6-6-2019

Tracking Highly-Maneuvering Targets and Data Fusion from IMM Estimator Tracks

Radu Visina

University of Connecticut - Storrs, radu.visina@uconn.edu

Follow this and additional works at: https://opencommons.uconn.edu/dissertations

Recommended Citation


https://opencommons.uconn.edu/dissertations/2224
Maneuvering targets are of special interest in remote tracking because of the difficulty in modeling the changes in their dynamic behavior. Sharp maneuvers present challenges. For some targets, such as those moving in a 2D plane, the proposed White Noise Turn Rate (WNTR) model should be included as one or more modes in an Interacting Multiple Model (IMM) estimator. This model is similar to the conventional coordinated-turn (CT) model, also known as the Wiener Process Turn Rate model, but with non-zero mean turn rate process noise, instead of zero-mean turn rate process noise. Such an IMM design, possibly coupled with a CT mode as well to accommodate longer turns with low turn rate, results in better estimation accuracy and improved covariance consistency. More sophisticated targets, such as those guided by feedback control systems in 3D space headed for a known or approximately-known destination, should be modeled using a guidance and destination model. In this dissertation, the model is based on the Proportional Navigation Guidance Control feedback control strategy implemented in real-world targets. It is shown that the structure of the model, especially when used in an IMM estimator, allows for an intuitive and robust description of target maneuvers in 3D space with significant improvement in tracking accuracy and with destination estimation capability. To strengthen the capability of tracking systems that operate using multiple independent trackers estimating the same target's state, a data fusion method dedicated to information coming from IMM tracks is developed. The Fusion Center receives local IMM “inside information”: mode-conditioned estimates and mode probabilities representing the Gaussian mixture density information at the local trackers. The method is capable of on-demand fusion, where the fusion is performed without memory of past fused estimates. Simulations show that this fusion estimator has fused covariance consistency matching the ideal consistency of the Centralized Measurement Fusion because of the computation and inclusion of crosscovariances in the fusion. It also has significantly lower peak error, and slightly less average error than fusion with moment-matched (outside) information.
APPROVAL PAGE

Doctor of Philosophy Dissertation

Tracking Highly-Maneuvering Targets and Data Fusion from IMM Estimator Tracks

Presented by

Radu Spiridon Visina, B.S.

Major Advisor ................................................................. Yaakov Bar-Shalom

Co-Major Advisor .......................................................... Peter Willett

Associate Advisor ......................................................... Krishna Pattipati

University of Connecticut

2019

iii
# Table of Contents

1 Introduction ............................................... 1
   1.1 Outline ............................................. 2
   1.2 Background on Multiple-Model Estimators for Tracking Sharply-Maneuvering Ground Targets ............................................. 2
   1.3 Background on Tracking Highly-Maneuvering Targets with Destination Information ............................................. 5
   1.4 Background on Data Fusion using IMM Inside Information ............................................. 8
   1.5 Summary of Contributions ............................................. 9
      1.5.1 Contributions Related to Tracking Sharply-Maneuvering Targets in 2D Space ............................................. 9
      1.5.2 Contributions Related to Tracking Maneuvering Targets in 3D Space ............................................. 10
      1.5.3 Contributions Related to Track-To-Track Fusion for Maneuvering Targets ............................................. 10
   1.6 Publications to Date ............................................. 12
      1.6.1 Journal Papers ............................................. 12
      1.6.2 Conference Proceedings Papers ............................................. 12

2 Multiple-Model Estimators for Tracking Sharply-Maneuvering Ground Targets ............................................. 13
   2.1 Problem Formulation ............................................. 13
      2.1.1 State and Measurement Equations ............................................. 13
   2.2 Models for Sharply-Maneuvering Targets ............................................. 14
      2.2.1 CT Model ............................................. 15
      2.2.2 Non-Zero Mean, White Noise Turn Rate Model ............................................. 16
      2.2.3 Mode mixing with Prior Distributions ............................................. 17
   2.3 Filter Design ............................................. 18
      2.3.1 IMM-CT Design ............................................. 18
      2.3.2 White Noise Turn Rate IMM Design ............................................. 18
Acknowledgements

I would like to thank my major co-advisors Dr. Yaakov Bar-Shalom and Dr. Peter Willett. Their knowledge and leadership has been tantamount in my learning experience at the University of Connecticut. I am especially grateful to them for allowing me to pursue research in challenging engineering problems and for believing that I would find solutions to these problems. Dr. Krishna Pattipati, the associate advisor, was my major advisor when I first enrolled in the Graduate school as a Masters student. Dr. Willett and Dr. Pattipati were professors in my undergraduate classes as well; their skill in teaching, as well as their passion for the subjects, are top reasons for my decision to enroll at UCONN for my graduate studies. I also appreciate Dr. Bar-Shalom’s recommendation after completing my first research paper that I pursue the Ph.D.

I would like to thank the entire ECE Faculty, the Department of Computer Science and Engineering, and the Department of Statistics for offering courses that enriched my technical learning experience. I am especially thankful to Dr. Dipak Dey: his experience with Bayesian statistics helped develop the idea to transform mode probabilities of the Interacting Multiple Model Estimator into log-ratios for the purpose of data fusion using IMM inside information.

I am grateful for my parents Gabriel M. and Victoria M. They are giant inspirations and foundational supporters for this accomplishment; I followed in their footsteps as a result of admiration and the awe of science and mathematics passed down to me.

My wife Sarah and son Theodorin are the strongest people that would have taken this ride with me. I love them to the galactic center and back.

I thank my entire family for their love and patience (Alan and Rebecca W. et al., Luana P. and Alex V. et al.). I would also like to thank my friends — Kaimei Yang, Andrew Finelli, Michael Kowalski, Qin Lu, Wenbo Dou, all my UCONN mates, and all those at home, thanks for cheering me on, helping me when I was confused, and forgiving me when I didn’t call back!
Chapter 1

Introduction

Maneuvering targets are of special interest in remote tracking because of the difficulty in modeling the changes in their dynamic behavior. Sharp maneuvers present challenges. For some targets, such as those moving in a 2D plane, the proposed White Noise Turn Rate (WNTR) model should be included as one or more modes in an Interacting Multiple Model (IMM) estimator. This model is similar to the conventional coordinated-turn (CT) model, also known as the Wiener Process Turn Rate model, but with non-zero mean turn rate process noise, instead of zero-mean turn rate process noise. Such an IMM design, possibly coupled with a CT mode as well to accommodate longer turns with low turn rate, results in better estimation accuracy and improved covariance consistency.

More sophisticated targets, such as those guided by feedback control systems in 3D space headed for a known or approximately-unknown destination, should be modeled using a guidance and destination model. The Proportional Navigation based estimation model discussed below describes the motion of such targets with the ability to model maneuvers as an approximately-Gaussian process noise in the target’s destination, where the destinations can switch to “fake” ones if the target exhibits evasive maneuvers on its way to its true destination. It is based on the concept of the Proportional Navigation guidance feedback control strategy implemented in real-world targets, where such a target reaches its destination by continuously attempting to reduce the angle rate of the line of sight (LOS) vector to zero. It is shown that the structure of the model, especially when used in an IMM estimator, allows for an intuitive and robust description of target maneuvers in 3D space with significant improvement in tracking accuracy and with destination estimation capability.

To strengthen the capability of tracking systems that operate using multiple independent track-
ers estimating the same target’s state, a data fusion method dedicated to information coming from IMM tracks is developed. The Fusion Center (FC) receives local IMM “inside information”: mode-conditioned estimates and mode probabilities representing the Gaussian mixture density information at the local trackers. The method is capable of on-demand fusion, where the fusion is performed without memory of past fused estimates. It is derived using Bayesian principles and proposes a first-order Taylor series approximation of the evolution of data computed by independent IMM estimators. Additionally, the solution includes the novel log-ratio (Log-R) transformation of mode probabilities that allows for a Gaussian approximation and Bayesian fusion. Simulations show that this fusion estimator has accuracy that is slightly better than fusion using outside information (local moment-matched IMM outputs) and that the consistency of the fused covariance matches the ideal consistency of the Centralized Measurement Fusion (CMF) covariance because of the computation and inclusion of crosscovariances in the fusion.

1.1 Outline

The rest of the sections in this dissertation are organized as follows. Chapter 2 discusses the use of the WNTR model in multiple-model estimators for tracking ground targets that can perform sharp heading maneuvers. Chapter 3 develops the Proportional Navigation estimation model for tracking maneuvering targets in 3D space that are guided towards specific destinations. Chapter 4 describes a novel approach to data fusion for tracking maneuvering targets using mode-conditioned estimates and mode probabilities from local IMM Estimator trackers.

1.2 Background on Multiple-Model Estimators for Tracking Sharply-Maneuvering Ground Targets

Tracking of remote targets is a challenging task because measurement systems are inaccurate, they are unaware of the input forces that may be applied to targets, and the targets’ dynamic properties cannot be modeled exactly. Some real-world targets, such as land vehicles with continuous wheel tracks, are capable of making very sharp, evasive maneuvers, effectively maintaining constant speed while making large changes in heading. Although a very different type of algorithm for zigzagging targets was developed in [47], little prior work has dealt with this specific type of maneuvering target. Targets with moderate and relatively smooth turn capability have been extensively studied.

The coordinated-turn (CT) dynamic model has been widely used for tracking air targets [33].
In this model, the turn rate $\omega(t)$ is a state and, in continuous time, evolves as a Wiener process (it is sometimes referred to as the Wiener-process turn rate model [11]). The turn rate results in lateral accelerations (normal to velocity) that dominate the maneuvering capability of the target. However, there are some issues with this model when it is considered for tracking sharply-turning targets. First, the most popular discretization of the coordinated-turn model approximates the Wiener process as a random walk, and this approximation degrades with decreased measurement rate and increased target maneuverability. An alternate discretization is proposed in [41], but it is still approximate. A white noise turn rate model that circumvents this issue is proposed here for dealing with sharply-maneuvering targets.

Kinematic targets that experience accelerations normal to their velocity can be described by nonlinear recursive equations. Similarly, when measurement systems report polar coordinates, they are described by nonlinear functions of the target state. In the current application, the dynamic equation contains the nonlinear effect of a turn rate, $\omega(t)$, either as a state to be estimated or as a stochastic input. The magnitude of the angle that the target’s velocity vector can turn over a sampling interval is quite high ($60^\circ$, for example) and this results in relatively high nonlinearity.

The linear minimum mean-square error filter for state estimation, the Kalman Filter, is not optimal in such situations. The most common suboptimal solution, the Extended Kalman Filter (EKF), uses first-order Taylor series approximations of the dynamic and/or measurement equations [11]. Second-order correction terms were first specified in [6], and the authors of [45] have generalized the algorithm to handle nonlinear stochastic inputs and have provided computational shortcuts as well. Due to its performance, mathematical tractability, and since it requires no design parameters, the second-order EKF (SOEKF) is used here and, therefore, Kalman-based filters are used. There are, however, many other choices for nonlinear filters. The Unscented Filter [29] is a computationally-efficient method that utilizes numerical differentiation to estimate the first two moments of the state. The particle filter [26] runs Monte Carlo simulations at each time step to estimate and propagate an approximate, non-Gaussian posterior state probability density function (PDF) using Bayes’ Theorem. The Cubature filter [5] uses numerical integration and Bayes’ Theorem to find the posterior state PDF, and only the first two moments are carried to the next time step. To handle the nonlinear measurement equation, we can opt for any of the previously-mentioned methods, or we can convert the measurements from polar to Cartesian coordinates. A method of unbiased conversions, which includes second-order corrections, is given in [40] and is used here.

It is stated in [29] that a motivation of the Unscented Filter is the need to calculate Jacobians.
and Hessians in the SOEKF. However, once they are calculated, they are very convenient and accurate, and the SOEKF does not require any design parameters related to its first- and second-order Taylor series approximations.

Alone, the nonlinear methods described above do not implicitly handle the non-predictable nature of random inputs that are dealt with in the realm of maneuvering target tracking. For example, if a kinematic target moves in a straight line for a long time, then suddenly executes a sharp turn, it is not ideal to model the turn rate process noise as an integral of Gaussian white noise (a Wiener process) because this integration implies memory and low-pass response in a target’s turn behavior, and the sharp turn maneuvers do not exhibit such Wiener process behavior. One of the first algorithms for dealing with maneuvering targets used statistical decision theory to augment the state of a constant-velocity model with an acceleration variable when maneuvers are detected [39]. Input Estimation, which uses a history of measurement residuals to detect and estimate the magnitude of a step function input, was developed in [18].

When a target exhibits changes in the nature of its dynamics, then the system parameters can be modeled as Markov jump processes and a unified framework of estimating both state and parameters can be used, as in [49] and [21]. Multiple-model algorithms, where the target’s current system equation is one in a set of known “modes,” are a modern choice for tracking maneuvering targets. The optimal multiple model estimator, along with the first suboptimal approximation (GPB1, the Generalized Pseudo-Bayesian Multiple-Model Estimator of First Order), was presented in [2]. The GPB2 algorithm was first given in [19]. The IMM, which typically performs as well as GPB2 but with much less computation, was first introduced in [14]. Since its introduction, it has generally been accepted as the best overall algorithm for maneuvering target tracking (and other multiple-model system applications) when considering both accuracy and computational load [10, 11, 34], and it is used in this paper. Since then, the Viterbi algorithm has also been adapted to the area of multiple-model switching systems [25], and a B-Best mode history algorithm [49] has also been developed. However, while theoretically sound, the latter two algorithms are hard-decision based, and were shown [43] to have weak performance when the true target model is not among the models in the set designed for the algorithms. Variable-structure multiple model estimation, introduced in [30], has been heavily researched in the past decade, but its inclusion into the present problem is unnecessary due to the small number of modes in the model sets that are presented here. A very different approach was taken in [36] where a very unpredictable target is tracked using a non-Bayesian segmenting algorithm.
A thorough survey of maneuvering target tracking can be found in [33], [35], [31], [32], and [34]. In that survey, as well as in [10] and [11], it is often the case that the best algorithms (considering both accuracy and computational load) for tracking point targets with coordinated turn capability in 2D are IMMs having a low-process-noise White Noise Acceleration (WNA) mode and one or more Nearly-Coordinated Turn (CT) modes (also known as Wiener Process Turn Rate modes) in the model sets. In 2D, such an algorithm is known as IMM-CT. In [37], [50] and [23], significant improvements were made to IMMs that include mode models with state vectors that don’t contain the same state variables. The work of [23] encompasses the other two, and we will use it to correctly design prior distributions to be used in mixing the modes that contain the turn rate state $\omega(t)$ with those that do not.

Some of the filters presented here rely on non-zero-mean process noise models to represent straight, left-, and right-turning target modes. This corresponds to each mode having an approximately-known input. This method has been explored only in part in the literature. It was successfully applied in [43] by using multiple left- and right-turn modes, but there was no indication that this was a better method than using an IMM-CT with multiple maneuvering modes. Also, the authors in [43] treated the turn rate in each mode as precisely known with no variance – this probabilistically incorrect in the present application. This paper generalizes the concept of non-zero-mean process noise to handle the quick, sharp turns that some ground vehicles can perform, without assuming that the turn rate is known exactly in any mode.

1.3 Background on Tracking Highly-Maneuvering Targets with Destination Information

Estimating the state of a time-varying, non-linear stochastic system is a challenging problem when the input is not known and has time-varying statistical properties. When the input is non-zero, yet is unknown to an estimator, the estimation problem can be described as that of estimating the state of a maneuvering system (or target).

Decreased prediction uncertainty can be achieved when a target’s feedback control system can be modeled. Some authors have considered the problem of track smoothing for trajectories with known start and end points using linear motion models and general Bayesian algorithms [7, 17]. In this manuscript, the specific task of tracking a maneuvering target in 3D space is considered and is placed in the familiar framework of nonlinear Kalman filtering and the Interacting Multiple Model
Estimator (IMM). Even without direct knowledge of the plant input (lateral acceleration in this case), certain system properties (such as reference setpoints, feedback gains, energy constraints, and/or actuator constraints) can be inferred or estimated to some degree. This technique can provide the filter with a dynamic system description that is more accurate in its prediction than one where only the kinematic states are described. When only these are modeled, the unknown input is typically described as additive white noise, such as in a Kalman filter [11]. Instead of this simplistic approach, stochastic parameters of the feedback loop, such as reference setpoint and/or feedback gain, which yield random inputs, can be estimated. Uncertainty in the order of the loop compensator is analogous to uncertainty in the order of kinematic states of a kinematic target described by linear and/or circular motion. The resulting increased prediction accuracy can be very significant. We shall show how this increased prediction accuracy reduces the updated position and velocity estimation error in a 3D maneuvering state estimation scenario.

