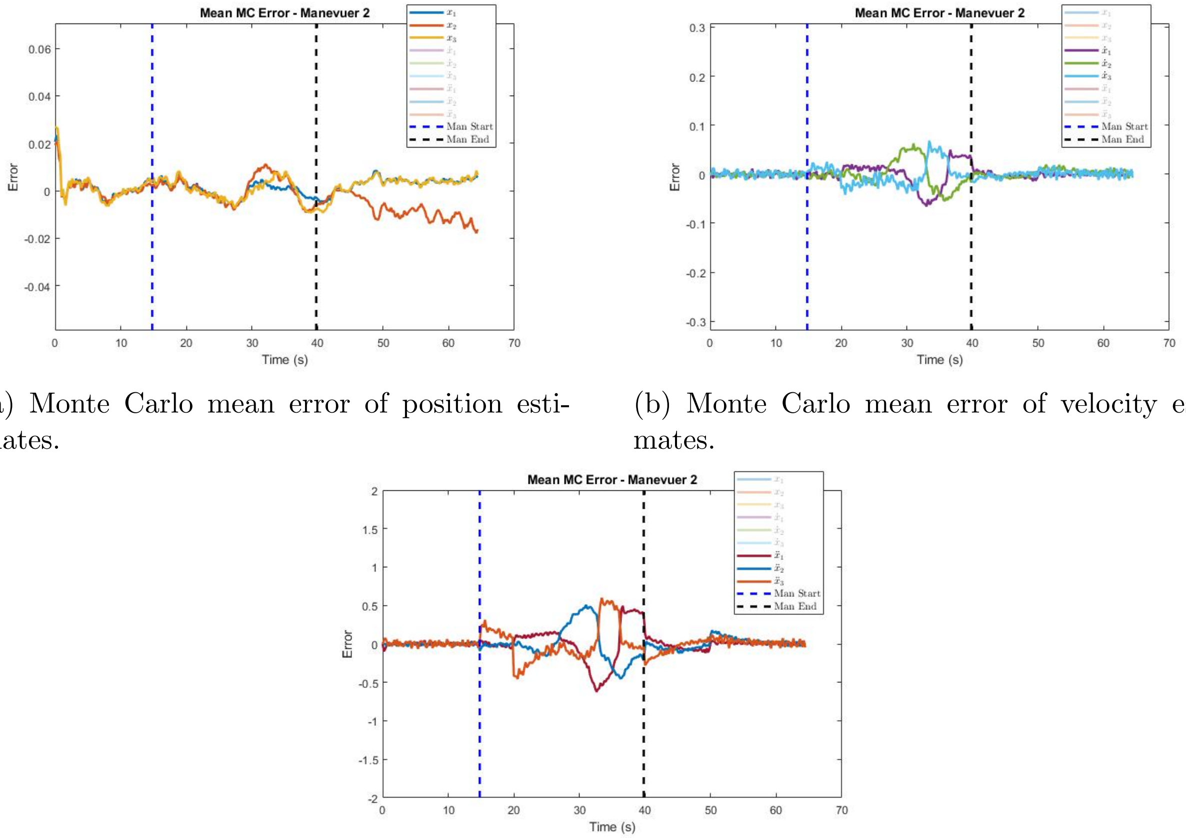
(a) Commanded UAV thrust

(b) Commanded UAV Euler angles.

Figure 2.20: UAV commands used to execute Maneuver 2.

The mean Monte Carlo error of EKF position (Figure 2.21a), velocity (Figure 2.21b), and acceleration (Figure 2.21c) estimates are shown below. Velocity and acceleration have zero-mean white noise error for the flight leading up to Maneuver 2. The position estimates are not entirely without error, but overall it is quite minor.

Once the maneuver begins, a small amount of consistent error can be seen in each state of the first 10 seconds. Then as the acceleration increases (see Figure 2.19c), the error increases for all states - indicating a lag in state estimation when there is high acceleration during the maneuver. After the maneuver ends, small amounts of acceleration and velocity estimate error continue until the transition period ends. Then, they return to zero-mean white noise. Once again, the position estimates begin diverging, likely due to radar positioning.



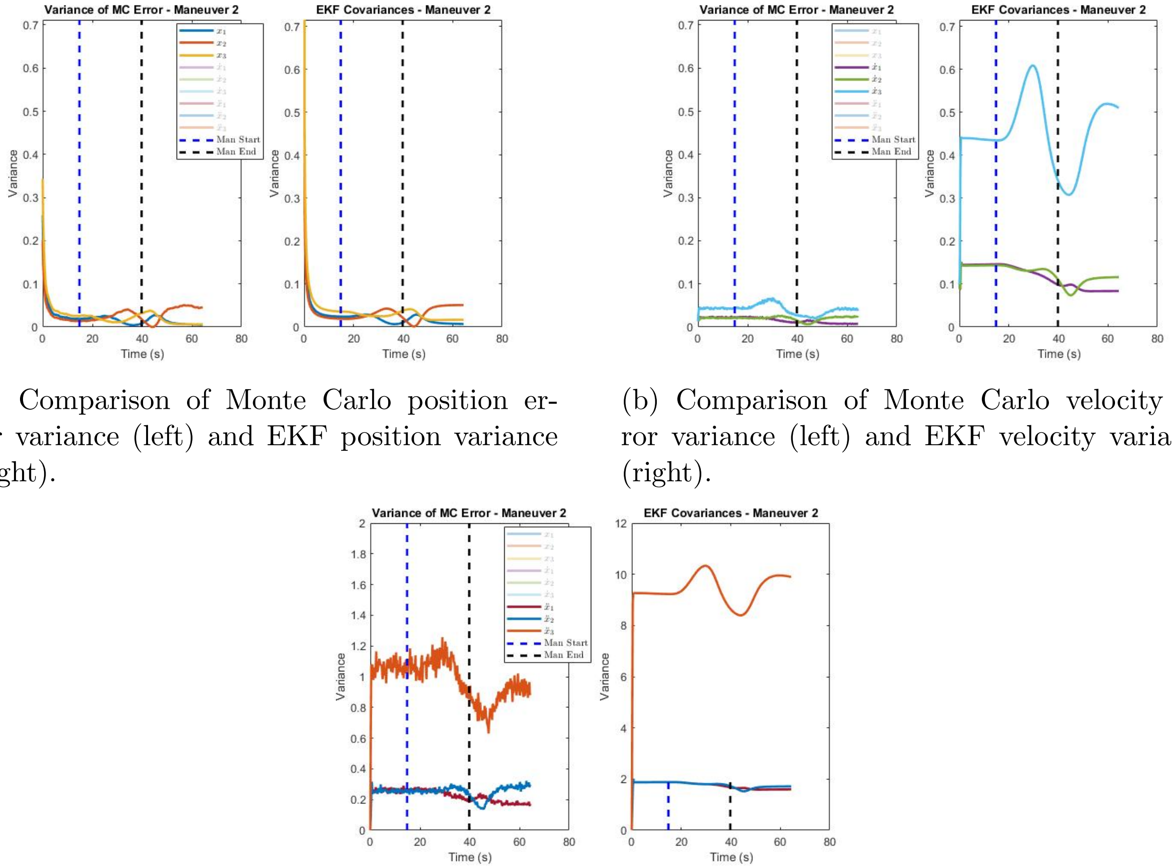
(a) Monte Carlo mean error of position estimates.

(b) Monte Carlo mean error of velocity estimates.

(c) Monte Carlo mean error of acceleration estimates.

Figure 2.21: Monte Carlo mean error when using basic EKF to estimate UAV states during Maneuver 2.

The Monte Carlo estimation error variance is compared to the EKF variance for position, velocity, and acceleration in Figures 2.22a, 2.22b, and 2.22c respectively. Each state converges to some variance (both in Monte Carlo mean error and EKF variance) before the maneuver, and then, as the maneuver begins, the state estimates experience changes in the variance. After both the maneuver and transition period are finished, the variances converge once more. Note that the EKF variance matches the shape of the Monte Carlo error variance, but scaled to a higher value for velocity and acceleration variances. This is to be expected with high process noise.



(a) Comparison of Monte Carlo position error variance (left) and EKF position variance (right).

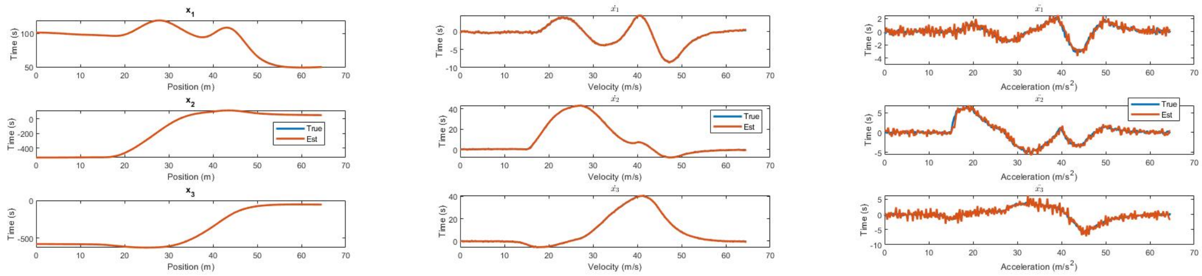
(b) Comparison of Monte Carlo velocity error variance (left) and EKF velocity variance (right).

(c) Comparison of Monte Carlo acceleration error variance (left) and EKF acceleration variance (right).

Figure 2.22: Comparison of Maneuver 2 Monte Carlo error variance to basic EKF error variance.

2.4.3 Maneuver 3

Estimates of UAV states throughout Maneuver 3 are shown in Figure 2.23. These estimates are an example of one of the data runs executed during the Monte Carlo. The results show EKF estimates following UAV states throughout Maneuver 3.



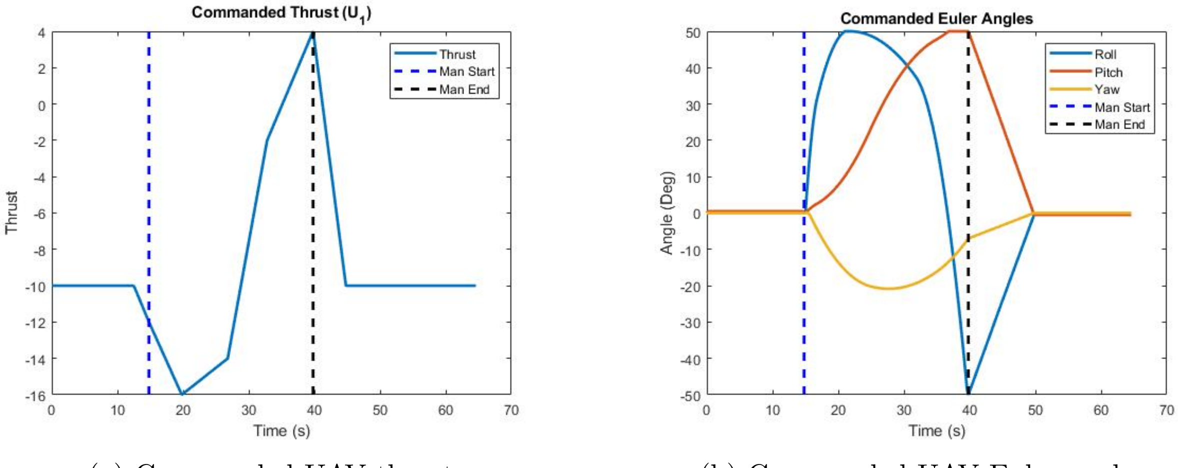
(a) An example of Maneuver 3 position estimates.

(b) An example of Maneuver 3 velocity estimates.

(c) An example of Maneuver 3 acceleration estimates.

Figure 2.23: Estimates of Maneuver 3 using EKF.

The commands used to create the total trajectory are shown in Figure 2.24. Commands specifically for Maneuver 3 are executed from 15 seconds to 40 seconds. Before the maneuver, thrust begins to move towards the initial thrust of the maneuver. After the maneuver, thrust and the Euler angles transition back to their constant values.

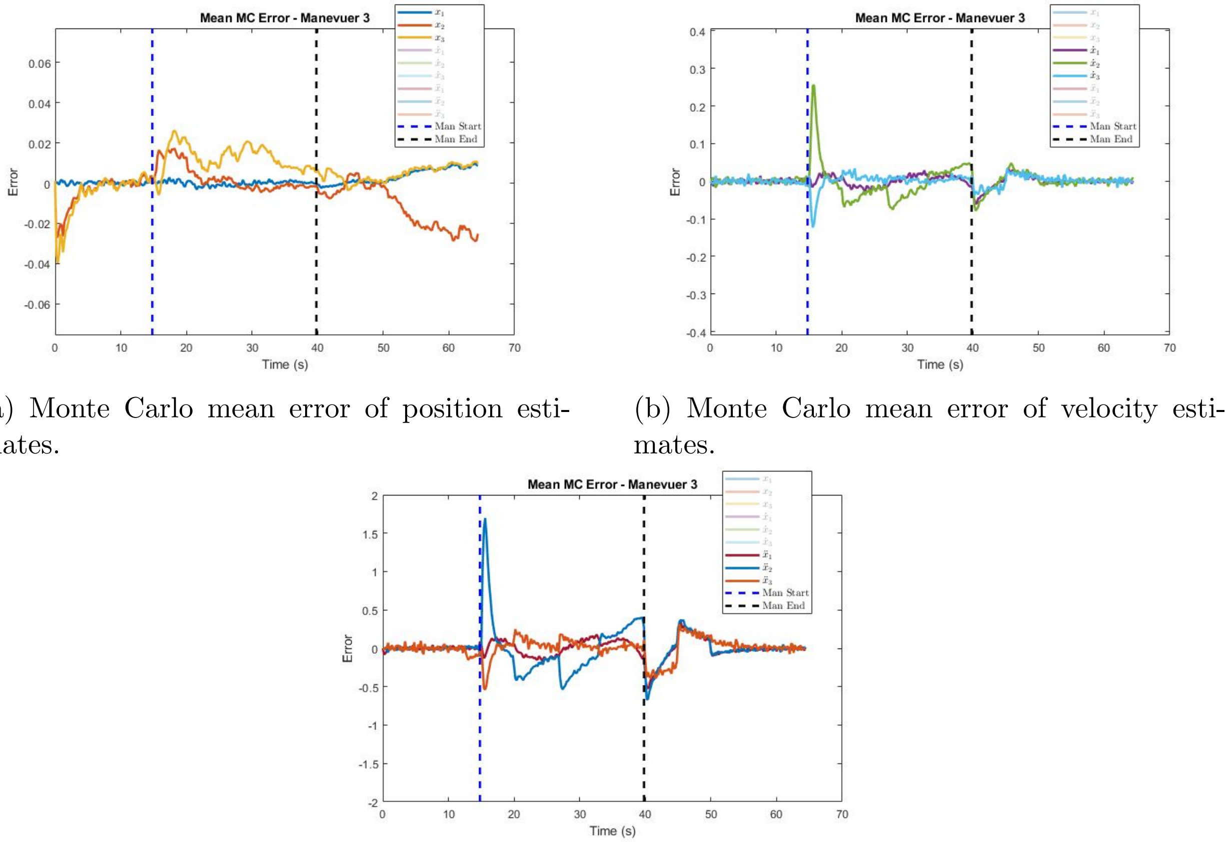


(a) Commanded UAV thrust

(b) Commanded UAV Euler angles.

Figure 2.24: UAV commands used to execute Maneuver 3.

The mean state errors found through the Monte Carlo simulation are shown in Figure 2.25. Each state has zero-mean white noise error for flight before the maneuver. Once the maneuver begins, consistent errors are introduced in each state. v_2 and a_2, which represent motion in the East direction, have the highest errors. After the maneuver is over, consistent error persists until the transition period ends. Then, acceleration and velocity errors return to zero-mean error. Once again, position estimates diverges towards the end of the trajectory.



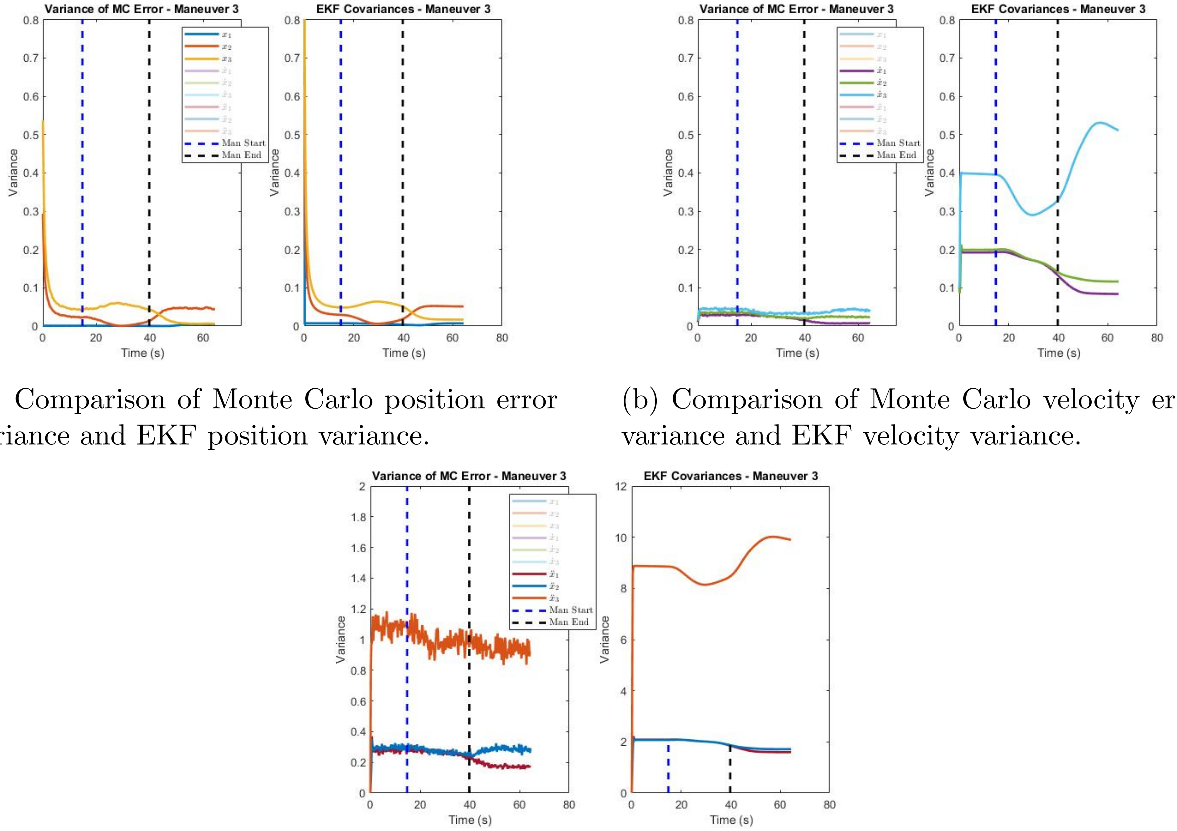
(a) Monte Carlo mean error of position estimates.

(b) Monte Carlo mean error of velocity estimates.

(c) Monte Carlo mean error of acceleration estimates.

Figure 2.25: Monte Carlo mean error when using basic EKF to estimate UAV states during Maneuver 3.

The Monte Carlo state error variance and the variance of each state calculated through the EKF are compared in Figure 2.26. The variances have similar shape for both the EKF and the Monte Carlo error, although the EKF is scaled in value for velocity and acceleration.



(a) Comparison of Monte Carlo position error variance and EKF position variance.

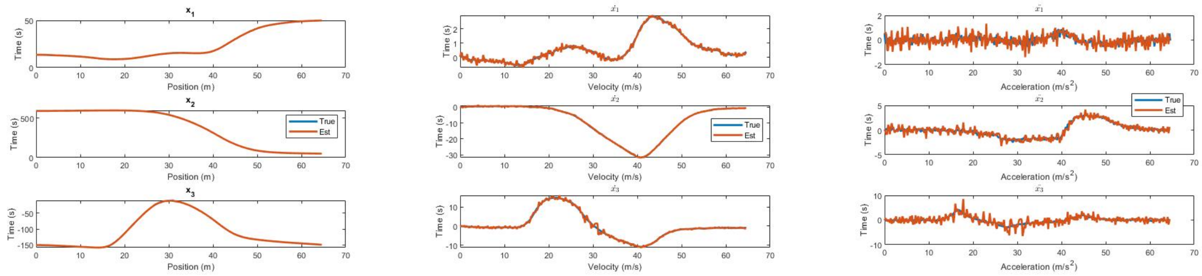
(b) Comparison of Monte Carlo velocity error variance and EKF velocity variance.

(c) Comparison of Monte Carlo acceleration error variance and EKF acceleration variance.

Figure 2.26: Comparison of Maneuver 3 Monte Carlo error variance to basic EKF error variance.

2.4.4 Maneuver 4

The states of the UAV were estimated during Maneuver 4, as shown in Figure 2.27. Maneuver 4 is the least dynamic maneuver, which causes some unnecessarily high noise on the acceleration estimates when using the same process noise values as the other maneuvers. Nonetheless, the EKF capably estimates the UAV states.



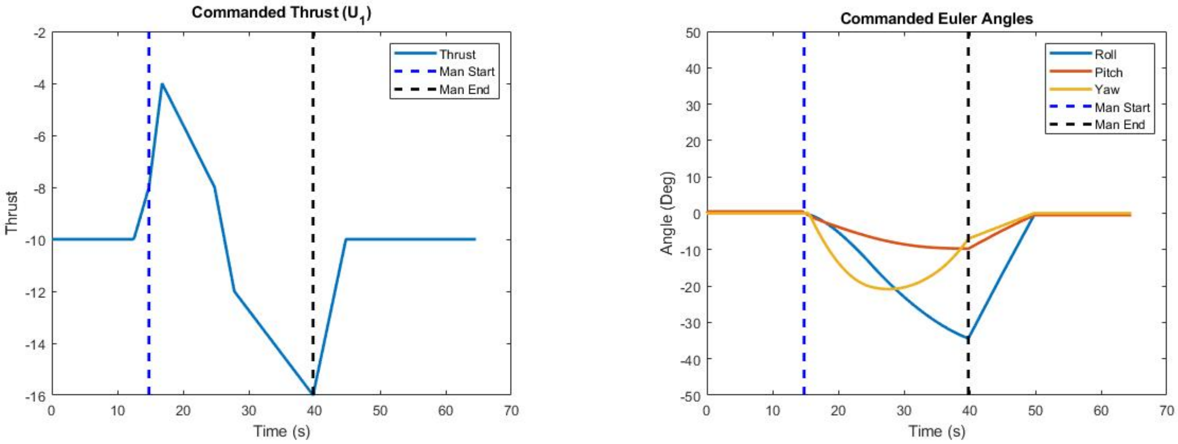
(a) An example of Maneuver 4 position estimates.

(b) An example of Maneuver 4 velocity estimates.

(c) An example of Maneuver 4 acceleration estimates.

Figure 2.27: Estimates of Maneuver 4 using EKF.

The commands used to make the UAV perform the total trajectory are shown in Figure 2.28. The start and end of Maneuver are denoted with the blue and black dotted line. Before and after the maneuver, there are small transition periods in order to start at the correct initial thrust and transition back to small constant Euler angles and the initial thrust.

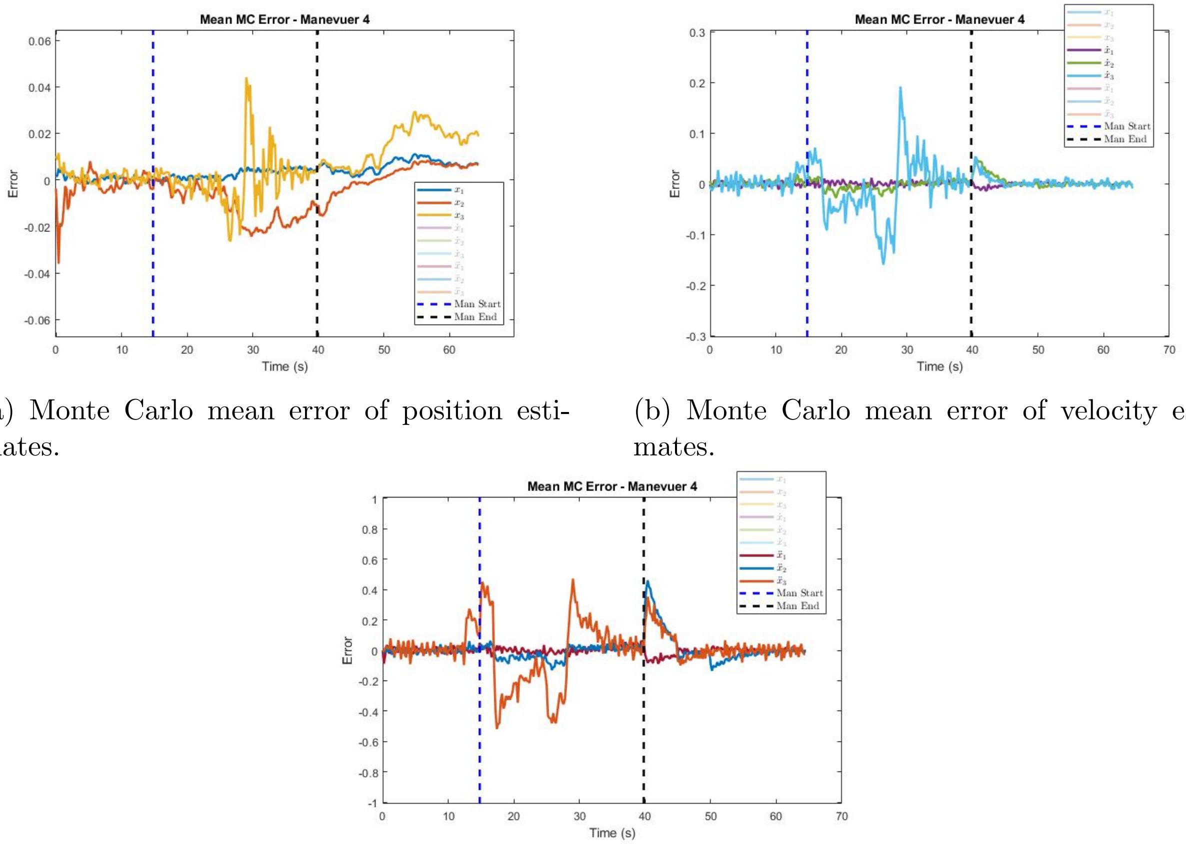


(a) Commanded UAV thrust

(b) Commanded UAV Euler angles.

Figure 2.28: UAV commands used to execute Maneuver 4.

The mean errors of the UAV states over time for Maneuver 4 are shown in Figure 2.29. Position, velocity, and acceleration estimates all reach zero mean error prior to the maneuver start. Maneuver 4 has less acceleration when compared to the other maneuvers, and therefore, has generally less consistent state errors throughout the maneuver. The highest errors are in the down direction. When the maneuver and transition period end, velocity and acceleration error return to zero-mean white noise, while position error continues to diverge slightly.



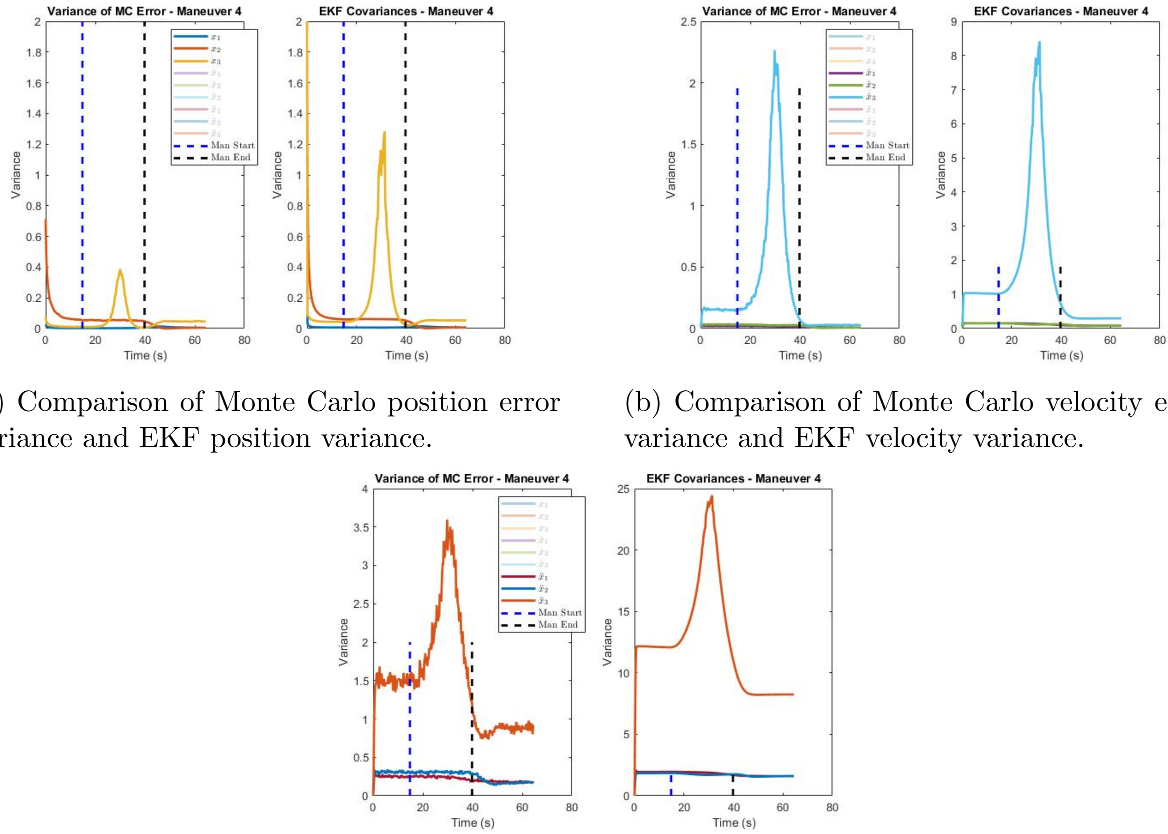
(a) Monte Carlo mean error of position estimates.

(b) Monte Carlo mean error of velocity estimates.

(c) Monte Carlo mean error of acceleration estimates.

Figure 2.29: Monte Carlo mean error when using basic EKF to estimate UAV states during Maneuver 4.

Figure 2.30 shows the variance of state errors from the Monte Carlo compared to the variances of states computed by the EKF. The most notable characteristic of this trajectory is the increase in variance for all states in the down direction (x_3, v_3, and a_3) during the maneuver. The variance of the EKF and Monte Carlo errors match in trend, although the EKF variance is higher in magnitude.



(a) Comparison of Monte Carlo position error variance and EKF position variance.

(b) Comparison of Monte Carlo velocity error variance and EKF velocity variance.

(c) Comparison of Monte Carlo acceleration error variance and EKF acceleration variance.

Figure 2.30: Comparison of Maneuver 4 Monte Carlo error variance to basic EKF error variance.

2.5 Conclusion

This section described UAV modelling, maneuver creation and simulation. The EKF formulation to estimation UAV position, velocity, and acceleration in the NED frame using radar measurements and a constant acceleration dynamic model was also explained. The EKF was then used to estimate UAV states during simulated flight, wherein the UAV maneuvers were performed. EKF performance was evaluated for each maneuver using a Monte Carlo simulation. While the EKF capably estimated UAV states throughout low dynamic portions of flight, the EKF produced consistent estimation error during the high dynamic maneuvers. The following chapter will explore the use of UAV maneuver simulations to create Hidden Markov Models (HMMs), which can be used for classification and to aid in state estimation.

Chapter 3

Modelling and Classification of UAV Maneuvers using Hidden Markov Models

3.1 Introduction

In the previous chapter, UAV maneuvers were developed in simulation. The states of the UAV were estimated using an EKF with a constant acceleration dynamic model. Now, the focus shifts to modelling the same maneuvers using a data driven approach - Hidden Markov Models (HMMs). The data created from maneuver simulations is used to create maneuver specific HMMs. In the following sections, HMMs are defined and the ways they can provide information about a maneuver through classification, state path tracking, and trajectory generation are explored.

3.2 Markov Chain

A Markov chain, shown in Figure 3.1, describes a system as Q discrete states [31]. At any moment in time the system must be in one of Q states, q_t. At uniform discrete time intervals, the system undergoes a transition - either to a different state within the model or to the same state. Each state corresponds to an output of the system that is some physical, observable event. An example of a system that could be modelled as a Markov Chains is a manufacturing systems. Yu and Bricker (1990) modelled a multi-stage manufacturing system as a Markov Chain, where each state of the system was defined as a separate part of the manufacturing process, such as machining, inspecting, and packing [32].

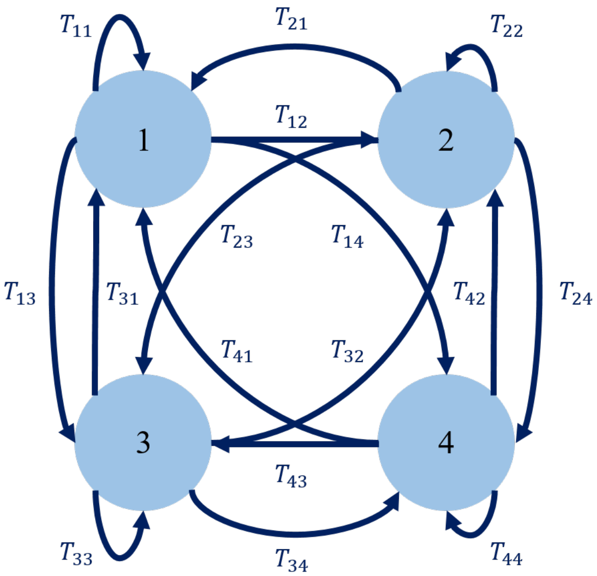


Figure 3.1: An example of a Markov Chain.

To fully describe a Markov chain at time t, the current state as well as all past states must be known. The probability of a series of states occurring is the joint distribution of the series of states from $t = 1$ to $t = T$ [31], which can be found using the chain rule:

$$p(q_{1:T}) = p(q_1)p(q_2|q_1)p(q_3|q_2, q_1)p(q_4|q_{1:3})...p(q_T|q_{1:T-1}). \tag{3.1}$$

First, the probability of the state at time $t = 1$ is found. This is multiplied by the probability of the state at time $t = 2$, given the state at $t = 1$, which is multiplied by the probability of the state at $t = 3$, given the states at $t = 1$ and $t = 2$, and so on. As t increases, finding the probability of a state sequence becomes more complicated, as does describing the model through its potential state series.

The Markov chain is be simplified using the first order Markov assumption, which assumes that the future is conditionally independent of the present given the past, or

$$q_{t+1} \perp q_{1:t-1}|q_t. \tag{3.2}$$

The joint distribution of a series of states can then be written as

$$p(q_{1:T}) = p(q_1) \prod_{t=1}^{T} p(q_t|q_{t-1}). \tag{3.3}$$

A Markov chain that utilizes the first order Markov assumption is known as a first order Markov chain. When $p(q_t|q_{t-1})$ is assumed to be independent of time, the first order Markov chain is also time invariant. A time invariant first order Markov chain at time t can be described using only the current state and $p(q_t|q_{t-1})$ for each state[31].

The state transition matrix denotes $p(q_t|q_{t-1})$ for each state in a compact form. For example, a Markov chain with four states has the transition matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, \tag{3.4}$$

where a_{ii} is the probability of remaining on the same state from $t-1$ to t and a_{ij} is the probability of the model transitioning to state j at time t given its current state i at $t-1$.

3.3 Hidden Markov Models

A Hidden Markov Model (HMM) models a system as a time invariant first order Markov Chain where the states are not directly observable. Instead, the states are latent variables [31][33]. The system produces data, O_t, known as an observation – the measurable emission of the HMM. O_t can be simple, like the result of a dice roll, or more complex, like a vector of radar measurements. The emission probability ($B_i(O) = p(O_t|q_t)$) relates the emission to the latent states. Figure 3.2 shows an example of an HMM. The model is obscured with only O_t, the observations emitted from the model, visible.

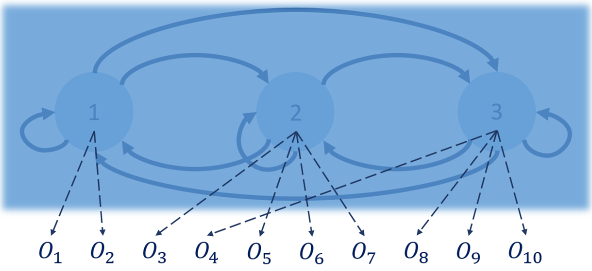


Figure 3.2: An example of a Hidden Markov Model.

The emission probability of a state S_i, $B_i(O)$, can be either continuous or discrete, depending on the characteristics of the observations. For example, Dimitrova et al. (2000) used discrete observation symbols and a corresponding observation symbol probability matrix to represent different discrete events occurring on a TV screen [34]. In contrast, Nefian et al. (2002) used a Gaussian Mixture Model as the emission probability function for audio and visual features in audio-visual speech recognition system [35].

Overall, a Hidden Markov Model has can be described through 5 elements [33]:

1. The number of states, Q.
2. The observations, O_t, emitted at each time step.
3. The emission probability, $B_i(O_t) = p(O_t|q_t = S_i)$, of each state.
4. The transition matrix, A, which contains the probabilities, $p(q_t = S_i|q_{t-1} = S_j)$, for each state.
5. The initial state distribution, π, which describes the probability of a state starting the HMM state sequence, $\pi_i = p(q_1 = S_i)$.

The structure of HMM excels at modelling long range dependencies across observation sequences [31]. As a result, HMMs are often used in time-series prediction, time-series generation, and time-series classification problems. HMMs are utilized in creation of predictive models of human behavior [36], gene finding in DNA [37], speech recognition [38], handwriting recognition [39], and stock market forecasting [40].

3.3.1 Model Training

Hidden Markov Model training is the process through which the parameters (A, B, π) of an HMM model, λ, are adjusted to maximize the probability of the training data. More specifically, training adjusts the parameters to maximize $P(O|\lambda)$, where O is the series observations used as training data. There is no known analytical solution to this problem. One method to train an HMM is to describe training as an optimization problem and use gradient techniques to find the parameters, but this method has generally not produced satisfactory results [41][33]. The most often used technique, and the technique used in this thesis, is the Baum-Welch algorithm [42][33].

The Baum-Welch algorithm is a variation of the Expectation-Maximization algorithm. It is an iterative process that is capable of satisfying the stochastic constraints of an HMM. Originally, the algorithm was formulated with only discrete emission probability functions in mind, but the formulation has now expanded to also include some continuous observation densities as well, including Gaussian Mixture Models [43][44][45]. For further reading on the implementation of HMM training, as well as how to implement it with continuous observation densities, [33] is recommended.

The number of states of the HMM (Q) and any hyperparameters required for the emission probability must be determined before beginning the Baum-Welch algorithm. For example, if the emission probability is represented as a GMM, the number of mixtures per GMM (M) is needed. The algorithm also requires initial values for the model parameters $\lambda_{init} = (A_{init}, B_{init}, \pi_{init})$.

The algorithm is divided into two stages: expectation and maximization. First, the expectation stage defines what "should" happen according to the current model parameters, $\lambda = (A, B, \pi)$, and the observation sequence, O. The expected frequency of visits to each state, expected number of state transitions, and expected observations for a certain state are found. Then, in the maximization stage, these expected values are used to re-estimate the model parameters and produce a new model, $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$.

It has been show that the newly re-estimated model will either be more likely than the previous model ($P(O|\bar{\lambda}) > P(O|\lambda)$) or, if λ was already the best model, the two will have equal likelihood ($P(O|\bar{\lambda}) = P(O|\lambda)$) [46][47]. As the process is iterated, λ will continue to increase in likelihood for the given training data. Typically, the process is stopped once a certain number of iterations is reached or the change in model likelihood from iteration to iteration is considered negligible.

The Baum-Welch algorithm is only capable of finding a local maxima [33]. Therefore, the training procedure is often performed multiple times with different model parameter initialization, or strategies are used to ensure the initial model parameters will produce worthwhile results.

3.3.2 Modelling UAV Maneuvers as HMMs

In this thesis, the UAV Maneuvers 1, 2, 3, and 4 are each modelled as an HMM. The observations of each model are the UAV position, velocity, and acceleration states. A Gaussian Mixture Model (GMM) was chosen as the probability density function for each state's emission probability. An example of a three dimensional GMM is shown in Figure 3.3.

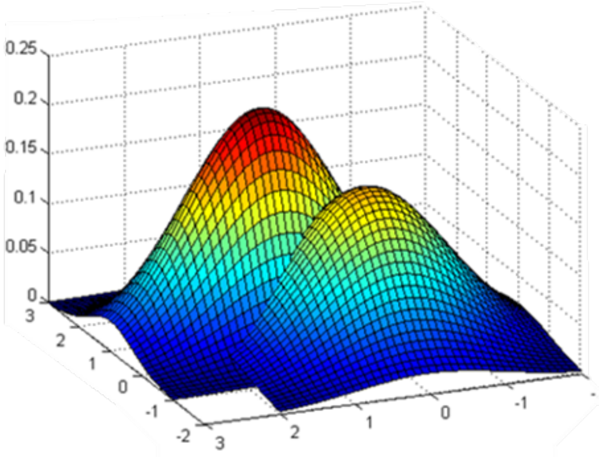


Figure 3.3: An example of a Gaussian Mixture Model [1].

The emission probability function is described as

$$B_i(O_t) = \sum_{i=1}^{M} p_i \mathcal{N}(O_t, \mu_i, \Sigma_i), \tag{3.5}$$

where B_i is the emission probability of state i, M is the number of mixtures, p_i is the weighting of each mixture, and $\mathcal{N}(O, \mu_i, \Sigma_i)$ is the multivariate Gaussian of a given mixture. The multivariate Gaussian is defined as,

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)), \tag{3.6}$$

where μ is mean vector, Σ is the covariance matrix, and k is the dimension of the data. For this application, the dimension of the data is nine, as the input to the multivariate Gaussian, $\mathbf{x}$, is the state vector of the UAV.

3.3.3 Training UAV HMMs

Data for training each maneuver's HMM was generated using maneuver simulations detailed in Section 2.2. Diverse training data for each maneuver was created by simulating the maneuver over 2000 times with different start and end positions and random process noise added to the acceleration of the vehicle. UAV position, velocity, and acceleration states were recorded at 5 Hz for each simulation. The true states of the UAV (not the filtered states) were used as training data.

Maneuvers 1-4 were randomly divided into a training and testing data sets, where 70% of the data was used for training and 30% was reserved for testing. Maneuver 5 was not used to create an HMM and instead is used as an unknown trajectory. Before UAV states were used as observations for the HMM, they were differenced and normalized using the global minimum and maximum of the training data set [48]. This allowed model observations to be invariant to UAV position and scaled in a way that is easier to train. The data pre-processing is shown as

$$\mathbf{x}_n = \frac{[\mathbf{x}_{t+1} - \mathbf{x}_t] - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}. \tag{3.7}$$

To find the initial parameters of the emission probabilities for model training, a random training set trajectory was chosen for each maneuver. A Gaussian Mixture Model was found using the Expectation-Maximization algorithm and data points from that trajectory. The GMM was calculated several times and with varying numbers of mixtures (from three to ten) and different initial conditions. The GMM with the highest log-likelihood was then selected to initialize HMM parameter, B_{init}. Each individual mixture was used to initialize a single state's emission probability, making the number of mixtures in each emission probability, M, one. This procedure is similar to the pre-training procedure of [21].

The transition matrix, A, was initialized simply as

$$A_{init} = \frac{1}{Q} I_{QxQ} \tag{3.8}$$

and π was initialized as

$$\pi_{init} = \frac{1}{Q} ones(1, Q). \tag{3.9}$$

After initialization and training, Maneuver 1 was modelled as a five state HMM (λ_1), Maneuver 2 was modelled as a six state HMM (λ_2), Maneuver 3 was modelled as a six state HMM (λ_3), and Maneuver 4 was modelled as a six state HMM (λ_4). λ_1 differs in number of states when compared to the other HMMs only because its 5 state produced superior results in training data estimation and classification, while the other models had better results with 6 states.

3.4 Classification of UAV Maneuvers

The Hidden Markov Model of each UAV maneuver can be utilized for maneuver classification in real time. The fundamental idea behind classification is to determine which model is most likely to have produced an observation sequence, **O**. The following section describes two classification schemes used to classify UAV maneuvers. Both schemes use the forward algorithm, a common HMM classification method, as the first step in the classification process. Then, either one-class classification or a confidence metric is used to complete classification and provide additional robustness to the classifier.

The classification schemes were tested using each maneuvers' HMM. UAV states are input into the classification schemes and are used to update the current maneuver classification. The models are first tested using the training data set of truth data (which they were originally trained on) to confirm the basic ability of the classification scheme. They are then tested using the test data set of truth data to ensure that the models are not overfit to training data.

After these initial results, the classification schemes are tested again using estimated UAV states generated from the EKF in Section 2.3. Classification ability is first evaluated using estimates of maneuver trajectories in the original training data set. Then, it is accessed using estimates of the test data set. These results will aid in examining if the classification schemes are robust to filtered data when trained on truth data. The estimated UAV states are often noisy in comparison to truth data and may have some lag in estimating the UAV states during the maneuver (see Section 2.4). Classification results from using estimated data are used evaluate whether HMMs can still be used with imperfect estimated data.

3.4.1 Classification Methods

Forward Algorithm [33]

In order to classify a model,the probability of an observation sequence given a model, $P(O|\lambda)$, must be determined. This probability is defined as

$$P(O|\lambda) = \sum_{allQ} P(O|Q,\lambda)P(Q|\lambda), \tag{3.10}$$

where $P(O|Q, \lambda)$ is the probability of the observation sequence occurring given the state sequence Q and the model λ and $P(Q|\lambda)$ is the probability of the state sequence given the model. The product of these probabilities is summed over all possible state sequences produced by the model to account for all possible variations. While accurate, this definition of $P(O|\lambda)$ is computationally intensive to calculate.

The forward algorithm is a faster way to find $P(O|\lambda)$ that uses the variable α. $\alpha_t(i)$ is updated for each state throughout the forward algorithm and is defined as the probability of the current state at time t and the observation sequence up to time t given the model, or

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, q_t = S_i|\lambda). \tag{3.11}$$

α is initialized using the first observation of the sequence, O_1, the probability of this observation originating from state S_i (also known as the emission probability $B_i(O_1)$), and the initial probability distribution of S_i, π_i:

$$\alpha_1(i) = \pi_i B_i(O_1), \qquad 1 \leq i \leq Q. \tag{3.12}$$

Next, α is propagated through the induction step, which is described as

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{Q} \alpha_t(i) a_{ij}\right] B_j(O_{t+1}), \qquad 1 \leq t \leq T-1, \quad 1 \leq j \leq N. \tag{3.13}$$

The algorithm uses a trellis structure, seen in Figure 3.4. At each time step, $\alpha_{t+1}(j)$ is updated using the emission probability of the observation ($B_j(O_{t+1})$) and the joint probability of S_j occurring at time $t+1$ and of the entire observation sequences occurring from time $t = 1$ to time t, $(O_1, O_2, ..., O_t)$. This probability is described as $\sum_{i=1}^{Q} \alpha_t(i) A_{ij}$.

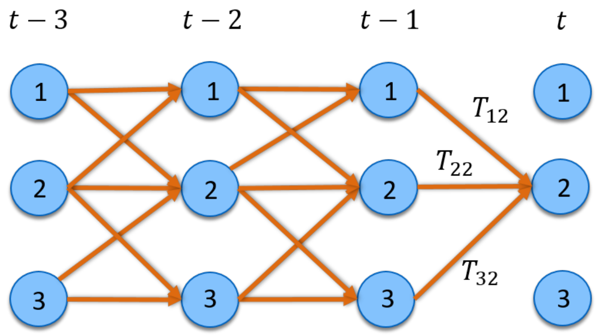


Figure 3.4: An illustration of the trellis structure used in the forward algorithm.

When the observation sequence reaches its final observation, the forward algorithm performs the termination step,

$$P(O|\lambda) = \sum_{i=1}^{Q} \alpha_T(i). \tag{3.14}$$

Each $\alpha_T(i)$ considers all possible paths that could have lead to that final state, S_i, as well as the probability of the entire observation sequence. Therefore, summing $\alpha_T(i)$ over all states produces $P(O|\lambda)$. If there is a need to know $P(O_{1:t}|\lambda)$, the probability partial observation sequence given the model, then $\alpha_t(i)$ can be summed each time it is updated with a new observation in the induction state,

$$P(O_{1:t}|\lambda) = \sum_{i=1}^{Q} \alpha_t(i). \tag{3.15}$$

When classifying a model, $P(O_{1:t}|\lambda)$ can be calculate for each model and compared. The model with the highest $P(O_{1:t}|\lambda)$ is considered the correct classification. The forward algorithm can also be modified to utilize logarithms within the algorithm and produce the log-likelihood of a model. This is the method used for classification in this thesis (and is further explain in [33]). The model with the highest log likelihood is considered the best classification for a given observation sequence.

Although the forward algorithm provides a method to compare HMMs for classification, an issue arises when classifying an observation sequence from an unknown system. Simple model comparison in classification guarantees the classification will always be one of the known models – even when this is incorrect. One-class classification and confidence measures provide two potential solutions to this problem.

One-Class Classification

One-class classification is typically used in two-class classification problems when data from only one class is readily available. Typically, the discriminant function for a classification system evaluates for all classes, but, as the name suggests, one-class classification evaluates the discriminant function for only one class [49]. For example, detecting an intruder in a system is a scenario in which the training data will only consist of the usual behavior for that system. The classifier must determine if activity is normal or not normal without having ever experienced an abnormal situation.

HMMs are used for some one-class classification applications [50] [51]. The one-class classification method used in this thesis is based on [52]. This simple method states that the log-likelihood of a model must exceed a selected threshold to be considered for classification. The threshold is determined through finding the log-likelihood of training data observations sequences (using the forward algorithm) and determining a log-likelihood value that most or all of the sequences exceed.

The classifier using one-class classification is shown in Figure 3.5. First, the observation sequence is input into each HMM and the log-likelihood of each model is compared. Initial classification occurs by finding which model produced the maximum log-likelihood. The corresponding maneuver is selected as the initial classification. Then, the log-likelihood of the classified model is input into one-class classification and is compared to the established threshold. If the log-likelihood exceeds the threshold, then the initial classification becomes the final classification. If the log-likelihood does not exceed the set threshold, the final maneuver classification is unknown.

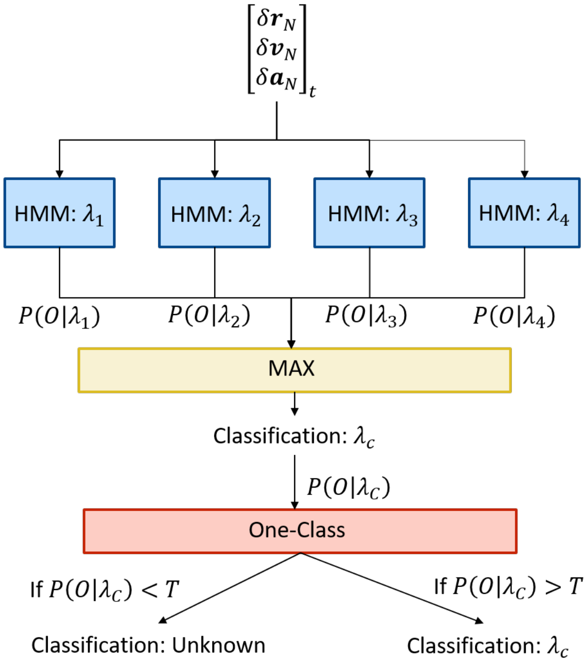


Figure 3.5: Classifier that uses the forward algorithm and one-class classification to determine a maneuver given a series of UAV states.

Confidence Measure

Another method to evaluate the forward algorithm's initial classification is using a confidence measure [53],

$$V = \log_{10} \sum_i \frac{P_i}{P_C}, \tag{3.16}$$

which compares the probability of the classified model, P_C, to the sum of the probabilities of the unclassified models, P_i. The measure shows whether the selected model is distinguished from the others when encountering a new observation sequence.

High confidence classification occurs when the confidence measure, V, is less than -2. This indicates a large difference between the log-likelihood of a maneuver's model and the log-likelihoods of the other models. When V is between -1 and -2, the classification has some uncertainty. If $-1 < V < 1$, then the classification is highly uncertain, because multiple models have similar log-likelihoods for the observation sequence. A score of $V > 2$ indicates high confidence that the model is not the correct classification, as the model has a distinctly lower log-likelihood than some other.

The classification scheme when using the confidence measure is shown in Figure 3.6. Like in Figure 3.5, after the log-likelihood of each model is calculated using the observations, the model that produces the maximum log-likelihood is found. The maneuver associated with this model becomes the initial classification. Then, the confidence measure is used to evaluate this initial classification by comparing this model's log-likelihood to the others. If the model currently classified from the forward algorithm has a confidence score greater than -1, then the maneuver is classified as unknown. If it's less than -1, the initial classification becomes the final classification.

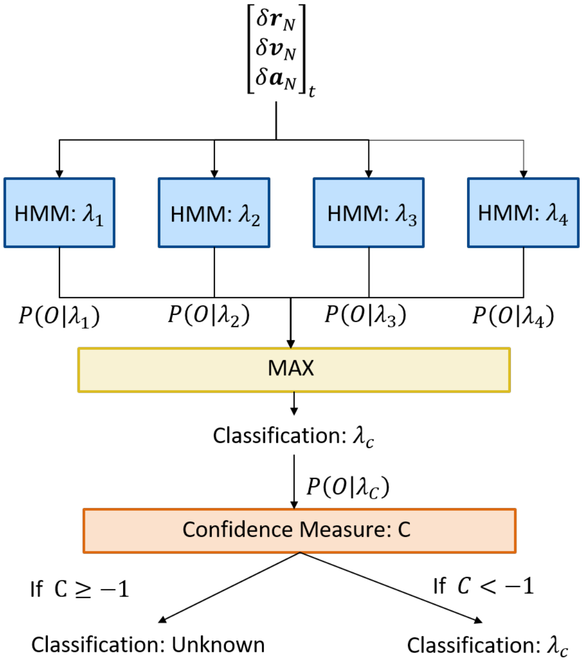


Figure 3.6: Final classifier used to determine a maneuver given a series of UAV states.

3.4.2 Classification Results: True States

Training Data

Both classifiers were first tested using the UAV truth data that used to train the HMMs. Table 3.1 shows the classification accuracy of the training data when using the forward algorithm and one-class classification, while Table 3.2 shows the classification accuracy when using the forward algorithm and confidence.

There are two accuracy measures shown in each table. The first (final accuracy) shows the accuracy of classification after the maneuver is complete. In other words, if the classifier was able to classify the maneuver at the end of the trajectory when it had all available information. The second accuracy measure (accuracy over time) shows what percent of time during the maneuver that the maneuver was correctly classified. For example if the maneuver was classified correctly for 20 out of 25 seconds, the trajectory would have 80% accuracy over time. The percentages for accuracy over time in Tables 3.1 and 3.2 are the mean accuracy over time across the data set.

Both classification schemes had 100% final accuracy in correctly identifying known Maneuvers 1 - 4. No maneuver was classified 100% of its flight time for either classification scheme. However, both classifiers showed very high accuracy over time. High overall accuracy is expected when evaluating with training data.

Table 3.1: HMM Classification Using One-Class: Training Truth

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	98.21%
2	100%	97.87%
3	100%	98.81%
4	100%	98.73%

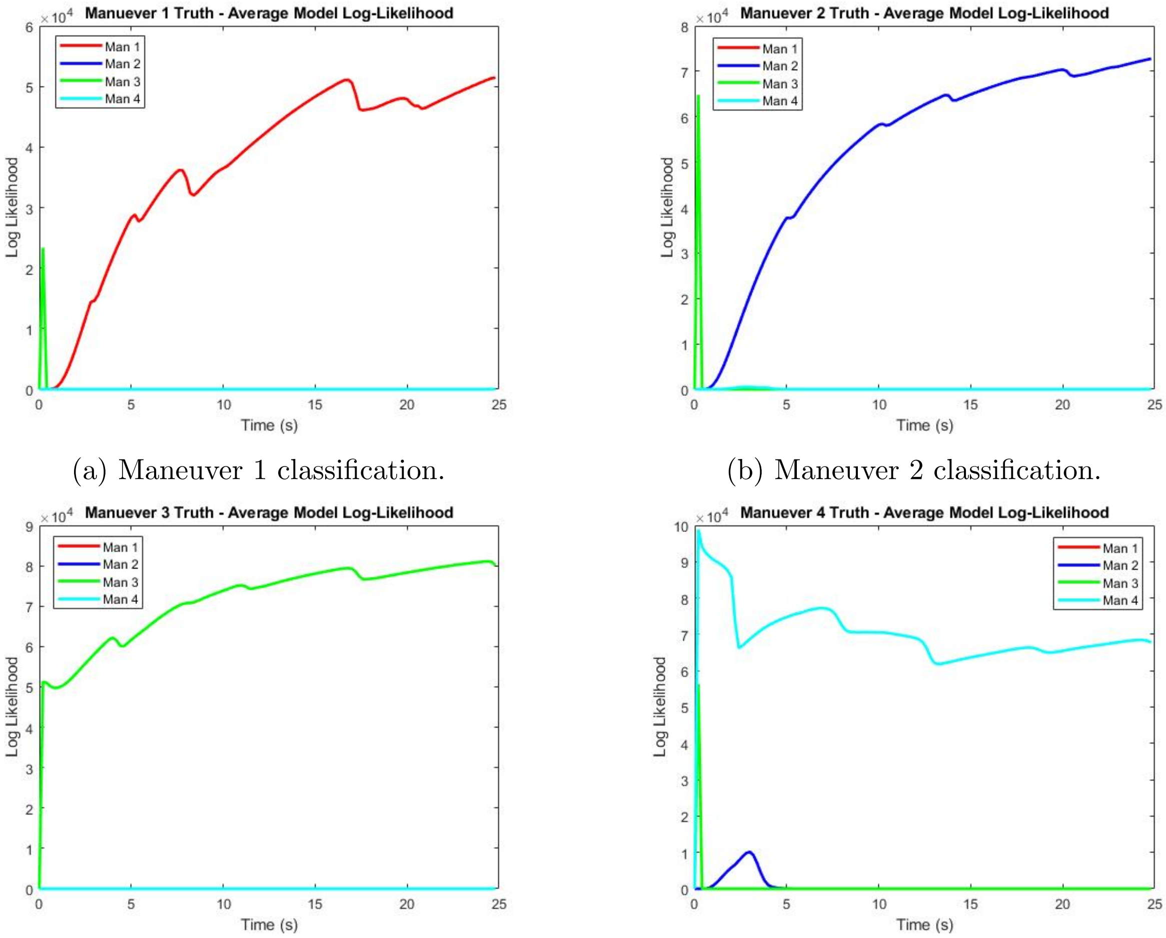
Figure 3.7 shows the average log-likelihood for each of the models throughout the classification process. The average log-likelihood at each time step is taken across the entire data set to produce these results. Figure 3.7a shows the log-likelihood of each HMM over

time when UAV states from Maneuver 1 are input into the classifier. Figure 3.7b shows log-likelihoods when Maneuver 2 observations are input, Figure 3.7c displays results from Maneuver 3 observations, and 3.7d shows Maneuver 4 observations. For figures of individual classification runs rather than the average refer to Appendix B.

Overall, Figure 3.7 shows that each model has a high log-likelihood when presented with observations from its corresponding maneuver, while the other models generally maintain low log-likelihoods. In some cases, λ_3 has an initial high log-likelihood in the first few instances of the maneuver, but the likelihood quickly declines. In Figure 3.7d, λ_2 has a high likelihood in the initial stages of Maneuver 4 (although in comparison to λ_4 it is still relatively low).

Table 3.2: HMM Classification Using Confidence: Training Truth

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	98.21%
2	100%	97.47%
3	100%	98.81%
4	100%	95.24%



(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

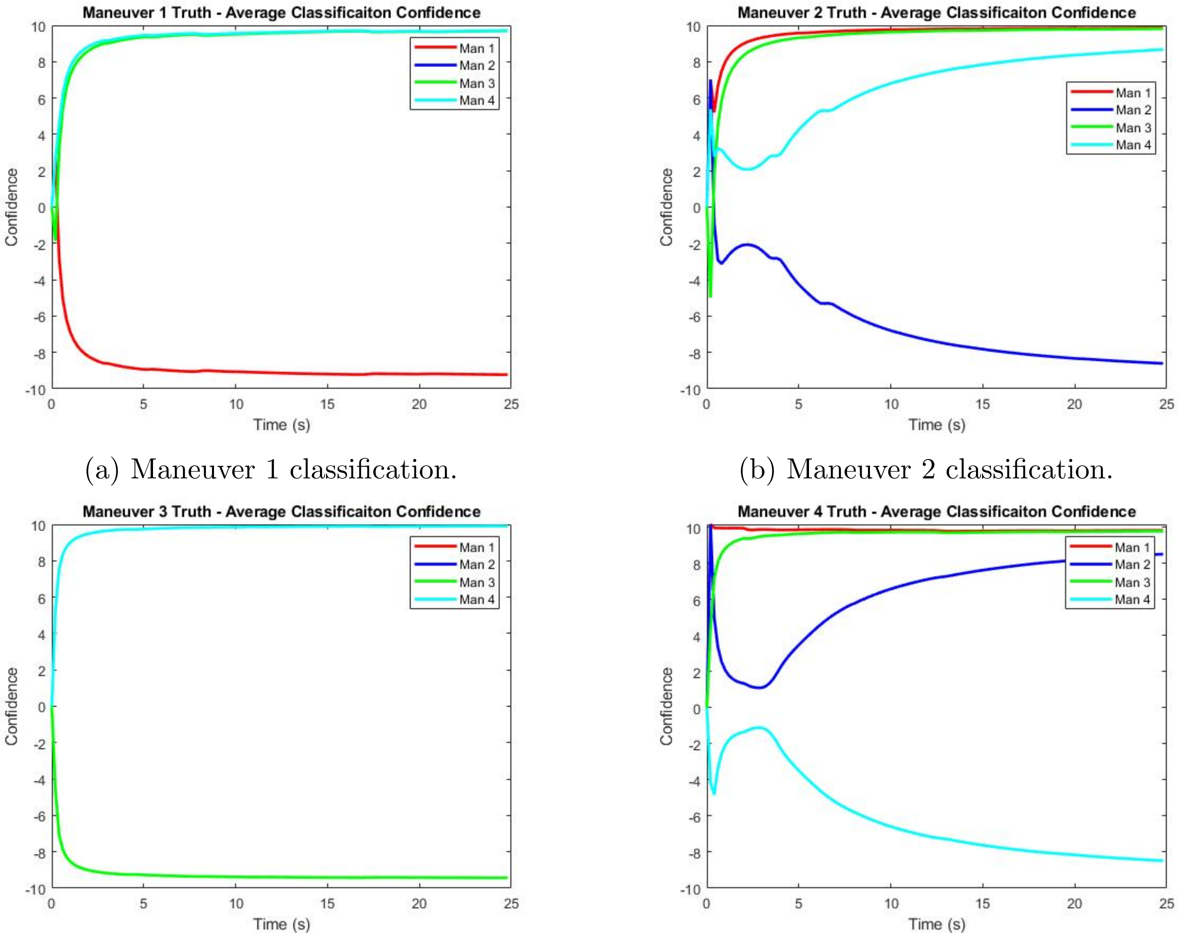
(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure 3.7: Average log-likelihood of each model over time: training data classification.

Figure 3.8 shows the confidence of each of the models throughout classification. Like Figure 3.7, Figure 3.8a refers to confidence of classification when observations from Maneuver 1 are presented, while Figure 3.8b uses observations from Maneuver 2, Figure 3.8c uses observations from Maneuver 3, and Figure 3.8d uses observations from Maneuver 4. As described in Section 3.4.1, a confidence value of less that -2 shows high confidence in a classification. As the confidence value of a model becomes more negative, it shows that the HMM has an even higher ratio of likelihood when compared to its counterparts and therefore, has higher confidence in its classification. Both λ_1 and λ_3 maintain high confidence in their classification throughout the entire maneuver.

λ_2 and λ_4, however, both display an initial uncertainty in the classification. These lower confidence values represent that both models find it likely the observation sequence could correspond to their respective models. The confusion may be caused by similarities in the maneuvers during the initial stages of flight. Recall from Sections 2.2.2 and 2.2.4, the position, velocity, and acceleration for each maneuver were shown. Figures 2.7c and 2.11c in particular show that Maneuvers 2 and 4 have similar initial accelerations for the first 5 -10 seconds of their maneuvers. As the maneuver is performed and the maneuvers lose their similarities, the confidence in classification increases to the point that both are highly confident.



(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure 3.8: Average confidence for training data classification.

Test Data

Tables 3.3 and 3.4 show the classification results for test data using one-class classification and the confidence measure respectively. This test data originates from the truth data set of the UAV. Both one-class and confidence classification have 100% final accuracy and high mean accuracy over time. These results are helpful in ensuring that the trained models are not overly fit to the training data and that both classification schemes have high accuracy.

The tables also contain additional results for an unknown maneuver. The unknown maneuver, Maneuver 5, has a final accuracy of 100% for both classifiers – meaning that both classifiers identified the maneuver as unknown at the end of its trajectory. Furthermore, the accuracy over time for each classifier was high (98.12% for one-class and 96.09% for confidence) indicating that for the majority of the time the UAV was performing Maneuver 5, the classifiers determined it was unknown either because no models exceeded the log-likelihood threshold for one-class classification or because every model had low confidence in its classification.

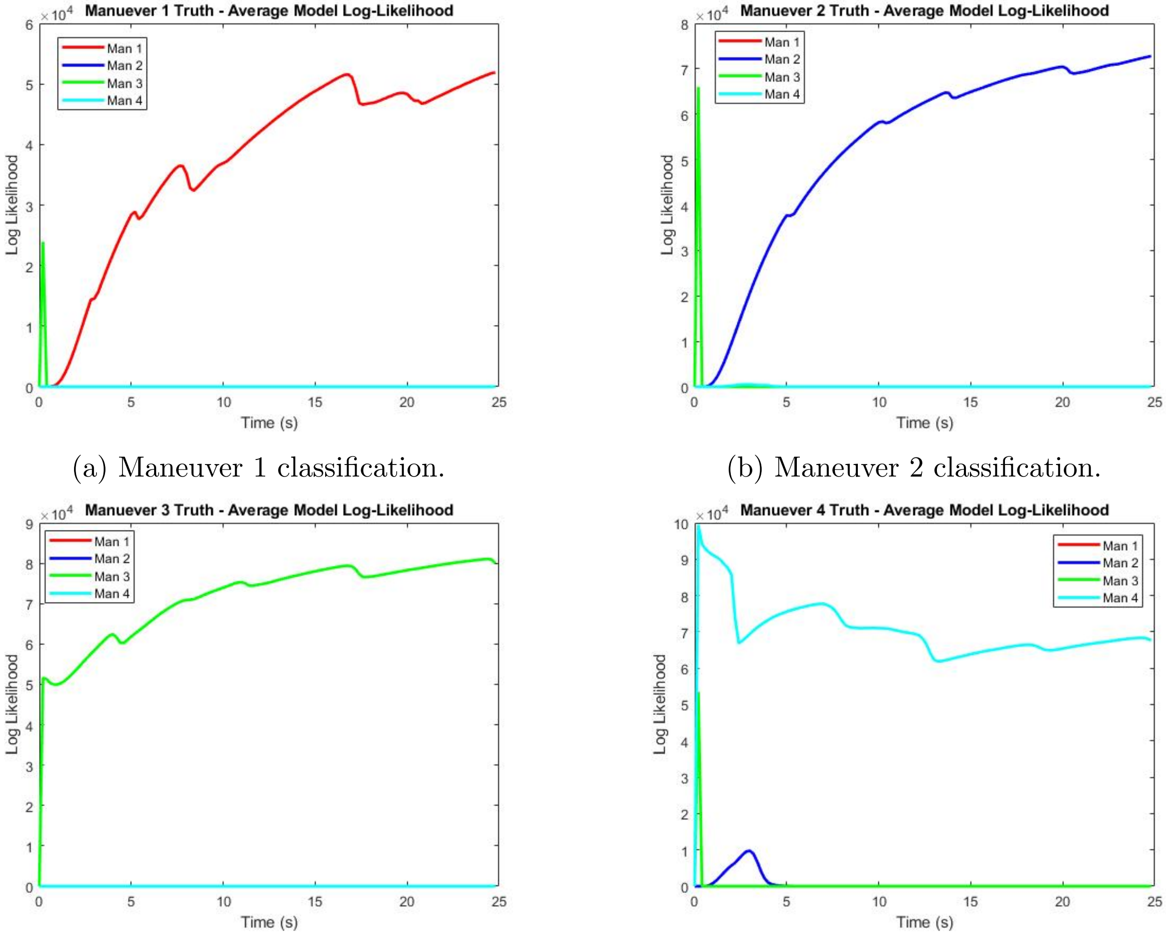
Table 3.3: HMM Classification Using One-Class: Test Truth

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	98.21%
2	100%	97.86%
3	100%	98.81%
4	100%	98.76%
5 (Unknown)	100%	98.12%

Table 3.4: HMM Classification Using Confidence: Test Truth

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	98.21%
2	100%	97.43%
3	100%	98.81%
4	100%	95.38%
5 (Unknown)	100%	96.09%

Figure 3.9 shows the average log-likelihood of each model throughout the classification process. Once again, each model has a high likelihood when input observations from its corresponding maneuver. The overall behavior of each model is very similar to the behavior when testing the training data. Appendix B displays the log-likelihood of individual test data maneuvers rather than the average.



(a) Maneuver 1 classification.

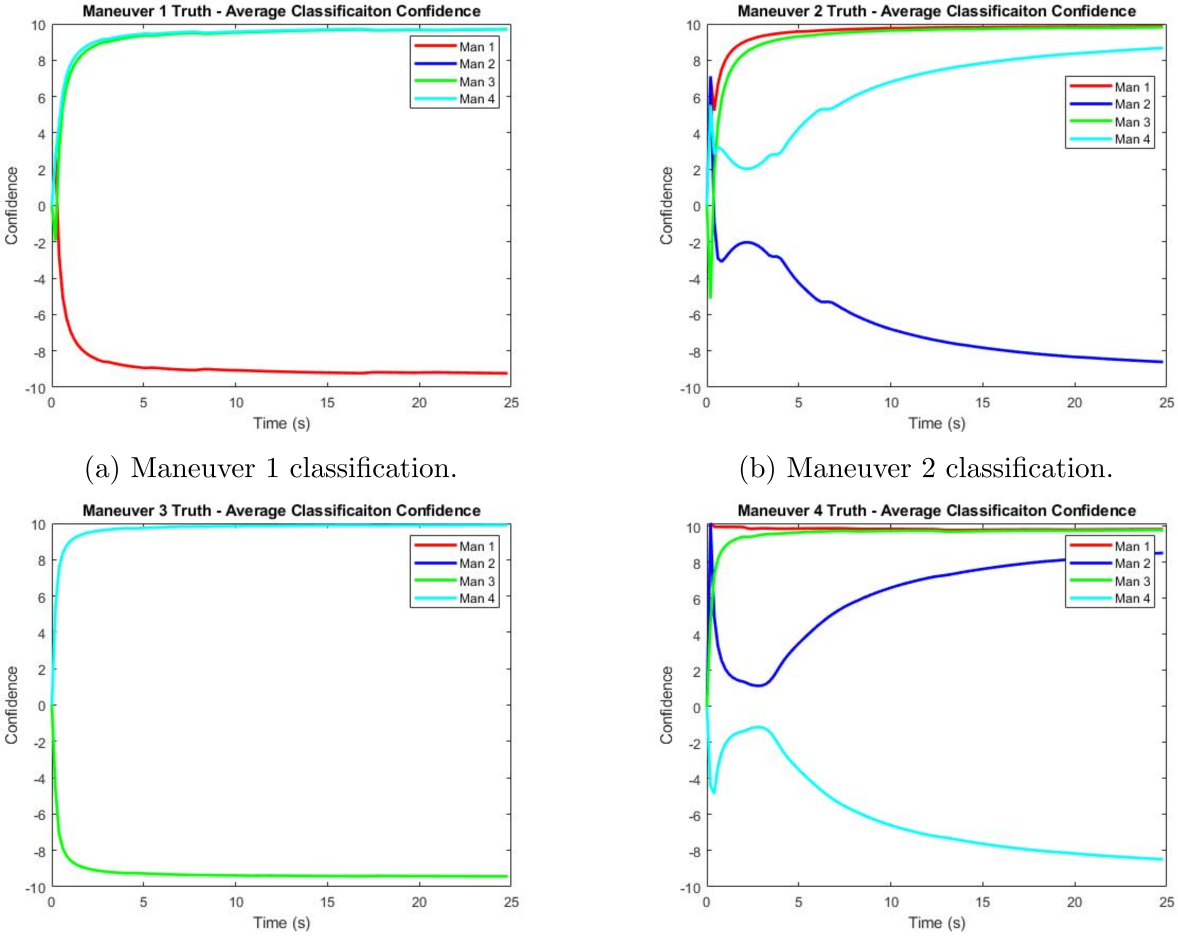
(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure 3.9: Average log-likelihood of each model: test data.

Figure 3.10 shows the average confidence of each model during classification. Both λ_1 and λ_3 are highly confident in their classification throughout the maneuver, while λ_2 and λ_4 are less confident in the initial stages of the maneuver. Later, both are able to find greater confidence.



(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

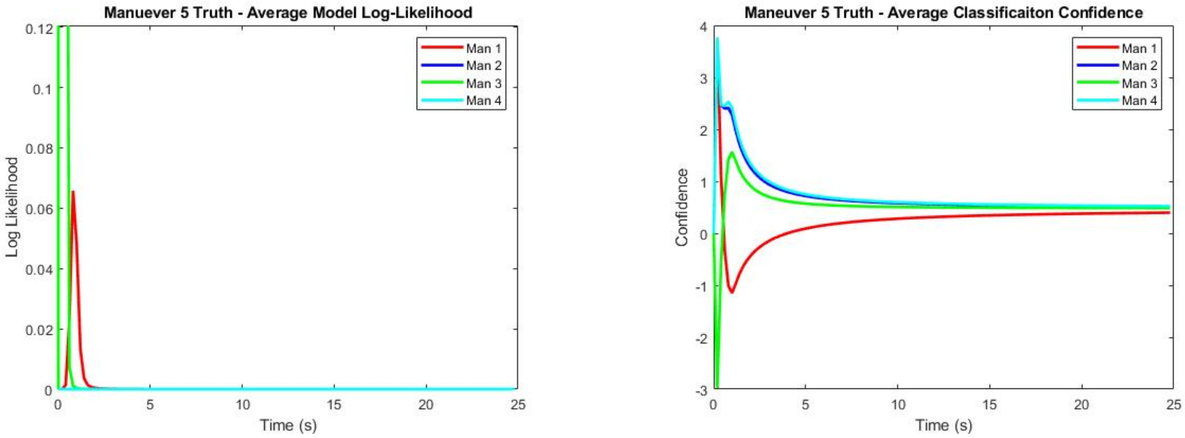
(d) Maneuver 4 classification.

Figure 3.10: Average confidence for testing data classification.

Maneuver 5 is used as an unknown maneuver as a way to test each classifiers' ability to discriminate between a known and unknown trajectories. Figure 3.11 shows the average log-likelihood of each model and the confidence of each model when receiving UAV states recorded during Maneuver 5. Figure 3.11a shows that although λ_3 had a high initial log-likelihood, it quickly drops off and remains low like the other models. For most of the maneuver, no model exceeds the threshold log-likelihood set by the one-class classifier, so the maneuver is classified as unknown.

In Figure 3.11b, there is some initial confidence that λ_3 is the the correct classification, but the metric quickly changes to not confident in the classification. λ_4 and λ_2 initially have high confidence that they are not the the correct classification. This is because λ_3 and

λ_1 have higher log-likelihoods at the beginning of the trajectory, while λ_4's and λ_2's log-likelihood remained low in comparison throughout. As the maneuver continues, however, all model log-likelihoods remain low. This creates a low confidence score for all models, because their log-likelihoods are similar. Essentially, there is no distinction between models. As a result, each model's confidence score is greater than -1, so the maneuver is classified as unknown.



(a) Maneuver 5 Log-likelihood.

(b) Maneuver 5 Confidence.

Figure 3.11: Average log-likelihood and confidence of each maneuver's HMM when presented with Maneuver 5 (unknown maneuver).

3.4.3 Classification Results: EKF

The next section discusses the classification results when utilizing Extended Kalman Filter estimates of UAV states rather than maneuver truth data to test the classifiers. The section is divided into a training and testing data section, but it should be noted that the training data section is not testing the true training data of the models. Rather, it is the original training data set after being estimated using the Extended Kalman Filter from Section 2.3. Likewise, the test data section shows the results from classification using the EKF estimates of the original test data set. Note that the models were not re-trained on estimates of the reference data. Rather, this section serves as an evaluation whether HMMs trained on truth data can then be used to classify estimated data.

Training Data

The accuracy of the classifiers when using EKF estimates of the training data is shown in Tables 3.5 and 3.6. For both classifiers, each maneuver model maintains 100% final classification accuracy. However, the mean accuracy over time decreases for both classifiers. The greatest drop in mean classification accuracy over time is when using the one-class classifier to classify Maneuver 1. There is a decrease in accuracy over time for both classifiers when classifying Maneuvers 1 and 4.

It is not surprising that accuracy in classification drops as the observations become more noisy and contain state estimation errors. Both classifiers are capable of correctly identifying a maneuver for the majority of its flight indicating they are both robust and capable of classifying estimated data.

Table 3.5: HMM Classification Using One-Class: Training EKF

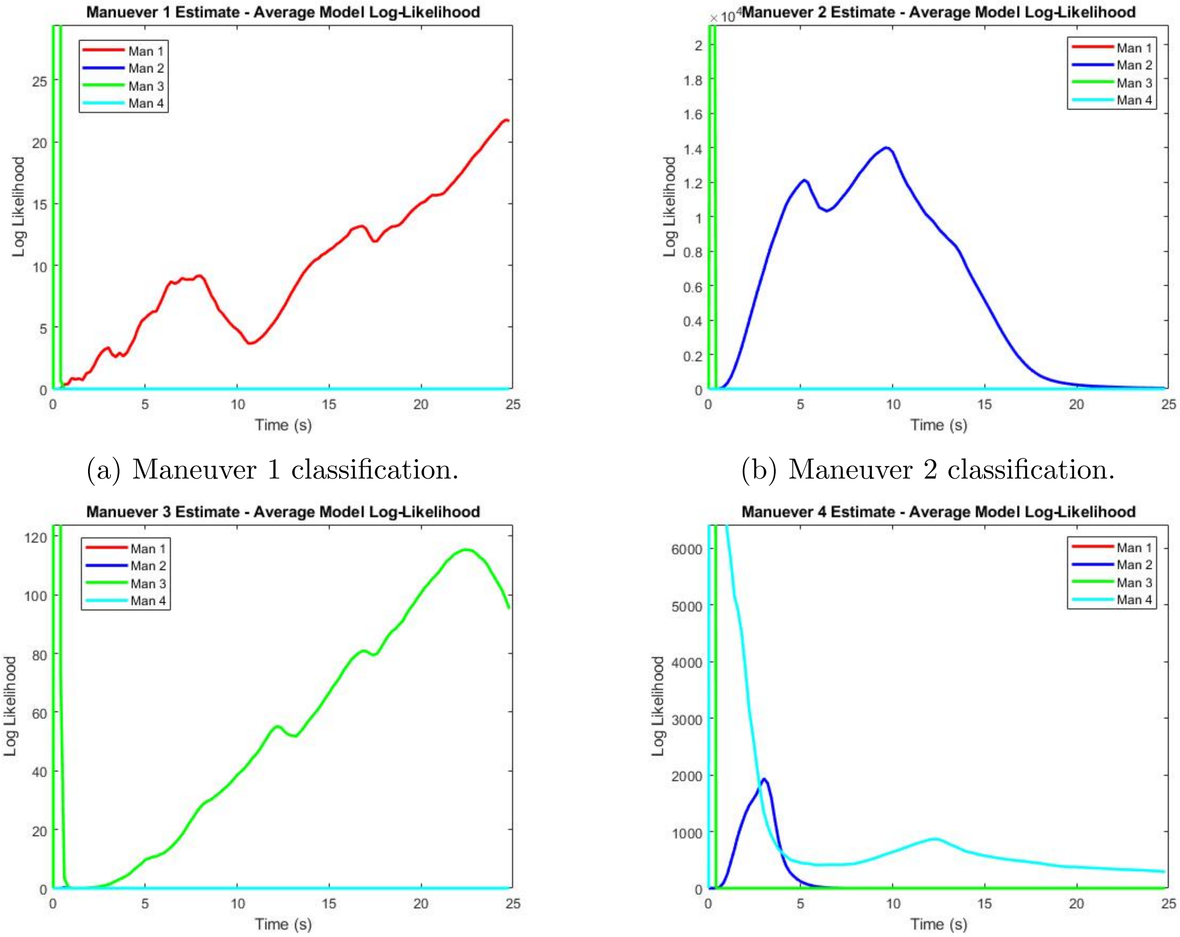
Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	76.45%
2	100%	97.68%
3	100%	94.29%
4	100%	89.40%

Table 3.6: HMM Classification using Confidence: Training EKF

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	86.94%
2	100%	97.64%
3	100%	96.04%
4	100%	80.83%

Figure 3.12 shows the log-likelihood of each model when presented with EKF estimates of the training data maneuvers. Compared to Figure 3.9, the models have a much lower log-likelihood. However, the distinction between the correct maneuver's model is still clear. Maneuver 4's HMM has a lower log-likelihood than Maneuvers 2's HMM for some portions

of flight. This lead to misclassification as Maneuver 2, which contributed to Maneuver 4's lower mean accuracy over time.



