INTERACTIVE MULTIPLE MODEL ESTIMATION FOR
UNMANNED AIRCRAFT SYSTEMS DETECT AND AVOID

BY

ADRIANO CANOLLA

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical and Aerospace Engineering
in the Graduate College of the
Illinois Institute of Technology

Approved

Advisor

Chicago, Illinois
December 2018
ACKNOWLEDGMENT

First and foremost, I want to thank to my academic advisor and dissertation committee chair Professor Boris Pervan for his endless support. I appreciate his vision for my research, matching my previous experience with his clear guidance and enthusiasm for this research. I must also thank my co-advisor, Dr. Michael Jamoom, for his technical expertise in the subject of this research, and endless availability and support for mentoring me in the process. This dissertation could not have been written without them.

I would also like to extend my deepest gratitude to the rest of my committee: Professors Matthew Spenko, David R. Williams and Yang Xu.

I gratefully acknowledge CAPES for funding and supporting this research as well as ANAC for approving my license that made this PhD possible. My colleague Antonio Quadros, and professors Paulo Greco Junior and Donizeti de Andrade which helped me on my application.

My fellow colleagues in the Navigation and Guidance Laboratory along these years: Professors Samer Khanafseh and Mathieu Joerger, Dr. Stefan Stevanovic, Dr. Yawei Zhai, Dr. Cagatay Tanil, Ryan Cassel, Josep Calavia, Jaymin Patel, Wengxiang Zhao, Elisa Gallon, Tejas Desai and Pau Martinez Jimenez who have made my time as a PhD candidate as memorable as possible.

I would like to thank my parents and my sister for always supporting me on my projects no matter where they take me. My girlfriend, Renata Mendonca for her encouragement through the difficult times during this PhD, her love and patience for the long time apart.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xi</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1. The Need for Detect and Avoid</td>
<td>1</td>
</tr>
<tr>
<td>1.2. Well Clear and Collision Avoidance</td>
<td>3</td>
</tr>
<tr>
<td>1.3. Problem Statement</td>
<td>5</td>
</tr>
<tr>
<td>1.4. Prior Work</td>
<td>6</td>
</tr>
<tr>
<td>1.5. Thesis Contributions</td>
<td>10</td>
</tr>
<tr>
<td>1.6. Thesis Outline</td>
<td>12</td>
</tr>
<tr>
<td>2. DYNAMIC MODELS FOR MANEUVERING INTRUDERS</td>
<td>13</td>
</tr>
<tr>
<td>2.1. Target Tracking: Maneuvering Trajectory vs Constant Velocity</td>
<td>13</td>
</tr>
<tr>
<td>2.2. Encounter Model</td>
<td>15</td>
</tr>
<tr>
<td>2.3. Kalman Filters</td>
<td>18</td>
</tr>
<tr>
<td>2.4. Relative Intruder State</td>
<td>21</td>
</tr>
<tr>
<td>2.5. Summary of Dynamic Models for Maneuvering Intruders</td>
<td>22</td>
</tr>
<tr>
<td>3. INTERACTIVE MULTIPLE MODEL</td>
<td>23</td>
</tr>
<tr>
<td>3.1. Multiple Model Adaptive Estimation</td>
<td>23</td>
</tr>
<tr>
<td>3.2. Basic Concepts of the IMM</td>
<td>24</td>
</tr>
<tr>
<td>3.3. IMM Mode Definition</td>
<td>28</td>
</tr>
<tr>
<td>3.4. IMM Estimation Error Sources</td>
<td>33</td>
</tr>
<tr>
<td>3.5. Perfect IMM</td>
<td>34</td>
</tr>
<tr>
<td>3.6. Transition Matrix Probabilities</td>
<td>34</td>
</tr>
<tr>
<td>3.7. Process Noise</td>
<td>35</td>
</tr>
<tr>
<td>3.8. Interactive Multiple Model Summary</td>
<td>36</td>
</tr>
<tr>
<td>4. HAZARD STATE ESTIMATION</td>
<td>37</td>
</tr>
<tr>
<td>4.1. DAA Well Clear</td>
<td>37</td>
</tr>
<tr>
<td>4.2. Time to Horizontal CPA</td>
<td>39</td>
</tr>
<tr>
<td>4.3. Horizontal Miss Distance</td>
<td>41</td>
</tr>
<tr>
<td>4.4. Vertical Separation</td>
<td>42</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.5. New Hazard State Prediction Method</td>
<td>43</td>
</tr>
<tr>
<td>4.6. Hazard States Summary</td>
<td>47</td>
</tr>
<tr>
<td>5. UAS SENSORS</td>
<td>49</td>
</tr>
<tr>
<td>5.1. Sensor Model</td>
<td>49</td>
</tr>
<tr>
<td>5.2. Sensor Field of Regard</td>
<td>50</td>
</tr>
<tr>
<td>5.3. Radar Declaration Range</td>
<td>51</td>
</tr>
<tr>
<td>6. IMM ANALYSIS</td>
<td>52</td>
</tr>
<tr>
<td>6.1. Simulation Steps</td>
<td>52</td>
</tr>
<tr>
<td>6.2. Simulation Parameters</td>
<td>53</td>
</tr>
<tr>
<td>6.3. Fixed Coordinated Turn vs Kinematically Constrained</td>
<td>54</td>
</tr>
<tr>
<td>6.4. Individual Kalman Filter Performance vs IMM</td>
<td>57</td>
</tr>
<tr>
<td>6.5. Perfect IMM Performance vs Regular IMM</td>
<td>61</td>
</tr>
<tr>
<td>6.6. Comparison of the IMM with Sensor Noise vs Zero Sensor Noise</td>
<td>64</td>
</tr>
<tr>
<td>6.7. Vertical Separation</td>
<td>66</td>
</tr>
<tr>
<td>6.8. Time to CPA</td>
<td>68</td>
</tr>
<tr>
<td>6.9. New IMM Prediction Method Analysis</td>
<td>70</td>
</tr>
<tr>
<td>6.10. IMM Analysis Summary</td>
<td>78</td>
</tr>
<tr>
<td>7. SAFETY EVALUATION</td>
<td>80</td>
</tr>
<tr>
<td>7.1. Risk evaluation methodology</td>
<td>80</td>
</tr>
<tr>
<td>7.2. Encounter Rate</td>
<td>81</td>
</tr>
<tr>
<td>7.3. Calculation of the Hazard State Covariance</td>
<td>83</td>
</tr>
<tr>
<td>7.4. Numerical Derivation of the Partial Derivatives</td>
<td>84</td>
</tr>
<tr>
<td>7.5. Hazard State Standard Deviations</td>
<td>85</td>
</tr>
<tr>
<td>7.6. Process Noise Analysis</td>
<td>87</td>
</tr>
<tr>
<td>7.7. Source for the Large Covariances on the IMM Prediction</td>
<td>94</td>
</tr>
<tr>
<td>7.8. Safety Evaluation Summary and Discussion</td>
<td>95</td>
</tr>
<tr>
<td>8. CONCLUSION</td>
<td>96</td>
</tr>
<tr>
<td>8.1. Summary of Accomplishments</td>
<td>96</td>
</tr>
<tr>
<td>8.2. Recommended Topics for Future Research</td>
<td>97</td>
</tr>
<tr>
<td>8.3. Closing</td>
<td>98</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>98</td>
</tr>
<tr>
<td>A. INERTIAL TO RELATIVE FRAME TRANSFORMATION</td>
<td>99</td>
</tr>
<tr>
<td>B. IMM DYNAMIC MODES</td>
<td>101</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Minimum RDR values</td>
<td>18</td>
</tr>
<tr>
<td>5.1 Sensor Performance Requirements [58, 59]</td>
<td>50</td>
</tr>
<tr>
<td>5.2 Sensor Performance Requirements</td>
<td>50</td>
</tr>
<tr>
<td>6.1 Mean RMS of CPA Error [ft] Estimation of $10^5$ Trajectories</td>
<td>61</td>
</tr>
<tr>
<td>6.2 Mean RMS of CPA Error [ft] Estimation of $10^5$ trajectories</td>
<td>63</td>
</tr>
<tr>
<td>7.1 Encounters type occurrence</td>
<td>82</td>
</tr>
<tr>
<td>7.2 $\sigma_{CPA}$ values for a single trajectory [ft]</td>
<td>86</td>
</tr>
<tr>
<td>7.3 $\sigma_{CPA}$ Values for Three Trajectories [ft]</td>
<td>87</td>
</tr>
<tr>
<td>7.4 Standard Deviation of Trajectory States - Regular Process Noise</td>
<td>91</td>
</tr>
<tr>
<td>7.5 Standard Deviation of Trajectory States - Reduced Process Noise</td>
<td>91</td>
</tr>
<tr>
<td>7.6 Standard deviation of the CPA position error [ft]</td>
<td>94</td>
</tr>
<tr>
<td>C.1 Process Noise Levels</td>
<td>110</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>ADS-B Airspace (from AOPA [6])</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Well Clear Threshold, and NMAC (not to scale)</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Linear Intruder Tracking</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Maneuvering Intruder Tracking</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Cumulative VFR Flight Hours, 2007-2008 Data (from Uncorrelated Encounter Model, 2008 [39])</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Encounter cylinder size (not to scale)</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>IMM Algorithm</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>WCT and NMAC</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>Horizontal Miss Distance</td>
<td>41</td>
</tr>
<tr>
<td>4.3</td>
<td>Trajectory with CPA and Vertical Separation Violations at Different Times</td>
<td>43</td>
</tr>
<tr>
<td>4.4</td>
<td>HMD Estimation</td>
<td>44</td>
</tr>
<tr>
<td>4.5</td>
<td>Future Maneuver Change</td>
<td>45</td>
</tr>
<tr>
<td>4.6</td>
<td>IMM Prediction Algorithm</td>
<td>46</td>
</tr>
<tr>
<td>4.7</td>
<td>IMM Prediction Data Structure</td>
<td>47</td>
</tr>
<tr>
<td>5.1</td>
<td>Spherical sensor measurements</td>
<td>49</td>
</tr>
<tr>
<td>6.1</td>
<td>Simulation Structure</td>
<td>53</td>
</tr>
<tr>
<td>6.2</td>
<td>Encounter Overview</td>
<td>54</td>
</tr>
<tr>
<td>6.3</td>
<td>Control Variables</td>
<td>55</td>
</tr>
<tr>
<td>6.4</td>
<td>CPA Estimation Error</td>
<td>56</td>
</tr>
<tr>
<td>6.5</td>
<td>IMM Mode Probability</td>
<td>56</td>
</tr>
<tr>
<td>6.6</td>
<td>Encounter Overview</td>
<td>57</td>
</tr>
<tr>
<td>6.7</td>
<td>Control variables</td>
<td>58</td>
</tr>
<tr>
<td>6.8</td>
<td>CPA Estimation Error Comparison</td>
<td>59</td>
</tr>
</tbody>
</table>
7.7 Scatterplot of CPA Position Error - Standard Noise $Q$ . . . . . . 92
7.8 Scatterplot of CPA Position Error - 0.1$Q$ . . . . . . . . . . . . 93
7.9 Scatterplot of CPA Position Error - 10$Q$ . . . . . . . . . . . . 93
ABSTRACT

This research presents new methods to apply safety standards to Detect and Avoid (DAA) functions for Unmanned Aircraft Systems (UAS), using maneuvering target tracking and encounter models.

Previous DAA research methods focused on predefined, linear encounter generation. The new estimation and prediction methods in this research are based on the target tracking of maneuvering intruders using Multiple Model Adaptive Estimation and a realistic random encounter generation based on an established encounter model.

When tracking maneuvering intruders there is limited knowledge of changes in intruder behavior beyond the current measurement. The standard Kalman filter (KF) with a single motion model is limited in performance for such problems due to ineffective responses as the target maneuvers. In these cases, state estimation can be improved using MMAE. It is assumed that the current active dynamic model is one of a discrete set of models, each of which is the basis for a separate filter. These filters run in parallel to estimate the states of targets with changing dynamics.

In practical applications of multiple model systems, one of the most popular algorithms for the MMAE is the Interacting Multiple Model (IMM) estimator. In the IMM, the regime switching is modeled by a finite state homogeneous Markov Chain. This is represented by a transition probability matrix characterizing the mode transitions. A Markov Chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the previous event.

This research uses the hazard states estimates (which are derived from DAA standards) to analyze the IMM performance, and then presents a new method to predict the hazard states. To reduce the prediction error, this new method accounts
for maneuvering intruders. The new prediction method uses the prediction phase in
the IMM algorithm to predict the future intruder aircraft states based on the current
and past sensor measurements.

The estimation and prediction methods described in this thesis can help ensure
safe encounters between UAS and manned aircraft in the National Airspace System
through improvement of the trajectory estimation used to inform the DAA sensor
certification process.
CHAPTER 1

INTRODUCTION

1.1 The Need for Detect and Avoid

In the 1990s and 2000s, unmanned aircraft systems (UAS) operations in the National Airspace System (NAS) were limited to mostly public functions like military operations and border security [20]. In 2012, as UAS civil and public applications expanded, the United States Congress mandated the Federal Aviation Administration, through the FAA Modernization and Reform Act, to develop necessary requirements for integration of UAS into the National Airspace System [64]. The FAA is estimating the fleet of small UAS (those that weigh less than 55 pounds) to increase from 32,800 in 2016 to over 540,000 in 2020 [4].

One of the challenges for the FAA to meet the UAS-NAS integration mandate is to ensure that safety targets are met. While a manned aircraft’s pilot relies on human vision to “see and avoid” non-cooperative intruders (those not employing a transponder or Automatic Dependent Surveillance-Broadcast, ADS-B) [23], a UAS requires sensors to “detect and avoid” intruder aircraft. If an intruder aircraft is cooperative, Air Traffic Control (ATC) may provide separation. A manned aircraft pilot could employ a Traffic Collision Avoidance System (TCAS) as a situational awareness aid to help the pilot detect an intruder then initiate an avoidance maneuver. Otherwise, it is the pilot’s responsibility to visually see the intruder and maneuver to maintain well clear. Anderson, et al., comprehensively detailed how the US Code of Federal Regulations (CFR) 14 CFR 91 relates to a pilot’s see and avoid responsibility [5].

The FAA Modernization and Reform Act of 2012 defines “sense and avoid capability” as “the capability of an unmanned aircraft to remain a safe distance from
and to avoid collisions with other airborne aircraft” [64]. “Sense and avoid” is a legacy term and is synonymous with DAA, which is the most current term used by the UAS standards community [58]. Non-cooperative DAA sensors include radar, Laser/Light Detection and Ranging (LIDAR), electro-optical (EO), acoustic, and infrared (IR) [66, 69]. The sensor must adequately provide information to the UAS DAA system, so the collision avoidance system can estimate whether or not a separation maneuver is required to ensure the UAS is well clear from intruder aircraft.

The FAA published 14 CFR 91.225 regarding the use of the ADS-B in May 2010. The final rule dictates that effective January 1, 2020, aircraft operating in airspace defined in 91.225 [2] are required to have an ADS-B system that includes a certified position source capable of meeting requirements defined in 91.227 [3].