A feedback control process model in a single-model or multiple-model estimator can improve the state estimation of a system that is truly under a feedback control policy. This technique is very useful for estimating systems whose dynamics reflect the goal of reaching a destination position, yet are constrained by energy and maneuverability. Although the exact control policy of the system is generally unknown, energy costs and actuator saturation are two typical attributes that are considered when designing the feedback control law. So, there may exist a model with observable stochastic parameters and/or observable random constant parameters that will approximate the true dynamics. For example, autonomous guidance systems are typically under some variant of Proportional Navigation [3], [46], which is designed to effectively engage a moving object with minimal actuation. Even if random maneuvers are performed along the way, the guidance system must eventually resume an engagement trajectory to reach the destination. Also, even if the control policy is not adequately described by Proportional Navigation, such a model is of practical importance in describing a maneuvering target in 3D space because a Gaussian random process can adequately model the semi-random changes and/or modeling uncertainties in the target’s turns, and it can describe turns in any plane. An effective remote state estimator would need to estimate the Proportional Navigation feedback gain parameter and the position of the target’s destination (if unknown), both of which may be time-varying. If the gain parameter and the destination are observable, then the position, velocity, and acceleration of the target can be estimated with decreased error due to the improved dynamic model.

Targets that experience accelerations normal to their velocity can be described by nonlinear
recursive equations. Similarly, when remote measurement systems report in spherical coordinates, they are described by nonlinear functions of the target state. In the current application, the dynamic equation contains the nonlinear effects of lateral, longitudinal, and gravitational accelerations. The minimum mean-square error filter for state estimation in linear systems with additive, white Gaussian noise is the Kalman Filter, but other techniques are required for estimation in nonlinear systems. The most common suboptimal solution is the Extended Kalman Filter (EKF), which uses first-order Taylor series approximations of the dynamic and/or measurement equations [11]. Second-order correction terms were first specified in [6], and the authors of [45] have generalized this Second Order EKF to handle nonlinear stochastic inputs and have provided computational shortcuts as well. The Unscented Kalman Filter (UKF) [29] is a flexible, computationally-efficient method that utilizes real-time, empirical evaluation of the first two moments of the state probability density, and it is used here in the simulations. The particle filter [26] runs Monte Carlo techniques at each time step to estimate and propagate an approximate, non-Gaussian posterior state probability density function (PDF) using point masses (“particles”) and Bayes’ Theorem. The Cubature filter [5] uses numerical integration and Bayes’ Theorem to find the posterior state PDF, and only the first two moments are carried to the next time step. To handle the nonlinear measurement equation, any of the previously-mentioned methods can be used; or, the measurements can be converted from spherical to Cartesian coordinates. A method of unbiased spherical-to-Cartesian conversions is given in [40].

When a system exhibits changes in the nature of its dynamics, such as a maneuvering target that performs occasional random turns, then the system parameters can be modeled as Markov jump processes and a unified framework of estimating both state and parameters can be used, as in [21] and [49]. One of the first methods for dealing with maneuvering targets that used statistical decision theory to augment the state of a constant-velocity model when maneuvers are detected is [39]. The theoretical (but infeasible) optimal multiple model estimator for Markov switching systems, along with the first suboptimal approximation (GPB1, the Generalized Pseudo-Bayesian Multiple-Model Estimator of First Order) was presented in [2]. The more accurate GPB2 estimator was first given in [19]. Input Estimation (not to be confused with the feedback control model proposed in this paper) uses a history of measurement residuals to detect and estimate the magnitude and onset time of a step function input [18]. The Interacting Multiple Model (IMM) estimator was first introduced in [14]. Since its introduction, the IMM has generally been accepted as the best overall filter for Markov switching systems when considering both accuracy and computational load [10, 11, 34]. A
survey of maneuvering target tracking can be found in [31], [32], [33], [34], and [35].

Conventional methods to estimate the state of a maneuvering target in 3D space include position and velocity as state variables, and some models include some combination of Cartesian acceleration, turn rate, turning plane, angular velocity of velocity vector, lateral accelerations, and/or longitudinal accelerations [33]. Some of the previous models mentioned in [33] were developed in [15], [16], and [42]. These previous methods did not consider dynamic models based on feedback control, and no model has shown to be effective in estimating turn maneuvers in a time-varying plane of rotation. However, some work has been done investigating the use of the Proportional Navigation policy as a model to predict the future location of airborne threats [24], but the authors did not make use of this prediction to improve current state estimation accuracy, prediction covariance was not provided, random maneuvers were not considered, and the destination was assumed to be known, though it may not be; the algorithm developed in this paper provides solutions to these important problems.

1.4 Background on Data Fusion using IMM Inside Information

The IMM Estimator is a powerful non-linear state estimator for targets whose dynamic evolution model changes according to a discrete-time, discrete-state Markov chain with known transition probabilities, and it may be used in local estimators for tracking maneuvering targets or other mode-switching systems. In this work, the posterior probability density function (PDF) of the state of a dynamic target, conditioned on information from local trackers (LT) implementing the IMM estimation algorithm [11], is derived for on-demand data fusion. The LT provide Gaussian mixture track information from inside their IMM algorithms (i.e. mode conditioned estimates and mode probabilities). The Fusion Center (FC) continuously updates a linearized system description of the IMM Estimator’s error and mode probability behavior to compute the required parameters of the likelihood functions of the state and mode. The fused posterior state PDF is a Gaussian mixture.

When new measurements from every sensor can be communicated to a Fusion Center (FC) at every measurement time, the optimal solution is to stack all new measurements into a single vector and run a single estimator, resulting in optimal centralized measurement fusion (CMF) [13]. However, data may need to be sent at arbitrarily low rates compared to the LT measurement intervals, requiring the transmission of recursively-computed local estimates and covariances. The
problem is difficult because of the dependent nature of the received state estimation errors. The correlation between the local estimation errors was described in [8], [9] as the recursively-computed crosscovariance matrices for linear, Gaussian estimators, and their incorporation into the standard fusion equations results in consistent fused covariances. The recursive computations described in the present paper yield the required matrices for IMM track fusion as well.

For the fusion of IMM mode-conditioned information, the extraction of track information from mode probabilities (i.e. the mixture weights) is an additional problem. This suggests that the received probabilities should be treated as conditioning information in the posterior (fused) PDF. To account for the dependency between the received probabilities, they are transformed into infinite-support log-ratios (Log-R) of probability pairs and finding an approximation of their evolution.

Fusion can be performed naively using the LT’s moment-matched IMM output estimate and covariance, but that method has poor performance during maneuvers and does not account for error correlations. Alternate Gaussian Mixture fusion approaches have been explored — Chernoff fusion [22], [27], [38], being a generalization of Covariance Intersection fusion [28], has received some attention in the literature, but does not utilize system model information (assumed available in the present paper) and may be too conservative in the computed mean-squared error (MSE). A heuristic method for high-rate fusion was developed in [1], but this method requires past fused tracks, does not utilize measurement and dynamic model information, and does not provide the crosscovariance for fusion at low rate.

The algorithm presented here requires trackers that agree on the dynamic modes of the target and the mode transition probabilities. The FC is assumed to know the parameters of the local IMM’s state and measurement equation parameters and the information from the trackers is for the same times (i.e. it is synchronous). It is also assumed that the state transition matrix is the same for both modes, so the method is ideal for target models that switch process noise covariance.

1.5 Summary of Contributions

1.5.1 Contributions Related to Tracking Sharply-Maneuvering Targets in 2D Space

This dissertation aims to advance the engineering subject of adaptive state estimation, with specific applications to the remote tracking of maneuvering targets. The primary contributions are the novel models developed for solving the problems, which are developed from first principles.
In the application of tracking targets in a 2D plane where the velocity vector can occasionally turn up to $60^\circ$ over one sampling interval with mostly straight-line motion (Sec. 2), the non-zero mean White Noise Turn Rate (WNTR) model is proposed and placed in the Interacting Multiple Model Estimator (IMM) to solve this problem, where left and right turns have, respectively, negative and positive turn rate means and added Gaussian turn rate noise. These models correspond to sharply-maneuvering modes and allow for quick, transient changes in the mode probabilities to increase the filter bandwidth for only a small number of sampling intervals.

1.5.2 Contributions Related to Tracking Maneuvering Targets in 3D Space

For tracking non-thrusting targets whose dynamics include destination seeking (Sec. 3), the new Proportional Navigation (PN) Estimation model, which is designed based on general knowledge of the Proportional Navigation guidance feedback control strategy, is developed. There are at least three reasons for describing such a model.

The first reason is the prevalence of PN in guidance control theory and applications. The second reason is its flexibility in describing maneuvers: “fake” destinations represent deviations of the trajectory from the optimal one, where the target’s temporary destination may be switched to somewhere in a space surrounding the true destination, and this space can be easily described by a Gaussian position density.

The third reason is the substantial increase in tracking accuracy. Another contribution of this section is the novel drag estimation model, which is based on the exponential model of Earth’s atmospheric pressure. This model does not require knowledge of the target’s drag coefficient and does not seek to estimate it, yet it estimates the drag acceleration itself with high accuracy due to the demonstrated observability. It is shown, through simulation, that state estimators using PN models have observability in the PN feedback gain, the target’s current destination position, and the drag acceleration.

1.5.3 Contributions Related to Track-To-Track Fusion for Maneuvering Targets

On the topic of maneuvering target data fusion using Interacting Multiple Model (IMM) inside information (Sec. 4), this dissertation shows that it is possible to fuse mode-conditioned estimates and mode probabilities form local IMM trackers with better performance than fusion using local
moment-matched IMM outputs (outside information). Fusion using outside information suffers from poor consistency in the fused covariance (50% higher than ideal according to the simulations) and has excessively high error during target maneuvers. This is done through three primary contributions.

The first contribution in this area is the unique proposal to model IMM Estimators tracking the same target as a single non-linear dynamic system which can be linearized using the first-order Taylor series expansion. The states of this system are the zero-mean local mode-conditioned state estimate errors (MCEEs) and the mode probabilities. Since the method is Bayesian, the parameters of the likelihood functions of the mode-conditioned state and of the mode are required. Under linear-Gaussian mode assumptions, the likelihood functions are Gaussian, showing that the required parameters are the covariances of the MCEEs and the means/covariances of the log-ratio transformations of the mode probabilities (to be described). The parameters must be computed under every mode hypothesis. No literature to date has proposed a method to derive on-demand fusion parameters for nonlinear systems (such as IMM estimators tracking the same target), and this paper shows that modeling the scenario as a joint system allows for linearization that recursively computes the required means and covariances. Due to the potential for an exponentially-growing mode hypothesis history, a mixing strategy is introduced that is similar to the IMM initial condition mixing but which runs at the Fusion Center (FC).

The second contribution is the log-ratio transformation of the mode probabilities. This transformation maps the mode probabilities to the real line, which allows them to be approximately modeled as jointly-Gaussian processes.

The third contribution involves modeling the evolution of local tracker matrices that are hidden to the Fusion Center. To compute the required means and covariances, the FC must compute expected values of the Kalman gains and innovation covariances at the local trackers under every mode hypothesis, and this is accomplished recursively using linearization, recursion, and mixing methods similar to those described above. According to simulations, the fusion strategy has performance close to that of Centralized Measurement Fusion (CMF), including ideal covariance consistency. Compared to fusion using outside information, it has less estimation error during target maneuvers, where the error of fusion using the outside information can be excessive.
1.6 Publications to Date

1.6.1 Journal Papers

Journal papers with primary authorship that have been accepted and published:


Journal paper with primary authorship that has been submitted:


1.6.2 Conference Proceedings Papers

Conference papers with primary authorship that have been accepted and published:


Conference paper with co-authorship that has been accepted and published:

Chapter 2

Multiple-Model Estimators for Tracking Sharply-Maneuvering Ground Targets

2.1 Problem Formulation

2.1.1 State and Measurement Equations

The target is modeled as a point in 2D Cartesian space with state vector

\[
\mathbf{x}(t) = [s(t) \quad \dot{s}(t) \quad \omega(t)]' = [x(t) \quad y(t) \quad \dot{x}(t) \quad \dot{y}(t) \quad \omega(t)]'
\]  

(2.1)

where \(s(t) = [x(t) \quad y(t)]'\) is the Cartesian position, \(\dot{s}(t) = [\dot{x}(t) \quad \dot{y}(t)]'\) is the Cartesian velocity, and \(\omega(t)\) is the turn rate of the velocity vector. In this paper, \(\omega\) is a state variable only in the Coordinated Turn (CT) model. Our goal is to find the most accurate estimates of the state \(\hat{x}(k|k)\) from noisy position measurements. We are also interested in finding an estimate of the mean-square error (MSE) of the state estimate, \(P(k|k)\). We are seeking (i) higher accuracy than can be obtained from the raw measurement system and (ii) consistency in the estimated MSE.

During a coordinated turn, the velocity vector rotates according to the turn rate \(\omega\). To model sharp maneuvers, such as those that can be undertaken by continuously-tracked land vehicles deliberately attempting to evade conventional tracking systems and/or incoming weapons, we shall model and simulate high coordinated turn rates for short time periods. Positive turn rate implies counter-clockwise (left) rotation of the velocity vector.
The dynamic state evolution is described by the discrete-time system

\[ x_{k+1} = f[x_k, v_k] \]

(2.2)

where \( f \) is the non-linear state transition function between samples and \( v(k) \) is a white process noise vector that represents random inputs or the model’s inaccuracies. The process noise vector \( v_k \) has mean \( \bar{v} \) and covariance matrix \( Q \).

The measurement equation in polar coordinates is

\[ z_p(k) = [r(k) \quad \theta(k)]' = h(x_s(k), x(k)) + w_p(k) \]

(2.3)

where \( z_p(k) \) is the vector that represents the noisy polar measurements of range \( r(k) \) and angle \( \theta(k) \), and \( x_s(k) \) is the state vector of the measurement sensor. The two-dimensional, additive, zero-mean, white, Gaussian vector \( w_p(k) \) describes the noise in the measurement, and

\[ R_p(k) = E[w_p(k)w'_p(k)] = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \]

(2.4)

is its covariance matrix.

The target motion is expected to consist mostly of straight-line motion, fast left/right turning capability, slow turning capability, fast deceleration (braking) capability, and relatively slow acceleration capability. Although established methods can be used, such as IMM-CT and its variants, these methods were mostly designed for aerospace applications where the maximum turns over a single sampling interval are only a few degrees, and where the turn rate itself cannot suddenly change because aircraft must bank to turn. However, it is critical that a filter can detect maneuvers and correct the state estimate as quickly as possible in the present application. It is also important to achieve very low tracking error during straight-line motion, where the dynamics are linear. All this should be done while maintaining the tracking accuracy better than the raw measurements.

### 2.2 Models for Sharply-Maneuvering Targets

Our algorithms are based on multiple-model (MM) methods. For each filter, \( n_r \) will be the number of modes in the model set and the first mode for all filters will use the non-maneuvering White Noise Acceleration (WNA) model. This linear model is typical for targets that perform only occasional accelerations and can be reviewed in [11].
2.2.1 CT Model

The CT model has been highly successful for tracking targets that are capable of true coordinated turns, such as aircraft. Omitting the time parameter, the continuous-time model is

$$
\begin{bmatrix}
\dot{s} \\
\dot{\omega}
\end{bmatrix} =
\begin{bmatrix}
0_{2\times2} & I_{2\times2} & 0_{2\times1} \\
0_{2\times2} & \omega U & 0_{2\times1} \\
0_{1\times2} & 0_{1\times2} & 0
\end{bmatrix}
\begin{bmatrix}
s \\
\dot{s} \\
\omega
\end{bmatrix}
+ 
\begin{bmatrix}
0_{2\times3} \\
0_{2\times1} \\
I_{3\times3}
\end{bmatrix}
\begin{bmatrix}
v_x \\
v_y \\
v_\omega
\end{bmatrix}
$$

(2.5)

where $I_{m\times n}$ is the $m \times n$ identity matrix, $0_{m\times n}$ is the $m \times n$ matrix of zeros, and

$$
U = 
\begin{bmatrix}
0 & -1 \\
1 & 0
\end{bmatrix}
$$

(2.6)

is a constant skew-symmetric matrix. The random sequences $v_x(t)$, $v_y(t)$, and $v_\omega(t)$ are the components of the process noise $v(t)$, with $v_\omega(t)$ the white noise turn acceleration that enters the system to yield a random walk (i.e. an approximate Wiener process) turn rate.

In discrete time, (2.5) is typically approximated as

$$
\begin{bmatrix}
s_{k+1} \\
\dot{s}_{k+1} \\
\omega_{k+1}
\end{bmatrix} =
\begin{bmatrix}
I_{2\times2} & \frac{1}{\omega_k} U (I - V_k) & 0_{2\times1} \\
0_{2\times2} & V_k & 0_{2\times1} \\
0_{1\times2} & 0_{1\times2} & 1
\end{bmatrix}
\begin{bmatrix}
s_k \\
\dot{s}_k \\
\omega_k
\end{bmatrix}
+ 
\begin{bmatrix}
\frac{T^2}{2} I_{2\times2} & 0_{2\times1} \\
TI_{2\times2} & 0_{2\times1} \\
0_{1\times2} & T
\end{bmatrix}
\begin{bmatrix}
v_{x,k} \\
v_{y,k} \\
v_{\omega,k}
\end{bmatrix}
$$

(2.7)

where $T$ is the measurement interval, $e^A$ is the matrix exponential of $A$, and

$$
V_k = e^{\omega_k T U} =
\begin{bmatrix}
\cos \omega_k T & -\sin \omega_k T \\
\sin \omega_k T & \cos \omega_k T
\end{bmatrix}
$$

(2.8)

is a rotation matrix. To use (2.7), the turn rate $\omega_k$ is evaluated at the last estimate of the turn rate, $\hat{\omega}_{k|k}$. It is evident that the evolution of $\omega_k$ in (2.7) is a random walk approximation of the Wiener process in (2.5). The turn rate is assumed constant throughout the state prediction interval $T$, so effects of the turn acceleration $v_k$ are not propagated to the position and velocity vectors within the sampling interval. This issue is discussed in [41] and an alternate discrete approximation is proposed. The discrete-time CT model given in (2.7) can also be improved by running a prediction-only filter for $m$ time intervals of length $\frac{T}{m}$ in between measurements for more accurate integration, at the expense of more computation [11].