(a) Maneuver 1 classification.

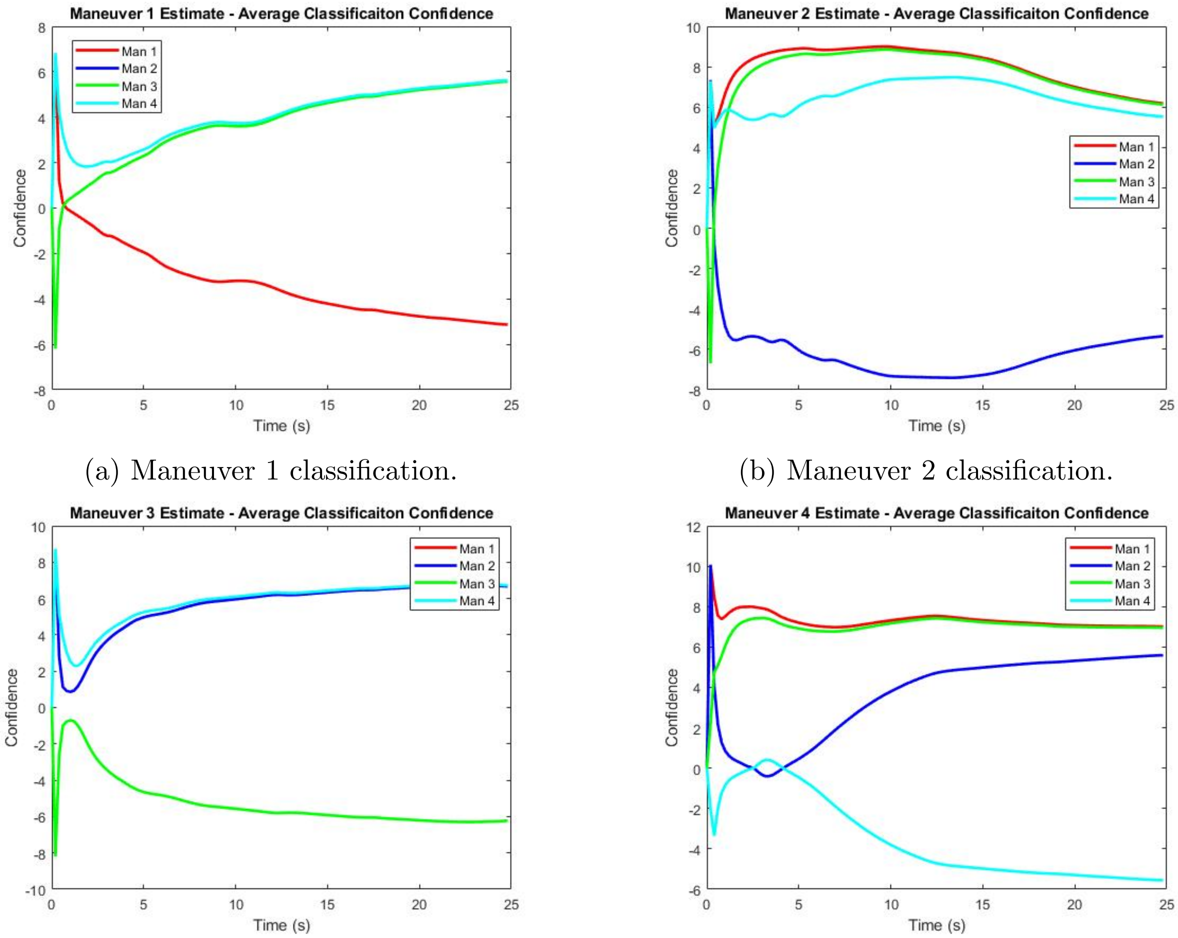
(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure 3.12: Average log-likelihood of each HMM when classifying training data estimates.

The average confidence of each model is displayed in Figure 3.13. The correct model takes more time to reach high confidence in its classification and generally has lower confidence when compared to classification with truth data. However, each model still reaches the "high confidence" score. λ_4 takes the longest amount of time to have high confidence, which coincides with its low accuracy over time when using the confidence measure. Note that Maneuver 4 is never misclassified as Maneuver 2 when using the confidence measure classifier. Both models begin to have confidence between -1 and 1, which leads to an "unknown" classification. Eventually, Maneuver 4's confidence decreases to less than -1, and the maneuver is correctly classified as Maneuver 4.



(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

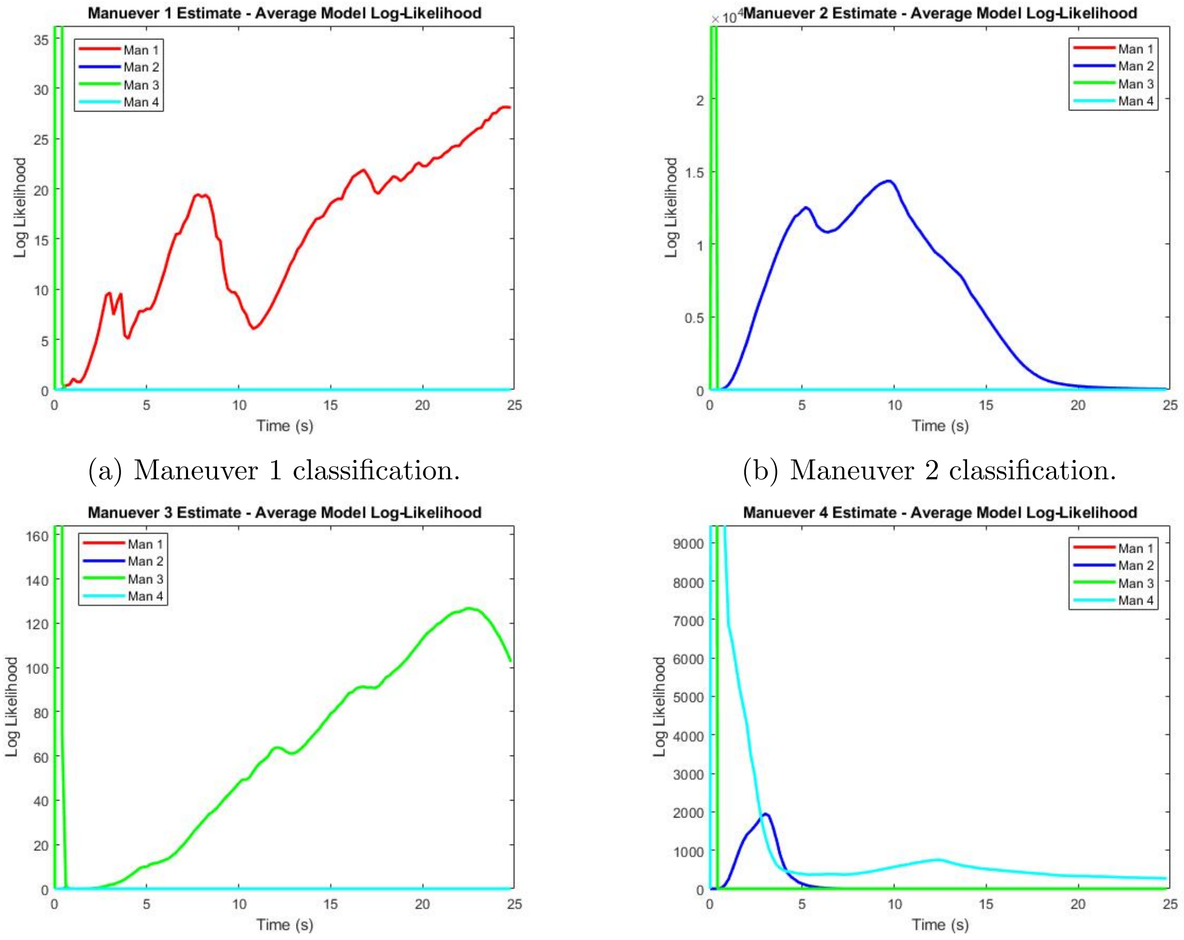
(d) Maneuver 4 classification.

Figure 3.13: Average confidence of each HMM when classifying estimates of training data.

Test Data

Tables 3.7 and 3.8 show the accuracy of the classifiers when using EKF estimates of the test data. Once again, both classifiers maintain a 100% final accuracy for each maneuver and also 100% final accuracy for identifying the unknown maneuver. The classification accuracy over time remains lower for both classification schemes than when using truth, particularly when classifying Maneuvers 1 and 4.

The average log-likelihood of each model throughout the classification process when using EKF estimates of test data is shown in Figure 3.14. Overall, the results are similar to that of Figure 3.12. The log-likelihood of the models are lower than when using truth data, but the models are clearly still capable of classification.



(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure 3.14: Average log-likelihood of each HMM when classifying estimates of test data.

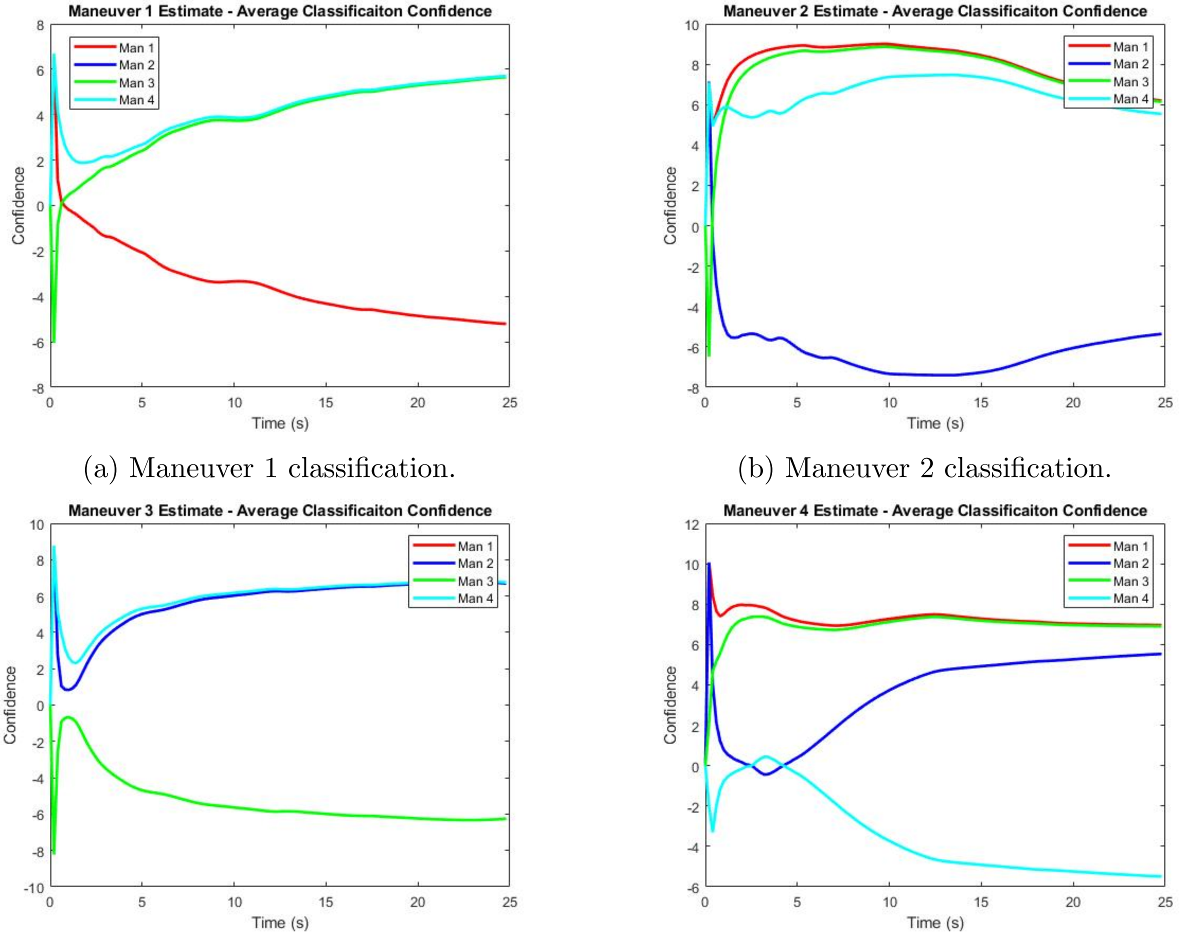
Table 3.7: HMM Classification Using One-Class: Test EKF

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	77.87%
2	100%	97.70%
3	100%	94.40%
4	100%	89.00%
Unknown	100%	94.21%

Figure 3.15 displays the confidence of each model in its classification when using EKF estimates of the original test data. Maneuver 1 has low initial confidence and is confused with Maneuver 3 before distinguishing itself causing a lower accuracy over time. Initial incorrect classifications of Maneuver 2 also occurs. Maneuver 4 is initially classified correctly, then loses confidence and is classified as unknown, before once again being classified as Maneuver 4 for the rest of the trajectory.

Table 3.8: HMM Classification Using Confidence: Test EKF

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	87.88%
2	100%	97.64%
3	100%	96.06%
4	100%	80.60%
Unknown	100%	91.59%



(a) Maneuver 1 classification.

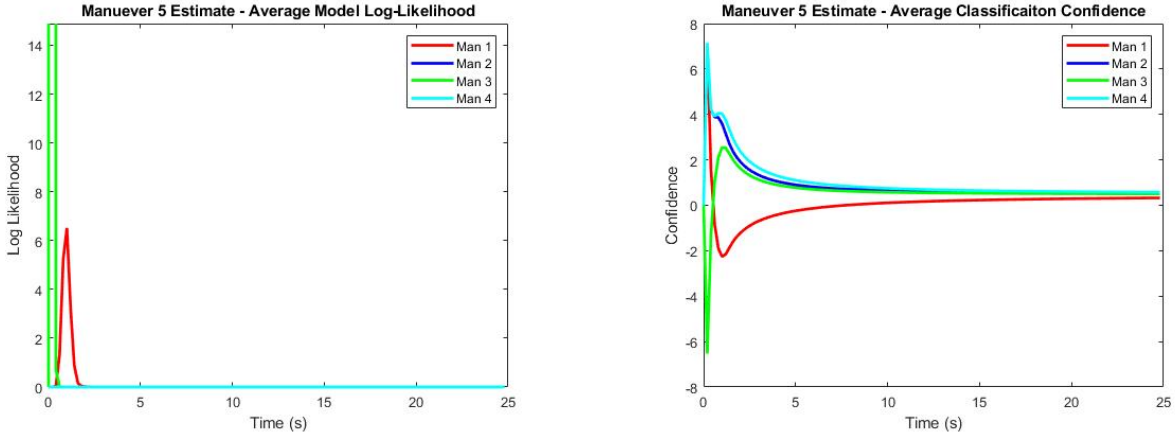
(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure 3.15: Average confidence of each HMM when classifying estimates of test data.

The average log-likelihood and confidence of each model when presented with EKF estimates of an unknown maneuver (Maneuver 5) are shown in Figure 3.16. λ_3 has a high log-likelihood in the initial 1 second of the maneuver, but then the likelihood quickly drops off. λ_1 also briefly has a high log-likelihood. For most of the maneuver however, no models have high log-likelihood. Like in Section 3.4.2, the confidence in each model is low for most of the maneuver, resulting in an unknown classification.



(a) Maneuver 5 Log-likelihood.

(b) Maneuver 5 Confidence.

Figure 3.16: HMM classification of Maneuver 5 (unknown maneuver) EKF estimates.

Appendix C shows the individual log-likelihood and confidence results for each of the EKF estimates of training and testing.

3.4.4 Final Classification Scheme

The one-class classifier and confidence classifier had similar levels of accuracy for each data set they were tested on – likely because both utilized the original model log-likelihood from the forward algorithm. Overall, the confidence measure classifier was chosen as the classifier used in this thesis. Because it compares the log-likelihood of each model through Equation (3.16), it is more likely to acknowledge ambiguities caused when multiple models have a high log-likelihood (like when classifying Maneuver 4 with EKF data). It also had slightly better performance when tested using EKF data.

3.4.5 Classification With Maneuver Changes

The confidence measure classifier was then tested on several scenarios in which the UAV changes from an initial maneuver it's performing mid-flight to a different final maneuver. The purpose of this situation is to test if the classifier is robust to attempts to confuse it. The same HMMs used in previous classification sections are used in this section as well. Ideally,

Table 3.9: Maneuver Changes

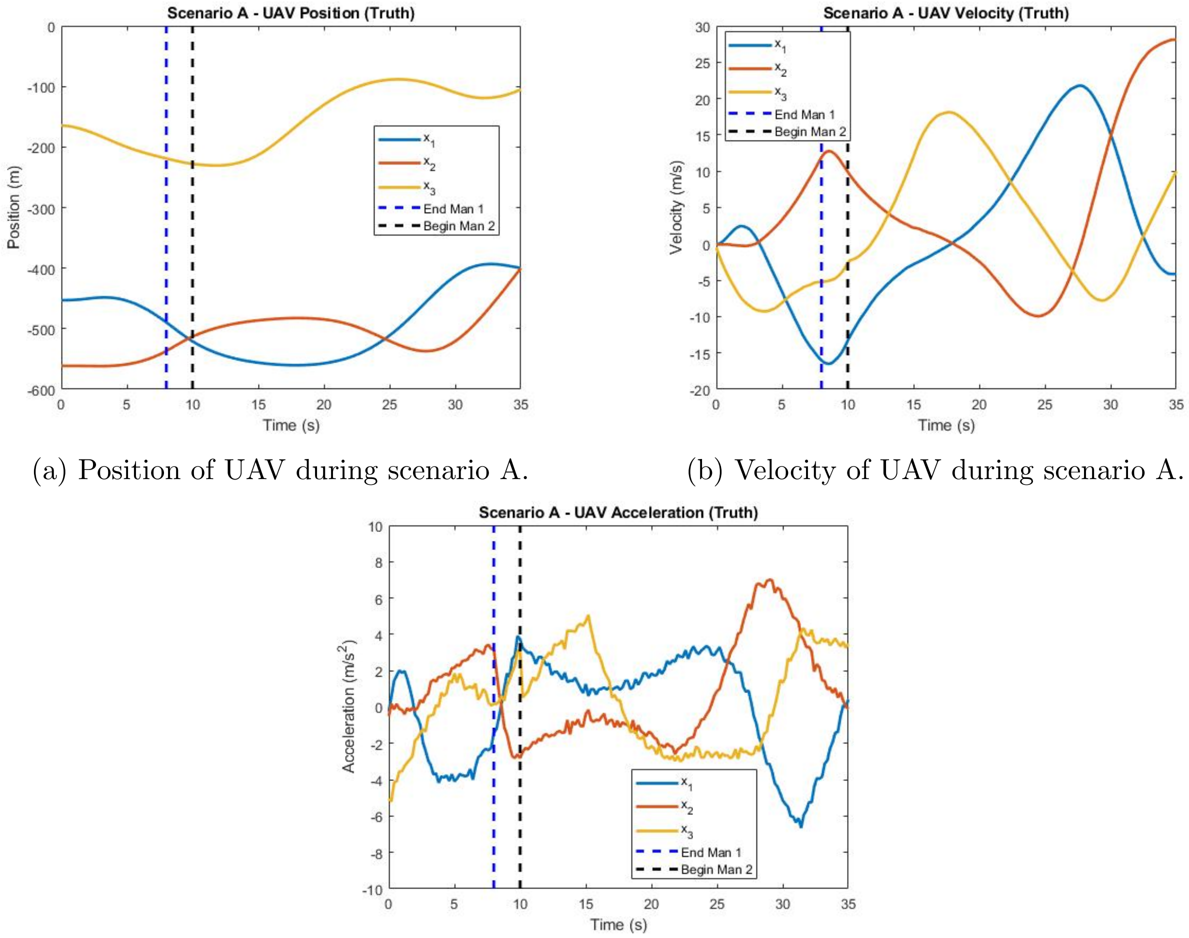
Scenario	Initial Maneuver	Final Maneuver
A	1	2
B	2	4
C	3	1
D	4	3

the confidence measure classifier would be able to correctly classify the initial maneuver as it's performed and then likewise, classify the final maneuver after the transition between the two.

The maneuver changes are shown in Table 3.9. The maneuver change flight was executed in three ways: transitioning maneuvers after the first 3 seconds of initial maneuver, transitioning 8 into the initial maneuver, and transition after 13 seconds of the initial maneuver. The transition period from the initial to final maneuver is 2 seconds, which allows for smoother flight and prevents unrealistic changes in acceleration.

An example of scenario A's maneuver change is shown in Figure 3.17, which shows the NED position, velocity, and acceleration of the UAV. Maneuver 1 is simulated for the first 8 seconds of flight. Then, from 8 to 10 seconds, the maneuver commands transition to the start of Maneuver 2. The blue dotted line marks the end of Maneuver 1, and the black dotted line denotes the beginning of Maneuver 2. Then, Maneuver 2 performed for its flight time of 25 seconds.

To create test data for the classifier, 243 trajectories of each scenario and transition time were created (243 trajectories for scenario A with the transition happening at $t = 3$, for example). The initial and final position of each of trajectory was varied and process noise was added to acceleration to simulate real world disturbances. Like before, classification accuracy of the classifier is tested on both simulated truth data and EKF estimates of that same truth data.



(a) Position of UAV during scenario A.

(b) Velocity of UAV during scenario A.

(c) Acceleration of UAV during scenario A.

Figure 3.17: UAV states throughout Scenario A when maneuver changes at 8 seconds

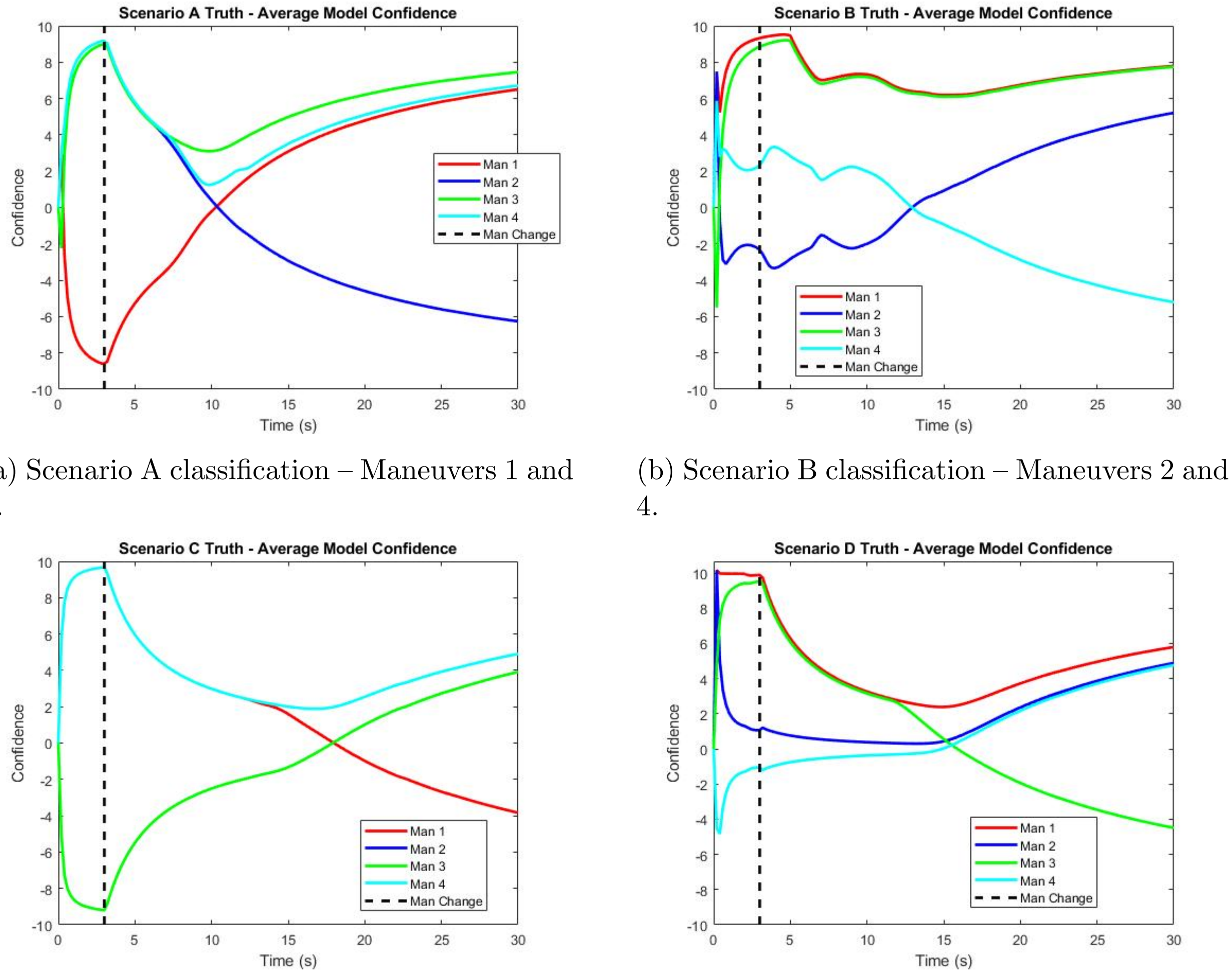
Truth Data

The following section describes the classification results for truth data of each scenario and at each transition time. For both the initial and final maneuver, the final accuracy and mean accuracy over time are shown. The final accuracy shows the accuracy of classification when all available information of the maneuver is known. In the case of the initial maneuver, this means the accuracy of classification at the beginning of the maneuver transition point. The mean accuracy over time shows the percentage of the total trajectory time that the maneuver classification was correct.

Table 3.10: Classification accuracy when maneuver transitions at 3 seconds - Truth Data.

Scenario	Final Accuracy		Mean Accuracy Over Time	
	Initial Maneuver	Final Maneuver	Initial Maneuver	Final Maneuver
a	100%	100%	91.26%	73.93%
b	100%	100%	87.70%	59.88%
c	100%	100%	94.43%	40.21%
d	18.93%	100%	55.92%	49.12%

Table 3.10 shows the accuracy of the confidence measure classifier when a maneuver transitions 3 seconds into flight. The initial maneuver has 100% final classification accuracy for every scenario, except scenario D. Scenario D begins with Maneuver 4, which, as seen before, has an uncertain classification in the initial moments of the trajectory. The final maneuver has 100% final classification for all scenarios. The mean accuracy over time is high for the initial maneuver in all scenarios (except scenario D). However, the final maneuver mean accuracy over time is considerably lower. Figure 3.18 shows the confidence of each model throughout the scenarios. After the transition point, which is denoted with a dotted black line, the confidence of the correct model transitions slowly from low confidence to high.



(a) Scenario A classification – Maneuvers 1 and 2.

(b) Scenario B classification – Maneuvers 2 and 4.

(c) Scenario C classification – Maneuvers 3 and 1.

(d) Scenario D classification – Maneuvers 4 and 3.

Figure 3.18: Average confidence of each HMM when classifying changing maneuvers with truth data (transition at 3 seconds.)

Recall that the forward algorithm (Section 3.4.1), which is used to find the HMM log-likelihood that is incorporated into the confidence measure, finds the likelihood of the entire observation sequence given the model. This means that all past data, including data from the currently initial maneuver, is incorporated into the log-likelihood of the model having produced the total trajectory. When there is no data from a different maneuver included in the observation sequence, the maneuvers are classified quickly. However, with data from a different maneuver, the observation sequence needs more data associated with the correct maneuver before model log-likelihood is high enough to result in correct classification. With a transition between two maneuvers, there needs to be an excess of “correct” data before the final maneuver can be correctly classified.

This is reflected in Figure 3.19, which shows the average log-likelihood of each model throughout scenario A. While the initial maneuver’s model (Maneuver 1) has a high average log-likelihood, the final maneuver (Maneuver 2) only begins to have a log-likelihood of near the same magnitude towards the end of the total trajectory. This contributes to the low confidence in the final maneuver’s model for the majority of its execution, as seen in Figure 3.18.

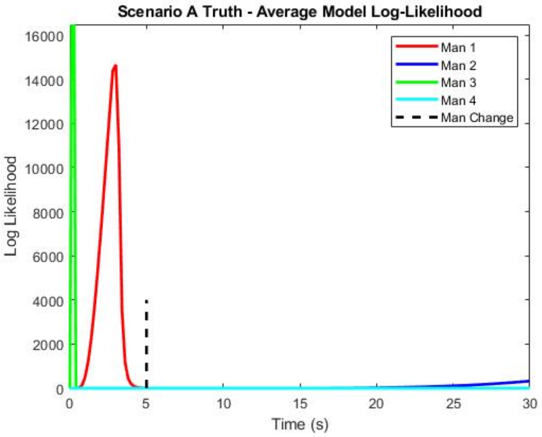


Figure 3.19: Average log-likelihood of each model throughout scenario A.

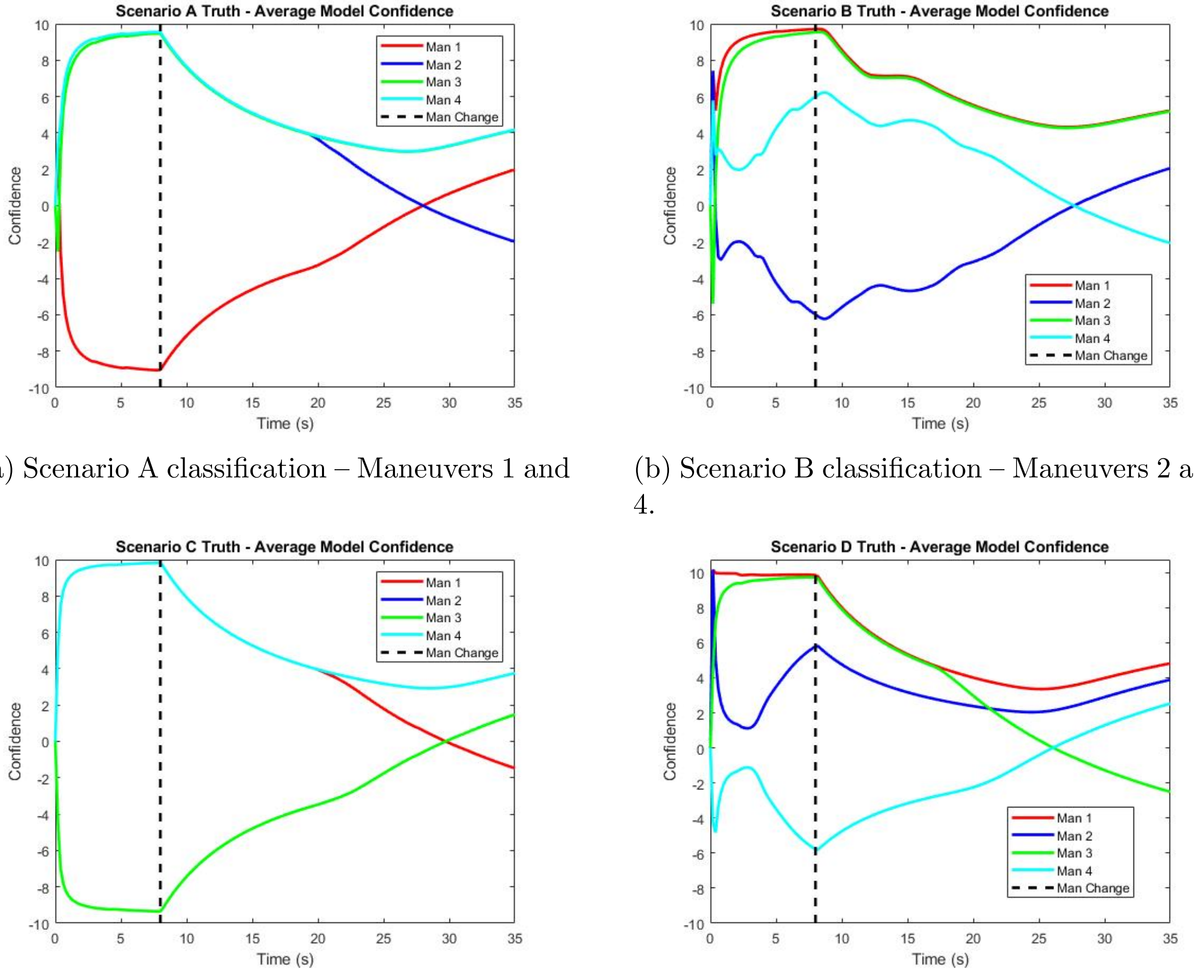
The final accuracy and mean accuracy over time for each scenario when the maneuver transitions at eight seconds is shown in Table 3.11. Both the initial and final trajectory have a high final accuracy, but the final maneuver’s mean accuracy over time is very low compared to that of the initial maneuver. The confidence classifier correctly identifies the correct maneuver in the final moments of the trajectory, if at all.

Figure 3.20 shows the confidence of each maneuver’s model in each scenario. The confidence of the initial maneuver model becomes very negative in the initial stages of the trajectory – indicating high confidence. After the transition point, the initial model classification becomes less confident over time, while the final maneuver model’s confidence grows.

Table 3.11: Classification accuracy when maneuver transitions at 8 seconds - Truth Data.

Scenario	Final Accuracy		Mean Accuracy Over Time	
	Initial Maneuver	Final Maneuver	Initial Maneuver	Final Maneuver
a	100%	100%	95.49%	15.75%
b	100%	100%	93.65%	17.10%
c	100%	100%	97.04%	7.84%
d	100%	100%	88.29%	24.09%

However, this process is lengthy. For most of the final maneuver's time, the maneuver is either misclassified as the initial maneuver or as "unknown".



(a) Scenario A classification – Maneuvers 1 and 2.

(b) Scenario B classification – Maneuvers 2 and 4.

(c) Scenario C classification – Maneuvers 3 and 1.

(d) Scenario D classification – Maneuvers 4 and 3.

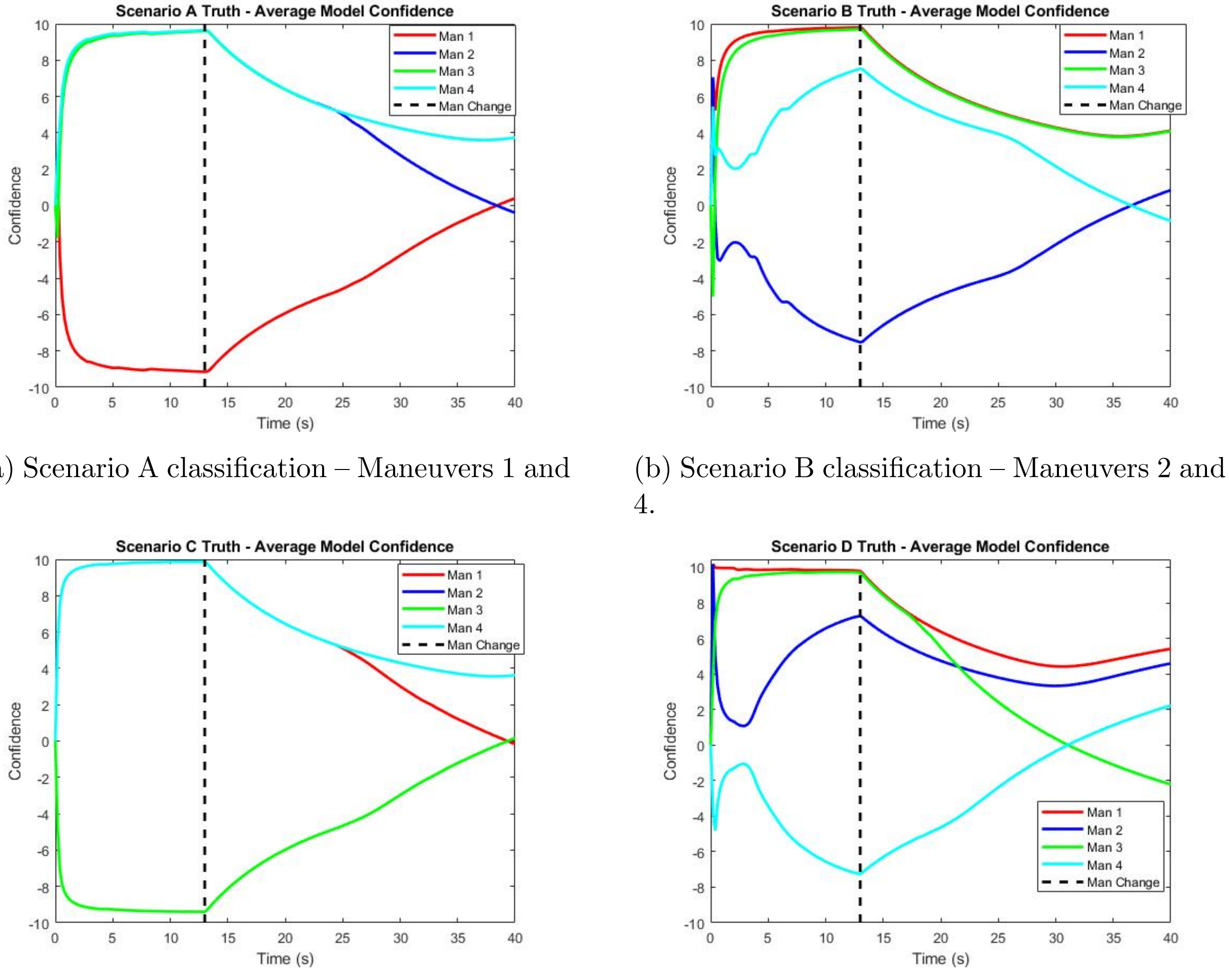
Figure 3.20: Average confidence of each HMM when classifying changing maneuvers with truth data (transition at 8 seconds.)

Table 3.12: Classification accuracy of each scenario when maneuver transitions at 13 seconds - Truth Data.

Scenario	Final Accuracy		Mean Accuracy Over Time	
	Initial Maneuver	Final Maneuver	Initial Maneuver	Final Maneuver
a	100%	0%	97.07%	0%
b	100%	13.58%	95.68%	4.15%
c	100%	0%	98.05%	0%
d	100%	100%	91.81%	22.14%

The final maneuver transition time took place at thirteen seconds. Table 3.12 shows the classification accuracy of the initial and final maneuver. All initial maneuvers' final accuracy and mean accuracy over time are high. However, the final maneuver has extremely low accuracy for both criteria. Scenario A and scenario B both have a final accuracy and mean accuracy over time of 0%.

The average confidence of each HMM in the different scenarios is shown in Figure 3.21. It can be seen that each initial maneuver reaches a high classification confidence by the maneuver transition point. From there, the certainty in the model decreases, while the confidence in the correct model increases. Once again, this change takes place slowly. In most cases, the confidence in the final maneuver's model does not reach the classification point of less than -1 even at the end of the total trajectory.



(a) Scenario A classification – Maneuvers 1 and 2.

(b) Scenario B classification – Maneuvers 2 and 4.

(c) Scenario C classification – Maneuvers 3 and 1.

(d) Scenario D classification – Maneuvers 4 and 3.

Figure 3.21: Average confidence of each HMM when classifying changing maneuvers with truth data (transition at 13 seconds.)

EKF Data

The following section explores the accuracy of the confidence classifier when a maneuver transitions partially through flight. Instead of classification with truth data, the classifier is tested with estimates of that truth data to evaluate how these factors affect accuracy.

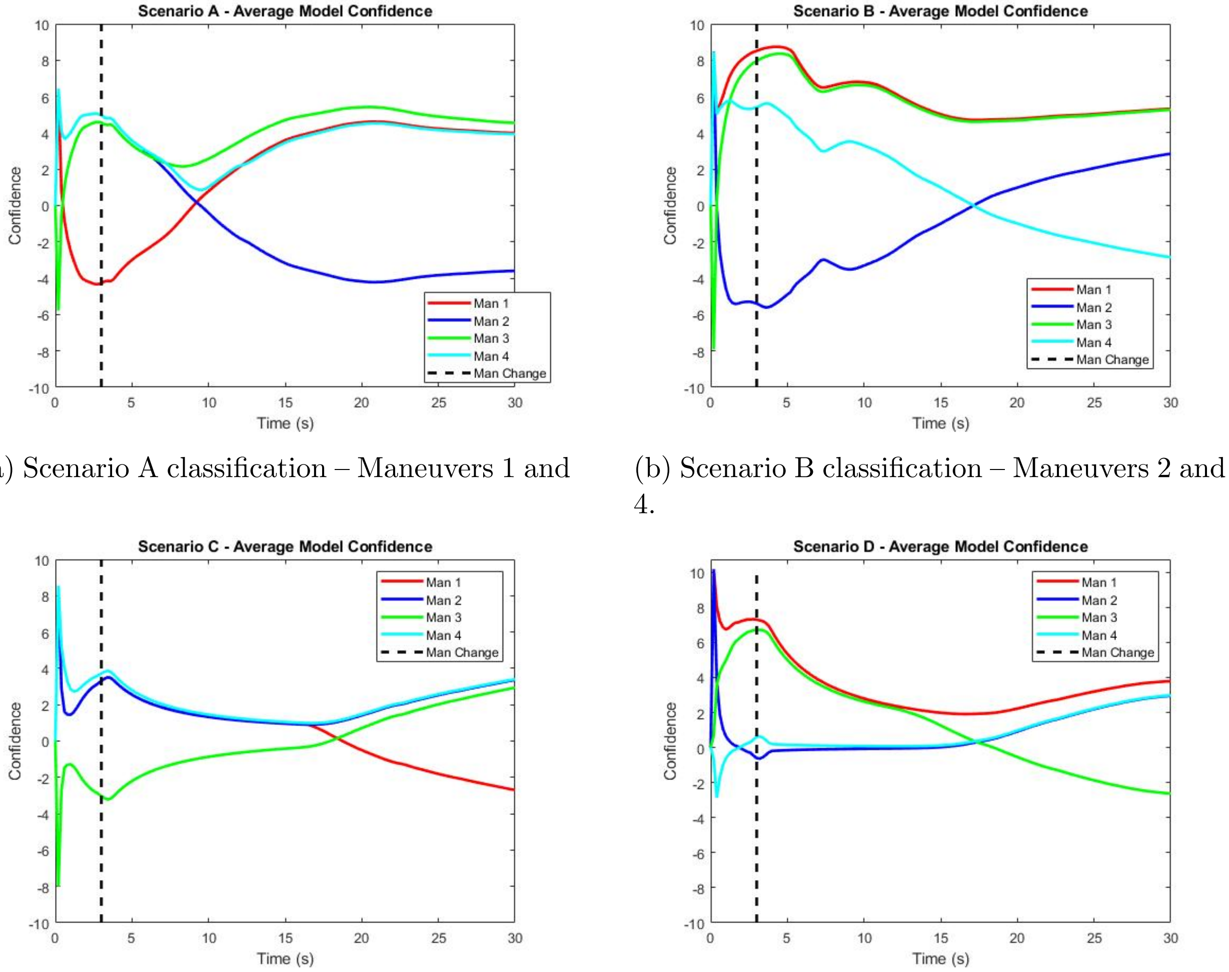
The classification accuracy of the initial and final maneuver when the maneuvers transition at three seconds are found in Table 3.13. Like with truth data, the final accuracy is high for both the initial and final maneuver (except the initial maneuver in scenario D). The

Table 3.13: Classification accuracy of each scenario when maneuver transitions at 3 seconds - Estimated Data.

Scenario	Final Accuracy		Mean Accuracy Over Time	
	Initial Maneuver	Final Maneuver	Initial Maneuver	Final Maneuver
a	99.59%	100%	84.30%	77.29%
b	100%	100%	88.00%	39.59%
c	88.48%	100%	80.75%	34.12%
d	8.23%	100%	19.20%	34.34%

mean accuracy over time is relatively high for the initial maneuver. However, it is quite poor for the final maneuver and slightly lower than when classifying with truth data.

The average confidence of each model is shown in Figure 3.22. The figure shows the slow transition between initial maneuver model confidence and final maneuver model confidence. In most cases, the final maneuver does not have a better confidence measure than the initial maneuver until at least fifteen seconds into it's own flight.



(a) Scenario A classification – Maneuvers 1 and 2.

(b) Scenario B classification – Maneuvers 2 and 4.

(c) Scenario C classification – Maneuvers 3 and 1.

(d) Scenario D classification – Maneuvers 4 and 3.

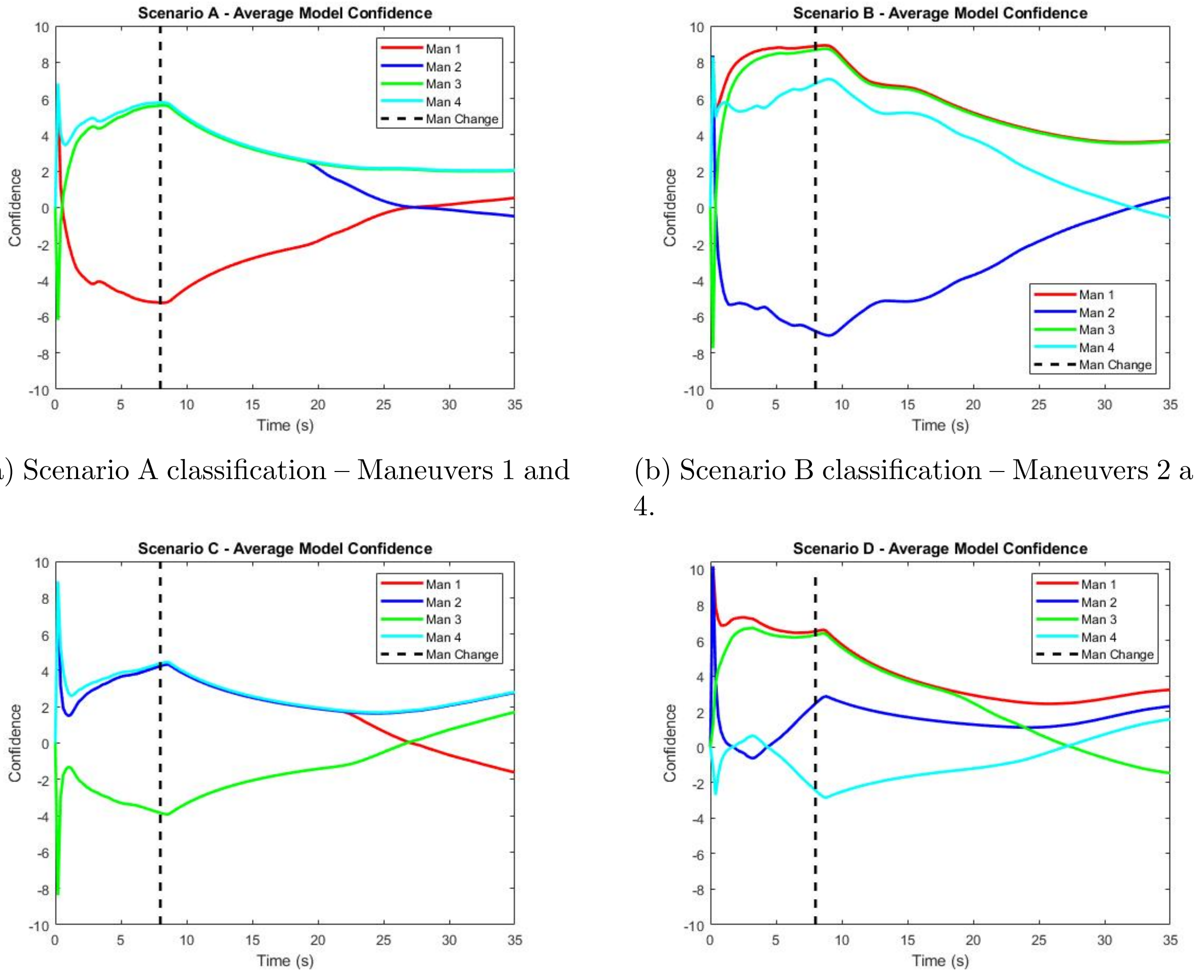
Figure 3.22: Average confidence of each HMM when classifying changing maneuvers with estimated data (transition at 3 seconds.)

Table 3.14 shows the classification results for each scenario when the maneuvers transition at eight seconds. The final maneuver's final accuracy and mean accuracy over time is low for each maneuver. Scenario A and B have a mean accuracy over time near 0%. This effects of a longer period of observation data from a different maneuver, combined with the noise and estimation errors typically found in estimated data, creates an observation sequence that is incredibly difficult for the classifier to correctly identify.

Table 3.14: Classification accuracy of each scenario when maneuver transitions at 8 seconds - Estimated Data.

Scenario	Final Accuracy		Mean Accuracy Over Time	
	Initial Maneuver	Final Maneuver	Initial Maneuver	Final Maneuver
a	100%	1.23%	90.90%	0.06%
b	100%	11.52%	94.02%	0.52%
c	97.12%	99.17%	88.29%	13.65%
d	99.18%	96.71%	48.94%	13.00%

Figure 3.23 shows the average model confidence during each scenario when using estimated data. The initial maneuver's model reaches high confidence before maneuver transition. After the transition, the final maneuver's model eventually has higher confidence than that of the initial maneuver's model, but it takes a considerable amount of time.



(a) Scenario A classification – Maneuvers 1 and 2.

(b) Scenario B classification – Maneuvers 2 and 4.

(c) Scenario C classification – Maneuvers 3 and 1.

(d) Scenario D classification – Maneuvers 4 and 3.

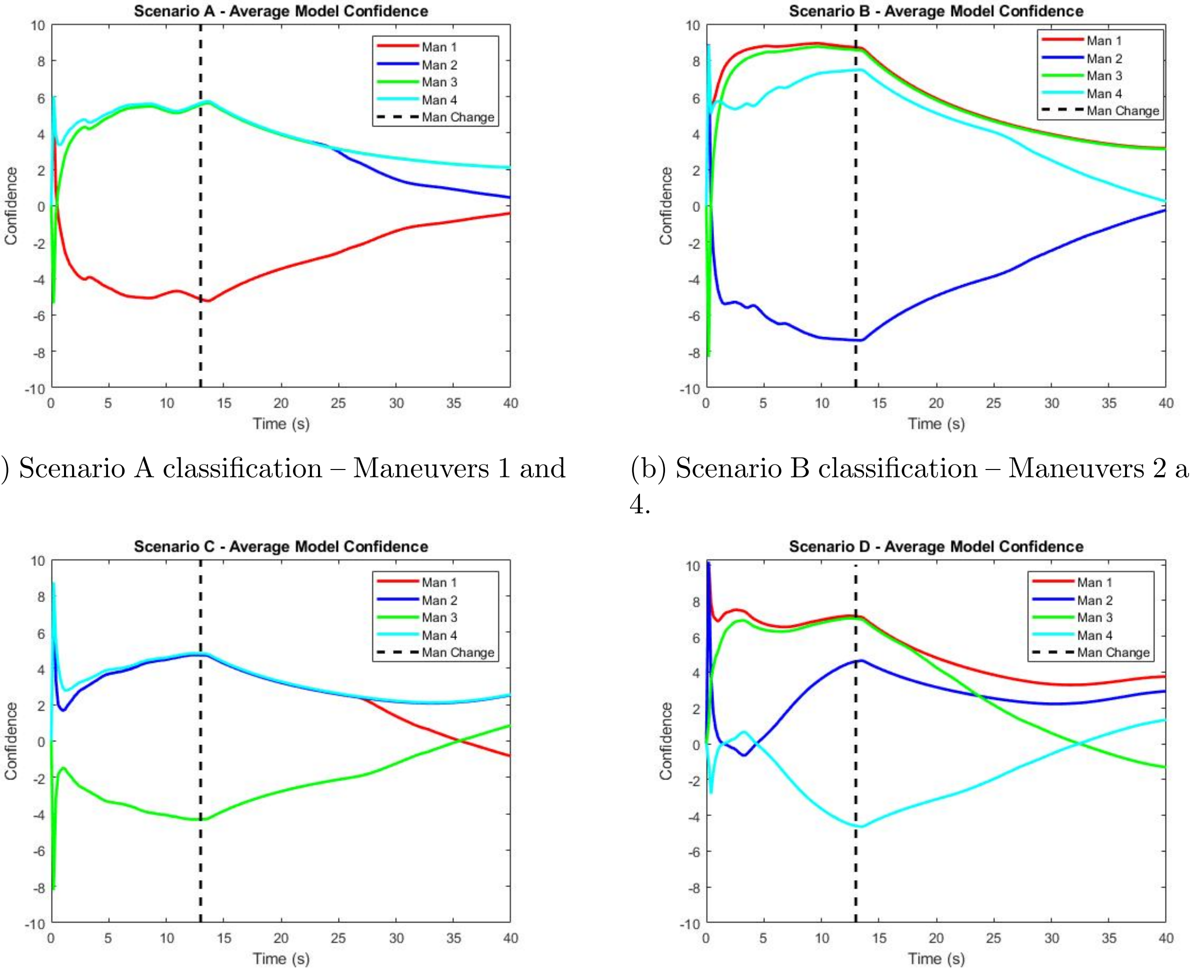
Figure 3.23: Average confidence of each HMM when classifying changing maneuvers with estimated data (transition at 8 seconds.)

The final situation considered is that the maneuver transition occurs at thirteen seconds. Table 3.15 shows the classification results when using estimated data. The final maneuver's accuracy is very low for both final accuracy and mean accuracy over time. Figure 3.24 shows the confidence of each model when using estimated data. Before the maneuver transition, the initial maneuver's HMM reaches high confidence. After the transition point, the initial maneuver's model does slowly change to low confidence. However, for most of the trajectory it remains below -1, which means the final maneuver is incorrectly classified as initial maneuver for most of the time it's performed. While in scenario C and D, the final maneuver

Table 3.15: Classification accuracy of each scenario when maneuver transitions at 13 seconds - Estimated Data.

Scenario	Final Accuracy		Mean Accuracy Over Time	
	Initial Maneuver	Final Maneuver	Initial Maneuver	Final Maneuver
a	100%	0%	93.80%	0%
b	100%	0%	95.87%	0%
c	100%	22.63%	93.67%	1.30%
d	100%	84.77%	66.33%	10.25%

eventually has higher confidence than the initial maneuver, this is not the case for scenario A and B.



(a) Scenario A classification – Maneuvers 1 and 2.

(b) Scenario B classification – Maneuvers 2 and 4.

(c) Scenario C classification – Maneuvers 3 and 1.

(d) Scenario D classification – Maneuvers 4 and 3.

Figure 3.24: Average confidence of each HMM when classifying changing maneuvers with estimated data (transition at 13 seconds.)

Future Improvements For Classification of Maneuver Transitions

Overall, HMM classification performed poorly when classifying the two maneuvers a trajectory transitions between. If the transition between maneuvers occurred early in the trajectory, the classifier was eventually able to correctly identify the final maneuver. When the transition occurred later, often the final maneuver was misclassified as “unknown” or as the initial maneuver. However, even in these cases, the classifier would slowly become less confident in the classification of the initial maneuver and more confident in the final maneuver as more observations of the final maneuver were input into the classifier.

The current classification scheme relies on the forward algorithm to produce the log-likelihood of each model throughout the observation sequence. The forward algorithm incorporates all past data into its log-likelihood output, which can cause a low model log-likelihood when there is an excess of data that comes from a different. This architecture causes the classifier to fail when classifying transitioning maneuvers. As it stands, an excellent way to defeat the current classifier would be for an adversary to change known maneuvers mid-way through flight and “confuse” the classifier.

However, modifications do exist that could increase the classifier’s accuracy when confronted with changing maneuvers. HMMs are often used to interpret ASL sentences from videos or to perform speech recognition on sentences. In both of these examples, there is a transition of models within the observation sequence. With ASL, there is a transition through different signs, while with speech, the phonemes change when speaking [13]. The issue is often solved through concatenating individual models or by use of embedded training. In embedded training, the models are trained using data that contains transitions. The training process allows the models themselves to find the boundaries of the transition points. Future iterations of this classifier could incorporate the embedded training process into its training process to improve results and robustness.

These results also suggest that it may be worthwhile to invest some time in maneuver detection, rather than using HMMs for detection. The HMMs succeed at classification, but detection may be difficult. Copious amounts of non-maneuver data may confuse the classifier and lead to worse overall classification results. A detection aspect of the classifier should be added to the architecture in future work.

3.5 HMM State Sequence

The observation sequence produced by an HMM is the result of an unknown HMM state sequence (where HMM states are referring to the states of the hidden Markov chain). Each HMM state has an associated emission probability function, which describes the expected observations of that state. The state sequence cannot be analytically solved, but can be estimated using some optimality criterion. The Viterbi algorithm is one such method that finds the best state sequence, Q, for a given model and observation sequence by maximizing $P(Q|O, \lambda)$. The following section details the Viterbi algorithm and shows example state sequences generated by each of the models.

3.5.1 Viterbi Algorithm

The Viterbi Algorithm [33] solves for the state sequence that maximizes the probability of the state sequence given the observation sequence and model, $P(Q|O, \lambda)$. The algorithm solves for δ, which is defined as

$$\delta_t(i) = \max_{q_1,q_2,\ldots q_{t-1}} P\left[q_1, q_2, ...q_t = i, O_1, O_2, ...O_t|\lambda\right]. \tag{3.17}$$

$\delta(i)$ the path with the highest probability at time t that ends on state S_i. As seen in Equation (3.18), $\delta_1(t)$ is initialized with the same method as α in the forward algorithm – using probability of the initial observation in state S_i ($B_i(O)$) and the initial probability of state S_i occurring (π_i). $\psi_1(i)$, a variable used to keep track of the best state sequence, is initialized as 0 for each state. The initialization process can be stated explicitly as,

$$\delta_1(i) = \pi_i B_i(O_1), \qquad 1 \leq i \leq Q \tag{3.18a}$$

$$\psi_1(i) = 0. \tag{3.18b}$$

Next, there is a recursion step, which is defined as,

$$\delta_t(j) = \max_{1 \leq i \leq Q} \left[\delta_{t-1} a_{ij}\right] B_j(O_t), \qquad 2 \leq t \leq T, \quad 1 \leq j \leq Q \tag{3.19a}$$

$$\psi_t(j) = \operatorname*{arg\,max}_{1 \leq i \leq Q} \left[\delta_{t-1}(i) a_{ij}\right], \qquad 2 \leq t \leq T, \quad 1 \leq j \leq Q. \tag{3.19b}$$

The transition probability, the emission probability of the observation at that time step, and the path that previously produced the highest probability, δ_{t-1}, are used to update $\delta_t(i)$. Meanwhile, $\psi_t(i)$ keeps track of which state produced the highest probability path (δ_{t-1}).

Finally, the termination step occurs when the observation sequence is terminated. This step is defined as,

$$P^* = \max_{1 \leq i \leq Q} \delta_T(i) \tag{3.20a}$$

$$q_T^* = \operatorname*{arg\,max}_{1 \leq i \leq Q} \delta_T(i). \tag{3.20b}$$

The final maximum $\delta_T(i)$ and the state associated with that δ_T is found. The state sequence that is optimal for the entire series of observations is found through backtracking,

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, ..., 1. \tag{3.21}$$

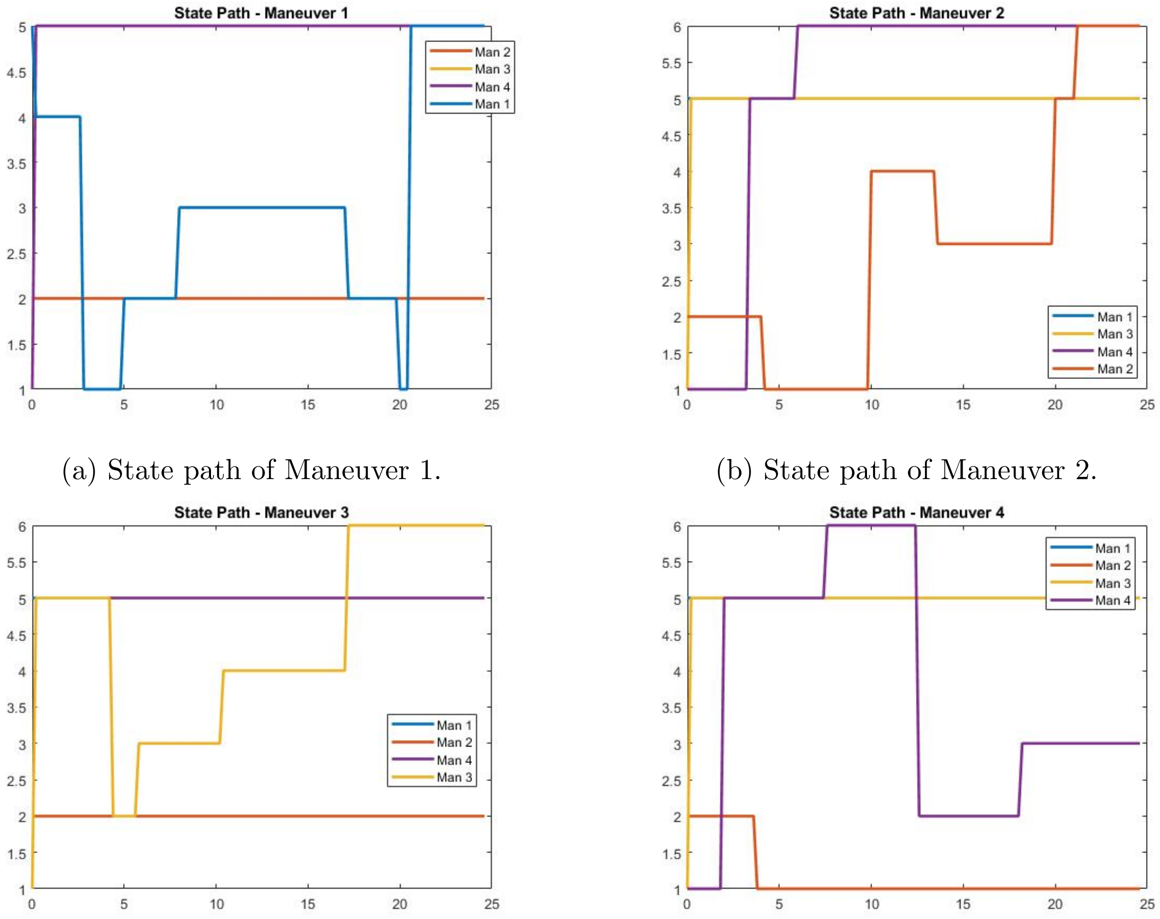
The state with the highest $\delta_T(i)$ initializes the backtrack, and then ψ_{t+1}, which stores information about the states that maximized $\delta_t(i)$ throughout the sequence, determines final state sequence.

Like the forward algorithm, the Viterbi algorithm can be implemented on a partial state sequence ($O_{1:t}$ rather than $O_{1:T}$). At any point in time, the termination and backtracking step can occur. However, since the algorithm sequence produced is initialized with the maximum $\delta_t(i)$ at the current time t and then backtracks, the entire state sequence may change depending on when the process is terminated. The state path found using only a partial observation sequence may not be the optimal path for the entirety of the observations.

3.5.2 Example State Paths

Test Data

Figure 3.25 shows the state path for each of the four HMMs when using true UAV states from the test data set and the Viterbi algorithm. These paths are the result of the algorithm being terminated once the final observation is reached. Figure 3.25a shows the state path of each model when receiving observations from Maneuver 1, while Figure 3.25b shows the state path found when given observations from Maneuver 2, and so on. In each figure, the state path of the model corresponding to the correct maneuver is the path that is most dynamic, while the paths of the incorrect model are often stagnant. Each correct HMM visits every state in its model (except Maneuver 4).



(a) State path of Maneuver 1.

(b) State path of Maneuver 2.

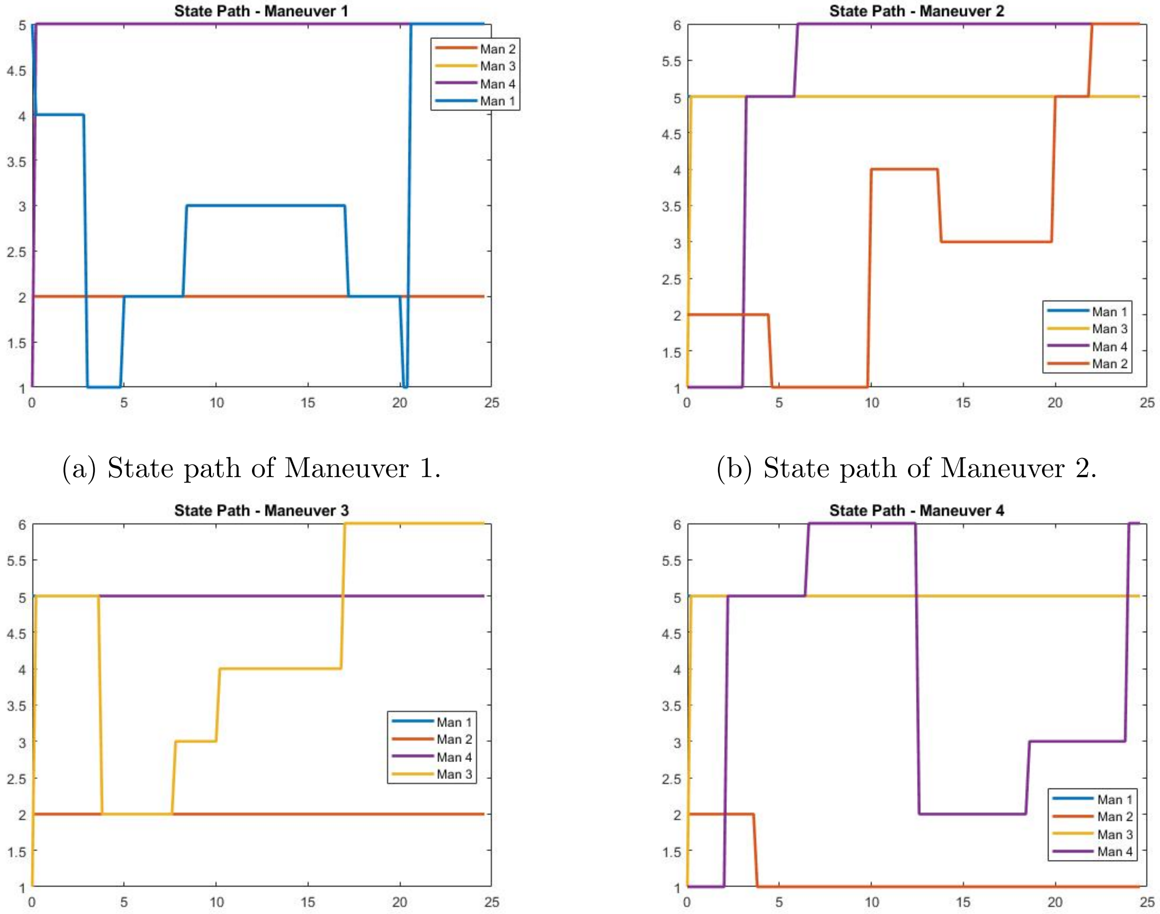
(c) State path of Maneuver 3.

(d) State path of Maneuver 4.

Figure 3.25: State path of each maneuver found using the Viterbi algorithm and a test data run of true UAV states.

EKF Test Data

In Figure 3.26, the model state sequence is shown for EKF estimates of the same maneuvers used in Figure 3.25. The state path does not match one-for-one with the state path found using true UAV states (Maneuvers 2 and 3 for example), as the duration spent in each state differs and transitions occur at different time steps. However, the same overall sequence of states occurs whether using true UAV states or estimated UAV states.



(a) State path of Maneuver 1.

(b) State path of Maneuver 2.

(c) State path of Maneuver 3.

(d) State path of Maneuver 4.

Figure 3.26: State path of each maneuver found using the Viterbi algorithm and a test data run of estimated UAV states.

3.6 HMM Estimates

A HMM models a system so that at each moment in time, the HMM is in one of Q latent states and, as a result, produces an observation. For a given observations sequence, the Viterbi algorithm finds the most likely HMM state sequence that would have produced those observations. In other words, each observation is now assigned to have originated from an HMM state. This state also has an emission probability function ($B_i(O)$) that describes what observations are most probable to be produced by that state. In a scenario where some elements of an observation (O_t) may be very certain while others are less certain, the question then becomes: can the emission probability be used to extract extra information about the observation at time t given the expected behavior of the HMM state?

In this thesis, the UAV position, velocity, and acceleration are the observations of the HMM. When UAV states are estimated during a dynamic maneuver, there may be instances in which some of the state estimates are more accurate than others. For example, Section 2.4 shows that position estimates typically have higher accuracy than velocity or acceleration estimates. Once the HMM state sequence corresponding to the observation sequence is found, this thesis aims to use the emission probability corresponding to each observation and the position estimates in these observations to extract information from the HMM.

The emission probability function used in each maneuver's HMM is a Gaussian Mixture Model. A Gaussian Mixture Regression (GMR), a nonlinear regression technique that is further described in the next section, is used to create HMM estimates of acceleration and jerk. The position estimate of a UAV is processed and used as an input to the GMR. Here, "processed" is referring to the data that has been differenced and normalized (as seen in Equation (3.7)). The GMR then produces expected processed velocity and acceleration values.

It is important to note that HMM estimates are limited by the ability of the emission probability to accurately represent observations and the accuracy of the state sequence when determined by the Viterbi algorithm. In other words, the quality of the HMM training determines the level of accuracy of the emission probability. Likewise, because the HMM estimates are updated in real time rather than when the sequence is terminated, the HMM state path produced by the Viterbi algorithm may not be the optimal path for the full observation sequence, which could negatively affect the accuracy of HMM estimates.

3.6.1 Gaussian Mixture Regression

A GMR is a nonlinear regression technique that originates from the definition of a joint density Gaussian distribution, $f_{X,Y}$. A joint density function is also described as,

$$f_{X,Y} = f_{Y|X} f_X. \tag{3.22}$$

A property of joint Gaussian distributions states that $f_{Y|X}$ and f_X are also Gaussian [54]. Moreover, the regression for $X|Y$ is linear and based on components of the original joint density function. Sung (2004) showed that this property can be applied to a multivariate Gaussian Model or a multivariate Gaussian Mixture Model and developed the following process for a Gaussian Mixture Regression [55].

To perform a Gaussian Mixture Regression, μ and Σ are divided into the known values or inputs, k, and the unknown values or outputs, u:

$$\mu_m = [\mu_{k,m} \mu_{u,m}] \tag{3.23a}$$

$$\Sigma_m = \begin{bmatrix} \Sigma_{k,m} & \Sigma_{ku,m} \\ \Sigma_{uk,m} & \Sigma_{u,m} \end{bmatrix}. \tag{3.23b}$$

Then, the conditional expected value, $\hat{\xi}_{u,m}$, and estimated conditional covaraince matrix, $\hat{\Sigma}_{u,m}$, are found using the following equations,

$$\hat{\xi}_{u,m} = \mu_{u,m} + \Sigma_{uk,m}(\Sigma_{k,m})^{-1}(\xi_k - \mu_{k,m}) \tag{3.24a}$$

$$\hat{\Sigma}_{u,m} = \Sigma_{u,m} - \Sigma_{uk,m}(\Sigma_{k,m})^{-1}\Sigma_{ku,m}. \tag{3.24b}$$

Each mixture of the GMM is given a weight based on the probability of the known values,

$$\beta_m = \frac{p(\xi_k|m)}{\sum_{i=1}^{M} p(\xi_k|i)}. \tag{3.25}$$

The final conditional expected value, $\hat{\xi}_u$, and conditional covariance, $\hat{\Sigma}_u$, of the unknown values are calculated. The process of which is shown as

$$\hat{\xi}_u = \sum_{m=1}^{M} \beta_m \hat{\xi}_{u,m} \tag{3.26a}$$

$$\hat{\Sigma}_u = \sum_{m=1}^{M} \beta_m^2 \hat{\Sigma}_{u,m}. \tag{3.26b}$$

3.6.2 Estimation Generation

For this thesis, the inputs to the regression are the (processed) position states of the UAV, while the outputs are the conditional expected (processed) velocity and acceleration. The emission probability function of each HMM state is a multivariate Gaussian rather than a GMM. Therefore, the GMR process can be simplified. First, the mean vector and covariance matrix are divided into known and unknown values,

$$\mu = [\mu_{\mathbf{r}_n} \mu_{\mathbf{v}_n \mathbf{a}_n}], \quad \Sigma = \begin{bmatrix} \Sigma_{\mathbf{r}_n} & \Sigma_{\mathbf{r}_n \mathbf{v}_n \mathbf{a}_n} \\ \Sigma_{\mathbf{v}_n \mathbf{a}_n \mathbf{r}_n} & \Sigma_{\mathbf{v}_n \mathbf{a}_n} \end{bmatrix}. \tag{3.27}$$

The expected value of processed velocity and acceleration given the current position is found using

$$\begin{bmatrix} \hat{\mathbf{v}}_{n,HMM} \\ \hat{\mathbf{a}}_{n,HMM} \end{bmatrix} = \mu_{\mathbf{v}_n \mathbf{a}_n} + \Sigma_{\mathbf{v}_n \mathbf{a}_n \mathbf{r}_n} (\Sigma_{\mathbf{r}_n})^{-1} (\mathbf{r}_n - \mu_{\mathbf{r}_n}). \tag{3.28}$$

These values are used to produce HMM acceleration and jerk estimates through the equations

$$\hat{\mathbf{a}}_{HMM} = \frac{(\hat{\mathbf{v}}_{n,HMM} + \mathbf{v}_{min})(\mathbf{v}_{max} - \mathbf{v}_{min})}{\delta t}, \tag{3.29}$$

and

$$\hat{\mathbf{j}}_{HMM} = \frac{(\hat{\mathbf{a}}_{n,HMM} + \mathbf{a}_{min})(\mathbf{a}_{max} - \mathbf{a}_{min})}{\delta t}. \tag{3.30}$$

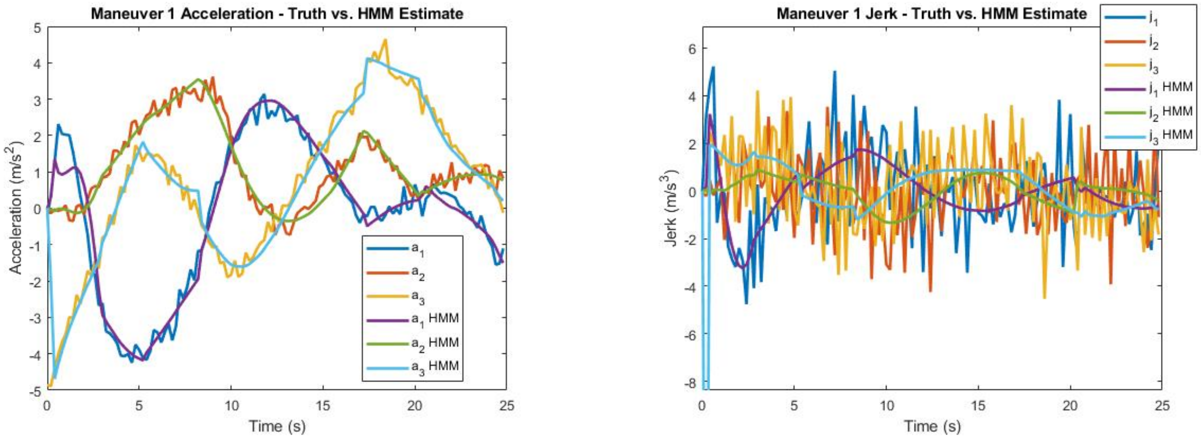
Therefore, the actual HMM estimates produced are acceleration and jerk ($\hat{\mathbf{a}}_{HMM}$ and $\hat{\mathbf{j}}_{HMM}$) rather than velocity and acceleration.

3.6.3 HMM Estimate Results

Figures 3.27, 3.28, 3.29, and 3.30 show true acceleration and jerk versus the HMM estimate for Maneuvers 1, 2, 3, and 4 respectively. These estimates were generated using a sample truth trajectory. Although both true acceleration and true jerk are noisy, the HMM estimates filter the noise. The noise on the true acceleration and jerk is the result of random process noise and differs during each simulation of a maneuver. The HMM estimates shows the ability to recognize trends in data without incorporating unnecessary noise.

Because the GMR is nonlinear, the HMM estimates are capable of following the acceleration and jerk even when there are nonlinearities or non-differentiable behaviors. For example in Figure 3.30, the HMM estimate of a_3 is capable of following the sharp peak at three seconds.

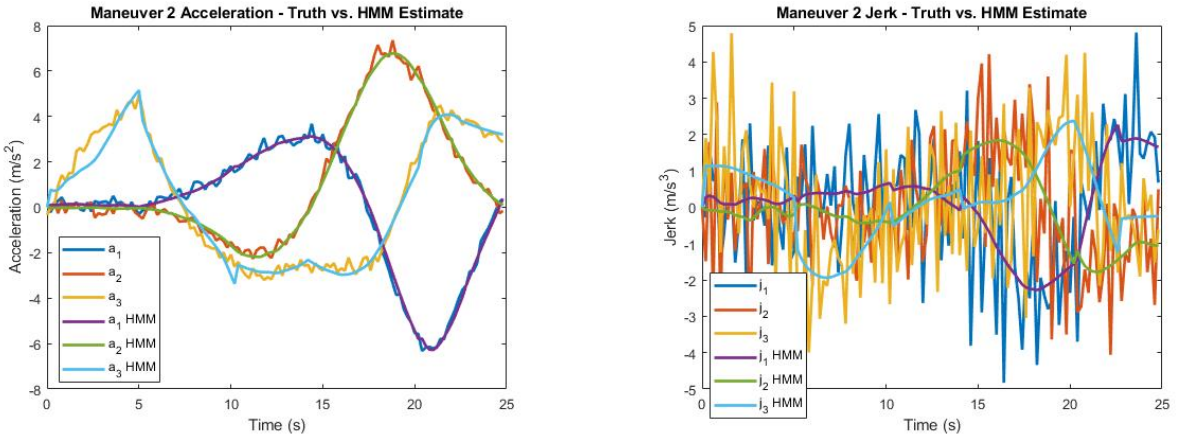
However, there are some clear errors and biases within the HMM estimates. The HMM estimate of a_3 in Maneuver 2 is clearly erroneous for the first five seconds of the trajectory. Similarly, the HMM estimate of a_1 for Maneuver 1 has difficulty following the initial peak seen with the true acceleration. These biases indicate less than optimal emission probability parameters developed during HMM training, and are the fault of the fundamental parameters of the HMM rather than the GMR process.



(a) Maneuver 1 acceleration: truth versus HMM estimates.

(b) Maneuver 1 jerk: truth versus HMM estimates.

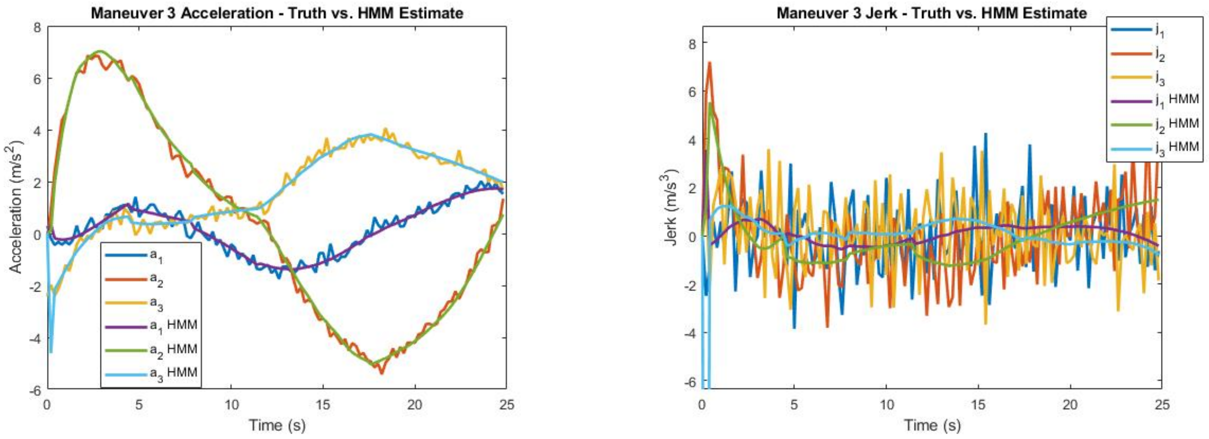
Figure 3.27: Example of HMM acceleration and jerk estimates of Maneuver 1.



(a) Maneuver 2 acceleration: truth versus HMM estimates.

(b) Maneuver 2 jerk: truth versus HMM estimates.