1.1.1 Airspace Classes. Figure 1.1 shows the airspace classes for which the new regulation mandates the use of ADS-B out (the broadcast part of Automated Dependent Surveillance-Broadcast equipment).

![Figure 1.1. ADS-B Airspace (from AOPA [6])](image-url)
Under the new requirements [2, 3], ADS-B Out performance will be required to operate in:

- Class A, B, and C airspace.
- Class E airspace within the 48 contiguous states and the District of Columbia at and above 10000 ft MSL (Mean Sea Level), excluding the airspace at and below 2500 ft above the surface.
- Class E airspace at and above 3000 ft MSL over the Gulf of Mexico from the coastline of the United States out to 12 nautical miles.
- Around those airports identified in 14 CFR part 91, Appendix D.

For the remainder of the airspace (Class E below 10000 ft and within 2500 ft AGL of terrain higher than 7500 feet MSL), it will still be possible to operate without this equipment, thus resulting in the need for the UAS sensors.

### 1.2 Well Clear and Collision Avoidance

Since a UAS will not have a pilot on board, it will have to replicate the pilot vision through appropriate sensors. DAA sensors include radar, Laser/Light Detection and Ranging (LIDAR), Electro-Optical (EO), acoustic, and Infrared (IR) [66,69]. Until 2014, the concept of keeping aircraft well clear of each other had never been fully defined, despite “well clear” being a widely recognized term by the FAA and International Civil Aviation Organization (ICAO) [15].

“Well clear” is a subjective term in the right-of-way rules, 14 CFR 91.113 [1,5]. In 2011, Weibel, et al., proposed well clear as an objective separation standard [65]. In 2013, the Second FAA SAA Workshop concluded that the concept of well clear is an airborne separation standard [21].
RTCA is a federal advisory committee that provides technical standard recommendations in response to FAA requests. SC-228 is specifically tasked to determine MOPS for UAS. Recently, the Radio Technical Commission for Aeronautics (RTCA) Special Committee-228 (SC-228), in their Phase I Detect and Avoid (DAA) Minimum Operational Performance Standards (MOPS), defined a Well Clear Threshold (WCT) as the time to horizontal closest point of approach (CPA), or Tau (τ), of 35 seconds, a horizontal miss distance (HMD) of 4000 ft, and a vertical miss distance of 450 ft [58]. In this thesis, Tau, HMD, and vertical separation (or vertical miss distance) are considered “hazard states,” which define the hazard of a loss of well clear between two aircraft.

When an intruder cannot remain well clear, a collision avoidance maneuver is required to avoid a near mid-air collision (NMAC). NMAC limits are defined as 500 ft laterally and 100 ft vertically from the own aircraft [21] and are shown, along with the WCT, in Figure 1.2.

![Figure 1.2. Well Clear Threshold, and NMAC (not to scale)](image)

This thesis focused on the Well Clear Threshold, although the same methodology could be applied to NMAC and collision avoidance limits.
1.3 Problem Statement

This research deals with the detect and avoid problem describing new methods to apply safety standards in Detect and Avoid (DAA) functions for Unmanned Aircraft Systems (UAS), using maneuvering target tracking and encounter models.

The detection and avoidance of non-cooperative intruders must be guaranteed by the UAS. This requires compliance with the requirements outlined in the UAS DAA Minimum Operational Performance Standards (MOPS) and Air-to-Air Radar MOPS [58, 59]. However, the MOPS requirements and previous DAA research were done based on encounters between a UAS ownship and a threatening intruder aircraft with assumed linear trajectories.

The new methods in this work builds on previous work by focusing on the impact of maneuvering intruders. The new intruder estimation and prediction methods are based on the target tracking of maneuvering intruders using Multiple Model Adaptive Estimation, specifically the Interactive Multiple Model Algorithm.

1.3.1 Target Tracking. The DAA system of the UAS (the own aircraft) must detect a non-cooperative intruder aircraft (a target), then estimate its trajectory. Uncertainty in the measurements due to sensor noise will lead to poor state estimates when the intruder is first detected by the sensors, but as the DAA system gets more sensor measurements, the trajectory estimation error decreases.

When tracking maneuvering intruders (with nonlinear trajectories), there is limited knowledge of changes in intruder behavior beyond the current measurement [49]. The standard Kalman filter (KF) with a single motion model is limited in performance for such problems due to ineffective responses to dynamics changes as the target maneuvers. In these cases, state estimation can be improved using Multiple Model Adaptive Estimation (MMAE) to account for different potential intruder
behaviors [10].

1.3.2 Multiple Model Adaptive Estimation. In MMAE, it is assumed that the current active dynamic model is one of a discrete set of $r$ models, each of which is the basis for a separate filter [10]. These filters run in parallel to estimate the states of targets with changing dynamics. This work assumes that for each model, the prior probability of being in each state and the probabilities of switching from model $i$ to model $j$ in the next timestep are known. The latter can be represented as a transition probability matrix of a Markov Chain characterizing the mode transitions. A Markov Chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the previous event. MMAE are also called hybrid estimators since they have both continuous (noise) and discrete (mode or model) uncertainties [10].

1.3.3 IMM Algorithm. The optimal approach to filtering the states of a multiple model system requires running filters for every possible mode sequence. That is, for $r$ modes, $r^k$ optimal filters must be implemented to process measurements at timestep $k$, which is computationally unfeasible except for very simple systems over short time intervals. In practical applications of multiple model systems, one of the most popular algorithms for the MMAE is the Interacting Multiple Model (IMM) estimator. The IMM estimator is a suboptimal hybrid filter that has been shown to achieve an excellent compromise between performance and complexity [27, 46]. The model changes according to a finite-state, discrete-time Markov Chain [10, 55]. In the IMM, each state estimate is computed under each possible current model using $r$ filters, with each filter using a different combination of the previous mode estimates. The regime switching is modeled by a finite state homogeneous Markov Chain, with a-priori known transition probabilities.

1.4 Prior Work
There have been several papers that provide overviews of the DAA problem. Previous comprehensive reviews of the UAS integration into the NAS problem include [16, 17, 52]. Yu and Zhang present the state of DAA sensors, decision algorithms, path planning, and path following with a journal literature review [66]. Kuchar and Yang outlined what at the time was an overview of air traffic conflict detection and resolution models [40].

Much of the previous work relating to DAA safety performance has tended to focus on risk ratio safety studies that derive from the development of TCAS. McLaughlin and Zeitlin presented a safety study that used encounter models to build collision avoidance risk ratios to determine the safety of TCAS version 6.4 [50]. Esplinadle, et al., described an MIT Lincoln Lab (MIT LL) safety study that used encounter models to build collision avoidance risk ratios to determine the safety of TCAS version 7.1 [19]. The Second DAA Workshop determined, using methodology described in the International Civil Aviation Organization (ICAO) Doc 9689, that UAS DAA systems should have two target levels of safety (TLS) based on catastrophic collision risk ratios: $10^{-9}$ midair collisions (MAC) per flight hour (FH) for cooperative airspace (where transponders are required) and $10^{-7}$ MAC/FH for all other airspace [21, 29].

A significant amount of work in the UAS DAA was done at the MIT LL, by Kochenderfer, et al., which developed an aircraft encounter model used to evaluate safety of collision avoidance systems [37–39]. This encounter model was the basis of the randomly generated encounters that are used in this thesis, and it is going to be further explained in the following chapter. Recently, a new version of the Correlated Encounter Model 2.0 has been released in 2018, by Underhill et al. with significant improvements [63].

Kochenderfer, Chryssanthacopoulos, and Billingsley of the MIT LL also looked at state uncertainty of a collision avoidance system, quantifying safety as probabil-
ity of NMAC accounting for avoidance maneuvers, and applying Markov decision processes for collision avoidance [11]. Heisley, et al., of the MIT LL developed an architecture with a future intent to test and certify DAA systems [26]. Owen, et al., of the MIT LL demonstrated and flight tested an approach to developing DAA radar models for requirements derivations that employed a phased-array technology [51]. Edwards and Owen of the MIT LL validated a radar-based DAA concept through modeling and flight test [18]. One of the oldest studies on the application of adaptive estimation was done by the MIT LL [13].

Also, the Air Force Research Laboratory (AFRL) and the Air Force Institute of Technology (AFIT) have made many contributions to DAA. In late 2006, AFRL flight tested an early DAA system based on EO cameras combined with the self-maneuvering passive ranging techniques. In 2009, AFRL conducted a flight test of their Multi-Sensor Integrated Conflict Avoidance (MuSICA) DAA system which included sensor fusion from ADS-B, TCAS, radar and EO [14]. In addition, AFIT and AFRL researched UAS collision avoidance trajectories that minimized the deviation from intended flight path while using a particle filter to track multiple intruders [62].

Some other prior work of note include the following. Kim, et al., of the Korean Pusan National University designed a 3D EO system for small UASs using a Kalman filter, Sequential Quadratic Programming, and Linear Parameter Varying approaches for tracking and measurement error reduction [36]. Genovese determined that IMM performance is dependent on filter model selection [25]. Lee, et al., of NASA Langley constructed a distributed traffic model to enable a probabilistic approach to risk assessment by computing collision rates based on Predator training missions in the Grand Forks Air Force Base area [41]. Munoz, et al., of NASA Langley presented DAIDALUS (Detect and Avoid Alerting Logic for Unmanned Systems), a reference DAA concept implementation, which is also outlined in Appendix G of the DAA
There has been a great amount of research done in the field of target tracking. For example, Li and Jilkov published a series of papers regarding the maneuvering target tracking problem [42–45]. Yaakov Bar-Shalom has published a book and several papers covering the MMAE/IMM methodology and performance [8–10,49].

Pulford from the University of Melbourne has published a survey on the maneuvering tracking methods [54]. Yuan et al., developed models and algorithms for tracking targets with coordinated turn motion [68], which were used later on this research. Additional research regarding coordinated turn models are shown in works by Yuan, et al. [67], and Zhai, et al. [70]. Radosavljevic researched the transition probabilities for the Interacting Multiple Model [55]. Silbert, et al., compared the state estimates of a Kalman filter to a Perfect IMM against a maneuvering target [60]. Some works using the Multiple Model approach for maneuvering targets are presented for different use scenarios as maneuvering targets in clutter [61], its use for maneuvering target tracking in GSM Networks [71] and the use of MMAE for inertial navigation during Mars entry [48]. There is some prior research regarding IMM performance. For example, Hwang, et al., conducted a comparative analysis of the hybrid estimation algorithms [28].

Recent work has been developed on new DAA technologies for small UAS (sUAS) applications. NASA researchers at the NASA Armstrong Flight Research Center flight tested a collision avoidance technology on a small UAS equipped with a micro ADS-B transponder [7]. Glaab and NASA Langley has conducted flight tests and experiments using Dedicated Short Range Communications (DSRC) [47]. NASA Ames Research Center published a Concept of Operations (ConOps) for NASA’s UAS Traffic Management (UTM) research initiative, focusing on safely enabling large-scale sUAS operations in low altitude airspace [53]. These preliminary findings evaluated
several essential sUAS prototype subsystems with the objective of determining the usability of DSRC systems developed for the automotive industry in potential DAA sUAS applications [47].

Jamoom’s work at the Illinois Institute of Technology’s Navigation Laboratory is the direct precursor to this research. Jamoom evaluated DAA sensor safety performance for large UAS using integrity and continuity as absolute measures of safety [31–35]. His research was based on intruder encounters that were assumed to be linear, either through constant relative velocity [31, 32, 34, 35] or constant relative linear acceleration [32, 33]. Jamoom used hazard state (Tau, HMD, vertical separation) estimate error covariances to establish the integrity risk of the DAA system not detecting an imminent loss of well clear, as well as the continuity risk of a false alert [31–35]. He analyzed his methodology using encounters that bordered on the WCT [31–35]. Jamoom also applied this methodology to multiple intruders [32].

This research describes a different approach than the prior work. First, this research uses the IMM to estimate intruder trajectories, which enables the estimator to account for maneuvering intruders and eliminates constant velocity and linear trajectory assumptions found in prior research. In addition, this research uses the MIT LL encounter model to generate maneuvering intruder trajectories for Monte Carlo analysis, which is an improvement over prior work that analyzed a few linear border case trajectories.

1.5 Thesis Contributions

The main research goal is to develop new methods for a UAS DAA system to estimate the trajectory of a maneuvering intruder and determine the safety performance of this system. In addition, this research aims to develop a new method for hazard state estimation, advancing previous work that used predefined, linear trajec-
tories [12,33]. There are five main contributions to knowledge in this thesis. These contributions are outlined in the following subsections.

1.5.1 Analysis of Maneuvering Encounter Trajectories to the DAA System. The first contribution of this work is the introduction of random trajectories using an established encounter model to account for realistic trajectories including maneuvers. While previous work used predefined trajectories or linear trajectories, this work takes advantage of the encounter model generated trajectories for the further development of the DAA estimation model.

1.5.2 Applied a Multiple Model Adaptive Estimation to the DAA Problem. Another contribution is the improvement of intruder tracking by the use of a Multiple Model Adaptive Estimation (MMAE) algorithm applied on the UAS DAA problem. The algorithm chosen among the MMAE was the Interactive Multiple Model (IMM).

1.5.3 Identified and Quantified the Interactive Multiple Model Error Sources. Another contribution is the assessment of the IMM’s different error sources when applied to the DAA problem. Before exploring the general performance of the algorithm for the problem itself, these error sources were identified and then explored in the analysis chapter, in order to quantify how much each one would influence the performance of a practical DAA system.

1.5.4 Developed a New Method for Hazard State Estimation. After using the hazard states estimation to analyze the IMM and general methodology performance, a new method for the hazard states estimation was developed. In order to reduce the prediction error due to the inherent maneuvers in this model, the new method accounts for maneuvering intruders. As opposed to the DAA MOPS hazard state formulas, which are based on linear and non-accelerated trajectories [58], the
new prediction method uses the IMM prediction phase in the algorithm to predict the future intruder aircraft states based on the current last sensor measurement.

1.5.5 Developed a Safety Evaluation Method. This work analyzed the potential for using the IMM in a safety analysis based on the loss of separation probability associated with the encounter rate. This is based on the estimated standard deviations generated by the IMM prediction method.