Even with effective nonlinear filtering methods, the continuous-time CT model itself is still not a match to reality. This is because the sharp turns are of short duration and, when they do happen,
the turn rate values are nearly independent of previous ones. We need to detect these maneuvers very quickly, but we don’t want to sacrifice too much accuracy when there are no turns. Just increasing the variance of the turn acceleration $\sigma_\omega^2$ needlessly increases the filter bandwidth during non-maneuvering, straight-line motion.

2.2.2 Non-Zero Mean, White Noise Turn Rate Model

If we want to detect occasional sharp maneuvers among nearly constant velocity motion, then it helps to model the turn rate process noise as a Gaussian mixture density (possibly multi-modal) consisting of 3 Gaussian terms with different means: no turn (zero mean), left turn (positive mean), and right turn (negative mean). With a recursive estimation algorithm that requires that the process noise is approximated as a white Gaussian process, this is not possible. Within a multiple-model estimator, however, this is very straightforward. In fact, we can approximate any non-Gaussian PDF using a Gaussian mixture [4]. Multiple-model estimators, such as the IMM, also allow us to describe the time correlation between modes in the form of a Transition Probability Matrix (TPM) [11]. We can match the TPM to the expected behavior of the target by using the mean sojourn time of mode $i$, $E[\tau_i]$, with

$$p_{ii} = 1 - \frac{T}{E[\tau_i]}$$  \hspace{1cm} (2.9)

being the probability of remaining in mode $i$. If $E[\tau_2] = T$, then $p_{22} = 0$. That means that the sharply-maneuvering mode is expected to last one sample and no longer.

To approximate multi-modal PDFs for the turn rate process noise, as we need for the occasional sharp maneuver, we can utilize non-zero mean process noise modes in a multiple-model estimator. We call a model of these modes the Non-Zero Mean White Noise Turn Rate (WNTR) model. The state is

$$\mathbf{x}_k = [s_k \ \dot{s}_k]^T$$ \hspace{1cm} (2.10)

and the motion is described by

$$
\begin{bmatrix}
\mathbf{s}_{k+1} \\
\dot{\mathbf{s}}_{k+1}
\end{bmatrix} =
\begin{bmatrix}
I_{2 \times 2} & \frac{1}{\omega_k} U (I - V_k) \\
0_{2 \times 2} & V_k
\end{bmatrix}
\begin{bmatrix}
\mathbf{s}_k \\
\dot{\mathbf{s}}_k
\end{bmatrix} +
\begin{bmatrix}
\frac{T^2}{2} I_{2 \times 2} & 0_{2 \times 1} \\
TI_{2 \times 2} & 0_{2 \times 1}
\end{bmatrix}
\begin{bmatrix}
v_{x,k} \\
v_{y,k} \\
v_{\omega,k}
\end{bmatrix}
$$ \hspace{1cm} (2.11)

In (2.11), $v_{\omega,k}$ is the white noise turn rate. The Second Order EKF will be used to propagate the state prediction $\hat{\mathbf{x}}_{k+1|k}$ and the state covariance prediction $\mathbf{P}_{k+1|k}$, with $\omega_k = \bar{v}_{\omega}$, the mean of the
white noise turn rate process noise. We propose that when tracking sharply-maneuvering targets, it is advantageous to have at least two WNTR modes to correspond to right and left turns.

An advantage of this non-zero mean process noise in a multiple model filter is that the measurement prediction $\hat{z}^{(i)}_{k+1|k}$ of each mode $i$ is unique, unlike the IMM-CT, where if $\hat{\omega}_{k|k}$ is zero, then the mode-conditioned measurement prediction of the maneuvering CT mode $\hat{z}^{(2)}_{k+1|k}$ is almost the same as that of the non-maneuvering mode $\hat{z}^{(1)}_{k+1|k}$, resulting in less ability to discriminate between modes. Utilizing WNTR modes guarantees that the mode-conditioned state predictions are unique at all times since the mean of the turn rate process noise of each mode is unique, which results in unique measurement predictions. Higher-magnitude measurement residuals (innovations) will result in smaller values of the likelihood functions for modes that are not in effect, thereby increasing the tracking accuracy during straight-line motion and allowing for fast detection/correction of turning maneuvers.

Even though the WNTR modes are designed for very sharp maneuvers, this filter can also perform well during coordinated turns, thanks to the mode mixing. For example, during a coordinated turn to the left, the estimator mixes in some of the left turn mode’s output, resulting in a balance between sharp turns and CV. While it will be shown that this does provide very good position and velocity estimates, this method does not estimate the turn rate, which means that it is not suited for situations where long-term target predictions are needed. To remedy this, we can also include a CT mode with low process noise in the mode set. This should provide the benefit of fast maneuver detection, as well as having a running estimate of $\omega$.

### 2.2.3 Mode mixing with Prior Distributions

If we introduce a CT mode in the IMM, then we must handle the mixing of modes that have different state vector components. The CT mode includes the $\omega$ state component, while the WNA and WNTR models do not. According to the work in [23], we should use a prior distribution of the $\omega$ state variable when transitioning from WNA/WNTR to CT. We must select a prior mean $\tilde{\omega}^{(i)}_k$ and variance $\tilde{\sigma}^2_{\omega,k}$ of the turn rate for each non-CT mode $i$ by asking “If the previous mode was WNA/WNTR, then what is the expected turn rate when starting the CT mode, and how much variance goes with that expected turn rate?”

To answer this question, we first note that the WNA model has a definite zero turn rate at all
When entering CT from WNTR, we have
\[
\tilde{\omega}_k^{(iWNA)} = 0
\]
\[
\tilde{\sigma}_{\omega,k}^{2(iWNA)} = 0
\]  \hspace{1cm} (2.12)

meaning that the expected turn rate when exiting the WNTR mode is the mean of the white noise turn rate and the variance of this prior is equal to the variance of the white noise turn rate. \(\tilde{v}_\omega^{(iWNTR)}\) and \(\sigma_{\omega,k}^{2(iWNTR)}\) are design parameters that are selected for each WNTR mode.

### 2.3 Filter Design

#### 2.3.1 IMM-CT Design

The first filter to test is a simple IMM-CT with mode 1 a WNA mode and mode 2 a CT mode matched to the possibility that the target can sharply turn its velocity vector within a single measurement interval. In mode 1, we set \(\sigma_x = \sigma_y\) to a low value of 0.1 m/s\(^2\) to cover small deviations from CV (and we use these linear acceleration standard deviation parameters in all subsequent modes designed in this paper). In mode 2, the CT mode model, we keep \(\sigma_x = \sigma_y = 0.1\) m/s\(^2\) and set \(\sigma_{\omega}^{(2)} = 60^\circ/s^2\) to handle sharp-turn maneuvers. This parameter, the turn acceleration process noise standard deviation, is integrated as a constant over a measurement interval \(T\) to result in the target’s most probable sharp turn rate, \(\sigma_{\omega}T\), which is \(60^\circ/s\) in this application. Initially, we set mean sojourn times as \(E[\tau_1] = E[\tau_2] = 10\) s, which results in a typical IMM-CT design. This filter is labeled IMM-CT-S. Then, we form a second IMM to better match to the the fast maneuvers by setting \(E[\tau_2] = 1\) s, which results in \(p_{22} = 0\), an unconventional IMM design. This filter is labeled IMM-CT-F.

#### 2.3.2 White Noise Turn Rate IMM Design

The IMM-WNTR filter has a WNA mode (mode 1), a right-turn WNTR mode (mode 2), and a left-turn WNTR mode (mode 3). The model parameters are: \(\tilde{v}_\omega^{(2)} = -60^\circ/s\), \(\tilde{v}_\omega^{(3)} = 60^\circ/s\), and \(\sigma_{\omega}^{(2)} = \sigma_{\omega}^{(3)} = 30^\circ/s\). We set \(E[\tau_1] = 10\) s, \(E[\tau_2] = E[\tau_3] = 1\) s. This gives \(p_{11} = 0.9\) and
\[ p_{22} = p_{33} = 0. \] We split the remaining probabilities by setting the TPM to
\[
\Pi_{WNTR} = \begin{bmatrix}
p_{11} & \frac{1-p_{11}}{2} & \frac{1-p_{11}}{2} \\
1 & 0 & 0 \\
1 & 0 & 0
\end{bmatrix}
\] (2.14)

Fig. 2.1 shows approximately what the maneuvering mode turn rate process noise looks like as a Gaussian sum mixture. Notice that by using the non-zero mean turn rate process noise, we can approximate a multi-modal process noise PDF.

### 2.3.3 IMM-WNTR-CT Design

To design a filter that will also provide an estimate of the turn rate, we add a CT mode (4th mode) to the IMM-WNTR to create the IMM-WNTR-CT. This filter has the same parameters as in Sec. 2.3.1 except \( \sigma_{\omega}^{(4)} = 1^\circ/s^2 \) and \( E[\tau_4] = 10 \) s. The TPM is
\[
\Pi_{CT-WNTR} = \begin{bmatrix}
p_{11} & \frac{1-p_{11}}{4} & \frac{1-p_{11}}{4} & \frac{1-p_{11}}{2} \\
0.5 & 0 & 0 & 0.5 \\
0.5 & 0 & 0 & 0.5 \\
1 - p_{44} & 0 & 0 & p_{44}
\end{bmatrix}
\] (2.15)
2.3.4 IMM-CT-LH Design

To ensure that adding two WNTR modes results in better performance than adding a single CT mode with high turn acceleration process noise, we will compare IMM-WNTR-CT to a 3-mode IMM-CT-LH (referring to “Low” & “High” turn process noises, respectively). It contains a WNA mode (mode 1), a low process noise CT mode (mode 2), and a high process noise CT mode (mode 3). We set $\sigma_\omega^2(2) = 1^\circ/s^2$, $\sigma_\omega^2(3) = 60^\circ/s^2$, $E[\tau_2] = 10$ s, and $E[\tau_3] = 1$ s. The TPM is

$$\Pi_{\text{CT-LH}} = \begin{bmatrix} p_{11} & \frac{1-p_{11}}{2} & \frac{1-p_{11}}{2} \\ \frac{1-p_{22}}{2} & p_{22} & \frac{1-p_{22}}{2} \\ 0.5 & 0.5 & 0 \end{bmatrix} \quad (2.16)$$

2.4 Simulations and Results

2.4.1 Simulated Trajectory and Measurements

For numerical simulations, we use a target trajectory shown in Fig. 2.2, along with an example of the noisy measurements. The sampling interval is $T = 1$ s. The target starts with $x_0 = 0$ m, $y_0 = 500$ m, $\dot{x}_0 = 0$ m/s, and $\dot{y}_0 = 5$ m/s. The simulated trajectory is designed to assess the algorithms’
<table>
<thead>
<tr>
<th>Label</th>
<th>Start Time (s)</th>
<th>Duration (s)</th>
<th>Turn Rate (°/s)</th>
<th>Long. Acc. (m/s²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>10.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>10.5</td>
<td>1.9</td>
<td>-60 (R)</td>
<td>-1 (br)</td>
</tr>
<tr>
<td></td>
<td>12.4</td>
<td>2.3</td>
<td>0</td>
<td>1 (acc)</td>
</tr>
<tr>
<td>C</td>
<td>14.69</td>
<td>10.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>24.8</td>
<td>1.2</td>
<td>-60 (R)</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>26.08</td>
<td>20.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>46.28</td>
<td>0.9</td>
<td>-30 (R)</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>47.18</td>
<td>11.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>58.58</td>
<td>30.6</td>
<td>5 (L)</td>
<td>0</td>
</tr>
<tr>
<td>I</td>
<td>89.18</td>
<td>10.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>J</td>
<td>99.68</td>
<td>1.0</td>
<td>0</td>
<td>-5 (br)</td>
</tr>
<tr>
<td></td>
<td>100.68</td>
<td>5.2</td>
<td>0</td>
<td>1 (acc)</td>
</tr>
<tr>
<td>K</td>
<td>105.88</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>115.88</td>
<td>0.8</td>
<td>80 (L)</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>116.68</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.1: Trajectory Description with maneuver labels. R = Right Turn, L = Left Turn, br = Longitudinal braking maneuver, acc = Longitudinal acceleration maneuver. 2 maneuver labels are omitted to reduce visual clutter in the simulation results.

behavior for targets that perform very sharp turns, rapid decelerations, moderately-sharp turns without deceleration, smooth coordinated turns, and longitudinal acceleration. Randomly-selected, fractional time intervals are used so that maneuvers do not necessarily start at the sampling instants. A description of the trajectory is given in Table 2.1.

The total simulation time is 126.68 s. The measurements are made in polar coordinates, $r_k, \theta_k$, with a sensor located at the origin. The measurement standard deviations are constant in polar coordinates and are given as $\sigma_r = 1.5$ m, $\sigma_\theta = 1.15^\circ$. The unbiased conversion of [40] is used to feed Cartesian measurements to all filters, resulting in all subsequent filter operations performed in Cartesian coordinates with position-dependent measurement variance; 1000 Monte Carlo runs are used for all performance evaluations.
Figure 2.3: Position estimate RMSE for 5 filters and raw measurements

Figure 2.4: Velocity estimate RMSE for 5 filters
2.4.2 RMSE and NEES Results

The position root-mean-square error (RMSE) across the five estimators and the raw measurements from Monte Carlo simulations is shown in Fig. 2.3. The velocity RMSE is shown in Fig. 2.4 and the Normalized Estimation Error Squared (NEES [11]) is shown in Fig. 2.5 for consistency evaluation. The NEES is calculated using only the position and velocity state components, and it is normalized by 4 to achieve a nominal value of 1 for ideal consistency. Maneuvering intervals are drawn with black rectangles along the time axis, referenced by the trajectory segment labels in Table 2.1. It can be seen from Fig. 4.5 that all filters, except IMM-CT-S and IMM-CT-F, achieve position RMSE that stays well below the raw measurements, despite relatively high measurement noise. Fig. 2.6 shows closer views of the performance measures for 3 maneuvers: the $-30^\circ$/$s$ sharp turn (segments, D, E), the the $80^\circ$/$s$ sharp turn (segments L, M), and the $5^\circ$/$s$ slow turn (segment H). IMM-CT-S does have the best performance during non-maneuvering segments, but IMM-CT-S and IMM-CT-F do not show comparable RMSE reduction after sharp maneuvers.

It can be seen from the results that when it comes to position and velocity error, the IMM-WNTR is a very accurate filter. It is slightly outperformed after the first extreme turn maneuver (see segment B), whose 2-second extreme turn and simultaneous braking mode is not matched to
Figure 2.6: Magnified plots showing RMSE and NEES during and after 3 key maneuvers. Axis units are the same as in Figs. 2.3, 4.6, and 4.7.
any of the filters. Apart from IMM-CT-S, which does not perform well after sharp maneuvers, it achieves the lowest RMSE in most other segments (see segments D, E, most of H, and K). It may actually be surprising that it even achieves the best overall performance during the slow coordinated turn (segment H). However, the IMM-WNTR is not a good choice if long-term predictions are needed, as is the case when measurements are slow to arrive. This is because the filter does not attempt to estimate the turn rate $\omega_k$. The NEES of the IMM-WNTR contains sharp peaks after maneuvers but recovers quickly after.

The IMM-CT-LH and the IMM-WNTR-CT are designed to handle both slow and sharp turns, and they are both capable of long-term prediction since they estimate the turn rate. The RMSE of IMM-CT-LH reaches very low values at times (see segments E, G, and I) but it contains higher peaks after maneuvers (see segments D, H, and L). Similar arguments can be made about its NEES. The IMM-WNTR-CT, designed to have the capability to track during sharp or slow turns, exhibits low RMSE (almost as good as IMM-WNTR) and it provides $\hat{\omega}_{k|k}$. It can be seen that it does not outperform IMM-CT-LH during straight-line motion, but it has better RMSE and NEES after all maneuvers.

### 2.4.3 Turn Rate Estimate

It is worth noting that IMM-WNTR-CT is capable of estimating the turn rate $\omega_k$. In Fig. 2.7, we see that this is true, although IMM-CT-LH seems to react to the change in $\omega$ faster. This assures that IMM-WNTR-CT can be used for long-term prediction if measurements are delayed.