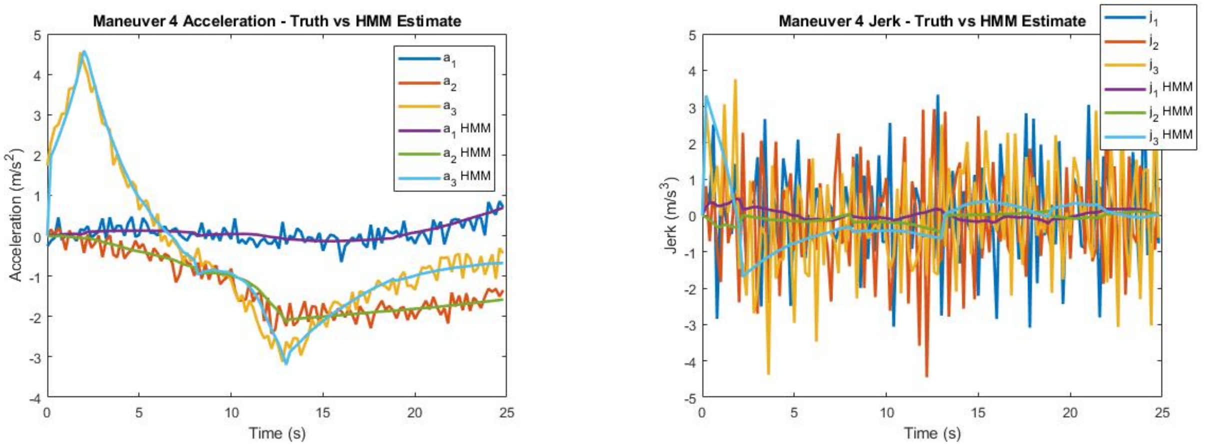
Figure 3.28: Example of HMM acceleration and jerk estimates of Maneuver 2.



(a) Maneuver 3 acceleration: truth versus HMM estimates.

(b) Maneuver 3 jerk: truth versus HMM estimates.

Figure 3.29: Example of HMM acceleration and jerk estimates of Maneuver 3.



(a) Maneuver 4 acceleration: truth versus HMM estimates.

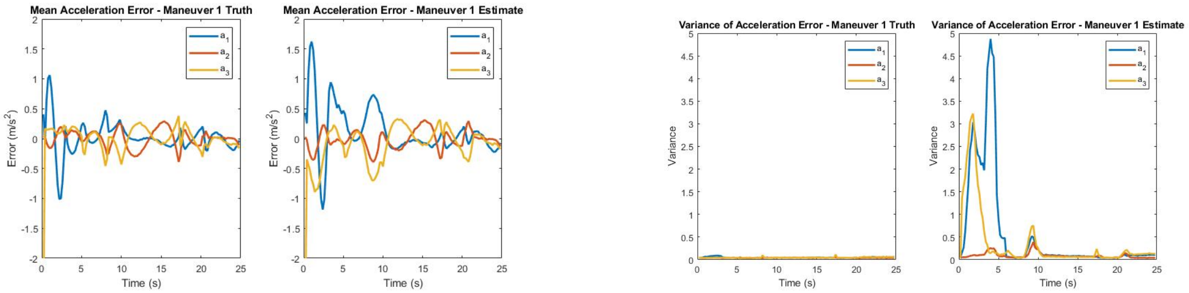
(b) Maneuver 4 jerk: truth versus HMM estimates.

Figure 3.30: Example of HMM acceleration and jerk estimates of Maneuver 4.

3.6.4 Characteristics of HMM Estimates

Maneuver 1

Figure 3.31a shows the HMM acceleration estimate error averaged over each trajectory from the test data set. The HMM estimates were generated from true position (left) and EKF estimated position (right). Figure 3.31b shows the error variance for both of these scenarios. HMM estimates generated from EKF position have generally the same structure of errors from HMM estimates generated from truth position. However, they have higher errors – particularly in the beginning of the maneuver. This is reflected in the variance of errors – where HMM estimates generated from EKF position have a much higher error variance. Both sets of errors, however, have clear biases and trends shown in the mean errors.

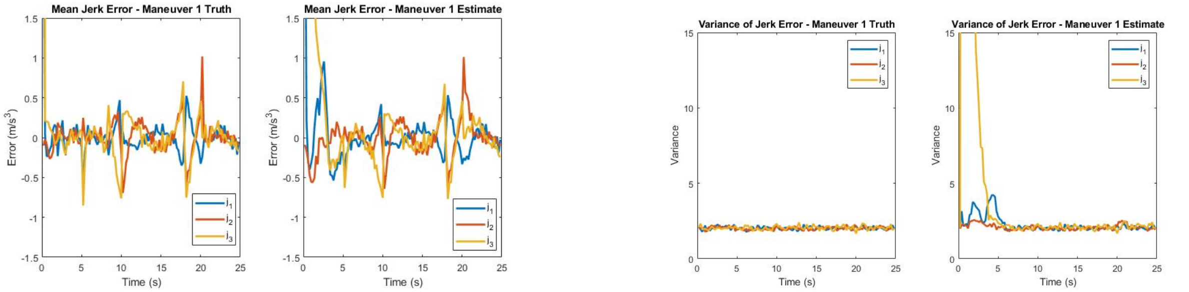


(a) Mean acceleration error over time of HMM estimates using true position (left) and estimated position (right).

(b) Acceleration error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.31: Characteristics of HMM acceleration estimates generated from λ_1.

The mean error of HMM jerk estimates generated from true position and estimated position are shown in Figure 3.32a, while the corresponding variances of these errors are shown in Figure 3.32b. The error of jerk estimates generated from truth do not have as much error as the HMM jerk estimate generated from EKF position. The areas of higher error correspond to areas of higher error for HMM acceleration estimates.



(a) Mean jerk error over time of HMM estimates using true position (left) and estimated position (right).

(b) Jerk error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.32: Characteristics of HMM jerk estimates generated from λ_1.

The variance in jerk errors is nearly constant over time for HMM jerk estimates generated from true position. Moreover, it is reflective of the variance seen in jerk noise. The error variance of jerk estimates generated from estimated position have large variance at specific times corresponding to high error in the HMM estimates, but other than that remain relatively constant like their counterparts.

The state sequence for each of the test truth data maneuvers and estimated test data maneuvers generated using the Viterbi Algorithm are shown in Figure 3.33. Note that the figure shows the state path for roughly 700 simulations of Maneuver 1. The change from state to state does not chatter. These are individual runs.

While the path sequence of the estimates follows the same path as the path sequence of true data, there is much more variation in the time period in which state transitions occur. Note the state transitions that have a wide scale of time in which they happen (zero to three seconds and three to seven seconds) correspond to areas in which jerk and acceleration estimates generated from position estimates have much higher error than jerk and acceleration estimates generated from true position. This suggests that some error incurred is not from a lack of optimal parameters for the emission probability functions but rather errors in the state sequence.

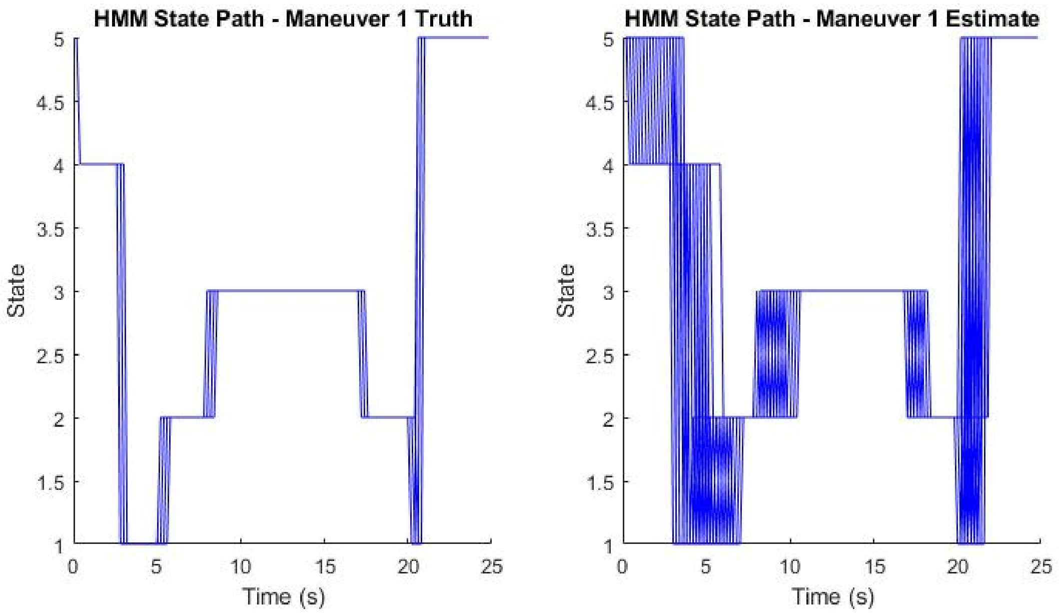
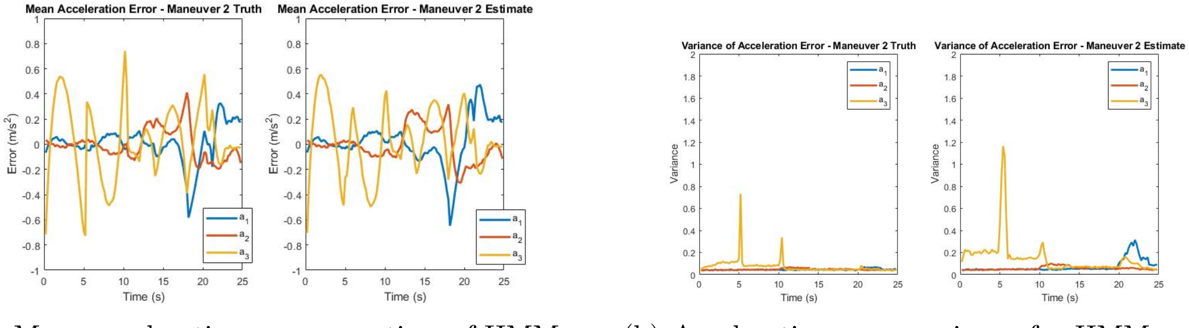


Figure 3.33: HMM states as estimated from Viterbi Algorithm using truth (left) and estimates (right)

Maneuver 2

The mean acceleration error over time for HMM estimates generated from true position and estimated position are shown in Figure 3.34a. Both cases have high error for a_3 in the first ten seconds of the maneuver, which is reflective of the incorrect HMM estimate shown in Figure 3.28a. The HMM estimates generated from estimated position also produce slightly higher errors towards the end of the maneuver.

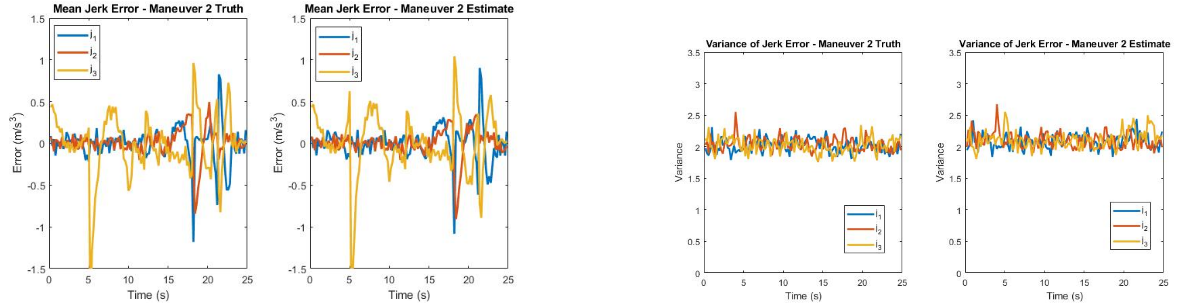


(a) Mean acceleration error over time of HMM estimates using true position (left) and estimated position (right).

(b) Acceleration error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.34: Characteristics of HMM acceleration estimates generated from λ_2.

The jerk HMM estimates generated from true position and estimated position, as well as their error variances are shown in Figure 3.35. j_3 has a high error in the initial and final stages of the maneuver, much like a_3. However the variance of error remains low during this period suggesting that the HMM estimate is consistent in its error across the data set.



(a) Mean jerk error over time of HMM estimates using true position (left) and estimated position (right).

(b) Jerk error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.35: Characteristics of HMM jerk estimates generated from λ_2.

Figure 3.36 shows the HMM state path found for each of the maneuvers using the truth test set (left) and estimated test set (right). Note that for the first five seconds of the maneuver, the state consistently stays at two for both data sets, suggesting that the error in a_3 and j_3 is an issue with having non-optimal parameters in state two's emission probability rather than an issue in finding the correct state sequence.

Another interesting note is there is one instance of "chatter" in which the Viterbi algorithm switches between two states in quick succession. This can be seen when using UAV state estimates at around 6 seconds. This shows in some cases the noise and error from the EKF estimates can cause the HMM to diverge (slightly) from the most likely state path found using true UAV states.

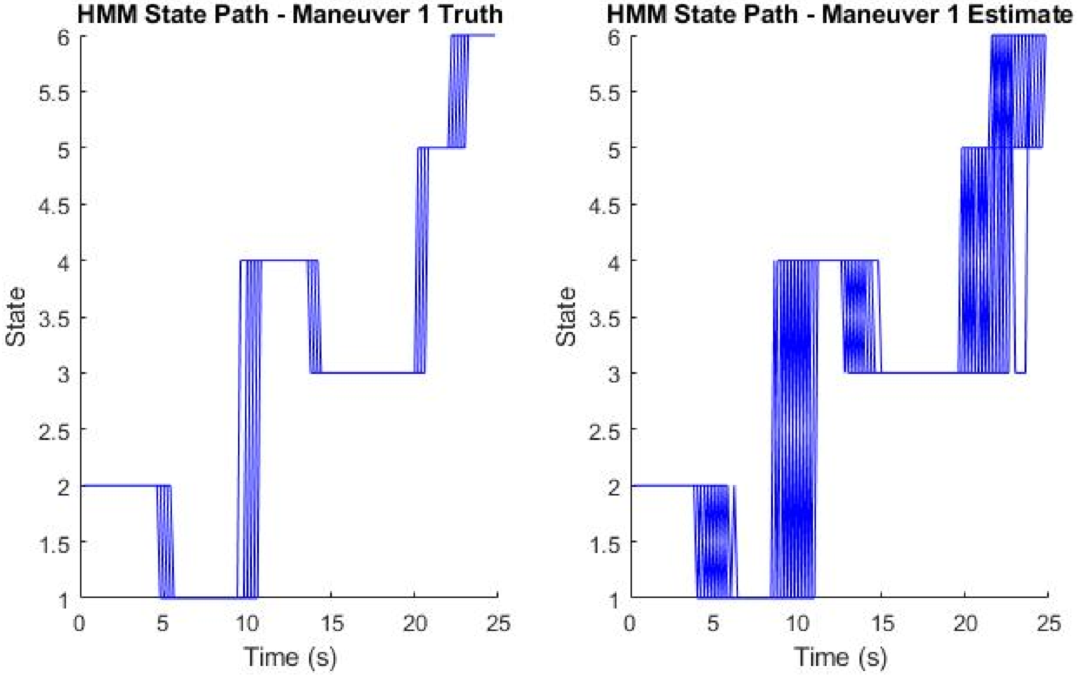
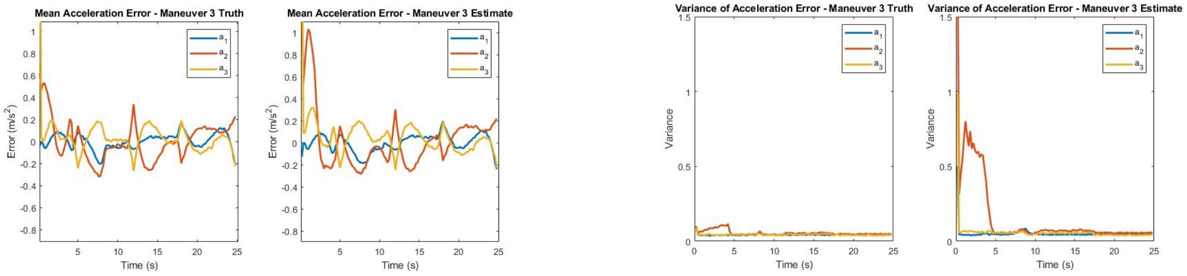


Figure 3.36: HMM states as estimated from Viterbi Algorithm using truth (left) and estimates (right)

Maneuver 3

The mean error over time and variance of mean error over time for HMM acceleration estimates generated using true position and estimated position are shown in Figure 3.37. The error in Maneuver 3 is consistent across both data sets except in the initial few seconds of flight. The variance in error for acceleration remains low, except for a_2. When there is high mean error for a_2, the error variance of a_2 is also high.

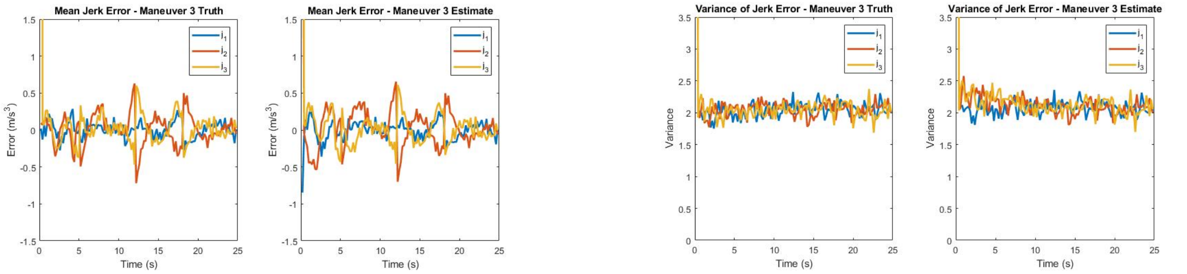


(a) Mean acceleration error over time of HMM estimates using true position (left) and estimated position (right).

(b) Acceleration error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.37: Characteristics of HMM acceleration estimates generated from λ_3.

Figure 3.38 shows the mean error over time for HMM jerk estimates generated from true position and estimated position. The variance of these errors are also shown. Error spikes are found in time periods corresponding to the HMM acceleration estimate errors. Overall, however, the errors remain low and the error variance consistent.



(a) Mean jerk error over time of HMM estimates using true position (left) and estimated position (right).

(b) Jerk error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.38: Characteristics of HMM jerk estimates generated from λ_3.

The states path found using truth as observations and estimates as observations are shown in Figure 3.39. Like previous models, the state path is mostly consistent in sequence when using either truth or estimates as observations. However, the time period over which transitions occur is inconsistent when using estimates. There are some slight "chatter" effects when using EKF estimates as opposed to truth data. The errors shown for Maneuver 3 do not correspond heavily with any state transitions, suggesting that most error is incurred from less than optimal emission probability parameters.

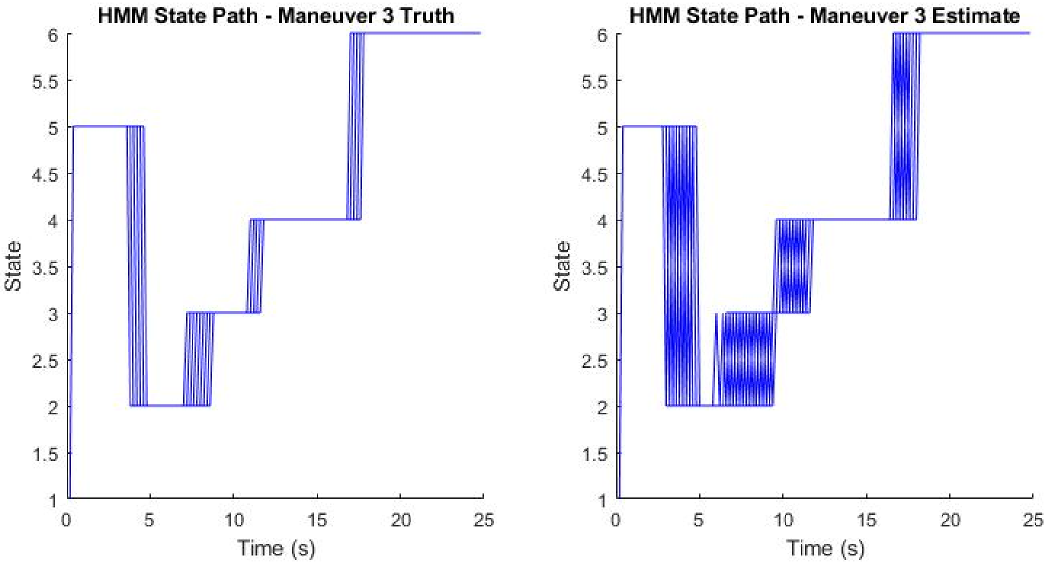
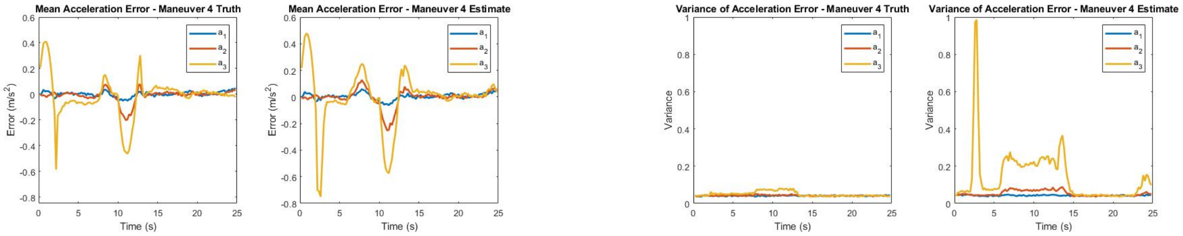


Figure 3.39: HMM states as estimated from Viterbi Algorithm using truth (left) and estimates (right)

Maneuver 4

The mean acceleration error and error variance for HMM estimates generated using either true position or estimated position are shown in Figure 3.40. There is consistent error in acceleration estimates, for both true and estimated position, during the same portions of the maneuver. This is especially severe with a_3. The error variance is much more variable with HMM estimates generated from EKF position.

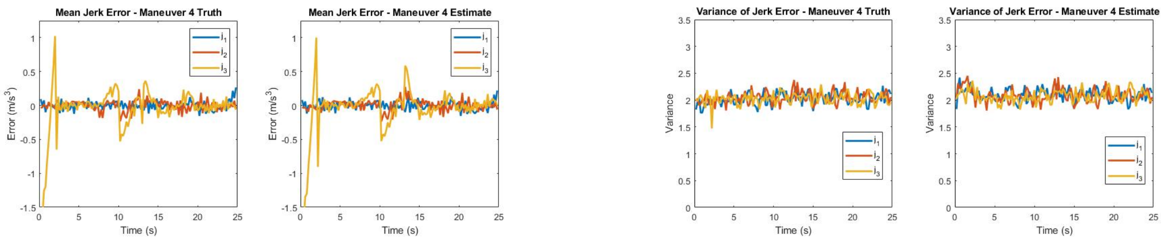


(a) Mean acceleration error over time of HMM estimates using true position (left) and estimated position (right).

(b) Acceleration error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.40: Characteristics of HMM acceleration estimates generated from λ_4.

The HMM jerk estimates have generally low error and consistent error variance, as shown in Figures 3.41a and 3.41b respectively. The spikes in error, particularly for j_3, relate to the error found in HMM acceleration estimates.



(a) Mean acceleration error over time of HMM estimates using true position (left) and estimated position (right).

(b) Acceleration error variance for HMM estimate when using true position (left) and estimated position (right).

Figure 3.41: Characteristics of HMM jerk estimates generated from λ_4.

The state sequence found using truth data and estimated data is shown in Figure 3.42. The sequence remains the same for both truth and estimated data until the final three seconds of the maneuver, in which the state sequence from truth remains in state three, while the state sequence from error transitions back to state six. This relates to a small increase in error and error variance seen in HMM acceleration estimates generated from position estimates. The sequence generated from estimated data also has higher variance in the times that the states transition, which may correspond to higher error in the HMM acceleration estimates generated from position estimates during the maneuver.

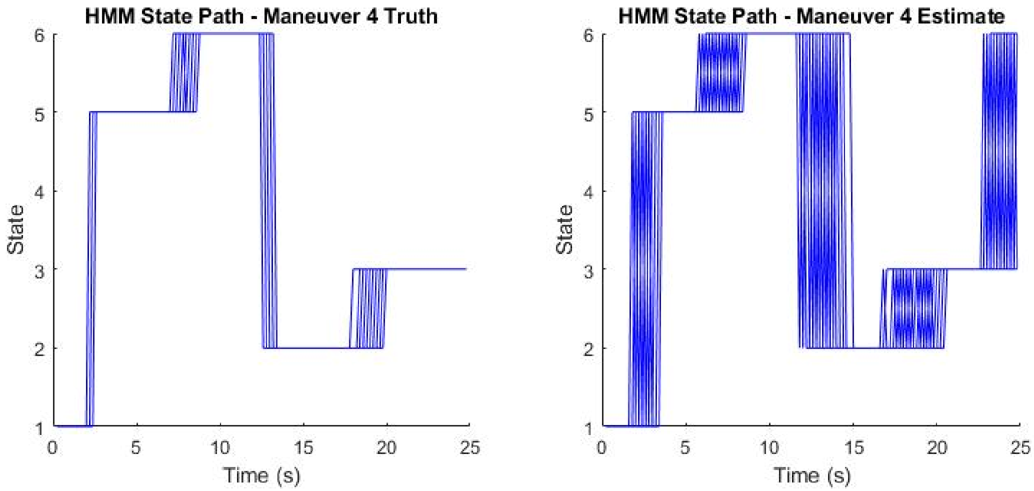


Figure 3.42: HMM states as estimated from Viterbi Algorithm using truth (left) and estimates (right)

3.7 Conclusion

This chapter summarized basic knowledge of HMMs, and explained modelling individual UAV maneuvers as HMMs. The HMMs were then used to classify maneuvers and provide estimates of acceleration and jerk. The following chapter will discuss how HMM classification and estimates are incorporated into the estimation of UAV states during dynamic maneuvers.

Chapter 4

EKF+HMM Estimation of UAV States

Previously, UAV maneuvers were estimated using an EKF and modelled with an HMM. Now, the two are combined into a novel EKF+HMM estimator. The EKF+HMM uses the benefits of HMMs (their ability to classify maneuvers and to generate estimates) to supply additional information to an EKF and improve UAV state estimates. The following sections will establish the EKF+HMM formulation and then show results of the EKF+HMM when compared to a standard EKF.

4.1 EKF + HMM Formulation

The EKF + HMM, shown below in Figure 4.1, aims to take advantage of a maneuver's HMM to aid in state estimation. A standard EKF runs throughout the entirety of the UAV flight while each HMM is updated. When there is no active maneuver classification, the EKF+HMM uses the standard EKF estimates. This is shown in Figure 4.1, where if the classification is unknown, the output of the filter is the EKF estimate. If the confidence of any HMM crosses the likelihood threshold, maneuver classification takes place. Classifications with high confidence (less than -1) result in the EKF+HMM incorporating HMM information into its estimates and instead outputting the EKF+HMM estimate. The following sections further break down the architecture and concepts of the EKF+HMM.

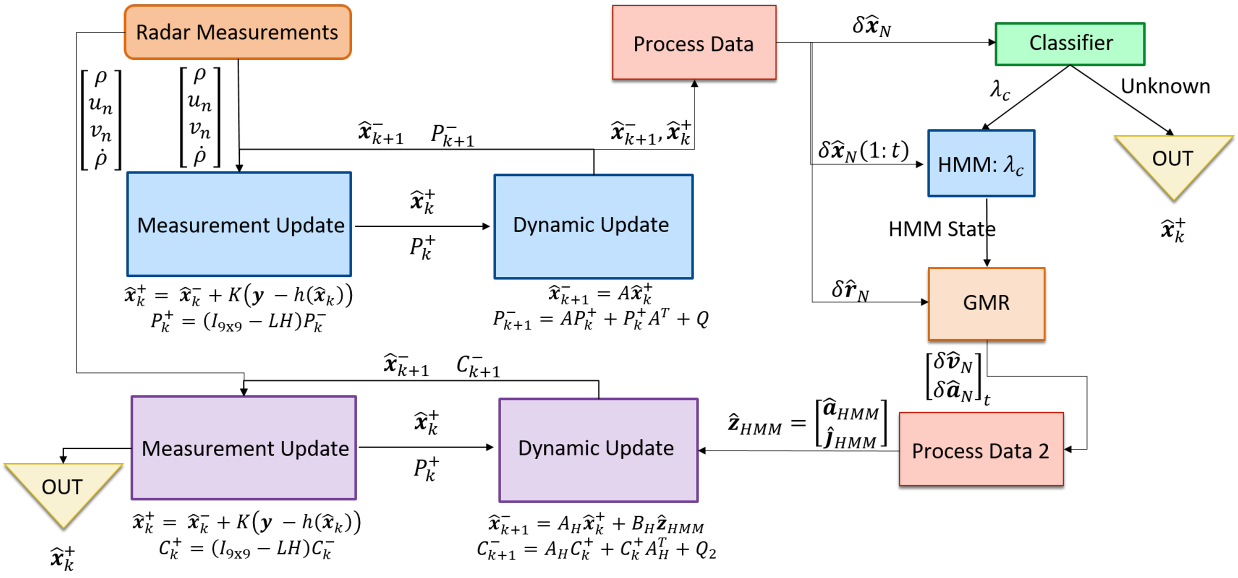

Radar Measurements
$\begin{bmatrix}\rho\\ u_n\\ v_n\\ \dot{\rho}\end{bmatrix}$
$\begin{bmatrix}\rho\\ u_n\\ v_n\\ \dot{\rho}\end{bmatrix}$
Process Data
$\delta\hat{\boldsymbol{x}}_N$
Classifier
λ_c
Unknown
$\hat{\boldsymbol{x}}_{k+1}^-$ P_{k+1}^-
$\hat{\boldsymbol{x}}_{k+1}^-, \hat{\boldsymbol{x}}_k^+$
$\delta\hat{\boldsymbol{x}}_N(1:t)$
HMM: λ_c
OUT
Measurement Update
$\hat{\boldsymbol{x}}_k^+$
P_k^+
Dynamic Update
HMM State
$\hat{\boldsymbol{x}}_k^+$
$\delta\hat{\boldsymbol{r}}_N$
GMR
$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K(\boldsymbol{y} - h(\hat{\boldsymbol{x}}_k))$
$P_k^+ = (I_{9x9} - LH)P_k^-$
$\hat{\boldsymbol{x}}_{k+1}^- = A\hat{\boldsymbol{x}}_k^+$
$P_{k+1}^- = AP_k^+ + P_k^+A^T + Q$
$\hat{\boldsymbol{x}}_{k+1}^-$ C_{k+1}^-
$\begin{bmatrix}\delta\hat{\boldsymbol{v}}_N\\ \delta\hat{\boldsymbol{a}}_N\end{bmatrix}_t$
Measurement Update
$\hat{\boldsymbol{x}}_k^+$
P_k^+
Dynamic Update
$\hat{\boldsymbol{z}}_{HMM} = \begin{bmatrix}\hat{\boldsymbol{a}}_{HMM}\\ \hat{\boldsymbol{j}}_{HMM}\end{bmatrix}$
Process Data 2
OUT
$\hat{\boldsymbol{x}}_k^+$
$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K(\boldsymbol{y} - h(\hat{\boldsymbol{x}}_k))$
$C_k^+ = (I_{9x9} - LH)C_k^-$
$\hat{\boldsymbol{x}}_{k+1}^- = A_H\hat{\boldsymbol{x}}_k^+ + B_H\hat{\boldsymbol{z}}_{HMM}$
$C_{k+1}^- = A_HC_k^+ + C_k^+A_H^T + Q_2$

Figure 4.1: The complete EKF+HMM scheme

4.1.1 Basic EKF

The EKF described in Section 2.3 remains a crucial part of the total EKF+HMM structure. This piece, displayed in Figure 4.2, estimates UAV states. These estimates are then used for HMM classification and HMM estimate generation. After each dynamic update, $\hat{\mathbf{x}}_{k+1}^-$ and $\hat{\mathbf{x}}_k^+$ are passed to the pre-processing block, which differences the data and normalizes it with values established from HMM training,

$$\mathbf{x}_n = \frac{[\mathbf{x}_{k+1}^- - \mathbf{x}_k^+] - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}. \tag{4.1}$$

These processed estimates are utilized as observation inputs for each HMM.

Even after a clear classification, the EKF estimates are still used as inputs to the classifier and HMM estimate generator. This prevents the EKF+HMM from further compounding error if the classification is incorrect or if the HMM estimates generated are invalid.

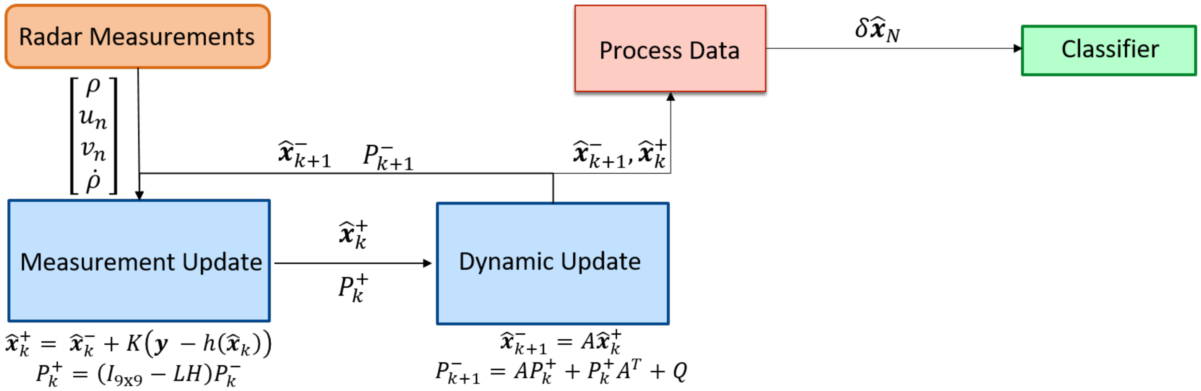

Radar Measurements
$\begin{bmatrix} \rho \\ u_n \\ v_n \\ \dot{\rho} \end{bmatrix}$
Process Data
$\delta\hat{\boldsymbol{x}}_N$
Classifier
$\hat{\boldsymbol{x}}_{k+1}^-$ P_{k+1}^-
$\hat{\boldsymbol{x}}_{k+1}^-, \hat{\boldsymbol{x}}_k^+$
Measurement Update
$\hat{\boldsymbol{x}}_k^+$
P_k^+
Dynamic Update
$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K(\boldsymbol{y} - h(\hat{\boldsymbol{x}}_k))$
$P_k^+ = (I_{9\text{x}9} - LH)P_k^-$
$\hat{\boldsymbol{x}}_{k+1}^- = A\hat{\boldsymbol{x}}_k^+$
$P_{k+1}^- = AP_k^+ + P_k^+A^T + Q$

Figure 4.2: EKF+HMM Scheme

Chapter 3 includes analysis of the ability of the classifier when using estimates of states rather than true states, as well as HMM estimates generated from estimated position. Overall, the classification accuracy remained high and the HMM estimates generated had only slightly more error than those generated from truth, showing that this is a viable method to use within the EKF+HMM scheme.

4.1.2 HMM Classification and Estimation Generation

The processed estimates from the EKF are then passed to the HMM portion of the EKF+HMM system, which can be viewed in Figure 4.3. The processed estimates are classified as either a known maneuver or unknown maneuver. If there is no known maneuver corresponding to the estimates, the EKF+HMM output will be the EKF output.

If the classifier identifies a known maneuver with high confidence, the HMM for that maneuver, λ_c, will be selected to generate estimates. The processed EKF estimates are used to update the Viterbi algorithm for λ_c, which finds the current HMM state. A Gaussian Mixture Regression is performed using the emission probability, B_i, for that state and the processed position estimates from the EKF. The output of the GMR is passed through the second data processing block, where it produces λ_c's estimates of acceleration and jerk, $\hat{\mathbf{a}}_{HMM}$ and $\hat{\mathbf{j}}_{HMM}$. These estimates become inputs to the EKF+HMM.

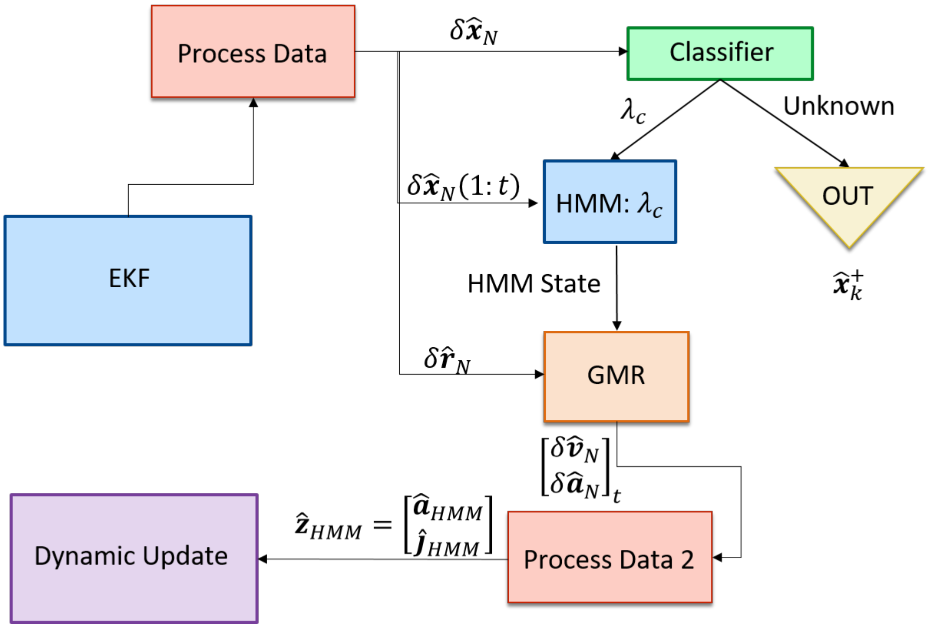

Process Data
$\delta\hat{\boldsymbol{x}}_N$
Classifier
λ_c
Unknown
$\delta\hat{\boldsymbol{x}}_N(1:t)$
HMM: λ_c
OUT
$\hat{\boldsymbol{x}}_k^+$
EKF
HMM State
$\delta\hat{\boldsymbol{r}}_N$
GMR
$\begin{bmatrix}\delta\hat{\boldsymbol{v}}_N \\ \delta\hat{\boldsymbol{a}}_N\end{bmatrix}_t$
$\hat{\boldsymbol{z}}_{HMM} = \begin{bmatrix}\hat{\boldsymbol{a}}_{HMM} \\ \hat{\boldsymbol{j}}_{HMM}\end{bmatrix}$
Dynamic Update
Process Data 2

Figure 4.3: A block diagram of how the HMMs interact with the other components of the EKF+HMM

During initial implementation of the EKF+HMM, the HMM estimates would occasionally produce high error estimates that were outliers. These outliers are most often the result of a misclassification. When a HMM is presented with data that is unusual for that model and its training data set (as would happen in the case of misclassification), abnormal acceleration and jerk estimates occur. The outliers were filtered out of the HMM estimates by ensuring that $\delta\hat{\mathbf{a}}_{HMM}$ did not exceed 15 from one time step to the next, as this is an acceleration change outside of what maneuvers were capable of performing.

4.1.3 EKF+HMM Structure

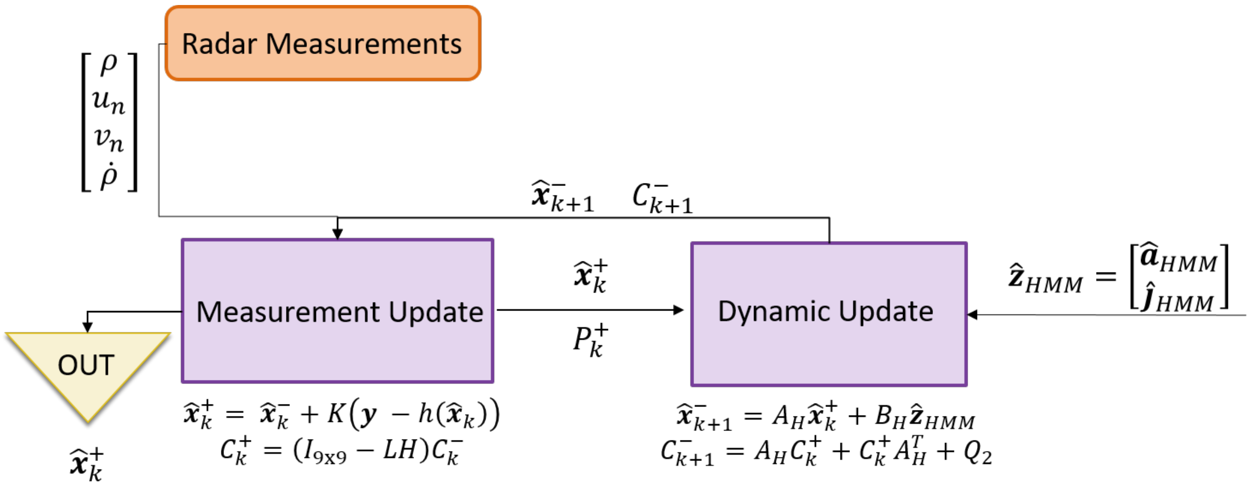

Radar Measurements
$\begin{bmatrix} \rho \\ u_n \\ v_n \\ \dot{\rho} \end{bmatrix}$
$\hat{\boldsymbol{x}}_{k+1}^{-}$ C_{k+1}^{-}
Measurement Update
$\hat{\boldsymbol{x}}_k^{+}$
P_k^{+}
Dynamic Update
$\hat{\mathbf{z}}_{HMM} = \begin{bmatrix} \hat{\boldsymbol{a}}_{HMM} \\ \hat{\boldsymbol{j}}_{HMM} \end{bmatrix}$
OUT
$\hat{\boldsymbol{x}}_k^{+}$
$\hat{\boldsymbol{x}}_k^{+} = \hat{\boldsymbol{x}}_k^{-} + K(\boldsymbol{y} - h(\hat{\boldsymbol{x}}_k))$
$C_k^{+} = (I_{9\text{x}9} - LH)C_k^{-}$
$\hat{\boldsymbol{x}}_{k+1}^{-} = A_H\hat{\boldsymbol{x}}_k^{+} + B_H\hat{\mathbf{z}}_{HMM}$
$C_{k+1}^{-} = A_H C_k^{+} + C_k^{+} A_H^{T} + Q_2$

Figure 4.4: The new EKF with HMM estimates incorporated into the dynamic update.

Figure 4.4 shows the EKF+HMM portion of the EKF+HMM system. The measurement update of the EKF+HMM remains the same as the EKF described in Section 2.3. However, the dynamic update is changed to incorporate the HMM estimates as inputs to the dynamic model. The new dynamic update is represented as

$$\hat{\mathbf{x}}_{k+1} = A_H\hat{\mathbf{x}}_k + B\hat{\mathbf{u}}_k, \tag{4.2}$$

where A_H is the new dynamic model, $\mathbf{x}_k$ is the state vector, B is the input matrix, and u is the input. The equation is expanded to

$$\begin{bmatrix} \hat{\mathbf{r}}_{k+1} \\ \hat{\mathbf{v}}_{k+1} \\ \hat{\mathbf{a}}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{r}}_{k} \\ \hat{\mathbf{v}}_{k} \\ \hat{\mathbf{a}}_{k} \end{bmatrix} + \begin{bmatrix} \frac{\delta t^2}{2} & \frac{\delta t^3}{6} \\ \delta t & \frac{\delta t^2}{2} \\ 1 & \delta t \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}}_{HMM} \\ \hat{\mathbf{j}}_{HMM} \end{bmatrix}. \tag{4.3}$$

Note that the estimate for acceleration, $\hat{\mathbf{a}}_{k+1}$, does not incorporate any states in its dynamic update. Instead, it only uses the inputs $\hat{\mathbf{a}}_{HMM}$ and $\hat{\mathbf{j}}_{HMM}$.

The state covariance matrix for the EKF+HMM, C, is updated with the equation

$$C_{k+1}^{-1} = A_H C_k^+ + C_k^+ A_H + Q_2, \tag{4.4}$$

where Q_2 is the process noise for the EKF+HMM. Because A_H does not incorporate any state information in its dynamic update of $\mathbf{a}$, the variance of the acceleration states are not continuously updated for the duration of the EKF+HMM estimation process. Instead, the acceleration variance is re-initialized during each dynamic update using only values from the process noise matrix, Q_2, which causes the acceleration variances to be constant.

This process noise matrix, Q_2, is initially created using Bryson's trick, where the maximum error variance of the training HMM estimates was used as the process noise of the system. Q_2 further tuned to produce the best possible estimates. Q_2 is populated both on the diagonals of the matrix and on the off-diagonals of the matrix. This characteristic means that there are still some dependencies between the acceleration states and the position and velocity states. Therefore, although acceleration estimates in the dynamic update incorporate no state information, the acceleration estimates are updated in the measurement update step of the filter based on the position and velocity state errors. It's important to note that this would not be possible without the correct structure of the Q_2 matrix. If the process noise matrix only contained values on the diagonals of the matrix, the acceleration estimates could not be updated using radar measurements.

4.2 Results

The EKF+HMM system was implemented on each of the test data set maneuvers. The total EKF+HMM estimation error of each maneuver was found. The sum of the absolute value of estimation error was found for each trajectory. The trajectory with the lowest sum of absolute error (Scenario 1 - "best") and high sum of absolute error (Scenario 2 - "worst") were used to depict the results. The following section shows a Monte Carlo simulation of each of these two trajectories.

The Monte Carlo simulation consisted of the maneuver being simulated 1000 times the same start and end position. Both the EKF and the EKF+HMM estimated the states of the UAV. The filters were initialized with the true states of the UAV. Process noise of the EKF and of the EKF+HMM were kept uniform across all maneuvers (although Q and Q_2 were different). Radar position and noise parameters remained constant. The mean error of the Monte Carlo simulation of both filters is compared.

4.2.1 Maneuver 1

Scenario 1

The mean accuracy over time for this Maneuver 1 trajectory was 94.79%. Figure 4.5 shows the average classification confidence of each maneuver HMM over time. There is a misclassification at the beginning of the trajectory, but the classification is quickly corrected to a high confidence classification of Maneuver 1. The average confidence of Maneuver 1's model quickly becomes less than -1, meaning that HMM estimates begin being incorporated into the EKF+HMM in about 2.5 seconds.

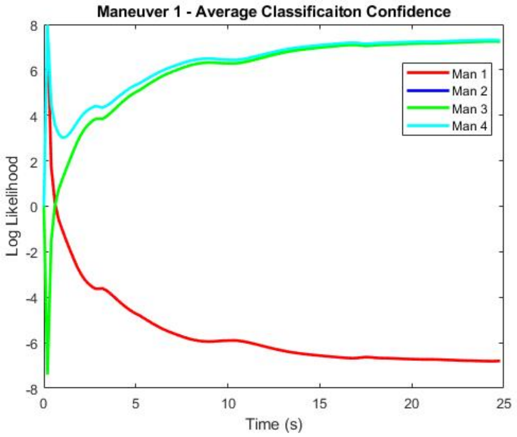
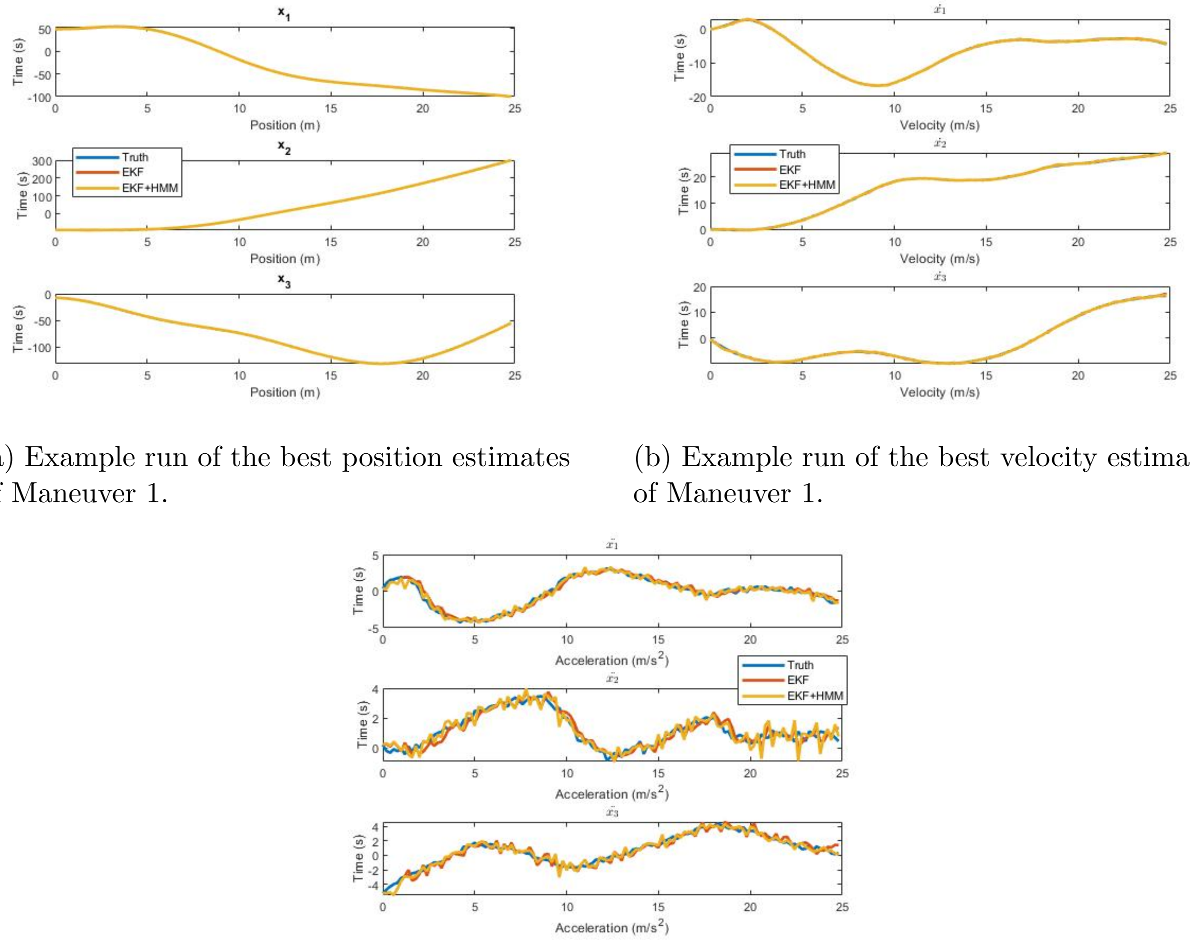


Figure 4.5: Confidence of classification over time.

Figure 4.6 presents an example run of the Monte Carlo simulation. Truth, EKF estimates, and EKF+HMM estimates of each UAV state (position, velocity, and acceleration) are shown. The greatest difference in the EKF and EKF+HMM can be seen in the acceleration estimates. The EKF estimates tend to lag behind the truth more so than the EKF+HMM estimates. Both filters have similar noise in their estimation.

Note that the EKF+HMM does not have high initial error in its estimation despite the misclassification shown in Figure 4.5. The HMM estimates that resulted from the misclassification were filtered out using the fault detection discussed in Section 4.1.2.



(a) Example run of the best position estimates of Maneuver 1.

(b) Example run of the best velocity estimates of Maneuver 1.

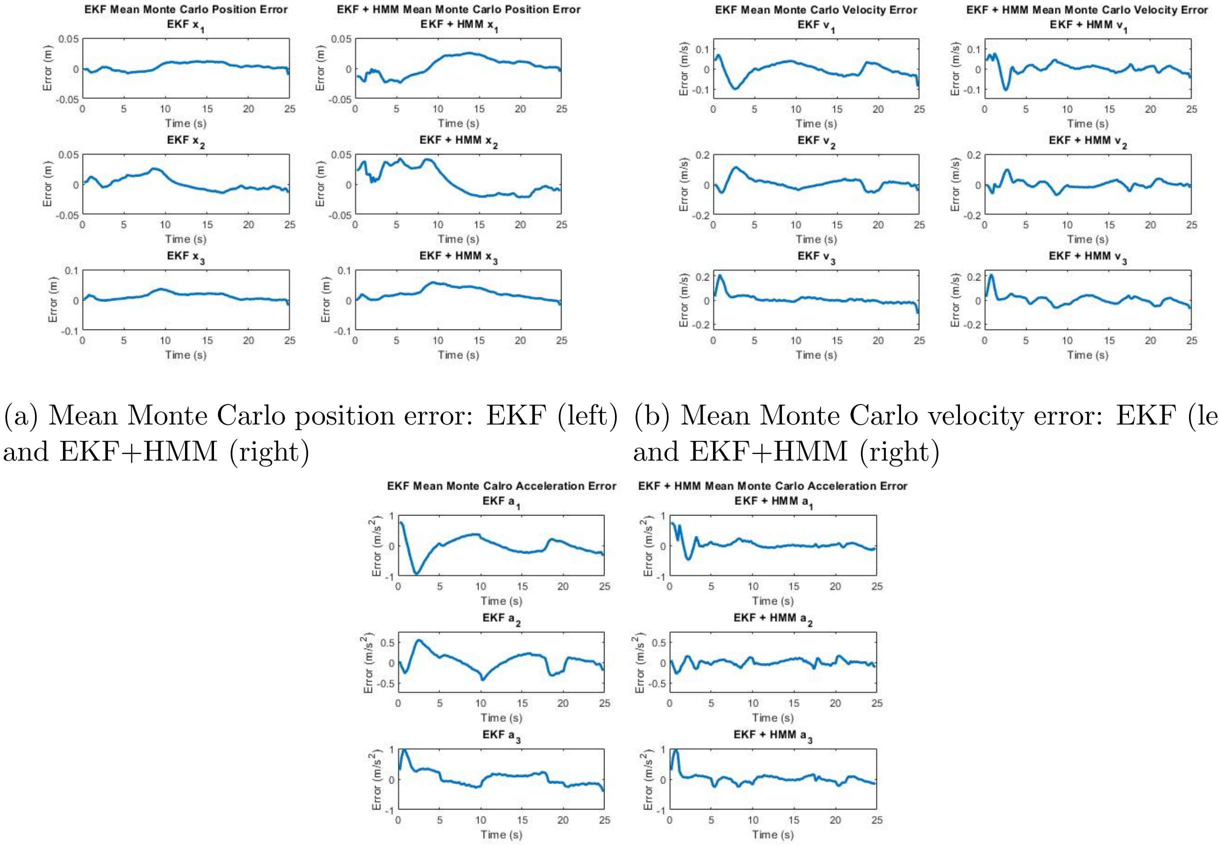
(c) Example run of the best acceleration estimates of Maneuver 1.

Figure 4.6: Example run of EKF+HMM for Maneuver 1.

Figures 4.7a, 4.7b, and 4.7c show the mean position, velocity, and acceleration error of the EKF estimates (left) and the EKF+HMM estimates (right). The EKF has less overall position estimate error; this is the result of a higher values in the covariance matrix for position than the EKF+HMM, which leads to greater correction during the radar measurement update step of the EKF.

The velocity estimates of the EKF+HMM for v_1 and v_2 are superior to the EKF. The EKF has errors that originate from the filter lagging when estimating the UAV states - which creates enduring errors that build over time. The EKF+HMM does not lag as much as the EKF and therefore, has lower magnitude of errors. However, error is introduced through incorrect HMM estimates, which create small consistent error in the EKF+HMM estimates. These errors are less detrimental than EKF errors when estimating v_1 and v_2. However, when estimating v_3, the EKF+HMM performs worse than the EKF.

The EKF+HMM has superior performance in estimating acceleration states of the UAV during Maneuver 1. The acceleration estimate most heavily relies on the dynamic update, because there are no direct acceleration measurement using the radar. The HMM estimates aid in the EKF+HMM's dynamic model predicting acceleration with more accuracy. EKF+HMM acceleration estimation error is not entirely zero-mean; there are clear errors shown that originate from the HMM estimates. However, these errors are much less than the errors of the EKF.



(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

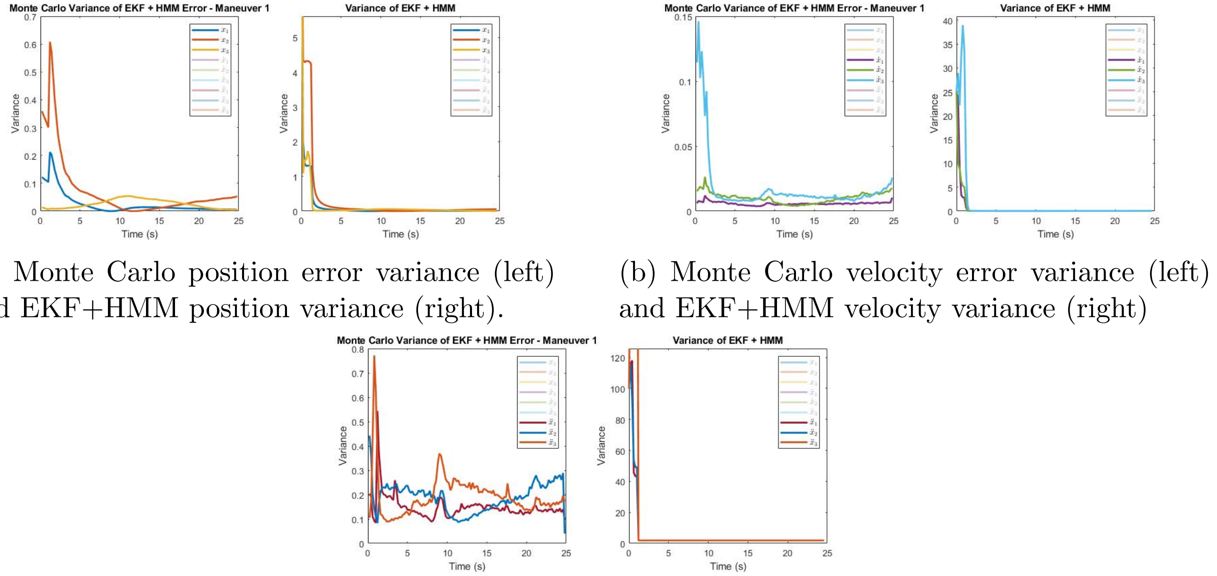
(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.7: Comparison of Monte Carlo errors for EKF and EKF+HMM.

To further examine the EKF+HMM, the Monte Carlo error variance for each state of the EKF+HMM is compared against the variance calculated by the EKF+HMM in Figure 4.8. The state variances of the EKF+HMM are initially high, because there is no classification and therefore, the EKF+HMM matches the EKF formulation. Once classification occurs, the variance of the EKF+HMM states drops drastically. The quick misclassification of Maneuver 3 causes the EKF+HMM variance to drop in two stages. It experiences an initial drop when classified as Maneuver 3. Then, the variances rises, because the confidence is not at or below −1 for any model. This causes the EKF+HMM to return to the standard EKF formulation for a few seconds. The EKF+HMM variances drop for a final time when the maneuver is classified as Maneuver 1 with high confidence and the EKF+HMM once again incorporates HMM estimates.

The position and velocity variance track the general trends of the error variance, while the acceleration variance of the EKF+HMM stays completely constant despite in spikes in acceleration error variance. This occurs because the acceleration variance is re-initialized at each dynamic update in the EKF. Because the EKF+HMM acceleration variance remains constant, tuning the EKF+HMM process noise matrix is especially important.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right)

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.8: Variance of errors versus EKF+HMM variance.

Scenario 2

This section displays the "worst case" Maneuver 1 EKF+HMM. The mean accuracy over time for the Monte Carlo was 70.27%. Figure 4.9 shows the average confidence in each model during classification. There is a quick initial misclassification of Maneuver 1 as Maneuver 3, but it is corrected almost immediately. The maneuver reaches a confidence of less than -1 slower than Scenario 1 however - indicating that multiple models may have high log-likelihoods throughout the initial portion of the maneuver.

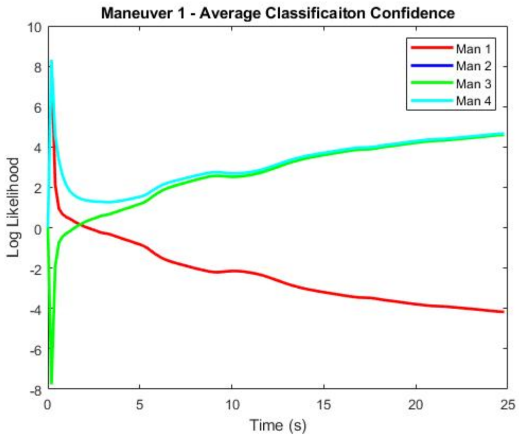
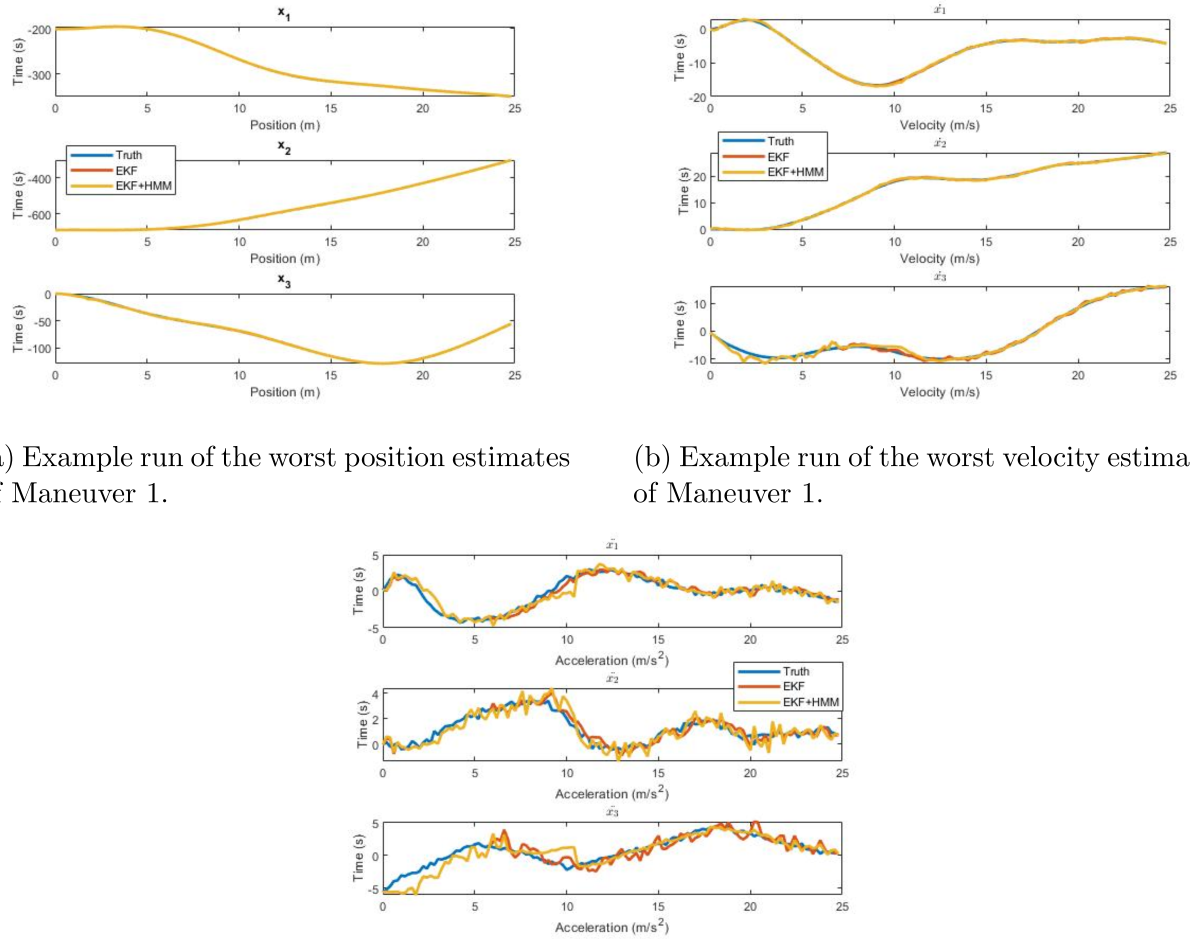


Figure 4.9: Confidence of classification over time.

Figure 4.10 shows an example of EKF and EKF+HMM estimates over time when compared to truth for this maneuver. The HMM estimates that could have been incorporated into the EKF+HMM estimates during the Maneuver 3 misclassification were filtered out through fault detection, because they created too much of a change in acceleration. Instead, the EKF+HMM matches the EKF for the first 6 seconds of the trajectory. Then, seen in Figures 4.10c and 4.10b, from 6 to 10, the EKF+HMM has significant error when estimating the North and Down position velocity and acceleration. This error is the result of faulty HMM estimates. The EKF+HMM is able to recover from the error and continues estimating UAV states with slightly more accuracy than the EKF for the remainder of the maneuver.



(a) Example run of the worst position estimates of Maneuver 1.

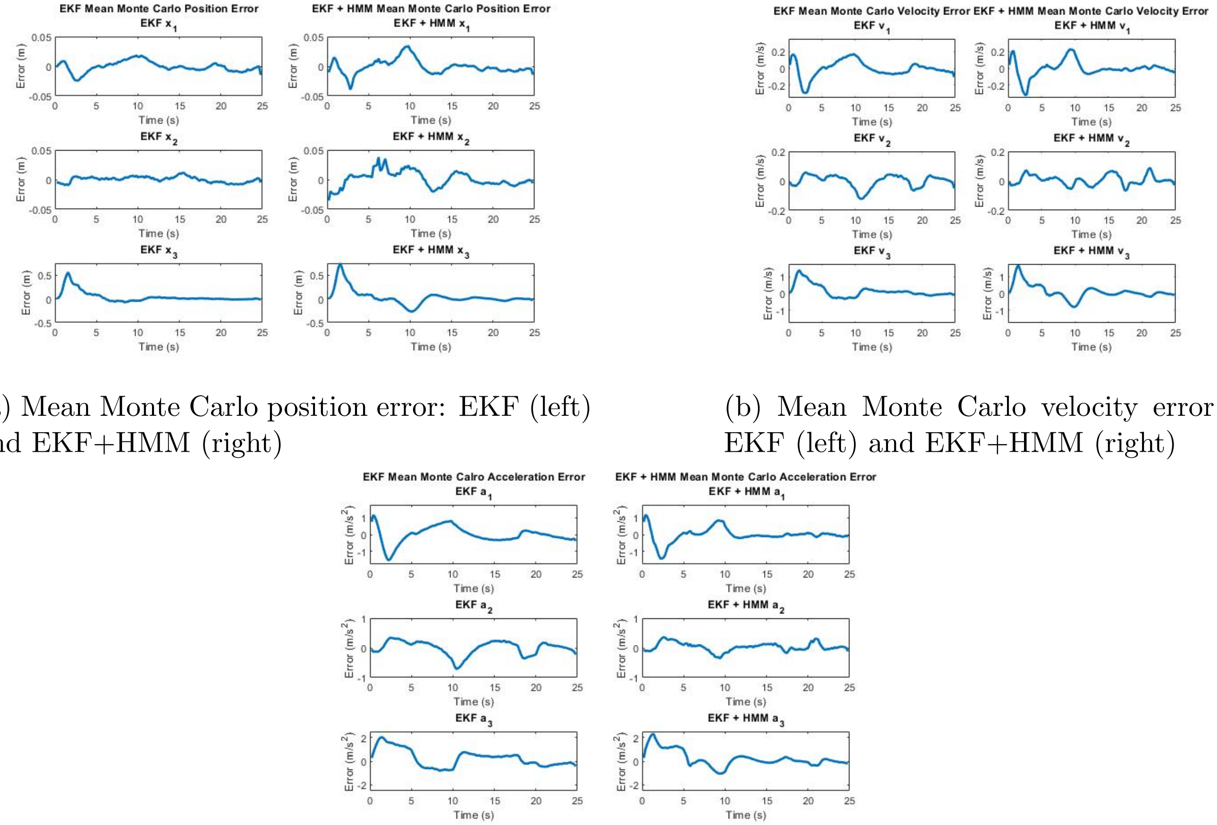
(b) Example run of the worst velocity estimates of Maneuver 1.

(c) Example run of the worst acceleration estimates of Maneuver 1.

Figure 4.10: Example run of EKF+HMM for Maneuver 1.

The mean Monte Carlo position error of both the EKF and EKF+HMM are shown in Figure 4.11a. Overall, the EKF has better position estimates than the EKF+HMM. THe EKF+HMM position estimates contain some error as the result of incorrect HMM acceleration and jerk estimates that are incorporated into the position state through the dynamic model. Another factor to this increased error may be that the variances and covariances of the covariance matrix are typically lower for the EKF+HMM than the EKF, which causes the radar measurements to weight less heavily in correcting the states.

The mean Monte Carlo velocity estimate errors are shown in Figure 4.11b. The EKF+HMM has similar or increased error for its estimates of v_1 and v_3 for the first 12 seconds of flight. The estimate error of v_2 is less with the EKF+HMM than the EKF for the entirety of the maneuver, but both estimates have very clear non-zero trends in their data. The acceleration estimates of the EKF+HMM, in contrast, are almost entirely better than the EKF estimates, as seen in Figure 4.11c. There are similar errors in the early stages of flight, but after 12 seconds, the errors of the EKF+HMM are lower than the EKF.



(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

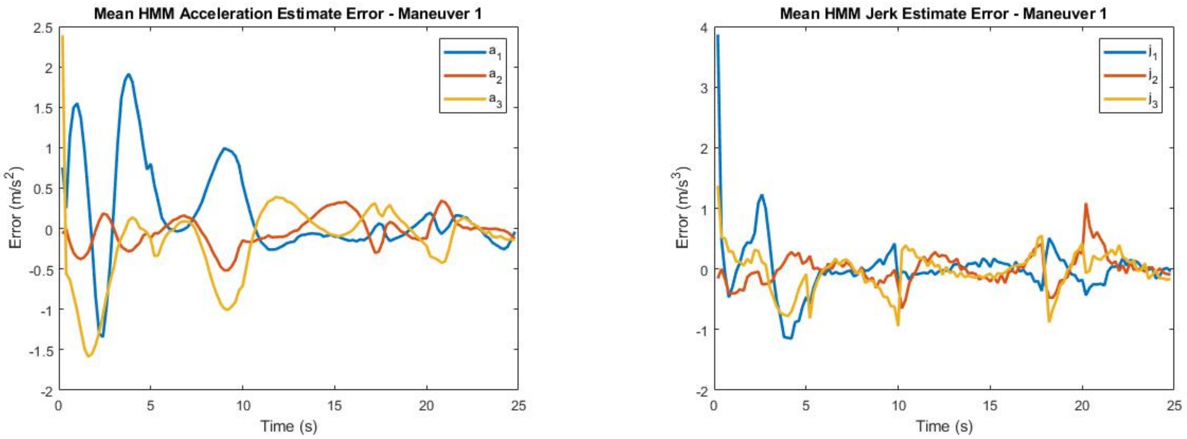
(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.11: Comparison of Monte Carlo errors for EKF and EKF+HMM.

The average error of HMM estimates of acceleration and jerk are shown in Figure 4.12. HMM estimates serve as inputs to the dynamic model of the EKF+HMM. The errors shown are before any incorporation into the EKF+HMM or any correction from measurements.

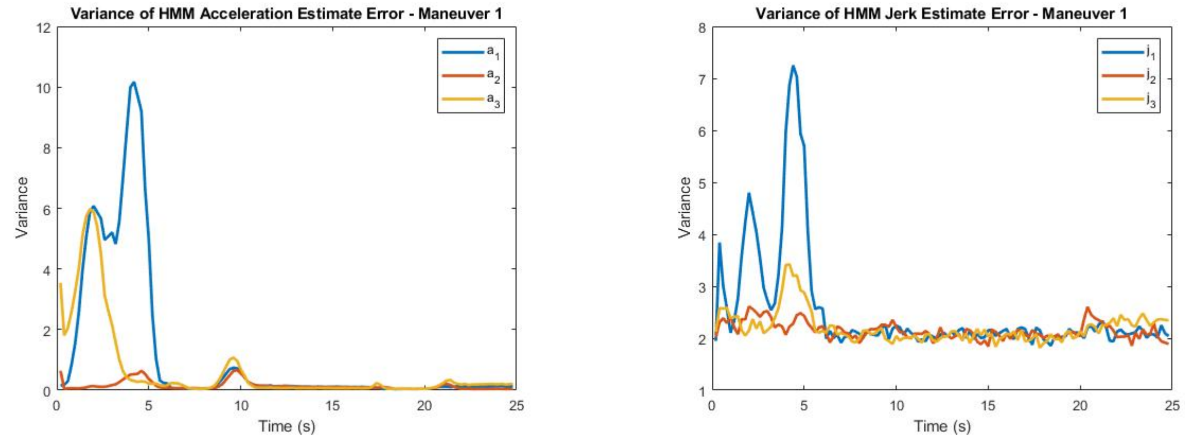
The effects of these errors, particularly the HMM acceleration estimate error, are reflected in Figure 4.11. Figure 4.13 shows the variance of these HMM estimate errors. The areas of high variance show times at which the HMM estimates were especially erroneous.



(a) Mean error of HMM acceleration estimate for the worst run.

(b) Mean error of HMM jerk estimate for the worst run.

Figure 4.12: Mean error of HMM estimates.



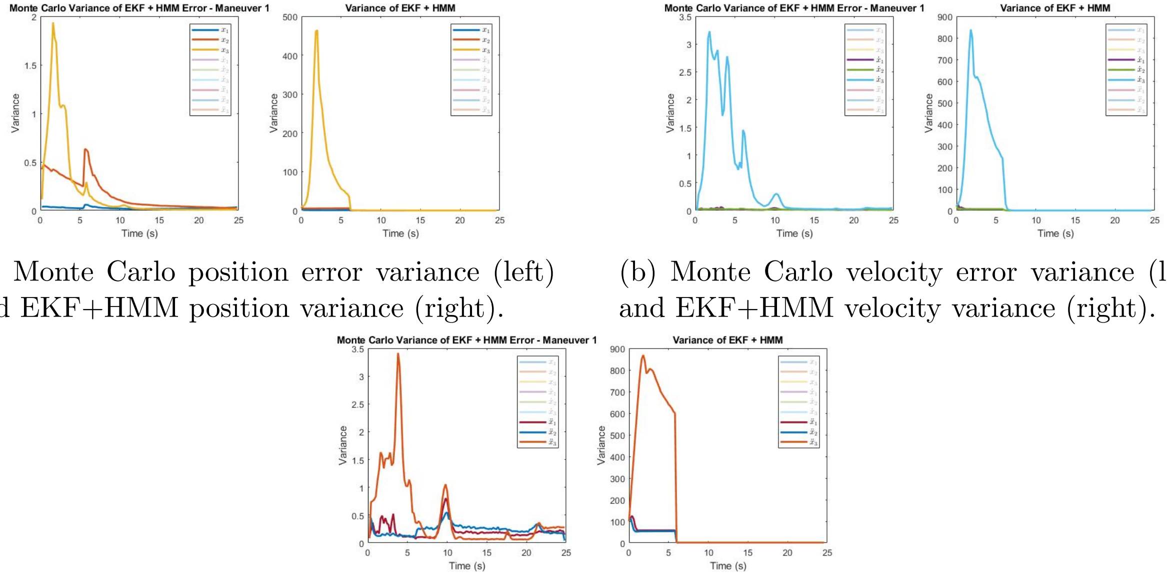
(a) Variance of HMM acceleration estimate error for the worst run.

(b) Variance HMM jerk estimate error for the worst run..

Figure 4.13: Error variance of HMM estimates.

The Monte Carlo error variance of the EKF+HMM is compared to the EKF+HMM calculated variance in Figure 4.14. Areas of high EKF+HMM state variance correspond to times that the EKF+HMM uses the EKF dynamic update instead of incorporating HMM estimates. This reflects times of uncertain classifications, where models have a low confidence (or no confidence) in the current classification. Once the maneuver is fully classified, both the experimental and calculated variance drop dramatically.

Note that the HMM acceleration estimate error variance (Figure 4.13a) is much greater than the Monte Carlo acceleration error variance (Figure 4.14c) before the maneuver is classified correctly. However, after correct classification the graphs are nearly identical – indicating that the current measurement correction does not have much affect on the EKF+HMM acceleration estimate. This could be changed depending on the tuning of the process noise matrix.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right).

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.14: Variance of errors versus EKF+HMM variance.

4.2.2 Maneuver 2

Scenario 1

The confidence in each model's classification is shown in Figure 4.15. There is an initial high confidence classification of Maneuver 3, before the system quickly corrects to a high confidence classification of Maneuver 2. This misclassification of Maneuver 3 is a common trend in the maneuver classification and may suggest λ_3 is overfit to a similar start data point on all maneuvers.

The mean classification accuracy over time for the Monte Carlo simulation was 96.83%. The high confidence classification of Maneuver 2 is almost immediate. Therefore, the EKF+HMM begins using HMM estimates early in the trajectory.

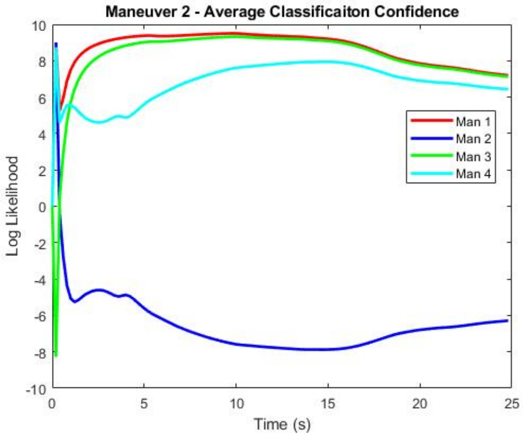
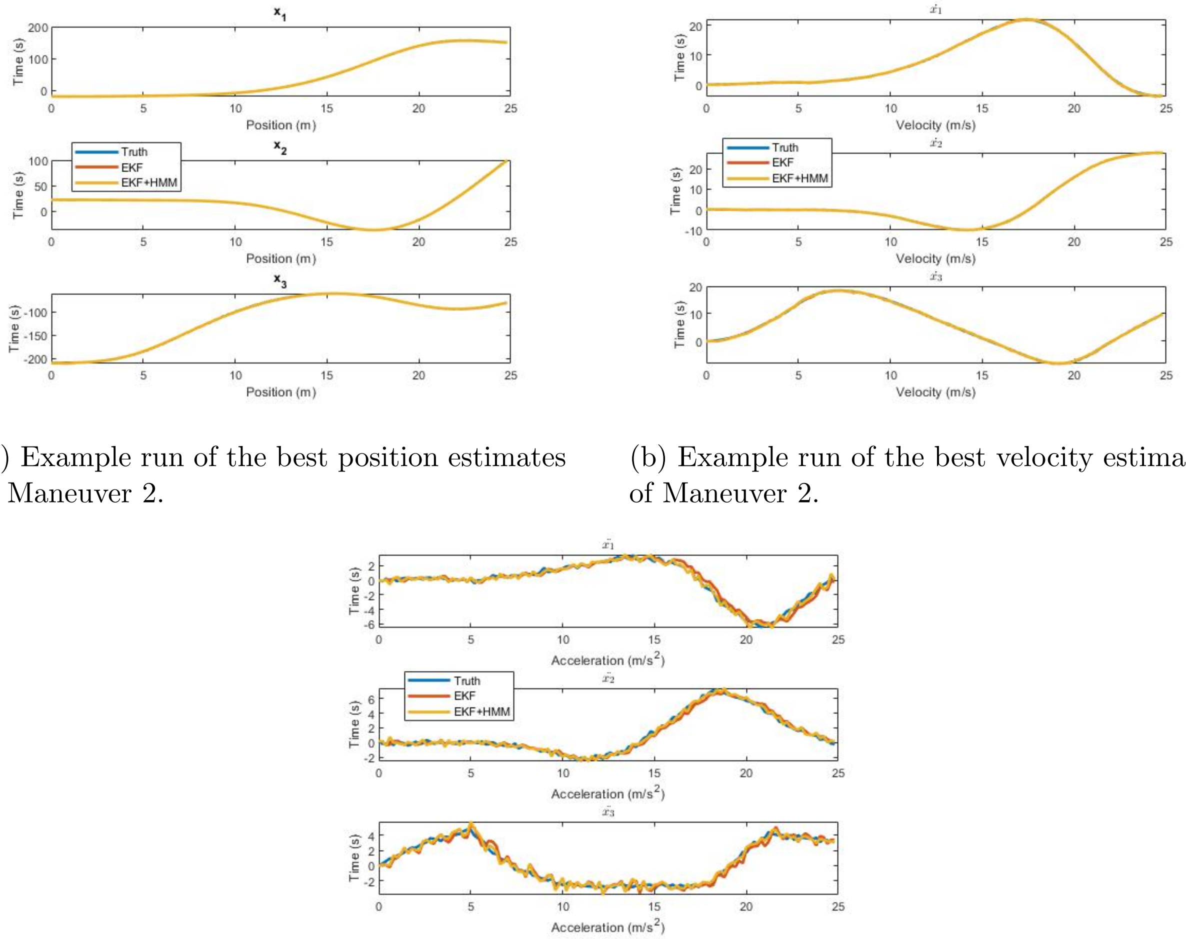


Figure 4.15: Confidence of classification over time.

Figure 4.16 shows an example run of the Monte Carlo - with the position, velocity, and acceleration shown for true states as well as EKF and EKF+HMM estimates. It is difficult to distinguish the position and velocity estimates of the EKF+HMM and EKF. However, differences in the acceleration estimates are more easy to determine from this figure. Both estimates have a similar amount of noise, but the EKF lags behind truth and the EKF+HMM when estimating acceleration states.



(a) Example run of the best position estimates of Maneuver 2.

(b) Example run of the best velocity estimates of Maneuver 2.