1.6 Thesis Outline

After this introductory chapter, Chapter 2 describes the dynamic modeling methodology required to employ the Interactive Multiple Model (IMM) algorithm, including how the MIT LL Encounter Model is used within this research. In Chapter 3, the MMAE is introduced as an approach to account for target tracking uncertainties and different maneuver regimes. The IMM is detailed as a practical MMAE algorithm for presenting a good compromise between complexity and performance. In Chapter 4, the Detect and Avoid (DAA) Minimum Operational Performance Standards (MOPS) definition for DAA Well Clear (DWC) is used to derive states necessary to define the hazard associated with a loss of well clear. Chapter 5 details the sensor models used in the following chapters. In Chapter 6, the analysis of different aspects of the IMM algorithm are presented. The analysis starts with a comparison of the individual Kalman filters that correspond to each individual mode. It continues with an IMM error source analysis and finishes with the new IMM prediction performance as well as overall performance in the DAA problem. Chapter 7 introduces the methodology for a safety analysis based on the encounter rate associated with the probability of Loss of Well Clear. Finally, Chapter 8 provides conclusions and opportunities for future research.
CHAPTER 2
DYNAMIC MODELS FOR MANEUVERING INTRUDERS

This chapter describes the dynamic modeling methodology required to employ the Interactive Multiple Model (IMM) algorithm. It will first describe the constant velocity assumptions currently used, the value of modeling maneuvering intruders and how it relates to target tracking research. Next, it introduces the MIT LL Encounter Model and explains how it relates to this dynamic modeling methodology. Finally, it presents maneuvering intruder dynamics estimator and the component Kalman filters that correspond to each individual IMM mode (or model). This lays the foundation for the next chapter, development of the IMM algorithm for the maneuvering intruder problem.

2.1 Target Tracking: Maneuvering Trajectory vs Constant Velocity

Classical tracking methods using a single sensor involve a dynamic model of the target and state estimation based on measurements obtained by the system's sensor. In the case where the dynamic model of the target is well known, a Kalman filter is usually employed [42]. In Figure 2.1, a target tracking of a linear target is presented.

When tracking maneuvering intruders (with nonlinear trajectories), there is limited knowledge of changes in intruder behavior. An example of a nonlinear maneuvering intruder with a linear estimation of its future trajectory is depicted in Figure 2.2. The limited knowledge of intruder behavior adds to the measurement uncertainty already present due to sensor noise [49]. The standard Kalman filter (KF) with a single motion model is limited in performance for such problems due to ineffective responses to dynamics changes as the target maneuvers.
In these cases, state estimation can be improved using Multiple Model Adaptive Estimation (MMAE) to account for different potential intruder behaviors [10]. Among MMAE estimators, the IMM algorithm will be explored in the next chapter of this thesis and has been shown to be one of the most cost-effective estimation algorithms [30, 46, 49]. It has been successfully implemented in maneuvering target tracking applications, including some air traffic control systems [30].
2.2 Encounter Model

The methodology in this thesis will apply an encounter model in three ways: 1) generating intruder trajectories, 2) defining the modes (dynamic models) of the IMM algorithm, detailed in Section 3.3, and 3) generating the values of the Transition Matrix of the Markov Chain, detailed in Section 3.6.

This research will employ the MIT LL Encounter Model, which is based on transponder-equipped aircraft using a Visual Flight Rules (VFR) Mode A code (code 1200) observed by radars across the United States [39]. Encounter models are simulations that rely on models that accurately reflect the geometries and dynamics of aircraft encounters at close range. This model is representative of encounters between a cooperative aircraft (employing a transponder/ADS-B) and conventional non-cooperative aircraft, which will be flying under VFR and will behave similar to those using a VFR transponder code. The model uses dynamic variables to construct random aircraft trajectories that are statistically similar to those observed in the radar data as depicted in Figure 2.3. These will be useful for generating encounters that are both realistic and with the same frequency of maneuvers that are expected for such encounters. The encounter model gives as outputs the initial encounter model conditions [39]:

- Airspace class, $A$
- Flight level, $L$
- Horizontal velocity, $v$
- Horizontal linear acceleration, $\dot{v}$
- Vertical velocity, $\dot{z}$
- Turn rate, $\dot{\psi}$
and the control variables for each timestep:

- Linear acceleration, $\dot{v}$
- Turn rate, $\dot{\psi}$
- Vertical velocity, $\dot{z}$

Due to the wide variety of possible dynamic model implementations, details for trajectory construction are not provided with the encounter model. This thesis uses point-mass kinematics to update the aircraft states. In order to generate the encounter, the user must define the initial position and heading angle of the intruder. This thesis will use an “encounter cylinder” (detailed in the next subsection) to generate these user-defined initial conditions.

After the user defines the initial position and heading angle on the encounter cylinder, the simulated intruder trajectory is initialized with the encounter model.
initial conditions. Then the new trajectory states are updated based on the control variables generated, and the process is repeated through the point of closest approach.

2.2.1 Encounter Cylinder. The initial position of the intruder aircraft is randomly placed on the surface of an “encounter cylinder” centered on the own aircraft, at a random heading angle. With the random angle assigned, the horizontal velocity \( \dot{v} \) and horizontal linear acceleration \( \ddot{v} \) are converted to ownship-centered Cartesian horizontal position and velocity \( x, y, \dot{x}, \dot{y} \) intruder states. The probability of being initialized on the top, bottom, or side surfaces of the encounter cylinder is proportional to the surface area of each. Sampling rejection is applied for the initial position conjugated with the initial control variables from the encounter model. If the intruder is not in an inward trajectory from the surface, the initialization is repeated.

The encounter cylinder is assumed to have a ±3000 ft height, since, for alerting, the DAA MOPS treats radar-only intruders within 3000 feet vertically as co-altitude [58]. The radius of the cylinder is based on the minimum Radar Declaration Range (RDR) in clean air in the head-on direction, according to the Air-to-Air Radar MOPS [59]. The Air-to-Air MOPS defines the RDR as the “minimum value for the maximum range at which the track accuracy requirements for a radar generated intruder need to be met.” The RDR is dependent on intruder category, which is based on speed and size. The intruder categories are as follows [59]:

- Small, which includes gliders, balloons, etc., with a representative true airspeed up to 100 knots.
- Medium, which includes single-engine aircraft with a representative true airspeed up to 130 knots.
- Large, which includes dual-engine and larger aircraft with a representative true airspeed up to 170 knots.
For an ownship capable of a standard turn rate of 3 degrees/second, the RDR in clean air in the head-on direction should be according to Table 2.1 [59]:

<table>
<thead>
<tr>
<th>Intruder Category</th>
<th>Max Airspeed</th>
<th>Minimum RDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>100kt</td>
<td>5.4NM</td>
</tr>
<tr>
<td>Medium</td>
<td>130kt</td>
<td>6.0NM</td>
</tr>
<tr>
<td>Large</td>
<td>170kt</td>
<td>6.7NM</td>
</tr>
</tbody>
</table>

The final encounter cylinder with its variable size is shown in Figure 2.4.

Figure 2.4. Encounter cylinder size (not to scale)

2.3 Kalman Filters

The Kalman filter has long been regarded as the optimal solution to many tracking and data prediction tasks [10, 24], such as Air Traffic Control (ATC), powerfully combining information in the presence of uncertainty.

The algorithm works in a two-step process [24]. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (corrupted with some error, which includes random noise) is observed, then in the update step, these estimates
are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its covariances, with no additional past information being required. The IMM methodology presented in the next chapter will apply multiple Kalman filters running in parallel, one for each predicted intruder aircraft motion model.

In a Kalman filter, each state space model can be expressed with equations the following form [24]:

\[ x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + q_{k-1} \]  \hspace{1cm} (2.1)

\[ z_k = H_kx_k + r_k \]  \hspace{1cm} (2.2)

where:

- \( x_k \in \mathbb{R}^n \) are the system states (intruder Cartesian position, velocity, and acceleration in inertial coordinates) at timestep \( k \).
- \( F_{k-1} \) is the transition matrix of the dynamic model.
- \( G_{k-1} \) is the transition matrix of the dynamics of the own aircraft.
- \( u_{k-1} \) are the own aircraft states in inertial coordinates.
- \( q_{k-1} \sim N(0; Q_{k-1}) \) is the process noise at timestep \( k - 1 \).
- \( z_k \in \mathbb{R}^m \) is the cartesian relative intruder measurement at timestep \( k \).
- \( H_k \) is the measurement model matrix.
- \( r_k \sim N(0; R_k) \) is the measurement noise on timestep \( k \).
• The initial distribution for the state is \( x_0 \sim N(0; \hat{P}_0) \), where \( \hat{P}_0 \) is typically large based on the lack of initial knowledge about the target.

As previously mentioned, the Kalman filter has two steps: the **prediction** step, where the next state of the system is predicted given the previous measurements, and the **update step**, where the current state of the system is estimated given the measurement at that timestep. The equations for these steps are as follows [10]:

**Prediction:**

\[
\bar{x}_k = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1} \\
\bar{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}
\]

**Update:**

\[
S_k = H_k\bar{P}_kH_k^T + R_k \\
K_k = \bar{P}_kH_k^T S_k^{-1} \\
\hat{x}_k = \bar{x}_k + K_k(z_k - H_k\bar{x}) \\
\hat{P}_k = \bar{P}_k - K_k S_k K_k^T
\]

where:

• \( \bar{x}_k \) is the predicted state mean at the timestep \( k \) before the measurement.

• \( \hat{P}_k \) is the predicted state covariance at the timestep \( k \) before the measurement.
• $\hat{x}_k$ is the predicted state mean at the timestep $k$ after the measurement.

• $\hat{P}_k$ is the predicted state covariance at the timestep $k$ after the measurement.

• $S_k$ is the measurement prediction covariance at timestep $k$.

• $K_k$ is the filter gain, which tells how much the predictions should be corrected at timestep $k$.

2.4 Relative Intruder State

Different from fixed radar applications (air traffic control as an example) in which the tracker is static, the UAS DAA system will contain tracking sensors mounted on the ownship unmanned air vehicle. In this way, standard maneuvers such as a coordinated turn (CT) are viewed differently in the aircraft relative frame. The transitions between these reference frames must be considered.

The inertial to relative frame transformation is detailed in Appendix A. The final state-state transition equation (2.9) is analogous to the Kalman filter state prediction equation (2.3).

$$x_{rel}^{k+1} = F_I^k x_{rel}^k + G_k u_k$$  (2.9)

where:

• $x^{rel}$ is the state vector of the intruder aircraft in the own aircraft-centered relative frame.

• $F^I$ is the state transition matrix of the intruder aircraft.
2.5 Summary of Dynamic Models for Maneuvering Intruders

This chapter described the dynamic modelling to be used in the IMM algorithm. Starting with the maneuvering target problem, it was shown how the Encounter Model is used to generate the intruder trajectories while in the next chapter it will be used to generate the Markov Chain Matrix along the modes/models for the Kalman filters. The concept of the Encounter Cylinder was introduced, in which a random position of the intruder aircraft is used to generate the encounter, with a variable maximum range depending on the aircraft intruder category. Finally, the Kalman filters with the relative ownship-centered intruder state approach were also introduced.
CHAPTER 3
INTERACTIVE MULTIPLE MODEL

In a target tracking problem, system model parameters are only partially known and must be estimated through sensor measurements and any available previous knowledge about the target dynamics. One possible approach to account for target tracking uncertainties and different maneuver regimes is to use Multiple Model Adaptive Estimation (MMAE). Since an optimal MMAE would be computationally unfeasible, the IMM is proposed as a DAA estimator for maneuvering targets since it is shown to present a good compromise between complexity and performance.

In this chapter after introducing the MMAE, the IMM will be presented in detail, including its dynamic modes and the Markov Chain matrix. Finally, the Perfect IMM filter is introduced as an analysis tool for the IMM estimator.

3.1 Multiple Model Adaptive Estimation

In a typical state estimation problem, the uncertainties consist of additive white noises with known statistical properties. The system model (consisting of the state transition matrix, measurement matrix, input and noise covariance) is assumed to be known. In the target tracking problem, the mentioned parameters are only partially known and are time-varying. The approach for the use of a Multiple-Model Adaptive Estimation (MMAE) technique is to “adapt” to certain types of uncertainties beyond only additive white noises (process and measurement) [10, 55]. The MMAE uses a parallel bank of filters to provide multiple estimates, where each filter corresponds to a unique dynamic model. These filters (or modes) can differ in noise levels or even in their structure. The MMAE, which uses all of these filters, is called a hybrid estimator since they have both continuous (noise) and discrete (modes) uncertainties. There are likelihood functions that give the associated hypotheses that
each filter is the correct one, and the final estimation (state and covariance) is based on these modes running in parallel with this probabilistic weighting.

3.2 Basic Concepts of the IMM

The optimal approach to filtering the states of a multiple model system requires running filters for every possible model sequence. That is, for \( r \) models, \( r^k \) optimal filters must be implemented to process measurements at timestep \( k \). This is almost always computationally unfeasible. In practical applications of multiple model systems, one of the popular practical algorithms for the MMAE is the Interacting Multiple Model (IMM) estimator.

The IMM estimator is a suboptimal hybrid filter that has been shown to achieve an excellent compromise between performance and complexity [27,46], whose model changes according to a finite-state, discrete-time Markov Chain [10,55]. In the IMM, each state estimate is computed under each possible current (timestep \( t \)) model using \( r \) filters, with each filter using a different combination of the previous (timestep \( t - 1 \)) model-conditioned estimates (which are the mixed initial conditions). The regime switching is usually modeled by a finite state homogeneous Markov Chain, with a-priori known transition probabilities. This algorithm can be divided into five stages: (1) calculation of mixing probabilities, (2) computation of mixed initial states and covariances, (3) filtering, (4) mode probability update and (5) state estimate combination (output only, stage five is not part of the recursion). A flowchart of the IMM algorithm is depicted in Figure 3.1 and it is important to note that even though it depicts only two different modes/filters (\( KF1 \) and \( KF2 \)) there can be as many modes as required.
The terms in Figure 3.1 are:

- \( KF_1 \) and \( KF_2 \) are examples of models in the algorithm (can vary in number).
- \( x_n(k-1|k-1) \) and \( P_n(k-1|k-1) \) are the previous states and covariances, at timestep \( k-1 \), for each individual mode \( n \).
- \( x_{nm}(k-1|k-1) \) and \( P_{nm}(k-1|k-1) \) are the mixed states and covariances, at timestep \( k-1 \), for each individual mode \( n \).
- \( x_n(k|k) \) and \( P_n(k|k) \) are the calculated states and covariances, at timestep \( k \), for each individual mode \( n \).
- \( x(k|k) \) and \( P(k|k) \) are the combined states and covariances, at timestep \( k \).
- \( \mu(k) \) is the mode probability vector at timestep \( k \).
- \( M(k|k) \) is the mixing probability matrix at timestep \( k \).
• $\Lambda(k)$ is the likelihood function at timestep $k$.

• $z(k)$ is the DAA sensor measurement at timestep $k$.

3.2.1 The IMM Estimator. As presented in the previous section, the algorithm cycle consists of the following steps:

(1) Calculation of mixing probabilities:

$$M_{ij}(k-1)(k-1) = \frac{1}{\bar{c}_j} p_{ij}(k-1) \mu_i(k-1) \quad i, j = 1, ..., r \quad (3.1)$$

where the normalizing constants are:

$$\bar{c}_j = \sum_{i=1}^{r} p_{ij}(k-1) \quad (3.2)$$

and $r$ are the number of modes, $p_{ij}$ is the Markov Chain Matrix and $M$ is the mixing probability matrix.