### 2.4.4 Inspection of Mode Probabilities

In Figs. 2.8, 2.9, and 2.10, we plot the mode probabilities for IMM-CT-S (a conventional IMM-CT), IMM-CT-F (an IMM-CT with zero probability of remaining in the maneuvering mode), and IMM-WNTR, respectively. We show these as examples of how the IMM treats short-duration maneuvers in terms of its maneuver detection capability. The mode probabilities for IMM-CT-S vary considerably compared to the other two because of the specified long mean sojourn time $E[\tau_2]$, which is required for conventional IMM-CT algorithms. For IMM-CT-F, we see that since $p_{22} = 0$ and because of the random walk turn rate of the discrete CT model, the algorithm is never capable of detecting a maneuver. In the IMM-WNTR, which also has zero probability of returning to left- or right-turn modes, we can still see small perturbations in the mode probabilities (Fig. 2.10). These small changes in mode probabilities are enough to open up the filter bandwidth when maneuvers
Figure 2.7: Average $\hat{\omega}_{k|k}$ from 3 filters

Figure 2.8: IMM-CT-S Average mode probabilities
Figure 2.9: IMM-CT-F Average mode probabilities

Figure 2.10: IMM-WNTR Average mode probabilities
are detected.

2.4.5 Computational Complexity

The 3-mode IMM-WNTR and the 4-mode IMM-WNTR-CT are computationally efficient in the sense that the only extra computation needed, when compared to a single-mode filter or a 2-mode IMM, is having to run additional filters through the IMM, and this scales between linearly \(O(n_r)\) and quadratically \(O(n_r^2)\) with the number of additional modes, since each nonlinear filter only runs once per measurement arrival in an IMM, but the mixing step requires interaction between every pair of modes. As \(n_r \to \infty\), \(O(n_r^2)\) will dominate.

As an example, running 1000 Monte-Carlo simulations on a single-threaded MATLAB platform using a 32-bit, 2.9 GHz system with 3 GB of RAM, the average computation times per measurement interval are: 5.6 ms for IMM-CT-S, 3.9 ms for IMM-CT-F, 10.5 ms for IMM-CT-LH, 18.6 ms for IMM-WNTR, and 24.0 ms for IMM-WNTR-CT.

2.5 Conclusions

This paper presents a novel non-zero mean, white noise turn rate dynamic model that can be used within a MM estimator to track ground vehicles, moving in an open field, in two dimensions. Due to their nature, some targets, such as land vehicles with continuous wheel tracks, are capable of performing sharp turn maneuvers as high as \(60^\circ/s\) while maintaining almost constant speed, and this presents a challenge for conventional state estimator designs, including multiple-model methods that were originally designed for targets with relatively slow coordinated-turn capability. The nature of the short-lived maneuvers requires the design of Transition Probability Matrices that have a distinct form compared to those for tracking coordinated-turn targets. Having sharp left- and right-turn modes in an IMM with the TPMs described in this paper is similar to specifying a multi-modal turn rate process noise PDF, and the result is a filter that performs extremely well in terms of position and velocity RMSE at the expense of not being able to perform accurate long-term prediction. When also including a slow-turn CT mode in the IMM-WNTR filter, an estimate of the turn rate is provided and long-term predictions can be more accurate. The proposed filters are computationally inexpensive, and any nonlinear filtering algorithm, as well as any multiple-model estimator, can be used with the dynamic models presented here.
Chapter 3

Tracking Highly-Maneuvering Targets with Destination Information

A method to estimate/track the state of a high-speed target that can be modeled as a feedback control system in 3D space is developed. The target’s lateral acceleration input is not known, but it is known that the target aims to reach a certain destination position at a future time. The trajectory may also involve random evasive maneuvers. If the parameters of the modeled control loop are observable, the proposed model reduces the prediction and estimation errors. If the target’s destination is known, then the accuracy can be increased further. The model, based on Proportional Navigation (PN guidance), is nonlinear and can be used in a multiple-model estimator if the target can also perform random maneuvers along its route to the final destination. This is applied to a 3D maneuvering aerial target state estimation problem with a target capable of high-magnitude, random lateral accelerations under a PN control policy. It is shown that due to the observability of the feedback control parameters, the filter significantly reduces the error of estimated position/velocity and provides a flexible estimation model for laterally-maneuvering point targets in a 3D fluid.

Some authors have considered the problem of track smoothing for trajectories with known start and end points using linear motion models and general Bayesian algorithms [7, 17]. In this dissertation, the specific task of tracking a maneuvering target in 3D space is considered and is placed in the familiar framework of the IMM estimator. Even without direct knowledge of the plant input (lateral acceleration in this case), certain system properties (such as reference setpoints, feedback gains, energy constraints, and/or actuator constraints) can be inferred or estimated to
some degree. This model can provide the filter with a dynamic system description that is more accurate in its prediction than one with only kinematic states.

Autonomous guidance systems are typically under some variant of Proportional Navigation [3], [46], which is designed to effectively engage a moving object with minimal actuation. Even if random maneuvers are performed along the way, the guidance system must eventually resume an engagement trajectory to reach the destination. Previous tracking methods did not consider dynamic models based on feedback control, and no model has shown to be effective in estimating turn maneuvers in a time-varying plane of rotation. However, some work has been done investigating the use of the Proportional Navigation policy as a model to predict the future location of airborne threats [24], but the authors did not make use of this prediction to improve the current state estimation accuracy, prediction covariance was not provided, random maneuvers were not considered, and unknown destinations were not considered; this dissertation develops solutions to these important problems.

The PN dynamic model is similar to the PN guidance model except that the tracker cannot control the actuator of the target, and it does not know the PN control gain. The PN gain, destination, drag acceleration, position, and velocity are simultaneously estimated from spherical measurements. The case of known destination is also considered.

### 3.1 System Model

#### 3.1.1 Continuous-Time Kinematics

The system is modeled as a kinematic point target, moving in a 3D inertial Cartesian space (Earth-fixed, flat Earth)\(^1\) with position vector \(s(k)\). The accelerations of the target are lateral (due to turning) and longitudinal (due to drag), plus gravity. The Coriolis acceleration is ignored since its magnitude is negligible with respect to gravity and lateral acceleration capability, and the target is assumed to have no thrusting capability. To derive a discrete-time model that includes the effect of all the aforementioned accelerations, we start with the target position vector in continuous time as

\[
s(t) = [x(t) \ y(t) \ z(t)]^T
\]

\(^1\)The Flat Earth assumption is not valid in a real-world, long-distance scenario. The assumption is made for simplicity and elucidation. See [13] for coordinate transformations.
and form the sum of the continuous-time accelerations — the longitudinal $a_d(t)$ (from drag), the lateral $a_\Omega(t)$, and gravity $a_g$. The total acceleration is

$$\ddot{s}(t) = a_\Omega(t) + a_d(t) + a_g$$

(3.2)

The gravitational acceleration vector is

$$a_g = [0 0 -g]'$$

(3.3)

The magnitude of the gravitational acceleration, $g$, is taken as constant. The velocity dependence is incorporated into the total acceleration as (see [33])

$$\ddot{s}(t) = \Omega(t) \times \dot{s}(t) - a_d(t) \frac{\dot{s}(t)}{S(t)} + a_g$$

$$= \left[ K_\Omega(t) - \frac{a_d(t)}{S(t)} I_{3\times3} \right] \dot{s}(t) + a_g$$

(3.4)

where $a_d(t)$ is the magnitude of drag acceleration (simply referred to as “drag” from here on; its dependence on various factors is given in the sequel), $\Omega(t)$ is the angular velocity of the target’s velocity vector, $\times$ denotes the vector cross-product, $\dot{s}(t)$ is the target velocity vector, $S(t) = ||\dot{s}(t)||$ is the target speed, and $I_{n\times n}$ is the $n \times n$ identity matrix. $K_\Omega(t)$ is the cross product skew-symmetric pre-multiply matrix corresponding to $\Omega(t)$, given by

$$K_\Omega(t) = \begin{bmatrix}
0 & -\Omega_z(t) & \Omega_y(t) \\
\Omega_z(t) & 0 & -\Omega_x(t) \\
-\Omega_y(t) & \Omega_x(t) & 0
\end{bmatrix}$$

(3.5)

From (3.4), the continuous-time evolution of position and velocity (PV) in matrix form is

$$\dot{x}^{PV}(t) = A^{PV}(t)x^{PV}(t) + \begin{bmatrix} 0_{1\times3} \\ a_g \end{bmatrix}$$

(3.6)

with

$$x^{PV}(t) = [s(t)']' \dot{s}(t)']'$$

(3.7)

$$A^{PV}(t) = \begin{bmatrix}
0_{3\times3} & I_{3\times3} \\
0_{3\times3} & K_\Omega(t) - \frac{a_d(t)}{S(t)} I_{3\times3}
\end{bmatrix}$$

(3.8)

The drag, $a_d(t)$, can be written in terms of the drag coefficient $c_d(t)$ as

$$a_d(t) = \frac{c_d(t) p[z(t)] A S(t)^2}{2m}$$

(3.9)
with \( m \) the mass of the target vehicle (assumed constant), \( \rho[z(t)] \) the air mass density as a function of altitude \( z(t) \), and \( A \) the target’s aerodynamic cross-section area. The base coefficient of drag is designated \( c_{d0} \) and the total drag is the sum of the base drag plus the lift-induced drag \( c_i(t) \) [20], which results in increased drag when the target performs turns, as

\[
c_d(t) = c_{d0} + c_i(t) \quad (3.10)
\]

With \( \epsilon \) the efficiency factor and \( R \) the wing aspect ratio, the lift-induced drag is given as

\[
c_i(t) = \frac{[2a\Omega(t)]^2}{m\rho(t)AS(t)^2\pi\epsilon R} \quad (3.11)
\]

The air density is modeled as exponentially decreasing with altitude as

\[
\rho[z(t)] = \rho_0 \exp\left[-\frac{z(t)}{Z}\right] \quad (3.12)
\]

with given atmospheric constants \( \rho_0 \) and \( Z \) [48].

In the estimator, \( c_{d0}, \epsilon, R, m, \) and \( A \) are not known, and the altitude \( z(t) \) is a stochastic process (and part of the state). To describe the evolution of the drag as a state variable in continuous time, the derivative of (3.9) with respect to time is needed. First, let

\[
c_d^* = \frac{c_dA}{2m} \quad (3.13)
\]

Then,

\[
\frac{da_d(t)}{dt} = c_d^* \left[ \frac{dS(t)^2}{dt} \rho[z(t)] + S(t) \frac{d\rho[z(t)]}{dt} \right] \quad (3.14)
\]

Expanding the derivative in the first term,

\[
\frac{dS(t)^2}{dt} = \frac{d}{dt} \left[ \dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2 \right] = 2 \left[ \ddot{x}(t)\dot{x}(t) + \ddot{y}(t)\dot{y}(t) + \ddot{z}(t)\dot{z}(t) \right] = 2\dot{s}'(t)\dot{s}(t) \quad (3.15)
\]

The term \( \dot{s}'(t)\dot{s}(t) \) is the inner product of velocity and acceleration. Substituting (3.4) for the acceleration \( \ddot{s}(t) \) and using the fact that the inner product of the orthogonal vectors is zero (velocity is orthogonal to lateral acceleration),

\[
\frac{dS(t)^2}{dt} = 2\dot{s}'(t) \left[ \Omega(t) \times \dot{s}(t) - a_d(t)\frac{\dot{s}(t)}{S(t)} + a_g \right] = -2S(t)a_d(t) - 2g\dot{z}(t) \quad (3.16)
\]
Using (3.9) and (3.13), the first term of (3.14) becomes
\[ c_d \frac{dS(t)^2}{dt} \rho[z(t)] = -c_d^2S(t)a_d(t)\rho[z(t)] - c_d^22g\dot{z}(t)\rho[z(t)] \]
\[ = -2a_d(t) \frac{a_d(t)}{S(t)} - 2g\dot{z}(t)\frac{a_d(t)}{S(t)^2} \]  
(3.17)

The second term of (3.14) becomes
\[ c_d S(t)^2 \frac{d\rho[z(t)]}{dt} \]
\[ = \left[ \dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2 \right] \frac{d}{dt} \left\{ \rho_0 \exp \left[ -\frac{z(t)}{Z} \right] \right\} \]
\[ = -c_d^2S(t)^2 \frac{\dot{z}(t)}{Z} \rho_0 \exp \left[ -\frac{z(t)}{Z} \right] \]
\[ = -c_d^2S(t)^2 \frac{\dot{z}(t)}{Z} \rho[z(t)] \]
\[ = -\frac{\dot{z}(t)}{Z} a_d(t) \]  
(3.18)

Finally, (3.14) can be evaluated as
\[ \frac{da_d(t)}{dt} = -2a_d(t) \frac{a_d(t)}{S(t)} - 2g\dot{z}(t)\frac{a_d(t)}{S(t)^2} - \frac{\dot{z}(t)}{Z} a_d(t) \]
\[ = -D(t)a_d(t) \]  
(3.19)

with
\[ D(t) = 2a_d(t) \frac{a_d(t)}{S(t)} + \left( \frac{2g}{S(t)^2} + \frac{1}{Z} \right) \dot{z}(t) \]  
(3.20)

which depends on the current drag, speed, and vertical velocity. The dependence on the Mach number and the increase in drag due to turn acceleration (i.e. lift-induced drag) can also be modeled, but they are beyond the scope of this paper, so in the next section, the remaining drag uncertainty will be modeled by describing \( a_d(t) \) as a non-linear Markov process.

### 3.1.2 Discrete-Time Implementation

Over a measurement interval \( T \), \( a_d(t) \) and \( \Omega(t) \) will be approximated as constants. Using the matrix exponential and the discrete time step \( k = 0, 1, \ldots \), the discrete-time evolution with additive white noise acceleration \( v_a(k) \) can be written as
\[ x^{PV}(k+1) \approx P^{PV}(k)x^{PV}(k) + \begin{bmatrix} \frac{T^2}{2}I_{3\times3} \\ TI_{3\times3} \end{bmatrix} v_a(k) + \begin{bmatrix} \frac{T^2}{2}a_g \\ Ta_g \end{bmatrix} \]  
(3.21)
with the matrix exponential

\[ F^{PV}[x(k)] = \exp \left( T A^{PV}[x(k)] \right) \quad (3.22) \]

to be evaluated at the latest state estimate. The white noise acceleration \( v_a(k) \) has covariance

\[ E[v_a(k)v_a(k)'] = \sigma_a^2 I_{3 \times 3} \quad (3.23) \]

and

\[ A^{PV}[x(k)] = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & K\Omega(k) - \frac{a_d(k)}{s_f(k)} I_{3 \times 3} \end{bmatrix} \quad (3.24) \]

is used as an approximation to (3.8) evaluated the most recent estimates of drag, speed, and angular velocity of the target’s velocity vector. To fully implement (3.21), the evolution of \( \Omega(k) \) must be specified. This will done in Sec. 3.1.3 as part of the Proportional Navigation estimation model.

Similarly, (3.19) is discretized in time as

\[ a_d(k + 1) = \exp \left[ -TD(k) \right] a_d(k) + Tv_d(k) \quad (3.25) \]

with \( D(k) \) approximated in (3.25) using the most recent estimates of drag, speed, and vertical velocity. The additive white noise jerk \( v_d(k) \) covers un-modeled changes in drag and its variance is

\[ E[v_d(k)^2] = \sigma_d^2 \quad (3.26) \]

The noise variances in (3.23), (3.26) are filter design parameters [11].

### 3.1.3 Proportional Navigation Estimation Model

In this application, the dynamic model for the proposed filter corresponds to a point target under an implementation of Proportional Navigation (PN) [46]. This model will provide, as shown later, an expression for \( \Omega(k) \) to be used in (3.21). A simple 2D example is shown in Fig. 3.1. The 14-dimensional state vector consists of stacked position, velocity, drag \( a_d(k) \), PN gain \( N(k) \), previous position \( s(k-1) \), and destination \( s_f(k) \)

\[ x(k) = [s(k)', s(k)', a_d(k), N(k), s(k-1)', s_f(k)']' \quad (3.27) \]

The PN model is based on discrete-time Pure Proportional Navigation (PPN), a common guidance control implementation that maintains orthogonality between the applied acceleration and the velocity vector.
Figure 3.1: 2D example of PN principle. If the LOS angular velocity is $\omega_{LOS}$, then the target turns its velocity vector at a rate $N\omega_{LOS}$ (by applying lateral accelerations) to attempt to regulate $\omega_{LOS}$ to zero.