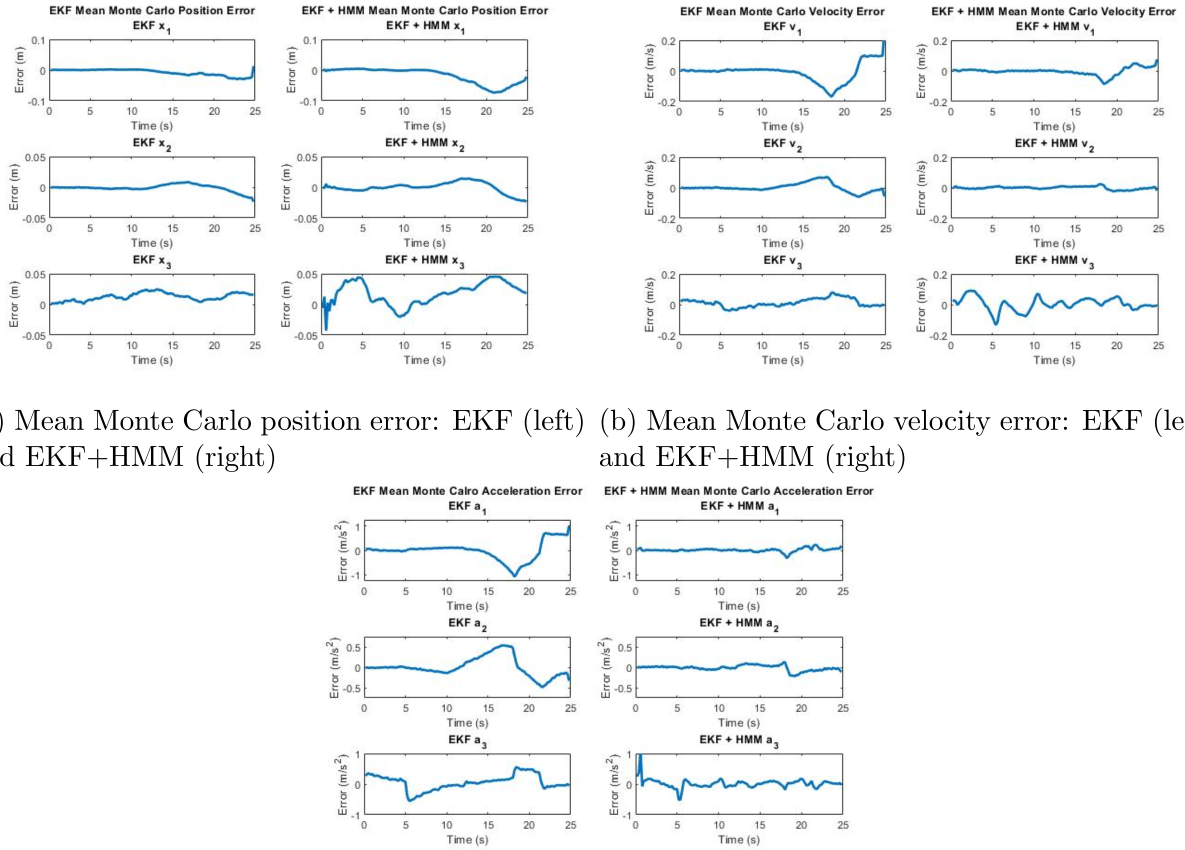
(c) Example run of the best acceleration estimates of Maneuver 2.

Figure 4.16: Example run of EKF+HMM for Maneuver 2.

The position error of the EKF and EKF+HMM for the Monte Carlo simulation is compared in Figure 4.17a. The errors of the EKF+HMM are slightly higher compared to the EKF. The EKF+HMM clearly improves estimates for v_1 and v_2, as seen in Figure 4.17b. The EKF+HMM increases error in v_3 estimates, while the EKF (although not zero-mean) has much lower error. The increased EKF+HMM error for v_3 is the result of incorporating incorrect HMM estimates.

Finally, the EKF and EKF+HMM mean acceleration errors are compared (Figure 4.17c). The EKF+HMM greatly improves all estimates of acceleration in comparison to the EKF.

Although some error trends seen in the EKF estimates are still present in the EKF+HMM errors, they are greatly reduced. Note the similar error in the two filters for a_1 and a_2 at 18 seconds. Both experience a sharp error, but the EKF+HMM greatly reduces the overall effects of this error. The EKF+HMM acceleration estimate with the most error is a_3. Once again, this is a reflection of incorrect HMM estimates of acceleration that are not fully corrected through radar measurements.



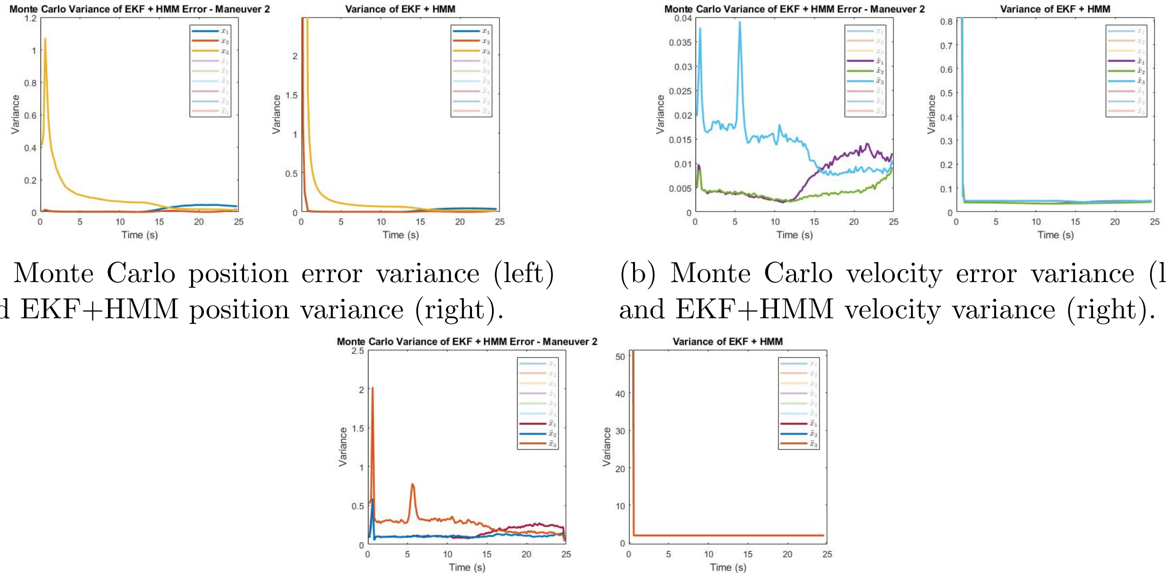
(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.17: Comparison of Monte Carlo errors for EKF and EKF+HMM.

The variance of the EKF+HMM was evaluated using the EKF+HMM error variance of the Monte Carlo simulation and the filter state variance in Figure 4.18. The position variance follows the position error variance almost exactly, while the velocity variance remains constant for most of the maneuver and slightly greater than the velocity error variance. The acceleration variance remains completely constant once the maneuver has been classified with high confidence.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right).

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.18: Variance of errors versus EKF+HMM variance.

Scenario 2

The maneuver's Monte Carlo simulation had a mean classification accuracy over time of 96.82%. The confidence in Maneuver 2's HMM reached less than -1 early in the maneuver, as seen in Figure 4.19. An example of estimating this test run of Maneuver 2 is shown in Figure 4.20. Overall, the EKF+HMM still performed well in comparison to the EKF. This is perhaps the most clear when comparing the estimation of a_3 in Figure 4.20c.

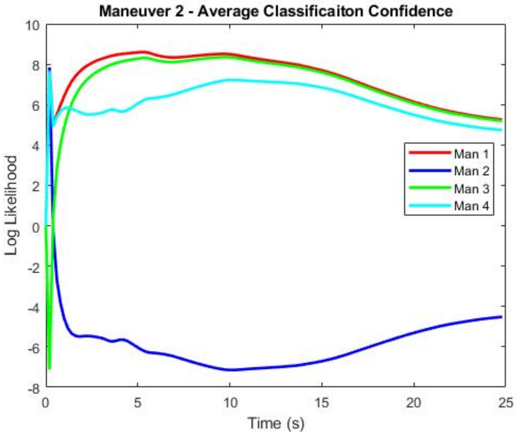
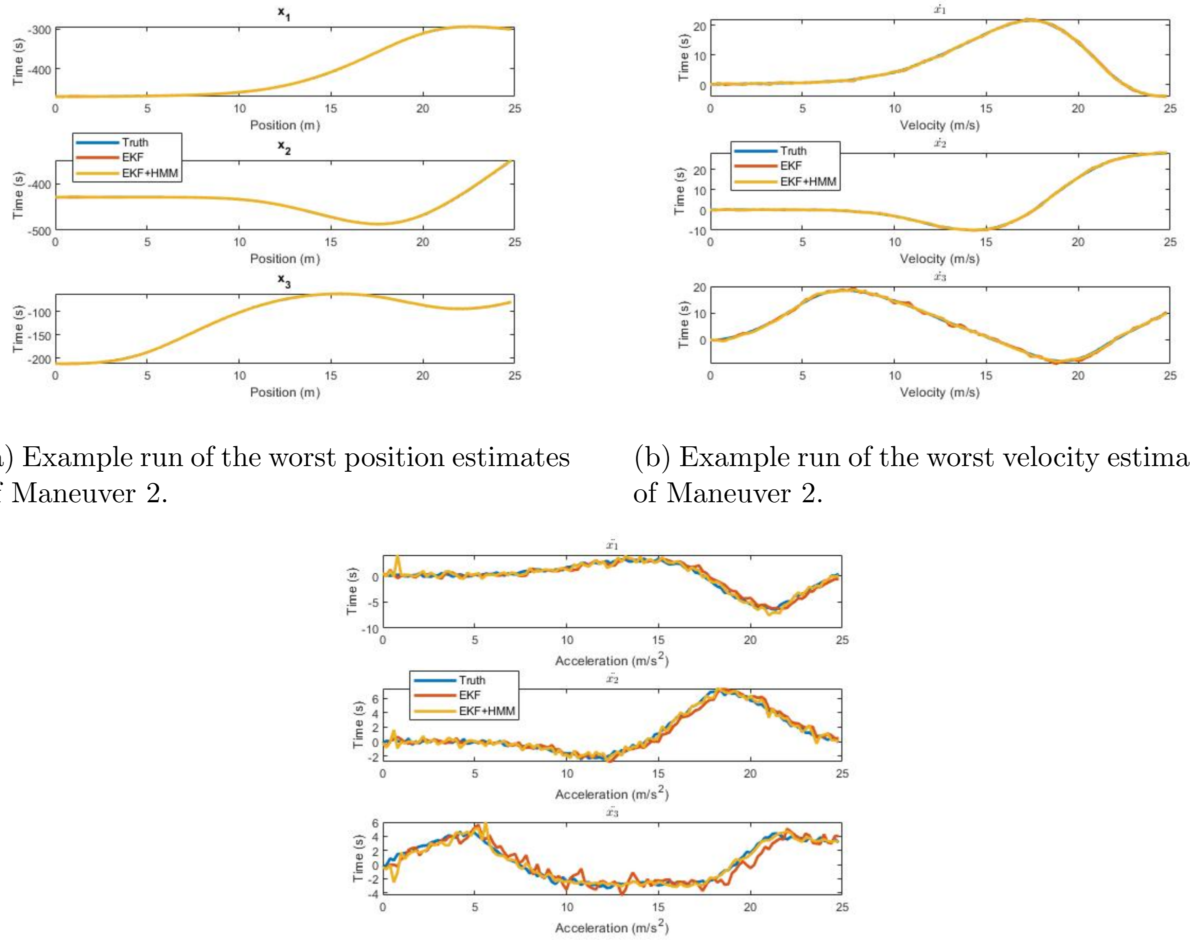


Figure 4.19: Confidence of classification over time.



(a) Example run of the worst position estimates of Maneuver 2.

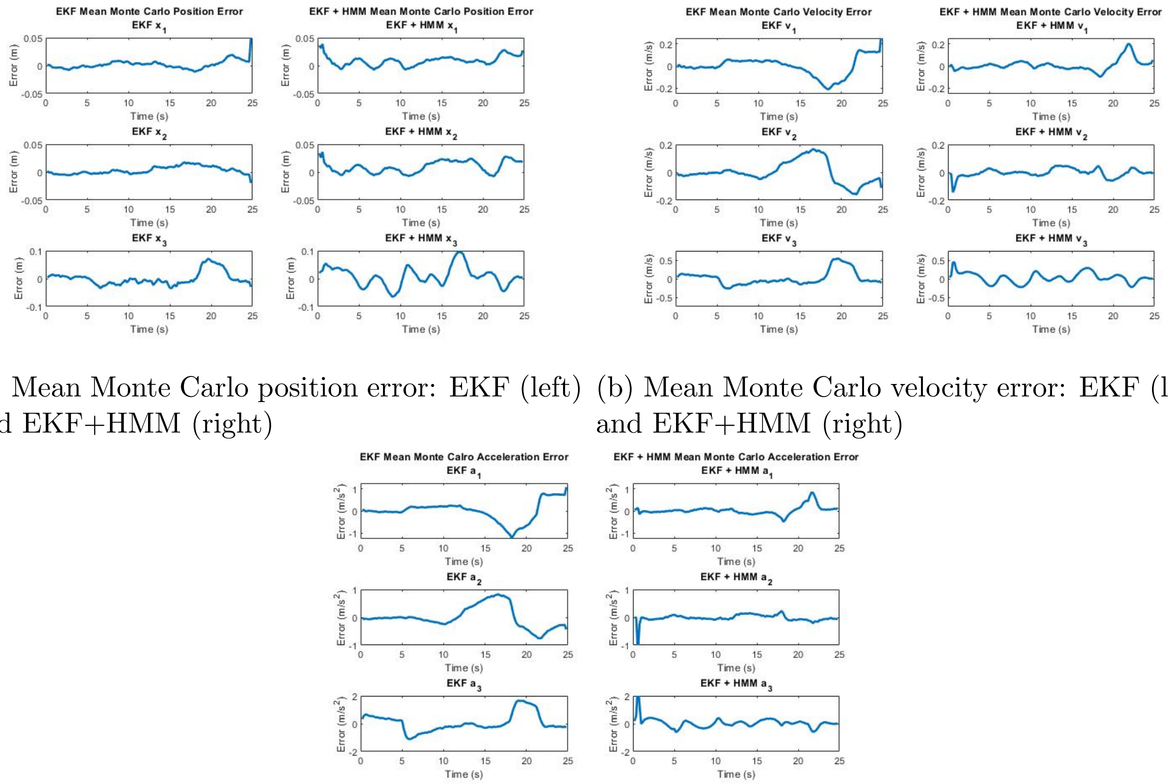
(b) Example run of the worst velocity estimates of Maneuver 2.

(c) Example run of the worst acceleration estimates of Maneuver 2.

Figure 4.20: Example run of EKF+HMM for Maneuver 2.

Figure 4.21a shows the Monte Carlo position mean error of both the EKF and the EKF+HMM. The position estimation error of both filters remained relatively low. The EKF had close to zero-mean position error throughout most of the maneuver, before additional error was introduced in the final 8 seconds. This increase is because acceleration is more dynamic in the final portions of Maneuver 2. In contrast to the EKF, the EKF+HMM position estimation errors were not at or near zero-mean for any portion of the trajectory.

The velocity errors of the EKF+HMM (Figure 4.21b) exhibit similar error trends seen in the position estimates. In the case of v_1 and v_2, these estimation errors are less than the EKF errors. The EKF+HMM estimation error of v_3 still exceeds that of the EKF. The acceleration error of the EKF+HMM once again displays similar trends to the EKF, but these errors are much smaller than those of the EKF, as seen in Figure 4.21c. Note, also, a sharp increase in EKF+HMM acceleration error as a result of Maneuver 3 misclassification, which caused incorrect HMM estimates to be used as inputs.



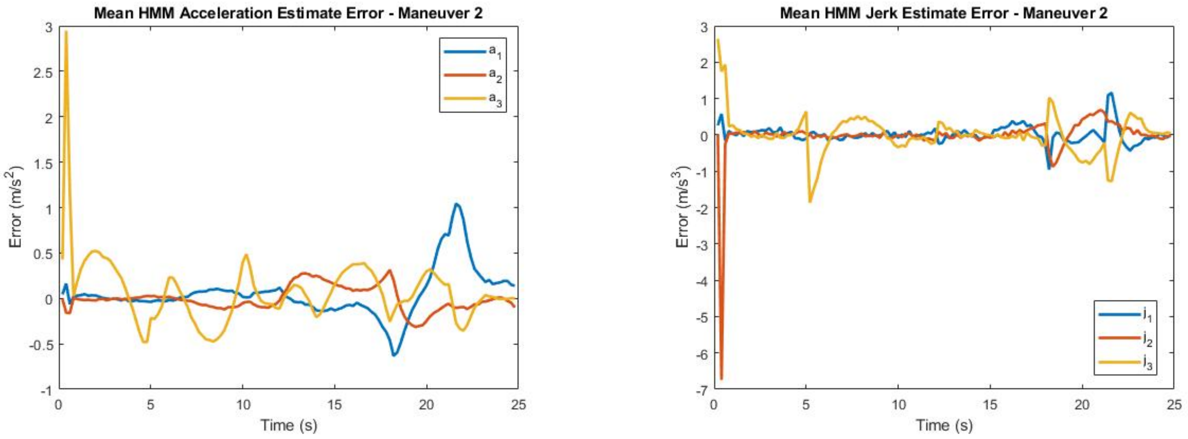
(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.21: Comparison of Monte Carlo errors for EKF and EKF+HMM.

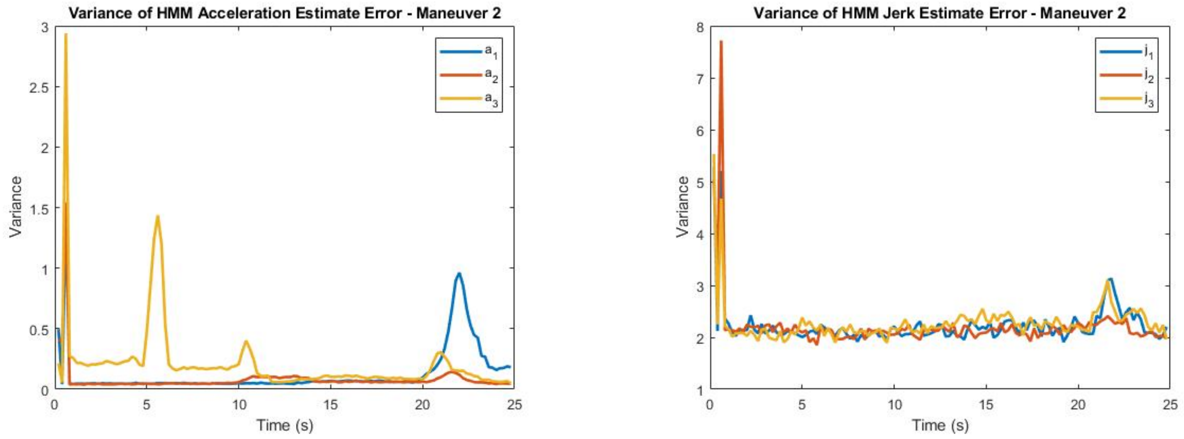
The mean error of acceleration and jerk HMM estimates are shown in Figure 4.22. Both jerk and acceleration have a high initial error that is the result of the incorrect Maneuver 3 classification. The error trends exhibited in the mean acceleration very closely align with the mean EKF+HMM acceleration error found in Figure 4.21. The variances of the HMM estimates' error are displayed in Figure 4.23.



(a) Mean error of HMM acceleration estimate for the worst run.

(b) Mean error of HMM jerk estimate for the worst run.

Figure 4.22: Mean error for HMM estimates.

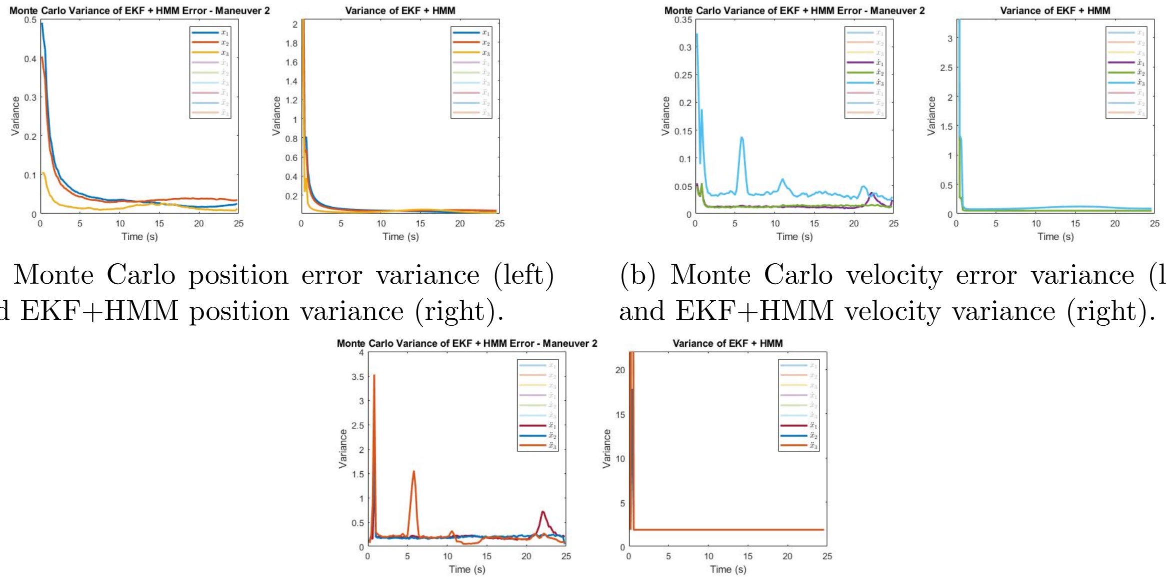


(a) Variance of HMM acceleration estimate error for the worst run.

(b) Variance HMM jerk estimate error for the worst run.

Figure 4.23: Error variance for HMM estimates.

The states' variances calculated from the EKF+HMM is compared to the Monte Carlo error variance in Figure 4.24. The results follow a similar pattern to the previous Maneuver 2 Monte Carlo simulation. Both position and velocity variance match well with the error variance, while the acceleration variance remains constant and greater than its error variance.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right).

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.24: Variance of errors versus EKF+HMM variance.

4.2.3 Maneuver 3

Scenario 1

The “best” test trajectory of Maneuver 3 (in terms of the EKF+HMM) is discussed in this section. Throughout the Monte Carlo simulation, the maneuver had a mean classification accuracy over time of 96.14%. The average confidence in Maneuver 3’s HMM (shown in Figure 4.25) is initially quite high, but then drops below the “confident” threshold for a period of about 2 seconds, because the other maneuvers produce high log-likelihoods for Maneuver 3’s observations. Maneuver 3 is occasionally misclassified as Maneuver 2 during this time period, which is likely due to similarities between the maneuvers’ initial few seconds. After the 3 seconds of confusion, the confidence in Maneuver 3 increases and remains there for the rest of UAV flight.

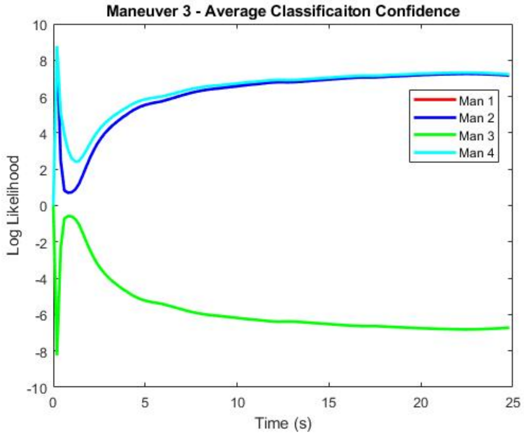
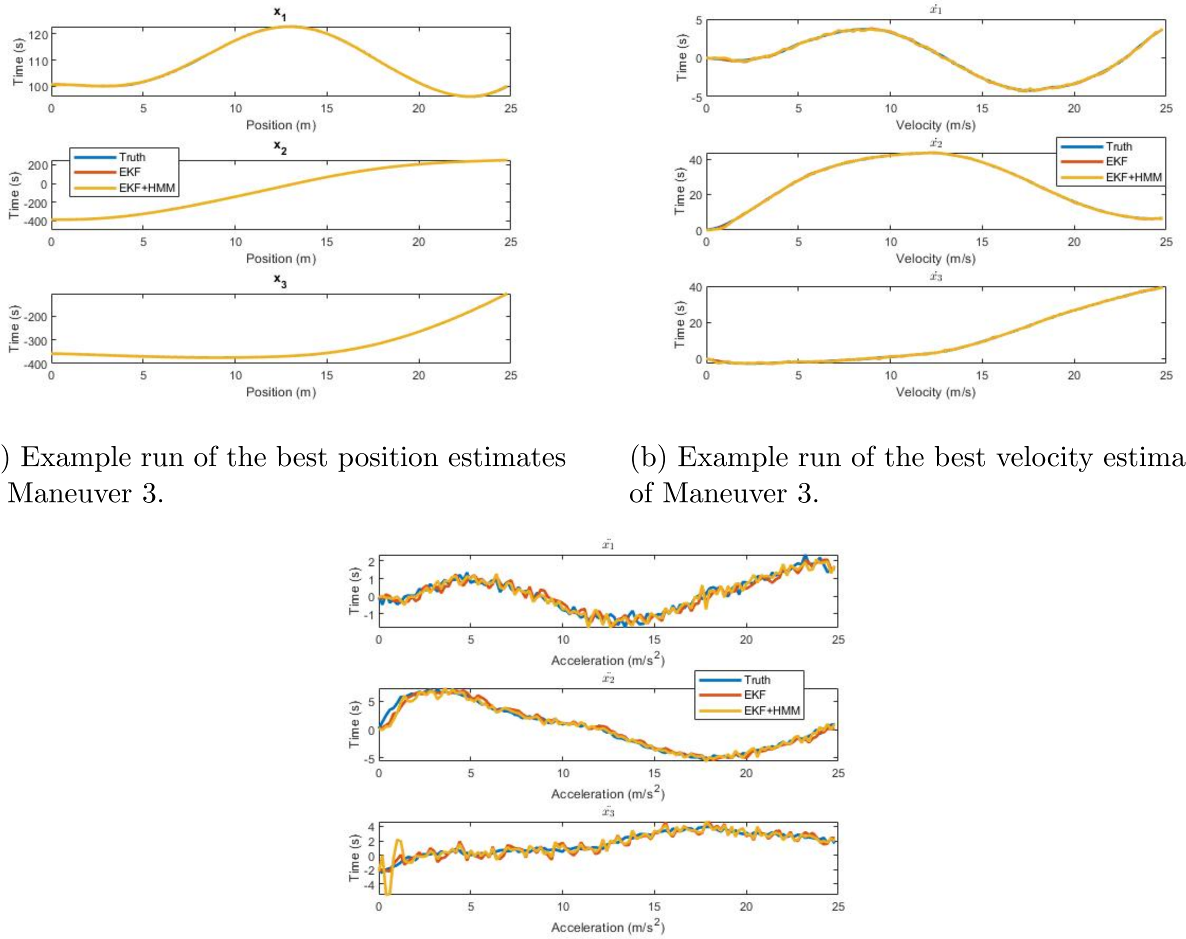


Figure 4.25: Confidence of classification over time.

An example of one of the runs used in the Monte Carlo is shown in Figure 4.26. The EKF and EKF+HMM estimates are difficult to distinguish. Notice that the EKF lags the EKF+HMM acceleration estimates slightly.



(a) Example run of the best position estimates of Maneuver 3.

(b) Example run of the best velocity estimates of Maneuver 3.

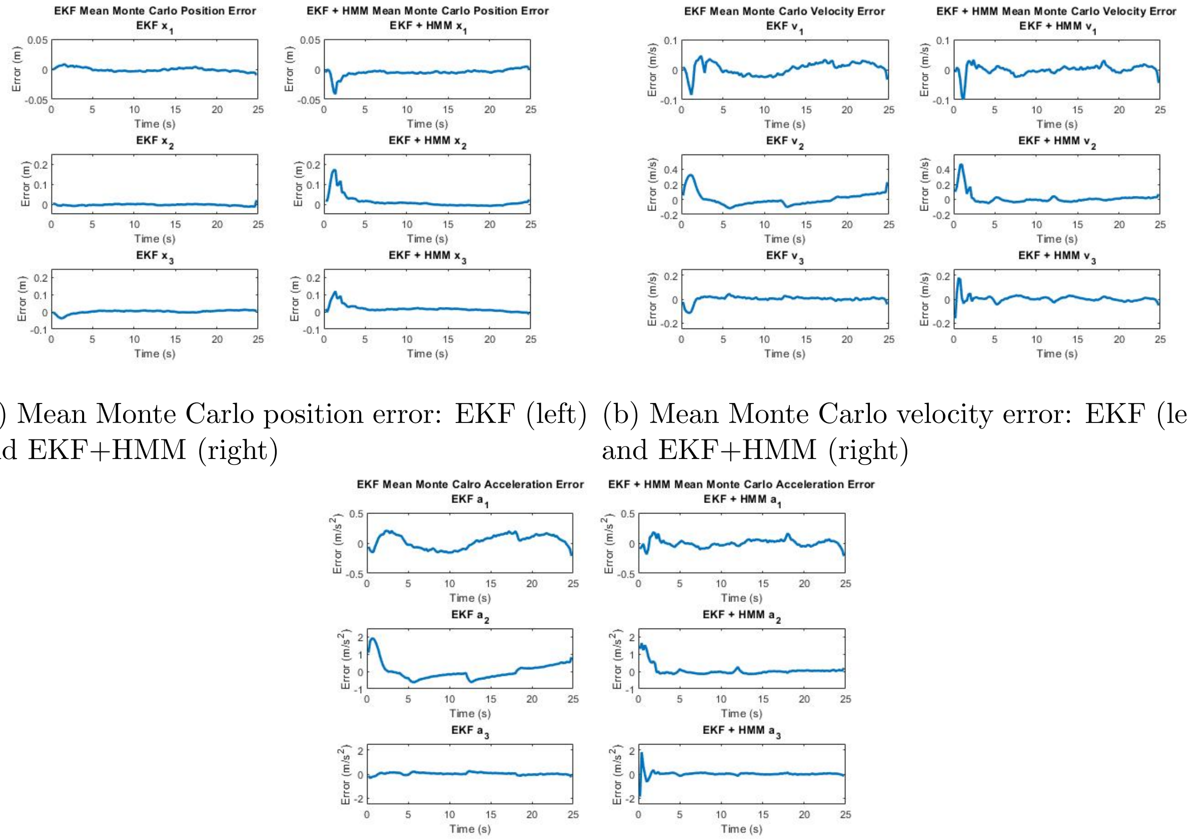
(c) Example run of the best acceleration estimates of Maneuver 3.

Figure 4.26: Example run of EKF+HMM for Maneuver 3.

The Monte Carlo position error of both the EKF and EKF+HMM is compared in Figure 4.27a. Both filters have low error throughout the maneuver. The position error of the EKF+HMM has a brief increase during the first 5 seconds of flight, which is a result of an occasional misclassification as Maneuver 2.

Likewise, in Figure 4.27b, the EKF and EKF+HMM estimation error are relatively low for velocity. The biggest difference in these two filters occurs when estimating v_2. The EKF has an error from estimation offset throughout UAV flight, but the EKF+HMM is capable of mitigating most of this error. In estimation of v_1, the EKF+HMM also decreases the error seen in the EKF. Only v_3, which is less dynamic, has better EKF estimates than EKF+HMM estimates.

Figure 4.27c shows the acceleration error for both the EKF and EKF+HMM. Here it is most clear the improvements of the EKF+HMM compared to the EKF. EKF error in estimating a_1 and a_2 are decreased (and in the case of a_2 are almost completely mitigated), while both filters have comparable errors when estimating a_3.



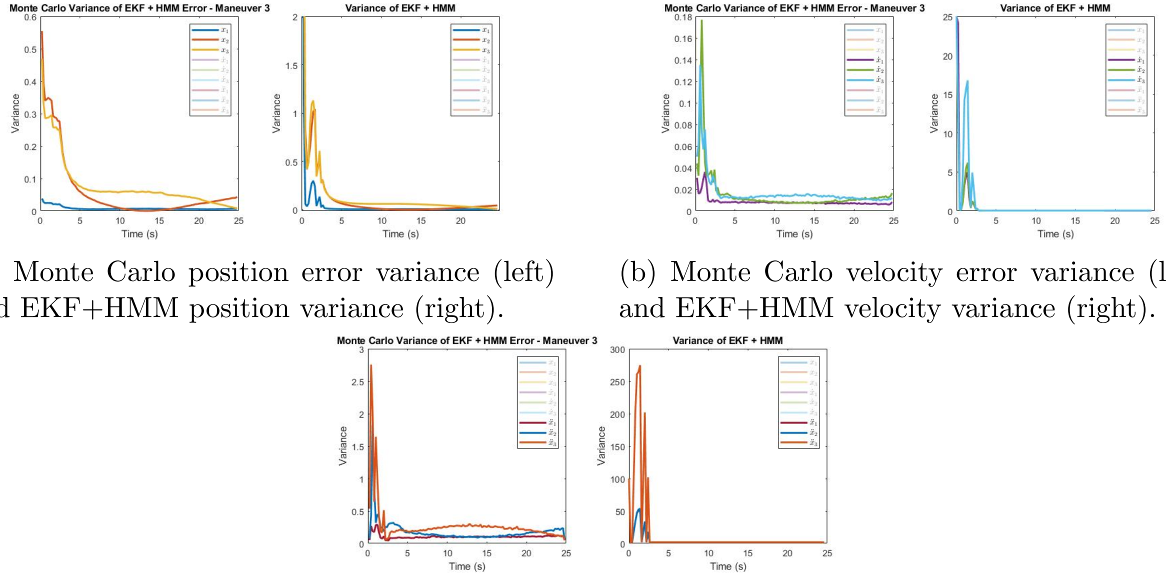
(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.27: Comparison of Monte Carlo errors for EKF and EKF+HMM.

A comparison of Monte Carlo EKF+HMM error variance to the EKF+HMM state variance can be seen in Figure 4.28. Overall, the EKF+HMM state variance followed the trends of the EKF+HMM Monte Carlo error variance, while typically being an order of magnitude greater (or more) in value.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right).

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.28: Variance of errors versus EKF+HMM variance.

Scenario 2

The "worst case" EKF+HMM estimation results for Maneuver 3 are explored in this section. Maneuver 3 had a mean classification accuracy over time of 96%. The average confidence in each HMM for classification over time (Figure 4.29) shows initial high confidence in Maneuver 3; this confidence drops briefly from time $t = 1$ to $t = 3$ before increasing once more. The results are similar to the confidence of the "best case" Maneuver 3 run.

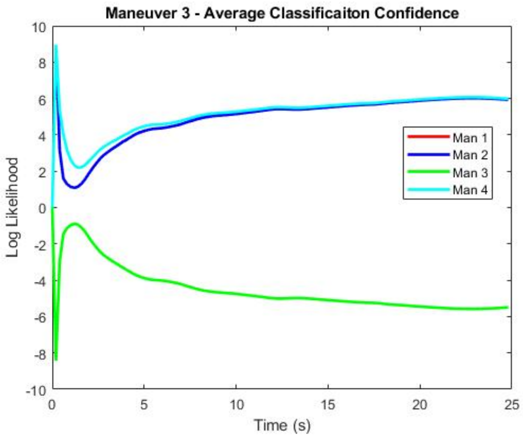
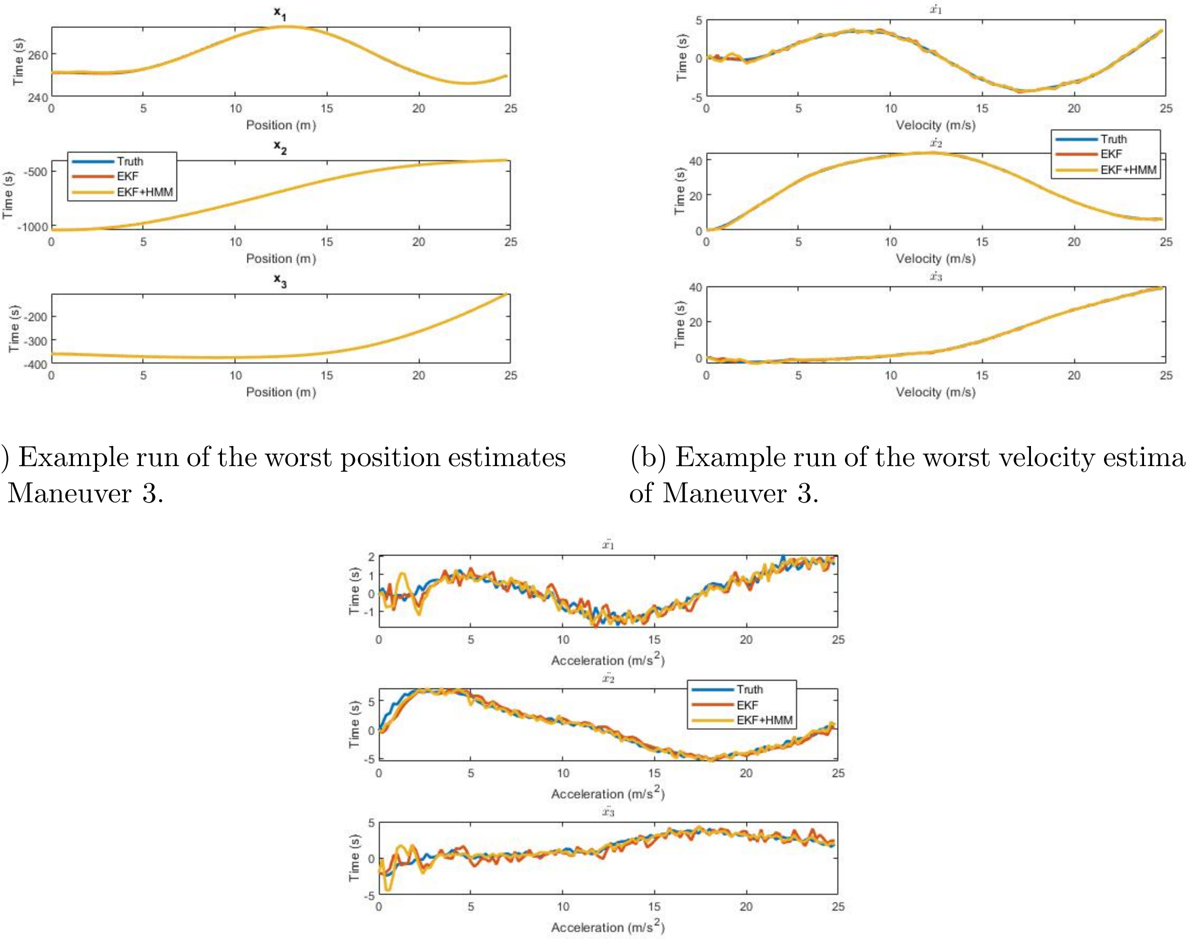


Figure 4.29: Confidence of classification over time.

Figure 4.30 shows an example data run during the Monte Carlo. Both the EKF+HMM and EKF estimate the maneuver well and with low noise. Some initial error in the EKF+HMM can be seen when estimating a_1 and a_3. During this example run, the EKF+HMM began using HMM estimate inputs, then returned to a simple EKF model, and finally switched back to the EKF+HMM model all within the first 3 seconds. This switching is a result in the fluctuations in the confidence of the Maneuver 3 classification and causes an increase in error for some EKF+HMM estimates.



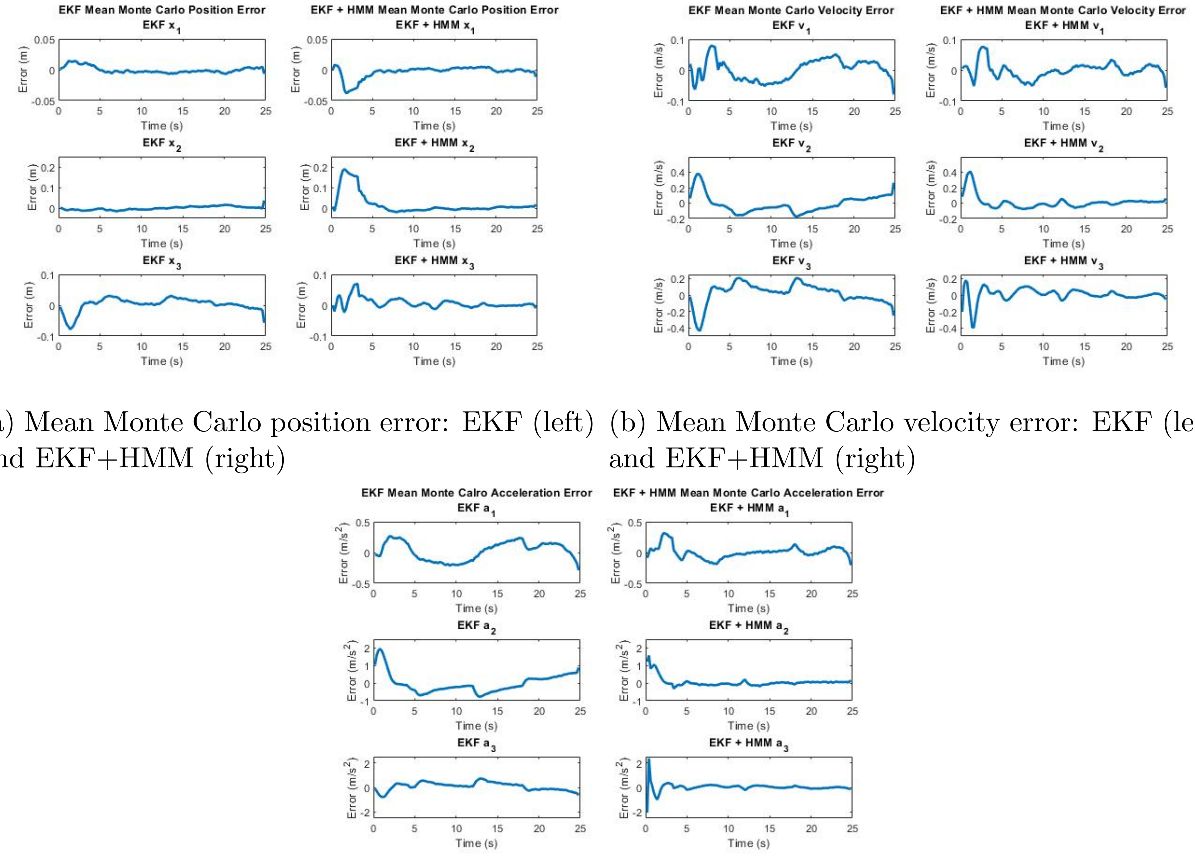
(a) Example run of the worst position estimates of Maneuver 3.

(b) Example run of the worst velocity estimates of Maneuver 3.

(c) Example run of the worst acceleration estimates of Maneuver 3.

Figure 4.30: Example run of EKF+HMM for Maneuver 3.

The mean Monte Carlo state estimation error for both the EKF and EKF+HMM are compared in Figure 4.31. The position estimates for both filters have similar error characteristics except in the initial phases of flight. During this time, the EKF+HMM has similar or increased error compared to the EKF when estimating velocity and acceleration. However, for the remaining portion of flight, the EKF+HMM has lower error in velocity and acceleration.



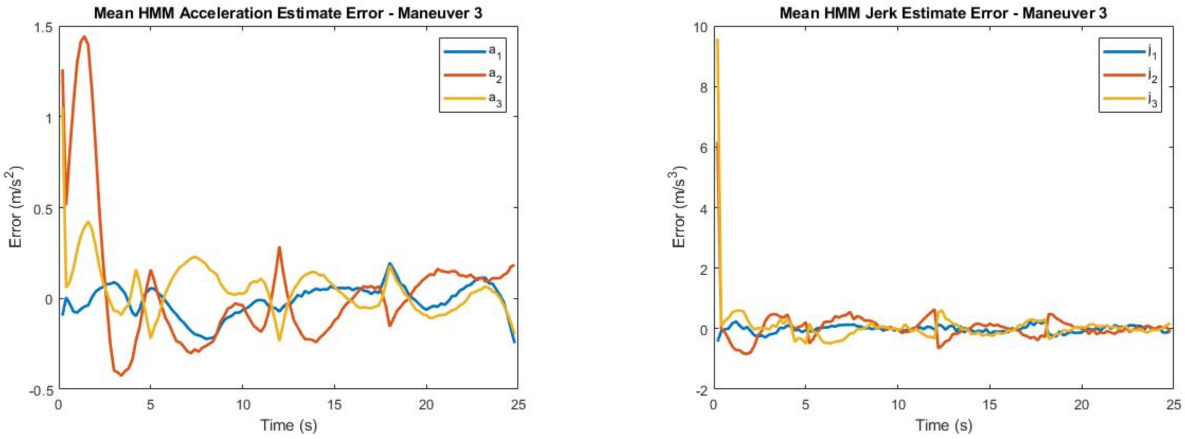
(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.31: Comparison of Monte Carlo errors for EKF and EKF+HMM.

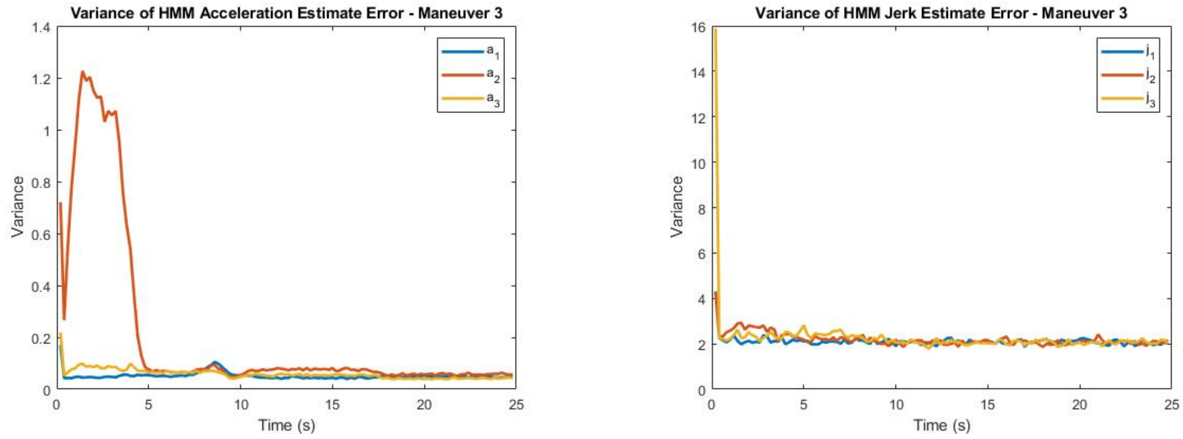
The average HMM acceleration and jerk estimate error are shown in Figure 4.32. Although both graphs shown high initial estimate error, most of this error is not incorporated into the filter. During this time period the confidence in Maneuver 3's HMM was low enough that its HMM estimates were not used as EKF+HMM inputs. The variance of these errors (Figure 4.33) show high variance in estimation errors during the first 5 seconds of flight. This corresponds to the uncertainty shown in the confidence measure of Maneuver 3.



(a) Mean error of HMM acceleration estimate for the worst run.

(b) Mean error of HMM jerk estimate for the worst run..

Figure 4.32: Mean error for HMM estimates.

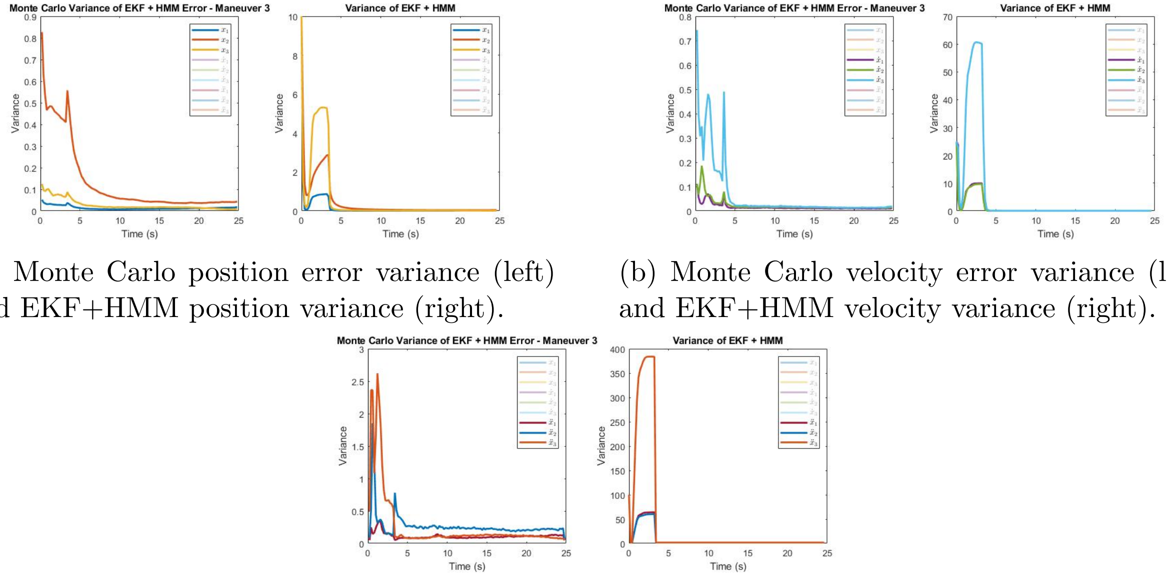


(a) Variance of HMM acceleration estimate error for the worst run.

(b) Variance HMM jerk estimate error for the worst run.

Figure 4.33: Error variance for HMM estimates.

Finally, the EKF+HMM Monte Carlo state error variance is compared to the EKF+HMM state variance in Figure 4.34. Like with the "best case" of Maneuver 3, the EKF+HMM variances follow the trend of the error variances while remaining an order of magnitude greater (or more) in value.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right).

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.34: Variance of errors versus EKF+HMM variance.

4.2.4 Maneuver 4

Scenario 1

The “best case” scenario of Maneuver 4 EKF+HMM estimation is discussed in this section. Maneuver 4 had an accuracy over time, on average, of 89.30%. The confidence in each maneuver’s HMM is shown in Figure 4.35, which shows some confusion between Maneuver 2 and Maneuver 4 in the first 5 seconds of flight before Maneuver 4 is established as the correct classification.

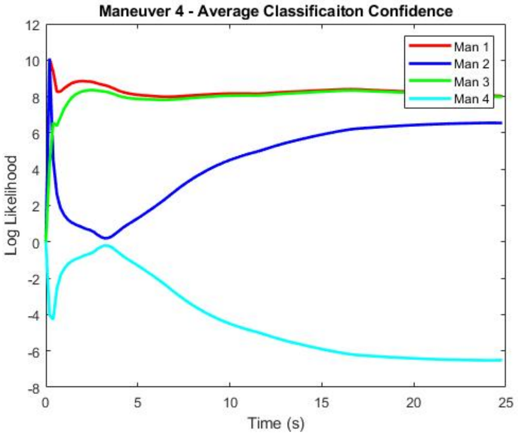
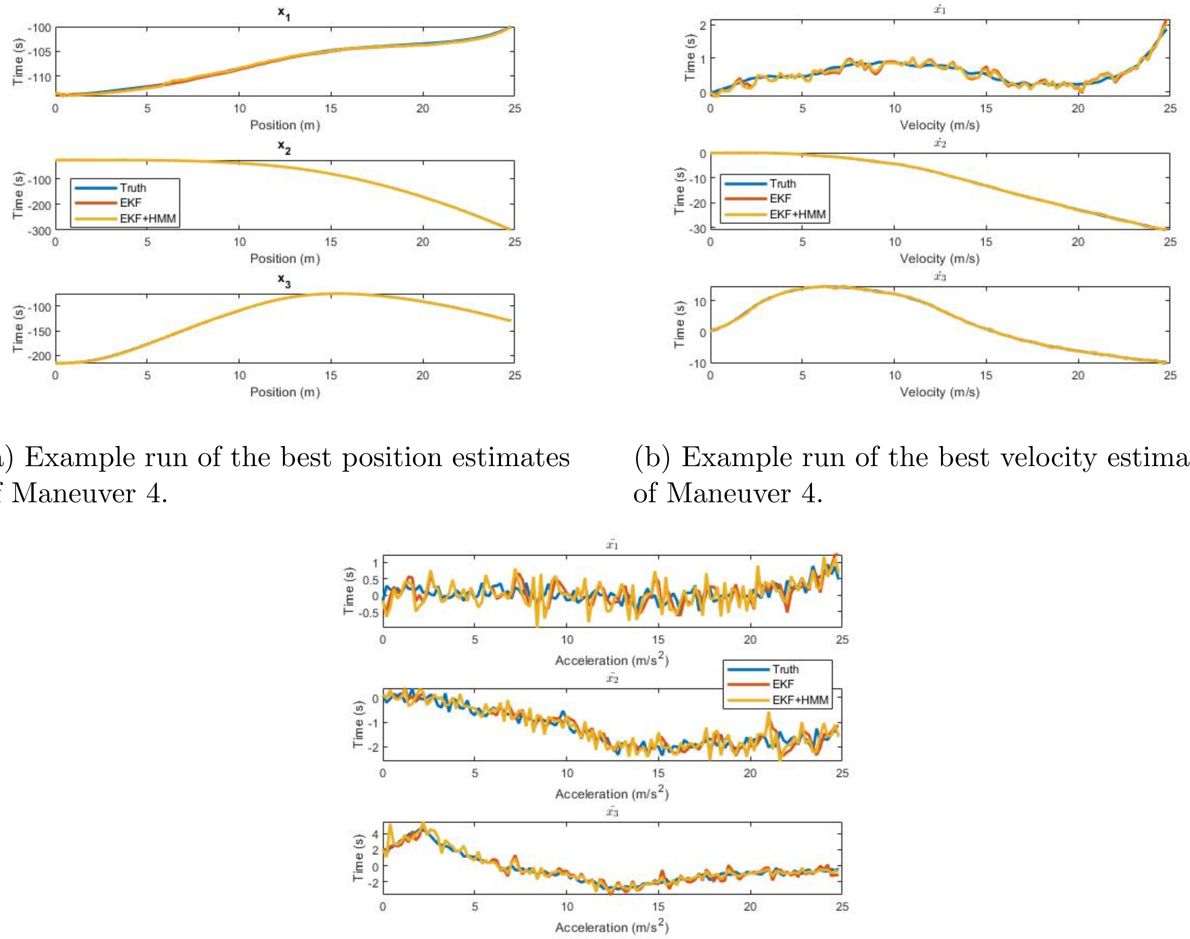


Figure 4.35: Confidence of classification over time.

Figure 4.36 shows an example Maneuver 4 trajectory when estimated by the EKF and EKF+HMM. The EKF+HMM is noisier than the EKF, but both filters are capable of estimating the UAV states. It's important to note that Maneuver 4 is the least dynamic of the maneuvers, so it is expected that the EKF should be capable of estimating the UAV states without any additional HMM estimate input, as established in Section 2.4.4.



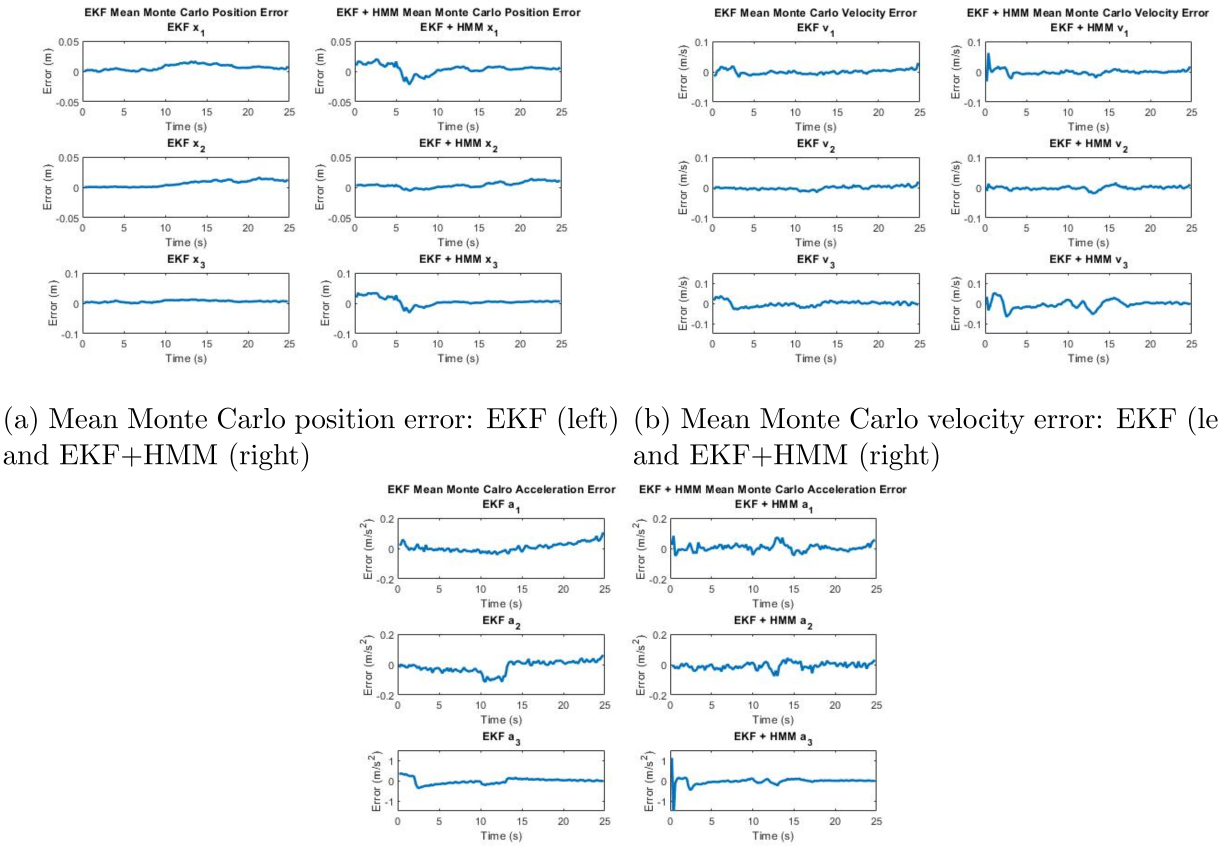
(a) Example run of the best position estimates of Maneuver 4.

(b) Example run of the best velocity estimates of Maneuver 4.

(c) Example run of the best acceleration estimates of Maneuver 4.

Figure 4.36: Example run of EKF+HMM for Maneuver 4.

The mean Monte Carlo state error for the EKF and EKF+HMM estimates are shown in Figure 4.37. Position error is low for both filters, but the EKF+HMM does have some initial error from initial HMM estimates. The velocity error is low for both filters as well, but the EKF+HMM does introduce some additional errors from 10 to 15 seconds. The acceleration error is minimal for both the EKF and EKF+HMM. Both of these systems struggle to estimate acceleration from 10 to 15 seconds.



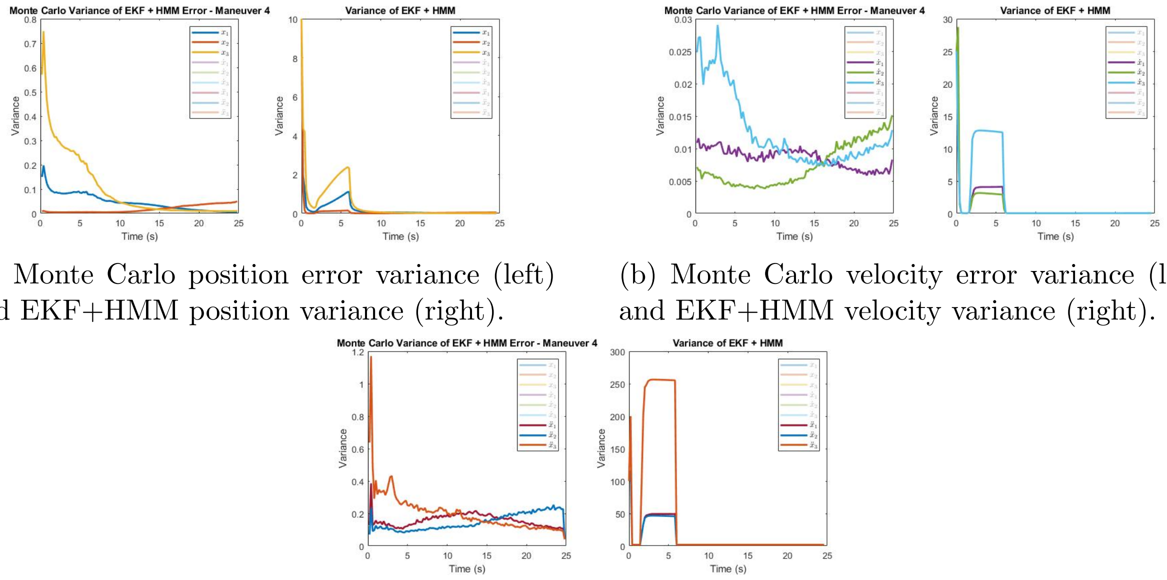
(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.37: Comparison of Monte Carlo errors for EKF and EKF+HMM.

The variance of each EKF+HMM state estimate's error is compared to the EKF+HMM state variance in Figure 4.38. Instances in which the EKF+HMM switches from using HMM inputs to using the typical EKF formulation cause in increase in the EKF+HMM state variance. These points of increase mirror the average confidence shown in Figure 4.35.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right).

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.38: Variance of errors versus EKF+HMM variance.

Scenario 2

The results of the "worst case" EKF+HMM estimation during Maneuver 4 is shown in this section. The maneuver had a mean classification accuracy over time of 76.80%. The maneuver was first correctly classified as Maneuver 4, but then incorrectly classified (with relatively high confidence – sometimes so much so that HMM estimates were used) as Maneuver 2, before being re-classified as Maneuver 4. The average confidence in each maneuver's HMM is shown in Figure 4.39.

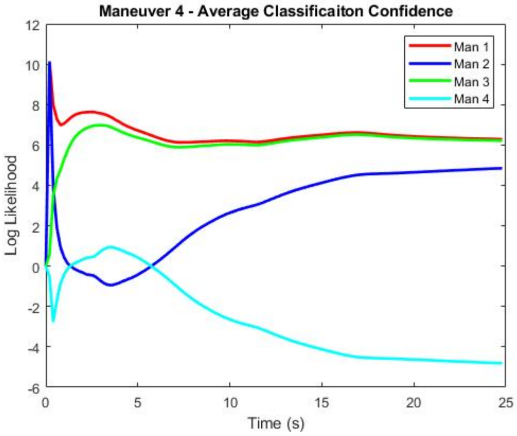
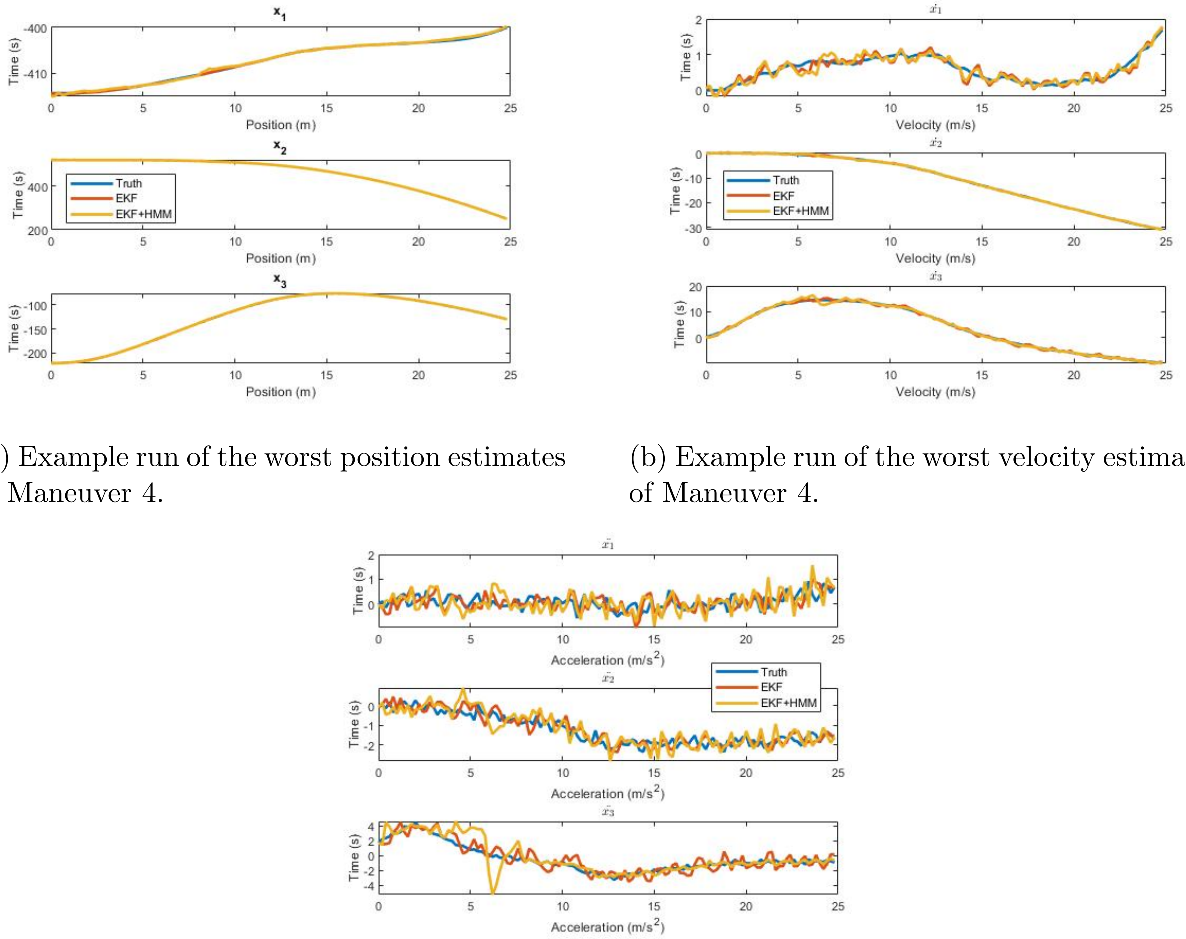


Figure 4.39: Confidence of classification over time.

An example of this worst case estimation event (Figure 4.40) shows high error in the EKF+HMM estimates of acceleration (particularly for a_3) in the first ten seconds of flight. After this period of high error, the EKF and EKF+HMM have comparable state estimates.



(a) Example run of the worst position estimates of Maneuver 4.

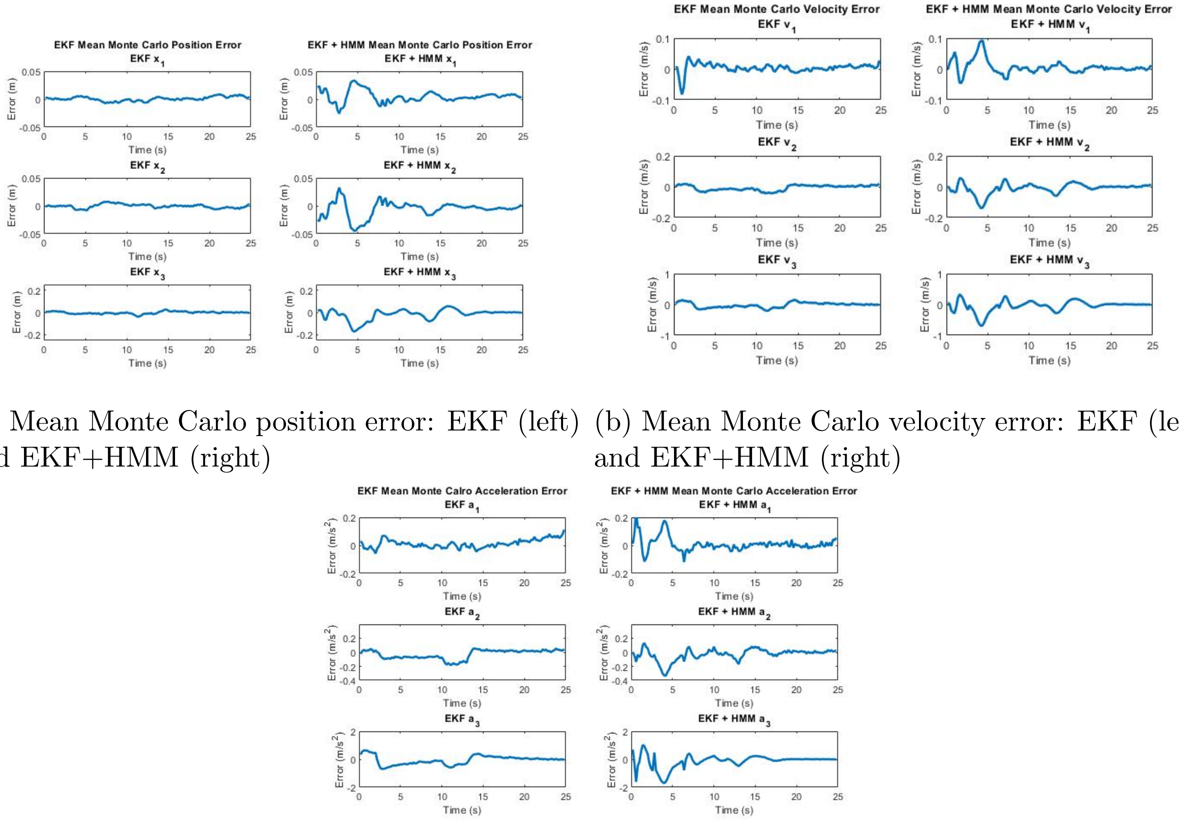
(b) Example run of the worst velocity estimates of Maneuver 4.

(c) Example run of the worst acceleration estimates of Maneuver 4.

Figure 4.40: Example run of EKF+HMM for Maneuver 4.

The EKF and EKF+HMM are compared through their mean Monte Carlo error in Figure 4.41. For all position, velocity, and acceleration estimation, the EKF+HMM performs worse than the EKF. The first 10 seconds of the maneuver involve a (high confidence) misclassification of Maneuver 2, which prompts Maneuver 2 to provide HMM estimates for acceleration and jerk that are not suitable for Maneuver 4 estimation. This causes the majority of EKF+HMM errors at the start of the trajectory.

Still, once the maneuver is correctly identified, the EKF+HMM performs worse. There are consistent errors in the Maneuver 4 HMM estimates of acceleration and jerk that negatively affect EKF+HMM estimation. In the case of other maneuvers, which are more dynamic than Maneuver 4, the HMM estimate errors may still be better than the EKF's constant acceleration model. However, because the EKF is already capable of estimating UAV states throughout Maneuver 4, there is little reason to use the EKF+HMM and risk additional error.



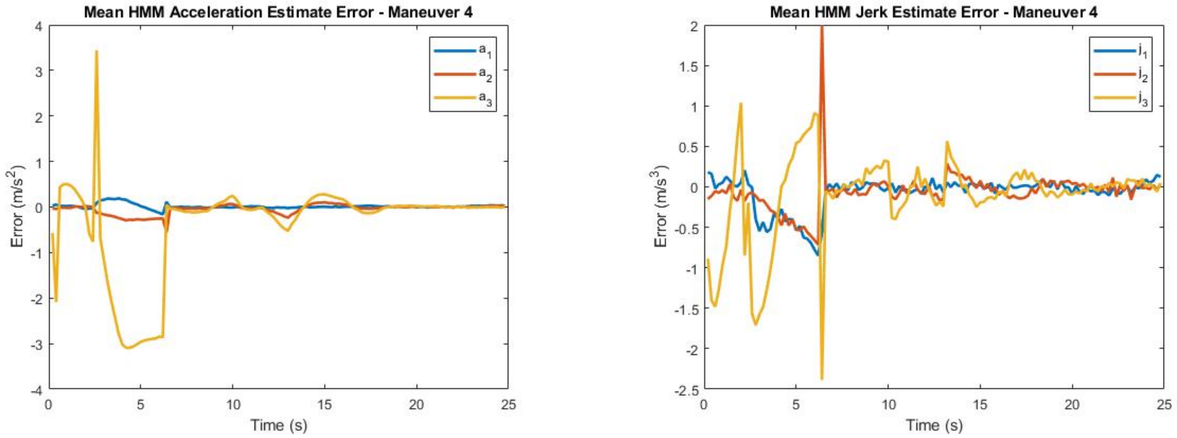
(a) Mean Monte Carlo position error: EKF (left) and EKF+HMM (right)

(b) Mean Monte Carlo velocity error: EKF (left) and EKF+HMM (right)

(c) Mean Monte Carlo acceleration error: EKF (left) and EKF+HMM (right)

Figure 4.41: Comparison of Monte Carlo errors for EKF and EKF+HMM.

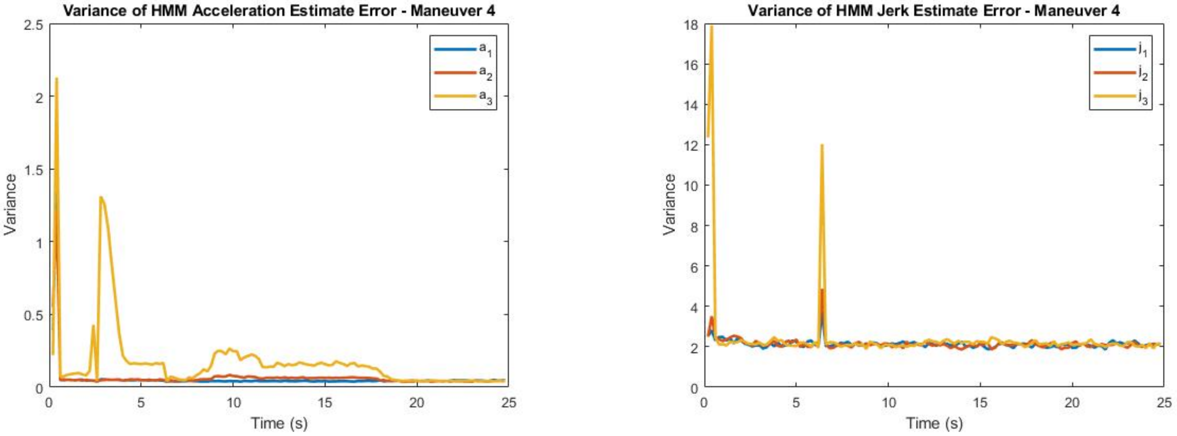
The mean and variance of the HMM estimation errors are shown in Figures 4.42 and 4.43. The high error and error variance in the initial stages of flight correspond to the incorrect classification and use of the incorrect HMM to generate estimates. These errors were not always incorporated into the EKF+HMM estimation. If confidence was greater than -1, then a standard EKF was used instead. After about 7 seconds, the error and error variance decrease greatly.



(a) Mean error of HMM acceleration estimate for the worst run.

(b) Mean error of HMM jerk estimate for the worst run..

Figure 4.42: Mean error for HMM estimates.

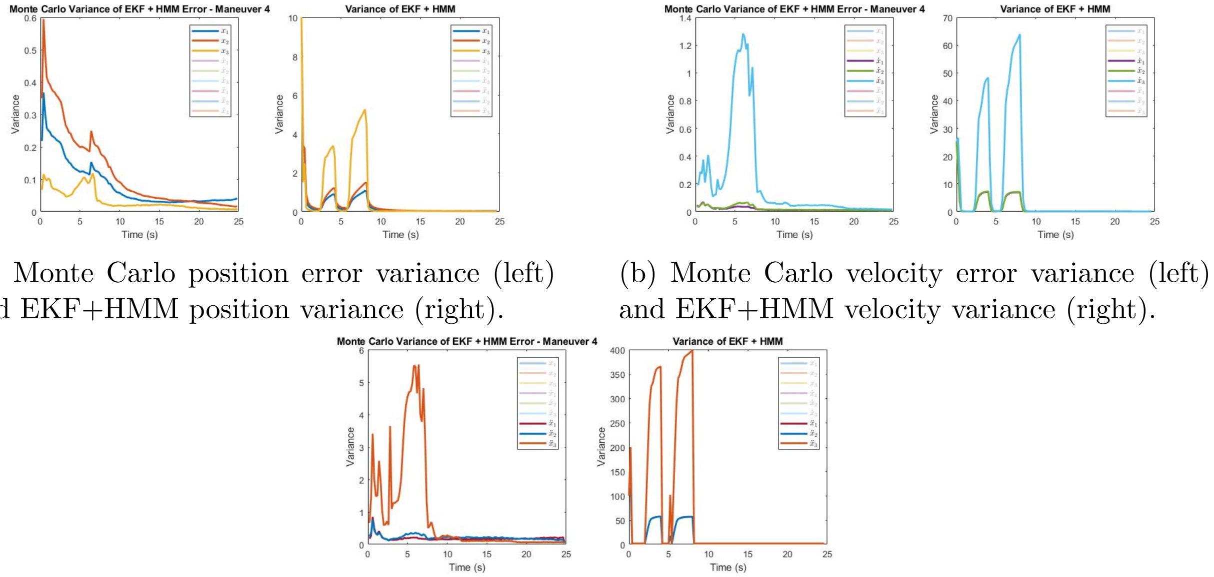


(a) Variance of HMM acceleration estimate error for the worst run.

(b) Variance HMM jerk estimate error for the worst run..

Figure 4.43: Error variance for HMM estimates.

A comparison between EKF+HMM state Monte Carlo error variance and the EKF+HMM state variance is shown in Figure 4.44. The high error variance during the initial 10 seconds of flight is reflected in the EKF+HMM state variance. This high EKF+HMM state variance is the result of switching between using HMM estimates when the confidence of the maneuver's HMM is in flux. After 10 seconds, both the EKF+HMM Monte Carlo error variance and the state variance settle to lower values.



(a) Monte Carlo position error variance (left) and EKF+HMM position variance (right).

(b) Monte Carlo velocity error variance (left) and EKF+HMM velocity variance (right).

(c) Monte Carlo acceleration error variance (left) and EKF+HMM acceleration variance (right).

Figure 4.44: Variance of errors versus EKF+HMM variance.

4.3 Conclusion

This chapter summarized the formulation of the EKF+HMM and showed results of the EKF+HMM when estimating simulated UAV maneuvers. The EKF+HMM takes advantage of a maneuver's HMM and the information it can provide. If a classification with high confidence can be made, the HMM for that maneuver is selected to supplement estimation efforts. The HMM generates estimates of acceleration and jerk based on EKF estimated position. The HMM estimates are used as inputs into the EKF+HMM dynamic model.

For each maneuver, the sum of the absolute value of the estimation error was found for each trajectory in the test data set. From this information, a "best case" (lowest sum of absolute error) and "worst case" trajectory (highest sum of absolute error) was found. A Monte

Carlo simulation was performed with each case and was used to compare the EKF+HMM to a standard EKF. Overall, it was found that the EKF+HMM aids in estimation of high dynamic maneuvers (like Maneuver 1, Maneuver 2, and Maneuver 3). Even in the "worst case" scenarios, the EKF+HMM performed at the same level as or better than the EKF.

The EKF+HMM is not perfect. Biased or incorrect HMM estimates can create consistent error in the state estimates, so true zero-mean error is rarely reached. But in situations where the EKF struggles to perform, the EKF+HMM has much lower error overall. However, in the case of less dynamic maneuvers (like Maneuver 4), where the EKF already is capable of good performance, the EKF+HMM was detrimental. Errors introduced from HMM estimates were much more consequential for this maneuver and created a worse performance of the EKF+HMM when compared to the EKF.

The EKF+HMM can be a valuable tool for increasing estimation accuracy when maneuvers are highly dynamic and the EKF suffers in performance. However, its addition in estimating a maneuver the EKF does not struggle with is more harmful than helpful.

Chapter 5

Experimental Results

The capabilities of HMM classification and the EKF+HMM system were also evaluated using real-world data. A UAV was used to perform several maneuvers, and an HMM was created for each maneuver - using either experimental data or simulated data as the training set. The maneuvers were classified using each type of HMM and their performance was compared. Finally, the estimation results of the EKF and EKF+HMM were compared when using real-world data. The performance of the EKF+HMM was compared when using HMMs trained on simulated data and HMMs trained on experimental data.

The following chapter describes the experimental set up, data processing, and maneuvers performed by the UAV, as well as how those manevuers were simulated. The data sets used to train the HMMs are described. Classification performance and EKF+HMM performance is evaluated.

5.1 UAV Maneuvers

UAV maneuvers were both simulated and performed experimentally. Both of these data sets were used as training data for HMMs that are later used to classify the experimental data. The following section describes the data collection process, as well as maneuver simulation. The maneuvers are shown, and the experimental and simulated data compared.