(2) Mixing: starting with $\dot{x}_i(k-1)(k-1)$, the mixed initial condition for each filter is calculated.

$$\dot{x}_{jm}(k-1)(k-1) = \sum_{i=1}^{r} \dot{x}_i(k-1)(k-1)M_{ij}(k-1)(k-1) \quad j = 1, ..., r \quad (3.3)$$

and the covariance

$$P_{jm}(k-1|k-1) = \sum_{j=1}^{r} M_{ij}(k-1|k-1)\{P^i(k-1|k-1) + [\dot{x}^i(k-1|k-1) - \dot{x}_{jm}(k-1|k-1)]' \cdot [\dot{x}^i(k-1|k-1) - \dot{x}_{jm}(k-1|k-1)] + \} \quad i, j = 1, ..., r \quad (3.4)$$
(3) Filtering: using the mixed inputs with the measurement \( z(k) \) to get \( \hat{x}_j(k|k) \) and \( \hat{P}_j(k|k) \) the from each Kalman filter corresponding to each different mode. The likelihood functions are given by:

\[
\Lambda_j(k) = p[z(k)|m_j(k), Z^{k-1}] \quad j = 1, \ldots, r
\]

are computed using the mixed initial conditions and covariances, that will give:

\[
\Lambda_j(k) = \mathcal{N}[z(k); \hat{z}_j[k|k-1; \hat{x}_{jm}(k-1)(k-1)], S_j[k; P_{jm}(k-1|k-1)]] \quad j = 1, \ldots, r
\]

(4) Mode probability update: this will give the likelihood of each model in the algorithm output.

\[
\mu_j(k) = \frac{1}{c} \Lambda_j(k) \bar{c}_j \quad j = 1, \ldots, r
\]

where \( c_j \) is shown in Equation 3.2 and

\[
c = \sum_{j=1}^{r} \Lambda_j(k) \bar{c}_j \quad j = 1, \ldots, r
\]

(5) Estimate and covariance combination: combination of the model estimates for output purposes, since it is not part of the algorithm recursion.

\[
\hat{x}(k|k) = \sum_{j=1}^{r} \hat{x}_j(k|k) \mu_j(k) \quad j = 1, \ldots, r
\]
\[
\mathbf{P}(k|k) = \sum_{j=1}^{r} \mu_j(k) \{ \mathbf{P}_j(k|k) + [\hat{x}_j(k|k) - \mathbf{\hat{x}}(k|k)][\hat{x}_j(k|k) - \mathbf{\hat{x}}(k|k)]' \} \quad j = 1, \ldots, r
\]

where:

- \( \mathbf{P}_j \) is the covariance of mode \( j \)
- \( \hat{x}_j \) is the state estimate of mode \( j \)
- \( \mu_j \) is the mode probability vector

### 3.3 IMM Mode Definition

Based on the various potential ways the MIT LL Encounter Model can incrementally build an intruder trajectory [39], models for straight level flight, turns, linear accelerations, and climbs and descents are required.

The initial approach was to define a different mode for each possible combination of control variable outputs from the Encounter Model. For the combination of the 3 different control variables, there are 8 different modes considering that each can be zero or non-zero.

The first CT filters considered were with fixed turn rates, based on a standard turn rate (defined as a 3° per second turn, which completes a 360° turn in 2 minutes) [22]. Adding to that, filters with kinematically constrained CT were added for comparison. The coordinated turn with a fixed turn rate required positive and negative turn rates, driving the total number of modes to 10. The following potential filters were considered, including two different types of CT filters:

- Constant Velocity 2D
- Coordinated Turn (fixed turn rate +) 2D
- Coordinated Turn (fixed turn rate -) 2D
- Constant Velocity 3D
- Constant Linear Acceleration 2D
- Coordinated Turn (fixed turn rate +) 3D
- Coordinated Turn (fixed turn rate -) 3D
- Coordinated Turn (kinematically constrained model) 2D
- Constant Linear Acceleration 3D
- Coordinated Turn (kinematically constrained model) 3D

Even after analyzing Encounter Model data to find the most likely value for
the turn rate (which was about 3.5 degrees per second) for the fixed turn rate CT
modes, the better state estimation performance of the kinematically constrained CT
modes drove the IMM to give them more weighting. Since the fixed CT modes did not
improve estimation, they were dropped in favor of the kinematically constrained CT
modes. An analysis of the performance of these different modes is done in Chapter 6.3.

This results in six models/modes, three different modes for 2D movement and
the same three modes in 3D with the addition of the vertical velocity \( \dot{z} \) to account
for climb/descent:

- Mode 1 - Constant Velocity
- Mode 2 - Constant Velocity 3D (non-zero \( \dot{z} \))
- Mode 3 - Coordinated Turn
• Mode 4 - Coordinated Turn 3D (non-zero \( \dot{z} \))

• Mode 5 - Constant Linear Acceleration

• Mode 6 - Constant Linear Acceleration 3D (non-zero \( \dot{z} \))

Modes 1 and 2, have six states:

\[
x = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T
\]  

(3.11)

Modes 3 through 6 have nine states:

\[
x = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}^T
\]  

(3.12)

The implementations of the Constant Velocity (CV) and Constant Acceleration (CA) filters are relatively straightforward. For the kinematically constrained Coordinated Turn (CT) model (modes 3 and 4), the state transition matrix is defined to model a constant-speed turn along the trajectory which is defined by the state estimates of velocity and acceleration. Then, the transition matrices are based on the dynamics for the intruder \( F^I \) and own aircraft \( F^O \) are:
\[ F^I = \begin{bmatrix}
1 & \frac{\sin(\omega T)}{\omega} & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\sin(\omega T)}{\omega} & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 & 0 \\
0 & 0 & 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \] (3.13)

\[ F^O = \begin{bmatrix}
0 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & T & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \] (3.14)
\[ G = F^I - F^O = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} - T & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos(\omega T) - 1 & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{\sin(\omega T)}{\omega} - T & \frac{(1-\cos(\omega T))}{(\omega^2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos(\omega T) - 1 & \frac{\sin(\omega T)}{\omega} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]  

(3.15)

where \( T \) is the simulation timestep and \( \omega \) is the turning rate.

\[ u = \begin{bmatrix} 0 & \dot{x}_{own} & 0 & 0 & \dot{y}_{own} & 0 & 0 & \dot{z}_{own} & 0 \end{bmatrix}^T \]  

(3.16)

where \( \dot{x}_{own}, \dot{y}_{own} \) and \( \dot{z}_{own} \) are the own aircraft velocity components. Here, we have assumed that the own aircraft has a constant velocity vector during the encounter. Strictly, this assumption is not necessary at this point in the general development. However, we write it here as an example because we will use it the performance analysis in later chapters.

The turning rate \( \omega \) is defined as:

\[ \omega = \frac{\|a\|}{\|v\|} = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \left/ \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \right. \]  

(3.17)

where \( a \) and \( v \) are the inertial acceleration and velocity vectors.
Considering the case where we have only relative position measurements, for example, the measurement matrix is:

\[
H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\] (3.18)

The matrices for the other modes are detailed in Appendix B.

### 3.4 IMM Estimation Error Sources

The IMM algorithm has estimation errors that are not directly related with its calculated covariances. The total error in the state estimation and trajectory prediction will be a sum of different error sources. Some will depend on the quality of the implementation of the algorithm, others on the inherently noisy measurements from the DAA system. In summary, we can divide the main contributors of IMM estimation error into four parts:

- **Modeling error**: This is where a maneuver, by its nature, does not follow the assumptions of the standard dynamic models used in the IMM [46].

- **Sensor noise**: Random noise on the sensor measurements.

- **Mode transition**: A relatively large estimation error is needed for the algorithm to recognize sudden changes in system modes, i.e. the adaptation might not be rapid enough. To quantify the magnitude of this effect, the next section will introduce the Perfect IMM, an analysis tool in which the true mode of the target is assumed to be known, for performance comparison.

- **Prediction error**: During an encounter, the IMM cannot predict future in-
truder pilot intent to maneuver. As a result, there is an additional error of the predicted states from the actual states, only known after an encounter has ended.

A real IMM, even with perfect measurements, would still have uncertainty in trajectory estimation due to modeling, mode transition and prediction errors.

3.5 Perfect IMM

A Perfect IMM (PIMM) filter is a theoretical IMM filter that cannot be realized for any real system, but is a useful analysis tool. The output from the intruder is defined by the encounter model, and the PIMM switches to the correct motion model at each mode change. Since the PIMM always selects one motion model as soon as the mode change occurs, it defines the lower bound on the IMM estimation error for any given sensor error model, making it useful for comparison studies of IMM performance. The importance of this lower bound is the knowledge that the IMM state estimation cannot improve beyond the PIMM results, given the same modes and conditions regardless of sensor quality.

3.6 Transition Matrix Probabilities

The IMM estimator assumes that the mode transition probabilities (which forms the Transition Matrix of the Markov Chain) governing the mode jumps is known. However, it is very difficult to determine the appropriate matrix quantities and identify a Markov transition law that optimally fits the unknown target motion [55]. Also, these Markov Chain Transition Matrix quantities are usually parameters selected during the design process of the algorithm [10].

Fortunately, for this UAS DAA application, there is the MIT LL Encounter model (based on actual radar observations [39]) to get realistic probabilities to pop-
ulate the Markov Chain Transition Matrix. To build the transition matrix, the MIT LL Encounter Model was run $10^6$ times and each mode transition at each timestep was recorded, in order to get all the likelihood of mode changes during the encounters. The resulting mode probability vector and Markov Chain Transition Matrix are as follows:

Initial mode probabilities:

$$\mu_{ip} = \begin{bmatrix} 0.55117 & 0.15393 & 0.10708 & 0.04781 & 0.09292 & 0.04709 \end{bmatrix} \quad (3.19)$$

Transition Matrix of the Markov Chain:

$$M_{ij} = \begin{bmatrix} 0.98038 & 0.00692 & 0.00562 & 0.00093 & 0.00508 & 0.00108 \\ 0.02536 & 0.96022 & 0.00246 & 0.00509 & 0.00200 & 0.00487 \\ 0.03263 & 0.00280 & 0.95063 & 0.00646 & 0.00624 & 0.00123 \\ 0.01280 & 0.01891 & 0.01456 & 0.94566 & 0.00172 & 0.00636 \\ 0.02551 & 0.00250 & 0.01098 & 0.00077 & 0.95314 & 0.00710 \\ 0.00926 & 0.01860 & 0.00204 & 0.01147 & 0.01049 & 0.94814 \end{bmatrix} \quad (3.20)$$

### 3.7 Process Noise

The process noise levels used in the modes in the IMM affect not only the final estimation results, but also the mode transition and mode probabilities. While small values can lead to slow mode transitions, higher values can lead to poor mode identification. The chosen values are dependent on the sensor noise levels and also the Markov chain matrix, and were selected based on the results of Monte Carlo runs, varying the process noise for different modes and optimizing it to minimize hazard state estimation error. A more detailed analysis of the chosen values for the process noise is done in Section 7.6.
3.8 Interactive Multiple Model Summary

The multiple model, or hybrid, system approach assumes the system to be in one of a finite number of modes. An optimal multiple model estimator would require an exponentially increasing number of filters, due to the conditioning for every mode history. The IMM is a practical algorithm that computes the state estimate for each possible current model using a mixing of the previous model-conditioned estimates, depending on the current model. No maneuver detection decision is needed since the algorithm performs a soft switching according to the latest mode probabilities. The best state estimates and covariances are based on these several modes running in parallel with a probabilistic weighting. The models and the Markov Chain matrix used in the IMM are based on MIT LL Encounter Model runs. Four different IMM estimation error sources were identified. The Perfect IMM switches to the correct motion model at each mode change, serving as a benchmark for the IMM estimation as it eliminates the mode transition error.
CHAPTER 4
HAZARD STATE ESTIMATION

In this chapter, the Detect and Avoid (DAA) Minimum Operational Performance Standards (MOPS) definition for DAA Well Clear (DWC) is used to derive states necessary to define the hazard associated with a loss of well clear. A UAS DAA system will use these “hazard states” to provide timely intruder alerts. The DAA MOPS is a technical standards document written by the Radio Technical Commission for Aeronautics (RTCA) Special Committee-228 (SC-228). In this chapter, first the hazard states are defined based on the DAA MOPS definition for a loss of DAA Well Clear. Then the three hazard states are introduced. They are time to horizontal CPA (or $\tau$), horizontal miss distance (HMD), and vertical separation ($z$). Finally, a new prediction method for these hazard states is presented based on the IMM algorithm.

4.1 DAA Well Clear

The DAA MOPS introduces the term DAA Well Clear (DWC) as “a temporal and/or spatial boundary around the aircraft intended to be an electronic means of avoiding conflicting traffic” [58]. The MOPS mathematically defines DWC in the following equation for a loss of DWC (LoWC) [58]:

$$\text{LoWC} = [0 \leq \tau_{mod} \leq \tau^{*}_{mod}] \cap [HMD \leq HMD^{*}] \cap [-z^{*} \leq z \leq z^{*}]$$ \hspace{1cm} (4.1)

with $\tau^{*}_{mod} = 35$ sec, $HMD^{*} = 4000$ft and $z^{*} = 450$ft.

These variables are:

- $\tau_{mod}$ is the Modified Time to Horizontal Closest Point of Approach with its threshold, $\tau^{*}_{mod}$.
- $HMD$ is the Horizontal Miss Distance with its threshold, $HMD^{*}$.
• \( z \) is the Vertical Separation with its threshold, \( z^* \).

Equation 4.1 means that a LoWC occurs when all three variables \((\tau_{\text{mod}}, HMD, z)\) are simultaneously violating their thresholds. The three thresholds \((\tau_{\text{mod}}^*, HMD^*, z^*)\) can be considered together as a Well Clear Threshold (WCT). In addition, the three variables that define the hazard associated with a LoWC \((\tau_{\text{mod}}, HMD, z)\), will be defined as \textit{hazard states}, with the caveat that this thesis will use Actual Time to Horizontal CPA, \( \tau_{\text{true}} \), instead of \( \tau_{\text{mod}} \), for reasons explained in the next section.

In this thesis, a LoWC will be defined by the following equation:

\[
\text{LoWC} = [0 \leq \tau \leq \tau^*] \cap [HMD \leq HMD^*] \cap [-z^* \leq z \leq z^*] \tag{4.2}
\]

with \( \tau^* = 35 \text{ sec} \), \( HMD^* = 4000\text{ft} \) and \( z^* = 450\text{ft} \) and where \( \tau = \tau_{\text{true}} \), the actual time to horizontal CPA.

Therefore, the \textit{hazard states} in this thesis are:

• \( \tau \), Actual Time to Horizontal CPA

• \( HMD \), Horizontal Miss Distance

• \( z \), Vertical Separation

For a LoWC, there must be a time where all violations in Equation 4.2 must simultaneously be true. As an example, if only HMD is violated, the intruder can still be well clear, safely above or below the intruder by several thousand feet.