[24] [46]. The position of the destination used in the PPN feedback control is $s_f(k)$ and the line of sight (LOS) from the target to the destination is

$$r(k) = s_f(k) - s(k) \quad (3.28)$$

The PN feedback control policy attempts to reduce the angular velocity of the LOS to zero by applying an angular velocity to the velocity vector that is proportional to the LOS angular velocity. The direction of this resulting angular velocity vector is determined by the unit vector

$$u(k) = \frac{\Omega(k)}{||\Omega(k)||} = \frac{r(k - 1) \times r(k)}{||r(k - 1)|| \cdot ||r(k)||} \quad (3.29)$$

and the magnitude (the turn rate) is

$$\omega(k) = ||\Omega(k)|| = \frac{N(k)}{T} \cos^{-1} \left( \frac{r(k - 1) \cdot r(k)}{||r(k - 1)|| \cdot ||r(k)||} \right) \quad (3.30)$$

resulting in

$$\Omega(k) = \omega(k)u(k) \quad (3.31)$$

$N(k)$ is the PN gain parameter to be estimated and

$$r(k - 1) = s_f(k) - s(k - 1) \quad (3.32)$$

requiring $N(k)$ and $s(k - 1)$ to be part of the state vector $x(k)$. The PN gain $N(k)$ will be modeled as a random walk to allow for an adaptive mean, given as²

$$N(k + 1) = N(k) + T \nu_N(k) \quad (3.33)$$

²For example, actuator saturation limits the lateral acceleration magnitude $S(k)\omega(k)$, which, in the estimation model, can be adapted to by a decrease in the mean of $N$. This bypasses the need for the estimator to know the actuator saturation limits explicitly and also favors a linear model for the evolution of this stochastic parameter.
with the variance of the zero-mean, white process noise $v_N(k)$ modeling the PN gain changes

$$E[v_N(k)^2] = \sigma_N^2$$  \hfill (3.34)

In modeling the evolution of the destination $s_f(k)$, we either treat it as a non-zero mean, white noise process when the destination is known to the estimator, or as a random walk when it must be estimated as a random parameter. As a white noise process, it is

$$s_f(k + 1) = \bar{s}_f + Tv_f(k)$$  \hfill (3.35)

and as a random walk, it is

$$s_f(k + 1) = s_f(k) + Tv_f(k)$$  \hfill (3.36)

The covariance of the zero-mean, white process noise $v_f(k)$ is

$$E[v_f(k)v_f(k)'] = \text{diag} \left[ (c_x ||r(k)||)^2, (c_y ||r(k)||)^2, (c_z ||r(k)||)^2 \right]$$  \hfill (3.37)

This covariance is time-varying and depends on the LOS distance $||r(k)||$. The scaling design parameters $c_x, c_y, c_z$ have dimension of inverse time. They model uncertainty of the target’s own destination sensing uncertainty, which decreases as the target approaches its destination since the target is probably using a noisy on-board sensor to measure the angle between the LOS and the velocity vectors.

The white noise version (3.35) will be referred to as a “destination-aware” PN (PN-DA) model and the random walk version (3.36) will be referred to as a “destination-unaware” (PN-DU) model. Both models estimate the PN gain $N(k)$ in the control system.

Note that the orthogonality constraint $\Omega \perp \dot{s}$ is not explicitly preserved in (3.29). In practical applications, this constraint typically exists since lateral acceleration caused by aileron deflection should be normal to velocity and should not affect the speed as long as it is moderate. If $\Omega \perp \dot{s}$ is maintained, the control law is termed Pure Proportional Navigation (PPN) [24]. This constraint can be enforced by either subtracting from the angular velocity vector $\Omega(k)$ its projection onto the velocity vector $\dot{s}(k)$, or by performing a repeated cross product operation on the turn axis unit vector $u(k)$ and normalizing the result as done in this paper:

$$u_{PPN}^\perp(k) = \frac{\dot{s}(k) \times u(k) \times \dot{s}(k)}{||\dot{s}(k) \times u(k) \times \dot{s}(k)||}$$  \hfill (3.38)

The destination process noise models (3.35) and (3.36) do not imply that the target sees its destination as a moving one. Proportional Navigation is capable of engaging accelerating destinations if the previous LOS $r(k - 1)$ is part of the state, but in the estimation model presented
here, only the previous target position $s(k-1)$ is part of the state. This is because it is assumed that the destinations, although they are random, are stationary points in the inertial reference frame. If they were treated as moving points, the noise due to estimation error would impart very high-magnitude turn rates that do not represent true dynamics at all. It is the destination position that is not known directly, and it will occasionally change in the simulated trajectory according to a Monte Carlo realization. Therefore, since destinations are assumed to be stationary with respect to the ground (albeit unknown), only the previous target position is part of the state vector, not the previous LOS. A natural extension to the present method is to actually use the previous LOS as a state variable to track accelerating objects since PN is capable of this function as well.

The dynamic state evolution can be fully described as:

$$x(k+1) = F[x(k)] x(k) + \Gamma v(k) + \begin{bmatrix} \frac{T^2}{2} a_g \\ T a_g \\ 0_{8 \times 1} \end{bmatrix}$$  \hspace{1cm} (3.39)$$

The complete process noise vector $v(k)$ is

$$v(k) = \begin{bmatrix} v'_a(k) \\ v_d(k) \\ v_N(k) \\ v'_f(k) \end{bmatrix}$$  \hspace{1cm} (3.42)$$

and its covariance is
\[
Q(k) = \begin{bmatrix}
\sigma^2_a I_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 3} \\
0_{1 \times 3} & \sigma^2_d & 0 & 0_{1 \times 3} \\
0_{1 \times 3} & 0 & \sigma^2_N & 0_{1 \times 3} \\
0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & E[v_f(k) v_f(k)']
\end{bmatrix}
\]  

(3.43)

To implement the non-zero mean, white noise model for the destination, the initial condition of the destination estimate \(\hat{s}_f(k-1|k-1)\) supplied to the filter at every time step is equal to a known destination \(\bar{s}_f\)

\[
\hat{s}_f(k-1|k-1) = \bar{s}_f
\]

(3.44)

with a modified initial condition covariance, using only the first 11 rows and columns of the unmodified covariance, as

\[
\bar{P}(k-1|k-1) = \text{diag} \left[ P_{11}^{11}(k-1|k-1), E[v_f(k) v_f(k)'] \right] \]

(3.45)

with \(E[v_f(k) v_f(k)']\) given in (3.37). Similarly, the process noise \(v(k)\) defined in (3.42) is truncated to remove the destination process noise \(v_f(k)\), the last 3 columns of (3.41) are removed, and the last 3 rows and columns of (3.43) are removed. The reason for implementing the white noise destination this way, as opposed to truncating (3.27) and (3.40), is to maintain the same dimension of the state vector as the random walk destination model \((n = 14)\) for mixing in the IMM. The mode with the known destination (i.e. the mode with the non-zero mean, white noise destination model) serves to bias the overall destination estimate toward the known destination.

Different practical motion models, other than 3D PPN, can be envisioned to actually be implemented in the remote target. For example, 2D PN can be implemented in the horizontal plane, while altitude remains nearly constant due to an altitude control system. However, such motion can also be handled by the PPN model since constant-altitude motion is achieved by a target whose velocity vector is horizontal, approaching a destination that is at the same altitude as the target. It is difficult, however, to envision a point target, under the acceleration of gravity, whose motion model ensures that it reaches its final destination, if this model is not in some way related to PN. In any case, uncertainties as to the dynamic mode in effect at time step \(k\) can be dealt with effectively by the use of multiple-model estimators such as the IMM, as is described in Sec. 3.2.
3.1.4 Discrete-Time Measurement Equation

Choosing the observation model is not a focus of this paper, but 3 or more dimensions should be available for measurement to ensure reasonable observability of the parameters [11]. For the development of a sufficient measurement model to accompany the process models in a simulated scenario, the noisy measurement vector is provided in spherical coordinates of range, azimuth angle, and elevation angle, which is a nonlinear function of the state vector (for simplicity, the sensor assumed to be at the origin of the coordinate system)

\[
\mathbf{z}(k) = \begin{bmatrix}
    r(k) \\
    \theta(k) \\
    \phi(k)
\end{bmatrix}
= \begin{bmatrix}
    \sqrt{x(k)^2 + y(k)^2 + z(k)^2} \\
    \tan^{-1} \frac{y(k)}{x(k)} \\
    \sin^{-1} \frac{z(k)}{r(k)}
\end{bmatrix} + \mathbf{w}(k)
\]

where the discrete-time measurement noise \( \mathbf{w}(k) \) is a zero-mean, white noise vector, uncorrelated from the current state and the process noise, and has covariance matrix \( \mathbf{R} \).

3.2 Multiple Model Method

3.2.1 Motivation

A single motion model is not adequate to describe a target that switches between different dynamic modes. Generally, if a target “maneuvers” in a classic sense, then it deviates from some predicted motion. In a 3D kinematic target problem, classic estimators may consider any accelerations (except gravity), or changes in acceleration, to be maneuvers [11]. In the PPN model derived here, a target on course to reach a known destination can be considered to be in a non-maneuvering mode, since the time-varying acceleration is now approximately deterministic. However, if the target is attempting to confuse a remote estimator, then it would no longer appear to be heading towards its final destination, but a PPN model may still be adequate in the estimator as a secondary maneuvering mode because of its ability to describe lateral accelerations in any plane of rotation by varying the fake final destination and feedback gain. To this end, the IMM estimator is used here to automatically balance the estimate between a mode \( M_1 \) with small values for \( c_x, c_y, c_z \) (considered a
low process noise, non-maneuvering mode since it is consistently headed for the same destination),
and a model \( M_2 \) with large values for \( c_x, c_y, c_z \) (considered a high process noise maneuvering mode
since the destination is modeled as stochastic) [11]. If there is prior knowledge of the target’s
final destination as a mean \( \bar{s}_f \), then \( M_1 \) should be a “small” additive white noise version of the
destination state evolution (3.35) and \( M_2 \) should be the random walk evolution (3.36); but, if this
information is not available, then two random walk modes with different levels of process noise
should be implemented as \( M_1 \) and \( M_2 \).

### 3.2.2 Mode Transition Probabilities of the IMM Estimator

The transition probability matrix (TPM) of the IMM is

\[
\Pi = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}
\]  

(3.47)

where \( p_{11} \) is the probability of staying in \( M_1 \), \( p_{12} \) is the probability of transitioning from \( M_1 \) to \( M_2 \),
and so on. We determine these probabilities by using the mean sojourn time (in sampling intervals)
of each mode \( E[\tau^i], i = 1, 2 \) [11] as

\[
\Pi = \begin{bmatrix} 1 - 1/E[\tau^1] & 1/E[\tau^1] \\ 1/E[\tau^2] & 1 - 1/E[\tau^2] \end{bmatrix}
\]  

(3.48)

If the target is close to its destination, it is assumed that it cannot afford to make any more
evasive maneuvers and will probably engage its destination directly. Therefore, when \( ||r(k)|| < r_1 \)
(where \( r_1 \) is a design parameter), \( M_1 \) can be known to be in effect with probability of 1. We can
do this in (3.48) by setting \( E[\tau^1] \to \infty \) and \( E[\tau^2] = 0 \).

### 3.3 Simulation and Results

Monte Carlo (MC) simulations, using 1000 trials in two separate scenarios, are used to establish the
value of the proposed estimators. A thorough description of the ground truth trajectory generation,
the measurement parameters, the estimator parameters, results, and discussions are given in the
following sections.

#### 3.3.1 UKF for Nonlinear Prediction and Measurement

The Unscented Kalman filter, as described in [29], is used for the non-linear PN prediction model of
Sec. 3.1.3 and the non-linear measurement model of Sec. 3.1.4. The UKF is a numerical solution to
the recursive non-linear system estimation problem that uses so-called “sigma points” to approximate the state probability density, propagate the points through the nonlinear state transition and measurement, and approximate the first two moments of the updated probability density. It is capable of accuracy comparable to the Second-Order Extended Kalman Filter but without the need to calculate Jacobian and Hessian matrices [6], [45]. Other nonlinear estimation methods are applicable as well, such as particle filters, but since the UKF is computationally more efficient, it is adopted here. The UKF weight parameter is selected as $W_0 = 0.5$.

### 3.3.2 Measurement Parameters

The measurements are provided as spherical coordinates of range $r(k)$, azimuth angle $\theta(k)$, and elevation angle $\phi(k)$. The covariance of the additive, white noise vector $w(k)$ is

$$
R = \text{diag} \left[ \sigma_r^2, \sigma_\theta^2, \sigma_\phi^2 \right] \\
= \text{diag} \left[ (10 \text{ m})^2, (0.001 \text{ rad})^2, (0.001 \text{ rad})^2 \right] 
$$

### 3.3.3 Ground Truth

The sensor’s position is fixed at the origin in the inertial frame. The target is headed for a final destination located at $\bar{s}_f = [-4 15 0]' \text{ km}$. It starts at initial position $s_0 = [450 18 30]' \text{ km}$. The initial velocity vector is $\dot{s}_0 = [-5.97 0 -0.59]' \text{ km/s}$ (speed $S = 6.0 \text{ km/s}$). The ground truth integration time step is $T_i = 10 \text{ ms}$ and the measurement time interval is $T = 1 \text{ s}$. Two different trajectory descriptions are simulated and different variations of the estimators are implemented for each scenario in the next sections. In all scenarios, the true PN gain is set to $N = 6$ (this is initially unknown to the filters). The drag in the true trajectory will be simulated using the direct definitions of drag as in (3.10) with $m = 2000 \text{ kg}$, $c_{d0} = 0.03$, $A = 0.48 \text{ m}^2$, $R = 1.5$, $\epsilon = 0.95$, $\rho_0 = 1.225 \text{ kg/m}^3$, and $Z = 1.2124 \cdot 10^{-6} \text{ m}$. Therefore, true drag will depend on the lateral acceleration magnitude, as well as the altitude and speed.

Even though the target will reach its destination in the Monte Carlo runs, the actual trajectories will be generated randomly and independently for each run. The target has decreasing finite kinetic energy, so some rules are implemented to ensure that the target reaches the final destination, yet still performs maneuvers. This allows for a robust performance analysis. These rules are as follows:

1. The target’s PPN simulation guides it to a destination close to the final destination, $s_f(k) =$
\( \bar{s}_f + v_f \) in one of two randomly selected dynamic modes \( M_1 \) or \( M_2 \).

2. Modes \( M_1 \) or \( M_2 \) are selected with equal probability and have different probability densities for their respective \( v_f \). If the trajectory is to be direct to the final destination with no other maneuvers, \( M_1 \) is selected with probability of 1.

3. In mode \( M_1 \), the deviation \( v_f \) from the final destination \( \bar{s}_f \) is a zero-mean, white random vector as in (3.37) with \( c_x = c_y = c_z = 10^{-4} \).

4. In mode \( M_2 \), the deviation \( v_f \) is a uniform random vector. The uniform intervals in the \( x \), \( y \), and \( z \) coordinates are described, respectively, by \([\bar{s}_{fx} - 0.03||r(k)||, \bar{s}_{fx} + 0.03||r(k)||]\), \([\bar{s}_{fy} - 0.03||r(k)||, \bar{s}_{fy} + 0.03||r(k)||]\), and \([0, \bar{s}_{fz} + 0.03||r(k)||]\), respectively. The uniform distribution is chosen because its finite support can prevent a destination with a \( z \) coordinate less than zero from being selected (since the final destination has a zero \( z \) coordinate).

5. The sojourn time of \( M_1 \) is selected from an independent exponential distribution with mean of 5 s.

6. The sojourn time of \( M_2 \) is selected from an independent exponential distribution with mean of 3 s.

7. The discrete-time equations (3.21), (3.22), (3.24) are implemented as a numerical integration of (3.4) with \( T_i = 0.01 \) s and \( \sigma_a^2 = 0 \). (3.29), (3.30), and (3.31) establish the angular velocity vector, while (3.9), (3.10), and (3.11) establish the drag, both at every time step.

8. The maximum lateral acceleration \( a_d \) generated as a result of (3.30) is limited to 25g to simulate actuator saturation and to verify robustness to this common non-linearity. The maximum allowable lateral jerk is 5g/s. The PN filter models both the PN gain \( N(k) \) and the destination \( s_f(k) \) as random walks to allow for adapting to this change.

9. When the distance of the LOS is less than 5 km, \( M_1 \) is selected with probability 1.

10. If a randomly-generated trajectory results in a target position below \( z = 0 \) or speed less than 1 km/s (i.e. if the target misses the destination), then the realization of the trajectory is discarded and it is replaced by a new realization. This may occur due to the random generation of aggressive maneuvers that render the destination unreachable.
The average lateral acceleration from Monte Carlo runs, when the trajectory contains random maneuvers, is about $3g$ and occasionally reaches a saturation maximum of $25g$. The average simulation time (considered completed when the target reaches its destination) is about 100 s. Figs. 3.2–3.4 show three projections of the same randomly-generated, maneuvering trajectory realization (Scenario 2) along with measurements and trajectory estimation according to the IMM-DA filter (to be described in the next section).

### 3.3.4 Filter Descriptions

Three filters are compared in two scenarios. Scenario 1 is designed to test the single-mode filters and the observability of the state by using a PN ground truth (without extra maneuvers, i.e. a direct trajectory). Scenario 2 tests the filters in a trajectory with extra maneuvers using multiple-model filters (generated using the rules given in Sec. 3.3.3). The filters are:

(i) PN-DA (destination-aware) UKF/IMM (see (3.35)), labeled UKF-DA in Scenario 1 and IMM-DA in Scenario 2

(ii) PN-DU (destination-unaware) UKF/IMM (see (3.36)), labeled UKF-DU in Scenario 1 and IMM-DU in Scenario 2
Figure 3.3: A realization of a randomly-generated true trajectory (Scenario 2), measurements, and estimated trajectory from IMM-DA filter, as viewed in the $x$–$z$ plane.

Figure 3.4: A realization of a randomly-generated true trajectory (Scenario 2), measurements, and estimated trajectory from IMM-DA filter, as viewed in the $y$–$z$ plane.
(iii) Baseline Kalman/IMM filters using the linear Wiener Process Acceleration (WPA) model [11], labeled KF-WPA in Scenario 1 and IMM-WPA in Scenario 2 (the process noise is selected based on the assumed lateral acceleration capability).