5.1.1 Experimental Set Up and Data Collection

A Tarot X8 Octacopter, shown in Figure 5.1, was used to execute four maneuvers. Although the Tarot X8 is a different type of UAV than used previously in the thesis (octacopter

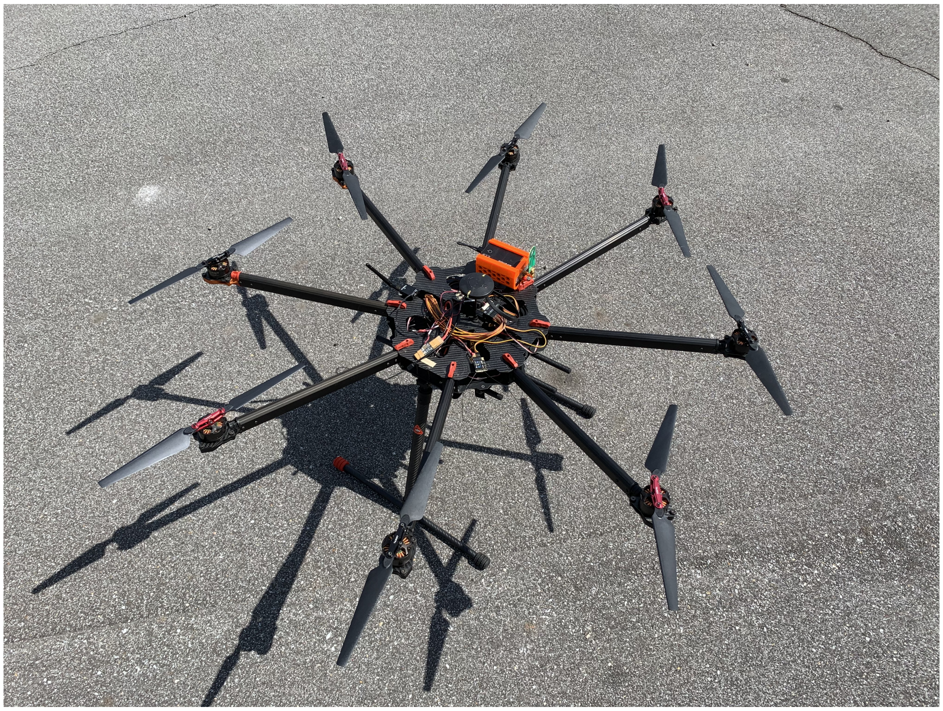


Figure 5.1: The Tarot X8 Octacopter used to perform maneuvers.

versus quadrotor), both the EKF and HMMs are general in their formulation and can easily change applications.

The octacopter used a Hex Cube flight computer and was equipped with a HEX Technology Here+ Rover for GNSS positioning. Throughout each maneuver, local GPS position and velocity were recorded at a 10 Hz rate. When processing the data, the measurements were interpolated to have a 5 Hz update rate consistent with the simulated maneuvers in previous chapters. UAV acceleration throughout a maneuver was found by numerically differentiating the GPS velocity. These measurements are used as the reference states of the UAV.

The maneuvers executed by the UAV are shown in Figure 5.2. During data collection, each maneuver was executed fifteen times, except for Maneuver 2, which was performed 11 times. As a result, Maneuver 2 was used as an "unknown" trajectory to test the classification system. Note that these maneuvers are not the same as those used in previous chapters. The previous simulated maneuvers were more dynamic and would not be safe for the octacopter.

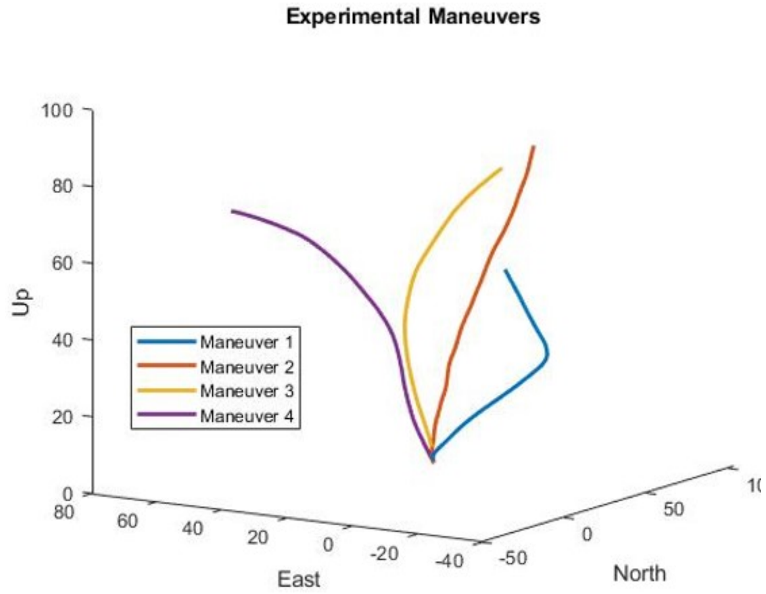


Figure 5.2: Maneuvers performed by the octacopter.

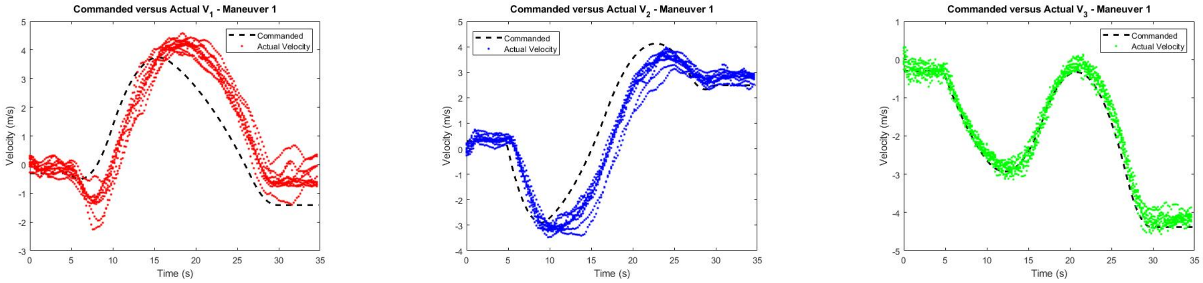
5.1.2 UAV Control

Text files of commanded velocities were created in the local ENU frame for each maneuver and converted to NED in post-process. The UAV was remote controlled into a start position, then the UAV mode was switched to follow text file velocity commands. Each maneuver began with five seconds of constant velocity commands and ended with another five seconds of constant velocity commands. Once in command velocity mode, the update rate of the UAV was 10 Hz.

The velocity control of the UAV was open loop and therefor unable to correct errors. At times, this resulted in high error or offset (or both) between the actual and commanded velocity. Initially, there was an attempt to characterize these errors. Figures 5.3 and 5.4 show the difference between commanded and actual NED velocity for Maneuvers 1 and 3 respectively. These two maneuvers were chosen because they illustrate the difficulty in consistently characterizing error between commanded and actual velocity.

The maneuvers were performed by the UAV at different orientations, different times of day, and sometimes on different days altogether. Because of this, environmental effects changed, which changed the UAV's response to commanded velocity as well. Figure 5.3 shows that both v_1 and v_2 are slightly delayed, and that v_1 has a higher magnitude than the commanded velocity. However, Figure 5.4 shows that v_1 has high error though relatively low delay, while v_2 and v_3 are capable of following commanded velocity well. The only consistency between the two data sets is seen with v_3.

These results suggest disturbances from the environment (wind) greatly affected UAV performance when following velocity commands. A strategy for future experimentation could be to only collect data on still days, which would result in the UAV more consistently and accurately following the commanded velocities. However, this would not allow the opportunity to properly access how HMM classification and EKF+HMM estimation responds to environmental disturbances.

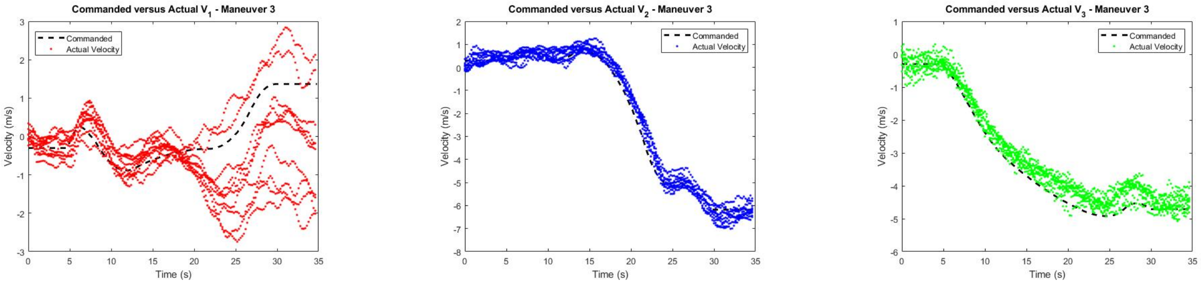


(a) Commanded versus actual North velocity of the UAV during Maneuver 1.

(b) Commanded versus actual East velocity of the UAV during Maneuver 1.

(c) Commanded versus actual Down velocity of the UAV during Maneuver 1.

Figure 5.3: Comparison of commanded versus actual velocity of the UAV during Maneuver 1.



(a) Commanded versus actual North velocity of the UAV during Maneuver 3.

(b) Commanded versus actual East velocity of the UAV during Maneuver 3.

(c) Commanded versus actual Down velocity of the UAV during Maneuver 3.

Figure 5.4: Comparison of commanded versus actual velocity of the UAV during Maneuver 3.

5.1.3 Simulation of UAV Maneuvers

The UAV maneuvers were simulated in MATLAB. Any specific attributes of the UAV or the environment (such as command velocity delay time) were not included in the simulation. This mimics a situation in which, while certain maneuvers may be previously known, the characteristics of the specific UAV and its environment are not.

Simulated data was created by differentiating the command velocity of the UAV to obtain a command acceleration. Zero mean white noise was added to the command acceleration before integrating to obtain velocity and position. To create training data, the North, East, and Down start and end positions of each data run was also varied. Heading was considered inconsequential as the data pre-processing procedure in Chapter 3 creates a position invariant data set. In total, 405 simulated trajectories were created for each maneuver.

5.1.4 Simulated and Experimental Maneuvers

An example of each of the four maneuvers is shown in the following sections using both simulated and experimental data. Each maneuver is 35 seconds total, with 5 seconds of constant commanded velocity at the beginning and end of the trajectory. For each maneuver, the UAV is ascending from an initial height of approximately 10 m. This method of performing maneuvers (beginning at a low altitude and ascending towards some target point) was the simplest way to carry out maneuvers in an experimental setting.

Maneuver 1

An example of the UAV executing Maneuver 1 is shown in Figure 5.5. The blue line shows simulated UAV position while the red shows an experimental run of the maneuver. Both begin at the same initial position. The simulated position, velocity, and acceleration of the UAV in the NED frame are displayed in Figure 5.6, while the experimental position, velocity, and acceleration are shown in Figure 5.7. The experimental data has greater noise in acceleration and velocity. Moreover, note the octacopter is unable to produce the quick change in a_2 near 10 seconds – unlike the simulated UAV. This creates a slightly wider turn for the experimental maneuver that can be seen in Figure 5.5.

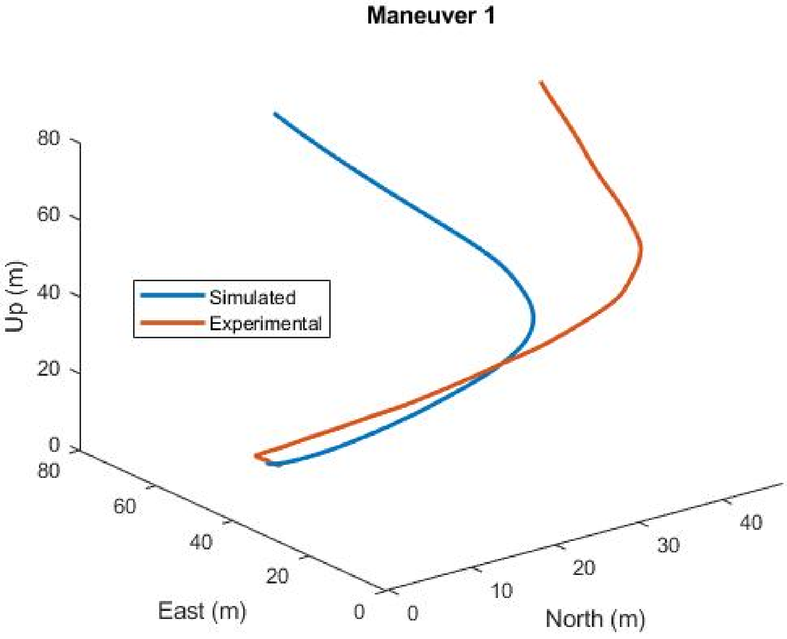
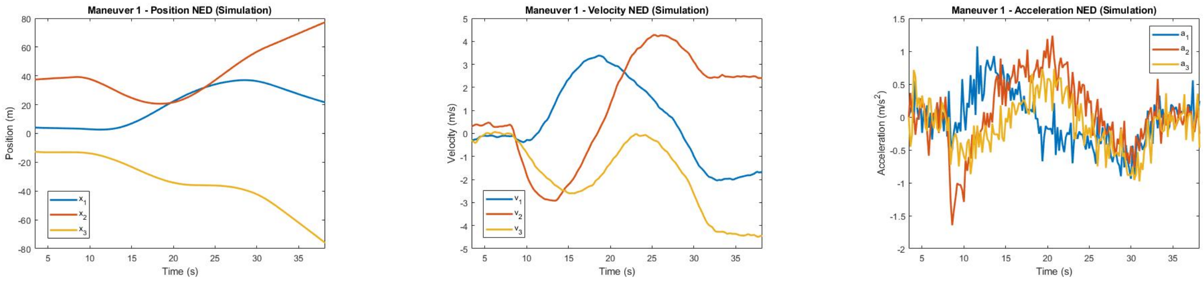


Figure 5.5: Simulated versus experimental position of UAV throughout Maneuver 1.

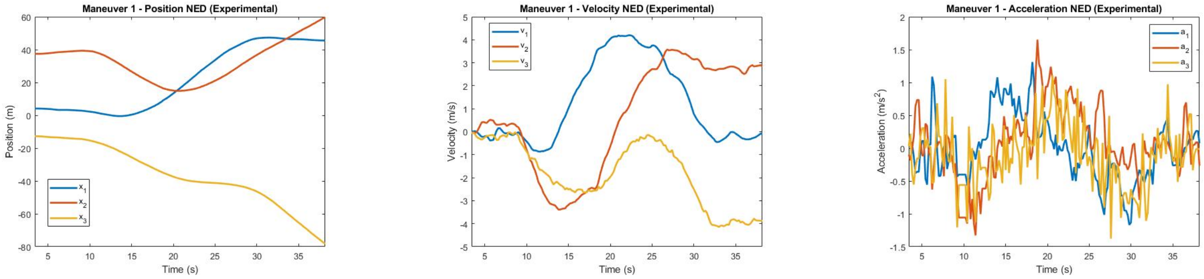


(a) Maneuver 1 simulated position over time (NED).

(b) Maneuver 1 simulated velocity over time (NED).

(c) Maneuver 1 simulated acceleration over time (NED).

Figure 5.6: The nine states of the UAV simulated throughout Maneuver 1.



(a) Maneuver 1 experimental position over time (NED).

(b) Maneuver 1 experimental velocity over time (NED).

(c) Maneuver 1 experimental acceleration over time (NED).

Figure 5.7: The nine states of the UAV throughout Maneuver 1.

Maneuver 2

Figure 5.8 illustrates simulated and experimental UAV flight for Maneuver 2. Once again, both runs begin from the same position, but the two paths have slightly different heading values. The simulated and experimental UAV position, velocity, and acceleration are shown in Figures 5.9 and 5.10. The greatest difference in the simulated and experimental data set is seen again with a_2. Unlike the simulated acceleration, the experimental a_2 changes little over time, which results in a great difference between simulated and experimental v_2 and x_2.

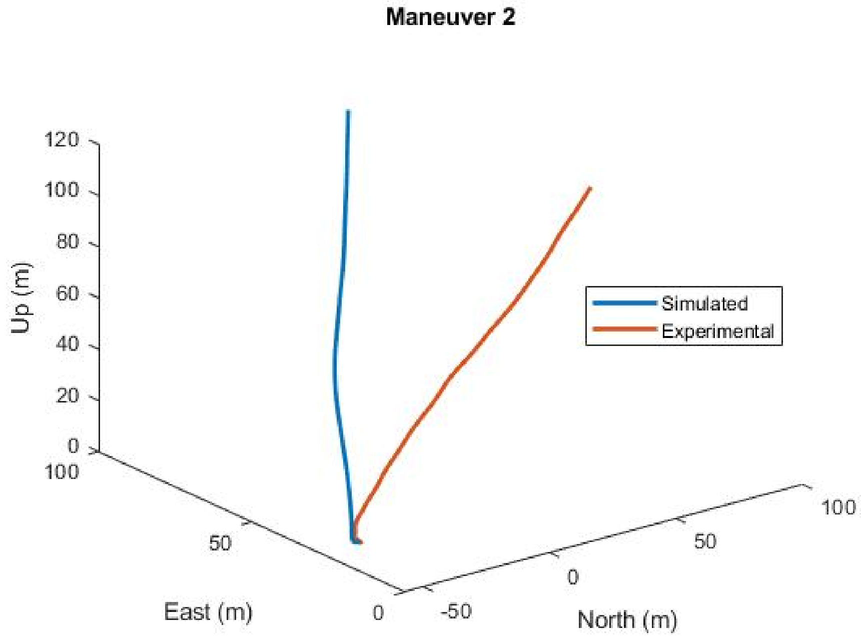
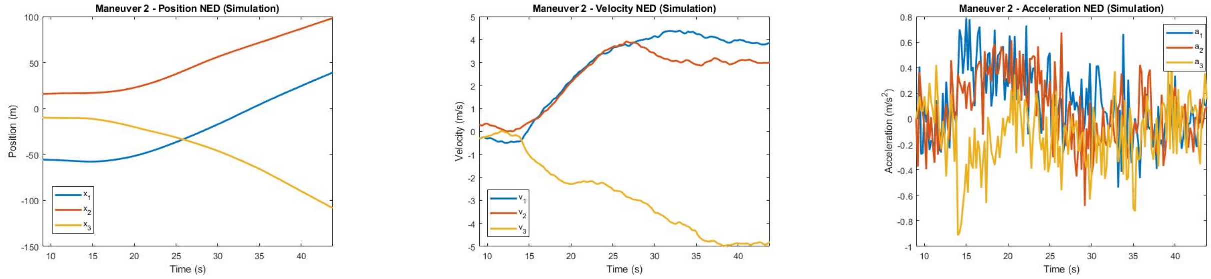


Figure 5.8: Simulated versus experimental position of UAV throughout Maneuver 2.

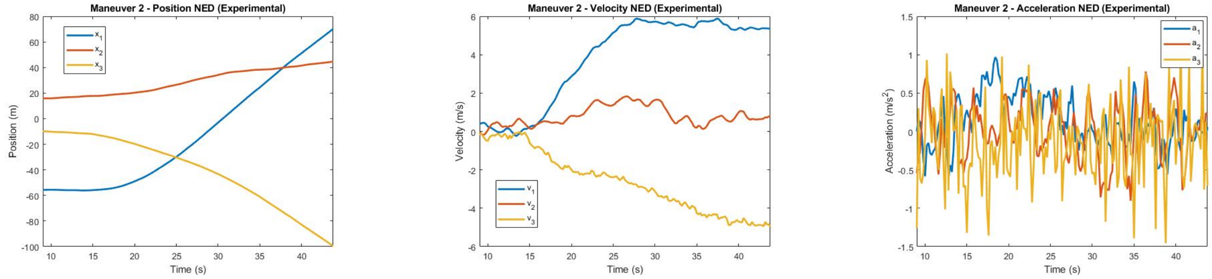


(a) Maneuver 2 simulated position over time (NED).

(b) Maneuver 2 simulated velocity over time (NED).

(c) Maneuver 2 simulated acceleration over time (NED).

Figure 5.9: The nine states of the UAV simulated throughout Maneuver 2.



(a) Maneuver 2 experimental position over time (NED).

(b) Maneuver 2 experimental velocity over time (NED).

(c) Maneuver 2 experimental acceleration over time (NED).

Figure 5.10: The nine states of the UAV throughout Maneuver 2.

Maneuver 3

The UAV path created from Maneuver 3 is displayed in Figure 5.11, which shows both the simulated and experimental UAV path. The simulated UAV states in the NED frame are shown in Figure 5.12, while the experimental UAV states are shown in Figure 5.13. Overall, the simulated and experimental states are similar, although experimental a_3 experiences much higher noise than simulated a_3.

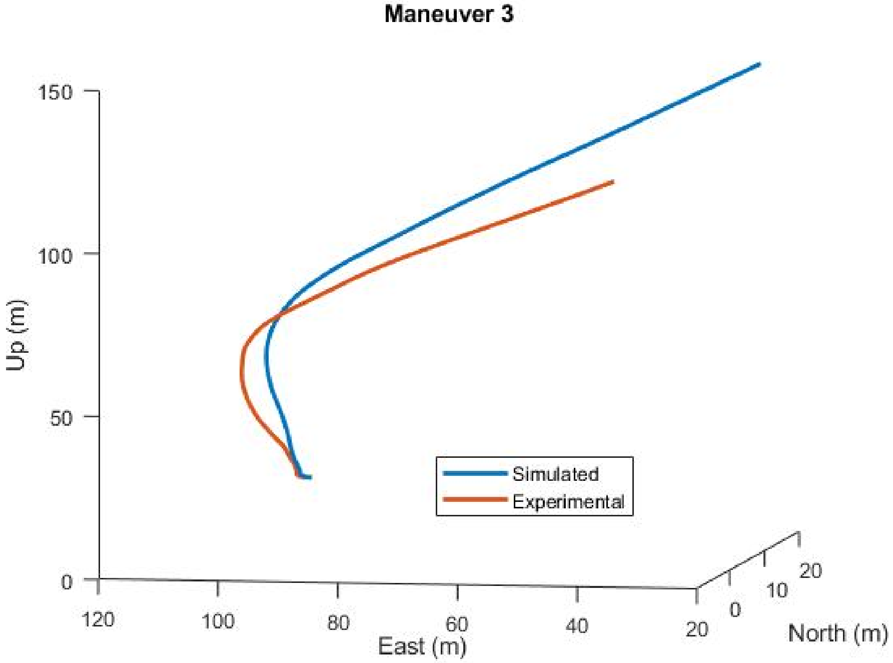
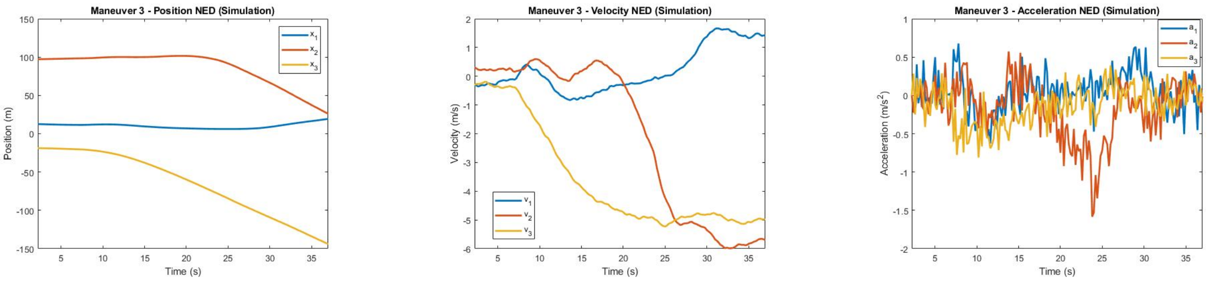


Figure 5.11: Simulated versus experimental position of UAV throughout Maneuver 3.

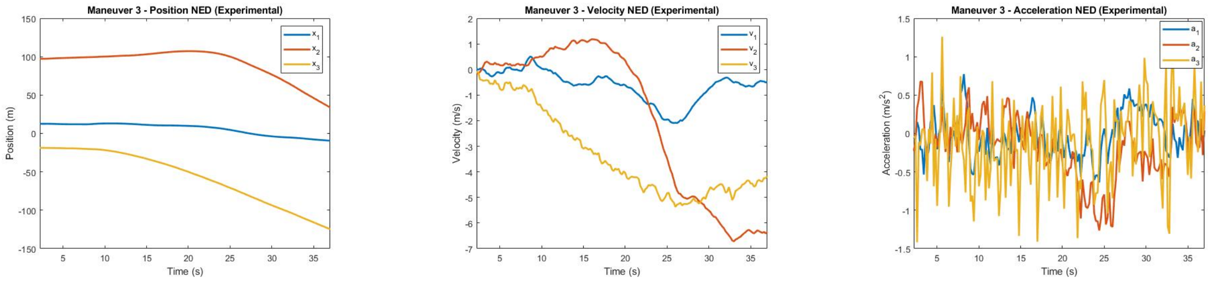


(a) Maneuver 3 simulated position over time (NED).

(b) Maneuver 3 simulated velocity over time (NED).

(c) Maneuver 3 simulated acceleration over time (NED).

Figure 5.12: The nine states of the UAV simulated throughout Maneuver 3.



(a) Maneuver 3 experimental position over time (NED).

(b) Maneuver 3 experimental velocity over time (NED).

(c) Maneuver 3 experimental acceleration over time (NED).

Figure 5.13: The nine states of the UAV throughout Maneuver 3.

Maneuver 4

The final maneuver carried out by the octacopter is Maneuver 4. Figure 5.14 shows the position of the UAV during the maneuver – both simulated and experimentally. The simulated and experimental states of the UAV in the NED frame can be viewed in Figures 5.15 and 5.16. The greatest difference between simulated and experimental data can be seen in a_1, v_1, and x_1. Initial commanded v_1 is relatively low, which the simulation is able to capture, but is not executed well by the UAV experimentally. This leads to a difference in v_1 between the two data sets that is so great it initially looks as if the velocity has been inverted.

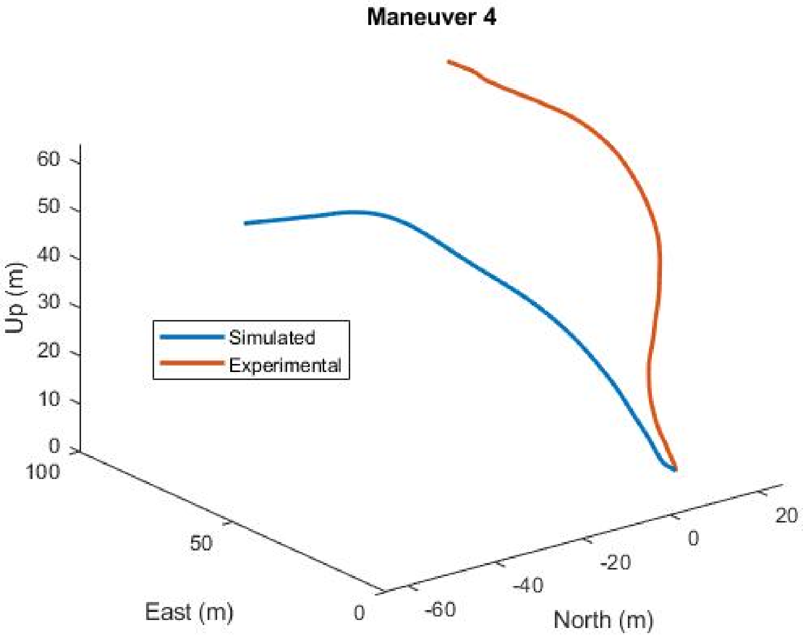
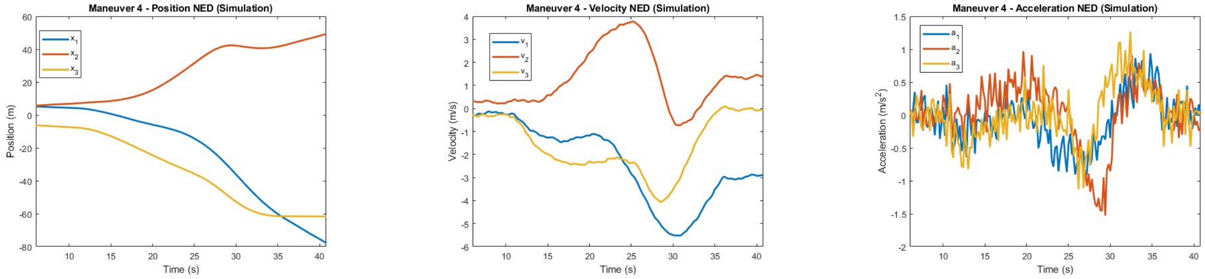


Figure 5.14: Simulated versus experimental position of Maneuver 4.

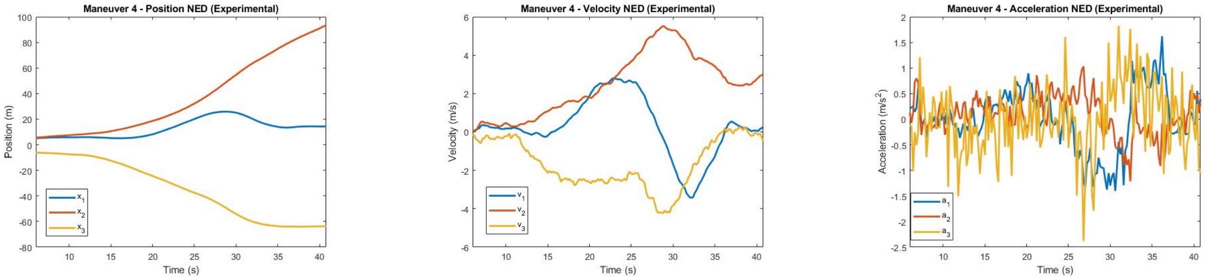


(a) Maneuver 4 simulated position over time (NED).

(b) Maneuver 4 simulated velocity over time (NED).

(c) Maneuver 4 simulated acceleration over time (NED).

Figure 5.15: The nine states of the UAV simulated throughout Maneuver 4.



(a) Maneuver 4 experimental position over time (NED).

(b) Maneuver 4 experimental velocity over time (NED).

(c) Maneuver 4 experimental acceleration over time (NED).

Figure 5.16: The nine states of the UAV throughout Maneuver 4.

5.2 Maneuver Classification

The purpose of this section is to evaluate the classification accuracy of two different types of HMM. The first set of HMMs were trained on experimental data (also known as reference data). Ten data runs of each maneuver were randomly selected and used as the training data set, while the remaining five runs were used as test data. This is similar to a scenario in which there is a sparse data set from real world data collection and a lack of knowledge of the underlying commands of the UAV.

The second set of HMMs were trained on simulated data and tested on experimental data. All simulated data was used as the training data, while all experimental data was used as the test data. This mimics a scenario in which information about a maneuver may be known, but experimental data of it has never been collected.

For both types of HMMs, Maneuvers 1, 3, and 4 were selected to be the known maneuvers, while Maneuver 2 was used as the unknown trajectory. The following sections first discuss the classification results of HMMs trained with reference (experimental) data and then the classification results of HMMs trained with simulated data.

The classification accuracy of both reference data and estimated reference data are shown for both sets of HMMs. Accuracy is characterized in two ways: final accuracy, whether the maneuver was classified correctly at the end of its flight, and mean accuracy over time, which shows what percentage of the trajectory the maneuver was correctly classified. For example, if the maneuver was correctly classified for 25 out of 30 total seconds, the accuracy over time would be 83.33%. Accuracy over time is found for each trajectory in the data set and then the average is taken to produce mean accuracy over time.

Technically each maneuver is 35 seconds in total. However, it should be noted that for the first five seconds of flight, each maneuver has the same commanded velocity and is identical. This makes classification impossible during these moments. Therefore, these five seconds are not included in the mean accuracy over time calculation, but are shown in the figures.

More information about the training process of HMMs can be found in Section 3.3.1. Maneuver classification with HMMs is discussed in detail in Section 3.4. The classification scheme used to classify maneuvers is the confidence measure classifier, which is discussed in Section 3.4.4.

5.2.1 Classification Accuracy of HMMs Trained on Experimental Data

The following section shows classification results when the confidence measure classifier uses HMMs trained with experimental data.

Reference Training Data

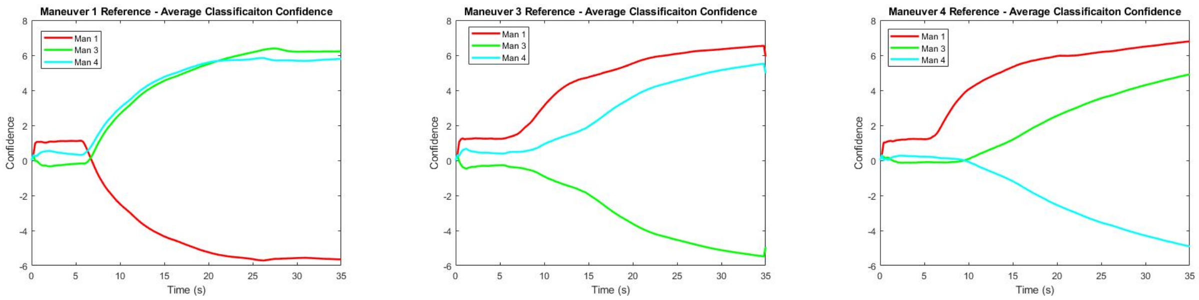
Classification accuracy was first tested on the training data; the results of this are shown in Table 5.1. Each maneuver had a 100% final accuracy, which is to be expected with a training data set. The mean accuracy over time is much lower for experimental training data when compared to classification of simulated training data in Section 3.4.2. However, all maneuvers are correctly identified for more than half of their flight time. Maneuver 1 has the greatest mean accuracy over time, while Maneuver 4 has the lowest.

Table 5.1: HMM Classification: Training Reference

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	91.27%
3	100%	82.86%
4	100%	68.67%

Figure 5.17 the average confidence of each maneuver during classification. As noted above, the first five seconds of each maneuver is identical. The confidence in classification for each model remains low and constant for the first five seconds of each maneuver – demonstrating how classification is impossible during this time. Then, the maneuvers begin to distinguish themselves. Note that Maneuver 1 almost immediately distinguishes itself,

while Maneuvers 3 and 4 require more observations over time for a correct and confident classification.



(a) Classification of Maneuver 1 reference training data.

(b) Classification of Maneuver 3 reference training data.

(c) Classification of Maneuver 4 reference training data.

Figure 5.17: Average confidence over time for each HMM during the classification process - *reference training data.*

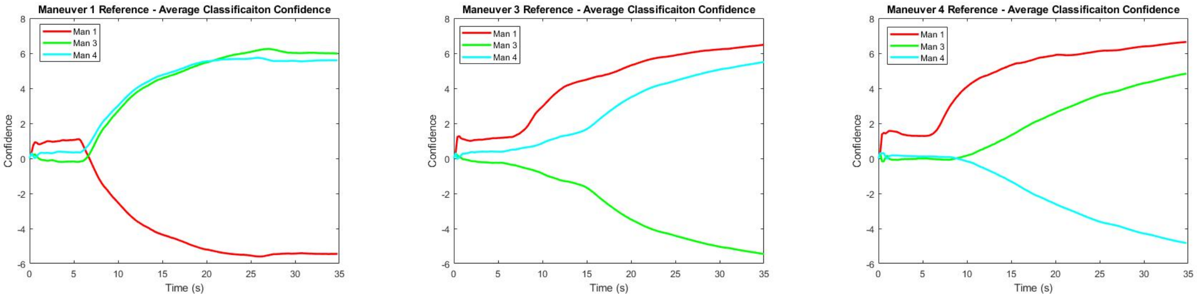
Reference Test Data

The classifier then tested by classifying the test data set of maneuvers. The results of this test is shown in Table 5.2. This table also contains results for classifying an unknown maneuver, Maneuver 2. In this case, a correct classification for Maneuver 2 means the classifier identifies it as unknown. Maneuver 2 is classified as unknown 90% of the time, but is occasionally mistaken for a known trajectory (Maneuver 4).

Table 5.2: HMM Classification: Test Reference

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	91.47%
3	100%	81.47%
4	100%	69.47%
Unknown (2)	90.00%	80.14%

Figure 5.18 displays the average confidence of each model during the classification of Maneuver 1, Maneuver 3, and Maneuver 4 for the reference test data set. Like before, Maneuver 1 is quickly classified correctly, while Maneuvers 3 and 4 require more time and observations to drop below a confidence measure of -1.



(a) Classification of Maneuver 1 reference testing data.

(b) Classification of Maneuver 3 reference testing data.

(c) Classification of Maneuver 4 reference testing data.

Figure 5.18: Average confidence over time for each HMM during the classification process - *reference test data*.

The average confidence of each maneuver HMM when receiving observations from Maneuver 2 (unknown) is displayed in Figure 5.19. The confidence of each maneuver's classification remains low throughout the flight. However, Maneuver 4 does show a confidence score that occasionally results in a misclassification.

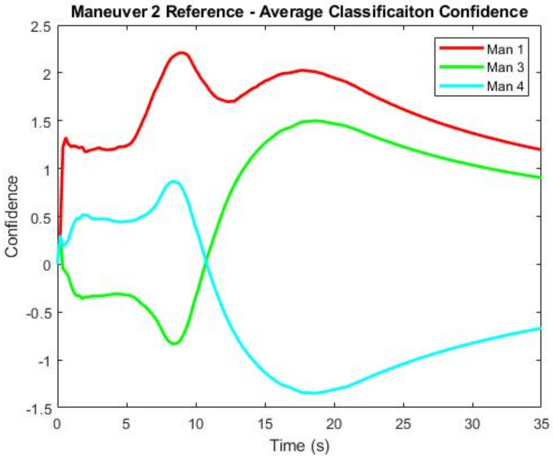


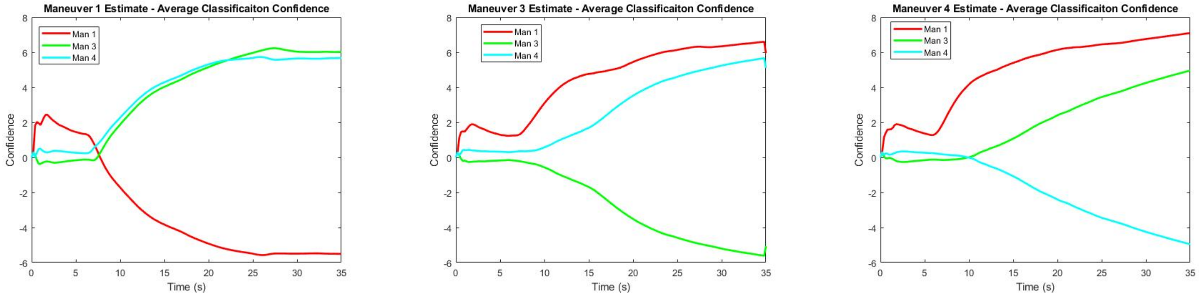
Figure 5.19: Average confidence of each HMM when given an unknown maneuver - *reference data*.

Estimated Training Data

The classifier was also tested using the EKF's estimates of reference data as observations. This section will cover the classification results when the HMM classifier was tested using estimates of the original training data and test data. Overall, the final accuracy of the classifier remained at 100% for each maneuver when given estimated data as observations, as seen in Table 5.3. The accuracy over time was slightly lower than when using reference data, but overall the decrease was nearly negligible. The average confidence of each model is shown in Figure 5.20 respectively. The confidence measure remains similar to that of reference training data.

Table 5.3: HMM Classification: Training EKF

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	87.47%
3	100%	77.67%
4	100%	66.27%



(a) Classification of Maneuver 1 estimated training data.

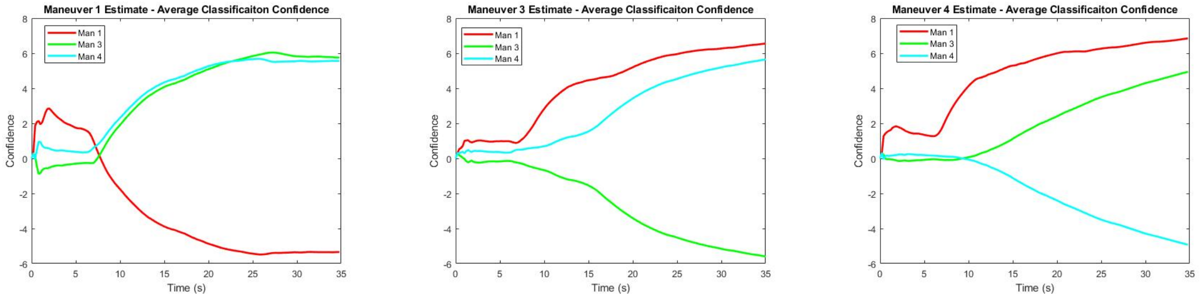
(b) Classification of Maneuver 3 estimated training data.

(c) Classification of Maneuver 4 estimated training data.

Figure 5.20: Average confidence over time for each HMM during the classification process - *estimated training data*.

Estimated Test Data

Estimates of test data were also used to test classification accuracy. Once again, the final accuracy of the known trajectories was 100%. The the unknown trajectory was only given the correct “unknown” classification 80% of the time, which is a lower accuracy than when using reference data as observations. The accuracy over time of each maneuver was similar to the results when using reference test data. These results can be viewed in Table 5.4. The average confidence of each model during classification of known trajectories is shown in Figure 5.21.



(a) Classification of Maneuver 1 estimated testing data.

(b) Classification of Maneuver 3 estimated testing data.

(c) Classification of Maneuver 4 estimated testing data.

Figure 5.21: Average confidence over time for each HMM during the classification process - *estimated test data*.

Maneuver 2, the unknown maneuver, was also estimated using the EKF and used to test the classification scheme's ability to classify a maneuver as unknown. The average confidence of each model when presented with estimated UAV states of Maneuver 2 is shown in Figure 5.22. The confidence of each model remains low, but Maneuver 4's model has a confidence score that reaches below -1, meaning there were some misclassifications. This is reflected in the lower final accuracy and mean accuracy over time of the unknown maneuver in Table 5.4.

Table 5.4: HMM Classification: Test EKF

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	87.61%
3	100%	78.00%
4	100%	66.54%
Unknown (2)	80.00%	72.87%

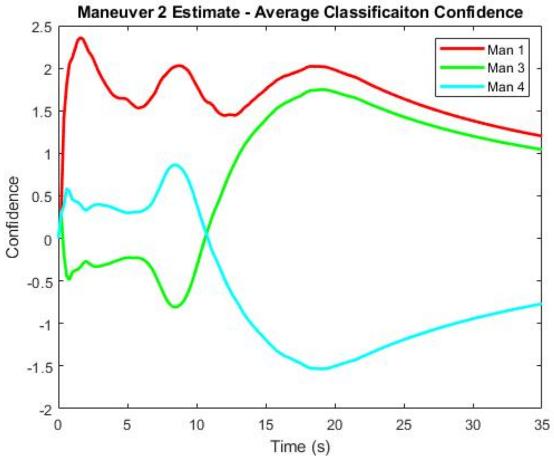


Figure 5.22: Average confidence over time of each HMM when given an unknown maneuver - *estimated data*.

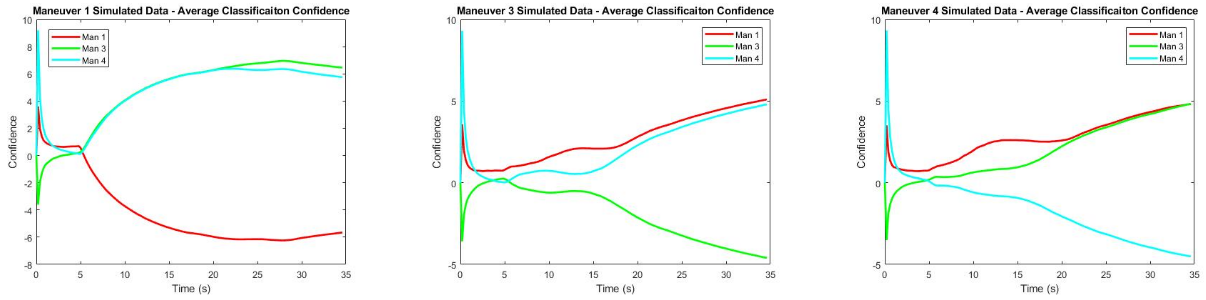
5.2.2 Classification Accuracy of HMMs Trained on Simulated Data

The following section shows the classification results when using simulation trained HMMs in the confidence measure classifier rather than HMMs trained with experimental reference data.

Simulated Training Data

The classification accuracy of HMMs trained on simulated data was first evaluated using the simulated training data set to understand the baseline accuracy. Table 5.5 shows the classification accuracy. Each maneuver had a final classification accuracy 100%, including Maneuver 2 which was successfully classified as unknown 100% of the time. Maneuver 1 had the highest mean accuracy over time of the known trajectories. Both Maneuvers 3 and 4 were correctly classified for a little over 60% of their trajectory.

The average confidence of each model over time is shown in Figure 5.23. In Figure 5.23a, it can be noted that Maneuver 1 is correctly classified almost immediately after the first 5 seconds of similar flight. Figures 5.23b and 5.23c show that both Maneuvers 3 and 4 remain "unknown" for a period at the beginning of their trajectory, which is similar to results when using HMMs trained with experimental data. There is similarity in these two maneuvers' acceleration for the first fifteen seconds of flight (as seen in Figures 5.12c and 5.15c), which may contribute to their initial low confidences.



(a) Classification of Maneuver 1 simulated training data.

(b) Classification of Maneuver 3 simulated training data.

(c) Classification of Maneuver 4 simulated training data.

Figure 5.23: Average confidence over time for each model - *simulated training data.*

The average confidence of each model when shown an unknown maneuver (Maneuver 2) is displayed in Figure 5.24. Almost immediately, each model has low confidence. After five seconds, there is a change in each model's confidence as Maneuver 2 transitions from constant commanded velocity to the beginning of the maneuver. However, all the models quickly settle to a confidence measure near zero soon after.

Table 5.5: Simulation Trained HMM Classification: Training Data

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	95.80%
3	100%	63.46%
4	100%	67.54%
Unknown (2)	100%	100%

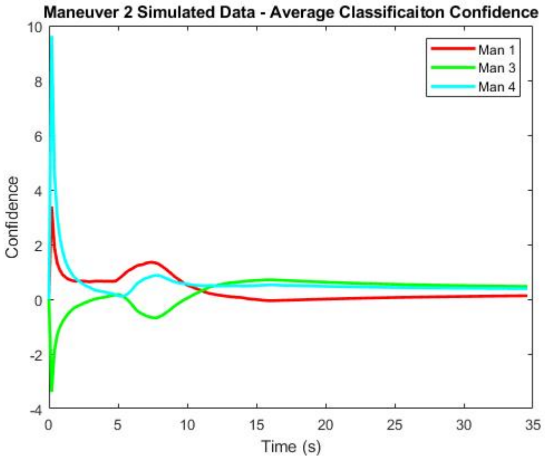


Figure 5.24: Average confidence of each HMM when given an unknown maneuver - *simulated training data*

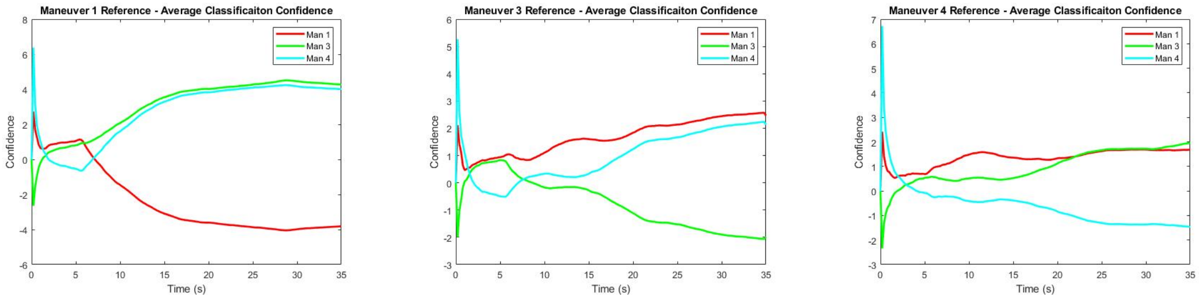
Experimental Test Data

After being tested on the original training set of simulated data, the classifier was tested on real world data. It was tested first on the reference data and then on the estimated reference data. Table 5.6 shows the accuracy of the simulation trained HMM classifier on the experimental reference data of all the maneuvers. Maneuvers 1 and 3 have a 100% final classification accuracy, and Maneuver 2 is correctly identified as unknown with 100% accuracy (both final accuracy and mean accuracy over time). However, Maneuver 4 has a final accuracy of only 73.33%. Most maneuvers are correctly classified for 50% or more of their flight time – except Maneuver 4.

The simulation trained HMM classifier had worse performance in final accuracy for Maneuver 4 and mean accuracy over time for all known maneuvers when compared to the experimental data trained HMM classifier. However, the simulation trained HMM does improve the classification accuracy of Maneuver 2 – both for final accuracy and mean accuracy over time.

Figure 5.25 shows the average classification confidence of each model for each known maneuver. Maneuver 1's model once again almost immediately has a confidence measure of less than -1, indicating high confidence in its classification. However, the model of Maneuver

3 and 4 require more time to become confident in their own classification. Maneuver 4's model, at times, lingers in the "unknown" confidence level. For some Maneuver 4 trajectories, the confidence of Maneuver 4's model was so low at the end of the trajectory, it resulted in an "unknown" classification (see Appendix E.2).



(a) Classification of Maneuver 1 reference data.

(b) Classification of Maneuver 3 reference data.

(c) Classification of Maneuver 4 reference data.

Figure 5.25: Average confidence over time using HMMs trained with simulated data - *reference data.*

The simulation trained models' average confidence when given reference data of an unknown trajectory (Maneuver 2) is shown in Figure 5.26. After an initial misclassification as Maneuver 3, Maneuver 2 is steadily classified as unknown for all models.

Table 5.6: Simulation Trained HMM Classification: Test Reference

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	86.76%
3	100%	45.22%
4	73.33%	35.73%
Unknown (2)	100%	100%

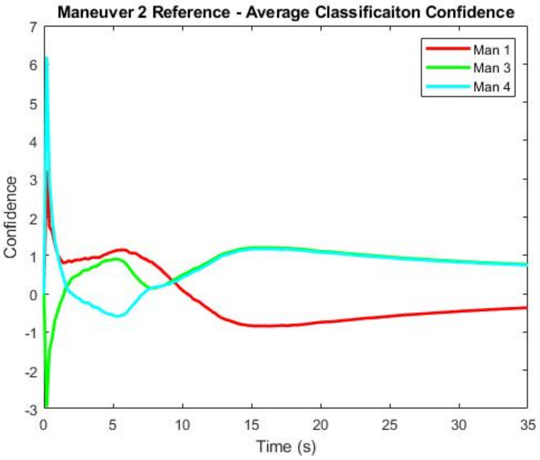


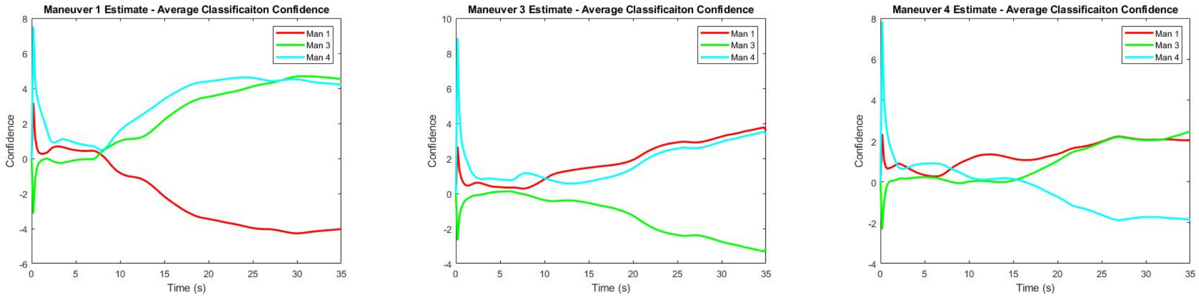
Figure 5.26: Average confidence of each simulation trained HMM when given an unknown maneuver - *reference data*.

The simulation trained models were then used to classify EKF estimates of the reference data. These results are shown in Table 5.7. The final classification accuracy stayed roughly the same as the reference data for Maneuvers 1 and 3, while decreasing for the unknown maneuver. The final accuracy of Maneuver 4 increased, as well as the mean accuracy over time for Maneuver 3 and Maneuver 4. Estimated data has less noise than reference data, which may have lead to the increases in mean accuracy over time.

Table 5.7: HMM Classification: Test Estimate

Maneuver	Final Accuracy	Mean Accuracy Over Time
1	100%	68.80%
3	100%	52.23%
4	80%	40.46%
Unknown (2)	81.82%	41.66%

Figure 5.27 shows the average confidence over time for each simulation trained model when classifying estimates of the reference data. The results are similar to that of Figure 5.25.



(a) Classification of Maneuver 1 estimated data.

(b) Classification of Maneuver 3 estimated data.

(c) Classification of Maneuver 4 estimated data.

Figure 5.27: Average confidence over time using simulation trained HMMs - *estimated reference data.*

Figure 5.28 displays the average classification confidence of each model, as the model reacts to an unknown maneuver. The classification scheme fails at times to correctly identify the maneuver as unknown. Note that the average classification confidence of Maneuver 1 falls below -1, meaning the unknown maneuver is misclassified as Maneuver 1 at times.

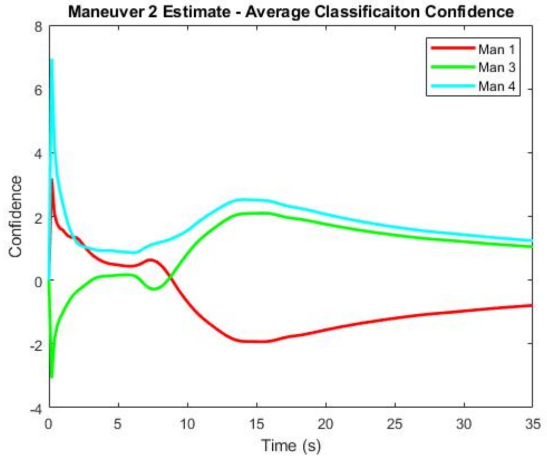


Figure 5.28: Average confidence of each model when classifying an unknown maneuver - *estimated data.*

5.2.3 Classification Accuracy During Maneuver-less Flight

The process by which experimental data was generated means there is data before each maneuver in which the UAV is operated via teleop control rather than through commanded velocities for specific maneuvers. Ideally, the confidence measure classification scheme would be capable of distinguishing these sections of flight as unknown maneuvers. The following two sections show results of simulation trained and experimentally trained HMM classifiers when presented with completely unknown data. Because these random segments of data often varied in length, individual results rather than average results are presented.

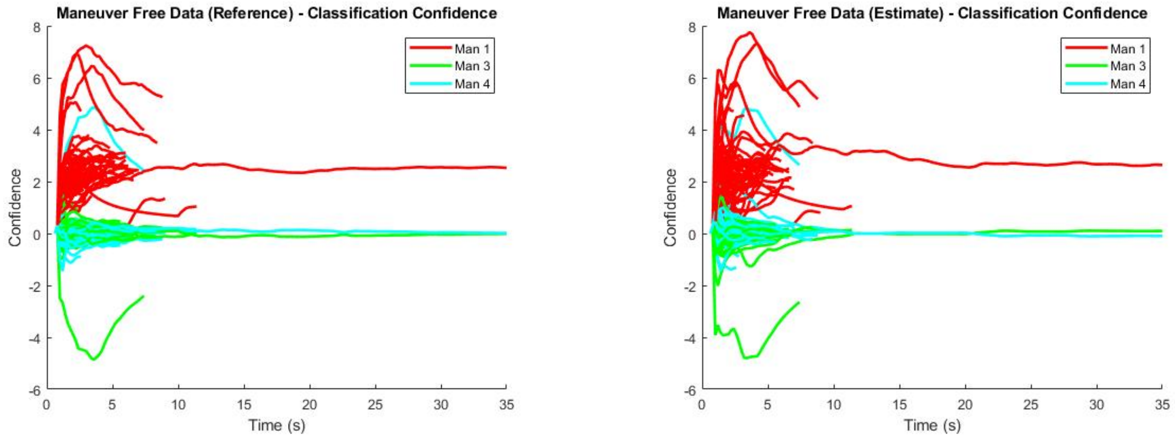
Experimental Data Trained HMM

The classifier using HMMs trained on experimental data was also tested on the random unknown trajectories. Both the reference data and estimated reference data of the unknown trajectories had high final accuracy for correctly classifying the trajectories as unknown. Note that they also both had high mean accuracy over time, meaning the trajectories were classified as unknown in the early portions of flight.

Table 5.8: Classification of unknown experimental data using HMMs trained on real world data.

Data Type	Final Accuracy	Mean Accuracy Over Time
Reference	98.21%	99.49%
Estimate	96.43%	98.40%

The confidence of each model when presented with unknown data is shown below in Figure 5.29. Both the reference and estimated data have a few instances in which the unknown trajectory is misclassified as Maneuver 3. For the majority of data, the models have low confidence (indicating they are unknown) or high confidence of incorrect classification.



(a) Confidence of each model when no maneuvers are performed - *Reference Data*.

(b) Confidence of each model when no maneuvers are performed - *Estimated Data*.

Figure 5.29: Confidence over time for each reference trained HMM when there is no maneuver.

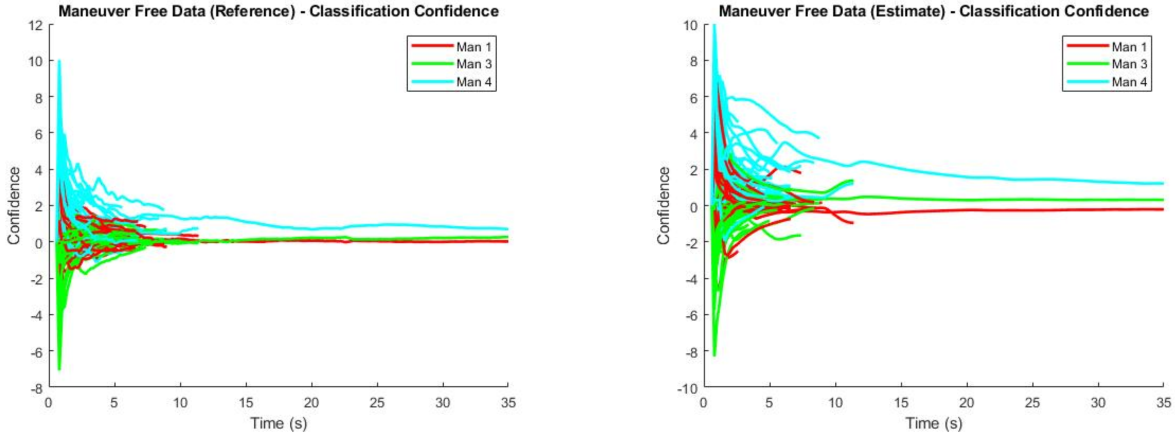
Simulation Data Trained HMM

Table 5.9 shows the classification results when the classifier (using simulation trained HMMs) was given completely unknown data. For both reference and estimated reference data, the classifier had high final accuracy – meaning the trajectories were successfully classified as “unknown”. The mean accuracy over time for reference data remained high, while estimated data did not perform as well.

Table 5.9: Classification of unknown experimental data using simulation trained HMMs.

Data Type	Final Accuracy	Mean Accuracy Over Time
Reference	100%	92.87%
Estimate	92.86%	79.60%

Figure 5.30 shows the confidence of each model as classification occurred on the random segments of data. Maneuver 3 had a tendency, for both simulated and experimental data, to be initially misclassified as the maneuver. Figure 5.30b shows an occasional misclassification of the unknown trajectories as Maneuver 1 or Maneuver 3, but overall the models seemed capable of correctly identifying he trajectory as unknown.



(a) Confidence of each model when no maneuvers are performed - *Reference Data*.

(b) Confidence of each model when no maneuvers are performed - *Estimated Data*.

Figure 5.30: Confidence over time for each simulation trained HMM when there is no maneuver.

5.2.4 Classification with Simulation versus Experimental Data Trained HMMs

Compared to HMMs trained with experimental data, simulation trained HMMs performed worse in classification of real-world data (both reference and estimated) by almost all metrics. The confidence measure classifier using simulation trained HMMs typically had the same or lower final accuracy and mean accuracy over time. The only instance in which the simulation trained HMMs performed better was when classifying Maneuver 2 as unknown.

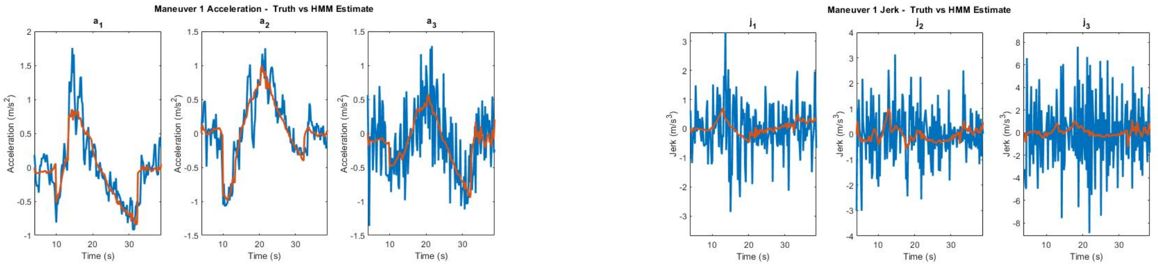
While generally worse than the experimental data trained HMMs, simulation data trained HMMs were still adequate at classification of real world data. These results show that it is preferred to have some experimental data when training HMMs. However, if experimental data is not available, simulated data can still provide valuable information for classification. Future work could explore if a combined simulation and experimental data set could further improve the accuracy of classification with real world data.

5.3 HMM Estimates

The HMMs of each maneuver can be utilized to generate acceleration and jerk estimates. This process is further discussed in Chapter 3. The following section shows HMM estimates of acceleration and jerk when produced from experimental data trained HMMs and simulation data trained HMMs. It also compares these estimates using mean error and error variance.

5.3.1 Estimates Generated From Experimental Data Trained HMM

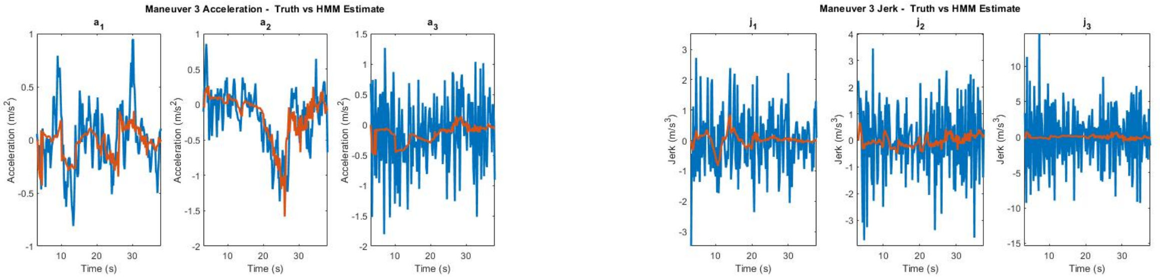
An example of the acceleration and jerk estimates produced for Maneuver 1, 3, 4 are show in Figures 5.31, 5.32, and 5.33. These estimates were produced using reference position data. In each instance the reference (in blue) has much higher noise than the estimates produced by the HMMs. The acceleration estimates are capable of following the general trend of the true acceleration. The accuracy of the jerk estimates, however, are more difficult to gauge because of the extremely high noise.



(a) Example acceleration HMM estimates compared to reference acceleration.

(b) Example jerk HMM estimates compared to reference jerk.

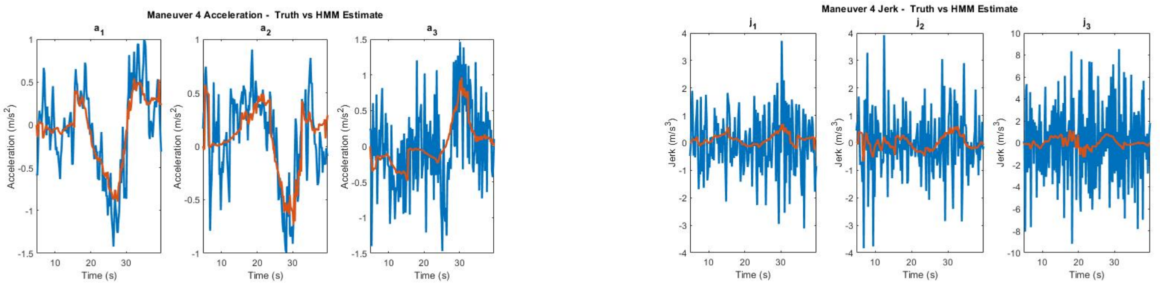
Figure 5.31: Acceleration and jerk estimates generated from Maneuver 1's HMM.



(a) Example acceleration HMM estimates compared to reference acceleration.

(b) Example jerk HMM estimates compared to reference jerk.

Figure 5.32: Acceleration and jerk estimates generated from Maneuver 3's HMM.



(a) Example acceleration HMM estimates compared to reference acceleration.

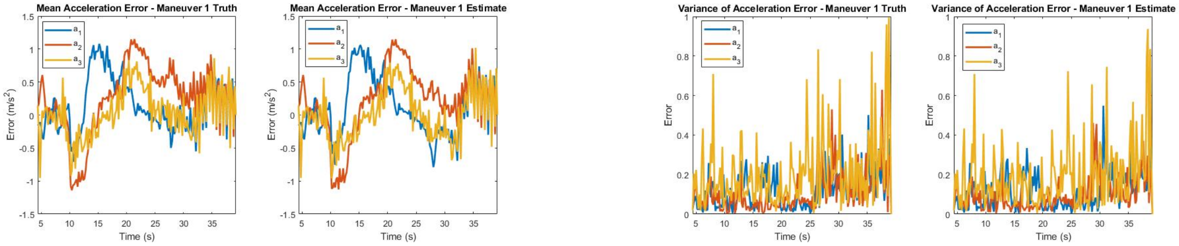
(b) Example jerk HMM estimates compared to reference jerk.

Figure 5.33: Acceleration and jerk estimates generated from Maneuver 4's HMM.

Maneuver 1

The mean acceleration error and error variance of the HMM estimates generated using either reference (test set) UAV position or estimated (test set) UAV position are shown in Figure 5.34. The mean errors follow a similar trend in both cases. The HMM estimate of a_1 has slightly more error towards the end of the maneuver when generated from estimated position. The variance of these errors are also similar, with an increase in error variance occurring towards the end of the maneuver for both scenarios.

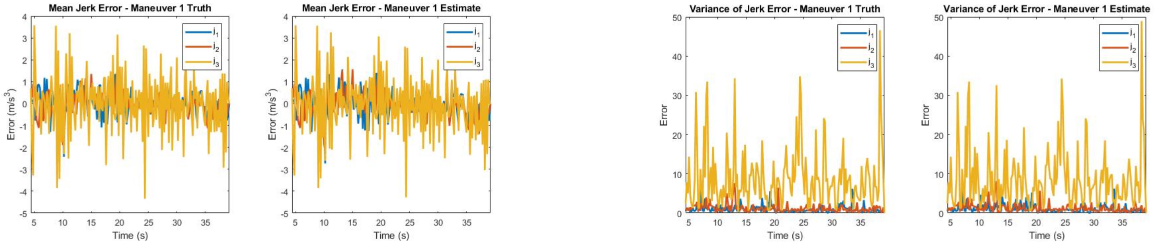
Figure 5.35 displays the mean jerk error and error variance for HMM estimates generated using reference UAV position and estimated UAV position. There is a slight bias in the error in the final ten seconds of the maneuver. The error variance also increases during this portion of the maneuver.



(a) Mean error of HMM acceleration estimates generated using reference position (left) and estimated position (right).

(b) Mean error variance of HMM acceleration estimates generated using reference position (left) and estimated position (right).

Figure 5.34: Mean error and error variance of HMM acceleration estimates generated for Maneuver 1.



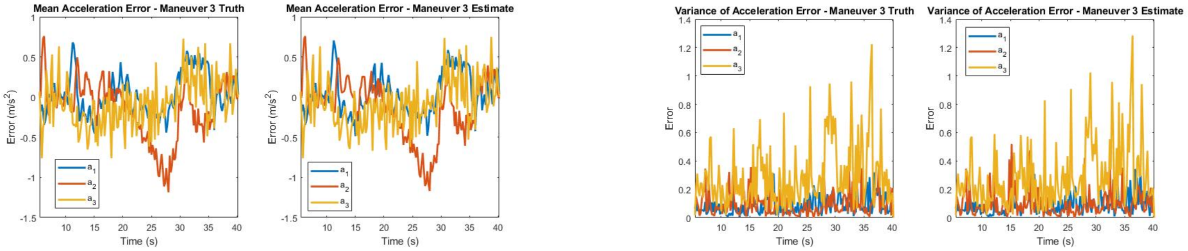
(a) Mean error of HMM jerk estimates generated using true position (left) and estimated position (right).

(b) Mean error variance of HMM jerk estimates generated using true position (left) and estimated position (right).

Figure 5.35: Mean error and error variance of HMM jerk estimates generated for Maneuver 1.

Maneuver 3

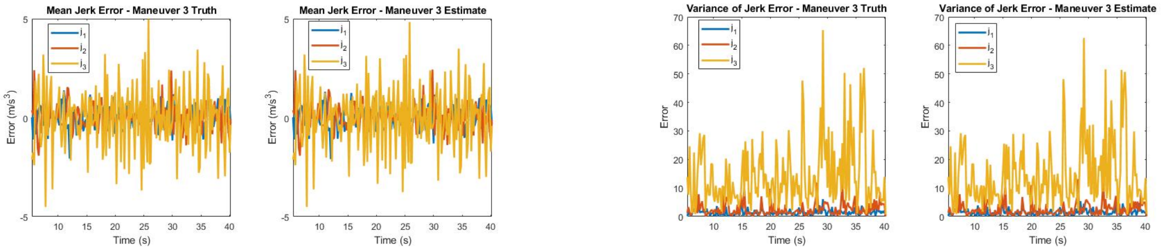
Maneuver 3 HMM acceleration estimation error is seen in Figure 5.36. The acceleration estimates produced have similar error, regardless if they were generated using true position or estimated position. The variance of these errors remain the same for both types of estimates as well. Figure 5.37 shows the jerk mean error and error variance for jerk estimates produced either by reference position or estimated position. The errors are similar for both estimates produced. They are close to zero-mean with high noise. The variances of these errors are consistent throughout the trajectory.



(a) Mean error of HMM acceleration estimates generated using true position (left) and estimated position (right).

(b) Mean error variance of HMM acceleration estimates generated using true position (left) and estimated position (right).

Figure 5.36: Mean error and error variance of HMM acceleration estimates generated for Maneuver 3.



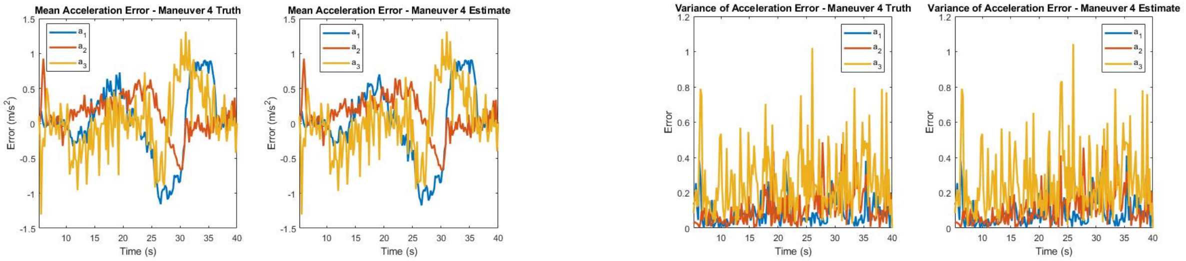
(a) Mean error of HMM jerk estimates generated using true position (left) and estimated position (right).

(b) Mean error variance of HMM jerk estimates generated using true position (left) and estimated position (right).

Figure 5.37: Mean error and error variance of HMM jerk estimates generated for Maneuver 3.

Maneuver 4

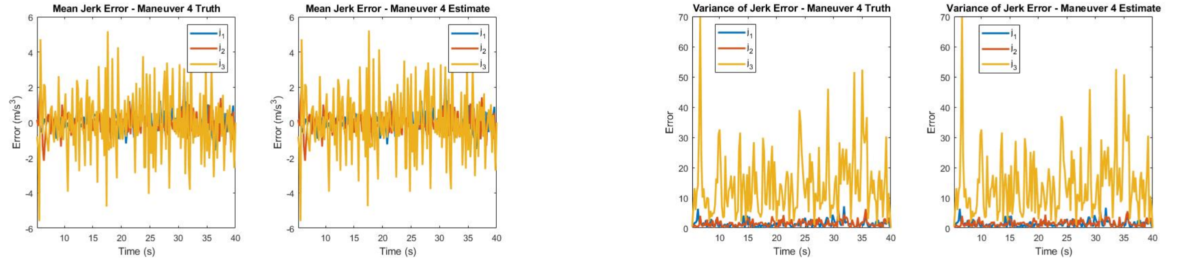
The mean error of the HMM acceleration estimates produced from Maneuver 4, as well as the error variance, is shown in Figure 5.38. The estimates generated from reference position or estimated position are indistinguishable in their mean error and error variance plots. Likewise the error of HMM jerk estimates are similar whether generated from reference or EKF position, as seen in Figure 5.39.



(a) Mean error of HMM acceleration estimates generated using true position (left) and estimated position (right).

(b) Mean error variance of HMM acceleration estimates generated using true position (left) and estimated position (right).

Figure 5.38: Mean error and error variance of HMM acceleration estimates generated for Maneuver 4.



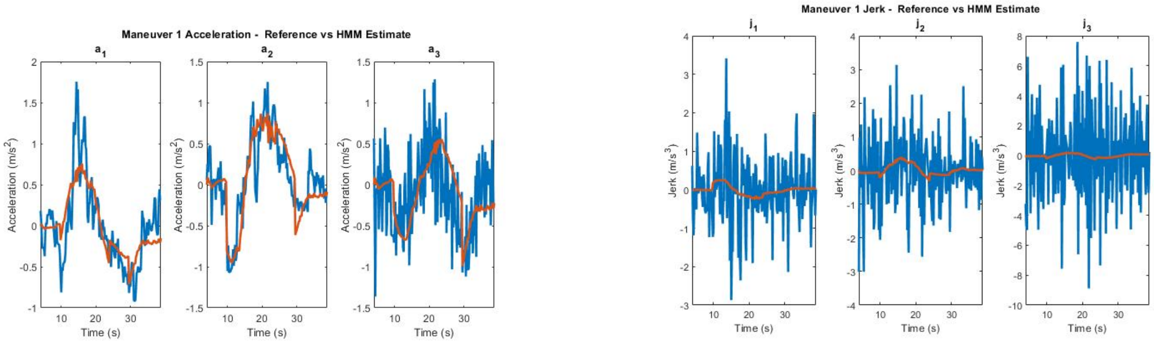
(a) Mean error of HMM jerk estimates generated using true position (left) and estimated position (right).

(b) Mean error variance of HMM jerk estimates generated using true position (left) and estimated position (right).

Figure 5.39: Mean error and error variance of HMM jerk estimates generated for Maneuver 4.

5.3.2 Estimates Generated From Simulation Data Trained HMM

Figure 5.40 shows an example of simulation trained HMM acceleration and jerk estimates for Maneuver 1. These estimates were generated using position reference data – not simulated data. The estimates are in orange while the reference data is in blue. The HMM estimates clearly filter much of the data, but are able to capture the general shape of the acceleration and jerk. Note that the HMM estimate of a_3 lags behind the reference, unlike estimates of a_1 and a_2. This could be the result of an inherent bias of the HMM's emission probability or an inconsistency resulting from the use of noisy and highly variable data.

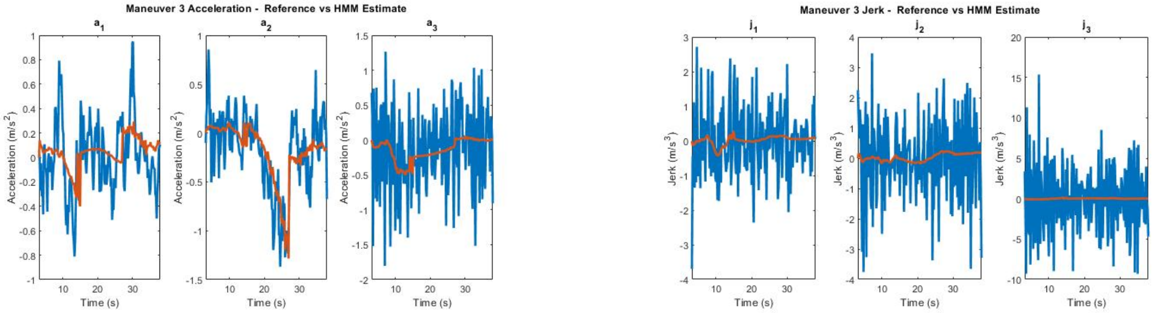


(a) Example simulation trained HMM acceleration estimates compared to reference.

(b) Example simulation trained HMM jerk estimates compared to reference.

Figure 5.40: Acceleration and jerk estimates generated from Maneuver 1's simulation trained HMM.

Figure 5.41 shows the acceleration and jerk estimates of Maneuver 3's simulation trained HMM when given reference positions of Maneuver 3. The estimates are once again capable of filtering most noise from the reference data. Some information seems to be lost in the estimates, such as the spike in a_1 near eight seconds or the oscillations in a_2 at the end of the trajectory. It's difficult to evaluate whether the jerk estimates capture the trends of the reference jerk, as the reference is noisy. There is some offset between estimate and reference, particularly for j_2, but overall the estimates are reasonable.

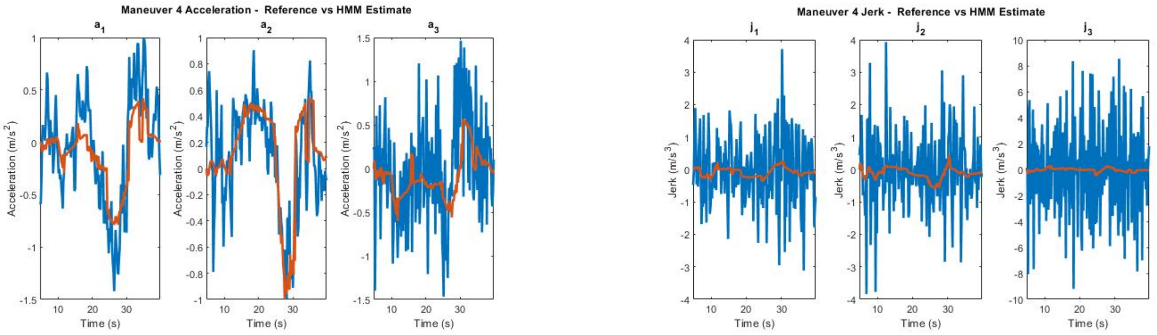


(a) Example simulation trained HMM acceleration estimates compared to reference.

(b) Example simulation trained HMM jerk estimates compared to reference.

Figure 5.41: Acceleration and jerk estimates generated from Maneuver 3's simulation trained HMM.

The acceleration and jerk estimates of Maneuver 4's simulation trained HMM is shown in Figure 5.42. These estimates were generated using reference position and are compared to the reference jerk and acceleration for the given maneuver. Maneuver 4's HMM estimates follow the reference well. In the case of a_1 and a_3, the magnitude of the estimates is less than that of the reference, but the general movement is still captured.



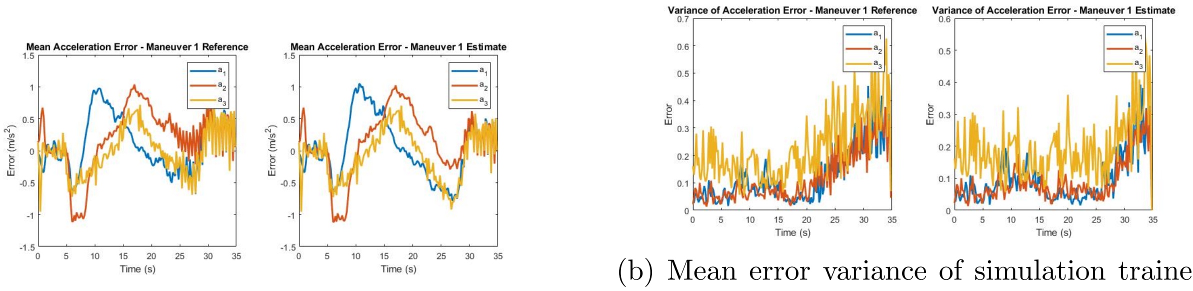
(a) Example simulation trained HMM acceleration estimates compared to reference.

(b) Example simulation trained HMM jerk estimates compared to jerk.

Figure 5.42: Acceleration and jerk estimates generated from Maneuver 4's simulation trained HMM.

Maneuver 1

The mean error and error variance of acceleration estimates generated by the simulation trained Maneuver 1 HMM are shown in Figure 5.43. The figure displays the mean error and error variance for estimates created using reference position and estimated position. The difference in error little difference in the two types of estimates. In both cases, there is error in acceleration estimates throughout the trajectory indicative of estimation lag. The error variance is small in the beginning of the trajectory, but at twenty to twenty-five seconds, the variance begins to grow.

