8-2016

Information inference for cyber-physical systems with application to aviation safety and space situational awareness

Sangjin Lee
Purdue University

Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Sangjin Lee

Entitled

Information Inference for Cyber-Physical Systems with Application to Aviation Safety and Space Situational Awareness

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Inseok Hwang
Chair

Jianghai Hu

Carolin E. Frueh

Martin J. Corless

Dengfeng Sun

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Inseok Hwang

Approved by: Weinong Wayne Chen 6/29/2016
Head of the Departmental Graduate Program Date

INFORMATION INFERENCE FOR CYBER-PHYSICAL SYSTEMS
WITH APPLICATION TO
AVIATION SAFETY AND SPACE SITUATIONAL AWARENESS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Sangjin Lee

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2016

Purdue University

West Lafayette, Indiana

"Progress, Not Perfection"

This dissertation is dedicated to my parents, Youngheung Lee and Sunye Seo.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my adviser, Professor Inseok Hwang, for his continuous support and guidance for my PhD study and related research. I am very grateful for his patience, motivation, and enthusiasm for research, from which I have learned a lot about right attitudes of a researcher. Thanks to his invaluable advice and guidance, I could successfully solve many challenging research problems and complete my PhD.

I would like to thank my other committee members, Professor Martin Corless, Professor Jianghai Hu, Professor Dengfeng Sun, and Professor Carolin Frueh for their insightful comments and encouragement. I also want to thank Dr. Nikunj Oza in the NASA Ames Research Center for his support in completing a NASA project that I participated in.

My sincere gratitude goes to my former advisers in Yonsei University, Professor Sang-Young Park and Professor Chandeok Park, for their valuable suggestions and guidance. They helped me a lot prepare myself to study in America and make right decisions for my academic career.

Last but not the least, I would like to extend my sincerest thanks and appreciation to my parents for their unconditional love and emotional support. Saving the most important for last, I wish to give my heartfelt thanks to my wife, Jooyoung, for her endless love, patience, and continual support of my academic endeavor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Lee, Sangjin PhD, Purdue University, August 2016. Information Inference for Cyber-Physical Systems with Application to Aviation Safety and Space Situational Awareness. Major Professor: Inseok Hwang.

Due to the rapid advancement of technologies on sensors and processors, engineering systems have become more complex and highly automated to meet ever stringent performance and safety requirements. These systems are usually composed of physical plants (e.g., aircraft, spacecraft, ground vehicles, etc.) and cyber components (e.g., sensing, communication, and computing units), and thus called as Cyber-Physical Systems (CPSs). For safe, efficient, and sustainable operation of a CPS, the states and physical characteristics of the system need to be effectively estimated or inferred from sensing data by proper information inference algorithms. However, due to the complex nature of the interacting multiple-heterogeneous elements of the CPS, the information inference of the CPS is a challenging task, where exiting methods designed for a single-element dynamic system (or for even dynamic systems with multiple-homogenous elements) could not be applicable. Moreover, the increasing number of sensor resources in CPSs makes the task even more challenging as meaningful information needs to be accurately and effectively inferred from huge amount of data, which is usually noise corrupted. Many aerospace systems such as air traffic control systems, pilot-automation integrated systems, networked unmanned aircraft systems, and space surveillance systems are good examples of CPSs and thus have the aforementioned challenging problems.

The goals of this research are to 1) overcome the challenges in complex CPSs by developing new information inference methodologies based on control, estimation, hybrid systems and information theories, and 2) successfully apply them to various complex and safety-critical aerospace systems such as air transportation systems,

space surveillance systems, and integrated human-machine systems, to promote their efficiency and safety.

1. INTRODUCTION

1.1 Background and Motivations

In recent years, engineering systems have become increasingly automated through the development of advanced sensing, communication, and computing technologies. Many sensors and computing devices in the systems are tightly coupled to effectively monitor and control their physical processes and environment. Such system consisting of interacting cyber elements (computing, sensing, and communication components) and physical processes is called a Cyber-Physical System (CPS) [1, 2]. Many of aerospace systems can be classified as CPSs including air traffic control systems, space surveillance systems, integrated pilot-automation systems, unmanned aircraft systems, etc. For example, in the air traffic control systems, many sensors (e.g., GPS and radars) are used to collect information about multiple aircraft (physical processes), and the collected information is used by air traffic management decision tools (computing components) to compute control strategies necessary for safe and efficient air traffic flow management [3].

In a CPS, estimation and inference of the system's states and physical characteristics are very important, as the information is critical for analyzing the system's current status and computing necessary control inputs to the system for achieving important functionalities. For example, position and velocity information of aircraft are crucial for the air traffic control system, as it can be used to predict the risk of collisions between aircraft, based on which air traffic controllers can issue a necessary advisory to resolve the risk, ensuring the safety of airspace operation [4]. However, in most applications, the states and physical parameters of a system are not directly observable and thus need to be inferred from noisy sensor measurements, which leads to state estimation and inference problems.

Due to the complex nature of the interacting multiple-heterogeneous elements of a CPS, designing proper information inference algorithms is a quite challenging task. Some of the main challenges are described as follows.

First, in CPSs, many sub-elements are integrated using communication networks, where a large amount of data including sensor measurements or control signals are transmitted among the elements. In such sensor networks and networked control systems, the limitation of communication bandwidth and power resource for data transmission has been a great issue, as they can largely affect the performance of the whole networked system [5, 6]. This issue can be dealt with by introducing an event-based sampling framework [7–10]. Rather than transmitting information at synchronous time intervals, the estimator and the sensor nodes can communicate only when some interesting events happen, thereby decreasing the data transmission rate. Although the event-based sampling is useful, the resulted filtering problem is quite challenging because the observations are sparse and irregular. This implies that the traditional filtering theories that assume the availability of observations at every regular sampling time are not applicable.

Second, many CPSs can be modeled as hybrid systems, which involve the interaction of discrete states (or modes) and continuous states [11, 12]. The discrete state dynamics describe logical behaviors of a CPS such as the logic of embedded controllers or transitions between multiple modes of operation of the system. For a given discrete state (or mode), the mode-conditioned continuous state dynamics describe the physical behavior of the system, such as the continuous response of a physical component to the control input. For example, in air traffic control applications, the dynamics of an aircraft can be represented as a hybrid system since the aircraft's behavior consists of both the logical behavior (discrete transitions between different flight modes (discrete states)) and the physical behavior (the aircraft's continuous motion (continuous state) corresponding to a specific flight mode). To monitor and control hybrid systems, it is necessary to estimate both the discrete state and continuous state using noisy measurements, which is challenging since it requires intractable computation of the exponentially increasing number of hypotheses for all the possible discrete state histories [13].

Another challenge arises when the states of CPSs are subject to abrupt changes due to system component failures or external environmental changes. For example, in space surveillance systems, the velocity of a spacecraft can abruptly change due to impulsive thrust applied for orbital transfers [14, 15]. Since the rapid change in the trajectory of the maneuvering spacecraft can cause dangerous and imminent safety issues to adjacent spacecraft, the abrupt changes should be accurately monitored for the safe operation of space. The state estimation of systems with abrupt state jumps is quite challenging because an estimation algorithm should be able to detect the abrupt changes in a timely manner and then appropriately adjust the estimation filters to compensate for the changes [16, 17]. The detection scheme also has to be robust to noisy measurements, because false detection due to noise can degrade the performance of the state estimation (e.g., tracking accuracy in the space surveillance example). These challenges necessitate the development of an adaptive estimation technique, which can perform both the robust state jump detection and the corresponding filter compensation in a systematic manner.

1.2 Objectives and Contributions

The contributions of this thesis are twofold: 1) theoretical development of new information inference (state estimation) algorithms that overcome the aforementioned challenges in complex CPSs, and 2) application of the developed algorithms to information inference problems of various CPSs in the filed of aerospace engineering.

1.2.1 Theories

The first objective of this thesis is to develop a new state estimation algorithm that can effectively deal with the challenges due to the event-based sampling structure of CPSs. To achieve this goal, we first propose a theoretical framework to mathematically formulate the continuous-time nonlinear event-based state estimation problem. In the proposed framework, Stochastic Differential Equations (SDEs) [18] are introduced to model the evolution of the continuous-time nonlinear dynamics, and a mathematical model is proposed to describe the generation of measurements made by event-based sampling. The event-based state estimation problem is then formulated to compute the probability density function (pdf) of the state of the SDEs, with the sequence of noisy measurements obtained by the event-based sampling. To solve the event-based state estimation problem, a numerical algorithm based on the Markov chain approximation method [19–21] is proposed. The proposed algorithm first discretizes the original state space into a grid space, and constructs a Markov chain on the grid space that approximates the evolution of the original SDEs. The pdf of the state of the SDEs is then approximated by computing the probability mass function of the state of the Markov chain. It has been shown that the evolution of the pdf of the original SDEs, which is complex due to the sparsity and irregularity of measurements made by the event-based sampling, can be accurately and effectively approximated by the proposed algorithm.

The second objective is to develop a new state estimation algorithm, called the Event-Based Hybrid State estimation (EBHSE) algorithm, that can address the challenges caused by the hybrid system structure of CPSs as well as the event-based sampling [22]. To mathematically describe the hybrid system structure with uncertainty, a mathematical model called the Stochastic Hybrid System (SHS) [12, 23] is introduced. Based on the model, the hybrid state estimation problem is formulated as to compute the probability density of the hybrid state (continuous and discrete) of the SHS with the noisy measurements generated at certain events by the event-based sampling. The optimal solution for the hybrid state estimation requires the computation of the exponentially increasing number of probabilities of the discrete state histories [13]. To deal with this computational complexity, we exploit the idea of the interacting multiple model approach [24] that keeps the number of the discrete state histories constant via a mixing technique. In addition, the sparsity and irregularity of measurements resulted from the event-based sampling lead to the evaluation of a multivariate integration over the measurement space, which is computationally demanding. To efficiently compute the integration, we propose a pseudo measurement generation method, where the multivariate integration is approximated as the weighted sum of simple Gaussian functions evaluated at each pseudo measurement.

Another objective of this thesis is to develop a new adaptive estimation algorithm for CPSs whose states are subject to abrupt changes. To achieve this objective, we first propose to model such systems as the SHS with two discrete modes, each of which describes the motion of the system with or without abrupt state jumps. Then, the abrupt state jumps are modeled as transitions between discrete modes and the probability of the transition is characterized by the discrete state dynamics. For the discrete state dynamics, we propose to use a continuous state-dependent transition model in the sense that the probability of abrupt state jumps is dependent on the value of the continuous state in many systems [25–28]. For example, for a geostationary satellite performing station-keeping maneuvers, impulsive maneuvers (i.e., abrupt state jumps) are likely to occur when its longitude or latitude (i.e., continuous state) approaches predefined bounds around a desired location [29]. Based on the SHS modeling, the estimation of systems with abrupt state jumps can be performed under the hybrid state estimation framework, where the discrete mode and the mode-conditioned continuous state are simultaneously estimated. The estimated discrete mode provides information on whether abrupt state jumps happen or not (abrupt change detection), and the estimated mode-conditioned continuous state is combined over the two modes to compensate for the effect of the abrupt jumps (filter adjustment).

1.2.2 Applications

The proposed state estimation algorithms have been applied to various information inference problems in three different CPSs in aerospace applications: 1) air traffic control systems, 2) space surveillance systems, and 3) pilot-automation integrated systems.

In air traffic control systems, a key requirement for safe and efficient air traffic flow management in terminal aerospace is accurate knowledge of the aircraft's states (e.g., position, velocity, and flight mode) [30–32]. Using the accurate state information of the aircraft, more efficient airborne spacing with reduced separation thresholds can be achieved, and thereby, air traffic flow near an airport can be effectively managed within its capacity [3]. Since the behavior of an aircraft is governed by hybrid dynamics (i.e., discrete transitions between flight modes and the continuous motion corresponding to a specific flight mode), the corresponding aircraft tracking problem is formulated as the hybrid state estimation problem. In addition, we introduce the event-based sampling framework [33] to the tracking problem as an attempt to reduce data transmission rate required for communication between the aircraft and air traffic controllers. The complexities due to the event-based sampling framework and the hybrid dynamics of the aircraft have been dealt with by the proposed EBHSE algorithm [22]. It has been shown that the proposed algorithm can produce accurate state estimation results with reduced communication channel usage.

As space has become highly congested by many space objects, the space surveillance system has become crucial for space situational awareness (SSA) with the objective of safe operation of space assets [34–36]. The space surveillance system is a good example of a CPS where multiple sensors (ground-based or space-based) are networked and controlled to track multiple physical objects (e.g., satellites or space debris). One of the challenging problems in SSA is to track a maneuvering spacecraft with impulsive burn, where the magnitude and the time of occurrence of impulsive maneuvers are usually unknown a priori [37–39]. To deal with this problem, we first formulate the tracking problem of the maneuvering spacecraft as the state estimation problem of a system with abrupt state jumps (i.e., abrupt changes in the velocity of the spacecraft due to impulsive maneuvers). Then, we apply the newly proposed adaptive estimation algorithm to the problem in order to compute accurate state estimates. Two illustrative scenarios are considered in this problem: 1) tracking of a geostationary satellite performing station-keeping maneuvers and 2) tracking of a spacecraft performing orbital transfers.

Another challenging issue in SSA is that observations can be made only for a small subset of space objects at a given time due to limited observational resources [40]. Therefore, during the time period without measurement updates, statistical properties of the trajectories of space objects need to be accurately predicted (known as uncertainty propagation) [41,42]. In particular, for satellites flying in close proximity, monitoring the uncertainties of neighboring satellites' states is a crucial task since the uncertainty information can be used to compute the collision probability between satellites with the objective of collision avoidance [43, 44]. In this sense, we develop an analytical closed-form solution to the uncertainty propagation problem for the satellite relative motion near general elliptic orbits [45]. Since the proposed analytical solution does not require any numerical integration, it allows satellite onboard computers having low computational capability to perform necessary computations efficiently.

The pilot-automation integrated system is an example of a CPS, in which heterogeneous elements (i.e., human and machine) are interacting in a complex way. One challenging problem in this system is to detect malicious interaction between the pilot and automation [46,47]. Due to the rapid advancement of the flight deck technology, pilot-automation interaction issues have become a core area of focus in today's aviation safety. The complexity of the advanced flight deck leads to new safety concerns such as dysfunctional interaction between the pilot and automation. To detect the interaction issues, we first propose to model the complex behaviors of the pilot and automation using a discrete event system and a hybrid system, respectively. We then propose to infer the intents of the pilot and automation using the intent inference and hybrid state estimation. The interaction issues are then identified by detecting mismatches between the inferred intents of the pilot and automation [48]. It has been shown that real pilot-automation interaction issues can be effectively identified in real-time by the proposed approach.

1.3 Outline of Thesis

The thesis is organized as follows. In Chapter 2, a theoretical framework of the event-based state estimation problem for CPSs is proposed, and the corresponding state estimation algorithm is developed. In Chapter 3, the event-based state estimation framework is extended to CPSs with the hybrid system structure to formulate the event-based hybrid state estimation problem. The EBHSE algorithm is then developed to solve the estimation problem and applied to an aircraft tracking problem in the air traffic control application. In Chapter 4, an adaptive state estimation algorithm for CPSs subject to abrupt state jumps is proposed and applied to tracking problems of impulsively maneuvering spacecraft in the SSA application. Another application in SSA, called the uncertainty propagation problem, is presented for satellite formation flying systems, and the corresponding analytical solution to the problem is derived in Chapter 5. In Chapter 6, a pilot-automation interaction issue detection algorithm is developed to identify anomalous interaction in the pilot-automation integrated systems. A summary and future research directions are discussed in Chapter 7.

2. EVENT-BASED STATE ESTIMATION FOR CYBER-PHYSICAL SYSTEMS

This chapter discusses the state estimation problem for CPSs in which the state needs to be estimated using information generated by the event-based sampling. In Section 2.1, the motivation and literature review for this problem are presented. In Section 2.2, the event-based state estimation problem is mathematically formulated. A numerical estimation algorithm based on the Markov chain approximation is proposed in Section 2.3. In Section 2.4, the proposed algorithm is demonstrated with an illustrative state estimation problem of 'bi-stable' system.