For all filters to be implemented, we set $\sigma_a = 0.1g$, $E[\tau_1] = 5s$, $E[\tau_2] = 3s$, $\sigma_N = 0.1$, and $\sigma_d = 10^{-5} \text{m/s}^3$. The parameters $c_x$, $c_y$, and $c_z$ are specialized for each filter.

The initial 6-dimensional position and velocity estimate $\hat{x}^{\text{PV}}_0$, the $6 \times 6$ position–velocity covariance $P^{\text{PV}}_0$, and the marginal $3 \times 3$ initial position covariance $P^{\text{P}}_0$ are initialized using two-point differencing as explained in [11]. All filters are initialized with $\hat{N}_0 = 4$, $\hat{a}_{d0} = 4 \text{m/s}^2$, and $\hat{s}_{f0} = [0\ 0\ 0]'$, which are values that differ from those in the true trajectory. This will allow for a fair evaluation of the observability of these stochastic parameters. With $l \triangleq (100 \text{km})^2$, the initial covariance is

$$P_0 = \begin{bmatrix}
P^{\text{PV}}_0 & 0_{6 \times 1} & 0_{6 \times 1} & 0_{6 \times 3} & 0_{6 \times 3} \\
0_{1 \times 6} & \left(4 \text{m/s}^2\right)^2 & 0_{1 \times 3} & 0_{1 \times 3} \\
0_{1 \times 6} & 0_{2 \times 1} & 0_{1 \times 3} & 0_{1 \times 3} \\
0_{3 \times 6} & 0_{3 \times 1} & 0_{3 \times 1} & P^{\text{P}}_0 & 0_{3 \times 3} \\
0_{3 \times 6} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 3} & I_{3 \times 3}
\end{bmatrix}$$

(3.50)

### 3.3.5 Scenario 1 Parameters and Results

The proposed estimator is first tested on a trajectory that does not significantly deviate the target from a direct path towards its destination, although some white noise is added to the destination mean at every time step in order to simulate the target’s own uncertainty. This is a direct trajectory and is realized by selecting mode $M_1$ with probability 1 (see rule 2 above). The maximum lateral acceleration in this mode is approximately $1.6g$ with an average of $1.36g$. The variance (3.37) of $v_f(k)$ is described for both the PN-DA mode and the PN-DU mode by $c_x = c_y = c_z = 10^{-3}$. For the baseline WPA Kalman filter, the white noise acceleration increment process noise standard deviation is selected to be $\sigma_a = 2g$.

Performance comparisons begin with the position root-mean-square error (RMSE) shown in Fig. 3.5. Since it is difficult to assess the performance near the end of the simulation from this plot, the data is presented in Fig. 3.6 as a ratio of the mean-square error (MSE) of each filter to the MSE of the raw measurements, in dB ($10 \log(\text{MSE}_{\text{est}}/\text{MSE}_{\text{meas}})$) — a filter that does not improve on its measurements is woeful. The velocity RMSE is shown in Fig. 3.7. Both the PN-DA and PN-DU filters outperform the baseline filter, especially when considering the velocity RMSE (this implies better position prediction accuracy).
Figure 3.5: Position RMSE of raw measurement and filter estimates plotted versus time (Scenario 1).

Figure 3.6: Position MSE ratio in dB plotted versus time (Scenario 1).
Next, consistency in the position and velocity estimates is investigated. To accomplish this, the averaged Normalized Estimation Error Squared (NEES) for position and velocity [11] is plotted in Fig. 3.8. The ideal value for the NEES in this case is 6 since it is calculated for the position and velocity only. The DA filter does seem to be a bit pessimistic in its covariance calculation near the end of the simulation.

It is very interesting to note the low estimation error of the PN-DU filter since it attempts to estimate the final destination, the PN gain, and the drag coefficient (the DA model does know the final destination as $\bar{s}_f$). This is encouraging since we intend to utilize the DU mode in the next section (Scenario 2) as part of an IMM to detect abrupt changes in the target’s current destination in a randomly-maneuvering target trajectory. The actual errors in the estimates of $N(k)$, $a_d(k)$, and $s_f(k)$ are not as interesting here as their observability and consistency is. To this end, the average NEES of $\hat{a}_d(k|k)$, $\hat{N}(k|k)$, and $\hat{s}_f(k|k)$ are plotted in Fig. 3.9. It can be seen that the NEES is reasonable for all three estimates, which indicates observability of these stochastic parameters.

Fig. 3.10 shows the actual and estimated drag for the two PN-based filters from a single MC run. The UKF-DU and UKF-DA filters provide an accurate estimate of the drag.
Figure 3.8: NEES of position and velocity estimates plotted versus time (Scenario 1). Values near 6 are ideal.

Figure 3.9: NEES of estimated drag, $\hat{a}_d(k|k)$ (ideal value of 1), PN gain $\hat{N}(k|k)$ (ideal value of 1), and destination $\hat{s}_f(k|k)$ (ideal value of 6) from DU model filter (Scenario 1)
3.3.6 Scenario 2 Parameters and Results

This trajectory is randomly generated according to Sec. 3.3.3, except that once $||\mathbf{r}(k)||$ has decreased to 100 km, the final destination is temporarily switched to a position above the final destination, located at $\mathbf{s}_f^* = \mathbf{s}_f + [1000 - 200 5000]'$. Then, when $||\mathbf{r}(k)||$ has decreased to 20 km, the target’s original final destination is restored and the target continues attempting to reach it. This simulates a very unexpected terminal maneuver. Note that each of the 1000 Monte Carlo runs has a different realization of the ground truth trajectory.

The three IMMs of this section are implemented as in Sec. 3.2 with $r_{\text{max}} = 5\text{ km}$. The first, labeled IMM-DA, has a DA model filter for $M_1$ and a DU model filter for $M_2$. The second IMM, labeled IMM-DU, has a DU model filter for $M_1$ and another DU model filter for $M_2$. The third IMM is a baseline for comparison, labeled IMM-WPA, with a low process noise WPA mode ($M_1$) and a high process noise WPA mode ($M_2$). This is appropriate because the PN control policy always applies a non-zero acceleration so that the target can reach its destination, but sometimes increases this acceleration to perform random maneuvers.

For $M_1$ of both IMM-DA and IMM-DU, we set $c_x = c_y = c_z = 10^{-4}$. For $M_2$ of IMM-DA and IMM-DU, $c_x = c_y = c_z = 0.033$. For $M_1$ of IMM-WPA, $\sigma_a = 0.1g$ and for $M_2$ of IMM-WPA, $\sigma_a = 5g$. 

Figure 3.10: Actual drag $a_d(t)$ and estimated drag $\hat{a}_d(k)$ in a single run (Scenario 1).
Plots of the position MSE Ratio, velocity RMSE, and position/velocity NEES are shown in Figs. 3.11, 3.12, and 3.13, respectively. IMM-DA and IMM-DU perform very well, as expected. Note, however, that the terminal maneuver (which occurs at the same distance away from the destination in every run, approximately between $t = 70\text{ s}$ and $t = 90\text{ s}$) does cause a loss of accuracy for all filters. During most of this maneuvering interval, the target’s lateral acceleration is saturated at $25g$. The IMM-DU is actually a better estimator during this maneuver since it is not biased toward the final destination as IMM-DA is, but after the final destination is restored, the only estimator that is effective is IMM-DA, but this is only with a few seconds left before the trajectories end around $t = 100\text{ s}$.

### 3.4 Conclusions

A new model for a maneuvering point target having an unknown input, high maneuverability, and a destination position has been developed, for the purpose of remotely tracking its kinematic state. While the states of such nonlinear systems are typically hard to estimate, it is known that the high lateral acceleration maneuverability is being exploited in a feedback control loop as a high gain to eventually reach the destination. It is shown here that it is valuable to estimate the parameters of a feedback control loop, such as feedback gain and/or destination position. The hypothesis is that
Figure 3.12: Position RMSE of raw measurement and filter estimates plotted versus time (Scenario 2).

Figure 3.13: NEES of position and velocity estimates plotted versus time (Scenario 2). Values near 6 are ideal.
a point target, traveling in 3D space toward a destination position, is under a Proportional Navigation control policy with a gain parameter $N$, and/or that Proportional Navigation is an adequate prediction model to encompass turning maneuvers in any plane of rotation, with aerodynamic drag also estimated as a state. When known, the destination position information provides the estimation system with a more accurate state prediction. The destination is observable when it is modeled as a random parameter. The proportional feedback gain parameter and drag are also observable and are estimated in real-time. It was shown that a 2-mode IMM, consisting of a low process noise PN model with non-zero mean, white noise destination (mode 1) and a high process noise PN model with random walk destination, results in a very accurate filter for systems exhibiting this feedback-controlled, yet stochastic, behavior. The control- and destination-aware estimation system significantly decreases the prediction error and improves the overall state estimation accuracy.
Chapter 4

Data Fusion using IMM Inside Information
4.1 List of Symbols and Acronyms Used In This Chapter

(·) Mean of (·)

(·) Mixed initial condition of (·) (IMM algorithm)

(·) Estimate of (·)

(·) Error of (·)

d_{jm}^{m}(k) The Gaussian-approximated process noise entering the log-odds

D_{jm}^{m}(k) The covariance of d_{jm}(k)

F The state transition matrix

g^{m}(k) The additive noise of the linearized joint IMM system

G^{m}(k) The covariance of g(k)

H_{jm} The measurement matrix at LT j

j The LT index (j = 1, 2 — used as a subscript)

J^{m}(k) The system (error and log-odds) transition Jacobian matrix

k The discrete time step

Log-R Log-ratio of a probability pair

m The target mode index of the received estimates and probabilities (m = 1, 2 — used as a superscript)

MCE(E) Local mode-conditioned estimate (error)

MCI Local mode-conditioned innovation (residual)

MCP(E) Local mode-conditioned prediction (error)

M The true target dynamic mode

µ_{jm}^{m}(k) Locally-computed probability of mode m

µ_{jm}^{m}(k) µ_{jm}^{m}(k)

µ_{jm}^{m}(k) Initial condition mixing weight (IMM algorithm)

n The target mode index hypothesis under consideration at the FC (n = 1, 2)

(·)^{m}n Any variable (·) derived under the hypothesis that the current mode is n

µ_{jm}^{m}(k) MCI

N^{m}n Spread-of-the-means term of a mixture’s covariance

ω_{jm}^{m} The Log-R of the mode based on µ_{jm}^{m}(k)[k]

Ω^{m}(k) The covariance of [ω_{1}(k) ω_{2}(k)]'

p(·) Any probability density function (PDF)

P(·) Any probability mass function (PMF)

P_{jm}^{m}(k) The covariance of the moment-matched fused estimate output error

P_{jm}^{m}(k) The covariance of the fused nth MCEE

P_{jm}^{m}(k) The locally-computed covariance of x_{jm}^{m}(k)

P_{jm}^{m}(k) The locally-computed covariance of x_{jm}^{m}(k)

P_{jm}^{m}(k) The FC-computed complete covariance/crosscovariance of x_{jm}^{m}(k)[k – 1]

P_{jm}^{m}(k) The FC-computed complete covariance/crosscovariance of the MCEE

φ_{jm}(k) The received information from LT j

π_{jm}^{lm} The Markov Chain transition probability from mode l to mode m

Q^{m} The covariance of the process noise under mode n

Rj Covariance of measurement error at LT j

S^{m}(k) The FC-computed complete covariance/crosscovariance of the MCI

S_{jm}^{m}(k) The locally-computed covariances of ν_{jm}^{m}(k)

TPM Transition Probability Matrix of Markov Chain

v^{m}(k) The zero-mean process noise under mode n

w_{jm}^{m}(k) The zero-mean measurement error at LT j

W_{jm}^{m}(k) Locally-computed Kalman gain matrices

x_{jm}^{m} The moment-matched fused estimate output

x_{jm}^{m} The fused nth MCE

x_{jm}^{m}(k) The nth mode’s mixed initial conditions (from IMM algorithm)

x_{jm}^{m}(k) The nth mode’s mixed initial conditions (from IMM algorithm)

x_{jm}^{m}(k) Local tracker MCP

x_{jm}^{m}(k) Local tracker MCPE

x_{jm}^{m}(k) Local tracker MCE

x_{jm}^{m}(k) Local tracker MCEE

x(k) The true target state

y^{m}(k) The “state vector” of the joint IMM system, computed at the FC

Y^{m}(k) The covariance of y^{m}(k)

z_{jm}^{m}(k) Measurement at LT j
4.2 Description of Target and Local Trackers

For clarity, only two LT and two dynamic modes will be considered, but the extension to multiple trackers and modes is possible. Local state estimation is performed by two trackers obtaining noisy observations of a target whose dynamics may switch between two different modes. Each tracker \( j = 1, 2 \) computes \( m = 1, 2 \) mode-conditioned estimates (MCE) and covariances (MCC) of \( x(k) \). With \( Z_j^k \) as the vector of all measurements at tracker \( j \), up to and including the present time step,

\[
Z_j^k = [z_j(0)' z_j(1)' \ldots z_j(k)']' \tag{4.1}
\]

the \( N_x \)-dimensional MCE, conditioned on the current target mode \( M(k) \) being \( m \), are denoted and defined as

\[
\hat{x}_{jm}^m(k|k) \triangleq E[x(k) | Z_j^k, M(k) = m] \quad j = 1, 2 \tag{4.2}
\]

and the MCC are

\[
P_{jm}^m(k|k) \triangleq E \left[ \hat{x}_{jm}^m(k|k) \hat{x}_{jm}^m(k|k)' \mid Z_j^k, M(k) = m \right] \tag{4.3}
\]

The true state of the target, when in mode \( M(k) = n \), evolves linearly in time as

\[
x^m(n)(k+1) = Fx^m(n)(k) + v^m(n)(k) \tag{4.4}
\]

The \( N_{z_j} \)-dimensional measurements of the target at each tracker are obtained according to

\[
z_j(k) = H_j x(k) + w_j(k) \tag{4.5}
\]

The trackers compute the probability of the target being in mode \( m \) at time step \( k \)

\[
\mu_m^m(k) \triangleq P\left(M(k) = m \mid Z_j^k \right) \tag{4.6}
\]

The evolution of the target’s dynamic modes is modeled as a Markov chain. Its known transition probability matrix (TPM [11]) is

\[
\Pi = \begin{bmatrix} \pi_{11}^1 & \pi_{12}^1 \\ \pi_{21} & \pi_{22} \end{bmatrix} \tag{4.7}
\]

and all of its rows must have a sum of one.

4.3 IMM Inside-Information Fusion

4.3.1 The Posterior Fused State PDF

Omitting the time step index \( k \) for brevity, the posterior state PDF of the target state using the information from two trackers can be described as a Gaussian mixture

\[
p(x | \phi_1, \phi_2) = \sum_{n=1}^{2} p(x | \phi_1, \phi_2, n) P(M = n | \phi_1, \phi_2) \tag{4.8}
\]

with the received information from the two trackers defined as

\[
\phi_j \triangleq \{ \hat{x}_j^1(k|k), \hat{x}_j^2(k|k), \mu_j^m(k) \} \tag{4.9}
\]

where \( \mu_j^2(k) \) is ignored in (4.9) due to its redundancy.