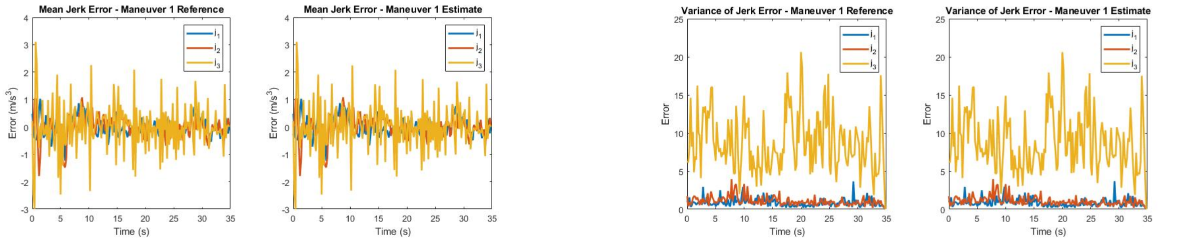


(a) Mean error of simulation trained HMM acceleration estimates generated using reference position (left) and estimated position (right).

(b) Mean error variance of simulation trained HMM acceleration estimates generated using reference position (left) and estimated position (right).

Figure 5.43: Mean error and error variance of simulation trained HMM acceleration estimates generated for Maneuver 1.

Figure 5.44 shows the mean error and error variance of jerk estimates from Maneuver 1's simulation trained HMM. j_3 maintains a mean error near zero, while j_1 and j_2 estimates produce error in the initial moments of the maneuver. All of the jerk estimates' error variances are consistent throughout the trajectory. j_3 has the highest error variance overall.



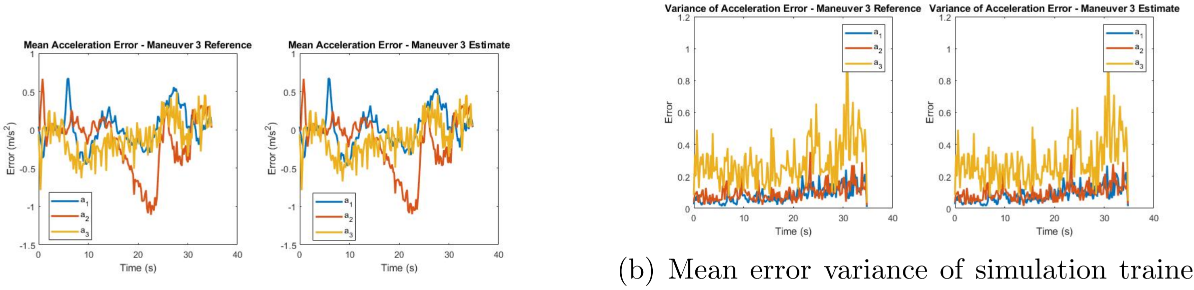
(a) Mean error of simulation trained HMM jerk estimates generated using reference position (left) and estimated position (right).

(b) Mean error variance of HMM jerk estimates generated using reference position (left) and estimated position (right).

Figure 5.44: Mean error and error variance of simulation trained HMM jerk estimates generated for Maneuver 1.

Maneuver 3

Figure 5.45 shows the mean error and error variance of HMM acceleration estimates. A simulation trained HMM was used to generate these estimates using real world reference data or estimated reference data. The mean error and error variance is consistent for estimates generated from reference or estimated data. Throughout the maneuver, the acceleration estimates have high error. The error variance remains consistent for the first half of the trajectory before slowly increasing throughout the second half of the trajectory.

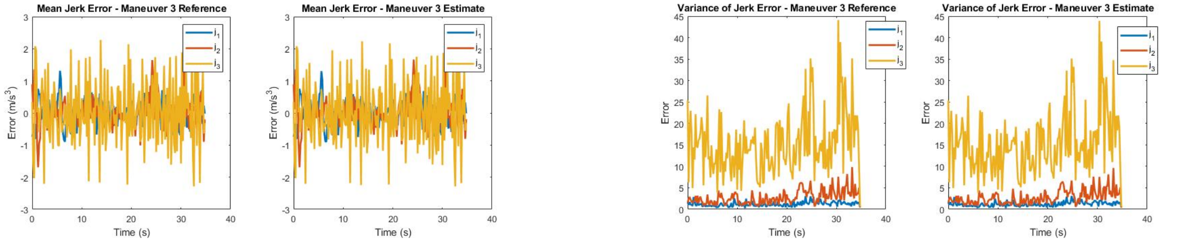


(a) Mean error of simulation trained HMM acceleration estimates generated using reference position (left) and estimated position (right).

(b) Mean error variance of simulation trained HMM acceleration estimates generated using reference position (left) and estimated position (right).

Figure 5.45: Mean error and error variance of simulation trained HMM acceleration estimates generated for Maneuver 3.

Mean error and error variance of HMM jerk estimates are shown in Figure 5.46. Both j_2 and j_3 have a mean error near zero – although the error is noisy. The mean error of j_1 contains some biases, but remains low. Jerk estimate error variances remain consistent until the final stages of flight, much like the acceleration error variances. Once again, there is little difference between estimates generated from reference position and those generated from estimated reference position.



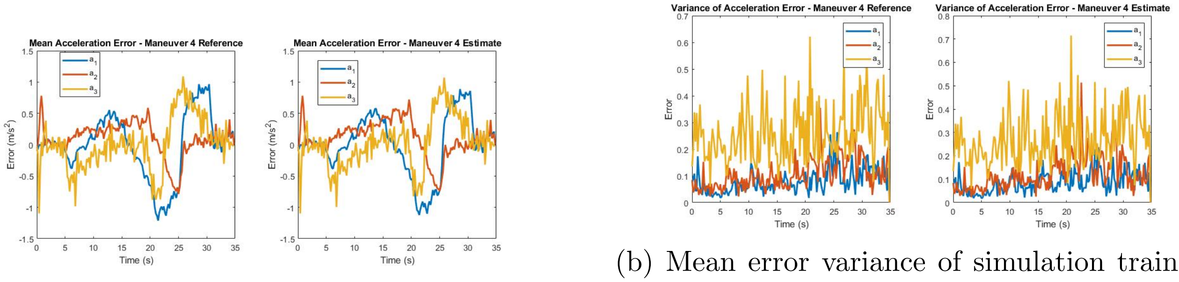
(a) Mean error of simulation trained HMM jerk estimates generated using reference position (left) and estimated position (right).

(b) Mean error variance of simulation trained HMM jerk estimates generated using reference position (left) and estimated position (right).

Figure 5.46: Mean error and error variance of simulation trained HMM jerk estimates generated for Maneuver 3.

Maneuver 4

Maneuver 4's simulation trained HMM was used to generate estimated acceleration and jerk from reference position and estimated reference position. The mean error and error variance of the acceleration estimates are shown in Figure 5.47. The error variance remains consistent for most of the maneuver.

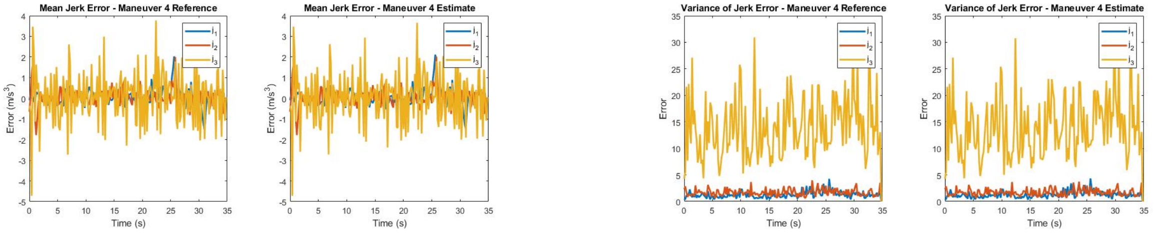


(a) Mean error of simulation trained HMM acceleration estimates generated using reference position (left) and estimated position (right).

(b) Mean error variance of simulation trained HMM acceleration estimates generated using reference position (left) and estimated position (right).

Figure 5.47: Mean error and error variance of simulation trained HMM acceleration estimates generated for Maneuver 4.

The mean error and error variance of Maneuver 4's jerk estimates are shown in Figure 5.48. The mean error for all jerk directions remains near zero, while the jerk variance is consistent throughout the trajectory.



(a) Mean error of simulation trained HMM jerk estimates generated using reference position (left) and estimated position (right).

(b) Mean error variance of simulation trained HMM jerk estimates generated using reference position (left) and estimated position (right).

Figure 5.48: Mean error and error variance of simulation trained HMM jerk estimates generated for Maneuver 4.

5.4 EKF + HMM Results

The following section presents results of the EKF+HMM when using experimental data as an input to the system. The section will compared EKF+HMM performance when using simulation data trained HMMs and experimental data trained HMMs. Process noise for each filter and radar position were kept constant for each run.

For each maneuver, each of the five test trajectories were estimated with the EKF+HMM that uses HMMs trained from experimental reference data. The sum of state estimation error over each run was found and used as the total state estimation error. Scenario 1A represents the trajectory that had the lowest total state estimation error, while Scenario 2A shows the trajectory that had the highest total state estimation error for the EKF+HMM. The results of both scenarios are shown and compared to a standard EKF. The same trajectories are used to test the EKF+HMM that uses HMMs trained from simulated data. These results are referred to as Scenario 1B and 2B. Once again these results will be compared to a standard EKF.

5.4.1 Maneuver 1

Scenario 1A - Experimental Data Trained HMM

In Scenario 1, the EKF+HMM had a classification accuracy over time of 80%. The confidence in each HMM throughout classification is shown in Figure 5.49. Each maneuver HMM has a low initial confidence. Maneuver 1 begins to fully distinguish itself at about 11 seconds.

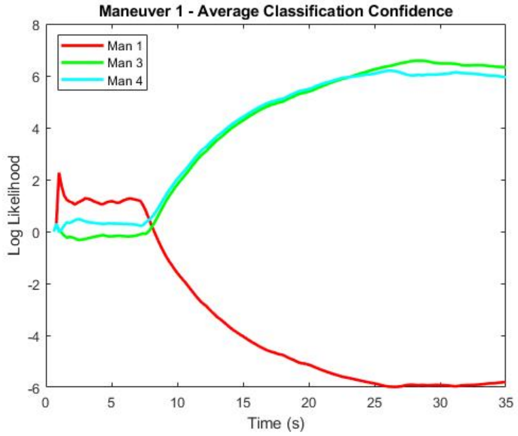
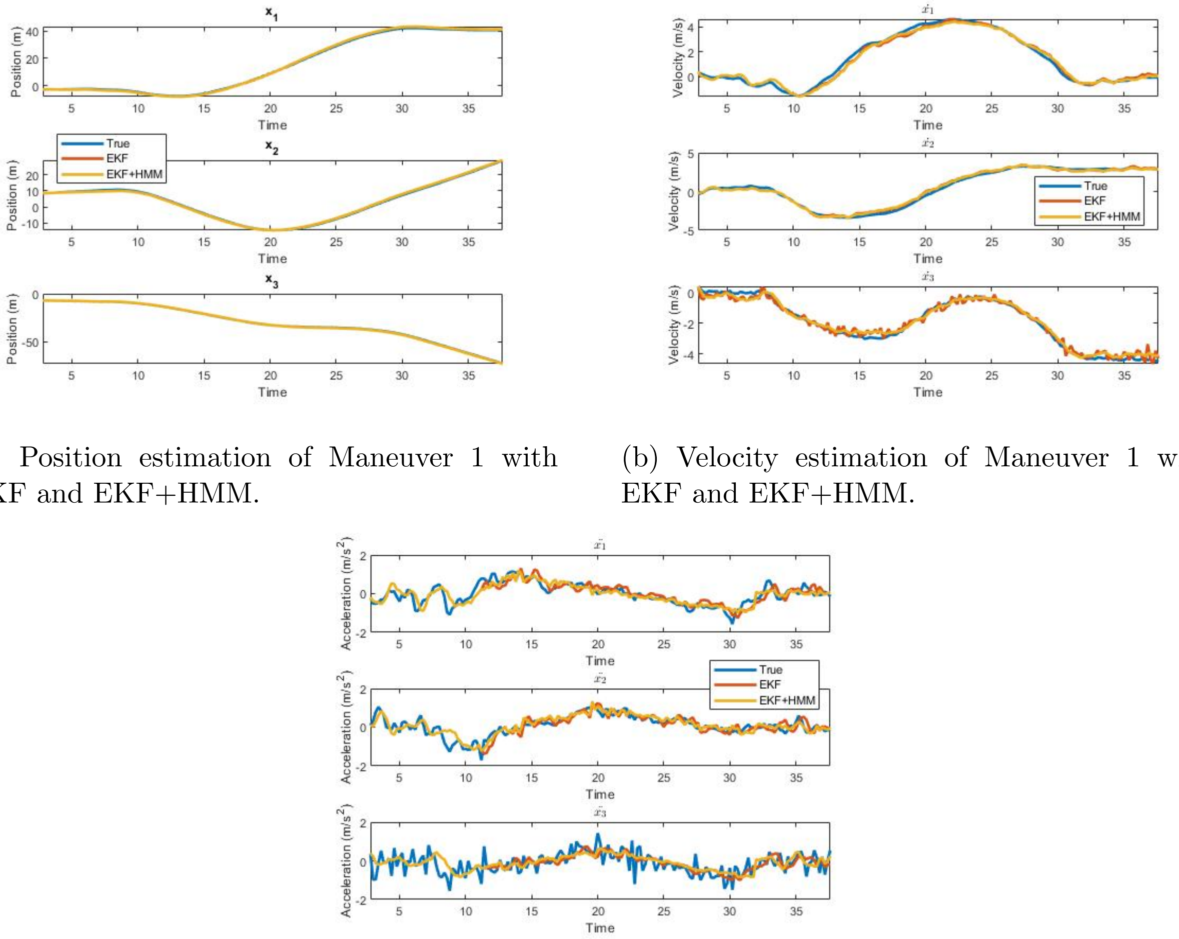


Figure 5.49: Maneuver 1: Confidence of classification over time.

The results of the EKF and EKF+HMM estimating UAV states throughout Maneuver 1 is shown in Figure 5.50. At 11 seconds, when the EKF+HMM begins to use HMM estimates, the clear distinction between EKF and EKF+HMM estimation can be seen. Both the EKF and EKF+HMM have lag in their estimation. This is especially evident in Figure 5.50b. When estimating acceleration, both filter noise in the reference data.



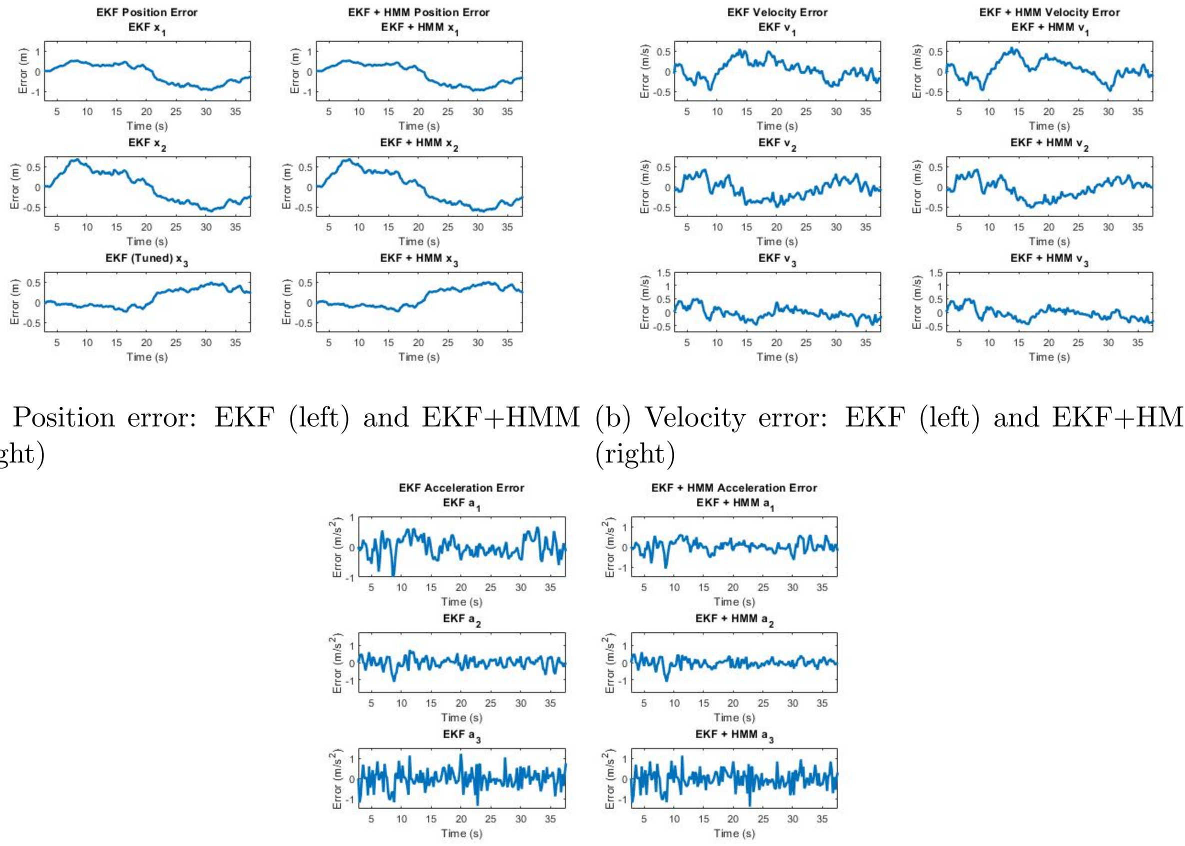
(a) Position estimation of Maneuver 1 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 1 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 1 with EKF and EKF+HMM.

Figure 5.50: Maneuver 1 UAV state estimation - Scenario 1A.

The EKF and EKF+HMM state errors are compared in Figure 5.51. Although the EKF+HMM provides HMM estimates to the EKF+HMM for over half of the maneuver, little difference is seen between the EKF and EKF+HMM. It could be that Maneuver 1 is not dynamic enough to need the additional information from HMM estimates of acceleration and jerk. The state errors show that, at the least, the HMM estimates are not detrimental to the UAV state estimation.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.51: Comparison of state errors for EKF and EKF+HMM.

The variance over time for each state in the EKF+HMM is displayed in Figure 5.52. When the EKF switches to the EKF+HMM, the acceleration variances become constant as they are no longer updated with acceleration state information.

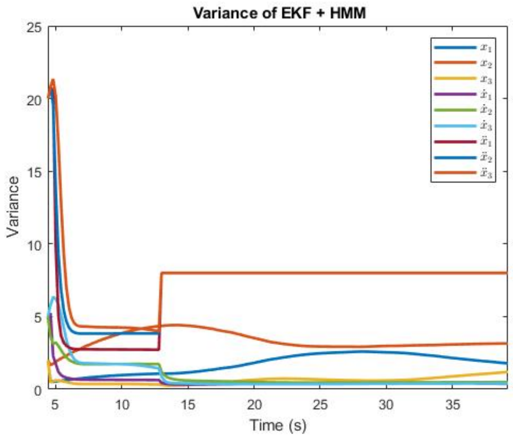


Figure 5.52: Variance of EKF+HMM states over time

Scenario 1B - Simulation Data Trained HMM

The same trajectory was then estimated with an EKF+HMM that used HMMs trained with simulated data. The classification accuracy over time of the classifier was 72.47%. This is slightly less that seen in Scenario 1A. Figure 5.53 shows the classification confidence over time for the trajectory.

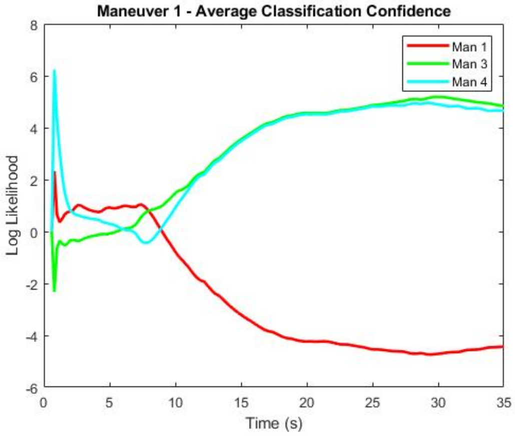
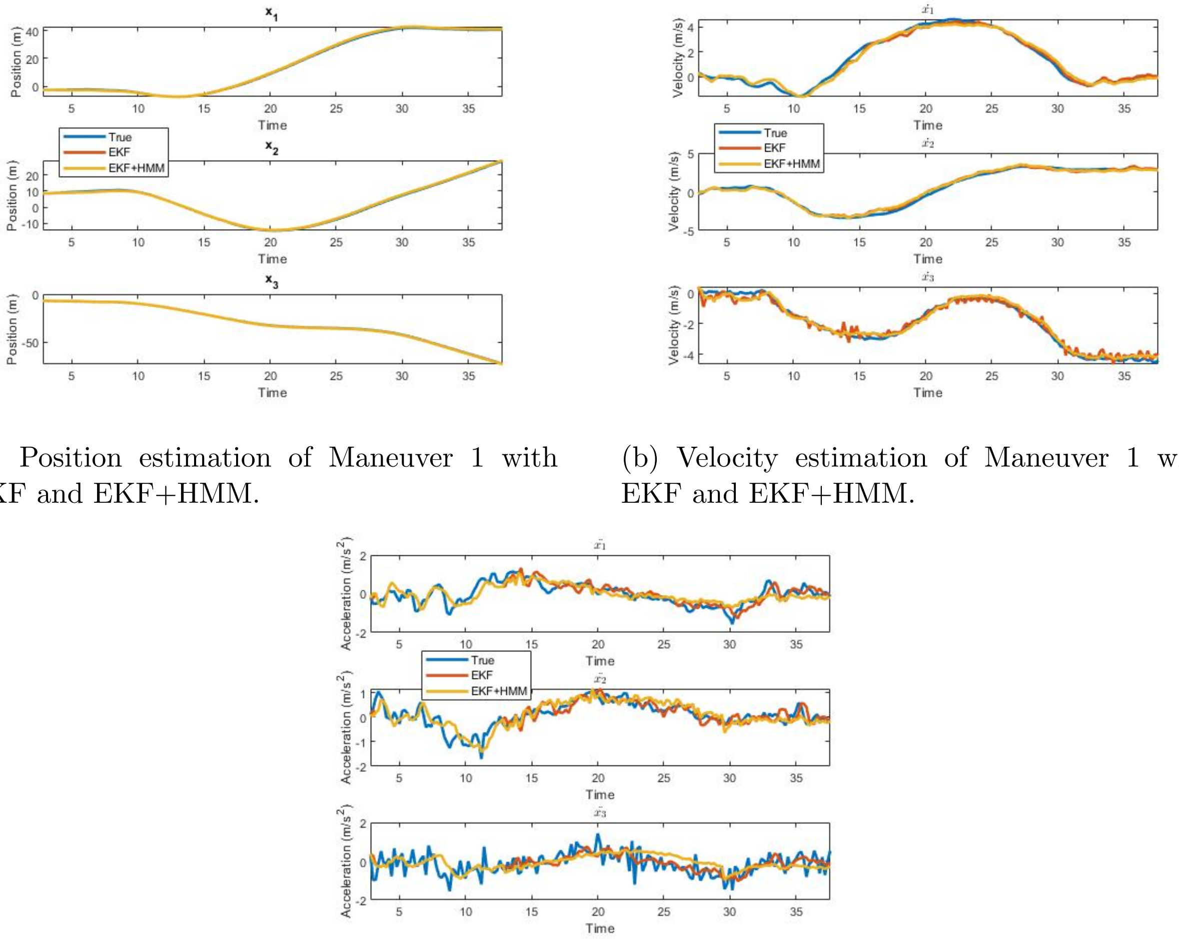


Figure 5.53: Maneuver 1: Confidence of classification over time using simulation trained HMM.

Estimation of the trajectory is shown in Figure 5.54. The EKF+HMM begins using HMM estimates around 13 seconds. In comparison to Scenario 1A, there is more offset error in acceleration estimation of a_3. Compared the standard EKF, the EKF+HMM filters more noise and disturbances, particularly in the acceleration estimation, but has roughly the same amount of lag as the EKF.



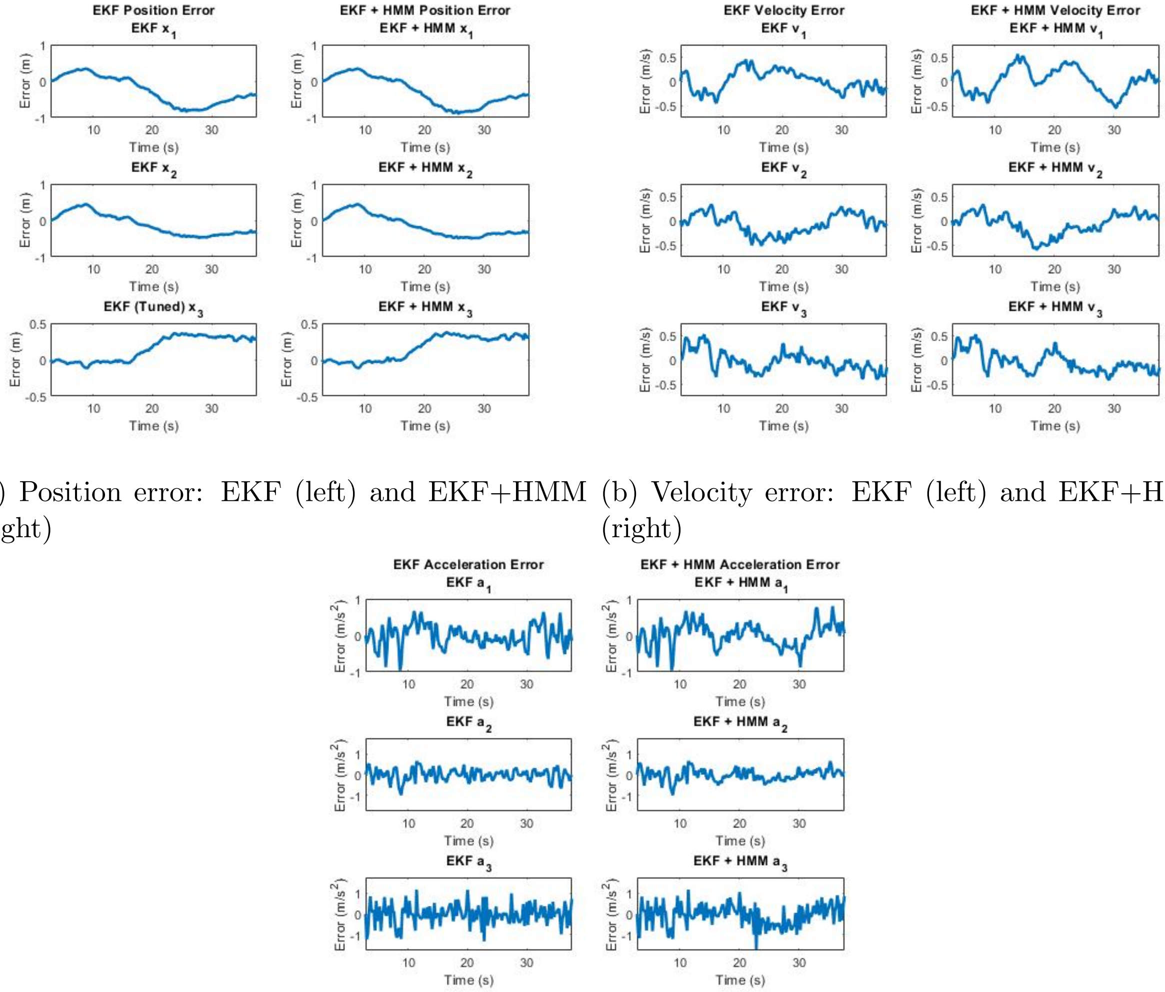
(a) Position estimation of Maneuver 1 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 1 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 1 with EKF and EKF+HMM.

Figure 5.54: Maneuver 1 UAV state estimation - Scenario 1B.

The state error of both the EKF and EKF+HMM are displayed in Figure 5.55. The EKF and EKF+HMM have similar error over time for most states. However, state error is greater for the EKF+HMM than the EKF when estimating acceleration states and v_1. The state variances of the EKF+HMM are shown in Figure 5.56.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.55: Comparison of state errors for EKF and EKF+HMM using simulation trained HMM.

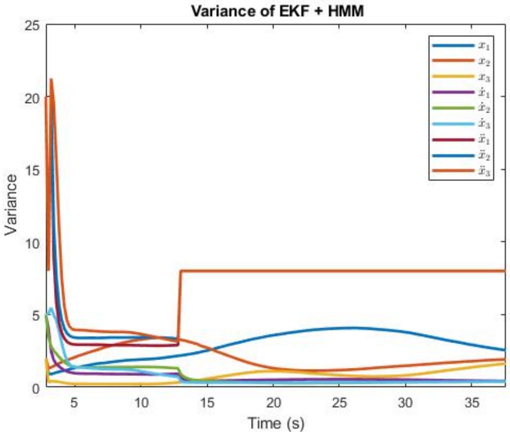


Figure 5.56: Variance of EKF+HMM states over time

Scenario 2A - Experimental Data Trained HMM

Scenario 2 shows the results of the Maneuver 1 test trajectory that had the greatest state error for the EKF+HMM. This maneuver had a classification accuracy over time of 91.33%. Confidence in each maneuver HMM throughout classification is shown in Figure 5.57.

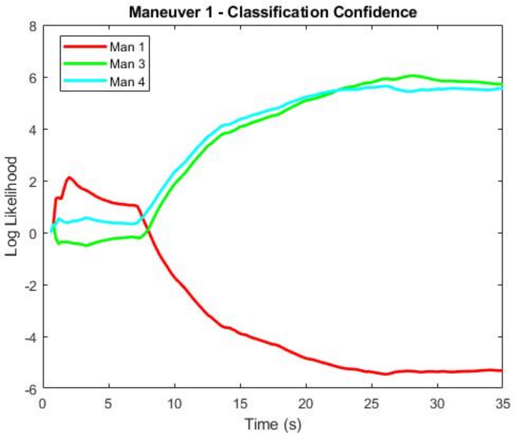
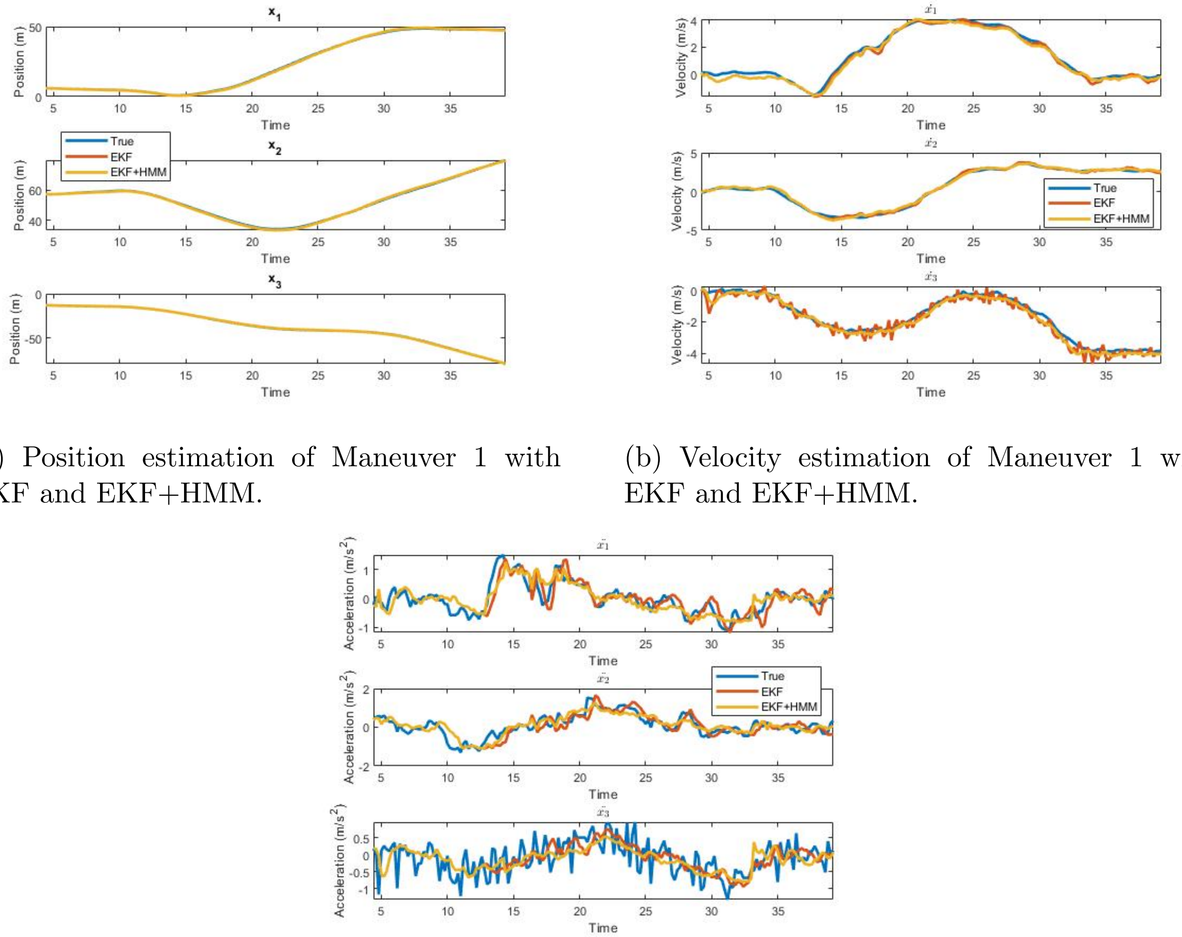


Figure 5.57: Maneuver 1: Confidence of classification over time.

Estimation of each state by the EKF and EKF+HMM is shown in Figure 5.58. This trajectory has reference states that are noisier than Scenario 1. An interesting difference is created between the EKF and EKF+HMM, where the EKF primarily attempts to follow the noise and disturbances when estimating a_1 and a_2, while the EKF+HMM filters most of the noise. Because the reference states are found from sensor measurements, it is difficult to know which filter is making the “right choice”. Note also that both filters still have lag when estimating UAV states.



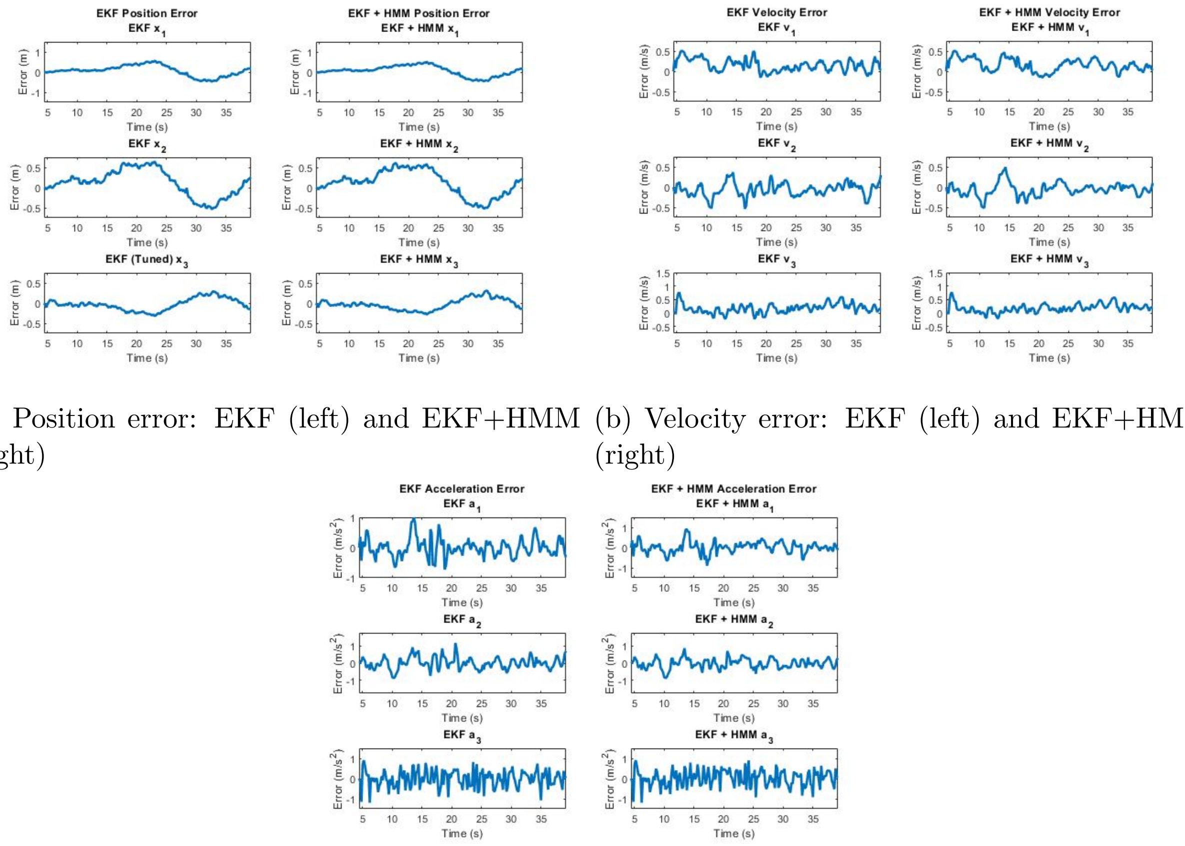
(a) Position estimation of Maneuver 1 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 1 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 1 with EKF and EKF+HMM.

Figure 5.58: Maneuver 1 UAV state estimation - Scenario 2A.

The state error over time of both filters (Figure 5.59) remains similar for each state. The EKF+HMM has slightly less noisy estimates of a_1 and a_2, while having increased error for v_1 and v_2. Still, these differences are small and do not make a discernible difference. Figure 5.60 shows the variance of each EKF+HMM state, which follows similar trends to Figure 5.52.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.59: Comparison of state errors for EKF and EKF+HMM.

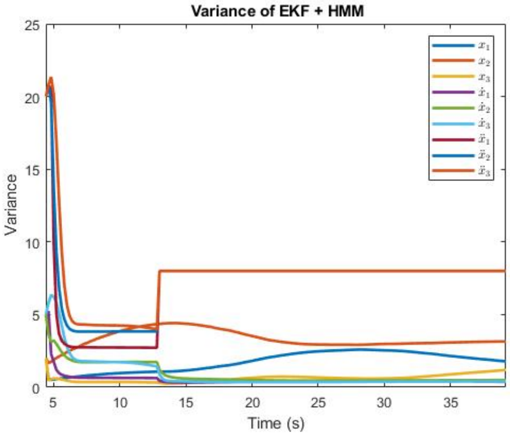


Figure 5.60: Variance of EKF+HMM states over time

Scenario 2B - Simulation Data Trained HMM

In Scenario 2B, the maneuver had a classification accuracy over time of 74.47%. The confidence of each simulation trained HMM over time is shown in Figure 5.61. The maneuver is classified early in its trajectory – meaning HMM estimates are used for much of the estimation process.

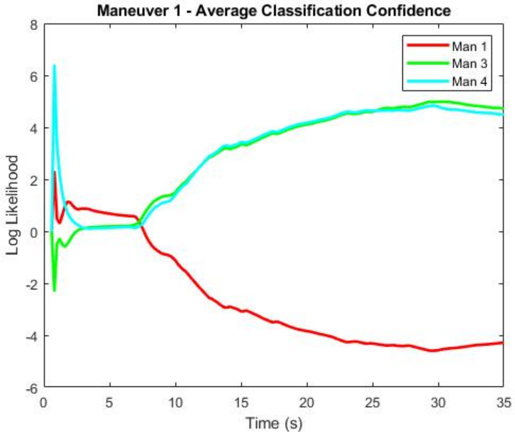
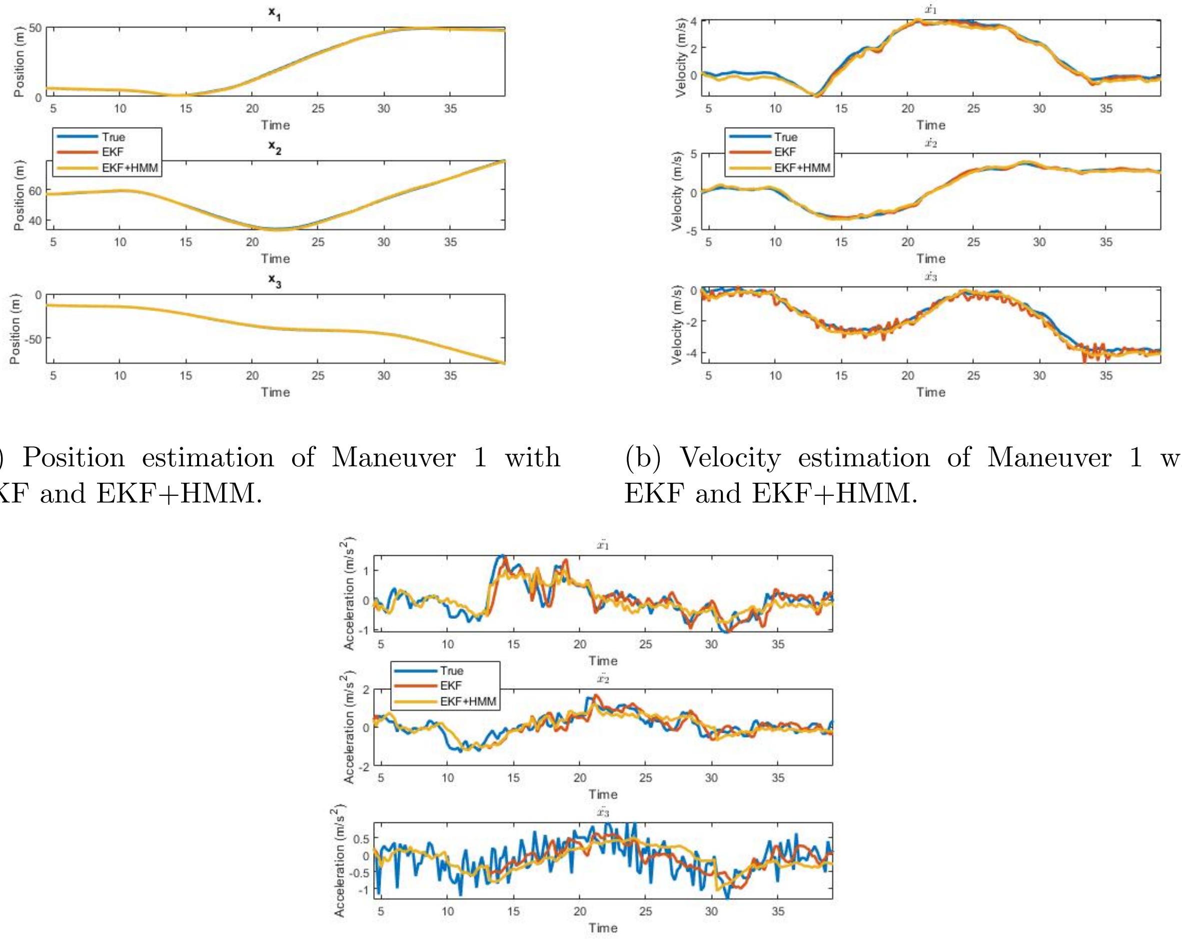


Figure 5.61: Maneuver 1: Confidence of classification over time using simulation trained HMM.

Figure 5.62 shows the EKF and EKF+HMM estimation of a Maneuver 1 trajectory. The EKF+HMM filters more noise and disturbance than the EKF. EKF+HMM estimation results are very similar to those seen in Scenario 2A when using experimental data trained HMMs. The greatest difference between EKF+HMM estimates in Scenario 2A and 2B is the increase of a_3 estimation error in Scenario 2B.



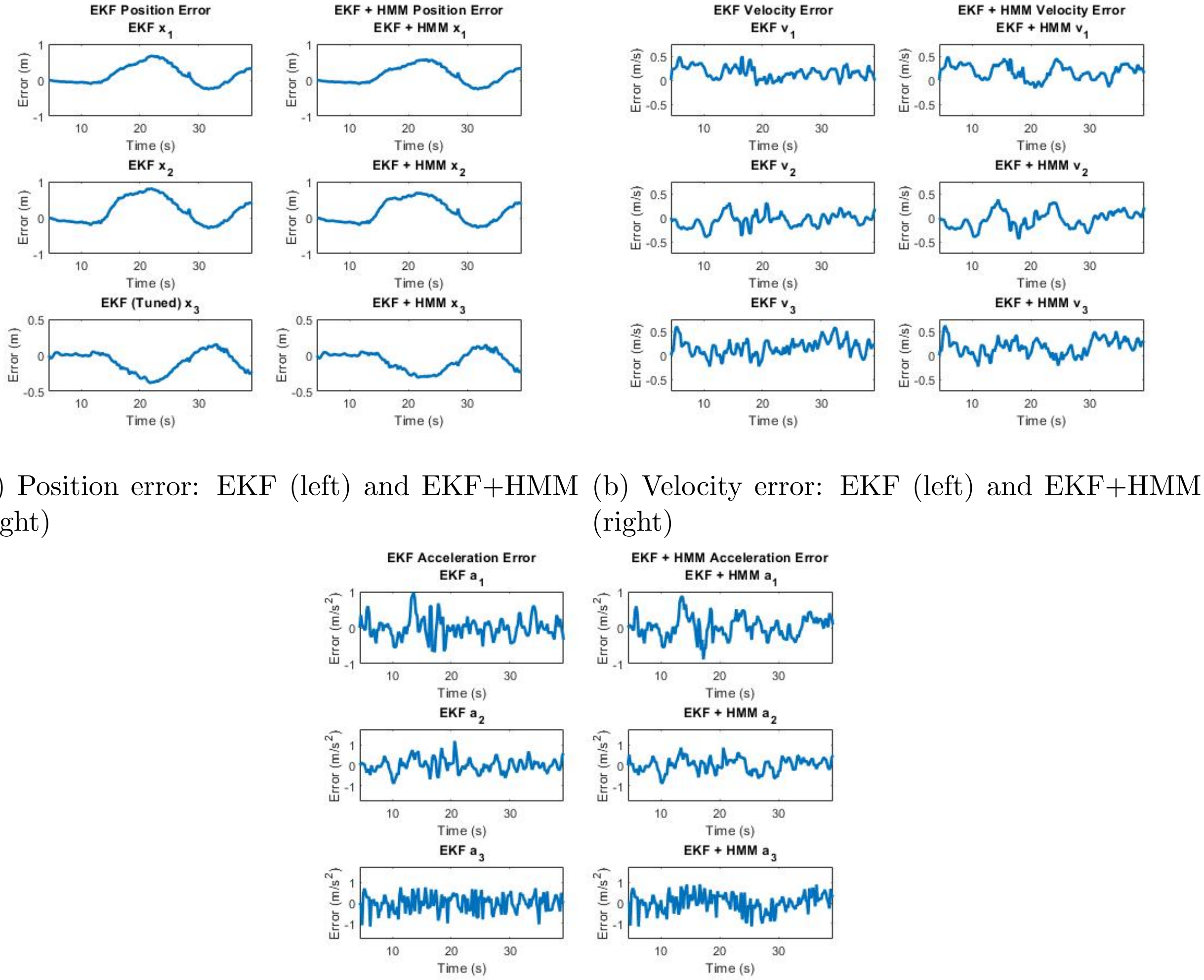
(a) Position estimation of Maneuver 1 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 1 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 1 with EKF and EKF+HMM.

Figure 5.62: Maneuver 1 UAV state estimation - Scenario 2B.

The state errors over time of the EKF and EKF+HMM are compared in Figure 5.63. The EKF and EKF+HMM are overall quite similar. However, EKF+HMM acceleration and velocity errors are slightly more than that of the EKF – likely because of erroneous HMM estimates. The state variance of the EKF+HMM is shown in Figure 5.64. They are similar to that of Scenario 2B.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.63: Comparison of state errors for EKF and EKF+HMM using simulation trained HMM.

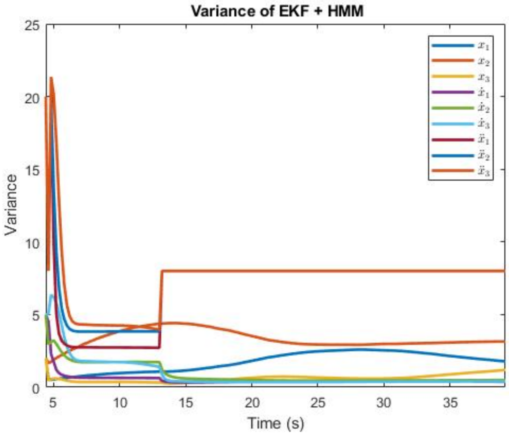


Figure 5.64: Variance of EKF+HMM states over time

5.4.2 Maneuver 3

Scenario 1A - Experimental Data Trained HMM

The "best case" EKF+HMM estimation of Maneuver 3 is discussed in this section. The classification accuracy over time was 66.67%. As shown in Figure 5.65, high confidence in the maneuver was not found until after 10 seconds.

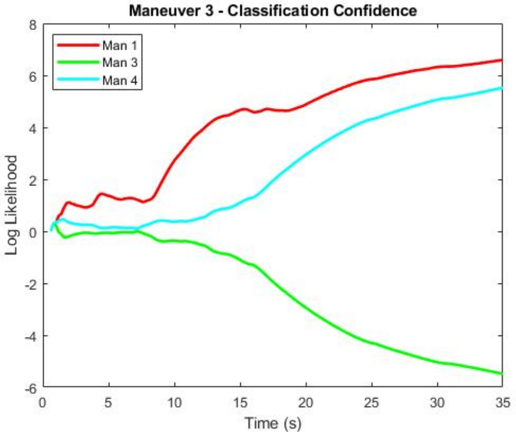
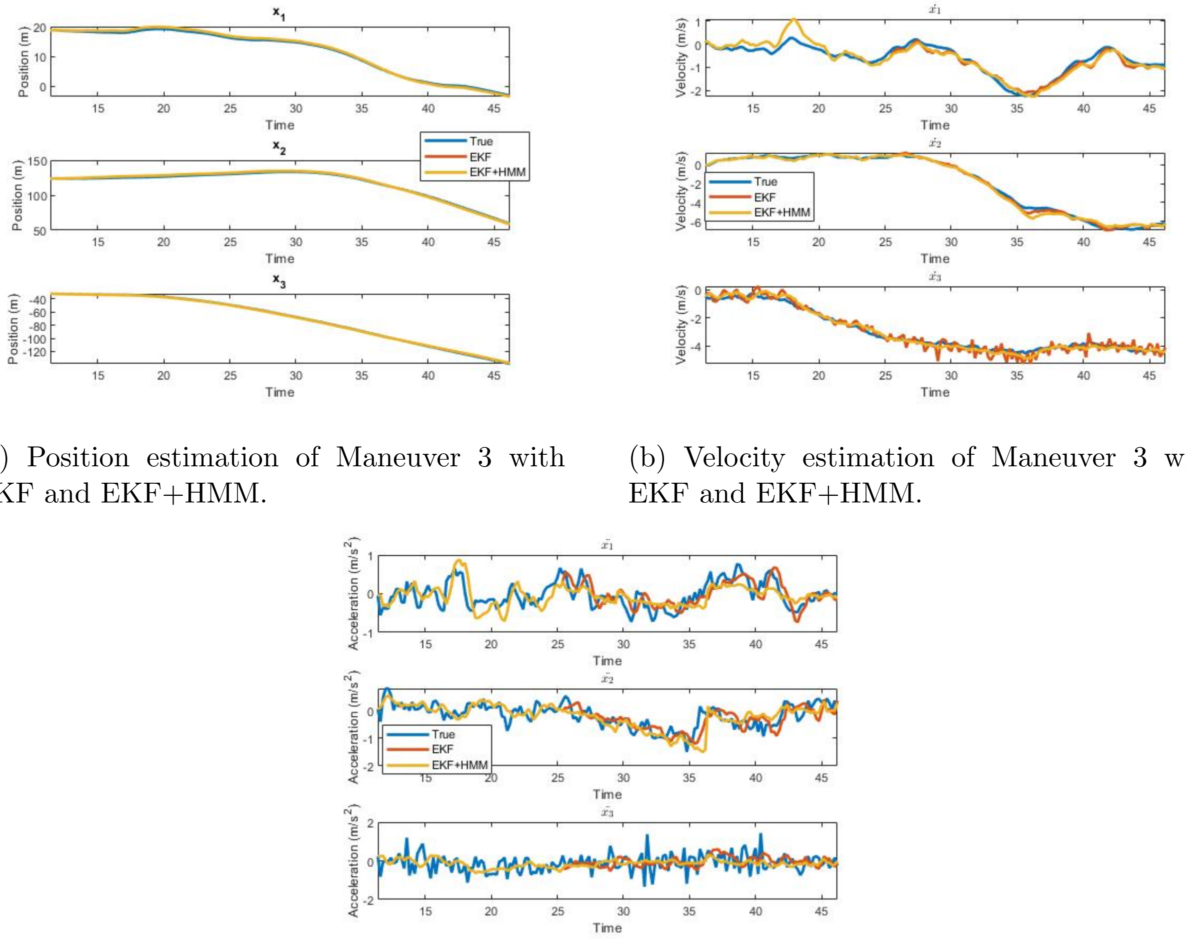


Figure 5.65: Maneuver 3: Confidence of classification over time.

Figure 5.66 shows the EKF and EKF+HMM state estimation of the UAV throughout Maneuver 3. The position and velocity estimates of both filters are quite similar, although the EKF is noisier at times. Like Scenario 2A in Maneuver 1, the EKF+HMM attenuates more noise than the EKF.



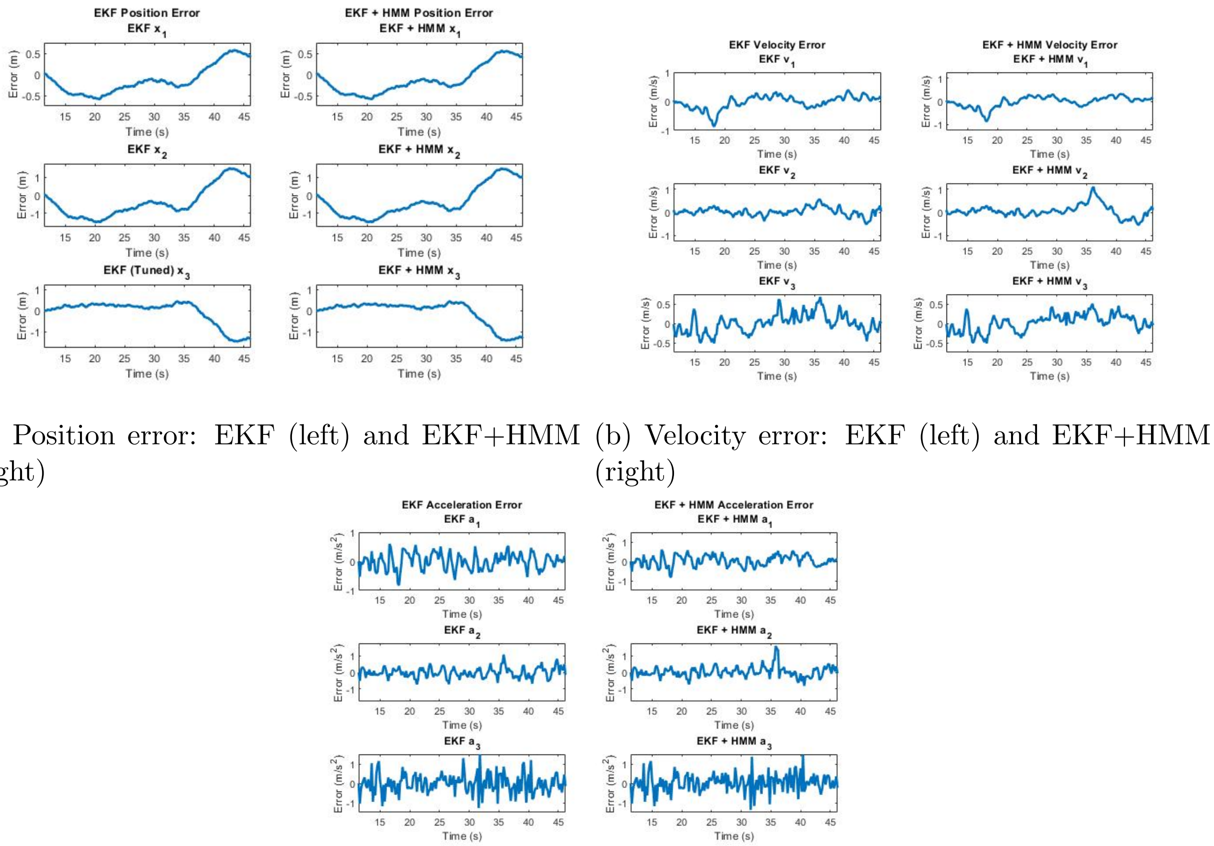
(a) Position estimation of Maneuver 3 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 3 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 3 with EKF and EKF+HMM.

Figure 5.66: Maneuver 3 UAV state estimation - Scenario 1A.

The errors of each filter state estimate is shown in Figure 5.67. Position estimation errors are nearly identical for both filters. When estimating velocity, the EKF+HMM has more error in the estimation of v_2, but comparable error for v_1 and v_3. When estimating acceleration, the EKF+HMM estimates are less noisy, but overall follow very closely with the EKF errors. The variance of each EKF+HMM state throughout the UAV maneuver is depicted in Figure 5.68.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.67: Comparison of state errors for EKF and EKF+HMM.

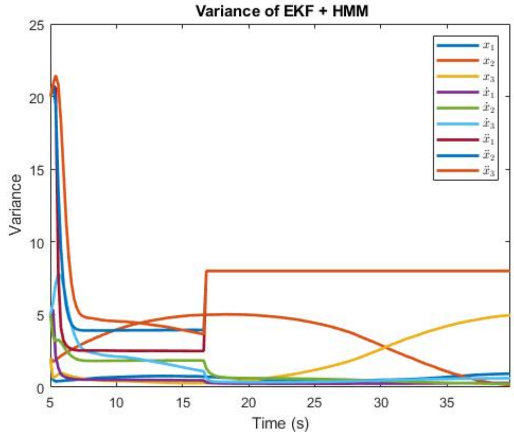


Figure 5.68: Variance of EKF+HMM states over time

Scenario 1B - Simulation Data Trained HMM

When using simulation trained HMMs, the classification accuracy over time of the same trajectory for Maneuver 3 was 49.56%. Figure 5.69 shows the confidence of each model over time for the maneuver. The classifier required more time and observations to have a confident classification.

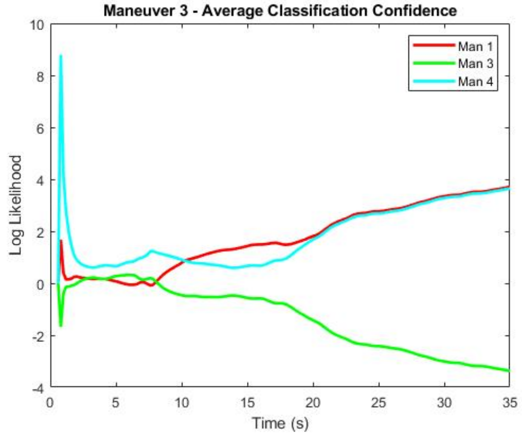
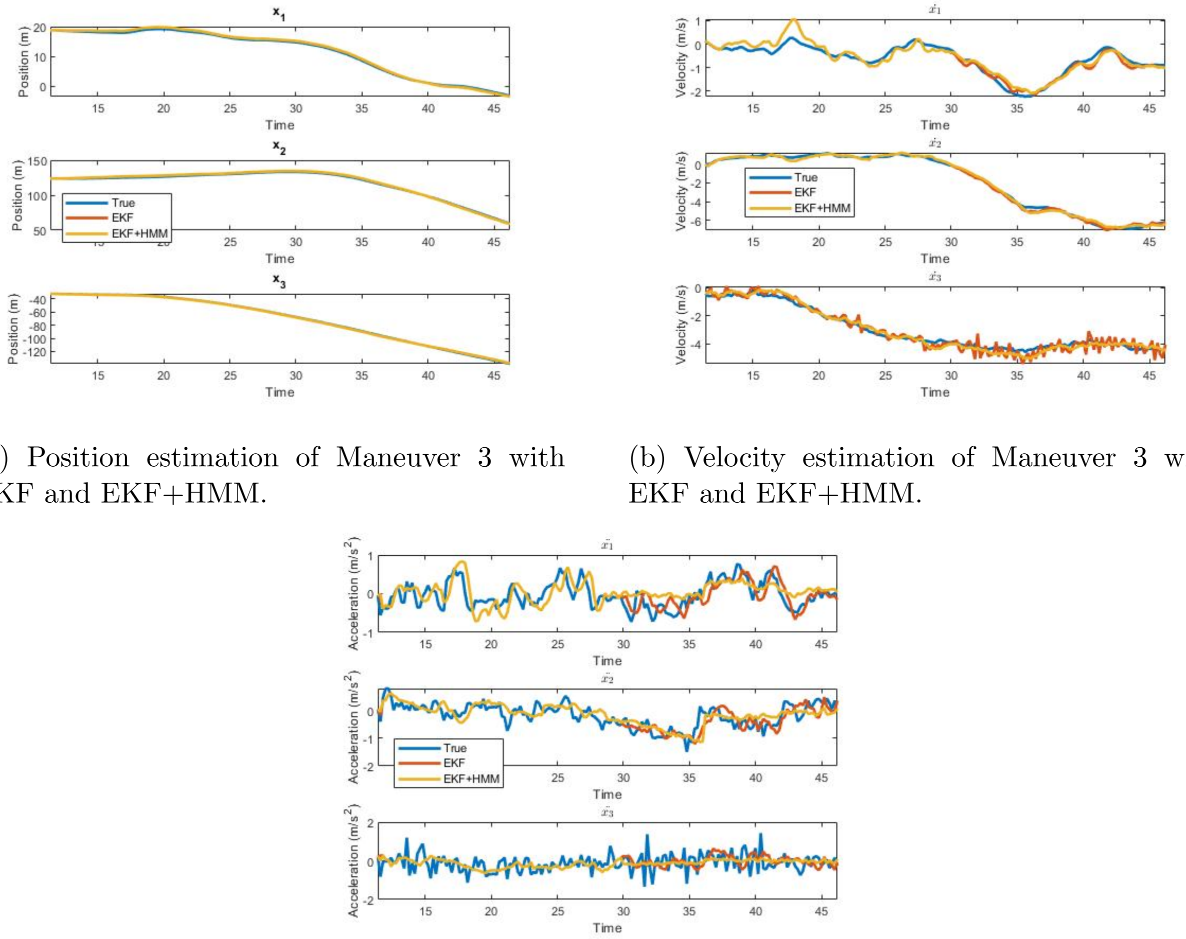


Figure 5.69: Maneuver 3: Confidence of classification over time using simulation trained HMM.

State estimation of the trajectory using the EKF and EKF+HMM is shown in Figure 5.70. The EKF+HMM estimates are extremely similar to the EKF+HMM estimates in Scenario 1A. The greatest difference between the two is the time taken for HMM estimates to be used. Because the simulation trained HMMs have such a low classification accuracy over time, HMM estimates are not incorporated into the EKF+HMM until later in the trajectory.



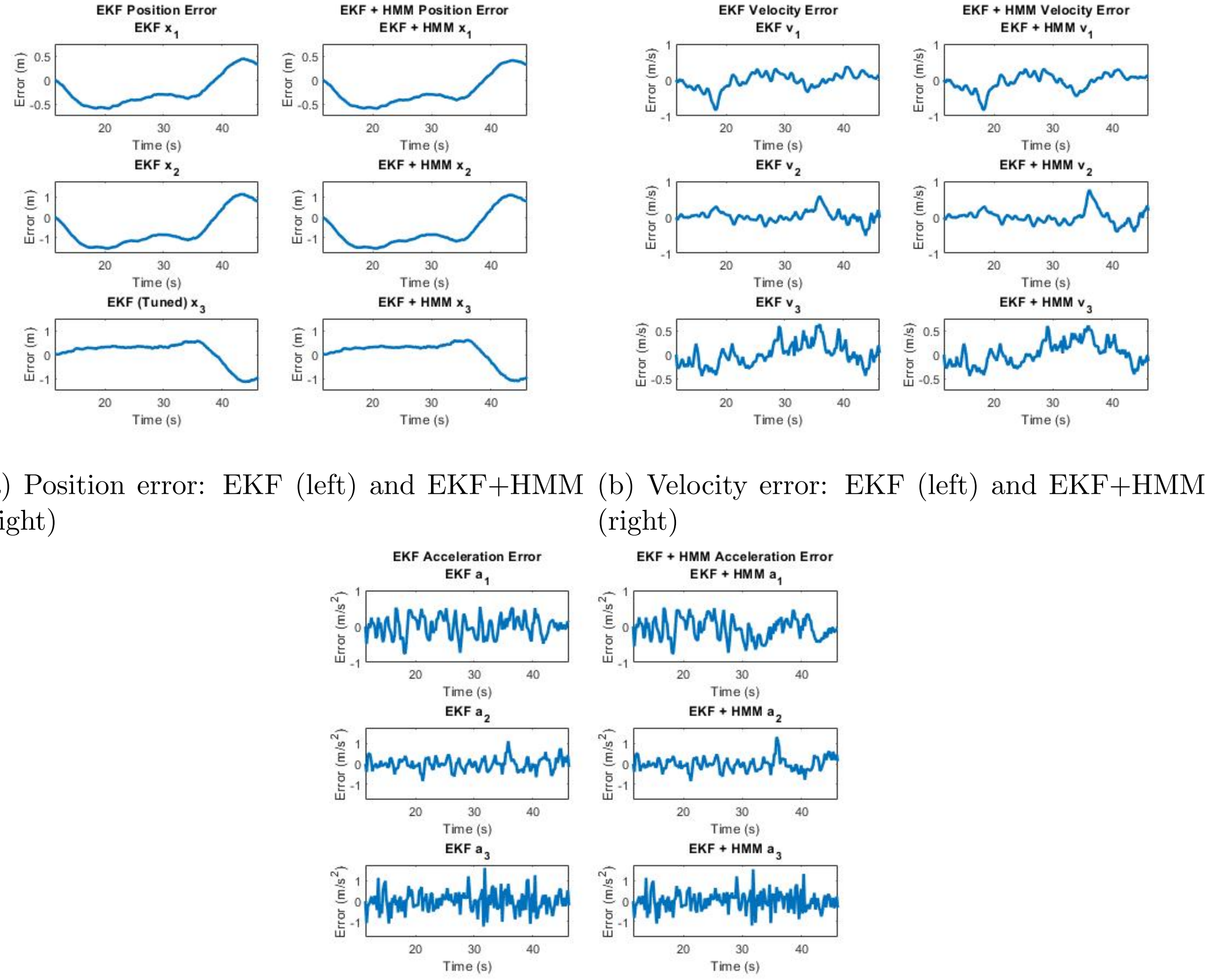
(a) Position estimation of Maneuver 3 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 3 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 3 with EKF and EKF+HMM.

Figure 5.70: Maneuver 3 UAV state estimation - Scenario 1B.

The state error over time for the EKF and EKF+HMM is shown in Figure 5.71. Once again, the EKF+HMM has greater error than the EKF for estimates of v_2 and a_2, but all other estimates remain comparable to the EKF. The state variance of the EKF+HMM over time is shown in Figure 5.72.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.71: Comparison of state errors for EKF and EKF+HMM using simulation trained HMM.

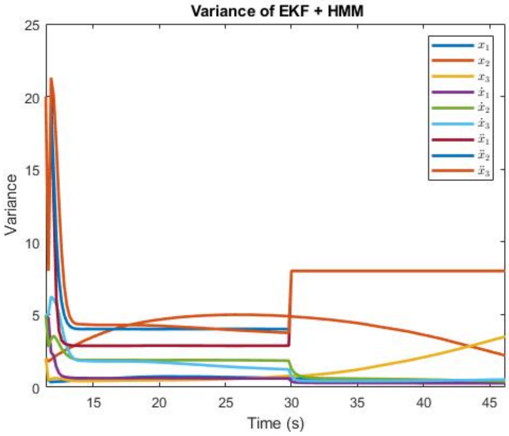


Figure 5.72: Variance of EKF+HMM states over time

Scenario 2A - Experimental Data Trained HMM

The "worst case" Maneuver 3 trajectory had a classification accuracy over time of 63.33%. The confidence in each maneuvers' HMM is shown in Figure 5.73.

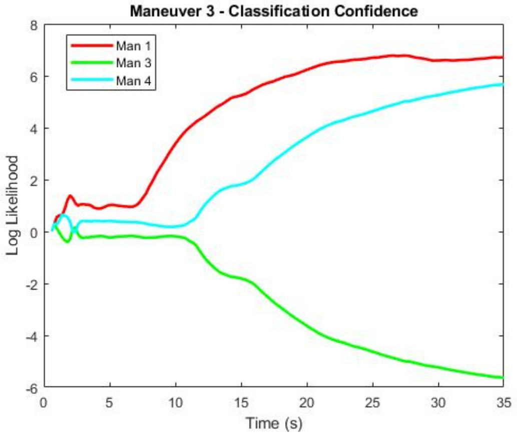
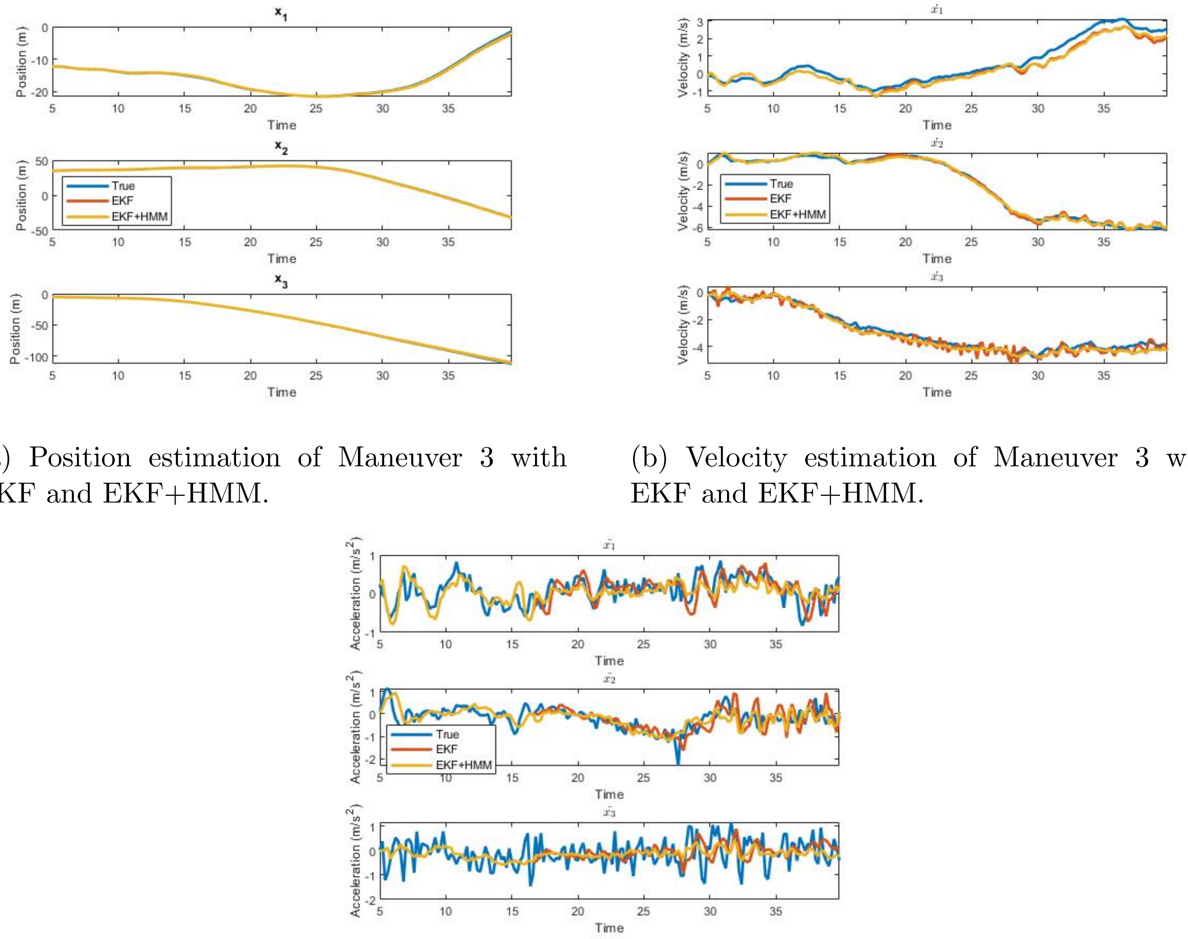


Figure 5.73: Maneuver 3: Confidence of classification over time.

The UAV states were estimated with the EKF and the EKF+HMM, as seen in Figure 5.74. Both filters have high error when estimating the velocity components v_1 and v_3. The acceleration estimates of the EKF and EKF+HMM are similar, but the EKF+HMM attenuates more changes in acceleration than the EKF.



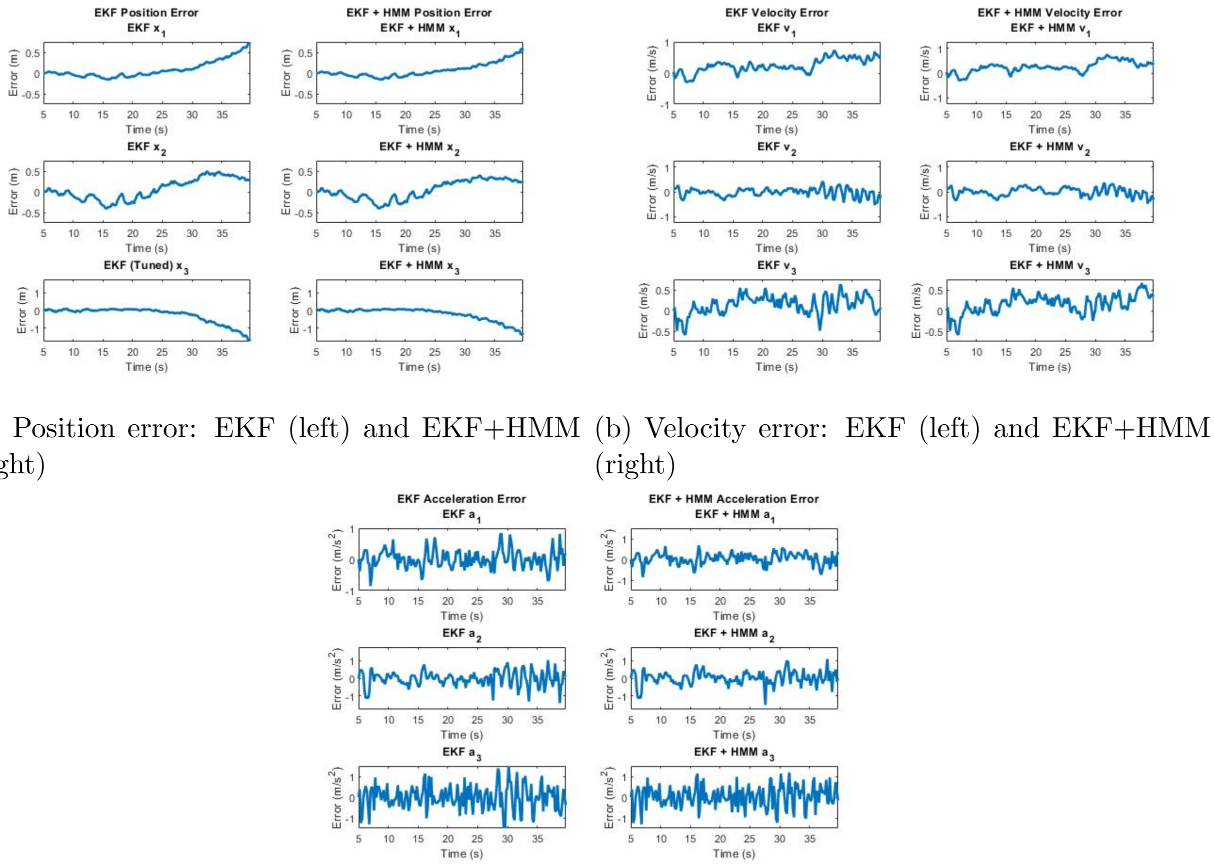
(a) Position estimation of Maneuver 3 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 3 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 3 with EKF and EKF+HMM.

Figure 5.74: Maneuver 3 UAV state estimation - Scenario 2A.

The comparison of EKF and EKF+HMM errors (Figure 5.75) shows similar errors for both filters. There is high error in position and velocity estimates, while acceleration estimate errors are noisy but near zero-mean. The EKF+HMM acceleration estimate error is less noisy than that of the EKF. Figure 5.76 shows the variance of each state for the EKF+HMM.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.75: Comparison of state errors for EKF and EKF+HMM.

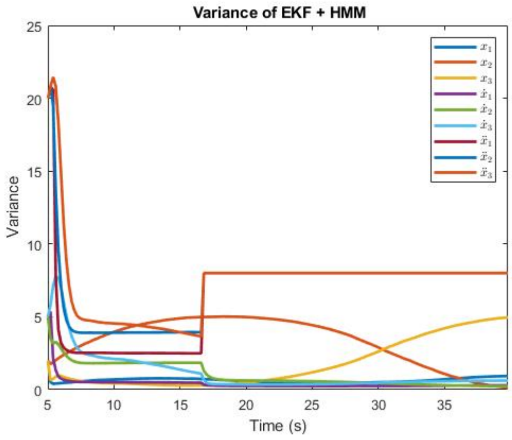


Figure 5.76: Variance of EKF+HMM states over time

Scenario 2B - Simulation Data Trained HMM

The confidence over time when classifying this maneuver with simulation trained HMMs is shown in Figure 5.77. The classification accuracy over time of the trajectory was 51.07%, which is less than that of the experimental data trained HMM.

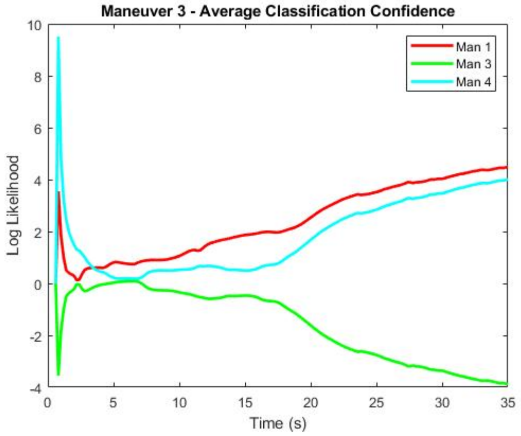
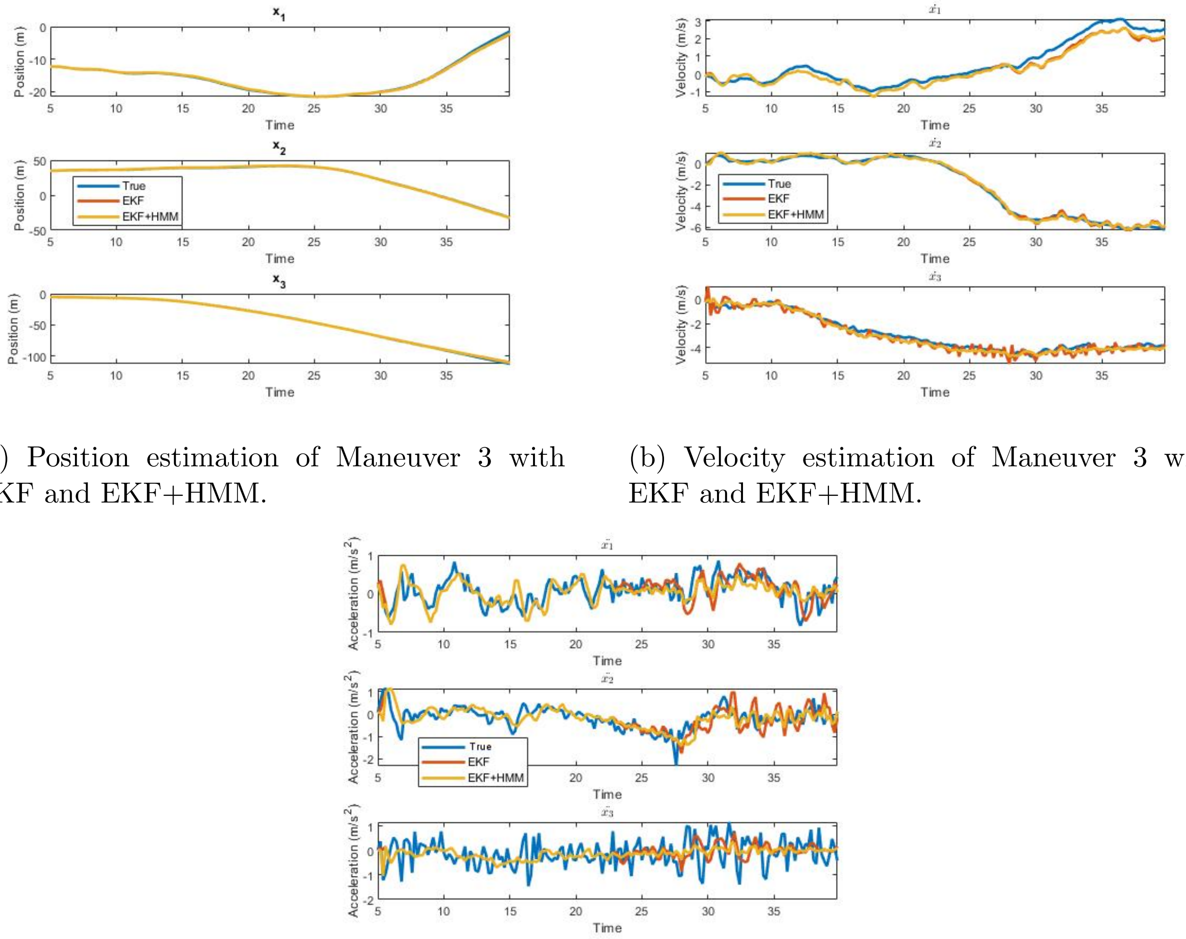


Figure 5.77: Maneuver 3: Confidence of classification over time using simulation trained HMM.