2.1 Background and Motivations

To save communication bandwidth and power resource for data transmission, event-based sampling and control have received much attention in the areas of networked control systems and wireless sensor networks. For example, in many environmental monitoring systems, e.g., greenhouse climate monitoring [49], various event-based sampling strategies have been applied to reduce the bandwidth of communication incurred by a large amount of wireless sensors monitoring CO_2 density. Because of the large number of sensors, time-based sampling methods, where measurements are taken at synchronous time intervals, are not appropriate since they require massive data transmission. To avoid such difficulty, the data sampling can be performed with the lower communication resource by using the event-based sampling strategy in which measurements are generated only when predefined events happen (e.g., 'critical' change in CO_2 density, see Figure 2.1). Another benefit of the event-based sampling is that the event-based sampling framework is appropriate for the observation with binary sensors. Those binary sensors generate binary signals only

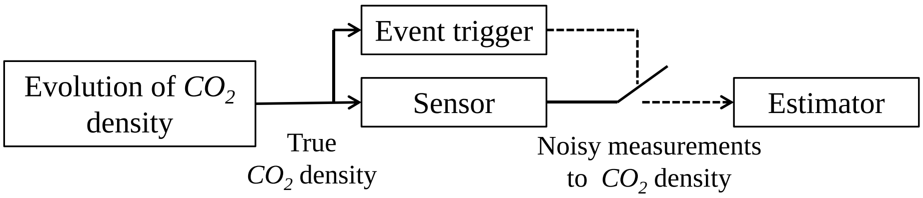

Evolution of CO_2 density
Event trigger
Sensor
Estimator
True CO_2 density
Noisy measurements to CO_2 density

Figure 2.1. Illustration of event-based sampling for CO_2 density monitoring

when triggered by some predefined events. The binary proximity sensor that gives one-bit signal regarding a target's presence or absence is a typical example [50, 51].

Although the event-based sampling is useful, the resulted filtering problem is quite challenging because the observations are sparse and irregular. This implies that the traditional filtering theories which assume the availability of observations at every regular sampling time are not applicable.

To challenge the difficulty in the state estimation with the event-based sampling, several filtering algorithms have been proposed [52–55]. In these previous works, the update of the state probability density is performed only when the outputs of a system are crossing predefined thresholds (i.e., when events happen). This approach, however, has a limitation in the sense that it ignores the information that no event is triggered between the consecutive threshold crossings. No threshold crossing implies that the outputs of a system must be within the upper and lower threshold bounds. Recently, there have been efforts to use the ignored information on the update of state estimates [33, 56–58]. In [33], a state estimation problem with the 'Send-on-Delta' sampling method was considered. It was shown that the estimation performance can be improved by using the implicit information that no output is generated between the two consecutive threshold crossings. References [56] and [57] proposed an event-based filtering algorithm which is based on the Gaussian sum filter. They showed that the divergence of error-covariance can be prevented by using the implicit information. However, all the work mentioned above [33, 56, 57] were based on the time-invariant linear dynamics and linear measurement equations in their formulations, which implies that they need to be extended to deal with general nonlinear cases, which could be nontrivial. In [58], a nonlinear event-based state estimation algorithm has been proposed which is based on particle filtering. Although the algorithm in [58] can deal with nonlinear dynamics, its event-based sampling model is limited to the 'Send-on-Delta' method.

Another concern is that the event-based state estimation problem has been formulated through different approaches such as level-crossings [59], measurement quantization [60], 'Send-on-Delta' [61], estimation with discrete sensors [62], and so forth. Because of the different formulations, it is not easy to apply the solution developed for a specific problem to other problems. Thus, it is important to develop a general framework for a large class of event-based state estimation problems with the corresponding general solution.

The contributions in this chapter are twofold. First, a general framework for the continuous-time nonlinear event-based state estimation problem is developed. The evolution of the continuous-time nonlinear dynamics is modeled by Stochastic Differential Equations (SDEs). Based on the dynamics, an event-based filtering problem is mathematically formulated such that the probability density function (pdf) of the SDE's state is computed through two steps: 1) the propagation of the pdf between consecutive measurements (i.e., with no event occurring); and 2) the update of the pdf when an event happens and corresponding new measurement arrives. Secondly, a numerical algorithm based on the Markov chain approximation method [19–21] is proposed to solve the filtering problem. The Markov chain's state space is constructed by discretizing the original state space. Then, with the appropriate choice of the transition probability of the Markov chain, the evolution of the SDE is approximated by that of the Markov chain so that the event-based state estimation of the SDE is solved by estimating the Markov chain's state. Unlike the previous research mentioned above, the proposed algorithm is general so that it can be applied to event-based state estimation problems with time-varying nonlinear dynamics and measurement models.

2.2 Problem Formulation for Event-Based State Estimation

In this section, the continuous-time nonlinear event-based state estimation problem is formulated. The nonlinear stochastic dynamics is represented using SDE and the measurements are given using an event-based sampling model. Filtering equations are then derived to compute the pdf of the SDE's state given measurements generated by the event-based sampling model.

2.2.1 Notations

In this chapter, we denote the n-dimensional Euclidean space by $\mathbb{R}^n$, the integers by $\mathbb{Z}$, the Euclidean norm by $\|\cdot\|$, the expectation of a random variable by $\mathbb{E}[\cdot]$, the probability of an event by $\Pr\{\cdot\}$, and the normalized probability density/mass function (pdf/pmf) by $p(\cdot)$ ($\tilde{p}(\cdot)$ for the unnormalized pdf/pmf).

2.2.2 Nonlinear Stochastic System Model

Let $\mathbf{x}(t) \in \mathbb{R}^{n^1}$ and $\mathbf{v}(t) \in \mathbb{R}^{n^2}$ be the state of the system and the noise process, respectively. The augmented state $\mathbf{X}(t)$ is then defined as $\mathbf{X}(t) := [\mathbf{x}(t)\ \mathbf{v}(t)]^T \in \mathbb{R}^n$, where $n = n^1 + n^2$ (note that the augmented state simplifies the description of the event-based observation model presented in the following section). The following SDE is given to describe the evolution of the augmented state as [18]:

$$d\mathbf{X}(t) = \mathbf{a}(\mathbf{X}(t), t)dt + \mathbf{b}(\mathbf{X}(t), t)d\mathbf{W}(t) \tag{2.1}$$

where $\mathbf{a} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}^n$ and $\mathbf{b} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}^{n \times n}$ are the drift term and volatility term, respectively, and $\mathbf{W}(t)$ is the standard n-dimensional Brownian motion. $\mathbf{a}$ and $\mathbf{b}$ are assumed to be continuous functions satisfying the Lipschitz conditions so that the solution to (2.1) exists and is unique.

2.2.3 Event-Based Observation Model

Define $\mathbf{y}(t)$ as a function of the augmented state $\mathbf{X}(t)$ as:

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{X}(t)) \tag{2.2}$$

where $\mathbf{y}(t) \in \mathbb{R}^l$ is a l-dimensional measurement vector and $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^l$ is a (piecewise) smooth bounded nonlinear function. Unlike the time-based sampling where the measurement $\mathbf{y}$ is transmitted to the estimator at every predefined time step, the event-based sampling generates the measurement only at the moment when some events happen. Those events are represented as conditions that $\mathbf{y}$ needs to satisfy in order to generate the measurement [56, 57]. Define τ_i as the i^{th} event epoch such that $i = 0, 1, 2, \ldots$ and $\tau_0 = t_0$, $\tau_i < \tau_j$, $i < j$, where t_0 is the initial time. During an interval $T_i = (\tau_i, \tau_{i+1})$, the trajectory of $\mathbf{y}(t)$ is allowed to reside only in the region $\mathcal{E}_i (\subset \mathbb{R}^l)$, which is defined by

$$\mathcal{E}_i = \{\mathbf{y} | \mathbf{y} \in \mathbb{R}^l, \mathbf{F}_i(\mathbf{y}) < \mathbf{0}\} \tag{2.3}$$

where $\mathbf{F}_i := [F_{i,1}\ F_{i,2}\ \ldots\ F_{i,n_i}]^T : \mathbb{R}^l \to \mathbb{R}^{n_i}$ is a (piecewise) smooth nonlinear vector function whose size n_i is dependent on each event instant, and $\mathbf{0}$ is $n_i \times 1$ zero vector. Each boundary $\partial\mathcal{E}_{i,k} := \{\mathbf{y} | F_{i,k}(\mathbf{y}) = 0, \mathbf{F}_i(\mathbf{y}) \leq \mathbf{0}\}$ represents an independent 'event' which is triggered when $\mathbf{y}(t)$ crosses to leave $\mathcal{E}_i$. It is assumed that the boundaries are exclusive to each other so that, at each event occurring time, only one event can occur, i.e., output $\mathbf{y}(t)$ can cross only one boundary at a time. The next event occurring time τ_{i+1} is then defined such that $\mathbf{y}(\tau_{i+1}^-) \in \mathcal{E}_i$ and $\mathbf{y}(\tau_{i+1}) \notin \mathcal{E}_i$, where τ_{i+1}^- is the left limit of τ_{i+1}. At time τ_{i+1} (assume that $\partial\mathcal{E}_{i,k}$ is triggered, i.e., $\mathbf{y}$(t) is crossing $\partial\mathcal{E}_{i,k}$), the information $F_{i,k}(\mathbf{y}(\tau_{i+1})) = 0$ and $\mathbf{F}_i(\mathbf{y}(\tau_{i+1})) \leq \mathbf{0}$ is generated and sent to an estimator. Note that $\mathbf{F}_i$ and n_i are allowed to vary at each event instant and its explicit expression is determined by the events predefined for specific problems. For example, consider the one-dimensional 'Send-on-Delta' example illustrated in Figure 2.2. Assume that, at time τ_i, the one-dimensional output $y(\tau_i) = y^i$ is generated.

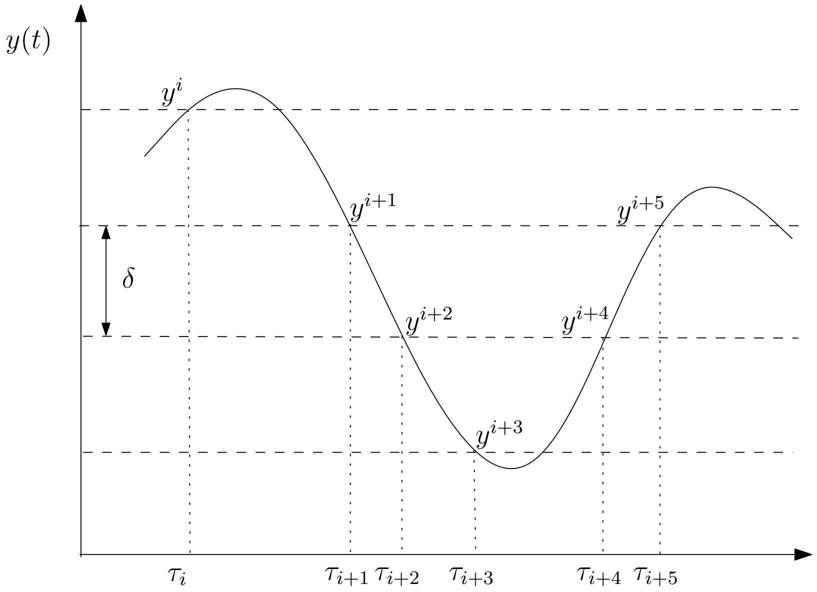


Figure 2.2. Send-on-Delta sampling

Then, the region $\mathcal{E}_i$ that $y(t)$ can move without triggering the next event is derived by defining $\mathbf{F}_i$ as:

$$\begin{aligned} F_{i,1} &= y(t) - y^i - \delta < 0 \\ F_{i,2} &= y^i - y(t) - \delta < 0 \end{aligned} \tag{2.4}$$

where δ is a predefined level. In this example, $y(t)$ crosses $F_{i,2}$ at time τ_{i+1}, and therefore, the information $F_{i,2}(y(\tau_{i+1})) = 0$ and $\mathbf{F}_i(y(\tau_{i+1})) \leq \mathbf{0}$ (i.e., $y^{i+1} := y(\tau_{i+1}) = y^i - \delta$) is sent to the estimator. Note that the size of δ can vary for each event instant although the size is assumed to be fixed in Figure 2.2.

2.2.4 Event-Based State Estimation Problem

The information that the estimator receives up to the current time t is a series of the event occurring times and the corresponding event information as:

$$\{\tau_{i+1}, F_{i,k_i}(\mathbf{y}(\tau_{i+1})) = 0, \mathbf{F}_i(\mathbf{y}(\tau_{i+1})) \leq \mathbf{0}\} \text{ with } i = 0, \ldots, N \tag{2.5}$$

such that $\tau_{N+1} \leq t < \tau_{N+2}$, where k_i refers to the index of the boundary ($k_i \in \{1, 2, \ldots, n_i\}$) that is triggered at τ_{i+1}. Based on the fact that the information is generated by the event-based sampling, the information (2.5) can be interpreted as:

$$\begin{cases} \mathbf{y}(s) \in \mathcal{E}_i, & \text{if } \tau_i < s < \tau_{i+1} \\ F_{i,k_i}(\mathbf{y}(s)) = 0 \text{ and } \mathbf{F}_i(\mathbf{y}(s)) \leq \mathbf{0}, & \text{if } s = \tau_{i+1} \end{cases} \tag{2.6}$$

for $s \in [t_0, t]$. Note that the information of $F_{i,k_i}(\mathbf{y}(\tau_{i+1})) = 0$ and $\mathbf{F}_i(\mathbf{y}(\tau_{i+1})) \leq \mathbf{0}$ is equivalent to $\mathbf{y}(\tau_{i+1}) \in \partial\mathcal{E}_{i,k_i}$. Define $Y(s)$ as:

$$Y(s) = \begin{cases} \mathcal{E}_i, & \text{if } \tau_i < s < \tau_{i+1} \\ \partial\mathcal{E}_{i,k_i}, & \text{if } s = \tau_{i+1} \end{cases} \tag{2.7}$$

The state estimation problem is then defined to compute the pdf $p(\cdot)$ as:

$$p(\mathbf{X}(t)|\mathbf{y}(s) \in Y(s),\ t_0 \leq s \leq t) \tag{2.8}$$

with the given initial pdf $p_{t_0}(\mathbf{X})$ (the condition for $p_{t_0}(\mathbf{X})$ will be presented in the following subsection). For brevity, denote (2.8) as $p_t(\mathbf{X}(t)|Y^t)$. From this, we define the estimate of $\mathbf{X}(t)$ conditioned on the information up to t as:

$$\mathbf{X}_{t|t} := \mathbb{E}[\mathbf{X}(t)|Y^t] \tag{2.9}$$

Note that the difference between the conventional estimation techniques and the event-based state estimation is that the former updates the pdf only when the new measurement data arrives (i.e., when $t = \tau_i$, $i = 1, 2, \ldots$) while the latter uses not only the measurement data but also the information generated by 'no measurements between the two consecutive measurements' (i.e., during $t \in (\tau_i, \tau_{i+1})$).

2.2.5 Solution to the Event-Based State Estimation Problem

Define $\forall i$, $E_i := \{\mathbf{X}|\mathbf{h}(\mathbf{X}) \in \mathcal{E}_i\}$ and $\partial E_{i,k_i} := \{\mathbf{X}|\mathbf{h}(\mathbf{X}) \in \partial\mathcal{E}_{i,k_i}\}$. The initial pdf $p_{t_0}(\mathbf{X})$ is given on E_0 (i.e., $p_{t_0}(\mathbf{X}) = 0$ if $\mathbf{h}(\mathbf{X}) \notin \mathcal{E}_0$). This is reasonable because $\mathbf{y}(= \mathbf{h}(\mathbf{X}))$ should remain in $\mathcal{E}_0$ until the first event occurring time $\tau_1(> t_0)$. The computation of (2.8) is then performed recursively by two steps. Assume that the pdf of the state at the event occurring time τ_i, $p_{\tau_i}(\mathbf{X}|Y^{\tau_i})$ is available from the last iteration of the algorithm.