Figure 4.1 depicts a lateral view of an encounter, showing the spacial elements of the WCT in relation to the much tighter near mid-air collision (NMAC) thresholds used for collision avoidance applications. The NMAC thresholds are 500 feet horizontally and 100 feet vertically [58]. Although the methodology in this thesis
will address the LoWC thresholds employed by a DAA system, it can also apply to NMAC thresholds used in collision avoidance applications.

![Figure 4.1. WCT and NMAC](image)

### 4.2 Time to Horizontal CPA

Tau is the time to horizontal CPA, based on horizontal ranges, not slant ranges. “Modified tau is the time to horizontal CPA with an added safety factor [56], including an approximation of potential intruder accelerations” [58]. Since the IMM already takes into account the estimated accelerations, the true tau will be introduced in Subsection 4.2.2.

#### 4.2.1 Modified Tau

The DAA MOPS uses modified tau to define a LoWC. It defines modified tau as follows for closing geometries [58]:

\[
\tau_{mod} = \frac{-(r^2 - DMOD^2)}{r \dot{r}} = \frac{DMOD^2 - r^2}{d_x v_{rx} + d_y v_{ry}}
\]  

(4.3)

where:

- \( r > DMOD \)
• $\tau_{mod} = 0$ for $r \leq DMOD$

• $\tau_{mod} = \infty$ for non-closing geometries where $r > DMOD$

• $r = \sqrt{d_x^2 + d_y^2}$ is the horizontal range between the aircraft

• $\dot{r} = \frac{d_x \cdot v_{rx} + d_y \cdot v_{ry}}{r}$ is the horizontal range rate between the aircraft

• $v_{rx} = \dot{x}_2 - \dot{x}_1$ is the relative horizontal velocity in the x direction

• $v_{ry} = \dot{y}_2 - \dot{y}_1$ is the relative horizontal velocity in the y direction

• $d_x = x_2 - x_1$ is the current horizontal separation on the x direction

• $d_y = y_2 - y_1$ is the current horizontal separation on the y direction

$DMOD$ is the distance modification, which is set equal to the $HMD$ threshold, $HMD^*$. If $DMOD$ does not equal $HMD^*$, the DAA MOPS explains that “alerts may oscillate on and off with unaccelerating ownship and intruder, which is an undesired behavior” [58]. The intent behind $DMOD$ has different explanations depending on the reference source. One of the explanations is from the TCAS II MOPS: “Distance Modification (DMOD) - Safety factor incorporated in range measurements to account for possible accelerations by the intruder. The value of distance modification varies with the sensitivity level for this own intruder set. The value is chosen such that a sustained acceleration of $g/3$ will produce this displacement in range threshold time” [56].

4.2.2 True Tau. Since this research intends to address maneuvering intruders and eliminate the assumption of linear, constant velocity intruder trajectories, this thesis will use true tau instead of modified tau.

The DAA MOPS defines actual time to horizontal CPA, $\tau_{true}$, as follows [58]:
\[ \tau_{\text{true}} = \max(0, -\frac{d_x v_{rx} + d_y v_{ry}}{v_{rx}^2 + v_{ry}^2}) \] (4.4)

where:

- \( v_{rx} = \dot{x}_2 - \dot{x}_1 \) is the relative horizontal velocity in the x direction
- \( v_{ry} = \dot{y}_2 - \dot{y}_1 \) is the relative horizontal velocity in the y direction
- \( d_x = x_2 - x_1 \) is the current horizontal separation on the x direction
- \( d_y = y_2 - y_1 \) is the current horizontal separation on the y direction

This thesis will utilize “true tau”, as defined in equation 4.4, and apply it in the proposed methodology in this chapter and subsequent chapters, in order to compare it to a new proposed methodology developed in subsequent chapters.

4.3 Horizontal Miss Distance

Figure 4.2 depicts the Horizontal Miss Distance (HMD) and the horizontal CPA \( r_{\text{CPA}} \), in a top-down view of an encounter.

![Figure 4.2. Horizontal Miss Distance](image-url)
The DAA MOPS, based on a linear trajectory with constant velocity, defines the HMD as follows [58]:

\[ HMD = \sqrt{(d_x + v_{rx}\tau)^2 + (d_y + v_{ry}\tau)^2} \]  

(4.5)

where:

- \( v_{rx} = \dot{x}_2 - \dot{x}_1 \) is the relative horizontal velocity in the x direction
- \( v_{ry} = \dot{y}_2 - \dot{y}_1 \) is the relative horizontal velocity in the y direction
- \( d_x = x_2 - x_1 \) is the current horizontal separation on the x direction
- \( d_y = y_2 - y_1 \) is the current horizontal separation on the y direction

4.4 Vertical Separation

The DAA MOPS defines current vertical separation, \( d_z \), as [58]:

\[ d_z = \text{abs}(z_1 - z_2) \]  

(4.6)

It also clarifies that “vertical separation for alerting performance requirements is based on relative altitudes,” which can be barometric for cooperative intruders or geometric for non-cooperative intruders [58]. It is relevant to emphasize that the condition for the LoWC in equation 4.2 shows that each one of the hazard states must be under its thresholds for a well clear violation. Figure 4.3 depicts an example of a trajectory where the vertical separation and CPA are violated but each one during different periods of the trajectory.
Figure 4.3. Trajectory with CPA and Vertical Separation Violations at Different Times

4.5 New Hazard State Prediction Method

To minimize prediction errors on the hazard states, a new method (instead of using the linear trajectory approximation) is to use the IMM prediction in the Kalman prediction step as the estimation for the hazard states. We will compare four different estimation and prediction combinations for the hazard states:

- True-MOPS: the true trajectory states using the MOPS equations 4.4, 4.5, and
4.6 for hazard state prediction.

- IMM-MOPS: the best IMM trajectory estimation and the MOPS equations 4.4, 4.5, and 4.6 for hazard state prediction.

- IMM-IMM: using the best IMM trajectory estimation and the new IMM prediction method to estimate future hazard states.

- True: only known after the encounter is completed (influenced by future mode switches).

The method can be used to estimate any of the defined hazard states, but HMD will be used as an illustrative example.

The True-MOPS, IMM-MOPS, and IMM-IMM trajectories are depicted in Figure 4.4. The difference between $HMD_{True-MOPS}$ (in black) and $HMD_{IMM-MOPS}$ (in blue) is due to the error in the IMM state estimation when compared to the true trajectory and states. The new HMD estimation method is shown as $HMD_{IMM-IMM}$ (in red). Under maneuvers, the hazard states estimation will be changing at each timestep.

![Figure 4.4. HMD Estimation](image)
Future aircraft intent (and possible maneuver changes) are unknown, so the true HMD can only be known after the encounter is completed. Using the knowledge of the current maneuver can improve the estimation based on the IMM prediction. After a maneuver occurrence in the future (as shown in Figure 4.5), the new estimation gets better as the IMM recognizes the maneuver and adjusts for a new subsequent prediction.

Figure 4.5. Future Maneuver Change

Without any sensor measurements in future (predicted) timesteps, the IMM algorithm predicts as described in on Figure 4.6:
Figure 4.6. IMM Prediction Algorithm

Since it is not possible to calculate the likelihood functions without the sensor measurements (λ and z(k) as shown in Figure 3.1), the approach is to use the normalizing factors (in equation 3.2) to update both the mixing probabilities (M(k|k)) and mode probabilities (μ(k)).

The changes start at step (3) Filtering, where each Kalman filter is calculated without its measurement input, thus only on its prediction phase and Equation 3.6 cannot be estimated. At step (4) Mode Probability Update, the new equation for μ_j(k) becomes:

$$\mu_j(k) = \sum_{i=1}^{r} p_{ij} \mu_i(k-1)$$  \hspace{1cm} (4.7)

which is similar to Equation 3.2.

This ultimately gives some weighting on the Markov Chain Matrix values, to account for the uncertainty of the possible future mode changes. Over time, the
mode probability tends to follow the probabilities of which mode is more likely to occur, based on the Markov Chain Matrix. A more detailed analysis of how these probabilities change will be done in following chapters.

4.5.1 Data Structure. For each timestep of the main IMM, the prediction algorithm is run until the end of the simulation. In comparison with the IMM, which deals with only the state estimation of the current timestep (diagonal of the matrix shown in Figure 4.7), there is a much larger dataset, estimating the states and covariances for each different filter preserving the IMM for all future timesteps.

![Figure 4.7. IMM Prediction Data Structure](image)

4.6 Hazard States Summary

In this chapter, three hazard states were defined based on the DAA MOPS definitions for a loss of DAA Well Clear. The hazard states are time to CPA ($\tau$), horizontal miss distance (HMD), and predicted vertical separation ($z$). These hazard states were defined as functions of the own aircraft and intruder current states.

The MOPS equations for the hazard states are based upon a linear trajectory with a constant velocity extension. Later chapters will show that a linear trajectory can be a problematic approximation for a nonlinear trajectory. To better estimate
nonlinear trajectories, a new hazard state prediction method was presented based on the prediction step within the IMM filters.
CHAPTER 5
UAS SENSORS

The intended function of the sensors aboard the UAS is to detect all airborne traffic within the sensor detection volume. The onboard system should complement other airborne surveillance sensors by providing detection of non-cooperative traffic [59]. For the purpose of this thesis, the analysis and simulations are done expecting non-cooperative intruders, without any equipment assumption for the intruder, and relying on the own aircraft’s DAA system as the only means of information regarding the maneuvering intruder.

5.1 Sensor Model

The DAA sensors measure (with an associated error) the intruder relative position in spherical coordinates. Taking from the DAA MOPS [58], these input measurements from a DAA radar includes relative slant range, relative range rate, relative bearing, and elevation angle \((\rho, \dot{\rho}, \theta, \phi)\), as in Figure 5.1.

Figure 5.1. Spherical sensor measurements

Table 2-5 of the Air-to-Air Radar MOPS [59] as well as Appendix Q of the DAA MOPS [58] outlines sensor performance characteristics in terms of standard
deviations of range, range rate, azimuth, and elevation. Based on the requirements outlined in both the DAA MOPS [58] and the Air-to-Air Radar MOPS [59], the DAA sensor will have a range as described in Table 2.1, with the sensor performance outlined in Table 5.1 at a sample rate of 1 Hz.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range Error</td>
<td>70 ft</td>
</tr>
<tr>
<td>Range Rate Error</td>
<td>10 ft/sec</td>
</tr>
<tr>
<td>Azimuth Error</td>
<td>1°</td>
</tr>
<tr>
<td>Elevation Error</td>
<td>1°</td>
</tr>
</tbody>
</table>

The transformation from spherical to Cartesian is non-linear but the MOPS also provides a table of values for 95% track accuracy presented in Cartesian [58]. This is shown in Table 5.2.

<table>
<thead>
<tr>
<th>Perf. Metric</th>
<th>Metric Threshold (95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal position</td>
<td>$125(\rho - 1\text{NM}) + 250'$ (for $\rho = 1 \text{ NM}$ through $6.7 \text{ NM}$)</td>
</tr>
<tr>
<td>Vertical position</td>
<td>$100(\rho - 1\text{NM}) + 150'$ (for $\rho = 1 \text{ NM}$ through $6.7 \text{ NM}$)</td>
</tr>
</tbody>
</table>

Based on the values in the table for accuracy in Cartesian coordinates, the standard deviations are extracted at each timestep considering the variable intruder aircraft position range along the encounter.

5.2 Sensor Field of Regard

The field of regard (FOR) is the total area that can be captured by a movable sensor. According to the Air-to-Air MOPS, sensors should be able to identify targets within the following field of regard (FOR) [59]:

• ± 110° with respect to the longitudinal axis

• ± 15° vertically referenced to the flight path

Considering that some of the worst-case scenarios found in preliminary analysis were from trajectories where the intruder was overtaking the own aircraft from behind, we expect improvement on the estimation errors when limiting the encounters to the FOR region. Also, an analysis of the likelihood of overtaking encounters can be made using the data from the Encounter Model, to evaluate the relevance of these FOR values. Due to the field of regard requirements from the MOPS [59], some encounters previously generated will have the intruder Aircraft lying outside of the FOR from the onset of the encounter. In order to adapt to the FOR requirements, new requirements for the initial intruder aircraft at its random initial position at the encounter cylinder are added to comply with the FOR in those analysis.

5.3 Radar Declaration Range

The Radar Declaration Range (RDR) was introduced in Section 2.2.1 to define the “encounter cylinder” dimensions. It is defined as the minimum value for the maximum range at which the track accuracy requirements are met. It is important to note that these would be the reference values at which a target tracked by the DAA sensors would be used by the UAS pilot [59].
CHAPTER 6
IMM ANALYSIS

This chapter contains analyses of different aspects of the IMM algorithm. First is a comparison with the individual Kalman filters that correspond to each individual mode. Next, is a comparison of the IMM vs the Perfect IMM filter. The effect of the sensor noise is also presented and the new, IMM-based hazard state prediction method is compared against the calculations done according to the MOPS [58].

6.1 Simulation Steps

The idea is to generate the intruder aircraft random trajectory using the outputs from the encounter model. The encounter model initial conditions are scalars (velocity, linear acceleration, turn rate, and the vertical velocity), and it is up to the user to provide an initial position and direction in relation to the own aircraft.

The initial position of the intruder aircraft is randomized on the surface of an “encounter cylinder” centered on the own aircraft, at a random heading angle. The dimensions of the cylinder are given by the minimum radar range [59] and the maximum height difference at which an “encounter” must be considered [58]. The surface areas of the top, bottom and side will define the proportion of encounters originating on that surface. Sampling rejection is applied for the initial position conjugated with the initial control variables from the encounter model. If the intruder is not in an inward trajectory from the surface, the initialization repeats until a suitable encounter is generated.

Using the encounter model output, the intruder trajectory is built using point-mass kinematics to update the aircraft states. The trajectory is then inserted into the IMM estimation algorithm, using the encounter model outputs as inputs for the true intruder trajectory. Finally, this process concludes with simulation outputs for
analysis as shown in Figure 6.1.

![Simulation Structure Diagram]

Figure 6.1. Simulation Structure

There are also flags within the simulation for separating out trajectories of interest, such as a WCT violation and an exit from the field of regard.

6.2 Simulation Parameters

For all the different encounters simulated, the own aircraft was assumed to have a linear trajectory with constant velocity of 200 kt (the MOPS modeling assumption for a UAS ownship below 10,000 ft altitude is an airspeed limit of 200 kt true airspeed [58]), while the generated trajectory for the intruder aircraft is randomly generated by the encounter model. Each of the trajectories were simulated according to the following conditions:

- Own aircraft: linear trajectory, constant velocity of 200 knots
- Intruder aircraft: randomly generated by the encounter model
- Variable size encounter cylinder based on the intruder velocity and RDR
• Simulation time: 60s / sample rate: 1Hz

• With sensor performance characteristics as in Table 5.2 and zero sensor noise

• With the field of regard requirements from Section 5.2

6.3 Fixed Coordinated Turn vs Kinematically Constrained

As previously noted in the Section 3.3, some of the initial modes were dropped in favor of the coordinated turn (CT) filters that had better performance. This section presents some of the results of that analysis.