The EKF and EKF+HMM estimation results are shown in Figure 5.78. Like in Scenario 2A, both the EKF and EKF+HMM have high error in estimating v_1 and v_3. The EKF+HMM results of Scenario 2B is similar to those of 2A except that the incorporation of HMM estimates begins later in Scenario 2B.



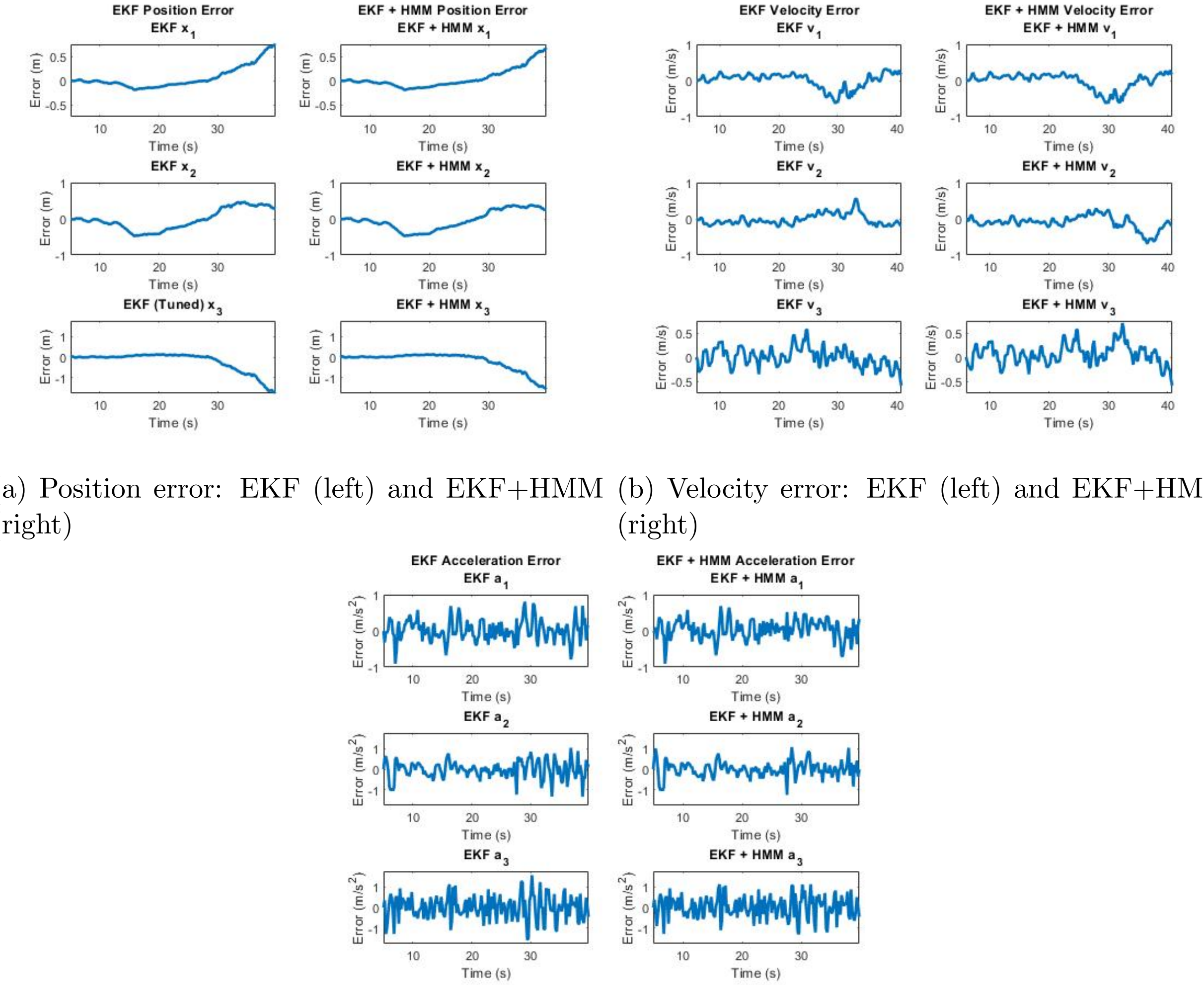
(a) Position estimation of Maneuver 3 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 3 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 3 with EKF and EKF+HMM.

Figure 5.78: Maneuver 3 UAV state estimation using simulation trained HMM in EKF+HMM.

The EKF and EKF+HMM estimation results are further shown in their state errors over time (Figure 5.79). The EKF+HMM has slightly less noisy acceleration estimate error, while having slightly higher or similar velocity error. The state variances of the EKF+HMM are displayed in Figure 5.80.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.79: Maneuver 3 UAV state estimation - Scenario 2B.

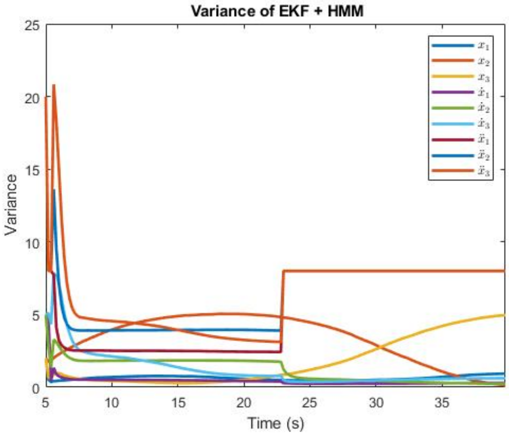


Figure 5.80: Variance of EKF+HMM states over time

5.4.3 Maneuver 4

Scenario 1A - Experimental Data Trained HMM

The results for the Maneuver 4 trajectory that produces the best EKF+HMM estimates (when using an experimental data trained HMM) are shown in this section. Figure 5.81 depicts the confidence of each maneuver's HMM throughout UAV flight. The maneuver had a classification accuracy over time of 82%.

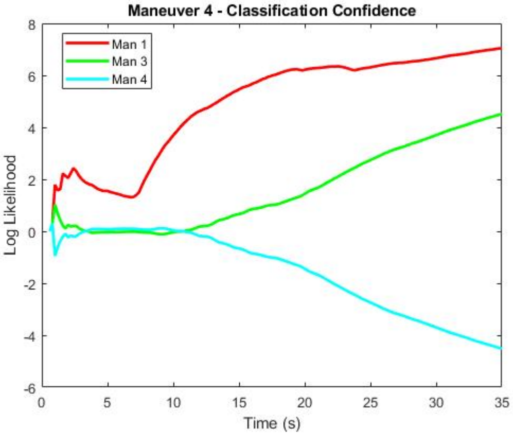
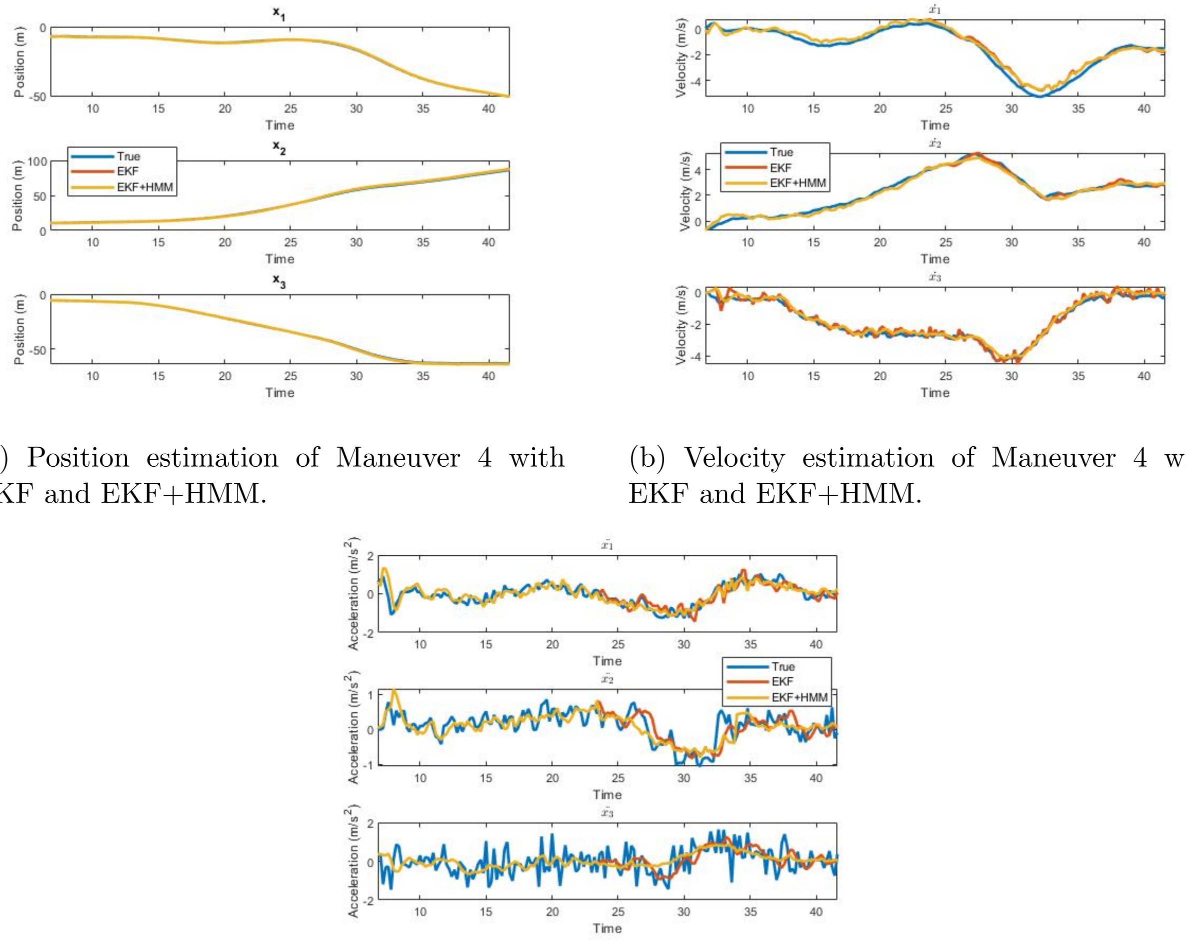


Figure 5.81: Maneuver 4: Confidence of classification over time.

Figure 5.82 shows UAV state estimation using the EKF and EKF+HMM. Both filters have lag when estimating the velocity states. However, the EKF+HMM more clearly differs from the EKF in estimating the acceleration states. The EKF follows many of the disturbances seen in the acceleration reference states, while the EKF+HMM filters most out.



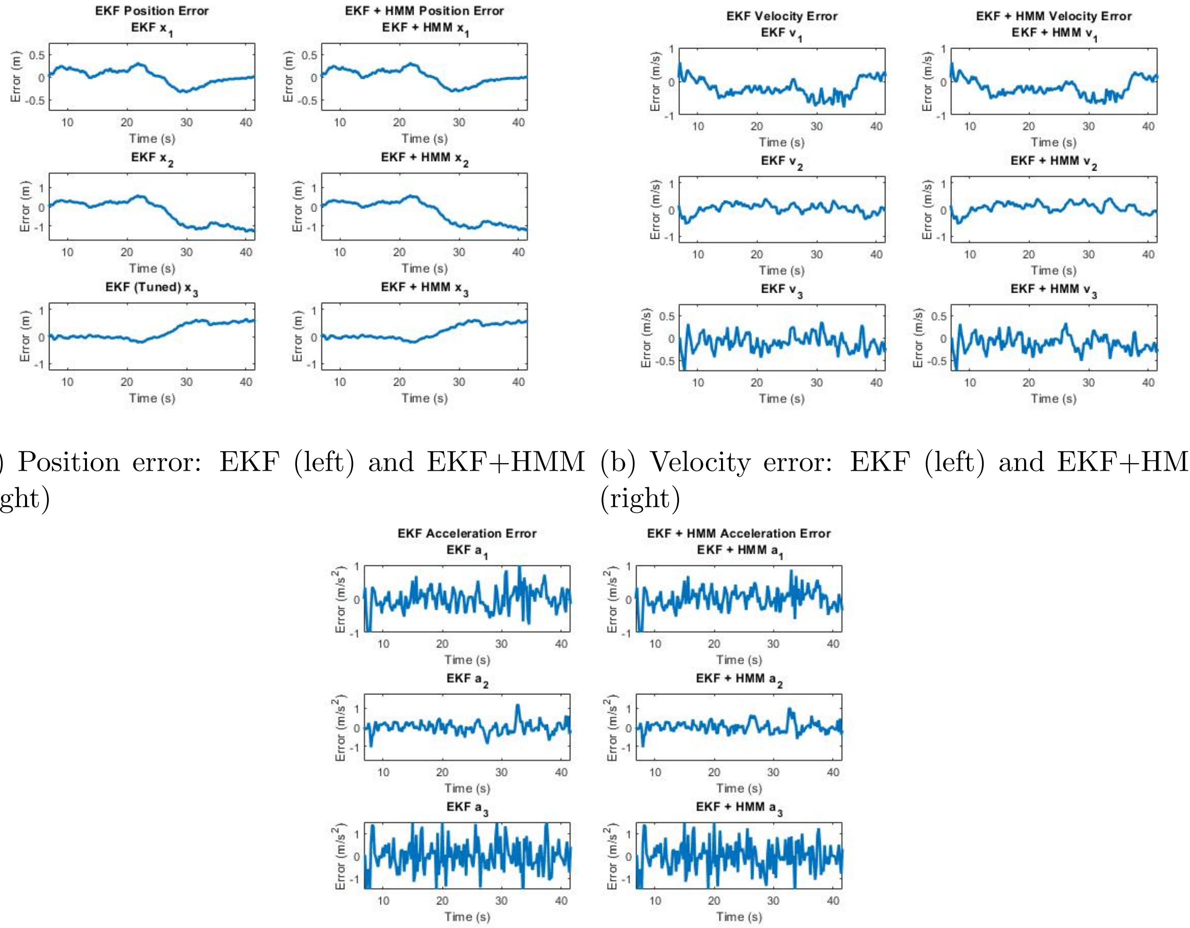
(a) Position estimation of Maneuver 4 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 4 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 4 with EKF and EKF+HMM.

Figure 5.82: Maneuver 4 UAV state estimation - Scenario 1A.

The estimation error of the EKF and EKF+HMM are compared in Figure 5.83. For the most part, the EKF and EKF+HMM errors are almost indistinguishable. However, some slight improvements in estimating a_1 can be seen towards the end of the maneuver when using the EKF+HMM. The EKF+HMM state variances are shown in Figure 5.84.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.83: Comparison of state errors for EKF and EKF+HMM.

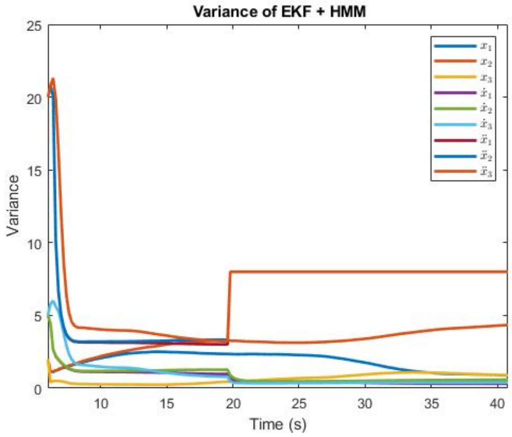


Figure 5.84: Variance of EKF+HMM states over time

Scenario 1B - Simulation Data Trained HMM

The same trajectory was then estimated using an EKF+HMM that utilizes simulation data trained HMMs. The maneuver was correctly classified as Maneuver 4 with a classification accuracy over time of 48.74%. The confidence of each HMM over time is shown in Figure 5.85.

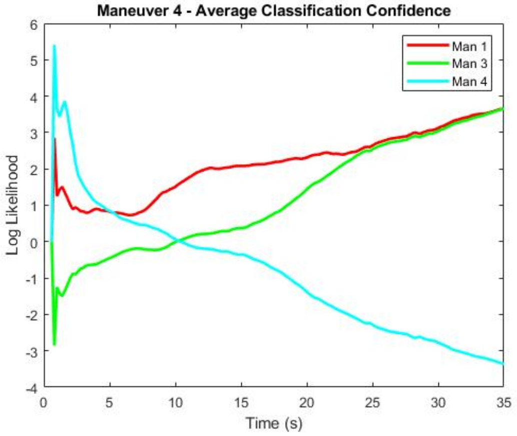
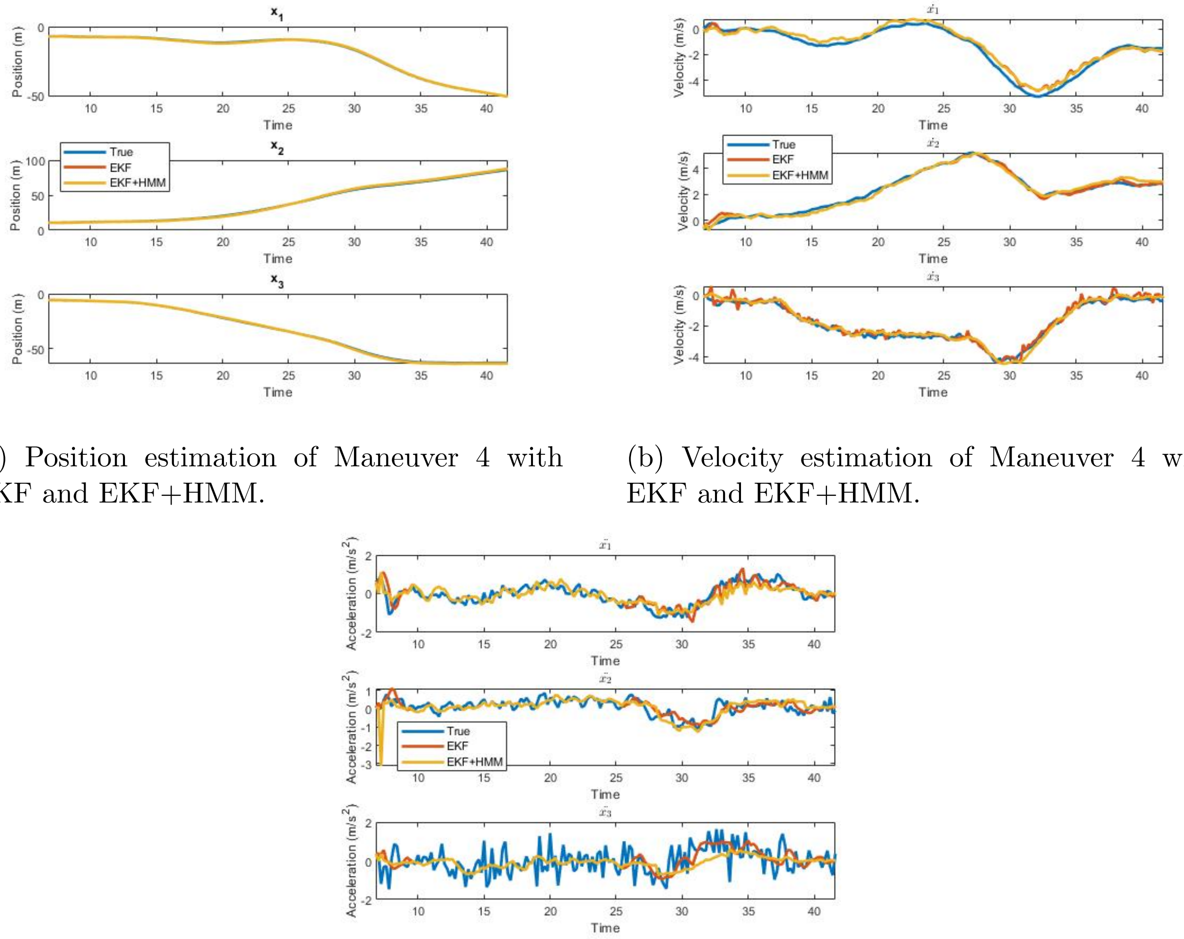


Figure 5.85: Maneuver 4: Confidence of classification over time using simulation trained HMM.

Figure 5.86 shows the EKF and EKF+HMM state estimates throughout the maneuver. HMM estimates are not used in the EKF+HMM until $t = 25$ seconds. When the HMM estimates are used, v_3 and a_3 EKF+HMM estimates become more offset than that of the EKF, while a_2 is more able to follow the reference.



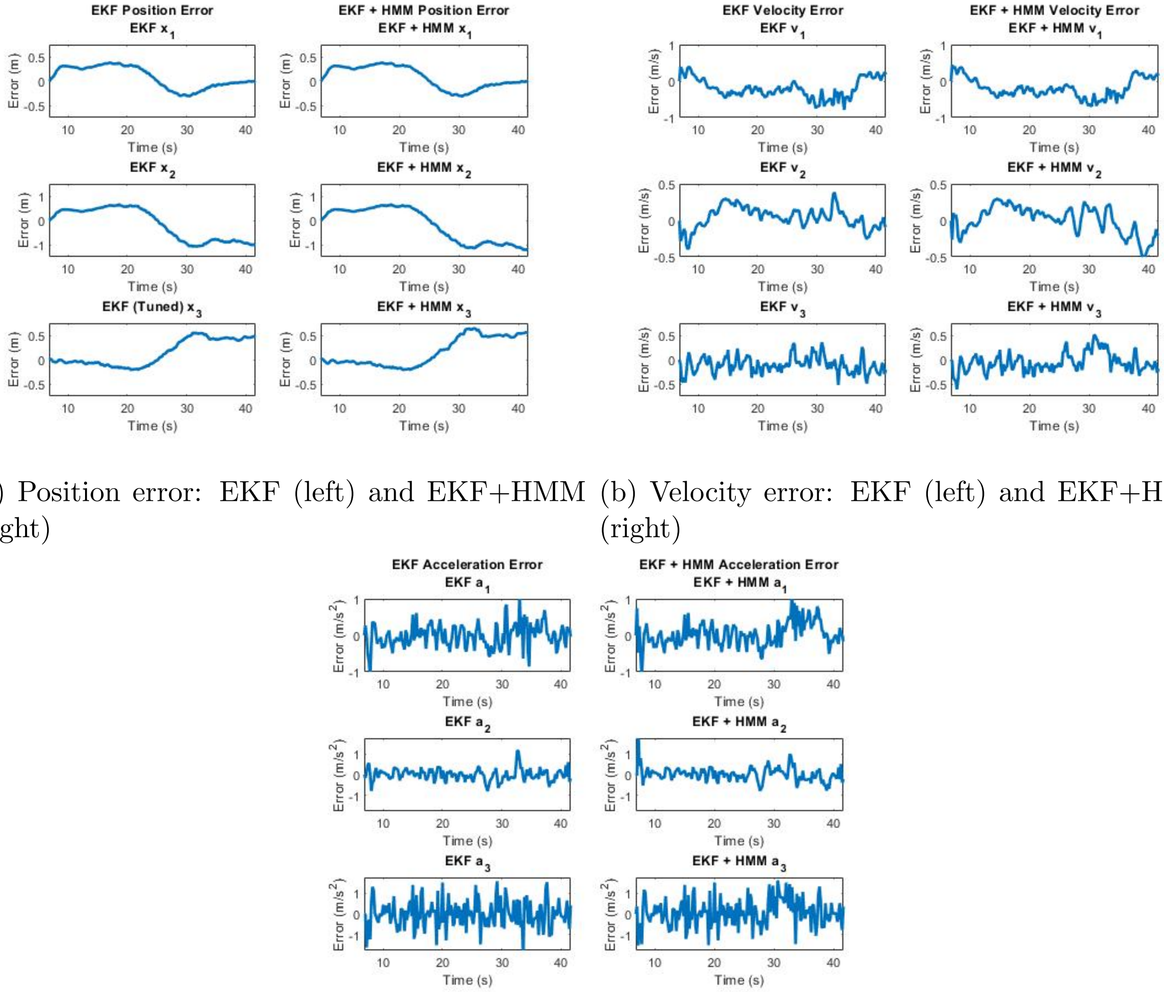
(a) Position estimation of Maneuver 4 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 4 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 4 with EKF and EKF+HMM.

Figure 5.86: Maneuver 4 UAV state estimation - Scenario 1B.

The state error of the EKF and EKF+HMM are shown in Figure 5.87. Position estimate errors are similar between the two filters. However, the EKF+HMM performs worse than the EKF for all velocity and acceleration estimates except v_1 and a_2. Note that this EKF+HMM is using estimates generated from a simulation trained HMM. The state variances of the EKF+HMM are shown in Figure 5.88.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.87: Comparison of state errors for EKF and EKF+HMM using simulation trained HMM.

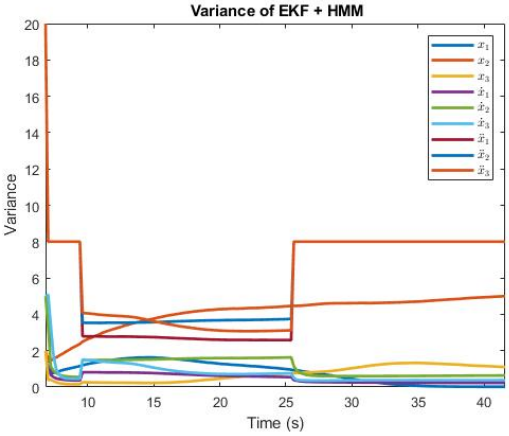


Figure 5.88: Variance of EKF+HMM states over time

Scenario 2A - Experimental Data Trained HMM

This trajectory of Maneuver 4 had a classification accuracy over time of 83.33%. The confidence of each HMM during this classification is shown in Figure 5.89. EKF and EKF+HMM estimation of the UAV states throughout this maneuver is shown in Figure 5.90. Both the EKF and EKF+HMM show ineffective estimation of velocity state. When estimating acceleration, the filters produce similar results for a_1 and a_2. When estimating a_3, the EKF+HMM underestimates the state while the EKF does not.

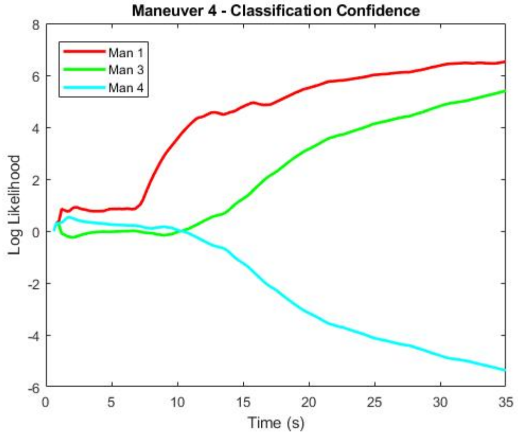
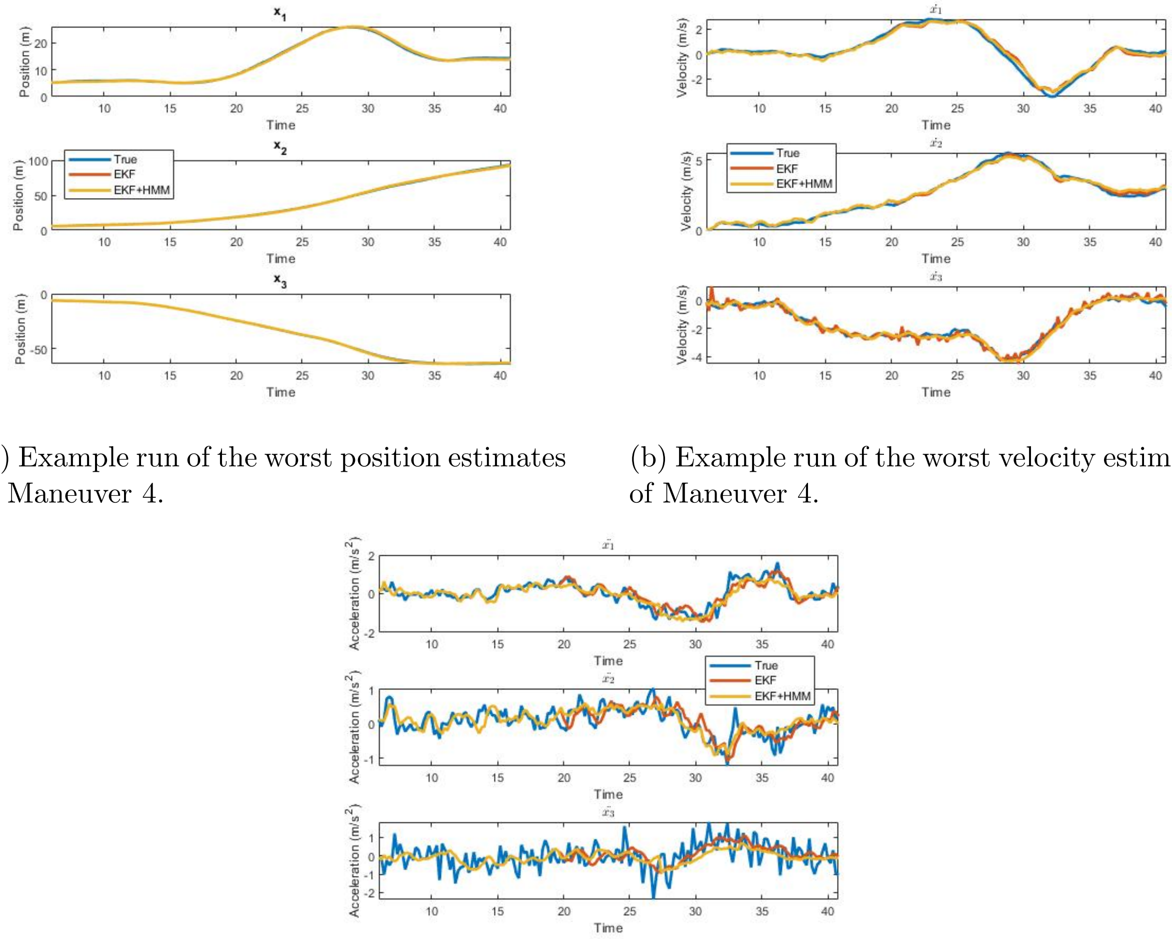


Figure 5.89: Maneuver 4: Confidence of classification over time.



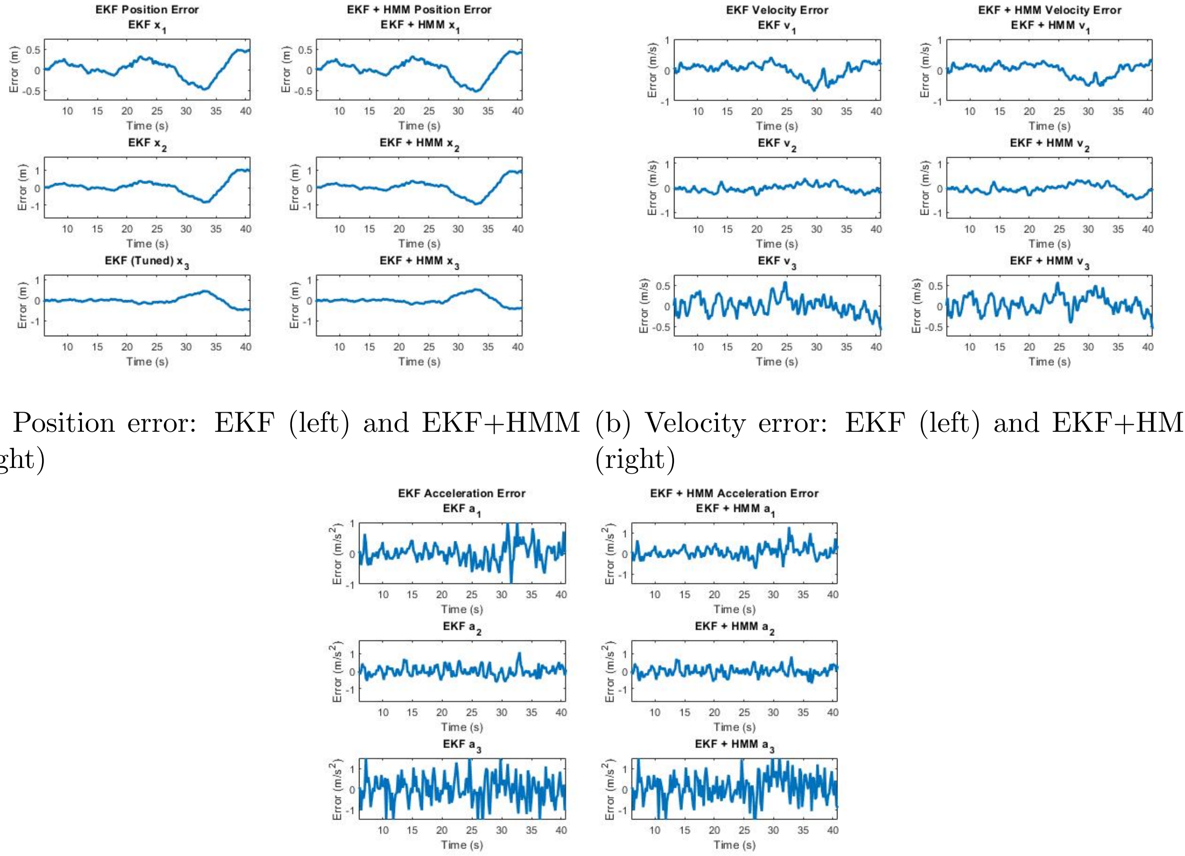
(a) Example run of the worst position estimates of Maneuver 4.

(b) Example run of the worst velocity estimates of Maneuver 4.

(c) Example run of the worst acceleration estimates of Maneuver 4.

Figure 5.90: Maneuver 4 UAV state estimation - Scenario 2A.

The state estimation errors of the EKF and EKF+HMM are displayed in Figure 5.91. The position and velocity estimates of the filters remain near identical. Acceleration estimates differ slightly, particularly when estimating a_1 and a_3. The EKF+HMM has lower estimation error for a_1 than the EKF. However, EKF+HMM a_3 error is increased for the final 10 seconds of the maneuver. The variance of the EKF+HMM states over time is shown in Figure 5.92.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.91: Comparison of state errors for EKF and EKF+HMM.

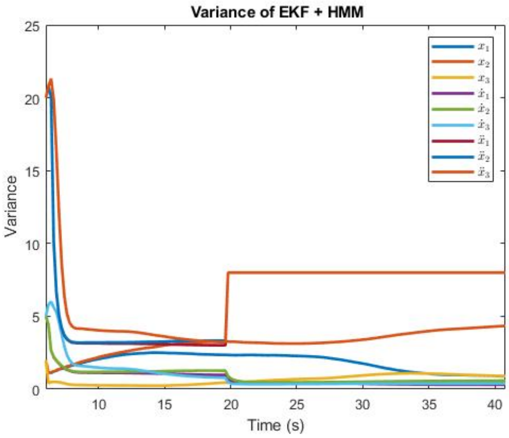


Figure 5.92: Variance of EKF+HMM states over time

Scenario 2B - Simulation Data Trained HMM

The same trajectory as Scenario 2A was estimated using an EKF+HMM with simulation trained HMMs. The classification accuracy over time of the trajectory was 40%, and the confidence over time of each model is seen in Figure 5.93. Note that there is an initial misclassification of the trajectory as Maneuver 3.

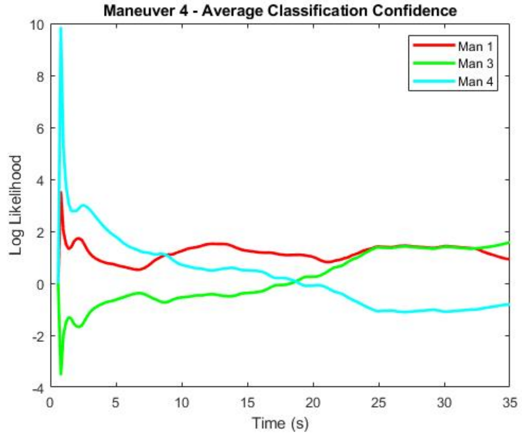
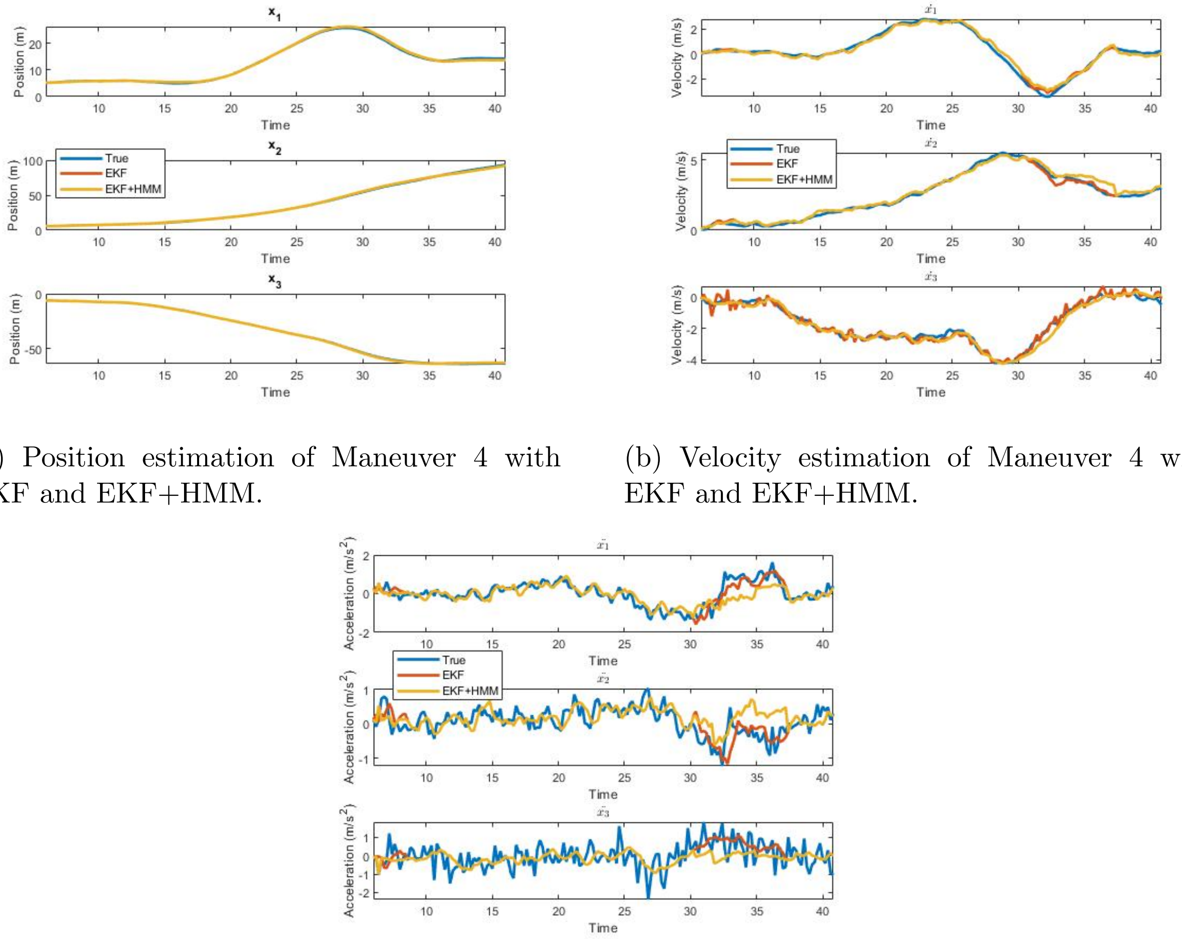


Figure 5.93: Maneuver 4: Confidence of classification over time using simulation trained HMM.

EKF and EKF+HMM estimation of the maneuver is shown in Figure 5.94. The EKF+HMM performs considerably worse than the EKF for all states. From thirty to thirty-five seconds, the EKF+HMM estimates are all offset from the reference. This indicates erroneous HMM estimates are worsening the EKF+HMM's ability to estimate the trajectory.



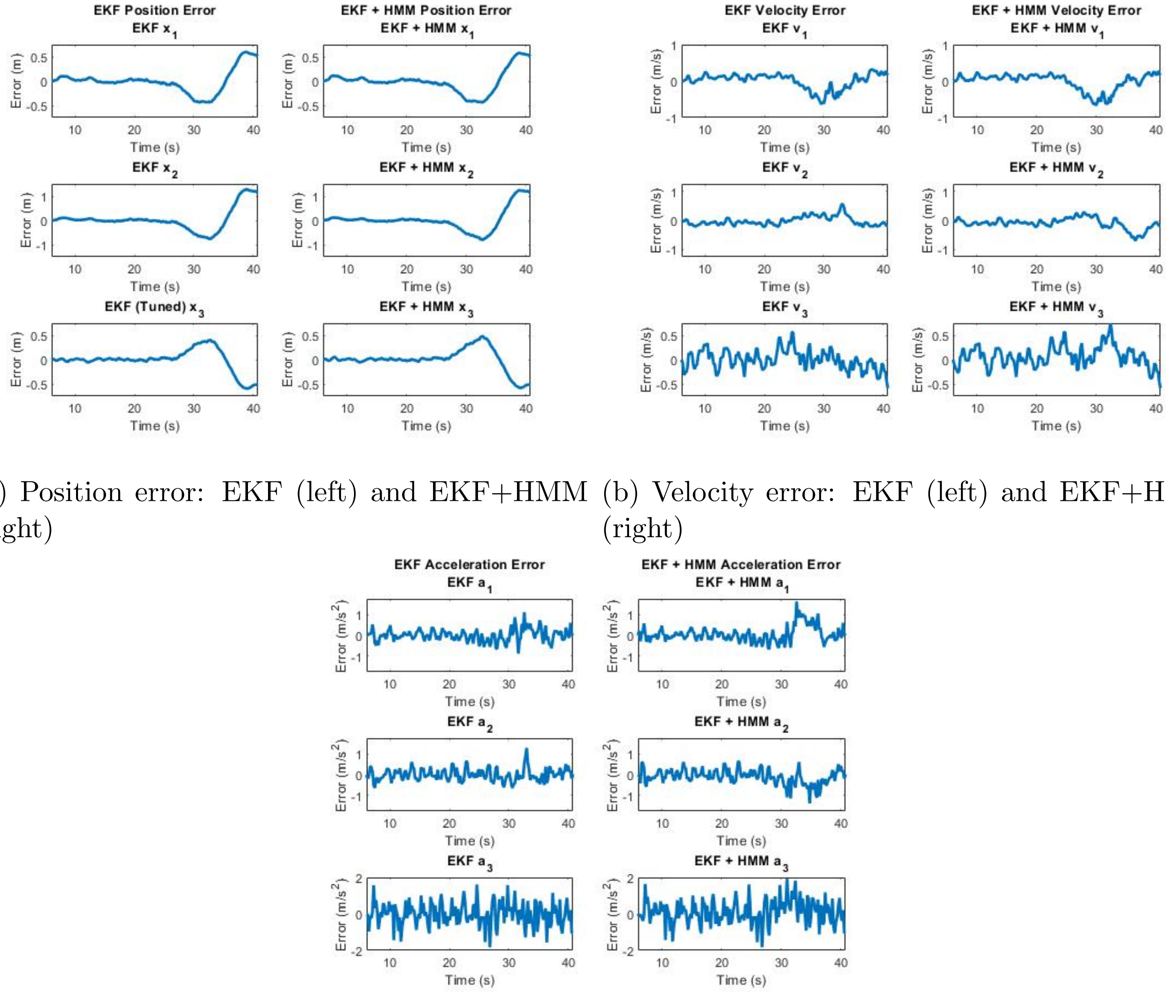
(a) Position estimation of Maneuver 4 with EKF and EKF+HMM.

(b) Velocity estimation of Maneuver 4 with EKF and EKF+HMM.

(c) Acceleration estimation of Maneuver 4 with EKF and EKF+HMM.

Figure 5.94: Maneuver 4 UAV state estimation - Scenario 2B.

Figure 5.95 shows the state errors of the EKF and EKF+HMM. Towards the end of the trajectory when the EKF+HMM begins using HMM estimates, the EKF+HMM errors become much greater than that of the EKF for velocity and acceleration. This was not the case in Scenario 2A – suggesting that these errors are the result of incorrect acceleration and jerk estimates from the simulation trained HMM only. Figure 5.96 shows the state variance over time of the EKF+HMM.



(a) Position error: EKF (left) and EKF+HMM (right)

(b) Velocity error: EKF (left) and EKF+HMM (right)

(c) Acceleration error: EKF (left) and EKF+HMM (right)

Figure 5.95: Comparison of state errors for EKF and EKF+HMM using simulation trained HMM.

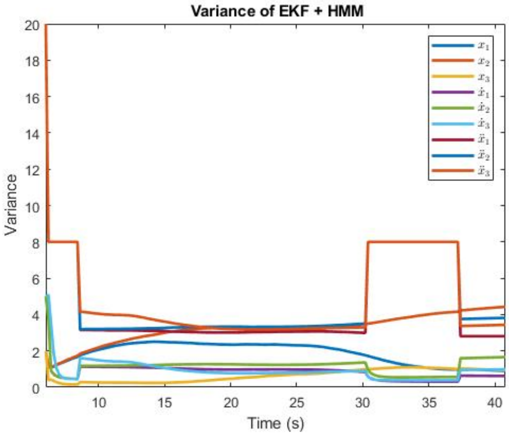


Figure 5.96: Variance of EKF+HMM states over time

5.5 Conclusion

An octacopter was used to perform four maneuvers and create an experimental data set of 15 maneuvers each. Despite having a much smaller training data set than used in simulation (10 trajectories versus 1700), HMMs were trained for 3 of the 4 maneuvers using experimental data. One maneuver, Maneuver 2, was not modelled and instead used as an unknown. The maneuvers were also modelled in simulation. The simulated data set was used to train another set of HMMs. The experimental data trained HMMs and simulation data trained HMMs were compared.

For classification, both sets of HMMs were capable of classifying real-world reference data and estimates of the reference data. The experimental data trained HMMs had superior final classification accuracy and mean accuracy over time for known maneuvers, but the simulation data trained HMMs maintained adequate classification accuracy results. Overall, both experimental data trained and simulation data trained HMMs proved to be viable options for classification of real world maneuvers.

Both types of maneuver HMMs were also used to generate HMM estimates of acceleration and jerk. These estimates were capable of filtering process noise out and finding the general trend of the reference data. The estimation performance of the experimental data trained HMMs was found to be superior to that of the simulation trained HMMs.

Then, an EKF+HMM estimated UAV states as the vehicle performed each maneuver – once using the experimental data trained HMM and once using the simulation data trained HMM. These estimates were compared to a standard EKF. For the experimental data trained HMM, the EKF+HMM generally did not make state estimates worse than the EKF, but it also did not make them better. When using the EKF+HMM with simulation data trained HMMs, the EKF+HMM performed at the same level as the EKF or worse than the EKF. Often estimates from the simulation trained HMM decreased the accuracy of the EKF+HMM. These results suggest that future work should focus primarily on the EKF+HMM using HMMs that have been trained with experimental data.

Because the reference data was created from less than perfect measurements, it is difficult to truly gauge the effectiveness of the EKF+HMM filter when compared to a standard EKF. Both experienced lag when estimating UAV states and had difficulty producing zero-mean estimation error. The greatest difference in the two filters is that the EKF+HMM often attenuated process noise and disturbances in the acceleration states, while the EKF was more likely to attempt to estimate these aspects of the acceleration. Future work should focus on creating better experimental reference data to better compare the EKF and EKF+HMM. Furthermore, performing more dynamic maneuvers with the octacopter could create a dataset where a standard EKF is insufficient, allowing for greater ability to view the improvements or failures of the EKF+HMM by comparison.

Chapter 6

Conclusion

This thesis presented a system that utilizes HMM estimates to supplement EKF estimation during a maneuver. Chapter 2 describes UAV modelling and simulation and establishes a baseline EKF to estimate the linear position, velocity, and acceleration of the UAV. The EKF was evaluated when estimating several UAV maneuvers. Chapter 3 explains how UAV maneuvers can be modelled as HMMs and describes the additional information HMMs can provide through classification and the generation of HMM estimates. Chapter 4 provides the architecture of the EKF+HMM and uses simulation to evaluate the EKF+HMM when compared to the baseline EKF. Finally, Chapter 5 implements the EKF, HMMs, and EKF+HMM on real world data and compares the effectiveness of an experimental data training set and a simulation data training set.

HMMs were shown to be an effective way to model UAV maneuvers. In both simulation and with real-world data, HMMs were capable of being trained using true UAV states (even when there was minimal data in the case of the real world maneuvers). HMMs were then utilized for classification by determining the maneuver model most likely to produce a series of UAV states and finding the confidence measure of the classification. HMM classification remained accurate even when using UAV state estimates as observations rather than true states. Classification was also accurate when HMMs were trained on simulated data, but tested on real world data.

HMMs also provided estimates of the UAV acceleration and jerk during a maneuver. This process involves finding the HMM state path and emission probability, and then performing a Gaussian Mixture Regression with the UAV position. HMM estimates did have some error, but overall they were able to follow the non-linear and sometimes non-differentiable true acceleration and jerk.

The HMM was then combined with an EKF to create the EKF+HMM. This system classifies maneuvers using EKF estimates of UAV states and the HMM confidence measure classifier. The HMM of the classified maneuver was then used to generate acceleration and jerk estimates that were input into the dynamic model of the EKF+HMM. The EKF+HMM aimed to create a better predictive model than the constant acceleration model used in the baseline EKF.

In simulation, the EKF+HMM was capable of improving UAV state estimation during high dynamic maneuvers when compared to the EKF. Although some error is introduced in the EKF+HMM through HMM estimates, these errors are much less than that of the EKF. During a maneuver with lower dynamics (Maneuver 4), the EKF is capable of more accurate estimation. For this maneuver, the EKF+HMM performed similar to the EKF and at times worse. Errors created from HMM estimates were greater than the EKF errors and caused degraded performance of the EKF. Overall, the EKF+HMM is an effective tool in estimating high dynamic maneuvers, but it is limited by the quality of the maneuver HMM parameters.

When using experimental data, the EKF+HMM was tested using HMMs trained with experimental data and HMMs trained with simulation data. When using experimental data trained HMMs, the EKF+HMM typically performed on par with the EKF. The EKF was more likely to estimate disturbances and noise – particularly when estimating acceleration. In contrast, the EKF+HMM often attenuated these disturbances. When simulation data trained HMMs were used in the EKF+HMM, the EKF+HMM tended to perform at the same level or worse than the EKF. Because the experimental data was quite noisy, UAV reference states are not necessarily the most accurate states. Therefore, it is difficult to evaluate which approach was more effective. Future experimentation should focus on ensuring accurate and precise UAV reference data.

6.1 Future Work

There are several interesting directions this work could be taken in in the future. A few possibilities are listed below.

- The most concrete need is better experimental data of the UAV maneuvers through improved sensors and experimentation techniques. Better reference data could provide a more comprehensive comparison between the EKF and EKF+HMM when estimating real-world data.

- Real world performance of more dynamic maneuvers would provide additional opportunities to assess the effectiveness of the EKF+HMM. In cases with highly variable acceleration and jerk, the EKF+HMM is expected to perform better than a constant acceleration EKF. Using a more maneuverable UAV for experimental data collection and designing more complex maneuvers would create a better data set for EKF+HMM evaluation.

- The EKF+HMM could also be improved through a process noise matrix that adapts to the quality of HMM estimates. Creation of a system to better predict whether or not an HMM estimate is valid and incorporating that information into the EKF+HMM could greatly increase the accuracy of the EKF+HMM.

- UAV modelling and estimation is a fairly structured problem. HMMs could be evaluated for systems with less structure and incorporated into something like a particle filter. An interesting application that comes to mind is navigation with a magnetometer.

- The HMM classification system already identifies unknown maneuvers. An interesting next step would be to save the data of each unknown maneuver, group like maneuvers together, and attempt to create HMMs of unknown maneuvers when “in the field.”

K-fold cross validation could be used to evaluate the number of trajectories needed to train a robust HMM for an unknown trajectory.

- Currently, HMM estimates are generated using position estimates of the maneuver. Potential future work could explore generating estimates about time in order to aid in trajectory prediction, as seen in [21].

Bibliography

[1] Is there a way to fit a 3d gaussian distribution or a gaussian mixture distribution to a vector?, 2016.

[2] Uas by the numbers, 2021.

[3] Man smuggled 13 pounds of meth from mexico using a drone. *NBC News*, 2017.

[4] C. Phillips and C. Gaffey. Most french nuclear plants 'should be shut down' over drone threat. *Newsweek*, 2015.

[5] G. Birch, J. C. Griffin, and M. K. Erdman. Uas detection classification and neutralization: Market survey 2015. 2015.

[6] D. Kingston and R. Beard. Real-time attitude and position estimation for small uavs using low cost sensors. In *AIAA Unmanned Unlimited Technical Conference, Workshop and Exhibit*, Chicago, Illinois, Sept 2004. ACM Press.

[7] P. Marantos, Y. Koveos, and K. J. Kyriakopoulos. Uav state estimation using adaptive complementary filters. *IEEE Transactions on Control Systems Technology*, 24(4):1214–1226, 2016.

[8] F. Hoffmann, M. Ritchie, F. Fioranelli, A. Charlish, and H. Griffiths. Micro-doppler based detection and tracking of uavs with multistatic radar. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6, 2016.

[9] J. Martin, D. Hall, and J. L. Crowley. Statistical gesture recognition through modelling of parameter trajectories. In Annelies Braffort, Rachid Gherbi, Sylvie Gibet, Daniel Teil, and James Richardson, editors, *Gesture-Based Communication in Human-Computer Interaction*, pages 129–140, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[10] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50:203–226, 2014.

[11] Trevor J. Darrell and lex P. Pentland. Recognition of space-time gestures using a distributed representation. Technical report.

[12] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[13] Thad E. Starner. Visual recognition of american sign language using hidden markov models. CRS Report No. RL33539, 1995.

[14] F. Bashir, A. Khokhar, and D. Schonfeld. Automatic object trajectory-based motion recognition using gaussian mixture models. In *2005 IEEE International Conference on Multimedia and Expo*, pages 1532–1535, 2005.

[15] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Transactions on Image Processing*, 16(7):1912–1919, 2007.

[16] Seungwon Choi, Suseong Kim, and Kim H. Jin. Inverse reinforcement learning control for trajectory tracking of a multirotor uav. *International Journal of Control, Automation and Systems*, 15:1826–1834, 2017.

[17] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[18] M. Schneider and W. Ertel. Robot learning by demonstration with local gaussian process regression. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 255–260, 2010.

[19] Sylvain Calinon and Aude Billard. Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 105–112, New York, NY, USA, 2005. Association for Computing Machinery.

[20] S. Calinon, F. Guenter, and A. Billard. On learning the statistical representation of a task and generalizing it to various contexts. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2978–2983, 2006.

[21] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007.

[22] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors, 2006.

[23] Tommaso Bresciani. Modelling, identification, and control of a quadrotor helicopter. Master's thesis, Lund University Department of Automatic Control, Lund Sweden, 2008.

[24] Teppo Luukkonen. Modelling and control of quadcopter. Technical report, Espoo Finland, 2011.

[25] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *2009 IEEE International Conference on Robotics and Automation*, pages 3277–3282, 2009.

[26] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment.*

[27] What speed do fpv racing drones fly? Available at `http://dronesuavreport.com/2018/09/21/what-speed-do-fpv-racing-drones-fly/`, 2018.

[28] Michael E. Hough. Reentry maneuver estimation using nonlinear markov acceleration models. *Journal of Guidance, Control, and Dynamics*, 40(7):1693–1710, 2017.

[29] R. J. Fitzgerald. Effects of range-doppler coupling on chirp radar tracking accuracy. *IEEE Transactions on Aerospace and Electronic Systems*, AES-10(4):528–532, 1974.

[30] M. A. Richards. *Principles of Modern Radar*, volume 1. Scitech Publishing, Edison NJ, 2010.

[31] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012.

[32] KEI-YUN CALVIN YU and DENNIS L. BRICKER. Analysis of a markov chain model of a multistage manufacturing system with inspection, rejection, and rework. *IIE Transactions*, 25(1):109–112, 1993.

[33] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[34] N. Dimitrova, L. Agnihotri, and G. Wei. Video classification based on hmm using text and faces. In *2000 10th European Signal Processing Conference*, pages 1–4, 2000.

[35] A. V. Nefian, L. Liang, X. Pi, L. Xiaoxiang, C. Mao, and K. Murphy. A coupled hmm for audio-visual speech recognition. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–2013–II–2016, 2002.

[36] Yves Boussemart and Mary Cummings. Predictive models of human supervisory control behavioral patterns using hidden semi-markov models. *Eng. Appl. of AI*, 24:1252–1262, 10 2011.

[37] Mario Stanke and Stephan Waack. Gene prediction with a hidden markov model and a new intron submodel. *Bioinformatics*, 19:215–225, 09 2003.

[38] B. H. Juang and L. R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[39] B. L. Van, S. Garcia-Salicetti, and B. Dorizzi. On using the viterbi path along with hmm likelihood information for online signature verification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1237–1247, 2007.

[40] M. R. Hassan and B. Nath. Stock market forecasting using hidden markov model: a new approach. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pages 192–196, 2005.

[41] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell System Technical Journal*, 62(4):1035–1074, 1983.

[42] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[43] L. Liporace. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Transactions on Information Theory*, 28(5):729–734, 1982.

[44] B. Juang. Maximum-likelihood estimation for mixture multivariate stochastic observations of markov chains. *AT T Technical Journal*, 64(6):1235–1249, 1985.

[45] Bing-Hwang Juang, S. Levinson, and M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.). *IEEE Transactions on Information Theory*, 32(2):307–309, 1986.

[46] Leonard E. Baum and George R. Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2):211 – 227, 1968.

[47] J. Baker. The dragon system–an overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):24–29, 1975.

[48] Samuel Douglass. Deep Learned Multi-Modal Traffic Agent Predictions for Truck Platooning Cut-Ins. Master's thesis, Auburn University, Auburn AL, 2020.

[49] O. Mazhelis. One-class classifiers : a review and analysis of suitability in the context of mobile-masquerader detection. *South African Computer Journal*, 2006(36), 2006.

[50] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: alternative data models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344)*, pages 133–145, 1999.

[51] Sung-Bae Cho and Hyuk-Jang Park. Efficient anomaly detection by modeling privilege flows using hidden markov model. *Computers and Security*, 22(1):45–55, 2003.

[52] Dit-Yan Yeung and Yuxin Ding. Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition*, 36(1):229–243, 2003.

[53] C. Lee and Yangsheng Xu. Online, interactive learning of gestures for human/robot interfaces. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 2982–2987 vol.4, 1996.

[54] J.M. Bibby K. V. Mardia, J. T. Kent. *Multivariate Analysis*. Academic Press, London, 1979.

[55] Hsi Guang Sung. *Gaussian mixture regression and classification*. PhD thesis, Rice University, Houston, TX, 2004.

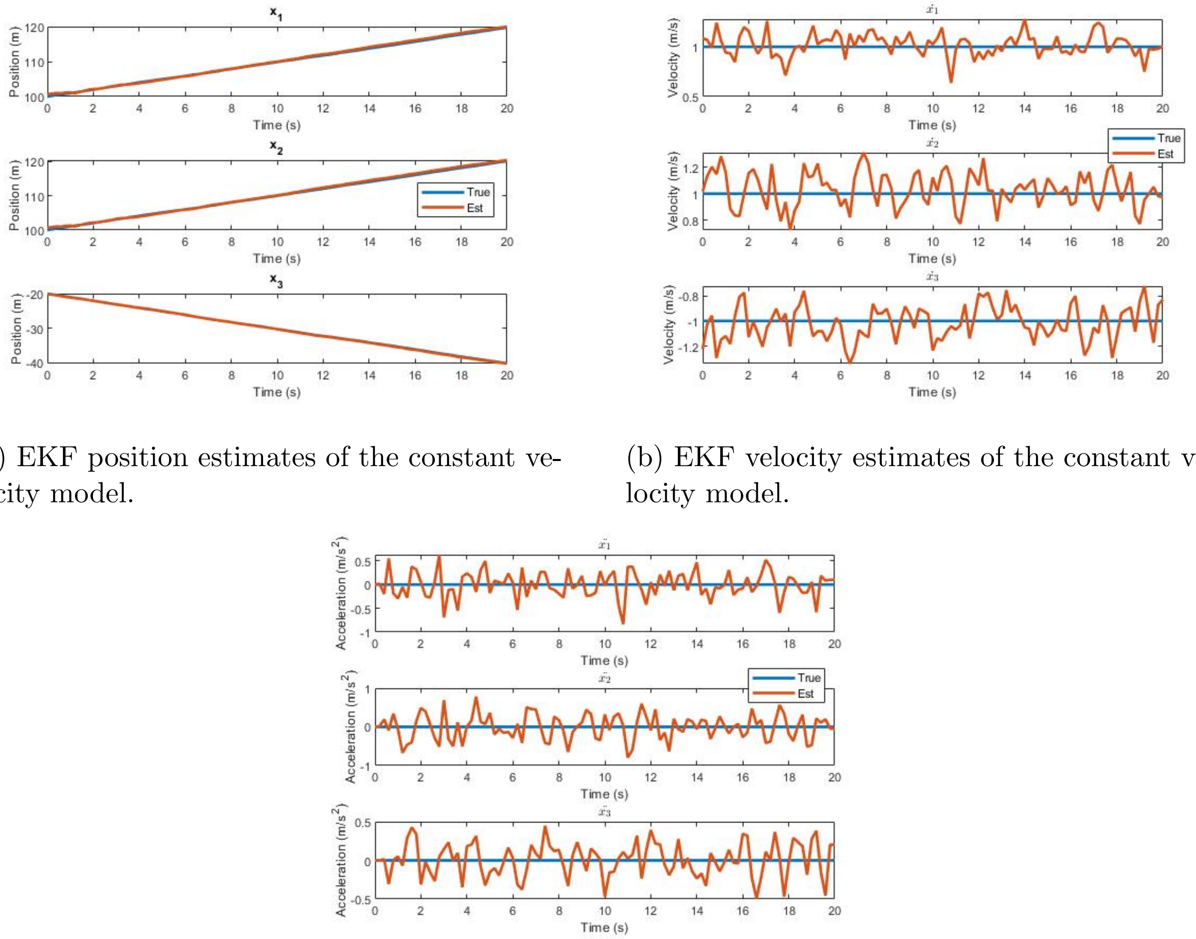
Appendices

Appendix A

Basic EKF Results

The following chapter provides initial results of the basic EKF used in this thesis, as seen in Section 2.3. A constant velocity (A.1), constant acceleration (A.2), and constant jerk (A.3) model were used to simulate a vehicle. Then, the vehicle's linear position, velocity, and acceleration is estimated with the EKF. A Monte Carlo simulation was performed for each scenario. The resulting mean error and error variance were found.

A.1 Constant Velocity Model

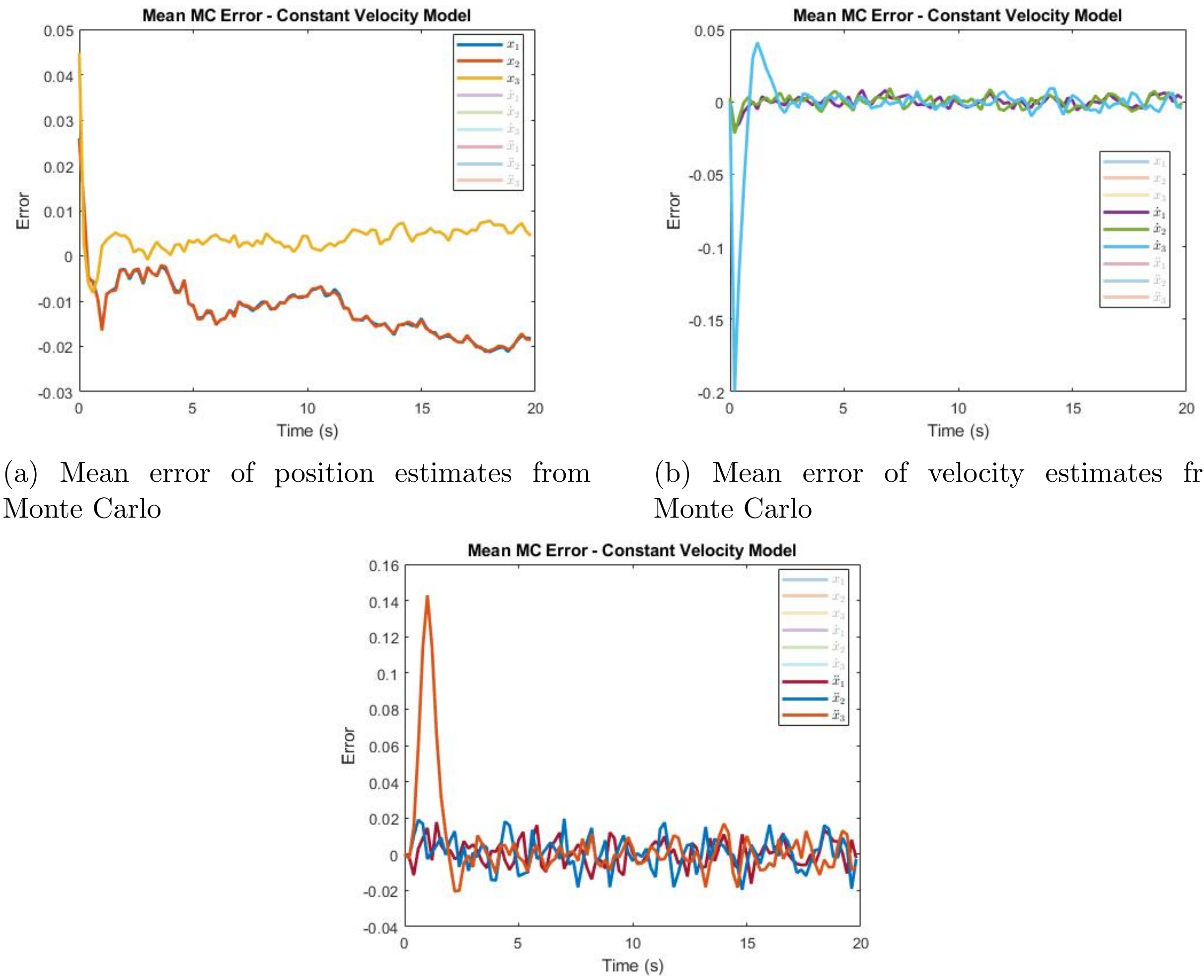


(a) EKF position estimates of the constant velocity model.

(b) EKF velocity estimates of the constant velocity model.

(c) EKF acceleration estimates of the constant velocity model.

Figure A.1: An example of the EKF estimating the states of a vehicle with a constant velocity dynamic model.

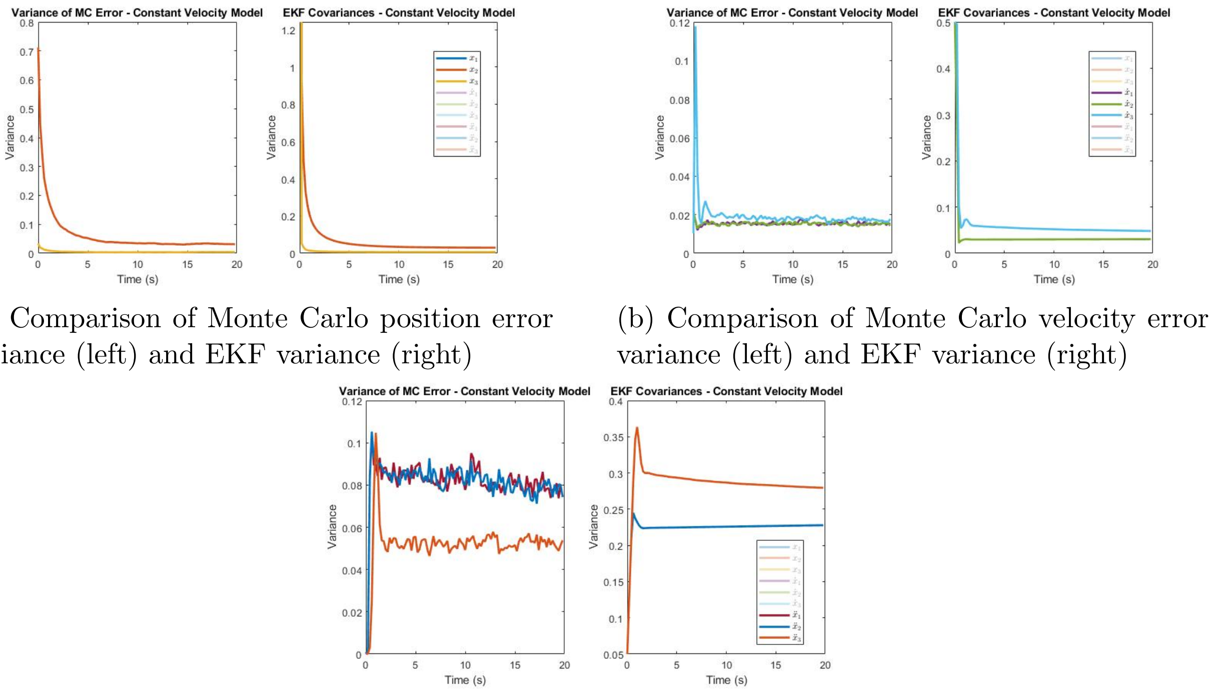


(a) Mean error of position estimates from Monte Carlo

(b) Mean error of velocity estimates from Monte Carlo

(c) Mean error of velocity estimates from Monte Carlo

Figure A.2: Mean Monte Carlo state errors over time when estimating the states of a vehicle with a constant velocity dynamic model.



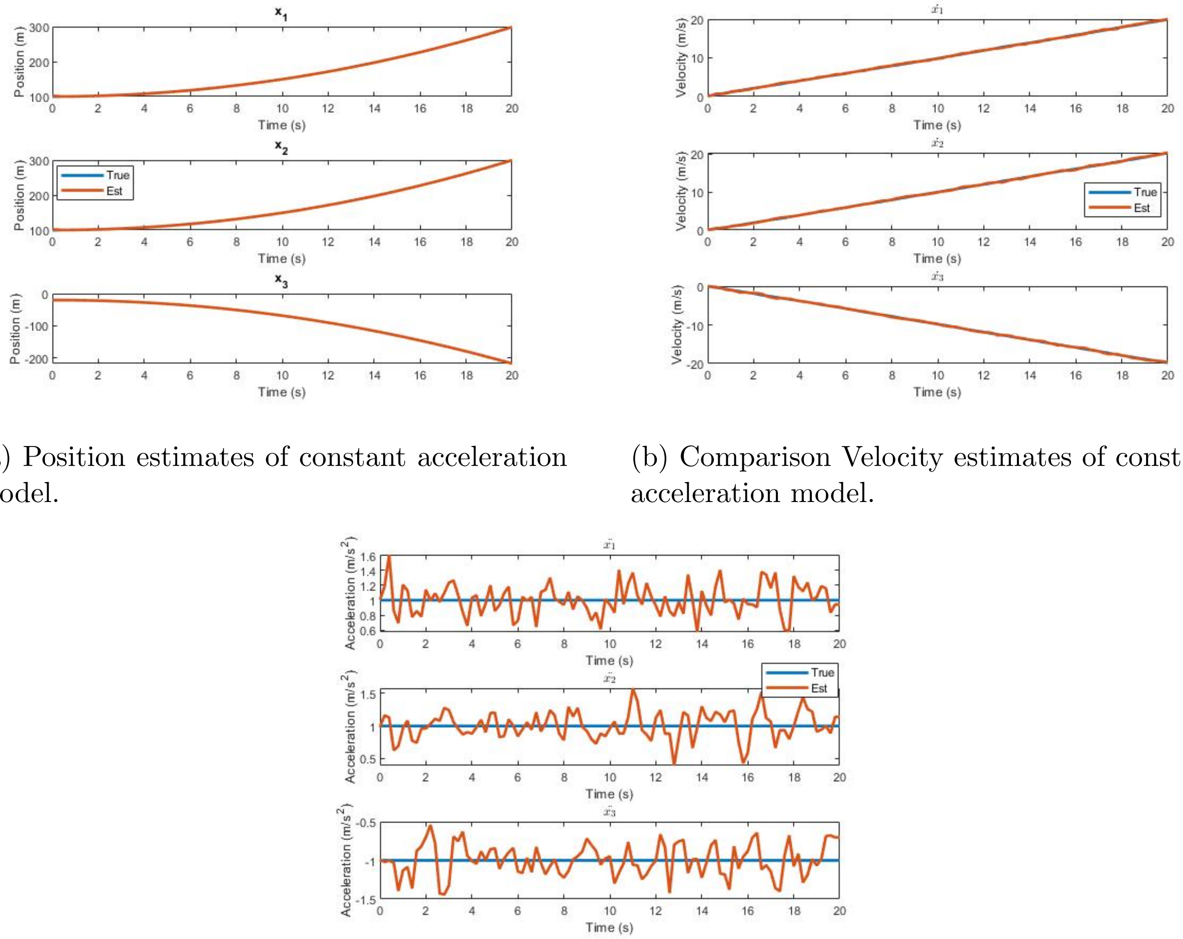
(a) Comparison of Monte Carlo position error variance (left) and EKF variance (right)

(b) Comparison of Monte Carlo velocity error variance (left) and EKF variance (right)

(c) Comparison of Monte Carlo acceleration error variance (left) and EKF variance (right)

Figure A.3: Comparison of Monte Carlo state error variance and calculated EKF variance.

A.2 Constant Acceleration Model

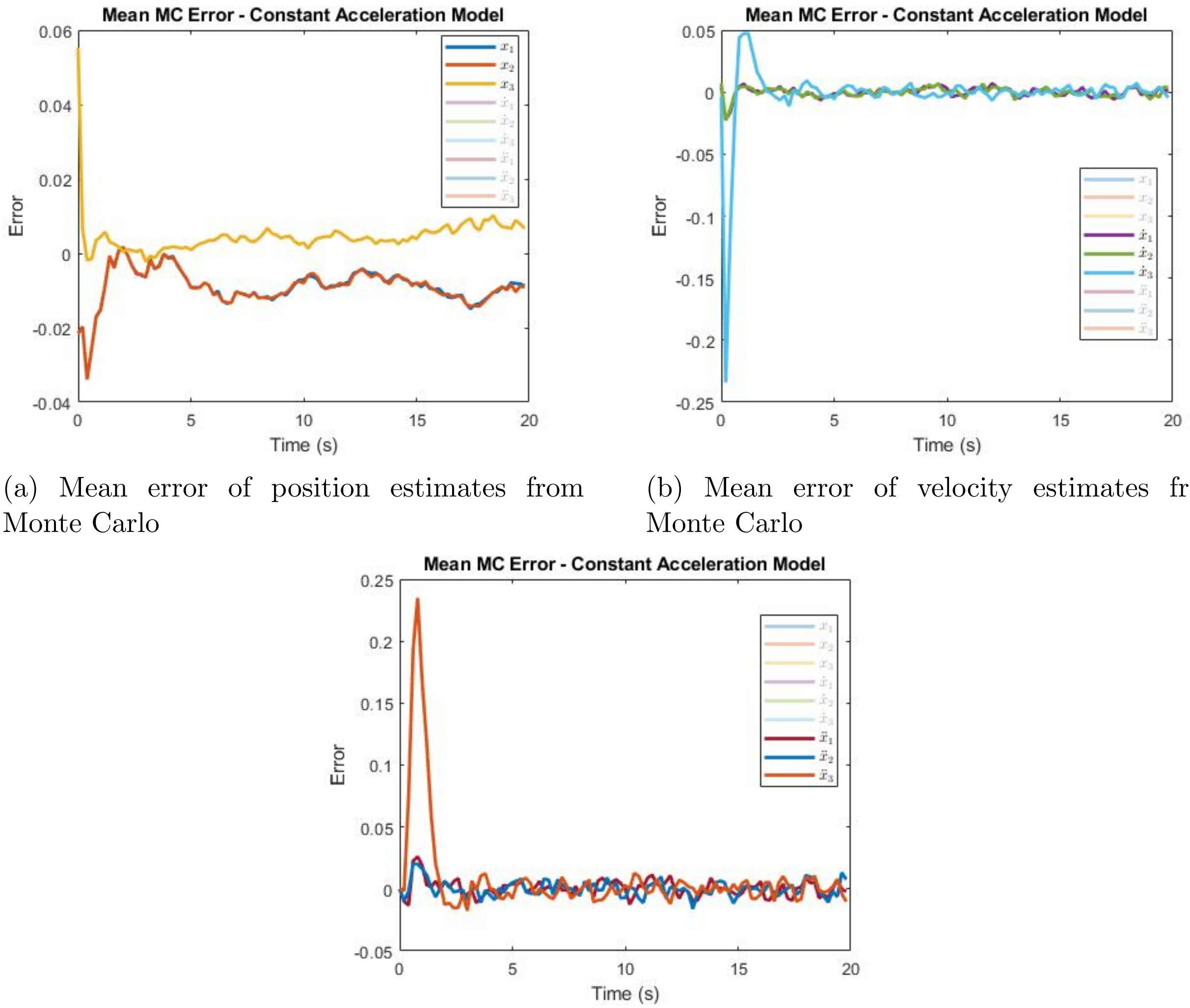


(a) Position estimates of constant acceleration model.

(b) Comparison Velocity estimates of constant acceleration model.

(c) Acceleration estimates of constant acceleration model.

Figure A.4: An example of the EKF estimating the states of a vehicle with a constant acceleration dynamic model.

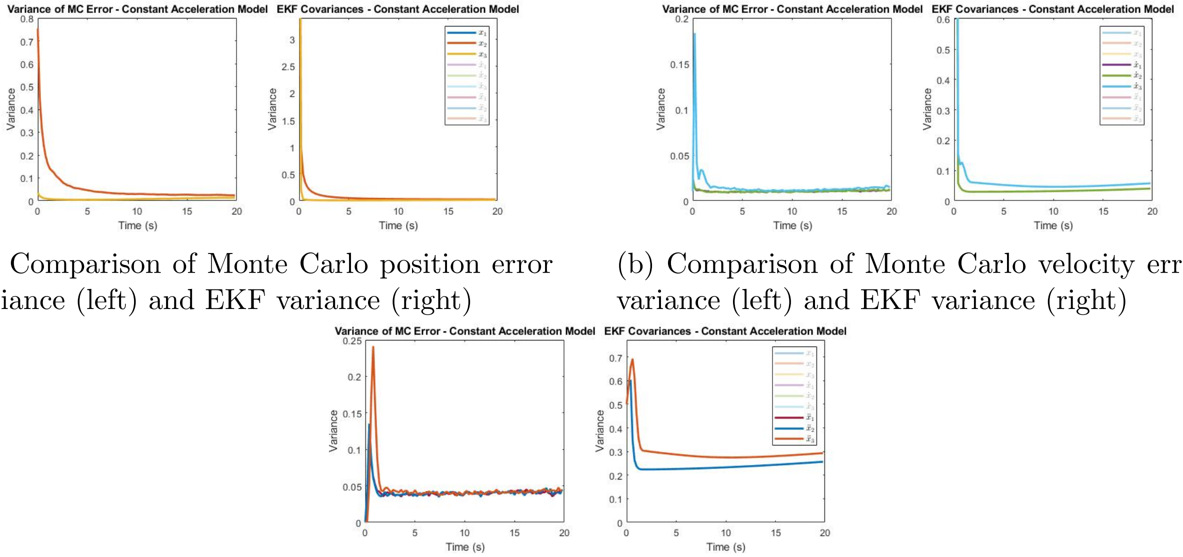


(a) Mean error of position estimates from Monte Carlo

(b) Mean error of velocity estimates from Monte Carlo

(c) Mean error of velocity estimates from Monte Carlo

Figure A.5: Mean Monte Carlo state errors over time when estimating the states of a vehicle with a constant acceleration dynamic model.