Propagation

$\forall t \in (\tau_i, \tau_{i+1})$, the available information is that the output $\mathbf{y}(t)$ is within $\mathcal{E}_i$. Thus, the pdf $p_t(\mathbf{X}|Y^t)$ is computed by propagating $p_{\tau_i}(\mathbf{X}(\tau_i)|Y^{\tau_i})$ within the domain E_i (i.e., $\bigcup_{k=1}^{n_i} \partial E_{i,k}$ plays as the absorbing barrier). Define the generator $\mathcal{L}$ and its adjoint $\mathcal{L}^*$ associated with the SDE (2.1) for a bounded and twice-differentiable function $f : \mathbf{X} \to \mathbb{R}$ as:

$$\mathcal{L}f(\mathbf{X}) = \sum_{i=1}^{n} a_i(\mathbf{X}, t)\frac{\partial f}{\partial X_i} + \frac{1}{2}\sum_{i,j=1}^{n}(\mathbf{b}(\mathbf{X}, t)\mathbf{b}(\mathbf{X}, t)^T)_{ij}\frac{\partial^2 f}{\partial X_i \partial X_j} \tag{2.10}$$

$$\mathcal{L}^* f(\mathbf{X}) = -\sum_{i=1}^{n} \frac{\partial[a_i(\mathbf{X}, t)f(\mathbf{X})]}{\partial X_i} + \frac{1}{2}\sum_{i,j=1}^{n}\frac{\partial^2[b_{ij}(\mathbf{X}, t)f(\mathbf{X})]}{\partial X_i \partial X_j} \tag{2.11}$$

where X_i is the i^{th} element of the state vector $\mathbf{X}$, a_i is the i^{th} element of the vector-valued function $\mathbf{a}$, and b_{ij} is the element in i^{th} row and j^{th} column of the matrix-valued function $\mathbf{b}$. The propagation of the pdf of the state is then performed by solving the Fokker-Plank equation for $t > \tau_i$ as:

$$\frac{\partial}{\partial t}p_t(\mathbf{X}) = \mathcal{L}^* p_t(\mathbf{X}) \tag{2.12}$$

with the initial condition

$$p_{\tau_i}(\mathbf{X}) = p_{\tau_i}(\mathbf{X}(\tau_i)|Y^{\tau_i})$$

and the boundary condition

$$p_t(\mathbf{X}) = 0 \text{ for } \mathbf{X} \in \bigcup_{k=1}^{n_i} \partial E_{i,k}$$

Updating

At time $t = \tau_{i+1}$, when the next event occurs, the propagated pdf is corrected by the new information that the output $\mathbf{y}(t)$ leaves $\mathcal{E}_i$ through $\partial\mathcal{E}_{i,k_i}$. That is, the pdf propagated in E_i is reduced onto $\partial E_{i,k_i}$. For the reduction, we consider the net flow of probability across the surface $\partial E_{i,k_i}$, which is given by a surface integral as [63]:

$$\text{Net flow of probability} = \int_{\partial E_{i,k_i}} \mathbf{n}(\mathbf{X}) \cdot \mathbf{J}(\mathbf{X}, t) d\mathbf{X} \tag{2.13}$$

where $\mathbf{n}$ is the unit normal vector to $\partial E_{i,k_i}$ outward from E_i and $\mathbf{J} := [J_1\ J_2\ \ldots\ J_n]^T$ is a probability current vector defined such that

$$J_i(\mathbf{X}, t) = a_i(\mathbf{X}, t)p_t(\mathbf{X}) - \frac{1}{2}\sum_j \frac{\partial}{\partial X_j} b_{ij}(\mathbf{X}, t)p_t(\mathbf{X}) \tag{2.14}$$

Note that the occurrence of an event implies that the probability is flowing out of E_i through the surface $\partial E_{i,k_i}$. In this sense, the probability density of $\mathbf{X}$ on $\partial E_{i,k_i}$ at $t = \tau_{i+1}$ is proportional to the net probability flow across $\mathbf{X}$. Therefore, the pdf $p_{\tau_{i+1}}(\mathbf{X}|Y^{\tau_{i+1}})$ can be computed as:

$$p_{\tau_{i+1}}(\mathbf{X}|Y^{\tau_{i+1}}) = \begin{cases} \frac{1}{c}\mathbf{n}(\mathbf{X}) \cdot \mathbf{J}(\mathbf{X}, \tau_{i+1}), & \text{for } \mathbf{X} \in \partial E_{i,k_i} \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

where $c = \int_{\partial E_{i,k_i}} \mathbf{n}(\mathbf{X})\cdot\mathbf{J}(\mathbf{X}, \tau_{i+1})d\mathbf{X}$ is a normalizing constant. The pdf $p_{\tau_{i+1}}(\mathbf{X}|Y^{\tau_{i+1}})$ is then propagated in E_{i+1} until the next event occurring time τ_{i+2}. The iterative state estimation steps are summarized in Table 2.1.

2.3 Numerical Algorithm for Event-Based State Estimation

To compute the pdf (2.8), it is necessary to solve (2.12) and (2.15). Because their analytical solutions, in general, are hard to obtain, they need to be solved numerically. In this section, a numerical algorithm based on Markov chain approximation is presented.

2.3.1 Discretization of the State Space

Assume that $U \subset \mathbb{R}^n$ is the bounded domain where the system (2.1) evolves. To construct the Markov Chain, the state space of the Markov Chain needs to be defined

Table 2.1. Filtering equations for event-based state estimation

```
1.Initialization
  Set an initial pdf $p_{t_0}(\mathbf{X})$ on $E_0$
2.Iteration
  for $(i = 0, ..., N)$
    while $t \in (\tau_i, \tau_{i+1})$ (Propagation)
      Compute the pdf $p_t(\mathbf{X})$ for $\mathbf{X} \in E_i$ using (2.12),
      with the initial condition: $p_{\tau_i}(\mathbf{X}) = p_{\tau_i}(\mathbf{X}(\tau_i)|Y^{\tau_i})$
      and the boundary condition: $p_t(\mathbf{X}) = 0$ for $\mathbf{X} \in \bigcup_{k=1}^{n_i} \partial E_{i,k}$
      if $t = \tau_{i+1}$ (Updating)
        Compute the pdf $p_{\tau_{i+1}}(\mathbf{X}|Y^{\tau_{i+1}})$ on $\partial E_{i,k_i}$ using (2.15)
      end
    end
    $i = i + 1$
  end
```

by discretizing the original state space U. Let $\mathcal{Q} \subset \mathbb{Z}^n$ be the Markov Chain's state space and define $\epsilon = [\epsilon_1 \ \epsilon_2 \ \dots \ \epsilon_n]^T$ as the grid size. Then, for all the points in the Markov Chain's state space, $\forall \mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T \in \mathcal{Q}$, its corresponding coordinate in U is mapped by

$$\mathbf{X}(\mathbf{q}) = [\epsilon_1 q_1 \ \epsilon_2 q_2 \ \dots \ \epsilon_n q_n]^T \tag{2.16}$$

The grid $G(\mathbf{q}) \subset \mathbb{R}^n$ is defined as an area centered at $\mathbf{X}(\mathbf{q})$ as:

$$G(\mathbf{q}) := \left\{ \mathbf{X} | \|X_i - X_i(\mathbf{q})\| \leq \frac{\epsilon_i}{2}, \forall i \right\} \tag{2.17}$$

where X_i is i^{th} element of $\mathbf{X}$ for all $i = 1, \dots, n$. $\forall \mathbf{q} \in \mathcal{Q}$, we define its neighborhood $N(\mathbf{q}) \subset \mathcal{Q}$ as:

$$N(\mathbf{q}) := \{\mathbf{q}' | \mathbf{q}' - \mathbf{q} \in \{-1, 0, 1\}^n, \text{ and } \mathbf{q}' \neq \mathbf{q}\} \tag{2.18}$$

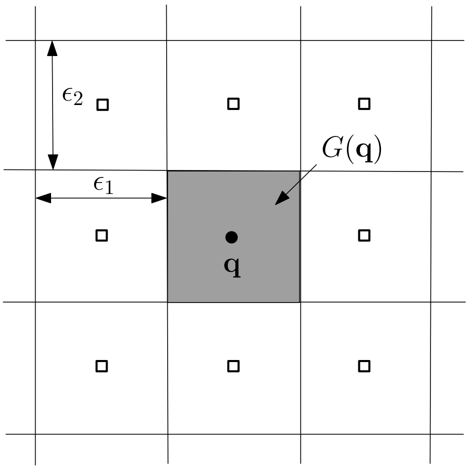


Figure 2.3. Illustration of a grid and the neighborhood of $\mathbf{q}$ in U

where $\{-1,0,1\}^n$ is n-tuple whose element takes a value in a set $\{-1,0,1\}$. The above definitions are illustrated in Figure 2.3 for a two-dimensional case ($n=2$), in which the empty squares represent the neighborhood of $\mathbf{q}$ and the shaded region is the grid $G(\mathbf{q})$. By using the above definitions, E_i, $\partial E_{i,k}$, and $\partial E_i := \bigcup_{k=1}^{n_i} \partial E_{i,k}$ are approximated in $\mathcal{Q}$ as:

$$\begin{aligned}
\mathcal{Q}_{E_i} :=& \{\mathbf{q}|\mathbf{X}(\mathbf{q}) \in E_i\} \\
\mathcal{Q}_{\partial E_i} :=& \{\mathbf{q}|\mathbf{X}(\mathbf{q}) \notin E_i, \mathbf{q} \in N(\mathbf{q}') \text{ for some } \mathbf{q}' \\
& \text{such that } \mathbf{X}(\mathbf{q}') \in E_i\} \\
\mathcal{Q}_{\partial E_{i,k}} :=& \{\mathbf{q}|\mathbf{q} \in \mathcal{Q}_{\partial E_i}, F_{i,k} \circ \mathbf{h}(\mathbf{X}(\mathbf{q})) \geq 0\}
\end{aligned} \tag{2.19}$$

Figure 2.4 illustrates the above definitions, where the shaded regions represent $\mathcal{Q}_{\partial E_i}$, which is an approximation of ∂E_i (black bold line).

2.3.2 Approximation of the Initial Pdf on the Markov State Space

To begin with, the initial pdf $p_{t_0}(\mathbf{X})$ needs to be approximated to $p_{t_0}(\mathbf{q})$ on the Markov Chain's state space $\mathcal{Q}$. Because the state space $\mathcal{Q}$ is a discretized space, the

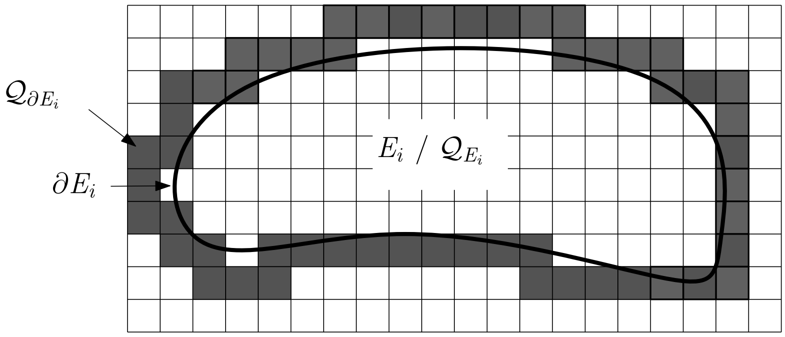


Figure 2.4. Illustration of discretization of the state space

pdf $p_{t_0}(\mathbf{q})$ is represented as a pmf. Assume that the initial pdf is given by $p_{t_0}(\mathbf{X})$ such that

$$\begin{aligned} p_{t_0}(\mathbf{X}) \geq 0, \ \forall \mathbf{X} \in E_0 \\ \int_{E_0} p_{t_0}(\mathbf{X}) d\mathbf{X} = 1 \end{aligned} \tag{2.20}$$

The pmf $p_{t_0}(\mathbf{q})$ approximating the pdf $p_{t_0}(\mathbf{X})$ is then computed as:

$$p_{t_0}(\mathbf{q}) = \int_{G(\mathbf{q})} p_{t_0}(\mathbf{X}) d\mathbf{X} \tag{2.21}$$

2.3.3 Construction of the Approximating Markov Chain

Upon defining the Markov state space through discretization of the original state space in Section 2.3.1, a Markov chain is constructed to approximate the evolution of the original system (2.1). Let $\{\mathbf{Q}_j\}$ be the Markov chain defined on $\mathcal{Q}$, where $j \in \mathbb{N}$ is the index of the Markov chain's time step. Let Δ be the time interval of one transition step of the Markov chain such that $\Delta = \lambda \|\epsilon\|^2$ for some fixed positive constant λ. To approximate the evolution of the pdf (2.8), the transition probabilities of the Markov chain need to be designed in such a way that its distribution converges to the distribution of (2.8), as the grid size ϵ and time step Δ approach to zero.

Propagation

For all j such that $\tau_i + j\Delta \in (\tau_i, \tau_{i+1})$, the transition probabilities of the Markov chain are designed to have the following property:

$$\Pr\{\mathbf{Q}_{j+1} = \mathbf{q}' | \mathbf{Q}_j = \mathbf{q}\} = \begin{cases} \pi(\mathbf{q}', \mathbf{q}), & \text{if } \mathbf{q}' \in N(\mathbf{q}) \cup \mathbf{q}, \forall \mathbf{q} \in \mathcal{Q}_{E_i} \\ 0, & \text{otherwise} \end{cases} \tag{2.22}$$

where $\pi(\cdot, \cdot)$ is a parameter to be designed. Equation (2.22) shows the fact that, at each time step, the state of the Markov chain is allowed to jump from $\mathbf{q}$ to either its neighborhood $N(\mathbf{q})$ or itself. Note that a legitimate Markov chain can be constructed with transition probabilities that allow its state to jump from one state to any other state, if only the resultant Markov chain weakly converges to the original system. In this paper, we designed the transition probabilities such that the Markov chain state can only jump from a state to the state itself or its neighborhood. In this setting, the construction of the Markov chain is then completed by designing the appropriate $\pi(\cdot, \cdot)$ in (2.22). The following theorem provides the weak convergence conditions that $\pi(\cdot, \cdot)$ must satisfy to ensure that the distribution of $\{\mathbf{Q}_j\}$ is consistent with the distribution of $\mathbf{X}(t)$. An example of designing the appropriate π is presented in the Appendix.

Theorem 2.3.1 *Assume that as ϵ, $\Delta \to 0$, the Markov chain $\{\mathbf{Q}_j\}$ with the transition probabilities given by (2.22) satisfies the following properties:*

$$\begin{aligned} &\frac{1}{\Delta}\mathbb{E}[\mathbf{X}(\mathbf{Q}_{j+1}) - \mathbf{X}(\mathbf{Q}_j) | \mathbf{Q}_j = \mathbf{q}] \to \mathbf{a}(\mathbf{X}(\mathbf{q}), j\Delta) \\ &\frac{1}{\Delta}\mathbb{E}[(\mathbf{X}(\mathbf{Q}_{j+1}) - \mathbf{X}(\mathbf{Q}_j))(\mathbf{X}(\mathbf{Q}_{j+1}) - \mathbf{X}(\mathbf{Q}_j))^T | \mathbf{Q}_j = \mathbf{q}] \\ &\qquad\qquad \to \mathbf{b}(\mathbf{X}(\mathbf{q}), j\Delta)\mathbf{b}(\mathbf{X}(\mathbf{q}), j\Delta)^T \end{aligned} \tag{2.23}$$

Then, $\forall j$, the distribution of $\{\mathbf{Q}_j\}$ defined on $\mathcal{Q}$ converges to the distribution of $\mathbf{X}(t)$ defined on U.