The FC does not have access to any \( z_j \) or any past \( \hat{x}_j^m \), but should provide the best fused estimate and its error covariance when receiving the latest MCE and mode probabilities from all LT. To do this, the posterior fused mode-conditioned estimates \( p(x | \phi_1, \phi_2, M = n) \) and posterior fused mode probabilities \( P(M = n | \phi_1, \phi_2) \) will be derived next.

\[[Note that \( m \) is the index of the MCE at the LT. The FC must consider the received MCE and probabilities under all mode hypotheses, so \( n \) is used in the multiple-model inference process at the FC, while \( m \) is used only to index the received information.\]
4.3.2 The Fused Mode-Conditioned State Estimates

First, we assume that conditioned on the received MCE, the received mode probabilities do not contribute additional information about any mode-conditioned state vector. So, the fused posterior mode-conditioned PDFs from (4.8) are approximately

\[p(x | \phi_1, \phi_2, n) \approx p(x | \tilde{x}_1^1, \tilde{x}_1^2, \tilde{x}_2^1, \tilde{x}_2^2, M = n)\]

\[= \frac{1}{a}p(\tilde{x}_1^1, \tilde{x}_2^1, \tilde{x}_2^1 | M = n, x)p(x | M = n) \quad n = 1, 2 \quad (4.10)\]

with \(a\) a normalizing constant and \(p(x | M = n)\) considered non-informative (i.e. diffuse). The likelihood of the state in (4.10) is the PDF of the LT MCE, conditioned on the true state \(x(k)\) and true mode \(M(k) = n\), given as

\[p(\tilde{x}_1^1, \tilde{x}_2^1, \tilde{x}_2^2 | M = n, x) \quad (4.11)\]

with mean

\[E \left\{ [(\tilde{x}_1^1)' (\tilde{x}_2^1)' (\tilde{x}_2^2)']' | M = n, x \right\} = [x' x' x']' \quad (4.12)\]

and covariance (to be computed recursively at the FC as described in Sec. 4.3.8)

\[P^{(n)} = E \left\{ \left[ \begin{array}{c} \tilde{x}_1^1' \\tilde{x}_2^1' \\tilde{x}_2^2' \\ \tilde{x}_1^1' \\tilde{x}_2^1' \\tilde{x}_2^2' \\ \tilde{x}_1^1' \\tilde{x}_2^1' \\tilde{x}_2^2' \end{array} \right] | M = n \right\} \quad (4.13)\]

With (4.13) computed, the solution to (4.10) is the standard Linear Minimum Mean-Square Error (LMMSE) fusion given by (see [13])

\[\hat{x}_F^n(k) = \left[ \tilde{L}'(P^{(n)^{-1}}L) \right]^{-1} \tilde{L}'(P^{(n)^{-1}}X) \quad (4.14)\]

with corresponding fused covariance computed by

\[P_F^n(k) = \left[ \tilde{L}'(P^{(n)^{-1}}L) \right]^{-1} \quad (4.15)\]

and

\[L_{4N_x \times N_x} = \left[ I_{N_x \times N_x} \ I_{N_x \times N_x} \ I_{N_x \times N_x} \ I_{N_x \times N_x} \right]' \quad (4.16)\]

The 4\(N_x\)-element vector \(\hat{X}\) in (4.14) is

\[\hat{X} = [(\hat{x}_1^1)' (\hat{x}_2^1)' (\hat{x}_2^2)']' \quad (4.17)\]

### 4.3.3 The Log-Ratio Transformation and Linearized Evolution

Local mode probabilities are computed at the LT by multivariate Gaussian PDF likelihoods evaluated at the latest local measurements. Therefore, the probabilities are themselves stochastic processes. Since finding a parametric joint PDF of these nonlinear transformations is not feasible, a solution is to transform the probabilities into log-ratios and use a multivariable Gaussian approximation of the transformed variables. This approximation is appropriate because log-ratios have infinite support and the multivariable Gaussian density can capture dependencies by non-zero covariances. The means, variances, and covariances are then readily computed as those of the difference of quadratic forms of Gaussian random variables (the innovations).

The single log-ratio at LT \(j\), denoted as \(\omega_j\), is selected to be the log of the ratio of the mode 1 probability to the mode 2 probability as

\[\omega_j = \ln \frac{\mu_1^j}{\mu_2^j} \quad (4.18)\]

Note that only a single log-odds \(\omega_j\) uniquely determines the probability pair, so the second log-odds does not need to be included in the analysis of the likelihood function as it certainly does not provide additional information. If there are more than two modes, then any mode probability can serve as the common denominator for all the log-ratios, but the rest of this paper will concentrate on the two-mode scenario only.

The Log-R transformation is one-to-one, and the probabilities can be recovered using

\[\mu_1^j = \frac{e^{\omega_j}}{e^{\omega_j} + 1} \quad \mu_2^j = \frac{1}{e^{\omega_j} + 1} \quad (4.19)\]
The transformation allows the new variables to be represented as a nonlinear first-order Markov process driven by the (approximately) white\(^2\) MCI \(\nu^n_1(k)\) with LT-computed covariances \(S^n_{m}(k)\), where the normalizing constants that compute the updated probabilities are canceled in the ratio

\[
\omega_j(k) = \frac{\pi^{11} e^{j^n(k-1)} + \pi^{21}}{\pi^{12} e^{-j^n(k-1)} + \pi^{22}} + \frac{1}{2} \ln \left[ \frac{S^{j^n}_2(k)}{S^{j^n}_1(k)} \right] + \frac{1}{2} \nu_j^n(k) S^{j^n}_1(k)^{-1} \nu_j^n(k) - \frac{1}{2} \nu_j^n(k) S^{j^n}_2(k)^{-1} \nu_j^n(k)
\]

The means of the Log-R processes \(\omega_1(k), \omega_2(k)\) are non-zero, and they have finite variance and non-zero correlation. The first term in \((4.20)\), conditioned on mode \(n\), has the first-order Taylor series expansion around \(\omega_j^n(k-1)\) (the mixed initial condition — see Sec. 4.3.9)

\[
\ln \frac{\pi^{11} e^{j^n(k-1)} + \pi^{21}}{\pi^{12} e^{-j^n(k-1)} + \pi^{22}} \approx \left[ \frac{\pi^{11} e^{j^n(k-1)} + \pi^{21}}{\pi^{12} e^{-j^n(k-1)} + \pi^{22}} - \frac{\pi^{12} e^{j^n(k-1)} + \pi^{21}}{\pi^{12} e^{-j^n(k-1)} + \pi^{22}} \right] \omega_j^n(k-1)
\]

The expected value of \(\nu^n\) found by (see \([11]\))

\[
\omega_j^n(k-1) \approx \left[ \nu_j^n(k-1) \right] \omega_j^n(k-1)
\]

The \(\mu_j^{m|n}(k-1)\) are the (actual) initial condition mixing weights at the local IMMs, which are functions of the Log-R, all conditioned on mode \(n\)

\[
\mu_j^{m|n}(k-1) = \frac{\pi^{11} e^{j^n(k-1)} + \pi^{21}}{\pi^{12} e^{-j^n(k-1)} + \pi^{22}}
\]

and \(\bar{\mu}_j^{m|n}(k-1)\) are computed according to \((4.22)\) by using \(\omega_j^n(k-1)\) instead of \(\omega_j^n(k-1)\).

The last two terms of \((4.20)\) are the difference of quadratic forms of the innovations, which are correlated between the modes and sensors. Since they are unknown to the FC and they are stochastic, they are considered to be a common additive noise for the Log-R of both sensors, and the mean and covariance of this noise are readily computed to form a Gaussian approximation.

Omitting \(k\) again, the mean and covariance can be derived by first defining the stacked vector of the zero-mean innovations as

\[
\nu^n = \left[ (\nu_1^n)^T \ (\nu_2^n)^T \right]^T
\]

The covariance of \((4.23)\) is \(S^{n}\), derived in Sec. 4.3.8. Together with the selection matrices

\[
\begin{align*}
L^1_1 &= [1 0 0 0] & L^2_1 &= [0 1 0 0] \\
L^1_2 &= [0 0 1 0] & L^2_2 &= [0 0 0 1]
\end{align*}
\]

the quadratic forms can be written as

\[
(\nu_j^n)^T S^n_2^{-1} \nu_j^n - (\nu_j^n)^T S^n_1^{-1} \nu_j^n = (\nu^n)^T \left[ (L^1_j)^T (S^n_2)^{-1} (L^1_j)^T - (L^2_j)^T (S^n_1)^{-1} (L^2_j)^T \right] \nu^n
\]

The expected value of \(S^n_2\), a hidden matrix computed at the LT, can be computed using the algorithm in Sec. 4.3.10. The 2-dimensional, white, non-zero mean Gaussian random process \(d\) will approximate the quadratic form noise, having mean and variance/covariance found by (see \([11]\))

\[
d^n \triangleq E[d \mid n] = \left[ E[\nu^T M_1 \nu] \ E[\nu^T M_2 \nu] \right] = \left[ \text{tr} [M_1 S^{n}] \ E[\nu^T M_2 \nu] \right]
\]

\[
(D^n \triangleq E[(d - \bar{d})(d - \bar{d})^T] | n) = \left[ 2\text{tr} [M_1 S^{n}]^2 \ 2\text{tr} [M_1 S^{n} M_3 S^{n}] \right.
\]

\[
\left. 2\text{tr} [M_1 S^{n} M_2 S^{n}] \ 2\text{tr} [M_2 S^{n}]^2 \right]
\]

The covariance between a zero-mean Gaussian vector and a quadratic form in the same vector is zero \([44]\) — this means that \(d\) is not correlated to the innovations or the process/measurement noise.

\(^2\)There is no proof if, or when, IMM mode-conditioned innovations are truly white, as they are in a correctly-matched single-mode Kalman filter \([11]\).
4.3.4 Fused Mode Probabilities

We assume that all the information about the mode is contained in the received mode probabilities. Omitting the time step $k$, using Bayes’ theorem, the posterior fused mode probabilities are

$$\mu_{\omega}^n = P(M = n \mid \mu_1^n, \mu_2^n, \mu_3^n)$$

$$= P(M = n \mid \omega_1, \omega_2)$$

$$= \frac{p(\omega_2 \mid \omega_1, M = n)P(M = n \mid \omega_1)}{b}$$

$$= \frac{p(\omega_1, \omega_2, M = n)\mu_1^n}{bp(\omega_1 \mid M = n)}$$

(4.28)

with $b$ the normalizing constant and the likelihood function of the mode based on the Log-R represented as

$$p(\omega_1, \omega_2 \mid M = n)$$

(4.29)

The goal here is to find the prior mean $[\bar{\omega}_n]$, and the covariance $\Omega_n^m$, conditioned on mode $n$, of the Gaussian approximation of (4.29) before any data arrives. From this, the marginal in the denominator of (4.28) is easily found and the likelihood can be evaluated for each mode $n = 1, 2$.

4.3.5 The System State of the IMM Trackers

The vector of the mode-conditioned estimate errors (MCEE) and log-ratios describes the internal behavior of two IMM trackers estimating the state of the same target for the purpose of computing the required parameters of (4.11) and (4.29). Conditioned on mode $n$, it is defined as

$$y^n(k) \triangleq \begin{bmatrix} x_1^n(k|k)' & x_2^n(k|k)' & \omega_1^n(k) & \omega_2^n(k) \end{bmatrix}'$$

(4.30)

with the stacked vector of errors from each sensor written for compactness as

$$x_1^n(k|k) \triangleq \begin{bmatrix} x_1^{1|n}(k|k)' & x_2^{1|n}(k|k)' \end{bmatrix}'$$

(4.31)

The mean of (4.30) is (considering that the MCE have zero-mean error according to the assumptions of the IMM estimator)

$$\bar{y}^n(k) \triangleq E[y^n(k)] = \begin{bmatrix} 0 & \omega_1^n(k) & \omega_2^n(k) \end{bmatrix}'$$

(4.32)

The covariance of (4.30) is

$$Y^n(k) \triangleq E[y^n(k)y^n(k)'] = \begin{bmatrix} P^n(k) & 0 & 0 & \Omega^n(k) \end{bmatrix}$$

(4.33)

where the zero off-diagonals are a result of the block-diagonal Jacobian and additive noise covariance to be derived in Sec. 4.3.7 and Sec. 4.3.8, respectively. Recursions yield $y^n(k)$ and $Y^n(k)$, under each hypothesis $n = 1, 2$, from which the parameters of the likelihood functions, (4.11) and (4.29), can be computed. This will be developed in Sec. 4.3.8. These parameters are not conditioned on any previous track information, but they do require knowledge of the measurement models, the dynamic models, and the TPM.4

4.3.6 The LT Mode-Conditioned State Estimate Errors (MCEE)

Recursive covariance computations can be used to find the covariance of the zero-mean MCEE. At every step $k$, there are two mode hypotheses, represented by $n = 1, 2$, and there is a mixing process analogous to the IMM algorithm that will be described in Sec. 4.3.9. As in Fig. 4.1, the error of the mode $n$ prediction at tracker $j$, conditioned on mode $n$ being the true mode, is

4The representation of (4.28) is not unique — either mode probability can be used as the prior, or the prior can be non-informative. The attractiveness of using a received probability as a prior is the ability to use as much information in the data as possible before the Gaussian approximation. In other words, the ability to directly factor in a probability as a prior can be advantageous from an accuracy perspective.

4Due to the recursive algorithm that computes the covariances, initial conditions do need to be provided. Standard covariance initialization methods can be used (see [11]) and the mode-conditioned mean of the Log-R can be initialized to zero. This can be accomplished offline.
\[
\begin{align*}
\mathbf{x}^m_j(k+1|k) & \triangleq \mathbf{F}\mathbf{x}(k) + \mathbf{v}^n(k) - \mathbf{F}\mathbf{x}^m_j(k|k) \\
& = \mathbf{F} \left[ \mathbf{x}(k) - \left[ \mu_j^{1m|n} \mathbf{x}^{1|n}_j(k|k) + \mu_j^{2m|n} \mathbf{x}^{2|n}_j(k|k) \right] \right] \\
& \quad + \mathbf{v}^n(k) \\
& = \mathbf{F} \left[ \mu_j^{1m|n} \mathbf{x}^{1|n}_j(k) + \mu_j^{2m|n} \mathbf{x}^{2|n}_j(k) \right] \\
& \quad + \mathbf{v}^n(k) \\
& \quad (4.34)
\end{align*}
\]

Since the Log-R can be considered system states, the weighting is a nonlinear function of the state variables, which can be linearized by using Jacobians (see Sec. 4.3.7).

The MCEE are propagated from the previous mode-conditioned prediction errors (MCPE) as

\[
\mathbf{x}(k) - \mathbf{x}^{m|n}_j(k|k) = (\mathbf{I} - \mathbf{W}^m_j(k)\mathbf{H}_j)\mathbf{x}^{m|n}_j(k|k-1) - \mathbf{W}^m_j(k)\mathbf{w}_j(k) \\
(4.35)
\]

The Kalman gains \(\mathbf{W}_j(k)\) are unknown to the FC directly, but expected values can be used in their place. See Sec. 4.3.10.

### 4.3.7 The System State Transition Jacobian

From (4.34) and (4.35), the Jacobians of the MCPEs with respect to the previous MCEE are

\[
\mathbf{J}_{\mathbf{x}^{m|n}_j(k|k)}(k) = \mu_j^{1m|n}(k)\delta_{i-j} \\
(4.36)
\]

where \(\delta_{i-j}\) is the Kronecker Delta function (i.e. the cross-sensor Jacobians in (4.36) are zero).

The Jacobians of the MCPEs with respect to the previous Log-R, evaluated at the mean of the errors (which are zero), are zero:

\[
\mathbf{J}_{\mathbf{x}^{m|n}_j(k|k)}(k) = \frac{\pi^{1m} \pi^{2m} e^{\omega j} \mathbf{F}}{\pi^{1m} \pi^{2m}} \mathbf{E} \left[ \mathbf{x}^{1|n}_j(k|k) - \mathbf{x}^{2|n}_j(k|k) \right] = 0 \\
(4.37)
\]

The Jacobians of the Log-R with respect to their previous values can be derived from (4.21). The Jacobians of the Log-R with respect to the previous MCEE are zero since the partial derivative of the quadratic form of innovations with respect to an innovation is scaled by that innovation, which is zero-mean. This is in agreement with the claim that \(\mathbf{d}_j(k)\) can be treated as white, additive noise.

Omitting \(k\), the complete Jacobian is

\[
\mathbf{J}^{1|n} = \begin{bmatrix}
\mu_1^{1|n} \mathbf{F} & \mu_2^{21|n} \mathbf{F} & 0 & 0 & 0 \\
\mu_1^{12|n} \mathbf{F} & \mu_2^{22|n} \mathbf{F} & 0 & 0 & 0 \\
0 & 0 & \mu_2^{11|n} \mathbf{F} & \mu_2^{21|n} \mathbf{F} & 0 \\
0 & 0 & \mu_2^{12|n} \mathbf{F} & \mu_2^{22|n} \mathbf{F} & 0 \\
0 & 0 & 0 & 0 & \mathbf{J}^{\omega_j^{m|n}(k+1)} \omega^{m|n}(k)
\end{bmatrix} \\
(4.38)
\]

with

\[
\mathbf{J}^{\omega_j^{m|n}(k+1)} \omega^{m|n}(k) = \begin{bmatrix}
\mu_1^{1|n} - \mu_1^{12|n} & 0 \\
0 & \mu_1^{2|1|n} - \mu_2^{12|n}
\end{bmatrix} \\
(4.39)
\]

### 4.3.8 Recursion for the System Mode-Conditioned Means and Covariances

Having computed \(\mathbf{J}^{1|n}(k)\), the linearized system description for (4.30) under mode \(n\) becomes

\[
\mathbf{y}^{m|n}(k+1) = \mathbf{K}(k)\mathbf{J}^{1|n}(k)\mathbf{y}^{n|n}(k) + \Gamma^{1|n}(k)\mathbf{g}^{m|n}(k) \\
(4.40)
\]

with

\[
\mathbf{K}(k) \triangleq \text{diag} (\mathbf{K}_1^1, \mathbf{K}_2^1, \mathbf{K}_1^2, \mathbf{K}_2^2) \\
\mathbf{K}_j^{m|n}(k) \triangleq \mathbf{I} - \mathbf{W}_j^{m|n}(k)\mathbf{H}_j \\
(4.41)
\]

The noise vector

\[
\mathbf{g}^{m|n}(k) = \begin{bmatrix} \mathbf{v}^{n|n}(k) & \mathbf{w}_1(k) & \mathbf{w}_2(k) & \mathbf{d}(k) \end{bmatrix}' \\
(4.42)
\]
with steady-state Markov chain probabilities.