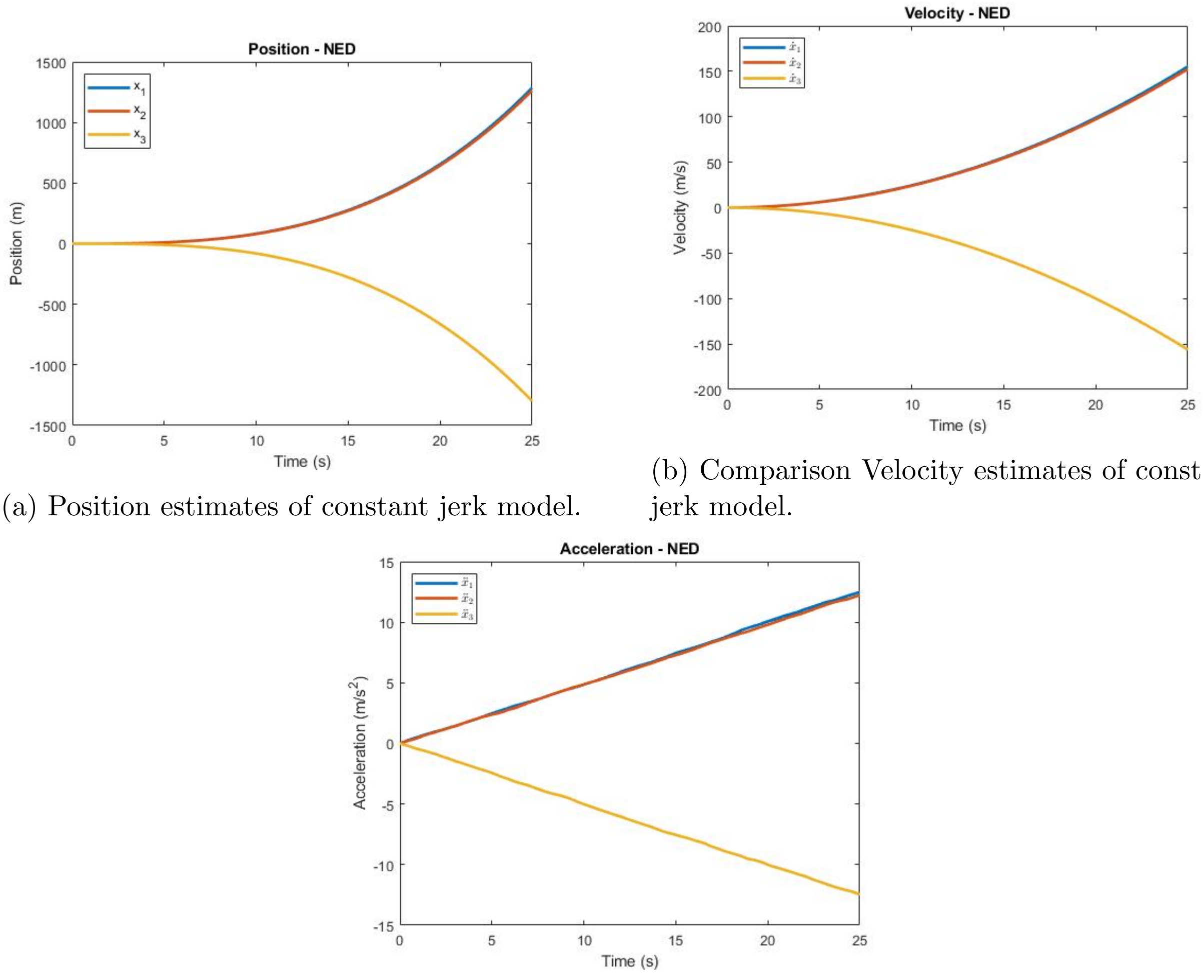
(a) Comparison of Monte Carlo position error variance (left) and EKF variance (right)

(b) Comparison of Monte Carlo velocity error variance (left) and EKF variance (right)

(c) Comparison of Monte Carlo acceleration error variance (left) and EKF variance (right)

Figure A.6: Comparison of Monte Carlo state error variance and calculated EKF variance.

A.3 Constant Jerk Model

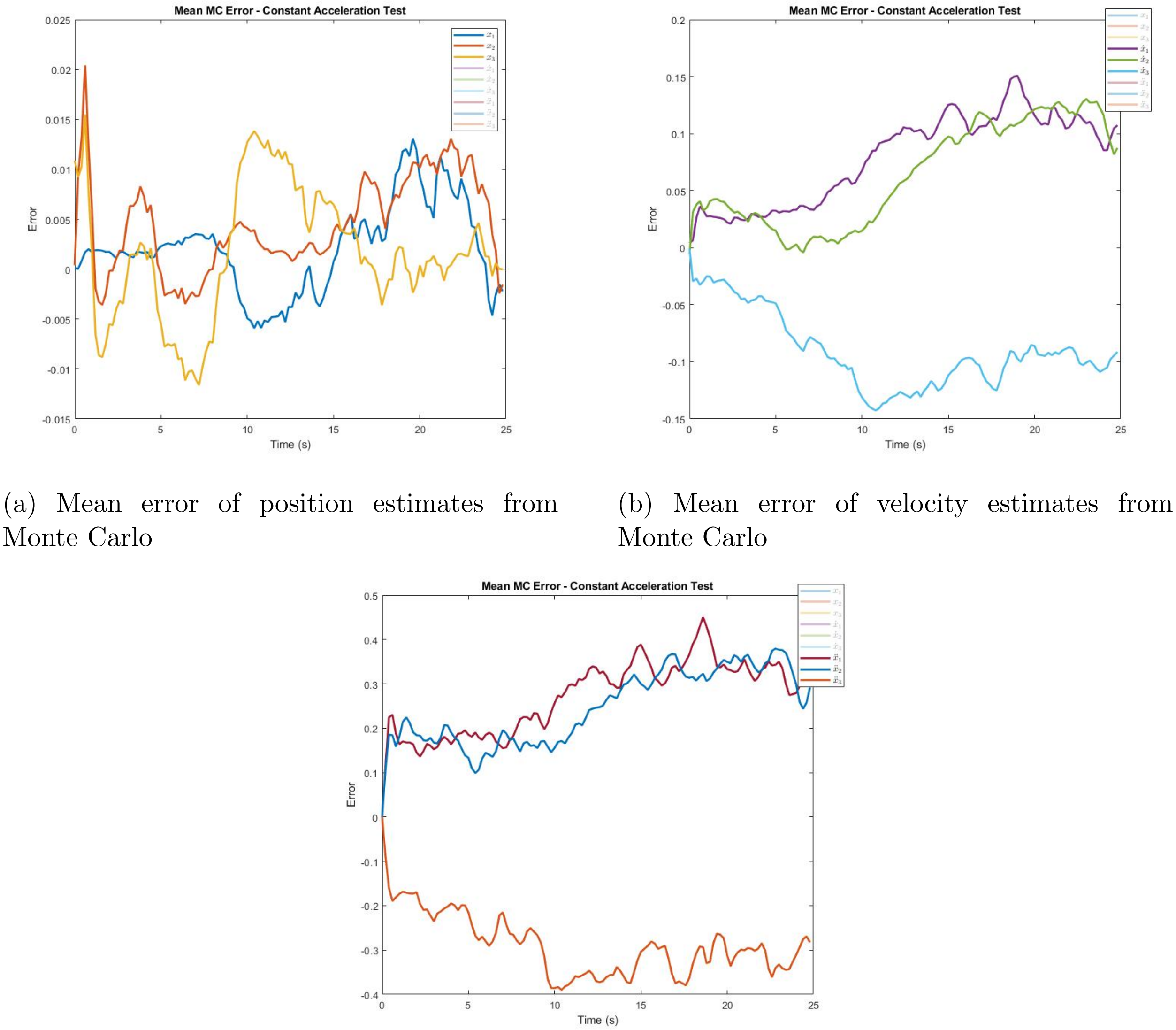


(a) Position estimates of constant jerk model.

(b) Comparison Velocity estimates of constant jerk model.

(c) Acceleration estimates of constant jerk model.

Figure A.7: An example of the EKF estimating the states of a vehicle with a constant jerk dynamic model.

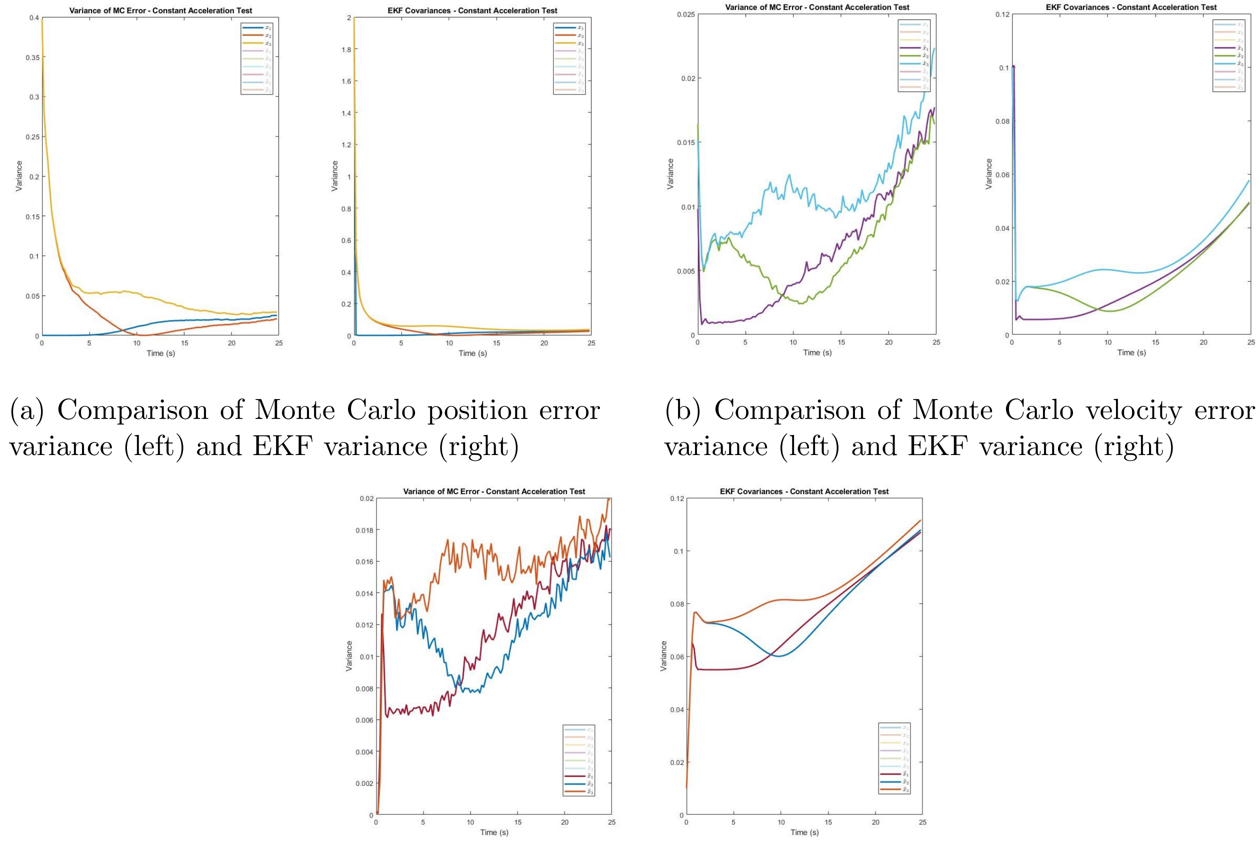


(a) Mean error of position estimates from Monte Carlo

(b) Mean error of velocity estimates from Monte Carlo

(c) Mean error of velocity estimates from Monte Carlo

Figure A.8: Mean Monte Carlo state errors over time when estimating the states of a vehicle with a constant jerk dynamic model.



(a) Comparison of Monte Carlo position error variance (left) and EKF variance (right)

(b) Comparison of Monte Carlo velocity error variance (left) and EKF variance (right)

(c) Comparison of Monte Carlo acceleration error variance (left) and EKF variance (right)

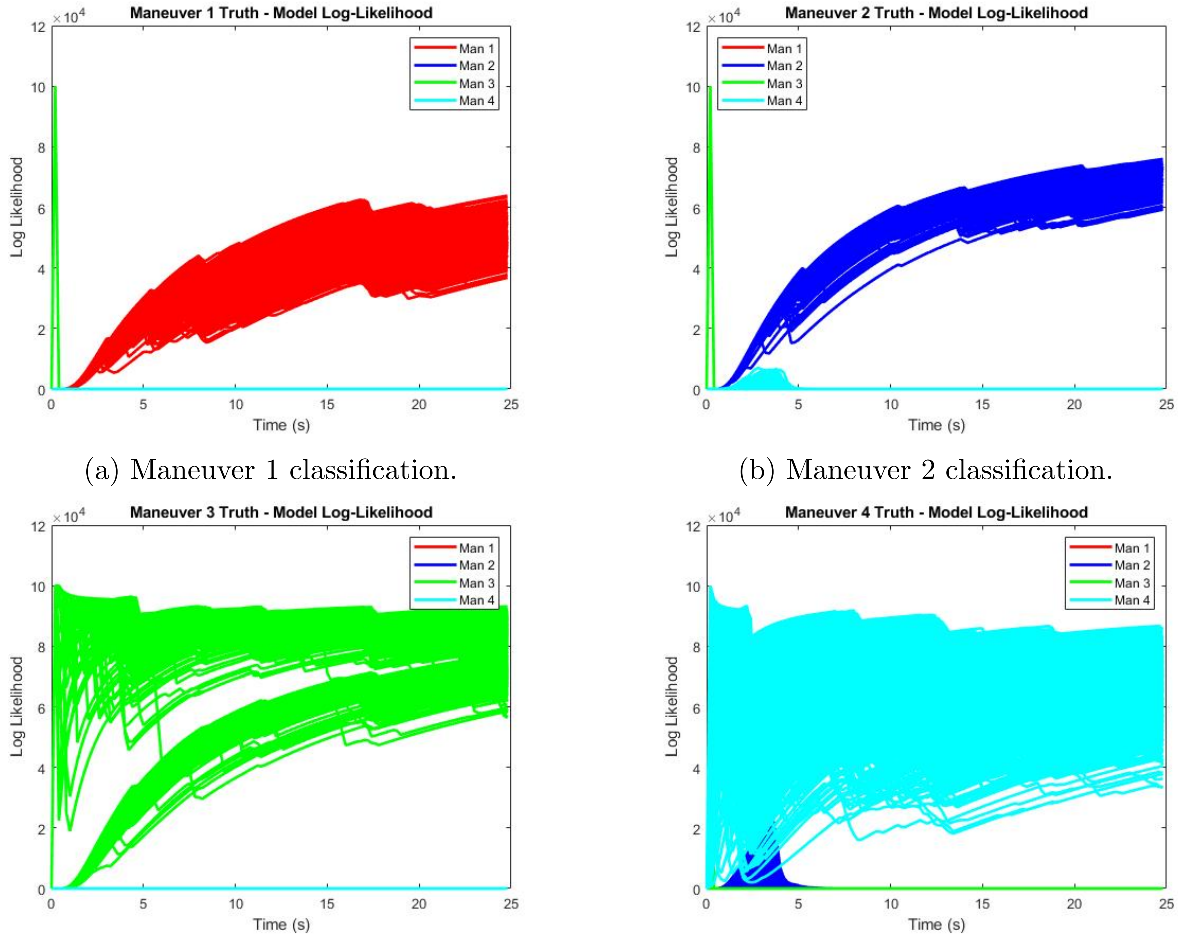
Figure A.9: Comparison of Monte Carlo state error variance and calculated EKF variance.

Appendix B

HMM Classification of UAV Maneuvers - Truth data

The following sections shows individual trajectory classifications for simulated truth data of UAV maneuvers.

B.1 Training Data



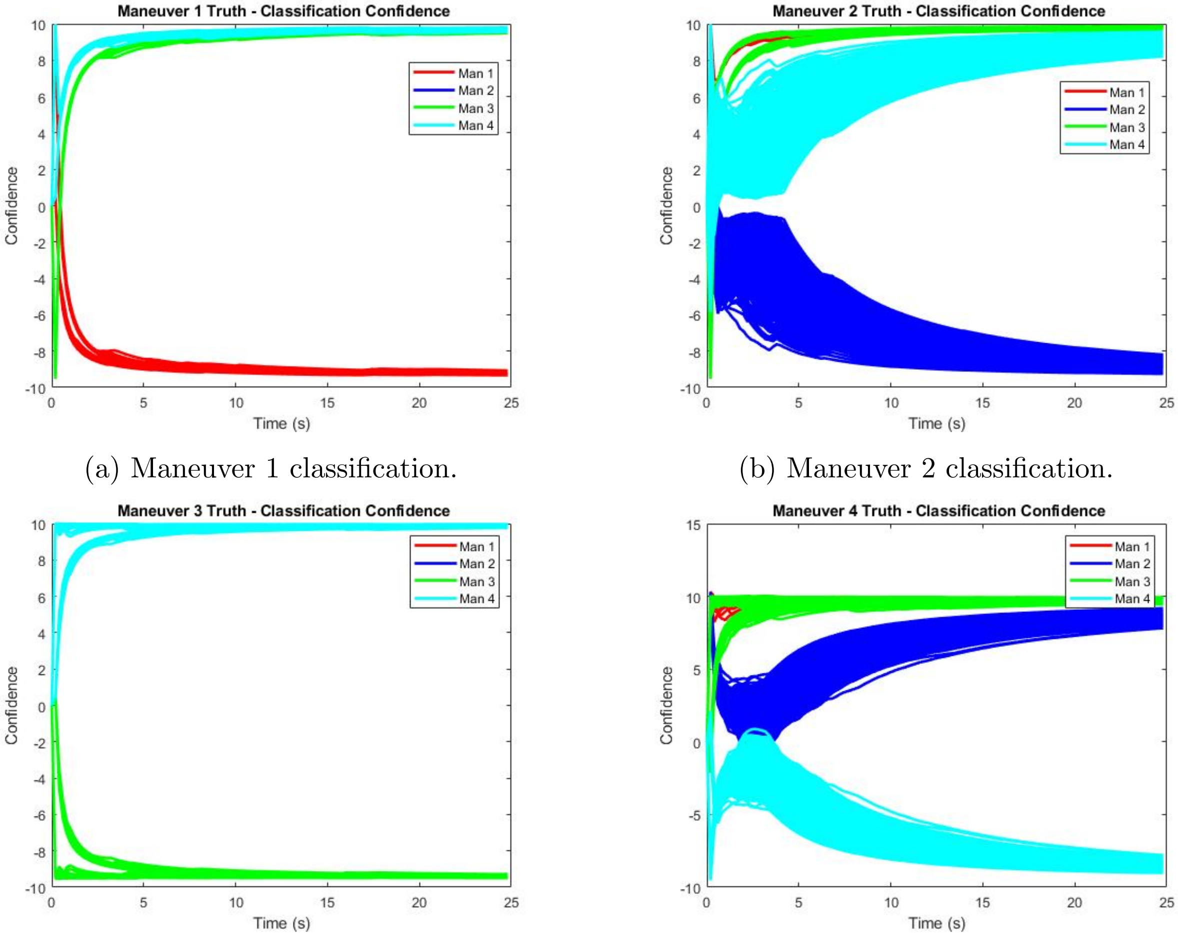
(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure B.1: Log-likelihood for training data classification.



(a) Maneuver 1 classification.

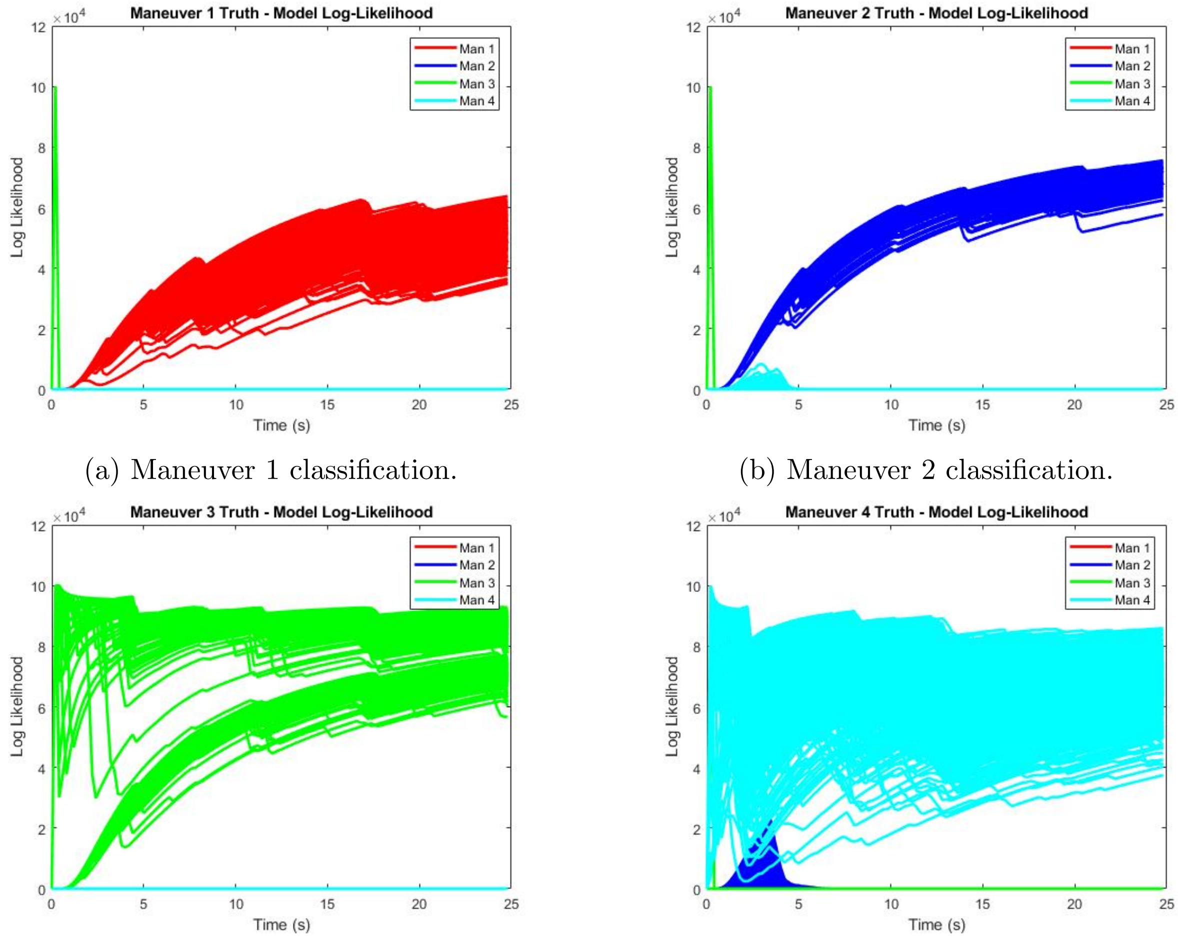
(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure B.2: Confidence for training data classification.

B.2 Test Data


Maneuver 1 Truth - Model Log-Likelihood

Maneuver 2 Truth - Model Log-Likelihood

Maneuver 3 Truth - Model Log-Likelihood

Maneuver 4 Truth - Model Log-Likelihood

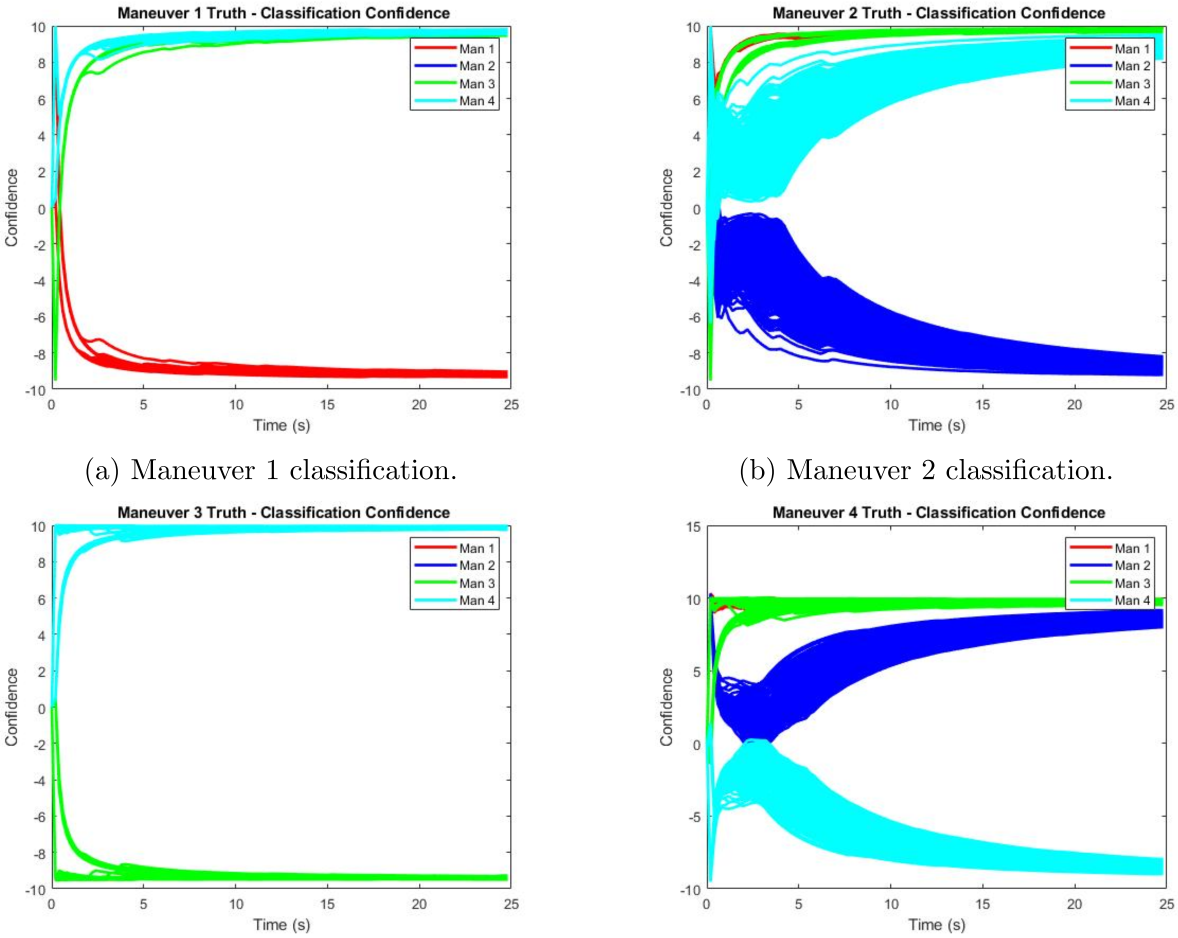
(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure B.3: Log-likelihood for testing data classification.



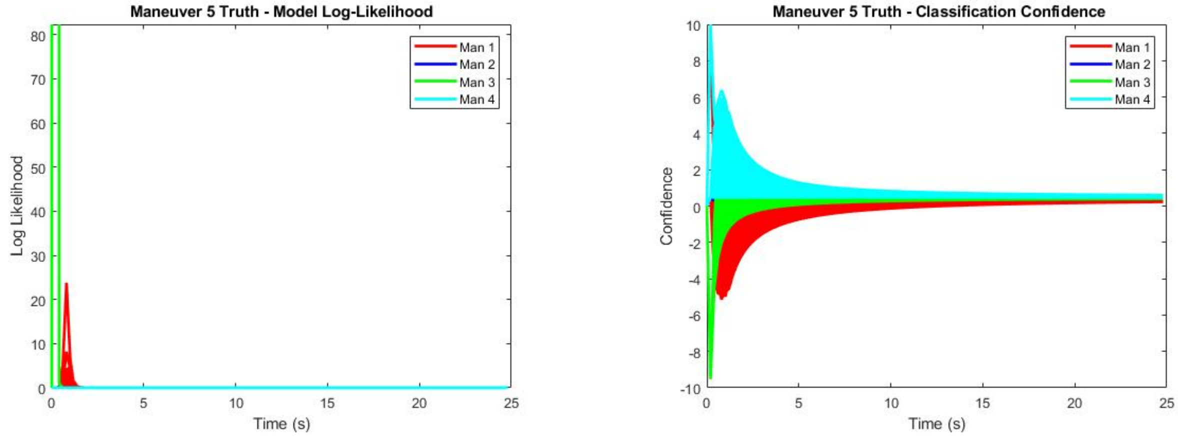
(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure B.4: Confidence for training data classification.



(a) Maneuver 5 Log-likelihood.

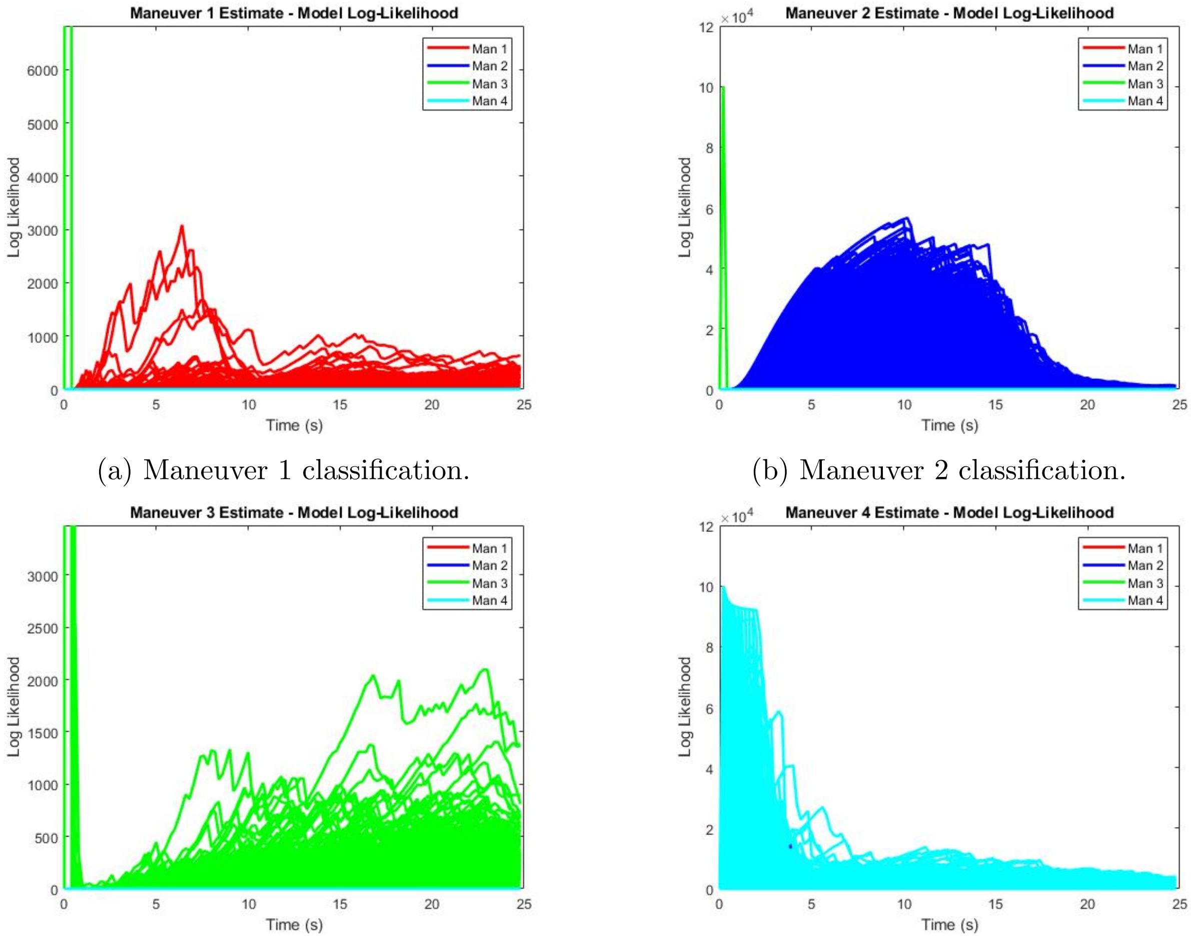
(b) Maneuver 5 Confidence.

Figure B.5: Maneuver 5 (unknown maneuver).

Appendix C
HMM Classification of UAV Maneuvers - EKF data

The following sections shows individual trajectory classifications for simulated estimated data of UAV maneuvers.

C.1 Training Data



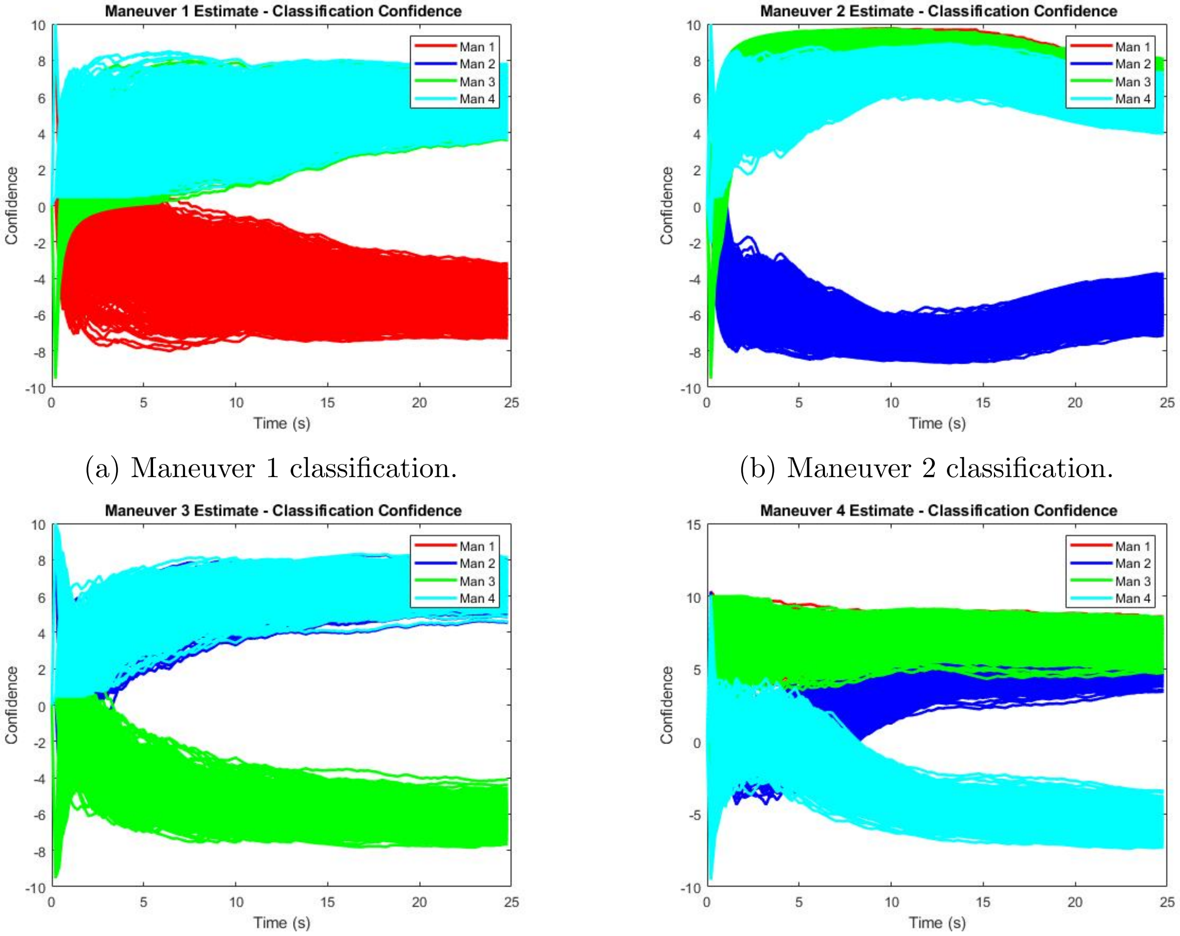
(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure C.1: Log-likelihood for train data classification (EKF sim).



(a) Maneuver 1 classification.

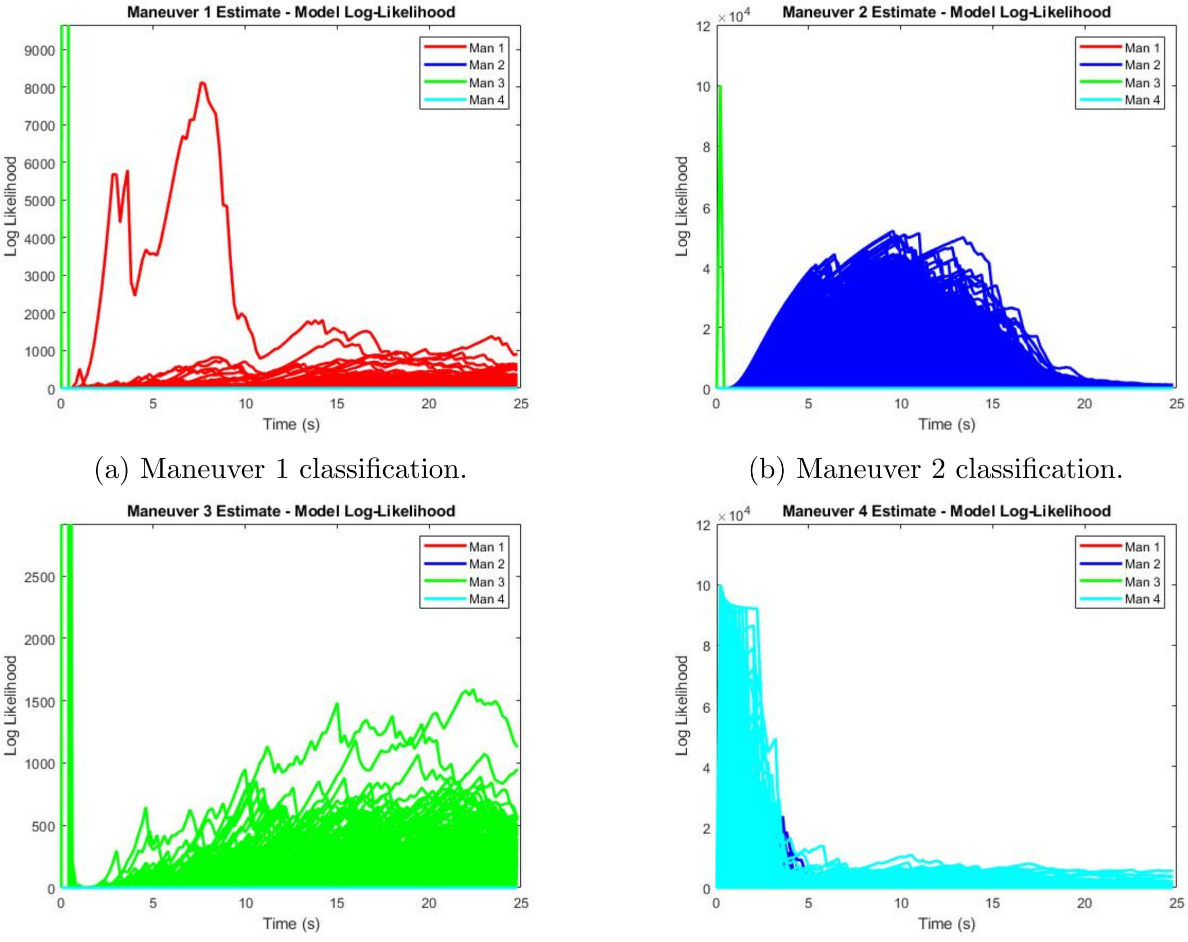
(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure C.2: Confidence for training data classification (EKF sim).

C.2 Test Data



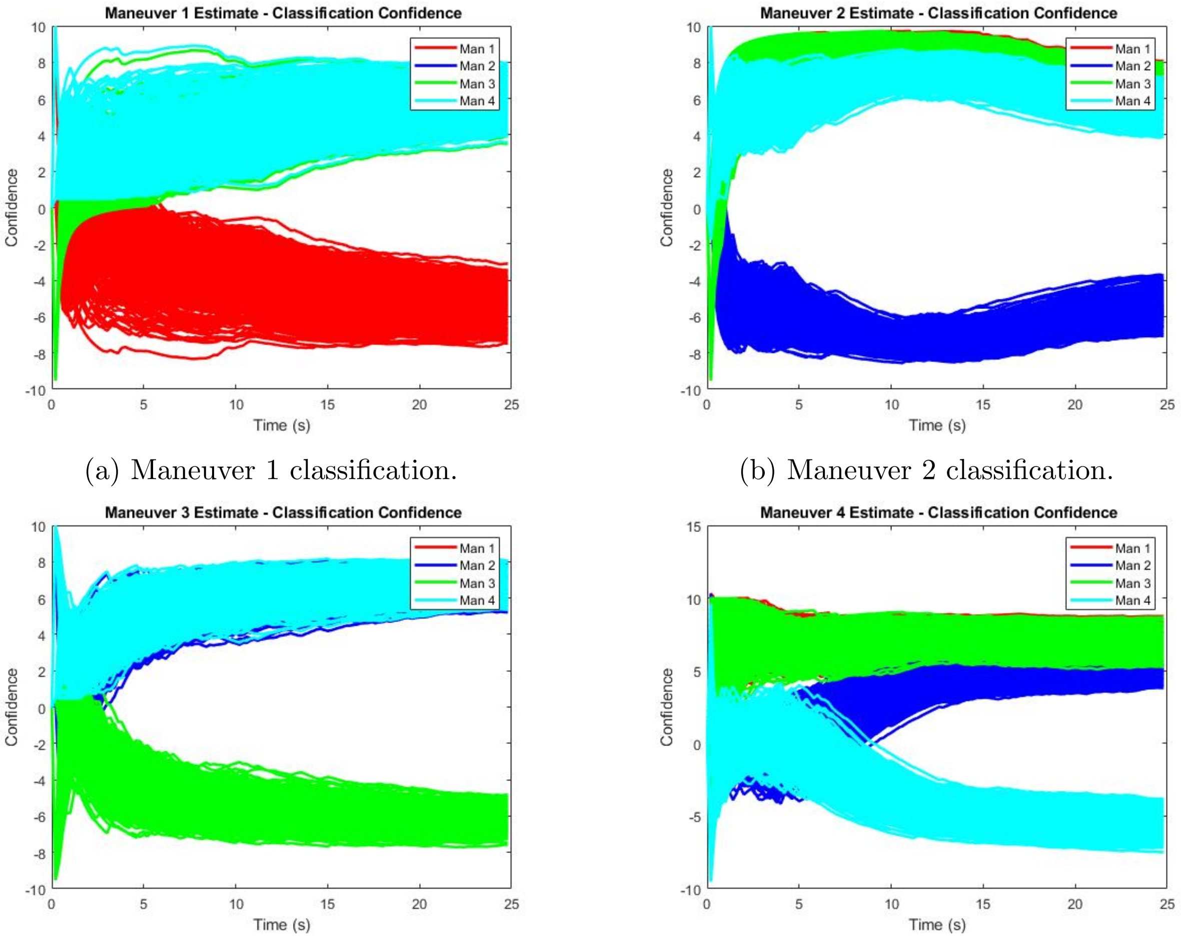
(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure C.3: Log-likelihood for testing data classification (EKF sim).



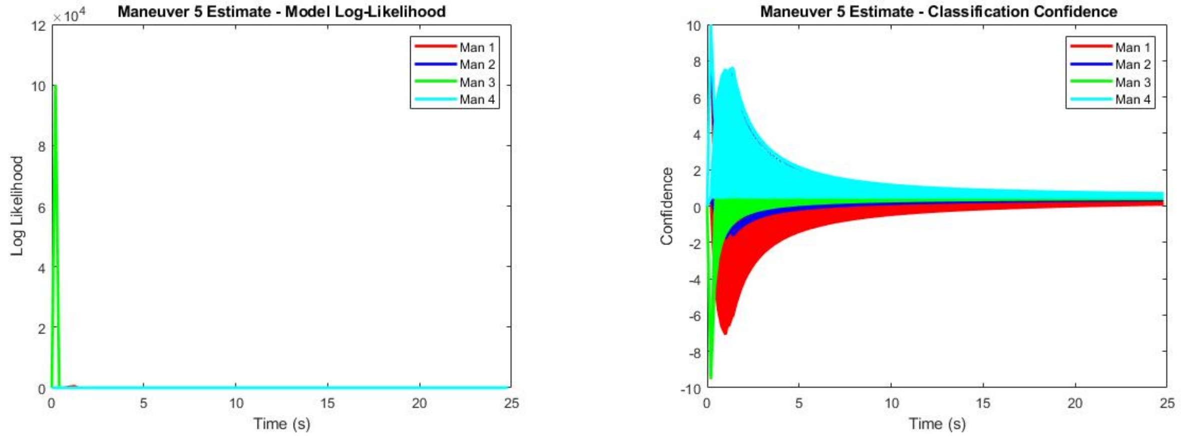
(a) Maneuver 1 classification.

(b) Maneuver 2 classification.

(c) Maneuver 3 classification.

(d) Maneuver 4 classification.

Figure C.4: Confidence for test data classification (EKF sim).



(a) Maneuver 5 Log-likelihood.

(b) Maneuver 5 Confidence.

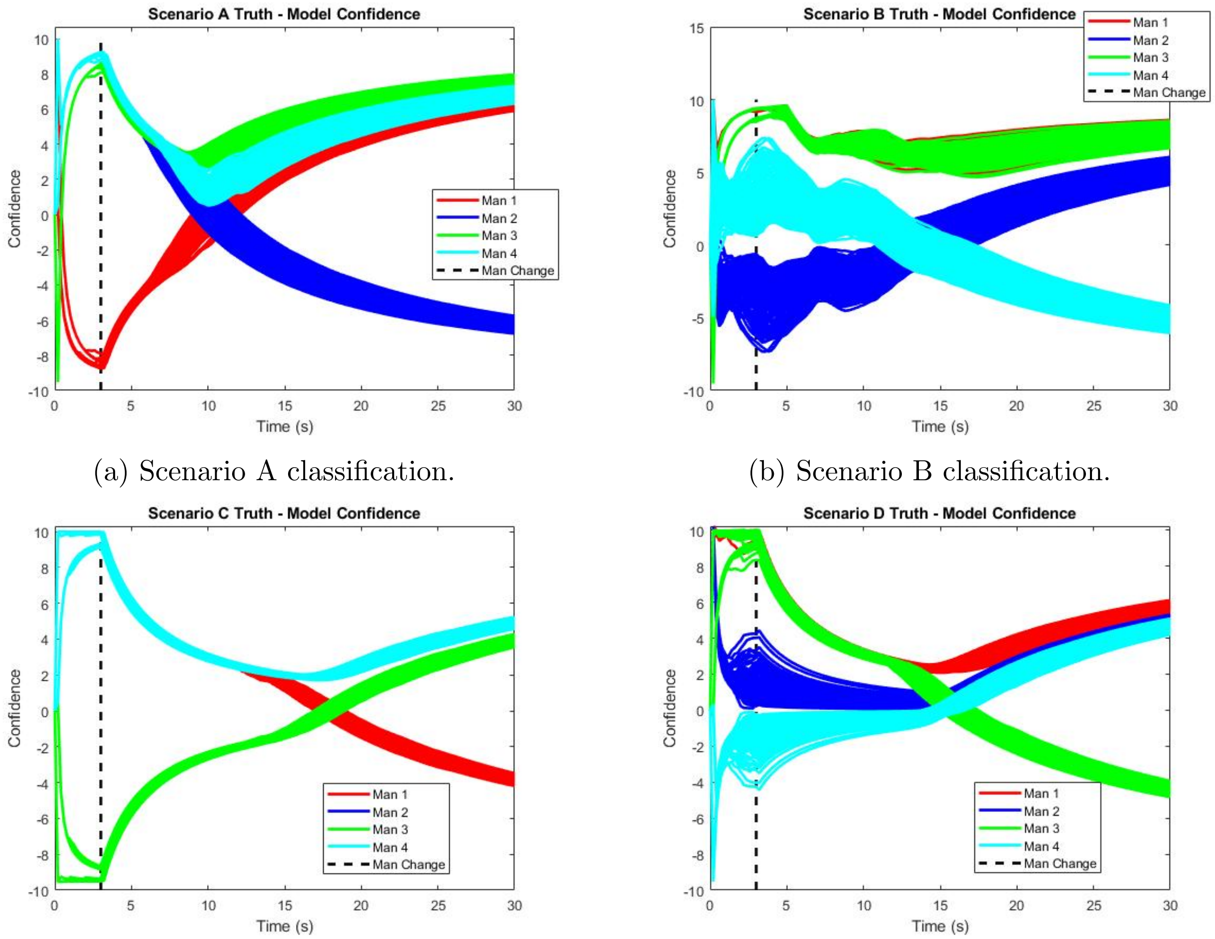
Figure C.5: Maneuver 5 (unknown maneuver) (EKF sim).

Appendix D

HMM Classification with Simulated Changing Maneuvers

The following section shows individual classification results of a scenario in which maneuvers transition throughout flight.

D.1 Truth Data



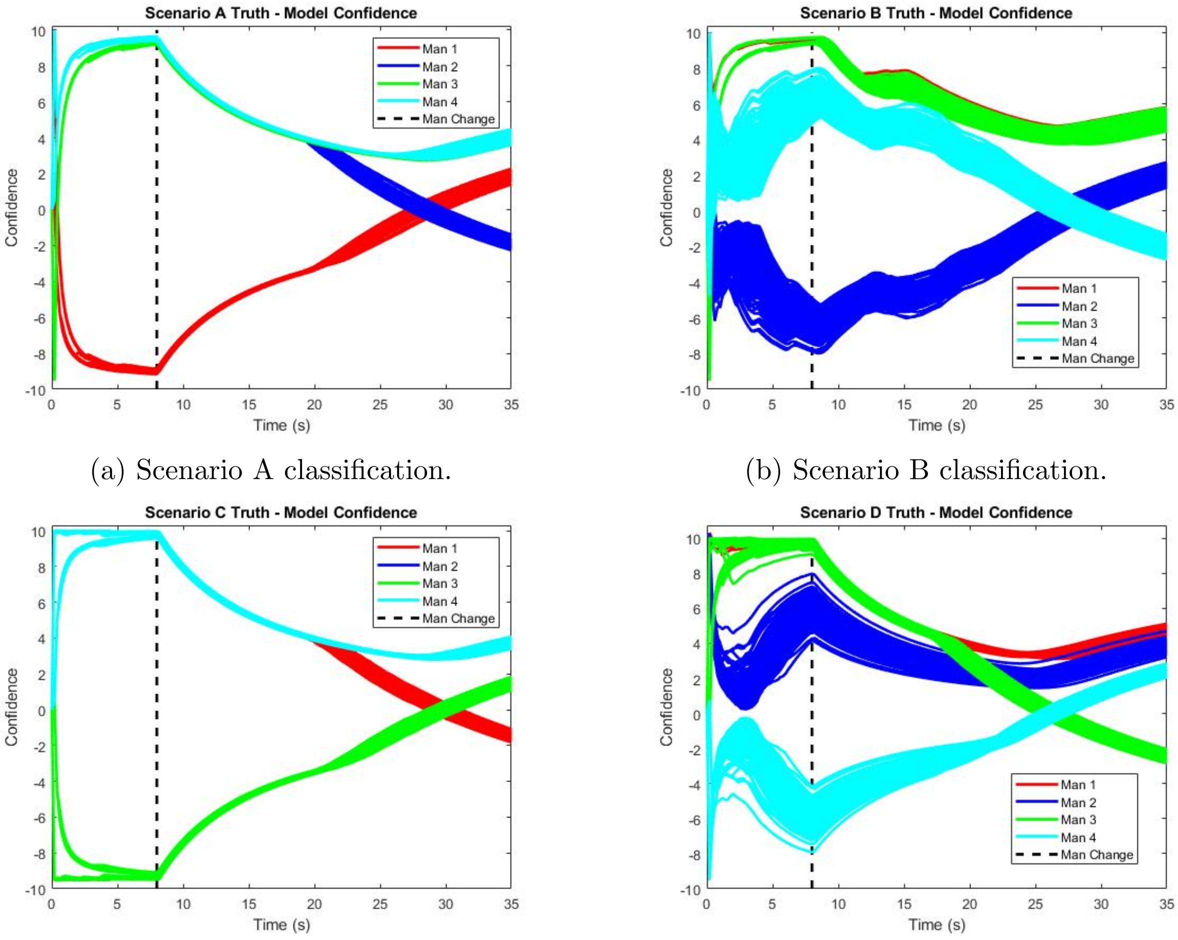
(a) Scenario A classification.

(b) Scenario B classification.

(c) Scenario C classification.

(d) Scenario D classification.

Figure D.1: Average confidence of each HMM when classifying a maneuver switch at 3 seconds - *truth data*.



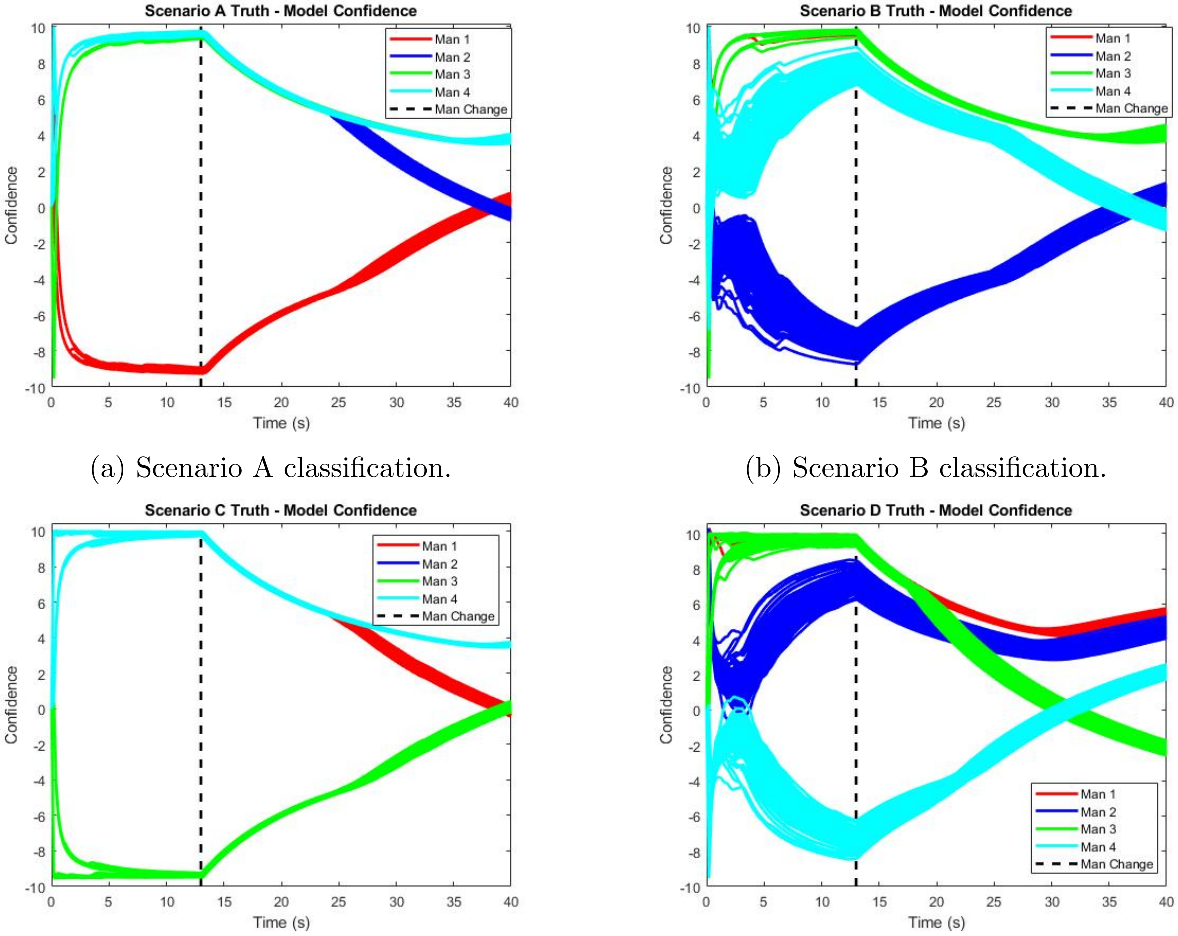
(a) Scenario A classification.

(b) Scenario B classification.

(c) Scenario C classification.

(d) Scenario D classification.

Figure D.2: Confidence of each HMM when classifying a maneuver switch at 8 seconds - *truth data*.



(a) Scenario A classification.

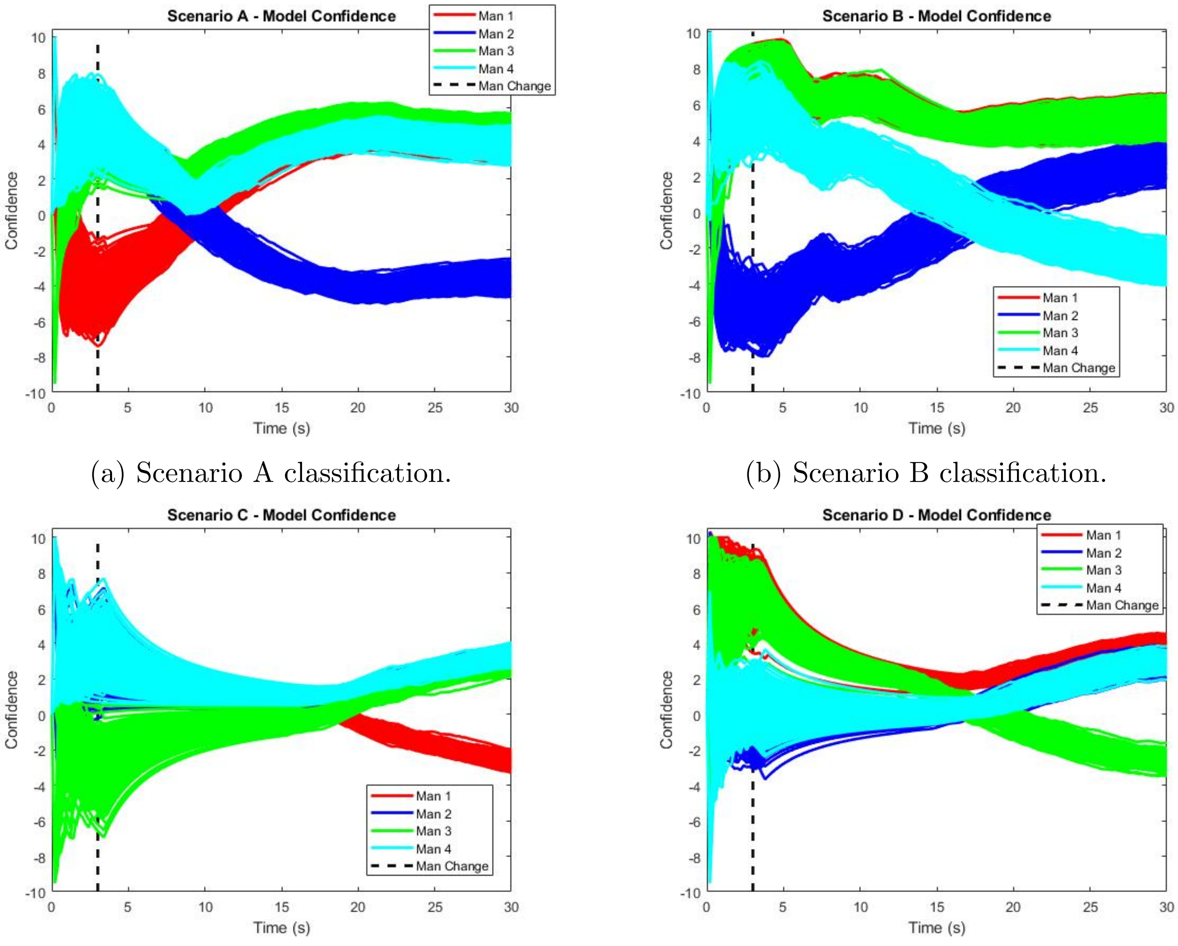
(b) Scenario B classification.

(c) Scenario C classification.

(d) Scenario D classification.

Figure D.3: Confidence of each HMM when classifying a maneuver switch at 13 seconds - *truth data*.

D.2 EKF Data



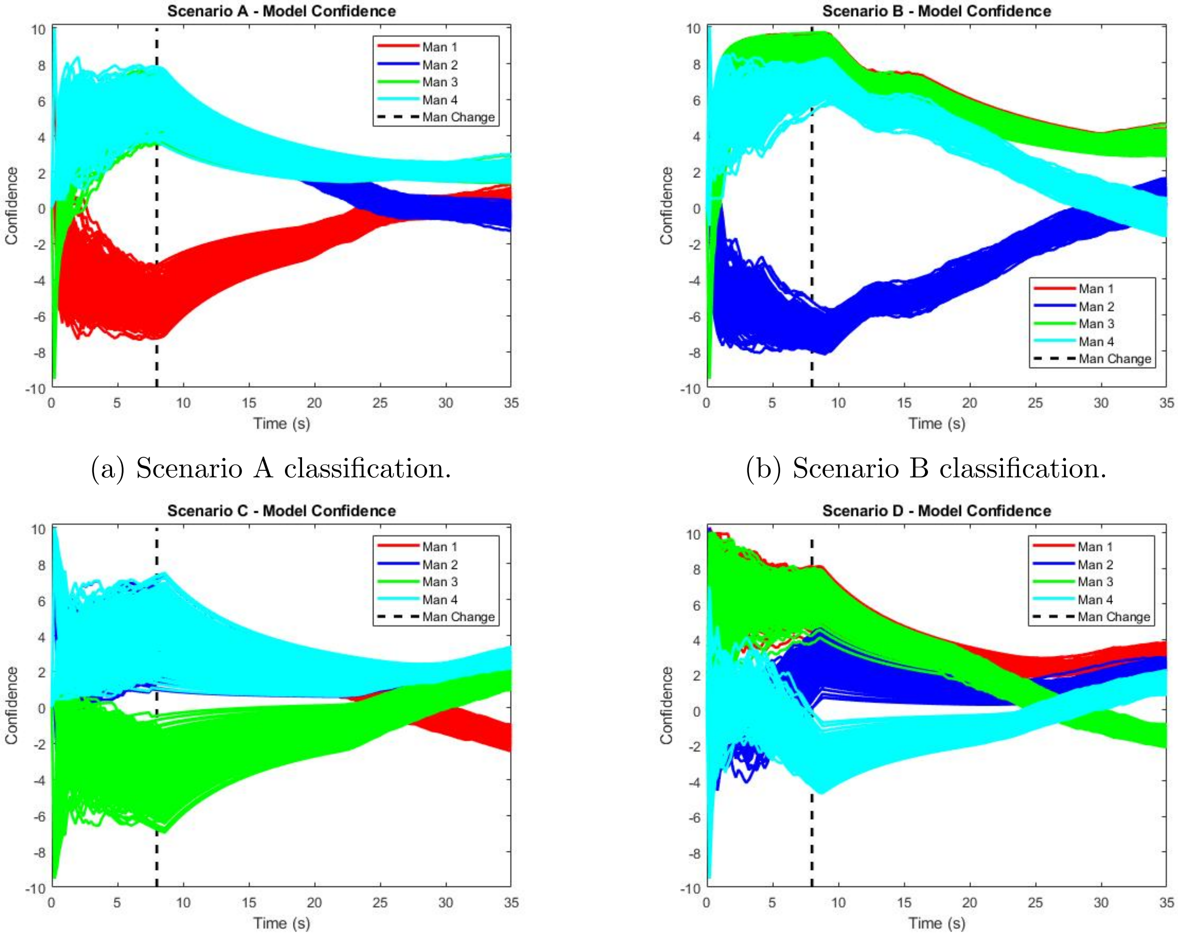
(a) Scenario A classification.

(b) Scenario B classification.

(c) Scenario C classification.

(d) Scenario D classification.

Figure D.4: Confidence of each HMM when classifying a maneuver switch at 3 seconds - *estimated data*.



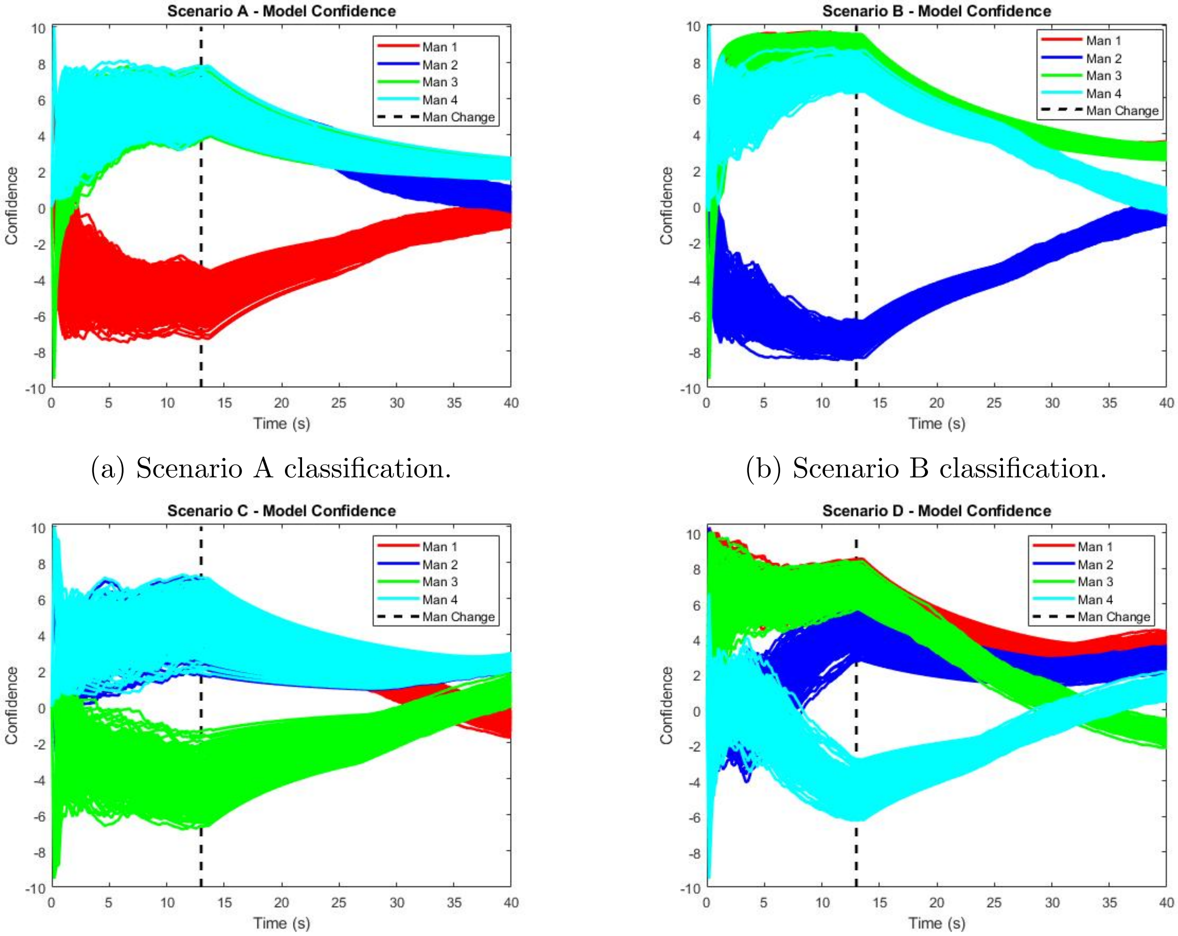
(a) Scenario A classification.

(b) Scenario B classification.

(c) Scenario C classification.

(d) Scenario D classification.

Figure D.5: Confidence of each HMM when classifying a maneuver switch at 8 seconds - *estimated data*.



(a) Scenario A classification.

(b) Scenario B classification.

(c) Scenario C classification.

(d) Scenario D classification.

Figure D.6: Confidence of each HMM when classifying a maneuver switch at 13 seconds - *estimated data*.

Appendix E

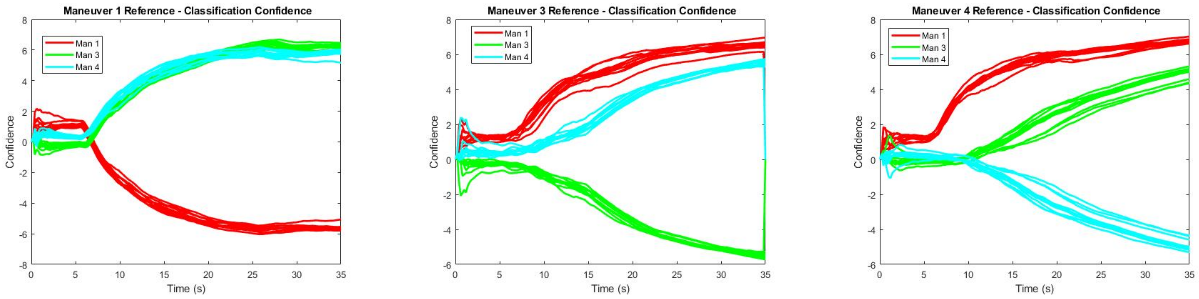
HMM Classification of Real World Data

The following sections shows classification results of individual trajectories for classification of real-world data.

E.1 Experimental Data Trained HMM

E.1.1 Reference Data

Training Data



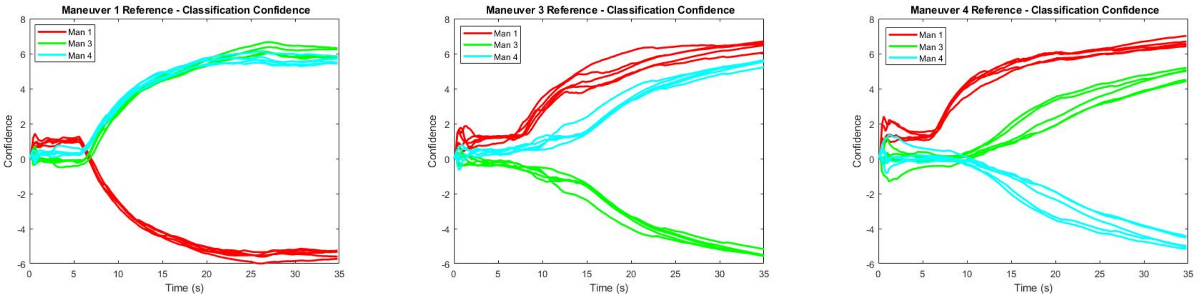
(a) Classification of Maneuver 1 reference training data.

(b) Classification of Maneuver 3 reference training data.

(c) Classification of Maneuver 4 reference training data.

Figure E.1: Confidence over time for each HMM during the classification process - *reference training data*.

Test Data



(a) Classification of Maneuver 1 reference testing data.

(b) Classification of Maneuver 3 reference testing data.

(c) Classification of Maneuver 4 reference testing data.

Figure E.2: Confidence over time for each HMM during the classification process - *reference test data*.

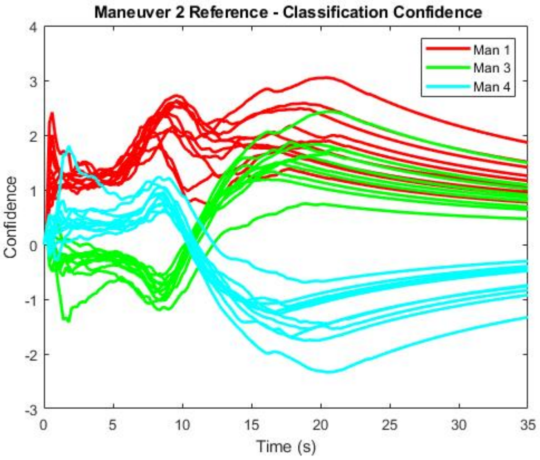
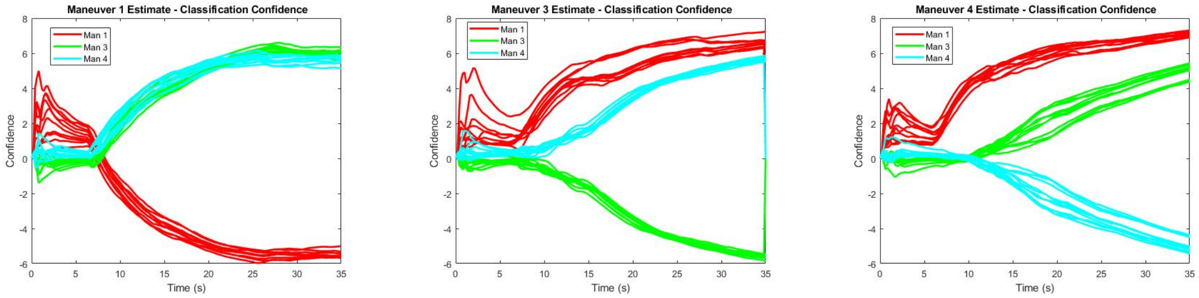


Figure E.3: Confidence of each HMM when given an unknown maneuver - *reference data*.

E.1.2 Estimated Data

Training Data



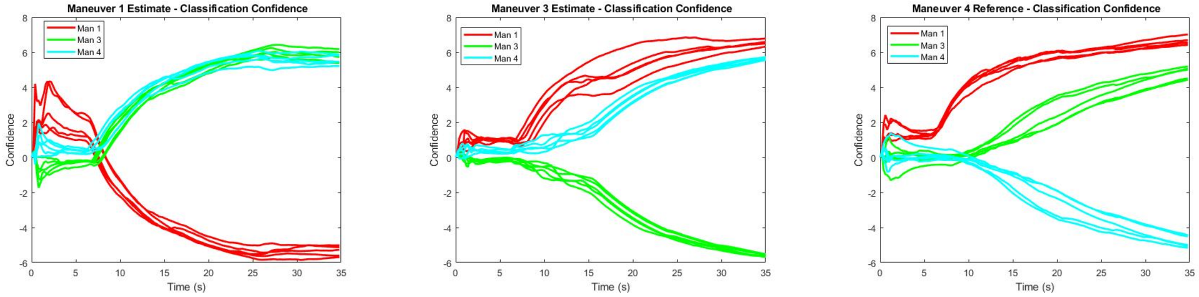
(a) Classification of Maneuver 1 estimated training data.

(b) Classification of Maneuver 3 estimated training data.

(c) Classification of Maneuver 4 estimated training data.

Figure E.4: Confidence over time for each HMM during the classification process - *estimated training data.*

Test Data



(a) Classification of Maneuver 1 estimated testing data.

(b) Classification of Maneuver 3 estimated testing data.

(c) Classification of Maneuver 4 estimated testing data.

Figure E.5: Confidence over time for each HMM during the classification process - *estimated test data.*

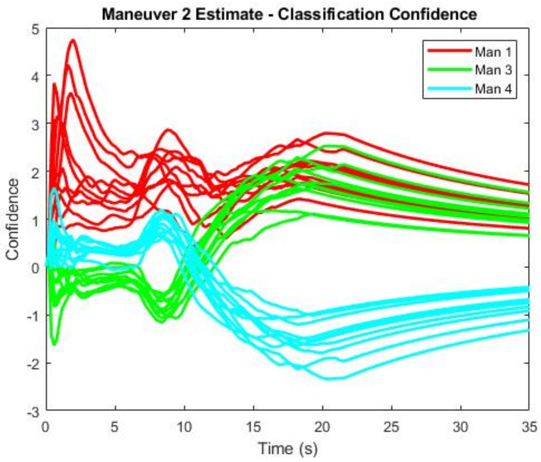
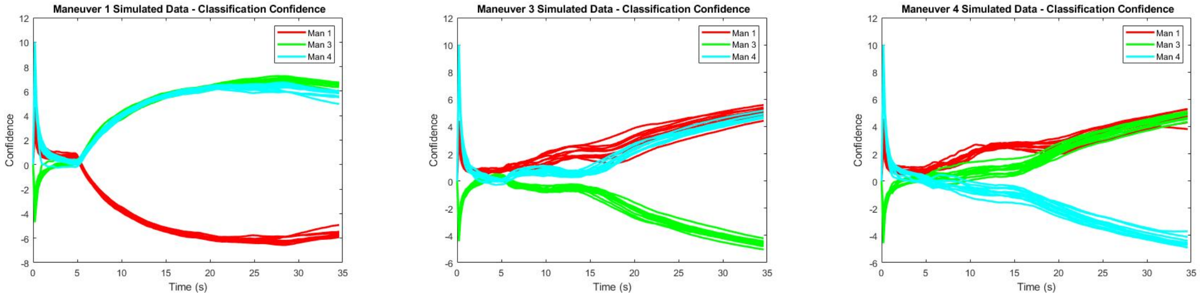


Figure E.6: Confidence over time of each HMM when given an unknown maneuver - *estimated data.*

E.2 Simulation Data Trained HMM

E.2.1 Training Data



(a) Classification of Maneuver 1 simulated training data.

(b) Classification of Maneuver 3 simulated training data.

(c) Classification of Maneuver 4 simulated training data.

Figure E.7: Confidence over time for each model - *simulated training data.*

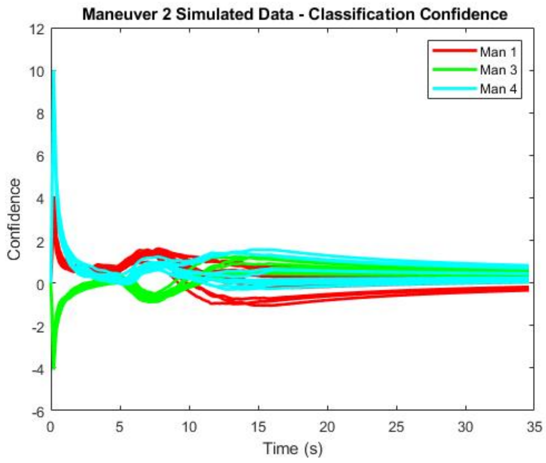
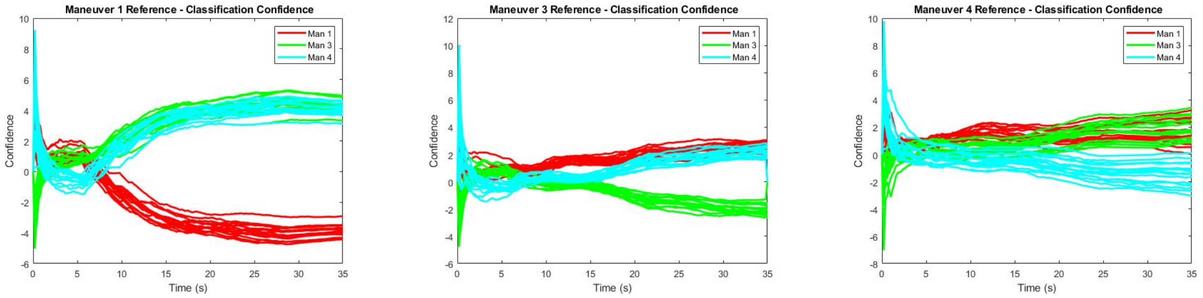


Figure E.8: Confidence of each HMM when given an unknown maneuver - *Simulated Training Data*

E.2.2 Test Data

Reference Data



(a) Classification of Maneuver 1 reference data.

(b) Classification of Maneuver 3 reference data.

(c) Classification of Maneuver 4 reference data.

Figure E.9: Confidence over time using HMMs trained with simulated data - *reference data*.

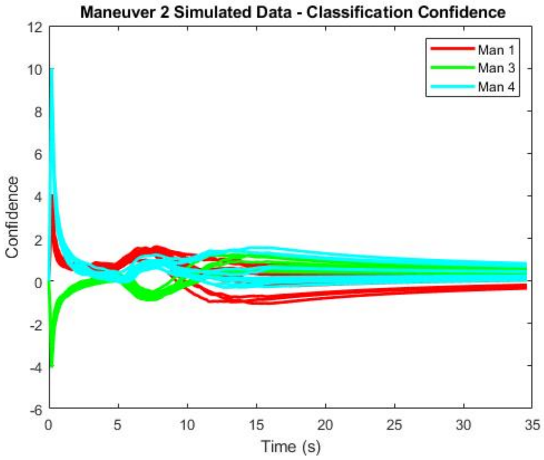
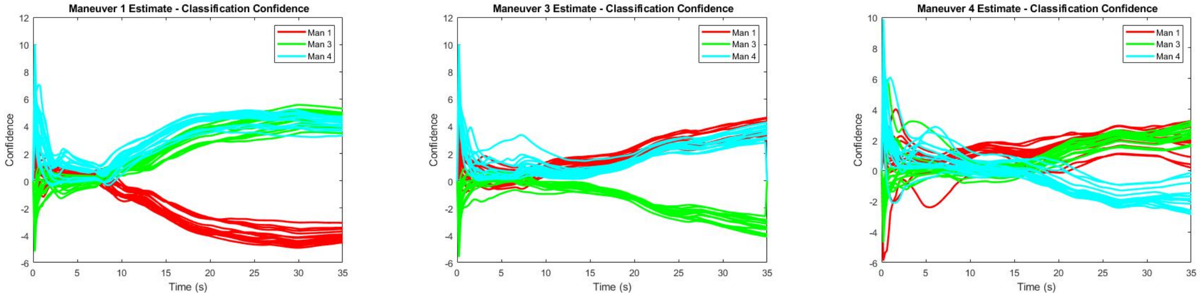


Figure E.10: Confidence of each HMM when given an unknown maneuver - *reference data*

Estimated Data



(a) Classification of Maneuver 1 estimated data.

(b) Classification of Maneuver 3 estimated data.

(c) Classification of Maneuver 4 estimated data.

Figure E.11: Confidence over time using simulation trained HMMs - *estimated reference data.*

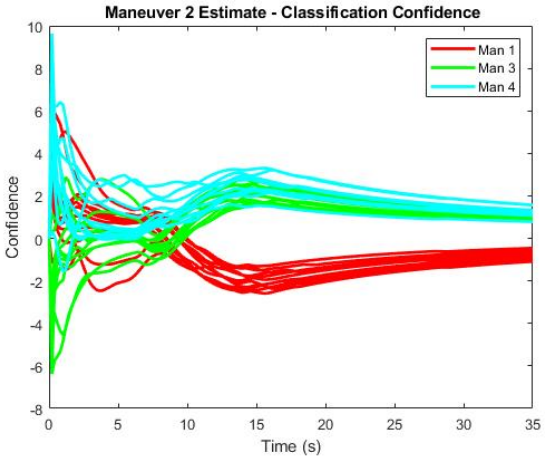


Figure E.12: Confidence of each model when classifying an unknown maneuver - *estimated reference data*.