Proof To prove the theorem, we need to show that for a bounded and twice-differentiable function $f : \mathbf{X} \to \mathbb{R}$,

$$\frac{\mathbb{E}[f(\mathbf{X}(\mathbf{Q}_{j+1})) - f(\mathbf{X}(\mathbf{Q}_j)) | \mathbf{Q}_j = \mathbf{q}]}{\Delta} \to \mathcal{L}f \tag{2.24}$$

as $\Delta, \|\epsilon\| \to 0$. By Taylor's theorem,

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{X}(\mathbf{Q}_{j+1})) - f(\mathbf{X}(\mathbf{Q}_j))|\mathbf{Q}_j = \mathbf{q}] \\
&= \sum_{i=1}^{n} \mathbb{E}[\mathbf{X}(\mathbf{Q}_{j+1}) - \mathbf{X}(\mathbf{Q}_j)|\mathbf{Q}_j = \mathbf{q}]_i \left.\frac{\partial f}{\partial X_i}\right|_{\mathbf{X}(\mathbf{q})} \\
&+ \frac{1}{2}\sum_{i,l=1}^{n} \{\mathbb{E}[(\mathbf{X}(\mathbf{Q}_{j+1}) - \mathbf{X}(\mathbf{Q}_j))(\mathbf{X}(\mathbf{Q}_{j+1}) - \mathbf{X}(\mathbf{Q}_j))^T|\mathbf{Q}_j = \mathbf{q}]_{il} \\
&\times \left.\frac{\partial^2 f}{\partial X_i \partial X_l}\right|_{\mathbf{X}(\mathbf{q})}\Bigg\} + H.O.T
\end{aligned} \tag{2.25}$$

Note that the $H.O.T$ has the order greater than $\|\epsilon\|^3$ and the fact that $\Delta = \lambda\|\epsilon\|^2$, which implies that

$$\lim_{\Delta,\|\epsilon\|\to 0} \frac{H.O.T}{\Delta} = 0 \tag{2.26}$$

Hence, from the properties (2.23) and (2.26),

$$\begin{aligned}
&\lim_{\Delta,\|\epsilon\|\to 0} \frac{\mathbb{E}[f(\mathbf{X}(\mathbf{Q}_{j+1})) - f(\mathbf{X}(\mathbf{Q}_j))|\mathbf{Q}_j = \mathbf{q}]}{\Delta} \\
&= \sum_{i=1}^{n} \mathbf{a}(\mathbf{X})_i \left.\frac{\partial f}{\partial X_i}\right|_{\mathbf{X}(\mathbf{q})} + \frac{1}{2}\sum_{i,l=1}^{n} (\mathbf{b}(\mathbf{X})\mathbf{b}(\mathbf{X})^T)_{il} \left.\frac{\partial^2 f}{\partial X_i \partial X_l}\right|_{\mathbf{X}(\mathbf{q})} \\
&= \mathcal{L}f
\end{aligned} \tag{2.27}$$

■

Assume that the pmf $p_{\tau_i}(\mathbf{q}|Y^{\tau_i})$ is available at τ_i. The propagation of the pmf is then performed iteratively by the following three steps: 1) to solve the Chapman-Kolmogorov forward equation which is the discretized version of the Fokker-Plank equation, 2) to zero out the probability mass at the absorbing barrier (i.e., $\mathcal{Q}_{\partial E_i}$), and 3) to re-normalize the rest of the distribution. In the first step, for $t = \tau_i + j\Delta \in (\tau_i, \tau_{i+1})$, the pmf $p_t(\mathbf{q}|Y^t)$ is propagated using the Chapman-Kolmogorov forward equation as:

$$\begin{aligned}
p_{\tau_i+(j+1)\Delta}(\mathbf{q}|Y^{\tau_i+j\Delta}) = \sum_{\mathbf{q}'\in\mathcal{Q}_{E_i}} &[\Pr\{\mathbf{Q}_{j+1} = \mathbf{q}|\mathbf{Q}_j = \mathbf{q}'\} \\
&\times p_{\tau_i+j\Delta}(\mathbf{q}'|Y^{\tau_i+j\Delta})]
\end{aligned} \tag{2.28}$$

In the second step, $\tilde{p}_{\tau_i+(j+1)\Delta}(\mathbf{q}|Y^{\tau_i+(j+1)\Delta})$ is obtained by

$$\tilde{p}_{\tau_i+(j+1)\Delta}(\mathbf{q}|Y^{\tau_i+(j+1)\Delta}) = \begin{cases} p_{\tau_i+(j+1)\Delta}(\mathbf{q}|Y^{\tau_i+j\Delta}), & \text{if } \mathbf{q} \in \mathcal{Q}_{E_i} \\ 0, & \text{otherwise} \end{cases} \tag{2.29}$$

Note that in (2.29) $\tilde{p}$ represents the unnormalized pmf. To zero out the probability mass at the absorbing barrier ($\mathcal{Q}_{\partial E_i}$) is reasonable since the state is not allowed to evolve across the boundary ∂E_i until the next event occurring time τ_{i+1}. The unnormalized pmf $\tilde{p}_{\tau_i+(j+1)\Delta}(\mathbf{q}|Y^{\tau_i+(j+1)\Delta})$ is then normalized in the last step to find $p_{\tau_i+(j+1)\Delta}(\mathbf{q}|Y^{\tau_i+(j+1)\Delta})$.

Updating

At the next event occurring time τ_{i+1}, the propagated pmf $p_{\tau_{i+1}^-}(\mathbf{q}|Y^{\tau_{i+1}^-})$ in $\mathcal{Q}_{E_i}$ is reduced to $\mathcal{Q}_{\partial E_{i,k_i}}$ using the new information that the output $\mathbf{y}(t)$ leaves $\mathcal{E}_i$ through $\partial\mathcal{E}_{i,k_i}$. Equation (2.15) is used to compute the updated pmf $p_{\tau_{i+1}}(\mathbf{q}|Y^{\tau_{i+1}})$. The probability flow vector at $\mathbf{q}$ for any time $t = \tau_i + j\Delta \in (\tau_i, \tau_{i+1})$ is approximated as:

$$\begin{aligned} \mathbf{J}(\mathbf{q}, t) \approx & \sum_{\mathbf{q}' \in N(\mathbf{q})} \mathbf{n}_{\mathbf{q}'\mathbf{q}}[\Pr\{\mathbf{Q}_{j+1} = \mathbf{q}'|\mathbf{Q}_j = \mathbf{q}\} p_{\tau_i+j\Delta}(\mathbf{q}|Y^{\tau_i+j\Delta})] \\ & - \sum_{\mathbf{q}' \in N(\mathbf{q})} \mathbf{n}_{\mathbf{q}'\mathbf{q}}[\Pr\{\mathbf{Q}_{j+1} = \mathbf{q}|\mathbf{Q}_j = \mathbf{q}'\} p_{\tau_i+j\Delta}(\mathbf{q}'|Y^{\tau_i+j\Delta})] \end{aligned} \tag{2.30}$$

where $\mathbf{n}_{\mathbf{q}'\mathbf{q}}$ is a unit vector which is parallel to $\mathbf{X}(\mathbf{q}') - \mathbf{X}(\mathbf{q})$. Using (2.15), the updated pmf $p_{\tau_{i+1}}(\mathbf{q}|Y^{\tau_{i+1}})$ is obtained as:

$$p_{\tau_{i+1}}(\mathbf{q}|Y^{\tau_{i+1}}) \propto \mathbf{n}(\mathbf{q}) \cdot \mathbf{J}(\mathbf{q}, \tau_i + j^*\Delta), \ \forall \mathbf{q} \in \mathcal{Q}_{\partial E_{i,k_i}} \tag{2.31}$$

where $j^* = \sup\{j|\tau_i + j\Delta \in (\tau_i, \tau_{i+1})\}$. Note that the unit normal vector $\mathbf{n}(\mathbf{q})$ can be approximated analytically by

$$\mathbf{n}(\mathbf{q}) \approx \frac{1}{c_1} \nabla(F_{i,k_i} \circ \mathbf{h})|_{\mathbf{X}=\mathbf{X}(\mathbf{q})} \tag{2.32}$$

where c_1 is a normalizing constant. As the grid size becomes smaller, $\mathbf{X}(\mathbf{q})$ will be closer to $\partial E_{i,k_i}$, and therefore, the normal vector at $\mathbf{X}(\mathbf{q})$ computed by (2.32)

Table 2.2. Numerical algorithm for event-based state estimation

```
1.Initialization
  Construct the pmf $p_{t_0}(\mathbf{q})$ on $\mathcal{Q}$ by discretizing the initial pdf
  $p_{t_0}(\mathbf{X})$, using (2.21).
2.Iteration
  for $(i = 0, ..., N)$
    $j = 0$
    while $(\tau_i + j\Delta < \tau_{i+1})$ (Propagation)
      Compute the pmf $p_{\tau_i+(j+1)\Delta}(\mathbf{q}|Y^{\tau_i+j\Delta})$ using (2.28) for $\forall\mathbf{q} \in \mathcal{Q}_{E_i}$
      if $(\tau_i + (j+1)\Delta < \tau_{i+1})$
        Zero out the probability mass at the absorbing barrier $\mathcal{Q}_{\partial E_i}$
        using (2.29) and normalize the resulted pmf
      end
      $j = j + 1$
    end
    (Updating)
    Correct the pmf $p_{\tau_{i+1}}(\mathbf{q}|Y^{\tau_{i+1}})$ using (2.31) for $\forall\mathbf{q} \in \mathcal{Q}_{\partial E_{i,k_i}}$
    $i = i + 1$
  end
```

will approach to true normal vector at $\partial E_{i,k_i}$. The numerical algorithm proposed in this section is summarized in Table 2.2. Given the pmf $p_t(\mathbf{q}|Y^t)$ computed from the propagation or update steps, the state estimate of $\mathbf{X}(t)$ is approximated as:

$$\mathbf{X}_{t|t} := \mathbb{E}[\mathbf{X}(t)|Y^t] \approx \mathbb{E}[\mathbf{X}(\mathbf{q})|Y^s] = \sum_{\mathbf{q}\in\mathcal{Q}} \mathbf{X}(\mathbf{q})p_s(\mathbf{q}|Y^s) \tag{2.33}$$

In (2.33), $s = t_0 + j\Delta$, where $j = \sup\{j|s \leq t\}$.

The computational complexity of the proposed numerical approach grows as M^d, where M is the number of grid points in each dimension and d is the dimension of the state space. This complexity is similar to other grid-based methods [64, 65] developed to find numerical solutions of the Fokker- Plank equation. In the methods, the number of grid points is unchanged as the pdf is propagated in the given whole domain. However, in our proposed approach, the number of grid points varies and are always are less than the number of grid points for the given whole domain, because the pdf needs to be propagated only in the subspace whose size and shape are changing according to the predefined events. By reducing the number of grid points, we can improve the computational time. In addition, the computational complexity can be further alleviated by using an adaptive grid [66,67] rather than a fixed grid, which will be included in our future work. Recently, it has been shown that the Markov chain approximation is comparable to or even better than the well-known particle filters in computational time and accuracy for approximating the evolution of the pdf [68]. This implies the practicality of the proposed algorithm in terms of the computational complexity.

2.4 Numerical Simulation

In this section, the proposed algorithm for the event-based state estimation is illustrated with an estimation of 'bi-stable' (also known as 'double-well') system, which is a well-known nonlinear stochastic model in biological applications (e.g., the behavior of membrane). Consider the one-dimensional bi-stable system described by the following SDE [69]:

$$dx = \alpha x(\beta - x^2)dt + \sigma_x dW_1 \tag{2.34}$$

where α, β, and σ_x are positive constants, and W_1 is the standard one-dimensional Brownian motion. We also consider a noise process modeled by the Ornstein-Uhlenbeck (O-U) process which is often used to model the perturbation of the membrane potential in biology as [70]:

$$dv = -\kappa v dt + \sigma_v dW_2 \tag{2.35}$$

where κ and σ_v are positive constants, and W_2 is the standard one-dimensional Brownian motion independent of W_1. Therefore, the two-dimensional augmented state is defined as $\mathbf{X} := [x \ v]^T$. The measurement model y is $y(t) = x(t) + v(t)$. In this example, the one-dimensional Send-on-Delta method is used as an event-based sampling (see Eq. (2.4) and Figure 2.2). That is, whenever the difference between the current $y(t)$ and the last transmitted value y^i exceeds the predefined threshold δ, $y(t)$ is sent to the estimator. To build the Markov chain, we design π using equations in the Appendix. For the numerical simulation, the values for constants are set as follows: $\alpha = 0.001$, $\beta = 50$, $\kappa = 1.2$, $\sigma_x = 0.5$, and $\sigma_v = 0.2$. The proposed Markov chain-based algorithm (denoted by 'EBMC') is compared with the particle filter-based algorithm (denoted by 'EBParticle') [58], both of which utilize the additional information generated between the two consecutive measurement arrivals (i.e., the information that the outputs of a system must be within the upper and lower threshold bounds defined by the events). Note that both the estimators receive the same number of measurements which are generated from the event-based sampling ('Send-on-Delta'). Figure 2.5 shows one realization of $x(t)$ and $v(t)$ which starts at $[0.3 \ 0]^T$. In Figure 2.6, the performance of the proposed algorithm (EBMC) is com-

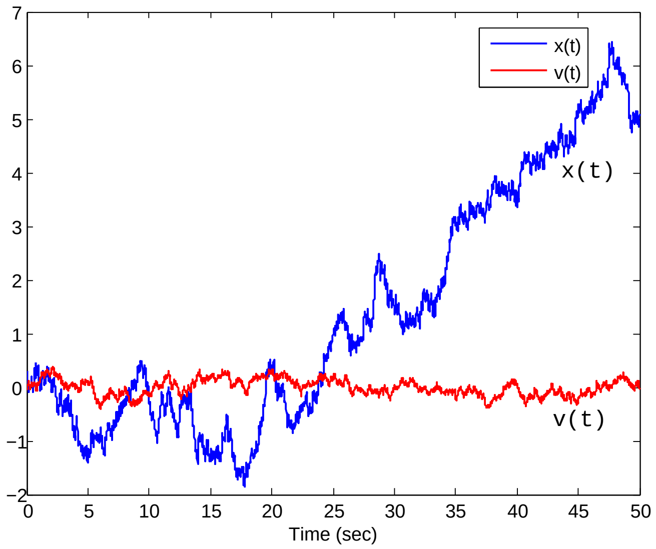


Figure 2.5. One realization of $x(t)$ and $v(t)$

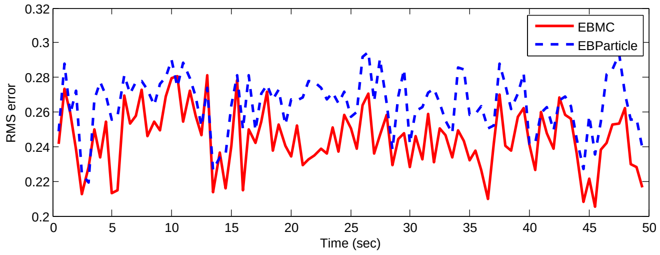


Figure 2.6. RMS errors of the proposed algorithm (EBMC) and the particle-based algorithm (EBParticle) (100 Monte Carlo simulations)

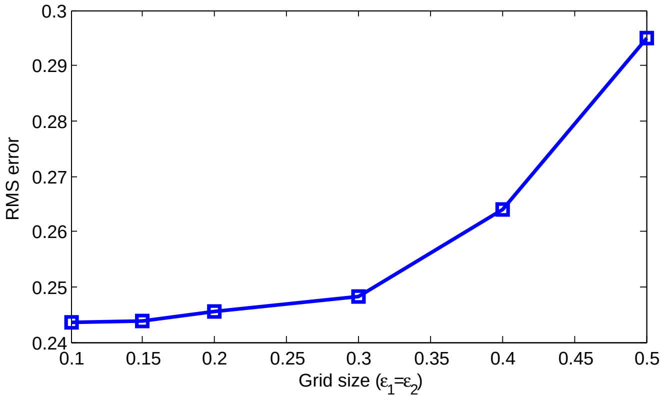


Figure 2.7. RMS state estimation error for different grid sizes of the EBMC where $\epsilon_1 = \epsilon_2$. For each grid size, 100 Monte Carlo runs are performed.

pared with the EBParticle in terms of state estimation error (i.e., difference between the true state $x(t)$ and the estimated state value). In the simulation, $\delta = 0.5$ for the 'Send-on-Delta', and $\lambda = 0.08$, $\epsilon_1 = \epsilon_2 = 0.2$ for the EBMC, and the number of particles for the EBParticle is 400 (the number is chosen so that the computation time of the EBParticle is almost the same as the EBMC). The simulation results show that the proposed estimation algorithm has smaller estimation errors than the EBParticle. In Figure 2.7, it is shown that as the grid sizes, ϵ_1 and ϵ_2 become smaller, the estimation error decreases. That is, as we have a finer grid, we can have more accurate approximation, which is consistent with Theorem 2.3.1.

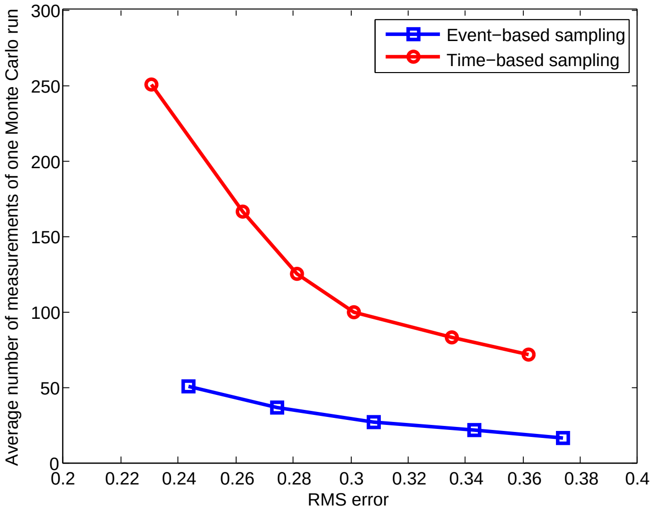


Figure 2.8. Comparison of estimation accuracy vs. number of measurements of event-based sampling (‘Send-on-Delta’) and time-based sampling. From the left to the right, δ for the ‘Send-on-Delta’ is 0.5, 0.6, 0.7, 0.8, and 0.9; and T for time-based sampling is 0.2 sec, 0.3 sec, 0.4 sec, 0.5 sec, 0.6 sec, and 0.7 sec. The entire simulation time is 50 sec.

We also compare the event-based sampling (‘Send-on-Delta’ in this example) with the time-based sampling (where measurements are taken at synchronous time intervals), to show the advantage of the event-based sampling with regard to the communication channel usage. By changing δ (for ‘Send-on-Delta’) and T (sampling time for time-based sampling), we perform several simulations with different number of measurements. For the estimation with the time-based sampling, we use the regular particle filter [71] (denoted by ‘RParticle’) which performs only its time update when there is no measurement transmission. Figure 2.8 shows the number of measurements used by the estimation algorithms (the EBParticle for ‘Send-on-Delta’ and the RParticle for time-based sampling). In the figure, it is found that, by using the event-based sampling, the desired estimation accuracy can be obtained with smaller number of measurements than time-based sampling, which motivates the use of event-based sampling.