A trajectory throughout the encounter cylinder is shown in Figure 6.2 with both cylinders shown; the encounter cylinder (blue) and the WCT (red). Figure 6.3 shows the control variables for this same trajectory.

![Figure 6.2. Encounter Overview](image)
In Figure 6.4, the CPA is used as a benchmark for the estimation performance of the 10 different filters initially considered in the IMM, meaning that each one is shown separately but is fed with the proper mixed initial conditions. It is important to clarify this point since next section will be using the individual Kalman filters outside the IMM context for analyzing the IMM performance itself. The difference between the poor performance of the fixed coordinated turn in comparison with other filters with variable and estimated variables is significant. Not surprisingly, having a fixed turn rate will not give a good performance in the cases where the actual turn rate is different from the fixed one. The original hypothesis to use the fixed CT filter was that it could present a good performance and be used for the IMM in the cases that it gets close to the selected average turn rate.

Figure 6.5 shows the mode probability change along the trajectory. Following the previous CPA error shown, it is evident that the IMM gives priority to the best performing filter that uses a kinematic constraint (KC) to estimate the turn rate.
In conclusion, the fixed turn rate filters were dropped in favor of the kinematically constrained ones. The latter will be used together with the remaining modes.
for all the following analysis in this thesis. This study reduced the number of IMM filters from 10 to the 6 listed in Section 3.3.

6.4 Individual Kalman Filter Performance vs IMM

This section explores the performance improvement of the IMM over the isolated Kalman filters. The isolated Kalman filters are just regular Kalman filters without using the mixed initial conditions, corrected by the IMM prediction of mode likelihood.

In order to test the adaptation to mode changes of the IMM against the isolated Kalman filters and see the performance of the algorithm in more extreme cases, the chosen trajectory depicted in Figure 6.6 has non-zero values for the three control variables generated from the encounter model, as shown in Figure 6.7.

![Figure 6.6. Encounter Overview](image-url)
The trajectory shows the intruder aircraft coming from the front of the ownship, accelerating, performing a turn and descending during different parts of the trajectory.

It is possible to observe a high CPA error for the individual Kalman filters under maneuver changes including turns, as seen in Figure 6.8. This is due to the sensitivity of the CPA prediction to the direction of the velocity vector and the poor estimation of it on the isolated KF.
In Figure 6.9, the IMM follows the CPA estimation (using the MOPS formula) change along the trajectory much closer than the individual Kalman filters. This particular trajectory presents the usual behavior of the estimation differences, but a more comprehensive conclusion can be done by performing a Monte Carlo simulation in the next section.
6.4.1 Monte Carlo Analysis. Even with the trajectory rejection sampling as detailed in Figure 6.1, there are still numerous trajectories that are not very interesting, going out of the cylinder after a few timesteps or just straight level flights that impose no danger to the WCT. For this particular analysis, only in the trajectories for which the WCT is violated are computed.

To analyze the general performance of the IMM compared with individual Kalman filters, $10^5$ simulations were run and the RMS of CPA error for each one of the filters was calculated. Since each of the KF or IMM modes takes some time to stabilize, the RMS values of the CPA estimation error starting at different timesteps $T_s$ were calculated to see how the values change along the simulation as well. Calculating the mean value of the RMS of the CPA estimation error gives us the following Table 6.1:
Table 6.1. Mean RMS of CPA Error [ft] Estimation of $10^5$ Trajectories

<table>
<thead>
<tr>
<th>T_s = 10s</th>
<th>T_s = 15s</th>
<th>T_s = 20s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>1318.9</td>
<td>1294.9</td>
</tr>
<tr>
<td>Mode 2</td>
<td>1318.9</td>
<td>1294.9</td>
</tr>
<tr>
<td>Mode 3</td>
<td>2411.7</td>
<td>2439.6</td>
</tr>
<tr>
<td>Mode 4</td>
<td>2409.1</td>
<td>2438.3</td>
</tr>
<tr>
<td>Mode 5</td>
<td>4046.3</td>
<td>3765.1</td>
</tr>
<tr>
<td>Mode 6</td>
<td>4046.3</td>
<td>3765.1</td>
</tr>
<tr>
<td>IMM</td>
<td>112.5</td>
<td>107.6</td>
</tr>
</tbody>
</table>

The difference between the performance of the IMM and the individual Kalman Filters is clear when looking at the Monte Carlo results, presenting a much smaller CPA estimation error for the IMM. In the following sections, the IMM performance is further explored against the Perfect IMM.

6.5 Perfect IMM Performance vs Regular IMM

The idea of the comparison between the IMM and the Perfect IMM is to see how much the mode transition error affects the performance of the IMM algorithm. Since the PIMM always selects the correct motion model as soon as the mode change occurs, it defines the lower bound on transient mode transition error. This analysis starts with a single trajectory example in order to see how the IMM filter compares to the performance of the PIMM. In Figure 6.10 the example trajectory is shown, and the control variables that define it are in Figure 6.11.
To represent one of the worst case scenarios, the trajectory chosen has the intruder aircraft maneuvering with all three control variables being non-zero and changing along the trajectory. The CPA estimation error is shown in Figure 6.12.
For most of the trajectory in Figure 6.12, the effect of the adaptation of the IMM following the ideal PIMM is fast enough to present a small difference. It can be observed that the estimation of the IMM follows the PIMM once it identifies the correct mode, as expected.

6.5.1 Monte Carlo Analysis. In order to analyze the general performance of the IMM versus the PIMM, a Monte Carlo run was done for $10^5$ simulations. The run is the same as presented in previous section 6.4.1, but now showing the PIMM results in Table 6.2.

Table 6.2. Mean RMS of CPA Error [ft] Estimation of $10^5$ trajectories

<table>
<thead>
<tr>
<th></th>
<th>$T_s = 10s$</th>
<th>$T_s = 15s$</th>
<th>$T_s = 20s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIMM</td>
<td>111.5</td>
<td>107.0</td>
<td>102.7</td>
</tr>
<tr>
<td>IMM</td>
<td>112.5</td>
<td>107.6</td>
<td>103.3</td>
</tr>
</tbody>
</table>
The difference between the performance of the IMM and the Perfect IMM is negligible when considering the RMS error. The main difference is caused by the adaptation lag between a mode change and the algorithm response, which can only be observed in individual trajectories, since in the table the numerical difference in these values are small. This behavior can be seen in individual trajectories, as shown in Figure 6.12.

### 6.6 Comparison of the IMM with Sensor Noise vs Zero Sensor Noise

In order to estimate the effect of the *sensor noise error* in the IMM, the next analysis is done adding the standard sensor noise as shown in Table 5.2. To represent a complex scenario, a trajectory in which all control variables are non-zero is chosen for this simulation. The control variables are shown in Figure 6.13.

![Control Variables](image)

*Figure 6.13. Control Variables for the Simulation*

The CPA estimation errors for this trajectory are shown in Figure 6.14 and Figure 6.15. In Figure 6.14 the results are for a simulation with zero sensor noise. In
Figure 6.15 the sensor based on MOPS performance limits is added, increasing the difference between the IMM-MOPS and the True-MOPS (with *true trajectory data*).

It is important to make clear the difference between the states estimates and the prediction method, for example in the IMM-MOPS, we are using the IMM for estimation and the MOPS equations for prediction. As expected, with an additional error source (as discussed in Section 3.4) the difference of the calculated hazard states will be greater due to sensor noise.

Figure 6.14. CPA Estimation - Zero Sensor Noise
The new IMM Prediction method, provides a better prediction than the MOPS CPA formula, even using True data, since the latter formula extrapolates the intruder trajectory linearly. The figures clearly show that when the new IMM-IMM method is used nominal sensor noise error does not affect substantially the final CPA prediction error.

6.7 Vertical Separation

The actual vertical separation is estimated using the IMM algorithm for the state estimation and compared against the true trajectory states. An example trajectory is simulated with the control variables as shown in Figure 6.16. This particular trajectory has the vertical velocity changing along the simulation in order to observe how the algorithm adapts to the new modes.
The vertical separation estimate error is shown in Figure 6.17. The results show a small mean error in this hazard state, which is not unexpected since the maneuvers along the vertical axis only allow for constant velocity modes, as well as being independent from the other modes on the x-y plane (constant acceleration and coordinated turn).
An additional analysis is done considering multiple trajectories and also prediction for the vertical separation in section 6.9.3.

6.8 Time to CPA

The simulation for the time to CPA follows the previous ones done in this chapter. A single maneuvering trajectory is selected, as shown in Figure 6.18 with its control variables.
The closest point of approach happens at the end of the simulation, at 60s. It can be observed that the current MOPS method for calculation of $\tau$ has poor performance under maneuvers, as shown in Figure 6.19 as IMM-MOPS. This calculation is done using the IMM to estimate the intruder states and the MOPS formulas for the time to CPA. In the same figure, the new prediction method is shown as IMM-IMM and will be detailed in the next section 6.9.
6.9 New IMM Prediction Method Analysis

The new method utilizes the IMM prediction phase to estimate the trajectory of the intruder in the future. The algorithm updates a new *estimation* for the current timestep and then projects to all future timesteps without the sensor measurement for the *prediction* phase. There is a fundamental difference between the nature of the calculations of the hazard states using the MOPS equations and the new method developed in this research. While the MOPS uses a continuous function to linearly extrapolate the current states into the future, the new method advances in discrete future timesteps to calculate the same hazard states.

6.9.1 IMM Prediction Mode Probability Change Behavior. An interesting aspect of the mode likelihood change in the IMM prediction is that in absence of new information of the intruder states - i.e., the absence of new sensor measurements - the mode probabilities over time regress to the initial fixed mode probabilities. This
is expected since the the normalizing factors multiply the Markov chain matrix at every timestep as previously explained in Chapter 4.5. If enough time has passed in the simulation, the values will tend to approach the steady state values presented in Equation 6.1.

\[
\mu_{ip} = \begin{bmatrix}
0.551 & 0.153 & 0.107 & 0.047 & 0.092 & 0.047 \\
\end{bmatrix} \quad (6.1)
\]

For the same trajectory, the mode probability behavior is analyzed over time. In Figures 6.20 and 6.21 it is shown the mode probabilities starting at five and 10 seconds respectively. A maneuver occurs between these timesteps resulting in different initial mode probabilities and its subsequent changes.

Figure 6.20. Mode Probabilities Starting at t=5s
This shows the regression to the initial mode probability values over time regardless of what current actual mode is.

6.9.2 Time to CPA. The time to the closest point of approach is not as sensitive as CPA. Using the new method, the time to CPA is a discrete calculation based on which of the timesteps in the future correspond to the closest point of approach. Since the CPA itself can change substantially after a maneuver while maintaining a similar $\tau$, the latter is not as sensitive to the specific prediction method used, but using the new method versus the MOPS formulas still produce does some difference.

Figure 6.22 shows the results for the average of the estimated vs true Tau for $10^5$ trajectories with the standard sensor noise applied.
The dashed line shown at $\tau = 35s$ represent the threshold for this hazard state in the WCT. The average error does not seem to be much different, even though it is still reduced by using the new prediction method (IMM-IMM). But for larger $\tau$, the likelihood of any maneuver happening at some point before CPA increases, and there will be a larger spread than using the IMM prediction method. Just looking at the average values can hide some of the largest variations that occur in the Tau estimation when in maneuvering trajectories. Figure 6.23 shows the standard deviation of the errors in Tau for this simulation.
6.9.3 Vertical Separation. Since the vertical axis only has linear constant velocity modes, and these are decoupled from the other possible maneuvers, vertical state estimation presents the smallest errors among the hazard states. Figure 6.24 presents the average of current estimation error on vertical separation, for $10^4$ trajectories.

Figure 6.23. Standard Deviation Error of Time to CPA Estimation - $10^5$ trajectories
At each timestep, the algorithm also predicts all future timesteps. For a comparison with the MOPS, a lookahead time of 25s was used in Jamoom’s work [31,58]. The MOPS estimation for the calculation then uses the assumption of constant linear velocity, while the IMM estimates the states in its prediction phase.
Using the **IMM estimation** provides good state estimation, and since the MIT LL encounter model outputs only constant velocity trajectories in the vertical axis, the MOPS formula provides decent vertical separation predictions. The average errors along all timesteps considered and through all trajectories are small, about 2.1ft for the IMM-MOPS and 0.6ft for the IMM-IMM. The slight improvement can be attributed to the mode probability regression to the initial mode probabilities, as shown previously at section 6.9.1. For the eventual case of an aircraft climbing or descending initially but changing to steady level flight later, the IMM will provide a better estimate.

### 6.9.4 Closest Point of Approach.

In order to compare the general performance of the IMM prediction estima-
tion method, $10^4$ different encounters were run comparing it and the DAA MOPS calculation against the True CPA value.

First, a histogram of all the position error values in the CPA estimation (concatenated for x and y positions) is presented in Figure 6.26. It can be observed here that the results from the IMM have a smaller deviation than those using the DAA MOPS calculation.

Next, a scatterplot of the same values is shown in Figure 6.27. It is evident from this latest figure that the results from the IMM Prediction are closer to the origin, around the region of the True CPA. It is interesting to note some of the dots aligned along nearly straight lines, which are all different timesteps in the same trajectory simulation.

As the estimators get more measurements, the CPA error estimation decreases as expected, moving closer to the center of the scatterplot. The prediction errors are
considerably reduced when using the IMM Prediction method when compared to the MOPS calculation.

![Figure 6.27. CPA position error scatterplot](image)

With these results, it is shown that the *mode transition error* can be significantly reduced by applying the new IMM prediction method.

### 6.10 IMM Analysis Summary

In this chapter, simulation results were presented in order to evaluate the IMM algorithm’s effectiveness in the DAA problem. Among the different approaches to these analyses, it was shown at first that the fixed coordinated turn (CT) modes provided poor estimation performance and were dropped in favor of kinematically constrained filters. The individual Kalman filters are shown to have a poor performance in the maneuvering target problem such as the one presented. The IMM performed much better than each individual Kalman filter.

The different error sources identified in Chapter 3.4 were analyzed in order
to estimate how they affect the IMM performance. The IMM was compared to the Perfect IMM, standard sensor noise was added and finally the new prediction method was evaluated. Three separate conclusions can be drawn from the hazard state estimation performance. The least affected is the vertical separation, which already exhibits good performance using the IMM estimation with either prediction method, but the error is still reduced slightly using IMM prediction. For the time to CPA, the IMM prediction results are better than the MOPS projections, the main difference being the reduction of the standard deviation of the prediction errors. The IMM prediction shows more consistent results while the MOPS prediction underperforms in the maneuver cases. Finally, for the CPA large differences in prediction performance were observed, with the new IMM-prediction achieving far better performance than the MOPS prediction formula.
CHAPTER 7
SAFETY EVALUATION

It is important to evaluate safety based on the likelihood of a LoWC, given
the MOPS DAA sensor performance characteristics outlined in Chapter 5 [58,59]. A
statistical analysis is possible to achieve two goals. The first goal is to determine if the
MOPS DAA sensor requirements are sufficient to achieve a target level of safety. The
second goal is to determine the level of safety for a given set of sensor characteristics.