Before every recursion update step (4.46)–(4.47), the hypotheses from the previous step must be merged, just as they are in

4.3.9 The Mixing Process for Hypothesis Merging

Before every recursion update step (4.46)–(4.47), the hypotheses from the previous step must be merged, just as they are in

The system’s covariance update is

\[ \mathbf{P}^{ln}(k+1) = \mathbf{P}^{ln}(k) + \Gamma^{ln}(k) \mathbf{G}^{ln}(k) \mathbf{G}^{ln}(k)' \Gamma^{ln}(k) \]

representing the covariance of the local MCPE, where only the first 4N_x rows and columns of \( \mathbf{J}^{ln}(k) \mathbf{Y}^{ln}(k) \mathbf{J}^{ln}(k)' \) are selected, the covariances of the MCIs can be computed as

\[ \mathbf{S}^{ln}(k+1) = \mathbf{H} \mathbf{P}^{ln}(k+1|k) \mathbf{H}' + \mathbf{L}_w \begin{bmatrix} \mathbf{R}_1 & 0 \\ 0 & \mathbf{R}_2 \end{bmatrix} \mathbf{L}_w' \]

with

\[ \mathbf{H} = \text{diag}(\mathbf{H}_1, \mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_2) \]

\[ \mathbf{L}_w = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \]

4.3.9 The Mixing Process for Hypothesis Merging

Before every recursion update step (4.46)–(4.47), the hypotheses from the previous step must be merged, just as they are in

the IMM estimation algorithm. In the absence of any previous observations, the mixing probabilities are computed from the
steady-state Markov chain probabilities \( \mu^{ln}_1, \mu^{ln}_2 \) as priors [11]

\[ \mu^{ln} = \frac{\sigma^{ln} \mu^{ln}_1}{\sigma^{ln} \mu^{ln}_1 + \sigma^{ln} \mu^{ln}_2} \]

With \( M(k+1) = n \) the event that the next mode is \( n \), the mixed initial conditions are

\[ \mathbf{y}^{ln}(k) = E[\mathbf{y}(k) \mid M(k+1) = n] = \sum_{l=1}^{2} \mu^{ln} \mathbf{y}^{ll}(k) \]

\[ \mathbf{Y}^{ln}(k) = E \left[ \left( \mathbf{y}(k) - \mathbf{y}^{ln}(k) \right) \left( \mathbf{y}(k) - \mathbf{y}^{ln}(k) \right)' \mid M(k+1) = n \right] \]

\[ = \sum_{l=1}^{2} \mu^{ln} \left[ \mathbf{Y}^{ll}(k) + \left( \mathbf{y}^{ll}(k) - \mathbf{y}^{ln}(k) \right) \left( \mathbf{y}^{ll}(k) - \mathbf{y}^{ln}(k) \right)' \right] \]

The notation \( k+1|k \) used in covariances computed at the FC serves only to show that they are related to the state predictions made at the LT. It is not intended to mean that the computations at the FC are conditioned on past measurements.
4.3.10 Typical Values of Stochastic Matrices

The local mode-conditioned innovation covariances $S^m_j(k)$ and Kalman gains $W^m_j(k)$ are required for (4.25), (4.41), and (4.45). To do this, it should be first noted that both $S^m_j(k)$ and $W^m_j(k)$ are stochastic matrices, computed from a mixed previous value and the spread-of-the-means (SOM) of the Gaussian mixture (i.e. the mixing process is measurement-dependent) [11]. Because the MCE depend on these stochastic matrices, the covariances as computed by the LT IMM algorithm behave like “state variables” of the system and are recursively updated. While a full probabilistic treatment is beyond the scope of this paper, the mean of each matrix can be computed through linearization and recursion, then treating the resulting matrices as having zero variance (i.e. invoking an approximate Certainty Equivalence property [12]).

This can be accomplished by dedicating a recursion process to finding the matrix means of the mixed initial condition covariances $P^m_j(k-1|k-1)$ and using them to find $S^m_j(k)$ and $W^m_j(k)$ using standard Kalman equations.

First, it is noted that mixed initial conditions $\bar{x}^m_j(k)$ and $P^m_j(k|k)$ do not depend on the state or mode at $k + 1$. Then, the recursion for the matrix mean can be linearized as

$$
\bar{P}^m_j(k|k) \triangleq E \left[ P^m_j(k|k) \right] 
= E \left[ P^m_j(k|k) \right] 
= \sum_{n=1}^{2} E \left[ \sum_{l=1}^{2} \mu^{lm}_j(k) P^m_j(k|k) \right] 
+ N^{lm}_j(k) \mid M(k) = n \cdot P \left[ M(k) = n \right] 
\approx \sum_{n=1}^{2} \left[ \sum_{l=1}^{2} \mu^{lm}_j(k) E \left[ P^m_j(k|k) \mid M(k) = n \right] \right] 
+ E \left[ N^{lm}_j(k) \mid M(k) = n \right] \mu^m_{o}$

(4.55)

Here, $\mu^{lm}_j(k)$ can be taken out of the expectation (as a first-order linear approximation) and evaluated using $\omega^{lm}_j(k)$ and (4.22); $N^{lm}_j(k)$ is the SOM. The expectation in the first term can be computed by

$$
P^m_j(k|k) \triangleq E \left[ P^m_j(k|k) \mid M(k) = n \right] 
= E \left[ P^m_j(k|k) \right] 
= FP^m_j(k-1|k-1)F' + Q' - W^m_j(k)S^m_j(k)W^m_j(k)'$

(4.56)

Omitting the step $k$, the expected value of the SOM can be derived starting with

$$
E \left[ N^{lm}_j \mid M(k) = n \right] = E \left[ \bar{x}^m_j - \bar{x}^m_j \right] \left[ \bar{x}^m_j - \bar{x}^m_j \right]' \mid M(k) = n
$$

(4.57)

and expanding the difference as

$$
\bar{x}^m_j - \bar{x}^m_j = \bar{x}^m_j - \sum_{a=1}^{2} \mu^{am}_j \bar{x}^m_j 
= \begin{cases} 
\mu^{2m}_j \bar{x}^m_j - \bar{x}^m_j & \text{if } l = 1 \\
\mu^{1m}_j \bar{x}^m_j - \bar{x}^m_j & \text{if } l = 2 
\end{cases}
$$

(4.58)

Since the two MCE have the same mean, $x$,

$$
E \left[ N^{lm}_j \mid M(k) = n \right] = \begin{cases} 
(\mu^{2m}_j)^2 \left( P^{11m}_{jj} + P^{22m}_{jj} - P^{12m}_{jj} - P^{21m}_{jj} \right) & \text{if } l = 1 \\
(\mu^{1m}_j)^2 \left( P^{11m}_{jj} + P^{22m}_{jj} - P^{12m}_{jj} - P^{21m}_{jj} \right) & \text{if } l = 2 
\end{cases}
$$

(4.59)

where each $P^{lm}_{jj}$ is a respective block of (4.13). Finally, expected values of the stochastic matrices can be computed as

$$
P^m_j(k|k-1) = FP^m_j(k-1|k-1)F' + Q^m$

(4.60)

$$
\bar{S}^m_j(k) = H^t_j P^m_j(k|k-1)H^t_j + R_j$

(4.61)

$$
W^m_j(k) = P^m_j(k|k-1)H^t_j \bar{S}^m_j(k)^{-1}$

(4.62)

---

6 While the zero-variance approximation is heuristic, simulation results show that the method is successful.
4.3.11 Summary

A linearized system description of two parallel IMM LT is depicted Fig. 4.1. This diagram shows the utility of the model: the white sequences, \( v_n(k) \) and \( d_{\mid n}(k) \), act as common inputs to both IMM subsystems, the measurement errors \( w_i(k) \) act as independent inputs to each IMM, and the MCEE \( \tilde{x}_{mj}(k\mid k) \) and log-ratios \( \omega_j(k) \) act as the outputs. It is the Gaussian PDF parameters of these outputs that are of interest.

Figure 4.1: Block diagram of the linearized model of the errors and log-ratios of two IMM LT (with only one shown explicitly) for the mode-conditioned error covariance, Log-R mean, and Log-R covariance computation. The computations for the stochastic matrices \( S^m_j(k) \) and \( W^m_j(k) \) are not shown.

4.4 Simulation Results

The simulations have the local IMM estimators tracking a target in 2D space capable of constant turn rates (i.e. coordinated turns, see [11]) observed by sensors that are measuring its Cartesian position. Two local trackers run IMM estimators and use two dynamic modes described in the sequel. Three scenarios are considered: the first has a deterministic target trajectory (ground truth) using a coordinated-turn model and fusion at full rate; the second has the same deterministic trajectory with fusion at a reduced rate (as a sanity check since the proposed method’s performance at fusion times is not affected by fusion rate), and the third simulates random trajectories driven by white noise, with a dynamic model matching that of the estimators and with fusion at full rate.

4.4.1 Ground Truth

The measurement interval is \( T = 1 \text{s} \). The target starts at \( x = 0, y = 0 \) with \( \dot{x} = 100 \text{m/s}, \dot{y} = 100 \text{m/s} \). The target:

1. Travels straight for 25.2 s
2. Performs a constant-rate left turn of 3°/s for 10.6 s.
3. Travels straight for 18 s
4. Performs a constant-rate right turn of −3°/s for 4.1 s.
5. Performs a constant-rate left turn of 1.3°/s 12.8 s.
6. Travels straight for 22.6 s

A plot of this constant-speed, variable-turn rate trajectory is shown in Fig. 4.4.
Figure 4.2: Block diagram of IMM inside information fusion. $\tilde{\omega}^{|in}$ is part of $y^{|in}$ and $Y^{|in}$ is block-diagonal with $P^{|in}$ and $\Omega^{|in}$ as its respective blocks. The relevant Gaussian likelihood functions used for fusion are shown in the fusion blocks.

Figure 4.3: Block diagram of conventional outside fusion method. The Gaussian likelihood function used is written in the fusion block.
4.4.2 Estimation Models

The state vector is composed of stacked position and velocity

\[ x(k) = [x(k) \ y(k) \ \dot{x}(k) \ \dot{y}(k)]' \]  

(4.63)

The estimation dynamic models are described as follows. Mode 1 is a 2D White Noise Acceleration (WNA) model, discretized from the continuous-time model [11]. It has a 2-dimensional process noise acceleration with intensity (power spectral density — PSD) \( \tilde{q}_1^2 = 0.01^2 \text{ m}^2/\text{s}^3 \), and Mode 2 is the same but with \( \tilde{q}_2^2 = 7.5^2 \text{ m}^2/\text{s}^3 \)

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

(4.64)

\[
Q^n = \begin{bmatrix}
\frac{1}{2}T^3 & 0 & \frac{1}{2}T^2 & 0 \\
0 & \frac{1}{2}T^3 & 0 & \frac{1}{2}T^2 \\
\frac{1}{2}T^2 & 0 & T & 0 \\
0 & \frac{1}{2}T^2 & 0 & T \\
\end{bmatrix} \tilde{q}^n
\]  

(4.65)

The TPM is

\[
\Pi = \begin{bmatrix}
0.95 & 0.05 \\
0.05 & 0.95 \\
\end{bmatrix}
\]  

(4.66)

The measurement parameters are

\[
H_1 = H_2 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\end{bmatrix}
\]  

(4.67)

\[
R_1 = \text{diag}((15 \text{ m})^2, (18 \text{ m})^2)
\]  

(4.68)

\[
R_2 = \text{diag}((20 \text{ m})^2, (25 \text{ m})^2)
\]  

(4.69)
4.4.3 Fusion Results

Figs. 4.5 and 4.6 show the position and velocity RMSE for the inside information fusion, outside information fusion, and centralized measurement fusion methods, along with the RMSE of the local sensor tracks. The inside fusion is slightly outperformed by the outside fusion during straight-line motion, just as the centralized fusion is, but inside fusion significantly outperforms outside fusion during maneuvers. The inside-information fusion clearly has performance in between that of centralized fusion and the conventional outside-information fusion.

The consistency is evaluated using the Normalized Estimation Error Squared (NEES, see [11]), divided by $N_x$ (the state dimension) and this is plotted for every time point in Fig. 4.7. Values near 1 are ideal and reflect a Chi-Square quadratic form resulting from estimation errors that are zero-mean and consistent with the state covariances. It is clear that the inside-information method achieves better consistency.

In Fig. 4.8, the mode probabilities of the inside-information fusion are compared to the local IMM mode probabilities and the centralized measurement fusion mode probabilities. The fused mode probabilities computed by the inside-information fusion slightly lead the probabilities of the local sensors when transitioning modes and, so, maneuvers can be detected quicker than they can be at the local sensors. Although the transient performance is encouraging, it can also be seen that the method as described in this paper results in fused mode probabilities that are “more sure” about the mode — centralized fusion is more conservative and only boosts this conviction slightly.

4.4.4 Reduced Rate Fusion

As a sanity check, it should be shown that fusion performance is not affected by the rate at which track data are transmitted. An advantage to the track-to-track fusion using inside information presented here is that it is not affected by previous tracks. Outside information fusion is known not to be affected by fusion rate because it utilizes the standard Gaussian fusion method without memory. As can be seen in Fig. 4.2, the Log-R mean/covariance and the MCEE covariance are recursively updated whether there is track information or not, and received tracks are not used in that computation (in the scenario presented here, $\bar{y}^m$ and $Y^m$ can even be computed offline).

Figs. 4.9 – 4.11 show the comparison of outside information fusion to the inside information fusion at a reduced rate of once every 5 measurement intervals, starting at $k = 4$. Looking closely, the performance at the fusion times matches the same performance as in Figs. 4.5 – 4.7. Again, it can be seen that the inside information fusion, like CMF, only outperforms outside information fusion during maneuvers, but the consistency of the MSE is significantly superior for inside info fusion.

4.4.5 Simulations Using Random, Model-Matched Target Trajectories

While the simulations of the previous sections were carried out using a single realization of the true target trajectory, the trajectory simulations of this section are randomized for every Monte Carlo run. This provides a better comparison of the
Figure 4.6: Velocity RMSE of local IMMs, outside fusion, inside fusion, and centralized measurement fusion

Figure 4.7: NEES of local IMMs, outside fusion, inside information fusion, and centralized measurement fusion
Figure 4.8: Computed probability of mode 1 from local IMMs (S1, S2), inside information fusion (IIF), and centralized measurement fusion (CMF)

Figure 4.9: Position RMSE of outside information fusion and inside information fusion at reduced rate, fusing tracks once every 5 measurement intervals
Figure 4.10: Velocity RMSE of outside information fusion and inside information fusion at reduced rate, fusing tracks once every 5 measurement intervals

Figure 4.11: NEES of outside information fusion and inside information fusion at reduced rate, fusing tracks once every 5 measurement intervals
Figure 4.12: Position RMSE of outside information fusion and inside information fusion at reduced rate, fusing tracks once every 5 measurement intervals

overall behavior of the algorithms and highlights the consistency of the inside-information fusion.

The random trajectories are created using the White Noise Acceleration model driven by zero-mean white noise having covariance given in (4.65).\textsuperscript{7} The mode $\eta$ is selected according to realizations of the Markov chain having TPM (4.66). The results are shown in Figs. 4.12 – 4.14. It can be seen that the position RMSE is pretty much equal for all three fusion methods, and that fusion with outside information has more velocity RMSE. Due to the matched target and estimator parameters, the NEES, normalized to nominal one, shows the overall MSE consistency of the IMM trackers (where CMF is simply an IMM with stacked measurement vectors). Fusion using inside information can be seen to be as consistent as centralized fusion, demonstrating that it is a fusion that accounts for error correlations (i.e. the crosscovariances) and provides consistent fused covariance output. Outside info fusion has NEES that is 50% higher than the ideal NEES of inside info fusion meaning that the fused estimate covariance from outside fusion is, on average, 33% smaller than it should be given the actual sample error covariance.

4.5 Conclusions

A system model of two IMM trackers estimating the state of a maneuvering target was presented for track-to-track fusion using information from inside the local IMM estimators. The fusion estimator produces a posterior mean and covariance, reduced from a Gaussian mixture, computed from IMM track information coming to a Fusion Center from two local trackers, with the target modeled as jumping between two dynamic modes. The linearized system model, together with the log-ratio transformation of the received mode probabilities, yields covariances and crosscovariances of the local mode-conditioned errors (and mode-conditioned means of the log-ratios). From these, the parameters of the likelihood functions of the mode-conditioned state and the mode are derived. Each fused mode-conditioned state estimate uses information from all received MCE. The result is on-demand Bayesian fusion capability with no previous fused track information needed. Compared to the naive fusion of moment-matched Gaussian track information (i.e., outside information fusion), the new method achieves performance closer to the centralized measurement fusion method and outperforms the naive fusion in both RMSE and covariance consistency, most notably when the target is in a maneuvering mode. Fusion using inside information was shown to be consistent on average as it accounts for the crosscovariance of the local estimate errors and mode probabilities, whereas fusion with outside information and no crosscovariance has a computed covariance that is 33% too small on average.

\textsuperscript{7}White noise is a requirement for the state of the system to be a Markov process, which is a requirement for the existence of an estimator [11].
Figure 4.13: Velocity RMSE of outside information fusion and inside information fusion at reduced rate, fusing tracks once every 5 measurement intervals

Figure 4.14: NEES of outside information fusion and inside information fusion at reduced rate, fusing tracks once every 5 measurement intervals
Bibliography


73