3. EVENT-BASED HYBRID STATE ESTIMATION FOR CYBER-PHYSICAL SYSTEMS

In this chapter, we extend the event-based state estimation problem to CPSs that have the hybrid system structure, and develop the corresponding state estimation algorithm, called the Event-Based Hybrid State Estimation (EBHSE) algorithm. Firstly, the motivation and literature review for this problem are presented in Section 3.1. We then mathematically formulate the event-based hybrid state estimation problem in Section 3.2. The EBHSE algorithm is proposed in detail in Section 3.3 and applied to an aircraft tracking problem in the air traffic control application in Section 3.4.

3.1 Background and Motivations

The stochastic hybrid systems (SHS) are dynamical systems which contain interacting continuous states and discrete states (or modes) with uncertainties. The SHS have been widely used in various applications such as communication networks [23], target tracking [13,72], chemical reactions [73], and manufacturing [74], to name a few. To monitor and control the SHS, it is necessary to estimate both the discrete state and continuous state using noisy measurements, which leads to the hybrid state estimation problem. Several hybrid estimation algorithms have been developed [24,25,75,76] including the Interacting Multiple Model (IMM) algorithm [24], the multiple model adaptive estimation algorithm [75], the state dependent transition hybrid estimation algorithm [25], etc. These hybrid estimation algorithms, however, assume that measurements are generated at synchronous time intervals (time-based sampling), and thus cannot effectively and efficiently deal with event-based sampling.

The objective of this chapter is to develop an event-based state estimation algorithm for the SHS. There are two main difficulties to solve the hybrid estimation problem with event-based sampling. Firstly, the optimal hybrid state estimation requires the computation of the exponentially increasing number of probabilities of the discrete state histories. To deal with this problem, we exploit the idea of the IMM approach which keeps the computational complexity constant via mixing [13]. Secondly, the computation of the mode-matched state estimates with event-based sampling leads to the evaluation of a multivariate integration over the measurement space, which is computationally demanding. To efficiently compute the integration, we propose a pseudo measurement generation method, where the multivariate integration is approximated as the weighted sum of Gaussian functions evaluated at each pseudo measurement. This enables to find a closed-form representation of the mode-matched posterior distribution and mode-matched likelihood with bounded complexity.

3.2 Problem Formulation for Event-Based Hybrid State Estimation

In this section, a mathematical model for a class of SHS and a event-based sampling model are presented. The event-based hybrid state estimation problem is then formulated as to find the probability density functions (pdfs) of both the continuous and discrete states of the hybrid system.

3.2.1 Stochastic Linear Hybrid System

We consider a discrete-time stochastic linear hybrid system which consists of the continuous state $x(k) = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$, the discrete state (mode) $q(k) \in \mathcal{Q} = \{1, 2, \ldots, n_q\}$, and the measurement vector $y(k) = [y_1, y_2, \ldots, y_p]^T \in \mathbb{R}^p$, where $k = 0, 1, \ldots$ is the discrete-time index. For each $q(k)$, the continuous state dynamics and the measurement equation are given by

$$\begin{aligned} x(k+1) &= A_{q(k)}x(k) + B_{q(k)}w_{q(k)}(k) \\ y(k) &= C_{q(k)}x(k) + v_{q(k)}(k) \end{aligned} \tag{3.1}$$

where $w_{q(k)}(k)$ and $v_{q(k)}(k)$ are zero-mean white Gaussian noise with covariances $Q_{q(k)}$ and $R_{q(k)}$, respectively. The discrete state transitions are governed by a finite state

Markov chain whose state space is equivalent to $\mathcal{Q}$. Define the probability vector $m(k) \equiv [m^1(k), \ldots, m^{n_q}(k)]^T$, where $m^i(k)$ denotes the probability that the system is in the i-th discrete state at time k. The discrete state transition is then characterized by the evolution of the probability vector $m(k)$, which is given by

$$m(k+1) = \Pi m(k) \tag{3.2}$$

where $\Pi = [\pi_{ij}]_{n_q \times n_q}$ is the transition probability matrix whose elements are defined by

$$\pi_{ij} \equiv p(q(k+1) = j | q(k) = i) \tag{3.3}$$

with $\sum_{j=1}^{n_q} \pi_{ij} = 1$, where we use $p(\cdot|\cdot)$ to denote a conditional probability.

3.2.2 Event-Based Observation Model

We assume that the measurements are sent to an estimator only when certain events happen in order to reduce the communication cost (e.g., bandwidth). Unlike conventional time-based sampling where the estimator receives the measurements at a fixed sampling interval, the duration between the acquisition of two consecutive measurements can vary in event-based sampling. The conditions (events) by which sensors decide when to transmit their measurements can be defined in various ways. In this research, we consider the send-on-delta (SOD) method where the sensors transmit their measurement data to the estimator only when its measurement value varies more than a given specified value δ. By adjusting the δ value, the data transmission rate can be reduced and therefore can save network bandwidth resources.

Suppose the i-th measurement output y_i (i.e., measurement from the i-th sensor) ($i = 1, 2, \ldots, p$) is sampled with period T_s (for simplicity, we assume that each sensor's output y_i is scalar but this assumption can be easily extended to a general vector measurement case). To illustrate the SOD, let us assume that the i-th sensor transmitted its measurement data $y_i(k)$ to the estimator at time k. At the next sampling time $k+1$, if the difference between the current measurement value $y_i(k+1)$

and the previously transmitted one (in this case, $y_i(k)$) is greater than δ_i, the sensor sends $y_i(k+1)$ to the estimator; otherwise it does not. Let $y_{i,last}(k-1)$ represent the last transmitted measurement from the i-th sensor up to time $k-1$. Here, we define an indicator function $\gamma : \mathbb{R}^p \to [0,1]$ such that

$$\gamma(y_i(k)) = \begin{cases} 1, & \text{if } |y_i(k) - y_{i,last}(k-1)| > \delta_i \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

where δ_i, $i = 1, 2, \ldots, p$, is a given threshold value for each sensor. Each sensor's measurement, $y_i(k)$, is sent to the estimator when $\gamma(y_i(k)) = 1$ (i.e., when the difference between the current value and the previously transmitted value exceeds the pre-defined threshold). Define two index sets $T(k)$ and $S(k)$ as:

$$\begin{aligned} T(k) &= \{i | \gamma(y_i(k)) = 1, \text{ for } i = 1, 2, \ldots, p\} \\ S(k) &= \{1, 2, \ldots, p\} \setminus T(k) \end{aligned} \tag{3.5}$$

Then, an information vector $I(k)$ available to the estimator at time k, is defined as:

$$I(k) = \{\{y_i(k) | i \in T(k)\}, S(k)\} \tag{3.6}$$

where $\{y_i(k) | i \in T(k)\}$ is the *explicit information* which is directly sent to the estimator, and $S(k)$ gives the *implicit information* that, for $i \in S(k)$, $y_i(k)$ remains in the $(-\delta_i, \delta_i)$ interval of the last transmitted sensor value $y_{i,last}(k-1)$. The measurements that the estimator received at time k (i.e., $y_i(k)$ for $i \in T(k)$) are stored to update $y_{i,last}(k)$. The network structure of the sensors and the estimator with the SOD sampling is depicted in Figure 3.1.

3.2.3 Event-Based Hybrid State Estimation Problem

Let $\mathcal{I}^k \equiv \{I(1), I(2), \ldots, I(k)\}$ denote the set of information (available to the estimator) up to time k. The hybrid estimation problem is then defined as to compute

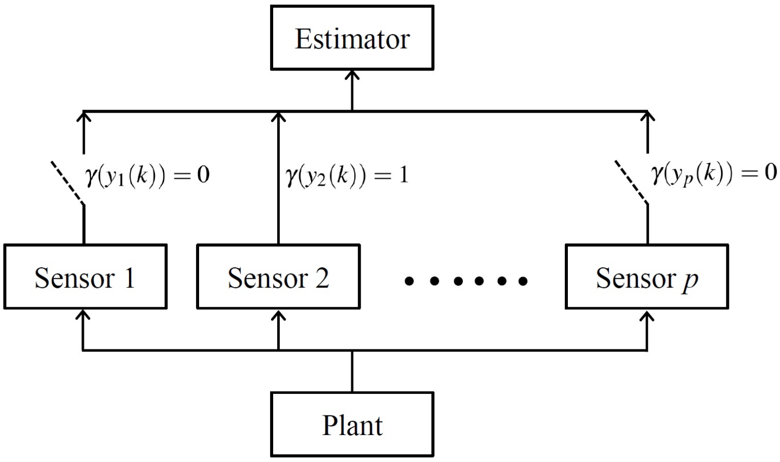


Figure 3.1. Networked system with the SOD sampling

both the continuous state pdf $p(x(k)|\mathcal{I}^k)$ and the mode probability $p(q(k)|\mathcal{I}^k)$. From the total probability theorem, $p(x(k)|\mathcal{I}^k)$ is computed by

$$p(x(k)|\mathcal{I}^k) = \sum_{i=1}^{n_q} p(x(k)|q(k) = i, \mathcal{I}^k)p(q(k) = i|\mathcal{I}^k) \tag{3.7}$$

Using the Bayesian approach, the hybrid state estimation problem can be formulated as follows: assume that at time $k-1$, the mode-conditioned continuous state pdfs $p(x(k-1)|q(k-1) = i, \mathcal{I}^{k-1})$ and the mode probabilities $m^i(k-1) = p(q(k-1) = i|\mathcal{I}^{k-1})$ have been computed for each mode $i \in \mathcal{Q}$. Then, the goal of the hybrid state estimation is to recursively compute $p(x(k)|q(k) = i, \mathcal{I}^k)$ and $m^i(k) = p(q(k) = i|\mathcal{I}^k)$ for all modes $i \in \mathcal{Q}$ using the new information vector $I(k)$ generated at time k. The hybrid state estimates are then obtained as:

$$\begin{aligned} \hat{x}(k) &:= E[x(k)|\mathcal{I}^k] \\ \hat{q}(k) &:= \underset{j}{\text{argmax}} p(q(k) = j|\mathcal{I}^k) \end{aligned} \tag{3.8}$$

where $E[\cdot|\cdot]$ denotes the conditional expectation of a random variable (i.e., $\hat{x}$ is the minimum mean square error estimate).

3.3 Event-Based Hybrid State Estimation Algorithm

In this section, a recursive filtering algorithm is developed for the event-based hybrid state estimation. There are the two main difficulties: the exponentially growing number of discrete state histories and the multivariate integration over the measurement space. To overcome these obstacles, we first exploit the IMM approach where the discrete state histories are approximated by a mode probability mixing step. In the mixing step, the mode-matched conditional posterior distributions are merged and approximated to a single Gaussian distribution to keep the number of mode histories constant. This enables to keep the exponentially growing number of the discrete state histories to the bounded computational complexity. Secondly, we also propose the pseudo measurement generation method which approximates the multivariate integration as the weighted sum of Gaussian functions evaluated at each pseudo measurement. This leads to a closed-form representation of the mode-matched posterior distribution and mode-matched likelihood.

Assume that from the last iteration at time $k-1$, the mode probabilities $m^i(k-1)$ are computed and the mode-conditioned continuous state pdfs are obtained as:

$$p(x(k-1)|q(k-1)=i,\mathcal{I}^{k-1}) = \mathcal{N}(x(k-1);\hat{x}^i(k-1),P^i(k-1)) \tag{3.9}$$

for $i = 1,2,\ldots,n_q$, where $\mathcal{N}$ represents the Gaussian distribution with the mean $\hat{x}^i(k-1)$ and the covariance $P^i(k-1)$.

Step 1: Mixing

The mixing probability $m^{i|j}(k)$ is computed by

$$\begin{aligned} m^{i|j}(k) &= p(q(k-1)=i|q(k)=j,\mathcal{I}^{k-1}) \\ &= \frac{p(q(k)=j|q(k-1)=i,\mathcal{I}^{k-1})p(q(k-1)=i|\mathcal{I}^{k-1})}{p(q(k)=j|\mathcal{I}^{k-1})} \\ &= \frac{\pi_{ij}m^i(k-1)}{\sum_{i=1}^{n_q}\pi_{ij}m^i(k-1)} \end{aligned} \tag{3.10}$$

Using the mixing probability, the initial conditions for a filter matched to mode j are obtained as:

$$\begin{aligned}\hat{x}^{0j}(k-1) &= \sum_{i=1}^{n_q} m^{i|j}\hat{x}^{i}(k-1)\\ P^{0j}(k-1) &= \sum_{i=1}^{n_q} m^{i|j}\left\{P^{i}(k-1)+[\hat{x}^{i}(k-1)-\hat{x}^{0j}(k-1)][\hat{x}^{i}(k-1)-\hat{x}^{0j}(k-1)]^{T}\right\}\end{aligned} \tag{3.11}$$

Step 2: Mode-matched estimation

For given mode j and the initial conditions ($\hat{x}^{0j}(k-1)$ and $P^{0j}(k-1)$), the mode-conditioned prior distribution $p(x(k)|q(k)=j,\mathcal{I}^{k-1})$ is computed as:

$$p(x(k)|q(k)=j,\mathcal{I}^{k-1}) = \mathcal{N}(x(k);\hat{x}^{j}(k|k-1),P^{j}(k|k-1)) \tag{3.12}$$

where

$$\begin{aligned}\hat{x}^{j}(k|k-1) &= A_j\hat{x}^{0j}(k-1)\\ P^{j}(k|k-1) &= A_jP^{0j}(k-1)A_j^T + B_jQ_jB_j^T\end{aligned} \tag{3.13}$$

From Bayes' theorem, the mode-conditioned posterior distribution $p(x(k)|q(k) = j,\mathcal{I}^{k})$ can be obtained as:

$$p(x(k)|q(k)=j,\mathcal{I}^{k}) = \frac{p(I(k)|x(k),q(k)=j,\mathcal{I}^{k-1})}{p(I(k)|q(k)=j,\mathcal{I}^{k-1})}p(x(k)|q(k)=j,\mathcal{I}^{k-1}) \tag{3.14}$$

Note that the computation of the likelihood function $p(I(k)|x(k),q(k)=j,\mathcal{I}^{k-1})$ is not trivial due to the implicit information contained in the information vector $I(k)$, unlike the conventional Kalman filter for time-based sampling where the likelihood function can be easily computed with residuals. From the explicit and the implicit information in $I(k)$, the likelihood function can be represented as a multivariate integral as [57,58]:

$$p(I(k)|x(k),q(k)=j,\mathcal{I}^{k-1}) = \int_{y(k)\in Y(k)} p(y(k)|x(k),q(k)=j,\mathcal{I}^{k-1})dy(k) \tag{3.15}$$

where $Y(k)$ is a set of $y(k)$ such that, for all $y(k) \in Y(k)$

$$\begin{aligned} y_i(k) = y_{i,last}(k) \text{ for } i \in T(k)\\ |y_i(k) - y_{i,last}(k-1)| \le \delta_i \text{ for } i \in S(k)\end{aligned} \tag{3.16}$$

Note that the direct evaluation of the multivariate integral in (3.15) is numerically intensive. To efficiently evaluate the multivariate integral in (3.15), we propose an approximate method based on pseudo measurement generation which discretizes the domain of the integral $Y(k)$ by a set of pseudo-measurements $y_{pseudo}(k)$ as:

1) Let $n_{S(k)}$ be the number of elements in $S(k)$ (i.e., the number of sensors not sending their measurements). Also, define an one-to-one map $\mathcal{M} : \{1, 2, \ldots, n_{S(k)}\} \rightarrow S(k)$ that connects a new index $l \in \{1, 2, \ldots, n_{S(k)}\}$ to i in $S(k)$ such that

$$\begin{aligned} \mathcal{M}(1) &= \text{smallest } i \text{ in } S(k) \\ \mathcal{M}(n_{S(k)}) &= \text{largest } i \text{ in } S(k) \\ &\text{and for } l_2 > l_1, \mathcal{M}(l_2) > \mathcal{M}(l_1) \end{aligned}$$

For example, if $S(k) = \{2, 5, 9\}$ (i.e., $n_{S(k)} = 3$), then $\mathcal{M}(1) = 2$, $\mathcal{M}(2) = 5$, and $\mathcal{M}(3) = 9$.