7.1 Risk evaluation methodology

To estimate the LoWC rate, a large collection of encounters is generated and
then it is determined which encounters lead to LoWC. This is an encounter where the
intruder aircraft penetrates the LoWC cylinder before exiting the encounter cylinder.
The density of air traffic in the region being simulated will influence the results. Most
of the encounters simulated do not enter the LoWC volume, but it is important to
simulate them to ensure the DAA system identifies the potential threat within the
required time for alerting purposes (and potential collision avoidance maneuvers).

By generating a large collection of encounters and determining which encoun-
ters lead to LoWC, the probability of LoWC, \( P(\text{LoWC}|E_i) \), can be estimated from
the simulation results:

\[
P(\text{LoWC}) = \sum_{i=1}^{N_e} P(\text{LoWC}|E_i)P(E_i)
\]  

where \( N_e \) is the total number of encounter types and \( P(E_i) \) is the probability mass
function (PMF) for encounter \( E_i \), \( i = 1,\ldots,N_e \). Because the PMF is unknown a
priori, the encounter model, which outputs radar samples of encounters in proportion
to \( P(E_i) \) is used instead. Therefore Equation (7.1) is now:
\[
\hat{P}(\text{LoWC}) = \lim_{N_s \to \infty} \frac{1}{N_s} \sum_{j=1}^{N_s} \hat{P}(\text{LoWC}|E_j)
\]  
(7.2)

where \(E_j\) are the encounters generated by the encounter model.

Finally the LoWC rate, denoted as \(\lambda_{\text{LoWC}}\), is dependent on the encounter rate, \(\lambda_{\text{enc}}\) [39]:

\[
\hat{\lambda}_{\text{LoWC}} = P(\text{LoWC}) \lambda_{\text{enc}}
\]  
(7.3)

\[
\lambda_{\text{enc}} = \rho_{tr} \bar{V}
\]  
(7.4)

The volume sweep rate \(\bar{V}\) depends on the size of the encounter cylinder and the average velocities of aircraft involved in encounters (to account for the volume of airspace travelled). The traffic density, \(\rho_{tr}\), can be expressed in aircraft per volume. The rate at which the intruder aircraft enters the encounter cylinder is the product of traffic density and the average volume of airspace the encounter cylinder sweeps out per unit time.

7.2 Encounter Rate

In order to estimate the traffic density, \(\rho_{tr}\), it is possible to use the values generated in the MIT LL encounter model research [39]. The highest density area is around Miami-Dade County with a density of approximately 0.003 aircraft per NM³, and is represented in the southeastern corner of Figure 7.1.
The next term to calculate in the equation (7.4) is the average volume of new airspace the encounter cylinder sweeps through per unit time, $\bar{V}$. This depends on the size of the encounter cylinder and on the average velocities of aircraft involved in encounters. Since the encounter cylinder has a variable size due to different RDR requirements, as shown in Section 2.2.1, one approach is to run a Monte Carlo simulation ($10^5$ runs) and estimating the likelihood of each type of encounter, as shown in Table 7.1.

Table 7.1. Encounters type occurrence

<table>
<thead>
<tr>
<th>Encounter Type</th>
<th>Occurrence</th>
<th>RDR [NM]</th>
<th>Max speed [kt]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small encounters</td>
<td>41%</td>
<td>5.4</td>
<td>100</td>
</tr>
<tr>
<td>Medium encounters</td>
<td>35%</td>
<td>6.0</td>
<td>130</td>
</tr>
<tr>
<td>Large encounters</td>
<td>24%</td>
<td>6.7</td>
<td>200</td>
</tr>
</tbody>
</table>

Weighting the cylinder size average based on the likelihood of each encounter
type, the average cylinder front section area has 11.7 NM$^2$. At 2339 NM$^3$/h (speed of 200kt, ±3000ft cylinder height and the weighted average cylinder range of 5.92 NM) and 0.003 aircraft per NM$^3$ (highest density in Miami), the result is 7 encounters per hour.

Note that this does not mean that collision avoidance maneuvering would be required 7 times per hour because not all encounters would require maneuvering or represent a LoWC. Most of these encounters are benign, but it is necessary to simulate the performance of the DAA system to ensure its safety. Using the same Monte Carlo run for the table above, it was recorded which encounters resulted in a Loss of Well Clear, by violation of all three hazard states simultaneously, getting a result of 0.6% of these encounters. In the absence of any DAA system, the expected rate of LoWC estimated is then approximately 0.07 LoWC/h, or 1 for every 23.4 hours. For estimating the performance of the IMM in predicting the LoWC condition, it is necessary to perform a covariance analysis which is done in the next sections.

### 7.3 Calculation of the Hazard State Covariance

The first step in order to find the probability of a given encounter resulting in a LoWC is to find the standard deviations ($\sigma_{CPA}, \sigma_\tau, \sigma_z$) for the hazard states estimation from the IMM.

The IMM algorithm outputs at each timestep are the best state estimates (and the IMM prediction for the future timestep), $x$, and its associated covariance matrix, $P$, in relation to the trajectory states, $[x, \dot{x}, \ddot{x}, ..., \dddot{z}]$. It is necessary to change $x$ and $P$ to hazard states.

A first order Taylor series expansion is used to obtain the approximated linearized hazard state estimates in relation to the trajectory state estimates, as shown by Jamoom [31]. The Taylor series partial derivative vectors are given by:
As an example, $a_{CPA}^T$ is the vector of partial derivatives of CPA (or HMD) with respect to the trajectory states:

$$a_{CPA}^T = \begin{bmatrix} \frac{\partial CPA}{\partial x} & \frac{\partial CPA}{\partial y} & \frac{\partial CPA}{\partial z} & \frac{\partial CPA}{\partial \dot{x}} & \frac{\partial CPA}{\partial \dot{y}} & \frac{\partial CPA}{\partial \dot{z}} & \frac{\partial CPA}{\partial \ddot{x}} & \frac{\partial CPA}{\partial \ddot{y}} & \frac{\partial CPA}{\partial \ddot{z}} \end{bmatrix} \tag{7.6}$$

The partial derivative vectors $a_{\tau}^T$ and $a_{z}^T$ are calculated in the same way. The full covariance matrix, $P_{CPA\tau z}$, of the hazard state is given by the following equation:

$$P_{CPA\tau z} = A_n P A_n^T \tag{7.7}$$

where $P_{CPA\tau z}$ is the covariance matrix of the hazard states. Then the hazard state estimate variances $\sigma_{CPA}^2, \sigma_{\tau}^2, \sigma_{z}^2$ are the diagonal elements of the matrix $P_{CPA\tau z}$.

Jamoom shows that these derivatives can be calculated for the constant velocity and for the constant linear acceleration cases [31]. The complexity of the motion model in this thesis, allowing for maneuvers and for accelerations not constrained to the velocity vector direction, makes an analytic derivation impossible. The numerical estimation of these derivatives is explored in the next section.

7.4 Numerical Derivation of the Partial Derivatives
The numerical derivation method is to use finite difference approximations in the partial derivatives for the $A_n$ matrix. In this method, a simple two-point estimation is to compute the slope of a tangent line through the points $(x, f(x))$ and $(x + \Delta, f(x + \Delta))$. Since the goal is to find the partial derivatives with respect to the original states $[x, \dot{x}, \ddot{x}, ..., \dddot{z}]$, the $\Delta$ is applied to these individual values and then after running the IMM finding the partial numerical derivatives, as in Figure 7.2 for an example for the calculation of $\frac{\partial CPA}{\partial x}$:

![Figure 7.2. Delta Hazard States](image)

Then calculating its numerical derivative:

$$\frac{\partial CPA}{\partial x} = \frac{CPA_\Delta - CPA}{\Delta} \tag{7.8}$$

The 3x9 matrix shown in equation 7.5 is then populated for all the derivatives similarly.

### 7.5 Hazard State Standard Deviations

Starting from a typical single encounter analysis, the following values for $\sigma_{CPA}$ were obtained. In Table 7.2 the lines represent the actual timestep while the columns represent the calculated timestep of the simulation. The values on the main diagonal represent the IMM estimation (current timestep) while all the other values represent the IMM prediction (for future timesteps). The table is truncated due to...
space constraints, but the idea from this result is showing that even with 10 seconds of prediction the IMM algorithm will produce high standard deviations.

Table 7.2. $\sigma_{CPA}$ values for a single trajectory [ft]

<table>
<thead>
<tr>
<th>Est. timestep</th>
<th>11.2</th>
<th>18.4</th>
<th>28.1</th>
<th>39.9</th>
<th>53.7</th>
<th>69.6</th>
<th>87.6</th>
<th>108.1</th>
<th>131.4</th>
<th>157.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.6</td>
<td>19.6</td>
<td>30.2</td>
<td>43.1</td>
<td>58.4</td>
<td>76.1</td>
<td>96.5</td>
<td>119.8</td>
<td>146.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.9</td>
<td>20.6</td>
<td>32.6</td>
<td>47.5</td>
<td>65.5</td>
<td>86.6</td>
<td>111.3</td>
<td>139.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.1</td>
<td>22.9</td>
<td>38.5</td>
<td>58.7</td>
<td>83.6</td>
<td>113.3</td>
<td>148.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current timestep</td>
<td>12.4</td>
<td>27.0</td>
<td>49.7</td>
<td>80.2</td>
<td>118.4</td>
<td>164.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.7</td>
<td>29.4</td>
<td>58.3</td>
<td>98.5</td>
<td>150.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.1</td>
<td>28.7</td>
<td>58.4</td>
<td>100.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.6</td>
<td>26.9</td>
<td>55.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.1</td>
<td>25.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As a comparison between different possible trajectories and its effect on these values, another result is shown in Table 7.3. Starting three different maneuvering trajectories at timestep 10s and running up to 20 seconds it is possible to take a look how the values evolve. Compared to Table 7.2, each column in this table shows only the prediction values starting with a single starting timestep, which would correspond to a row of the Table 7.2.
### Table 7.3. $\sigma_{CPA}$ Values for Three Trajectories [ft]

<table>
<thead>
<tr>
<th></th>
<th>Trajectory 1</th>
<th>Trajectory 2</th>
<th>Trajectory 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{CPA,10,10}$</td>
<td>12.9</td>
<td>13.0</td>
<td>11.7</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,11}$</td>
<td>24.1</td>
<td>19.9</td>
<td>29.4</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,12}$</td>
<td>42.4</td>
<td>30.6</td>
<td>58.3</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,13}$</td>
<td>67.7</td>
<td>44.9</td>
<td>98.5</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,14}$</td>
<td>100.1</td>
<td>63.1</td>
<td>150.2</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,15}$</td>
<td>139.9</td>
<td>85.9</td>
<td>213.6</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,16}$</td>
<td>187.6</td>
<td>113.8</td>
<td>289.0</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,17}$</td>
<td>243.5</td>
<td>147.5</td>
<td>376.5</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,18}$</td>
<td>307.8</td>
<td>187.7</td>
<td>476.0</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,19}$</td>
<td>381.0</td>
<td>235.1</td>
<td>587.6</td>
</tr>
<tr>
<td>$\sigma_{CPA,10,20}$</td>
<td>463.4</td>
<td>290.3</td>
<td>711.3</td>
</tr>
</tbody>
</table>

With the current results being considerably higher than what was expected from previous work [31], a further investigation on the source of the large standard deviation errors is necessary. The next section will explore the effect of the process noise on these sigma values.

### 7.6 Process Noise Analysis

In the IMM algorithm, if the process noise is set too high on the component filters the IMM disregards most of the mode’s dynamics since more trust is given to the sensor measurements. The mode probabilities do not change as the modes all perform equally poor. If the process noise is set to be too low, the IMM trusts each mode’s dynamics too much, changing the mode probability very frequently since each one can perform better for a single measurement. This mode probability instability is not desired since it is not the observed behavior of the aircraft trajectories. For all the algorithm runs in the previous chapters and this on, an intermediate value of the process noise was chosen after some Monte Carlo runs in order to settle on a value
for $Q$, as in equation 2.4, that resulted in the best estimations and stability for the model.

### 7.6.1 Single trajectory example

For Figures 7.3 through 7.6, the mode change behaviors are shown for different levels of process noise for the same trajectory. The Perfect IMM is depicted in Figure 7.3 as the correct reference for mode switching. Figure 7.4 depicts the change in mode probabilities for a standard level of process noise. All process noise levels of the different modes are presented in Appendix C.

![Figure 7.3. Mode Probability - Perfect IMM](Image)
In Figure 7.5, the process noise levels are 10x smaller. The mode probability instability starts to appear and even undesired modes take on higher probabilities for some timesteps.
In Figure 7.6, the process noise levels are 10x bigger than the standard. In this case, the dynamics do not matter much since the process noise levels are too high compared to the sensor noise levels. The modes with a natural higher probability (constant velocity) in the Markov chain matrix take priority even when they are not the correct mode.