2) To discretize $Y(k)$, let N_l be the designed number of grid points for the $\mathcal{M}(l)$-th sensor for $l = 1, 2, \ldots, n_{S(k)}$. The grid size ϵ_l is then computed as:

$$\epsilon_l = \frac{2 \times \delta_{\mathcal{M}(l)}}{N_l} \tag{3.17}$$

For example, if $S(k) = \{2, 5, 9\}$ (and thus, $\mathcal{M}(1) = 2$, $\mathcal{M}(2) = 5$, and $\mathcal{M}(3) = 9$), then we have

$$\epsilon_1 = \tfrac{2\times\delta_2}{N_1},\ \epsilon_2 = \tfrac{2\times\delta_5}{N_2},\ \text{and } \epsilon_3 = \tfrac{2\times\delta_9}{N_3}$$

3) To represent each grid point in $Y(k)$, define a discretization vector $d := [d_1, d_2, \ldots, d_{n_{S(k)}}]^T \in \mathbb{N}^{n_{S(k)}}$ such that

$$1 \leq d_j \leq N_j \text{ for } j = 1, 2, \ldots, n_{s(k)} \tag{3.18}$$

For example, if $S(k) = \{2, 5, 9\}$ and $N_1 = N_2 = N_3 = 3$, then all the discretization vectors d are $[1, 1, 1]^T$, $[1, 1, 2]^T$,...,$[3, 3, 3]^T$. Therefore, the total number of the discretization vectors (i.e., the total number of grid points in $Y(k)$) is equal to $N_{tot} = \prod_{j=1}^{n_{S(k)}} N_j$. The pseudo-measurements $y^d_{pseudo}(k) = [y^d_{pseudo,1}, y^d_{pseudo,2}, \cdots, y^d_{pseudo,p}]^T \in Y(k)$ (i.e., a single grid point in $Y(k)$) corresponding to each discretization vector d is defined as:

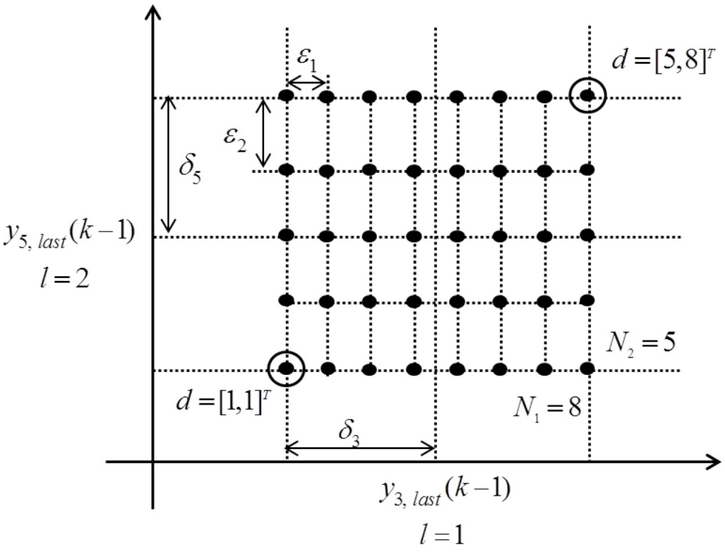


Figure 3.2. Illustration of grid points for pseudo-measurements

- $\forall i \in T(k)$

$$y^d_{pseudo,i} = y_{i,last}(k) \tag{3.19}$$

- $\forall i \in S(k)$

$$y^d_{pseudo,i} = y_{i,last}(k-1) - \delta_i + (d_{\mathcal{M}^{-1}(i)} - 1) \times \epsilon_{\mathcal{M}^{-1}(i)} \tag{3.20}$$

In Figure 3.2, the generation of grid points is illustrated. In the illustration, it is assumed that there are five sensors (i.e., $p = 5$) and only three sensors ($1, 2, 4$-th sensors) send their measurements to the estimator at time k. Therefore, $S(k) = \{3, 5\}$ (i.e., $\mathcal{M}(1) = 3$ and $\mathcal{M}(2) = 5$). The number of grid points for the sensors in $S(k)$ is given by $N_1 = 8$, $N_2 = 5$, and the total number of grid points is $N_{tot} = N_1 \times N_2 = 40$.

With the N_{tot} pseudo-measurements, the likelihood function in (3.15) can be approximated as:

$$\begin{aligned} p(I(k)|x(k), q(k)=j, \mathcal{I}^{k-1}) &= \int_{y(k)\in Y(k)} p(y(k)|x(k), q(k)=j, \mathcal{I}^{k-1}) dy(k) \\ &\approx \frac{D}{N_{tot}} \sum_{\forall d} p(y^d_{pseudo}(k)|x(k), q(k)=j, \mathcal{I}^{k-1}) \\ &= \frac{D}{N_{tot}} \sum_{\forall d} \mathcal{N}(y^d_{pseudo}(k); C_j x(k), R_j) \end{aligned} \tag{3.21}$$

where $D = \prod_{i\in S(k)}(2\delta_i)$. The following theorem provides a convergence property of the approximation.

Theorem 3.3.1 *Define the approximation error ϵ as:*

$$\epsilon = \left| \int_{y(k)\in Y(k)} p(y(k)|x(k), q(k)=j, \mathcal{I}^{k-1}) dy(k) - \frac{D}{N_{tot}} \sum_{\forall d} \mathcal{N}(y^d_{pseudo}(k); C_j x(k), R_j) \right| \tag{3.22}$$

Then,

$$\epsilon \to 0 \quad as \quad N_i \to \infty \;\; for \;\; i = 1, 2, \ldots, n_{S(k)} \tag{3.23}$$

The proof is presented in the Appendix. Note that the approximation in (3.21) can be considered as quasi-Monte Carlo approximation [77, 78], where the random samples of Monte Carlo methods are replaced by deterministic points. So, the convergence in (3.23) is deterministic as different from Monte Carlo approximations whose convergence is provided in a probabilistic way [79]. Note that the convergence property in Theorem 3.3.1 is also numerically demonstrated in the simulation section.

Using the approximated likelihood function, the mode-conditioned posterior distribution $p(x(k)|q(k)=j, \mathcal{I}^k)$ in (3.14) can be computed as:

$$\begin{aligned} p(x(k)|q(k)=j, \mathcal{I}^k) &\propto p(I(k)|x(k), q(k)=j, \mathcal{I}^{k-1}) p(x(k)|q(k)=j, \mathcal{I}^{k-1}) \\ &\approx \frac{D}{N_{tot}} \left(\sum_{\forall d} \mathcal{N}(y^d_{pseudo}(k); C_j x(k), R_j) \right) \\ &\quad \times \mathcal{N}(x(k); \hat{x}^j(k|k-1), P^j(k|k-1)) \end{aligned} \tag{3.24}$$

Note that from the identity of Gaussian distribution [80], the multiplication of two Gaussian distributions can be rewritten as:

$$\begin{aligned}&\mathcal{N}(y^d_{pseudo}(k); C_j x(k), R_j)\mathcal{N}(x(k); \hat{x}^j(k|k-1), P^j(k|k-1))\\&\qquad = \mathcal{N}(x(k); \lambda^d, \Lambda)\mathcal{N}(y^d_{pseudo}(k); \theta, \Theta)\end{aligned} \tag{3.25}$$

where

$$\begin{aligned}\Lambda &= \left(P^j(k|k-1)^{-1} + C_j^T R_j^{-1} C_j\right)^{-1}\\ \lambda^d &= \Lambda\left(P^j(k|k-1)^{-1}\hat{x}^j(k|k-1) + C_j^T R_j^{-1} y^d_{pseudo}(k)\right)\\ \theta &= C_j \hat{x}^j(k|k-1)\\ \Theta &= C_j P^j(k|k-1) C_j^T + R_j\end{aligned} \tag{3.26}$$

Because $\mathcal{N}(y^d_{pseudo}(k); \theta, \Theta)$ is constant for given d, $\hat{x}^j(k|k-1)$, and $P^j(k|k-1)$, it can be considered as weight w^d for $\mathcal{N}(x(k); \lambda^d, \Lambda)$. Then, the posterior distribution can be represented as the weighted sum of Gaussians which can then be approximated as a single Gaussian via moment matching:

$$\begin{aligned}p(x(k)|q(k) = j, \mathcal{I}^k) &= \sum_{\forall d} \frac{w^d}{\sum_{\forall d} w^d}\mathcal{N}(x(k); \lambda^d, \Lambda)\\ &\approx \mathcal{N}(x(k); \hat{x}^j(k), P^j(k))\end{aligned} \tag{3.27}$$

where

$$\begin{aligned}\hat{x}^j(k) &= \sum_{\forall d} \frac{w^d}{\sum_{\forall d} w^d}\lambda^d\\ P^j(k) &= \sum_{\forall d} \frac{w^d}{\sum_{\forall d} w^d}\left\{\Lambda + (\lambda^d - \hat{x}^j(k))(\lambda^d - \hat{x}^j(k))^T\right\}\end{aligned} \tag{3.28}$$

Step 3: Mode probability update

The mode probability is updated by using Bayes' rule as:

$$\begin{aligned}m^j(k) &= p(q(k) = j|\mathcal{I}^k)\\ &= \frac{1}{c}p(I(k)|q(k) = j, \mathcal{I}^{k-1})p(q(k) = j|\mathcal{I}^{k-1})\end{aligned} \tag{3.29}$$

where c is a normalizing constant; $p(I(k)|q(k)=j,\mathcal{I}^{k-1})$ is the mode-likelihood function computed using (3.21) by

$$\begin{aligned}p(I(k)|q(k)=j,\mathcal{I}^{k-1}) &= \int_{-\infty}^{\infty} p(I(k)|x(k),q(k)=j,\mathcal{I}^{k-1})p(x(k)|q(k)=j,\mathcal{I}^{k-1})dx(k)\\ &= \frac{D}{N_{tot}}\int_{-\infty}^{\infty}\left(\sum_{\forall d}\mathcal{N}(y^d_{pseudo}(k);C_jx(k),R_j)\right)\\ &\quad\times\mathcal{N}(x(k);\hat{x}^j(k|k-1),P^j(k|k-1))dx(k)\\ &= \frac{D}{N_{tot}}\sum_{\forall d}\mathcal{N}(C_j\hat{x}^j(k|k-1);y^d_{pseudo}(k),C_jP^j(k|k-1)C_j^T+R_j)\end{aligned} \tag{3.30}$$

and $p(q(k)=j|\mathcal{I}^{k-1})$ is the prior mode probability given by

$$\begin{aligned}p(q(k)=j|\mathcal{I}^{k-1}) &= \sum_{i=1}^{n_q} p(q(k)=j|q(k-1)=i,\mathcal{I}^{k-1})p(q(k-1)=i|\mathcal{I}^{k-1})\\ &= \sum_{i=1}^{n_q}\pi_{ij}m_i(k-1)\end{aligned} \tag{3.31}$$

Step 4: Output

The continuous state estimate $\hat{x}(k)$ and its covariance $P(k)$ are obtained using (3.28) and (3.29) as:

$$\begin{aligned}\hat{x}(k) &= \sum_{j=1}^{n_q}\hat{x}^j(k)m^j(k)\\ P(k) &= \sum_{j=1}^{n_q}\left\{P^j(k)+(\hat{x}^j(k)-\hat{x}(k))(\hat{x}^j(k)-\hat{x}(k))^T\right\}m^j(k)\end{aligned} \tag{3.32}$$

The discrete state estimate $\hat{q}$ can be computed with (3.8). The proposed event-based hybrid state estimation algorithm is summarized in Table 3.1.

3.4 Numerical Simulation

In this section, the proposed algorithm for event-based hybrid state estimation is demonstrated with an illustrative maneuvering aircraft tracking problem. Consider the aircraft's motion in the two-dimensional horizontal plane [13, 72]. Let

Table 3.1. Event-based hybrid state estimation algorithm

```
1.Initialization
 $\hat{x}^j(0)$, $P^j(0)$, $m^j(0)$ for $j = 1, 2, \dots, n_q$
2.Iteration
  for ($k = 1, 2, \dots$)
    Mixing
      Compute $\hat{x}^{0j}(k-1)$ and $P^{0j}(k-1)$ for $j = 1, 2, \dots, n_q$ using (3.11)
    Mode-matched estimation
      Compute mode-conditioned continuous state estimates $\hat{x}^j(k)$ and $P^j(k)$
      for $j = 1, 2, \dots, n_q$ using (3.28)
    Mode probability update
      Update mode probability $m^j(k)$ for $j = 1, 2, \dots, n_q$ using (3.29)
    Output
      Compute $\hat{x}(k)$ and $P(k)$ using (3.32)
  end
```

$x = [\xi, \dot{\xi}, \eta, \dot{\eta}]^T$ be the continuous states describing the aircraft's motion with the ξ-axis pointing the east and the η-axis pointing the north. In this example, the aircraft is assumed to have two flight modes ($n_q = 2$): 1) *Constant Velocity (CV) mode* ($q = 1$) and 2) *Coordinated Turn (CT) mode* ($q = 2$). The continuous state dynamics for each mode q is modeled as:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & T_s & 0 & -\frac{1}{2}\dot{\psi}_q T_s^2 \\ 0 & 1 & 0 & -\dot{\psi}_q T_s \\ 0 & \frac{1}{2}\dot{\psi}_q T_s^2 & 1 & T_s \\ 0 & \dot{\psi}_q T_s & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}T_s^2 & 0 \\ T_s & 0 \\ 0 & \frac{1}{2}T_s^2 \\ 0 & T_s \end{bmatrix} w_q(k) \\ y(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(k) + v_q(k) \end{aligned} \tag{3.33}$$

where $\dot{\psi}_1 = 0$ for mode 1 and $\dot{\psi}_2 = -1.3\ deg/s$ for mode 2; the sampling time $T_s = 2\ sec$;

$$\begin{gathered} Q_1 = \begin{bmatrix} 0.05^2 & 0 \\ 0 & 0.05^2 \end{bmatrix},\ Q_2 = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix} \\ R_1 = R_2 = \begin{bmatrix} 25^2 & 0 \\ 0 & 25^2 \end{bmatrix} \end{gathered} \tag{3.34}$$

The switching between the two modes is governed by a first-order Markov chain with the following transition probability matrix:

$$\Pi = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \tag{3.35}$$

Note that for both modes, the measurement $y \in \mathbb{R}^2$ is the aircraft's position (ξ, η). In this example, we assume that the measurements are sent to an estimator using the SOD method with given δ_1 and δ_2.

The proposed event-based hybrid state estimation algorithm (denoted by 'EBHSE') is compared to the conventional IMM approach (denoted by 'IMM') where each Kalman filter works as asynchronous filter (i.e., it ignores the implicit information that for sensors not sending their measurements, the measurements remain in the $(-\delta, \delta)$ interval from the last transmitted values). That is, each asynchronous Kalman filter propagates its mode-conditioned mean and covariance using (3.13) and updates them using the standard measurement update equation as:

$$\hat{x}^j(k|k) = \hat{x}^j(k|k-1) + K_j(k)(\tilde{y}(k) - \tilde{C}_j(k)\hat{x}^j(k|k-1))$$

where $K_j(k) = P^j(k|k-1)\tilde{C}_j(k)^T(\tilde{C}_j(k)P^j(k|k-1)\tilde{C}_j(k)^T + R_j)^{-1}$; $\tilde{y}(k)$ denotes a set of measurements $y_i(k)$ such that $i \in T(k)$; and $\tilde{C}_j(k)$ is the measurement matrix mapping the states x into $\tilde{y}(k)$. Note that each algorithm uses the same number of measurements obtained by the SOD method, but the proposed algorithm systematically utilizes the implicit information to improve the estimation accuracy. In addition, the difference between the proposed algorithm and the conventional IMM algorithm can be represented by the information vector $I(k)$ which they consider.

1) Conventional IMM approach with asynchronous Kalman filters:

$$I(k) = \{y_i(k)|i \in T(k)\}$$

2) Event-based hybrid state estimation:

$$I(k) = \{\{y_i(k)|i \in T(k)\}, S(k)\}$$

Operating scenario: the aircraft starts at the location $(1000\ m, 1000\ m)$ in the $\xi - \eta$ coordinates. The initial velocity is $(-10\ m/s, -10\ m/s)$ and the aircraft keeps a nearly constant velocity (i.e., *CV mode*) for 140 *sec*. Then, the aircraft makes a coordinate turn (i.e., *CT mode*) with $\dot{\psi}_2 = -1.3\ deg/s$ for 60 *sec*. After the coordinate turn, the aircraft flies at a constant velocity (i.e., *CV mode*) for another 120 *sec*. Figure 3.3 shows the result of a single run including the true trajectory of the aircraft with the estimated trajectories obtained by each estimation method (where the number of grids is chosen by $N_1 = N_2 = 10$ and $\delta_1 = 100\ m$ and $\delta_2 = 100\ m$ for the SOD algorithm). The mode estimation accuracies of both algorithms are compared in Figure 3.4, where the mode estimate $\hat{q}$ is obtained using (3.8). Figure 3.5 shows the

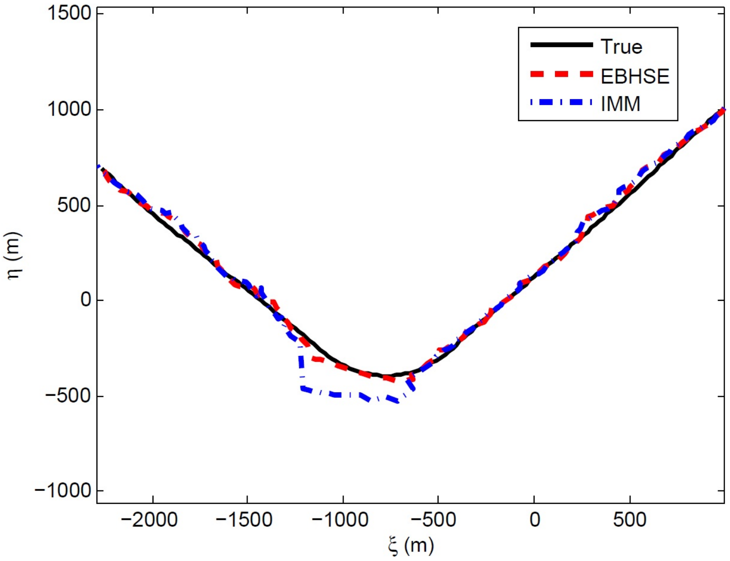
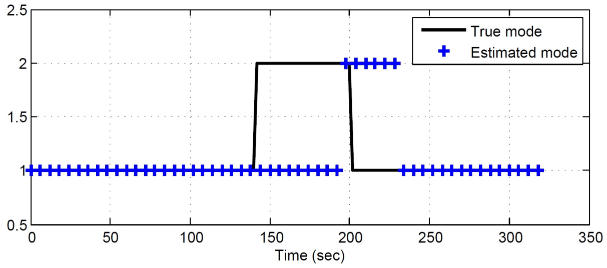
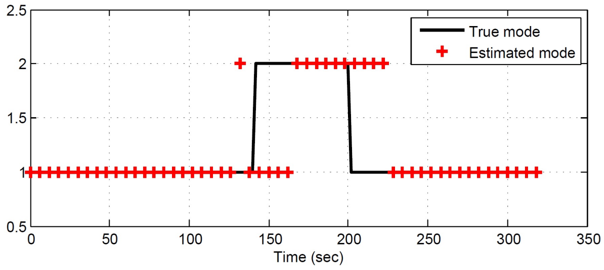


Figure 3.3. Actual and estimated trajectories of the aircraft (a single run)



(a) Estimated mode from the IMM



(b) Estimated mode from EBHSE

Figure 3.4. Comparison of mode-estimation accuracy (a single run) with $\delta_1 = \delta_2 = 100\ m$

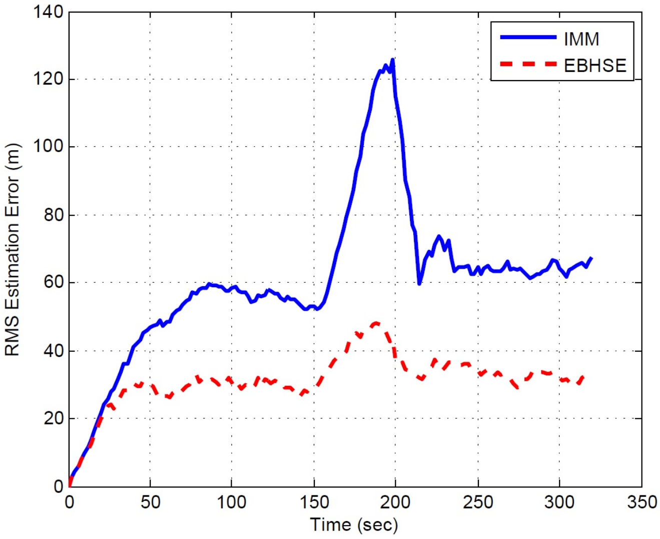


Figure 3.5. RMS position errors with 100 Monte Carlo runs with $\delta_1 = \delta_2 = 100\ m$

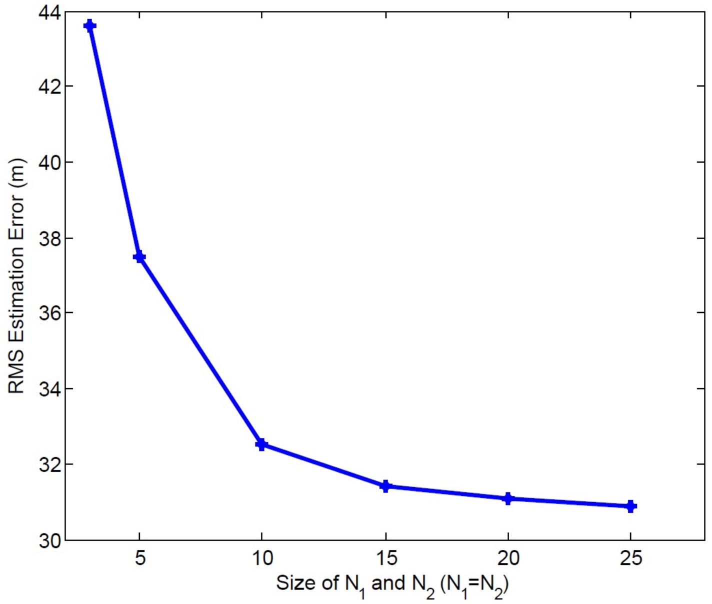


Figure 3.6. RMS position errors of EBHSE for different number of grid points ($N_1 = N_2$) with 100 Monte Carlo runs

RMS position estimation error for each method and other statistics of the results are summarized in Table 3.2.

It is obvious that the proposed algorithm produces more accurate estimates compared to the conventional IMM approach. In particular, during the coordinate turn, the aircraft's η coordinate remains within $2\delta_2$ (i.e., no measurement data of y_2 is provided to the estimator during the coordinated turn). This causes the conventional IMM to lose the estimation accuracy while the proposed algorithm keeps the accuracy by using the implicit information. The RMS error statistics for each method is summarized in Table 3.2 with the different values of δ_1 and δ_2. It is found that the estimation accuracy is improved as the size of δ_1 and δ_2 decreases. This result is reasonable because measurements are transmitted more frequently to the estimator with smaller δ_1 and δ_2. Figure 3.6 shows that as the number of grid points, N_1 and N_2 become larger, the estimation error decreases, validating the convergence of the approximation error in Theorem 3.3.1.

Table 3.2. Comparison of estimation performance (100 Monte Carlo runs)

		$\delta_1 = \delta_2 = 75\ m$	$\delta_1 = \delta_2 = 100\ m$	$\delta_1 = \delta_2 = 125\ m$	$\delta_1 = \delta_2 = 150\ m$
		EBHSE / IMM	EBHSE / IMM	EBHSE / IMM	EBHSE / IMM
RMS position error (m)	Peak	44.33 / 115.69	55.69 / 139.07	76.75 / 169.55	105.77 / 215.37
	Average	26.50 / 49.64	30.97 / 55.88	36.42 / 62.45	42.10 / 72.69
RMS velocity error (m/sec)	Peak	3.19 / 4.98	3.46 / 5.38	4.19 / 6.18	4.93 / 7.51
	Average	2.21 / 2.67	2.38 / 2.91	2.63 / 3.18	2.80 / 3.45
Average number of discrete state estimation errors (number of time steps)		8.50 / 9.38	8.88 / 10.55	9.41 / 11.09	9.96 / 11.43

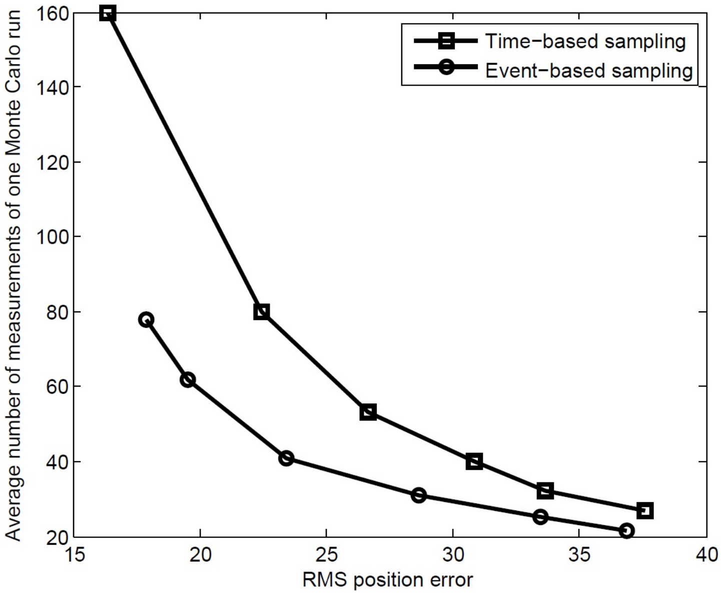


Figure 3.7. Comparison of estimation accuracy versus number of measurements of event-based sampling (Send-on-Delta) and time-based sampling. From the left to the right, $\delta_1 = \delta_2$ for the Send-on Delta is 30 m, 40 m, 60 m, 80 m, 100 m, and 120 m; and T_s for time-based sampling is 2 sec, 4 sec, 6 sec, 8 sec, 10 sec, and 12 sec.

To show the benefit of event-based sampling in terms of data transmission rate, we compare it to time-based sampling (where measurements are sent to the estimator at regular time intervals). To adjust the number of measurements for the comparison study, we change the sizes of δ_1 and δ_2 for the SOD, and the sampling time T_s for time-based sampling. In Figure 3.7, it is shown that the desired estimation accuracy can be attained with the smaller number of measurements by using event-based sampling. This result shows the advantage of event-based sampling over time-based sampling regarding to the communication channel usage.

4. APPLICATION TO SPACE SITUATIONAL AWARENESS I: TRACKING OF MANEUVERING SPACECRAFT

In this chapter, an adaptive state estimation algorithm for CPSs subject to abrupt state jumps is developed and applied to tracking problems of impulsively maneuvering spacecraft in the SSA application. First, the motivations of this research are discussed in Section 4.1. Then, in Section 4.2, we present a mathematical model for CPSs in which abrupt state jumps occur probabilistically conditioned on the state of the systems. Using the model, Section 4.3 proposes a new adaptive state estimation algorithm in detail. In Section 4.4, the proposed algorithm is applied to two illustrative examples, 1) tracking of a geostationary satellite performing station keeping maneuvers and 2) tracking of a spacecraft performing orbital transfers.

4.1 Background and Motivations

As space has become highly congested by many space objects, SSA has become crucial for the safe operation of space assets. One of the challenging tasks in SSA is the surveillance and tracking of spacecraft that have the capability to maneuver to perform various space missions. These tasks are essential to accurately predict the future trajectories of maneuvering spacecraft, and thus to effectively manage the safety of other spacecraft around the predicted future trajectories. It has been reported that there are more than 1000 currently operating spacecraft with the maneuver capability, and the number of such spacecraft is increasing [81,82]. This emphasizes the need to develop effective and efficient techniques for maneuvering spacecraft tracking.

The types of spacecraft maneuvers can be classified into two categories based on the type of the propulsion system used: 1) impulsive maneuvers using high-thrusts and 2) continuous maneuvers using low-thrusts. This study focuses on spacecraft with impulsive high-thrust maneuvers. The impulsive maneuvers usually occur using high thrusts which can provide high enough thrust to rapidly change the trajectories of the maneuvering spacecraft. The rapid changes of the maneuvering spacecraft's trajectory can cause more dangerous and imminent safety issues to adjacent spacecraft, and thus more efforts need to be made to accurately monitor the impulsively maneuvering spacecraft. In this sense, an emphasis is placed on the tracking of spacecraft implementing impulsive maneuvers.

The accurate tracking of maneuvering spacecraft is a challenging problem since the magnitude and the time of occurrence of maneuvers are usually unknown. To overcome these challenges, several approaches have been developed to detect orbital maneuvers of spacecraft. Kelecy and Jah [83] used an orbit determination technique based on batch least squares and the extended Kalman filter (EKF) to detect a single finite maneuver, where they checked inconsistency between filtered and smoother state estimates. Holzinger et al. [84] developed an optimal control performance metric based on the consistency check of measurement residuals to detect and characterize orbital maneuvers. A similar approach based on the optimal control framework was proposed and applied to GEO satellite maneuver detection and reconstruction problem by Lubey et al. [29]. The idea of the consistency check has been also applied to recorded two-line element set (TLE) data. Patera [37] proposed a moving window curve fit approach using the TLE data to detect the change of the maneuvering spacecraft's energy. Lemmens and Krag [85] also used the TLE data to check any inconsistency of the spacecraft's orbital elements due to maneuvers. Changes in orbital elements have also been explicitly used in [38] to estimate the maneuver time and magnitude of thrust using the Gaussian variation of parameters equations [86]. The performance of these approaches, however, is highly dependent on the tracking accuracy (i.e., the accuracy of estimated orbital elements or states), which implies the necessity of effective state estimation algorithms capable of providing accurate state estimates of the maneuvering spacecraft.

Many tracking algorithms have been developed to estimate the position and velocity of a maneuvering spacecraft (see Teixeira et al. [87] and references therein). One of the simplest approaches is to use a single Kalman filter based on a single dynamical model [87, 88]. However, these approaches may suffer from significant errors as a priori unknown maneuvers can cause filter divergence. To account for the maneuvers, multiple model adaptive estimation (MMAE) approaches have been developed where the maneuvers are considered as changes within multiple dynamics [75, 89, 90]. The interacting multiple-model (IMM) algorithm [72] is one of the most popular algorithms among them, and it has been successfully applied to maneuvering spacecraft tracking [34, 91]. However, these approaches also have limitations. They are applied to the cases where maneuvers occur for a reasonably long length of time (i.e., continuous low-thrust maneuvers) [91], or that the magnitude of a maneuver can take a value from an a priori known finite set [34]. These assumptions are not appropriate for the tracking of spacecraft with a priori unknown impulsive maneuvers.

Most importantly, it should be noted that, in many cases, the probability of occurrence of impulsive maneuvers is dependent on certain conditions on the state of the spacecraft. For example, for a geostationary satellite performing station keeping maneuvers, the impulsive maneuvers are likely to occur when its longitude or latitude approaches predefined bounds around a desired location. If this information is explicitly used in a tracking algorithm, the occurrences of impulsive maneuvers can be predicted more accurately, and thus the tracking accuracy can be improved. However, most of the aforementioned algorithms including the MMAE approaches do not incorporate the information explicitly, which emphasizes the need to develop a new tracking algorithm that can systematically exploit this information.

To address this problem, a new adaptive state estimation algorithm is developed in this study. First, a general mathematical model is proposed for a stochastic nonlinear dynamical system with unknown abrupt state jumps (e.g., abrupt changes in the velocity of a spacecraft due to impulsive maneuvers). A probabilistic model is also proposed to describe the abrupt state jumps whose probability of occurrence depends on the state of the system. Two extended Kalman filters are designed, each of which is matched to either a *no abrupt change model* or an *abrupt change model*, to account for the motion of the system with or without the abrupt jumps. The two filters are then systematically blended using a state-dependent transition probability which is derived using the probabilistic model for the abrupt state jumps. Through state-dependent blending, the information about the state-dependent probability of abrupt jumps is explicitly utilized in the proposed algorithm. The proposed algorithm is then applied to the tracking problem of impulsively maneuvering spacecraft.

4.2 Modeling of Stochastic Dynamical System Subject to Abrupt State Jumps

In this section, a mathematical model for a stochastic nonlinear dynamical system with unknown abrupt state changes (i.e. state jumps) is introduced, and a probabilistic model is presented to describe the abrupt state jumps whose probability of occurrence is dependent on the state of the system.

4.2.1 Stochastic Nonlinear Dynamical System with Unknown Jumps

Consider the following discrete-time nonlinear dynamical system and the observation equation

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k)) + \mathbf{B}\mathbf{w}(k) + \delta_{t,k+1}\mathbf{G}\mathbf{u} \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k)\end{aligned} \tag{4.1}$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the state, $\mathbf{y}(k) \in \mathbb{R}^p$ is the observation vector, $\mathbf{u} \in \mathbb{R}^m$ is the deterministic but unknown input vector, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^p$ are (piecewise) smooth bounded nonlinear functions, $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are l- and p-dimensional zero-mean white Gaussian noises with covariance $\mathbf{Q}(k)$ and $\mathbf{R}(k)$, respectively, $\mathbf{B} \in \mathbb{R}^{n\times l}$

and $\mathbf{G} \in \mathbb{R}^{n \times m}$ are constant matrices, and $\delta_{i,j}$ is the Kronecker delta with $\delta_{i,j} = 0$ for $i \neq j$ and $\delta_{i,j} = 1$ for $i = j$. The term $\delta_{t,k+1}\mathbf{G}\mathbf{u}$ in (4.1) represents abrupt jumps in the system whose magnitudes and times of occurrence (i.e., t in $\delta_{t,k+1}$) are unknown. Note that impulsive thrusts can produce abrupt jumps in the velocity of the spacecraft, and thus the spacecraft's motion during impulsive maneuvers can be described by (4.1) (in this case, $\mathbf{G}$ should be designed such that the abrupt jumps in the position are not allowed to occur).