Figure 7.6. Mode Probability - Large Process Noise Levels

In order to explore how the chosen process noise values affect the estimate error covariances, a comparison between the intermediate process noise and minimal process noise (but large enough to avoid mode instability issues) is also done in this section. Due to the large amount of data generated on each one of the runs, one of the challenges is choosing how to best represent the results obtained. For each run, there is an individual 9x9 covariance matrix for every current timestep and every future timestep. In order to synthesize the results, the presented values on the following tables are the square root of the diagonal values of the covariance matrix (the standard deviation of the estimated trajectory states), starting at timestep 10.
(\sigma_{10,10}, \text{ moving a couple timesteps forward } (\sigma_{10,11} \text{ and } \sigma_{10,12}) \text{ and finally showing what the values are at timestep } 20s \ (\sigma_{10,20}). \ \sigma^{(1)} \text{ at Table 7.4 shows the values for the regular process noise and } \sigma^{(2)}, \text{ in Table 7.5, for the reduced process noise.}

Table 7.4. Standard Deviation of Trajectory States - Regular Process Noise

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_{10,10}^{(1)})</th>
<th>(\sigma_{10,11}^{(1)})</th>
<th>(\sigma_{10,12}^{(1)})</th>
<th>(\sigma_{10,20}^{(1)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sigma_x)</td>
<td>2.3</td>
<td>5.7</td>
<td>10.8</td>
<td>131.6</td>
</tr>
<tr>
<td>(\sigma_y)</td>
<td>2.9</td>
<td>4.5</td>
<td>6.3</td>
<td>26.9</td>
</tr>
<tr>
<td>(\sigma_z)</td>
<td>1.8</td>
<td>2.1</td>
<td>2.3</td>
<td>3.9</td>
</tr>
<tr>
<td>(\sigma_{\dot{x}})</td>
<td>2.3</td>
<td>5.8</td>
<td>10.9</td>
<td>134.4</td>
</tr>
<tr>
<td>(\sigma_{\dot{y}})</td>
<td>2.9</td>
<td>4.5</td>
<td>6.4</td>
<td>27.5</td>
</tr>
<tr>
<td>(\sigma_{\dot{z}})</td>
<td>1.8</td>
<td>2.1</td>
<td>2.4</td>
<td>3.9</td>
</tr>
<tr>
<td>(\sigma_{\ddot{x}})</td>
<td>1.0</td>
<td>1.6</td>
<td>2.1</td>
<td>8.0</td>
</tr>
<tr>
<td>(\sigma_{\ddot{y}})</td>
<td>1.0</td>
<td>1.1</td>
<td>1.3</td>
<td>2.1</td>
</tr>
<tr>
<td>(\sigma_{\ddot{z}})</td>
<td>1.0</td>
<td>1.0</td>
<td>1.1</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 7.5. Standard Deviation of Trajectory States - Reduced Process Noise

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_{10,10}^{(2)})</th>
<th>(\sigma_{10,11}^{(2)})</th>
<th>(\sigma_{10,12}^{(2)})</th>
<th>(\sigma_{10,20}^{(2)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sigma_x)</td>
<td>0.70</td>
<td>1.82</td>
<td>3.61</td>
<td>50.23</td>
</tr>
<tr>
<td>(\sigma_y)</td>
<td>0.89</td>
<td>1.49</td>
<td>2.21</td>
<td>10.24</td>
</tr>
<tr>
<td>(\sigma_z)</td>
<td>0.56</td>
<td>0.63</td>
<td>0.70</td>
<td>1.13</td>
</tr>
<tr>
<td>(\sigma_{\dot{x}})</td>
<td>0.71</td>
<td>1.90</td>
<td>3.86</td>
<td>58.00</td>
</tr>
<tr>
<td>(\sigma_{\dot{y}})</td>
<td>0.91</td>
<td>1.59</td>
<td>2.43</td>
<td>11.95</td>
</tr>
<tr>
<td>(\sigma_{\dot{z}})</td>
<td>0.56</td>
<td>0.65</td>
<td>0.72</td>
<td>1.23</td>
</tr>
<tr>
<td>(\sigma_{\ddot{x}})</td>
<td>0.31</td>
<td>0.47</td>
<td>0.61</td>
<td>2.20</td>
</tr>
<tr>
<td>(\sigma_{\ddot{y}})</td>
<td>0.31</td>
<td>0.33</td>
<td>0.36</td>
<td>0.55</td>
</tr>
<tr>
<td>(\sigma_{\ddot{z}})</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
<td>0.30</td>
</tr>
</tbody>
</table>

The smaller process noise obviously reduces the standard deviations, but process noise needs to be limited to avoid instability. Regardless, the last column in
Table 7.5 suggests that using the **IMM prediction**, even by looking at only 10 seconds in the future will still lead to large predicted standard deviations.

### 7.6.2 Monte Carlo Analysis.

As done in the previous chapter, a Monte Carlo analysis \( (10^4 \text{ simulations}) \) was performed with the nominal sensor noise and comparing the IMM and MOPS methods for estimating the hazard states. The change here is on the process noise levels in order to see its effect on the CPA prediction error. Three different levels are compared, the optimized level of process noise \( (Q) \) to reduce the hazard states and also avoid stability issues, a value 10 times bigger and also 10 times smaller. The scatterplot of the CPA position estimation of these simulations are shown in figures 7.7, 7.8 and 7.9.

Figure 7.7. Scatterplot of CPA Position Error - Standard Noise \( Q \)
Since differences in the values are not easily identifiable from the scatterplots, the results are presented as standard deviations of the CPA position error for these Monte Carlo runs in Table C.1.
Table 7.6. Standard deviation of the CPA position error [ft]

<table>
<thead>
<tr>
<th></th>
<th>10Q</th>
<th>Q</th>
<th>0.1Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMM</td>
<td>1108.2</td>
<td>873.5</td>
<td>981.0</td>
</tr>
<tr>
<td>MOPS</td>
<td>5789.6</td>
<td>4973.0</td>
<td>6117.0</td>
</tr>
</tbody>
</table>

Beyond the difference in the estimation error values, there are the instability problems causing some extreme error values. In conclusion, the standard process noise levels seem appropriate given all the results shown.

### 7.7 Source for the Large Covariances on the IMM Prediction

The simulations for different process noise levels showed that even though it is possible to reduce the CPA standard deviation increase over time for the IMM prediction, it was still larger than expected when compared with the prior results presented by Jamoom for linear intruder trajectories [31]. However, the new IMM prediction method introduced in this work does show performance than the MOPS-prescribed algorithms in the previous chapters.

By analyzing each individual component of the final calculation of the deviations, it can be attributed to the IMM covariance only, since the numerical derivatives introduced earlier growth is reasonable or even absent. The covariance matrix in the IMM prediction grows fast even with minimal process noise levels.

Recalling Equation (3.10) for the covariance calculation in the IMM, we have:

\[
P(k|k) = \sum_{j=1}^{r} \mu_j(k) \{ P_j(k|k) + [\hat{x}_j(k|k) - \hat{x}(k|k)] [\hat{x}_j(k|k) - \hat{x}(k|k)]^T \} \quad j = 1, ..., r
\]

(7.9)

If the term multiplying the mode probability \( \mu_j(k) \) only contained the mode
covariances, $P_j(k|k)$, that equation would be similar to equation 3.9, in the sense that
the IMM covariance would be just the weighted average of each mode covariance,
as it is the case for the IMM state estimation. But the additional multiplicative
terms $((\hat{x}_j(k|k) - \hat{x}(k|k))[(\hat{x}_j(k|k) - \hat{x}(k|k)]^T)$ add to the individual mode covariance.
These terms account for the dispersion of the means of the component modes of the
IMM. In the IMM estimation phase, when measurements are still available, these
additional terms are typically not very large. But in the IMM prediction phase
this term makes the covariance of the IMM prediction increasingly large. That is the
main source of the large estimated standard deviations for the hazard states.

7.8 Safety Evaluation Summary and Discussion

In this chapter, the safety analysis methodology was introduced. Using the
encounter rate and the likelihood of a given encounter resulting in a LoWC, it is
possible to estimate the overall rate of LoWC per hour and compare it to safety
standards. The calculation of the hazard state covariances and standard deviation
advanced previous work, using numerical derivatives inside the IMM algorithm, but
the results for these were higher due to the uncertainty of future mode switches. An
additional analysis for the process noise was done to ensure that the process noise
levels used were adequate, and even though reducing its levels help decrease the
standard deviations, there is a stability lower limit for the process noise in the IMM.

A possible alternative to be explored in future work would be a generation of
LoWC trajectories (by using encounter rejection), looking at a specific moment along
the trajectory ($\tau = 35s$), and then computing the probabilities of correct detection of
the LoWC by the predicted hazard states. More suggestions for further work in this
area are presented in the final chapter.
CHAPTER 8
CONCLUSION

The focus of this dissertation has been to investigate the UAS DAA problem with maneuvering intruders. This dissertation has directly addressed the use of the Interactive Multiple Model algorithm on the tracking of maneuvering targets for the DAA use in UAS systems. These methods can provide better estimation tools than the current standardized requirements, and these new tools can be applied to current sensors in development or even allow a fresh look at the current sensor requirements.

This dissertation has also explored a new prediction method for the hazard states which define the Loss of Well Clear condition. Areas of contributions and future work are discussed in the following subsections.

8.1 Summary of Accomplishments

8.1.1 Analysis of Maneuvering Encounter Trajectories in the DAA Problem. This work introduced the use of random trajectories using an established encounter model to account for realistic trajectories including maneuvers. Taking advantage of the encounter model generated trajectories, a comprehensive analysis of the DAA problem was done.

8.1.2 Introduced the Multiple Model Adaptive Estimation use for DAA. This work used a Multiple Model Adaptive Estimation (MMAE) algorithm to deal with the maneuvering target tracking problem in the DAA context. The Interactive Multiple Model (IMM) modes were modeled according to the applicable problem variables.

8.1.3 Identification and quantification for the Interactive Multiple Model error sources. These error sources were identified and then explored in the analysis chapter. The analysis isolated each one of the different error sources in order to
quantify how much each one influenced the performance of a DAA system and the final performance of the algorithm considering all in effect.

8.1.4 Developed a New Method for Hazard States Estimation. After using the hazard states estimation as a benchmark for the IMM performance, a new method for the hazard state estimation was developed. As opposed to the DAA MOPS hazard state formulas, which are based on linear and non-accelerated trajectories [58], the new prediction method uses the IMM prediction phase in the algorithm to predict the future intruder aircraft states based on the last sensor measurement. This new method was shown to reduce significantly the estimation errors in the CPA, while slightly reducing the time to CPA error and vertical separation.

8.1.5 Safety Evaluation Method. This work introduced a method for performing a safety evaluation of the IMM using a loss of separation probability associated with the encounter rate.

8.2 Recommended Topics for Future Research

Some of the recommendations for future work are given in the following subsections including the continuation of the development of the safety analysis using the IMM in the DAA problem, performing flight tests in order to acquire real data for the validation of the proposed method and also extending the research for small UAS.

8.2.1 Safety Analysis and Process Noise. Further investigation on the process noise levels and optimization according to the expected flight regimes can be researched to improve the estimation for specific cases. These were shown to be sensitive to the variables involved in the problem at question. For a different set of encounter parameters those optimal values would be different. It was not conclusive if the process noise levels can still be further reduced and also if there is any other
methodology that can be developed to estimate the integrity of the IMM methods that do not depend directly on the standard deviations calculated using numerical derivatives. A possible alternative to be explored is the generation of LoWC trajectories (by using encounter rejection), looking at a specific moment along the trajectory ($\tau = 35s$), and then computing the probabilities of correct detection of the LoWC by the predicted hazard states.

8.2.2 Flight Tests. The Illinois Institute of Technology’s Navigation Laboratory has an octocopter and a LIDAR sensor that can be use to test the methods presented in this thesis and in previous work.

8.2.3 Extension of the research to the small UAS DAA. There is significant interest in the next steps on the certification effort are to develop small UAS Detect and Avoid requirements for beyond visual line of sight (BVLOS) operations.

8.3 Closing

Detect and Avoid capabilities must be ensured for the safe integration of the UAS in the National Airspace System. There is still a lot of work to be done not only in the implementation os these systems for the large UAS, but also a need for the full development of new standards for the integration of DAA for small UAS.
APPENDIX A

INERTIAL TO RELATIVE FRAME TRANSFORMATION
The inertial to relative frame transformation is performed as follows:

\[ x_{rel}^{k+1} = x_{k+1}^I - x_{k+1}^O \] (A.1)

\[ x_{k+1}^I - x_{k+1}^O = F_k^I x_k^I - F_k^O x_k^O \] (A.2)

\[ x_{k+1}^I - x_{k+1}^O = F_k^I x_k^I - F_k^O x_k^O - F_k^I x_k^O + F_k^I x_k^O \] (A.3)

\[ x_{rel}^{k+1} = F_k^I x_k^{rel} + [F_k^I - F_k^O] x_k^O \] (A.4)

\[ x_{k+1}^{rel} = F_k^I x_k^{rel} + G_k u_k \] (A.5)

where:

- \( x^{rel} \) is the state vector of the intruder aircraft in the own aircraft-centered relative frame.

- \( x^I \) is the state vector of the intruder aircraft in the inertial frame.

- \( x^O \) is the state vector of the own aircraft in the inertial frame.

- \( F^I \) is the state transition matrix of the intruder aircraft.

- \( F^O \) is the state transition matrix of the own aircraft.

We define \( [F^I - F^O] = G_k \) and \( x_k^O = u_k \) as our “input” velocity of our own aircraft to transform inertially-defined intruder maneuvers to the relative frame. Our final Equation (A.5) is analogous to the Kalman filter Equation (2.3).
APPENDIX B
IMM DYNAMIC MODES
For Modes 1-2:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T$$  \hspace{1cm} (B.1)

$$\mathbf{F}^O = \begin{bmatrix} 0 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$ \hspace{1cm} (B.2)

$$\mathbf{G} = \mathbf{F}^I - \mathbf{F}^O$$ \hspace{1cm} (B.3)

$$\mathbf{u} = \begin{bmatrix} 0 & \dot{x}_{own} & 0 & \dot{y}_{own} & 0 & \dot{z}_{own} \end{bmatrix}^T$$ \hspace{1cm} (B.4)

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$ \hspace{1cm} (B.5)

For Modes 3-6:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & z & \dot{z} & \ddot{z} \end{bmatrix}^T$$ \hspace{1cm} (B.6)
\[
F^O = \begin{bmatrix}
0 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & T & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 
\end{bmatrix}
\] (B.7)

\[G = F^I - F^O\] (B.8)

\[u = \begin{bmatrix}
0 & \dot{x}_{own} & 0 & 0 & \dot{y}_{own} & 0 & 0 & \dot{z}_{own} & 0
\end{bmatrix}^T\] (B.9)

\[H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}\] (B.10)

- Mode 1 - Constant velocity
\[
F^I =\begin{bmatrix}
1 & T & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & T & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

(B.11)

- Mode 2 - Constant velocity 3D

\[
F^I =\begin{bmatrix}
1 & T & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & T & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(B.12)

- Mode 3 - Coordinated Turn Model (kinematics constraints)
Mode 4 - Coordinated Turn Model (kinematics constraints) - 3D
\[
F' = \begin{bmatrix}
1 & \frac{\sin(\omega T)}{\omega} & \frac{1-\cos(\omega T)}{\omega^2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \frac{\sin(\omega T)}{\omega} & \frac{1-\cos(\omega T)}{\omega^2} & 0 & 0 \\
0 & 0 & 0 & 0 & \cos(\omega T) & \frac{\sin(\omega T)}{\omega} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\omega \sin(\omega T) & \cos(\omega T) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

• Mode 5 - Constant Linear Acceleration
\[ \mathbf{F}^f = \begin{bmatrix}
1 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & T & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & T & T^2/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]  
(B.15)

- Mode 6 - Constant Linear Acceleration - 3D
\[ F^J = \begin{bmatrix}
1 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & T & T^2/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 
\end{bmatrix} \quad (B.16) \]
APPENDIX C
PROCESS NOISE
Table C.1. Process Noise Levels

<table>
<thead>
<tr>
<th>Mode</th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
<th>$q_4$</th>
<th>$q_5$</th>
<th>$q_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>0.5</td>
<td>0.05</td>
<td>5</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.5</td>
<td>0.05</td>
<td>5</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.5</td>
<td>0.05</td>
<td>5</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.5</td>
<td>0.05</td>
<td>5</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.1</td>
<td>0.01</td>
<td>5</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.1</td>
<td>0.01</td>
<td>5</td>
<td>1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

\begin{align*}
Q_1 &= q_1 \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
Q_2 &= q_2 \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}
\end{align*}
\[
Q_3 = q_3 \ast \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]  
\text{(C.3)}

\[
Q_4 = q_4 \ast \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]  
\text{(C.4)}
\[ Q_5 = q_5 \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]  

\[ Q_6 = q_6 \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]
BIBLIOGRAPHY