4.2.2 Probabilistic Model for State-Dependent Abrupt Jumps

In many applications, abrupt jumps occur when certain conditions on the state of the system are satisfied. For example, for a spacecraft in a geostationary transfer orbit (GTO), impulsive maneuvers (i.e., abrupt jumps in velocity) are likely to occur when the spacecraft approaches the apogee of its orbit to transfer to a geostationary orbit (GEO). Another example is station-keeping maneuvers of geostationary satellites. When a geostationary satellite drifts out of a desired region (usually given as latitude/inclination and longitude bounds), impulsive maneuvers are performed to keep the position of the satellite within the desired region. Although those conditions are deterministic, actual abrupt jumps occur stochastically in practice due to navigation errors or external disturbances. In this sense, we build a probabilistic model for the abrupt jumps such that the probability of their occurrences is dependent on the state of system.

Let us denote the state-dependent jump probability as $\pi(\mathbf{x}(k))$. To model $\pi(\mathbf{x}(k))$, let us first consider an illustrating example of a spacecraft in a GTO. In this example, a maneuver occurs at the apogee of the GTO at about 42,164 km, which corresponds to the orbital radius of a GEO, as illustrated in Figure 4.1. The motion of the spacecraft could be modeled as a nonlinear dynamical system with abrupt state (i.e.,

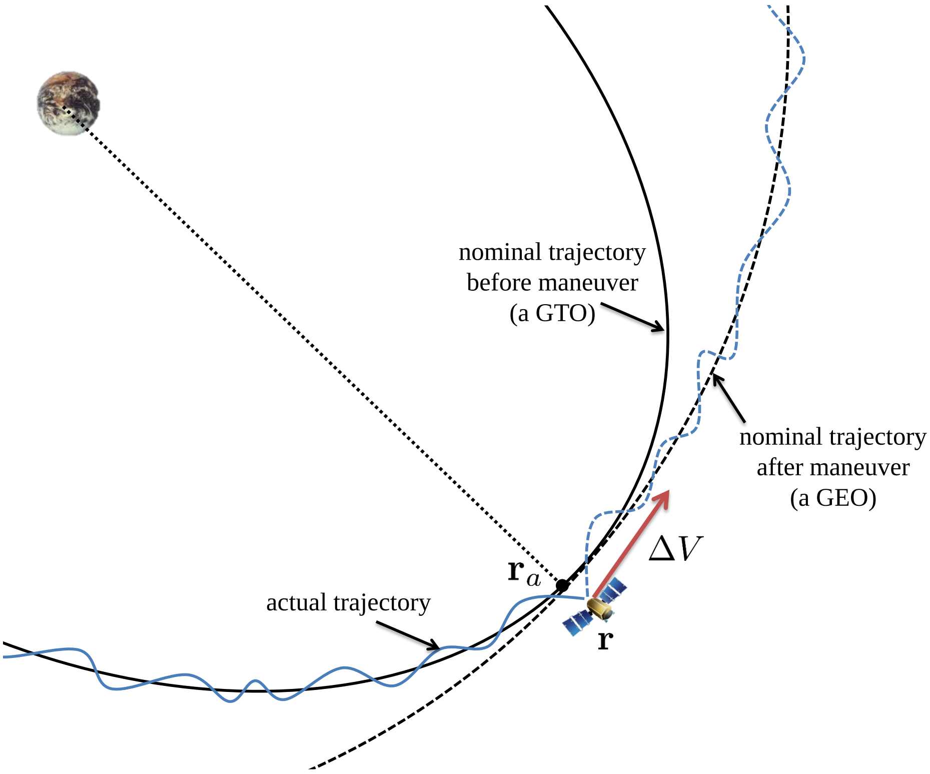


Figure 4.1. Impulsive maneuver from a GTO to a GEO at apogee with uncertainty

velocity) jumps as in (4.1). Let $\mathbf{r}(k)$ be the position of the spacecraft at time k and $\mathbf{r}_a$ be the location of the apogee, then an impulsive maneuver occurs when a condition

$$\mathbf{r}(k) - \mathbf{r}_a = \mathbf{0} \tag{4.2}$$

is satisfied. In practice, due to many factors such as orbital perturbations and navigation errors, the impulsive maneuver may occur around the apogee but not exactly at the apogee (see Figure 4.1). This uncertainty can be described by a probabilistic model where the probability of occurrence of the abrupt jump is conditioned on $\mathbf{r}(k) - \mathbf{r}_a$. It is reasonable that the jump probability is high when $\mathbf{r}(k)$ is close to $\mathbf{r}_a$. Suppose that the position $\mathbf{r}(k)$ can be represented as a function of the state of the spacecraft in a general form as $\mathbf{r}(k) = \mathbf{L}\mathbf{g}(\mathbf{x}(k))$, where $\mathbf{L}$ is a constant matrix and $\mathbf{g}$ is a smooth nonlinear function. For example, if the state itself is the position, i.e., $\mathbf{x}(k) = \mathbf{r}(k)$, then, $\mathbf{L}$ is the identity matrix and $\mathbf{g}$ is the identity function (more complex cases where $\mathbf{L}$ and $\mathbf{g}$ are nontrivial will be presented in the simulation section where station keeping maneuvers of a geostationary satellite are considered). Then, the jump probability can be modeled as a function $f(\mathbf{L}\mathbf{g}(\mathbf{x}(k)) - \mathbf{r}_a)$ in this example. Motivated by this example, the jump condition can be generally represented as:

$$\mathbf{L}\mathbf{g}(\mathbf{x}(k)) - \boldsymbol{\mu} = \mathbf{0} \tag{4.3}$$

and the corresponding state-dependent jump probability is modeled as:

$$\pi(\mathbf{x}(k)) = f(\mathbf{L}\mathbf{g}(\mathbf{x}(k)) - \boldsymbol{\mu}) \tag{4.4}$$

where $\mathbf{L} \in \mathbb{R}^{s\times q}$, $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^q$, $\boldsymbol{\mu} \in \mathbb{R}^s$, and $f : \mathbb{R}^s \to [0,1]$. Note that f should have the property that the state-dependent jump probability (4.4) is highest when the jump condition (4.3) is satisfied, i.e., $\mathbf{L}\mathbf{g}(\mathbf{x}(k)) = \boldsymbol{\mu}$, and gradually decreases as $\mathbf{L}\mathbf{g}(\mathbf{x}(k))$ moves away from $\boldsymbol{\mu}$. In this sense, we propose to use a multivariate Gaussian function to model f as:

$$\begin{aligned} \pi(\mathbf{x}(k)) &= f(\mathbf{L}\mathbf{g}(\mathbf{x}(k)) - \boldsymbol{\mu}) \\ &= \eta\mathcal{N}_s(\mathbf{L}\mathbf{g}(\mathbf{x}(k)); \boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{aligned} \tag{4.5}$$

where the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{s\times s}$ determines the rate of decrease in the jump probability as $\mathbf{L}\mathbf{g}(\mathbf{x}(k))$ moves away from $\boldsymbol{\mu}$, and $\eta \in [0,1]$ is a scaling parameter that needs to be chosen to satisfy that $0 \le \pi(\mathbf{x}(k)) \le 1$. It can be easily seen that the multivariate Gaussian function in (4.5) satisfies the desired property of f. In addition, the use of the multivariate Gaussian function facilitates the development of an analytical solution in the proposed adaptive estimation algorithm which will be presented in the next section.

4.3 State-Dependent Adaptive Estimation for Dynamical System with Abrupt State Jumps

In this section, an adaptive estimation algorithm is proposed to estimate the state of the nonlinear dynamical system (4.1) where abrupt jumps occur based on the state-dependent jump probability (4.5). We first formulate the state estimation problem as follows. Let $\mathbf{Y}^k \equiv \{\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(k)\}$ denote a set of measurements up to time k. Then, the state estimation problem is to compute the conditional pdf $p(\mathbf{x}(k)|\mathbf{Y}^k)$, and the minimum mean square error estimate $\hat{\mathbf{x}}(k)$ is obtained as:

$$\hat{\mathbf{x}}(k) := \mathbb{E}[\mathbf{x}(k)|\mathbf{Y}^k] \tag{4.6}$$

where $\mathbb{E}[\cdot|\cdot]$ denotes the conditional expectation of a random variable. This state estimation problem is nontrivial to solve due to the presence of abrupt state jumps, which causes the following challenges:

- First, the abrupt jumps in the state can cause filtering algorithms to diverge when the dynamical models used in the algorithms are not designed to account for the jumps.

- Second, incorporating the abrupt jumps into the dynamical models is difficult, since the magnitudes and the times of occurrence of the jumps are unknown a priori.

- Third, the knowledge of the state-dependent jump probability must be incorporated into the filter design, which is not straightforward.

To overcome the above difficulties, we propose an adaptive estimation algorithm based on multiple dynamical models [75]. Since the behavior of the system rapidly changes due to the abrupt jumps, a single dynamical model is not effective in accurately describing the complex behavior of the system. In this sense, two dynamical models (called the “No abrupt change model” and the “Abrupt change model”, respectively) are used in the proposed estimation algorithm as explained in the following section.

4.3.1 Two Dynamical Models for State Estimation

Let q denote the index of the two models (i.e., $q \in \{1, 2\}$). Both of the models have the following structure as:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{B}\mathbf{w}_q(k) \quad \text{for} \quad q = 1, 2 \tag{4.7}$$

where $\mathbf{w}_q(k)$ are zero-mean white Gaussian noises with covariances $\mathbf{Q}_q(k)$, respectively. As shown, the characteristics of the two models are determined by the value of $\mathbf{Q}_q(k)$, each of which is explained below.

- $q = 1$: ***No abrupt change model***

  The first model is a *no abrupt change* model where it is assumed that there is no abrupt jump (i.e., no impulsive maneuver). The covariance $\mathbf{Q}_1(k)$ is set to be equal to $\mathbf{Q}(k)$ (i.e., the same as the covariance of $\mathbf{w}(k)$ in (4.1), which means that this model is the same as the actual dynamics (4.1) when there is no abrupt jump (i.e., $\mathbf{Gu} = 0$). So, this model can accurately describe the behavior of the system when no abrupt jumps occur, but could cause significant errors when the unmodeled abrupt jumps occur.

- $q = 2$: ***Abrupt change model***

  The second model is an *abrupt change* model which accounts for possible jumps in the state of the system. In this model, a large covariance matrix $\mathbf{Q}_2(k)(> \rho I)$ is used to deal with the unknown abrupt jumps, where ρ is a positive constant and $\mathbf{Q}_2(k) > \rho I$ means that $\mathbf{Q}_2(k) - \rho I$ is positive definite. In this setting, the effects of the unknown abrupt jumps in the actual system ($\mathbf{Gu}$ in (4.1)) are covered by the process noise $\mathbf{w}_2(k)$ whose covariance $\mathbf{Q}_2(k)$ is set to a large number. In practice, the magnitude of each element in the covariance $\mathbf{Q}_2(k)$ can be chosen using the knowledge on the bounds on $\mathbf{Gu}$ in the actual system. For example, for impulsively maneuvering spacecraft, the covariance can be chosen using the bounds on impulsive thrusters that are currently operating in space [92]. Even though this model can account for the abrupt jumps, its accuracy is degraded when describing the behaviors of the actual system when no abrupt jumps occur. This is because the magnitude of the covariance $\mathbf{Q}_2(k)$ in the model is unnecessarily large compared to that of $\mathbf{Q}(k)$ in the actual system without the abrupt jumps.

If the occurrence times of abrupt jumps are known, one of the two models can be correctly chosen and used for the state estimation. However, the exact time is unknown in most cases (as assumed in this paper), and thus the two models need to be systematically blended to accurately describe the behaviors of the system. In this paper, we propose to use multiple model adaptive estimation [75], where multiple dynamical models are mixed using an adaptive weighting and a discrete transition between the models. Many algorithms have been developed for multiple model adaptive estimation. One of the most popular algorithms among them is the IMM algorithm [72] which has been shown to be efficient and effective in many applications. In particular, the IMM algorithm has produced excellent performance in maneuvering target tracking [22, 93, 94], which implies that it can also be applied to tracking an impulsively maneuvering spacecraft. However, the IMM algorithm and other similar approaches model the transitions between the multiple models as a Markov process with constant transition probabilities. So, those algorithms cannot explicitly incorporate a priori knowledge that the transitions (i.e., the transitions between the "No abrupt change model" and the "Abrupt change model" in this paper) are based on the state-dependent jump probability (4.5). In the following section, an adaptive estimation algorithm based on the two models (4.7) is proposed that can account for the state-dependent jump probability explicitly.

4.3.2 State-Dependent Adaptive Estimation

In the proposed algorithm, two extended Kalman filters (EKFs), each of which is matched to one of the dynamical models (4.7), are used. Define the mode probabilities $m^i(k)$ as:

$$m^i(k) := p(q(k) = i|\mathbf{Y}^k) \tag{4.8}$$

which denotes the probability that the i-th model (or called 'the i-th mode') correctly describes the actual system at time k given measurements up to time k. Let us assume that, from the last iteration at time $k-1$, the mode probabilities $m^i(k-1)$, $i = 1, 2$ are computed and the mode-matched probability density functions (pdfs) are obtained as:

$$p(\mathbf{x}(k-1)|q(k-1)=i,\mathbf{Y}^{k-1}) = \mathcal{N}_n(\mathbf{x}(k-1);\hat{\mathbf{x}}^i(k-1),\mathbf{P}^i(k-1)) \tag{4.9}$$

for $i = 1, 2$, where the mean $\hat{\mathbf{x}}^i(k-1)$ and the covariance $\mathbf{P}^i(k-1)$ are computed from the i-th EKF using the i-th model at time $k-1$ (note that the EKF assumes that the pdf is Gaussian, and thus computes only mean and covariance of the distribution). Then, using the new measurement $\mathbf{y}(k)$ obtained at time k, the mode-matched pdfs $p(\mathbf{x}(k)|q(k) = i, \mathbf{Y}^k)$ and the mode probabilities $m^i(k)$ for $i = 1, 2$ at time k can be recursively computed as shown in the following steps [26, 95].

Step 1: Mixing

The mixing probability $m^{i|j}(k)$ is computed as:

$$\begin{aligned} m^{i|j}(k) &= p(q(k-1)=i|q(k)=j,\mathbf{Y}^{k-1}) \\ &= \frac{p(q(k)=j|q(k-1)=i,\mathbf{Y}^{k-1})p(q(k-1)=i|\mathbf{Y}^{k-1})}{p(q(k)=j|\mathbf{Y}^{k-1})} \\ &= \frac{\gamma_{ij}(k-1)m^i(k-1)}{\sum_{l=1}^{2}\gamma_{lj}(k-1)m^l(k-1)} \end{aligned} \tag{4.10}$$

where $\gamma_{ij}(k-1) := p(q(k) = j|q(k-1) = i, \mathbf{Y}^{k-1})$ is the mode transition probability computed as:

$$\begin{aligned} &p(q(k)=j|q(k-1)=i,\mathbf{Y}^{k-1}) \\ &\qquad = \int_{\mathbb{R}^n} p(q(k)=j|q(k-1)=i,\mathbf{x})p(\mathbf{x}(k-1)=\mathbf{x}|q(k-1)=i,\mathbf{Y}^{k-1})d\mathbf{x} \end{aligned} \tag{4.11}$$

Define the state-dependent transition probability $\pi_{ij}(\mathbf{x})$ as:

$$\pi_{ij}(\mathbf{x}) := p(q(k)=j|q(k-1)=i,\mathbf{x}) \tag{4.12}$$

Note that $\pi_{12}(\mathbf{x})$ represents the probability of the transition from the “No abrupt change model” to the “Abrupt change model” conditioned on the state $\mathbf{x}$. It is obvious that this probability should be designed such that it increases as the jump probability of the actual system increases. This design strategy enables the proposed estimation algorithm to put more weight on the "Abrupt change model" as the actual system is likely to jump, and thus the proposed algorithm can perform more accurate state estimation. In this sense, we design $\pi_{12}(\mathbf{x})$ as equivalent to the state-dependent jump probability (4.5). The other terms $\pi_{11}(\mathbf{x})$, $\pi_{21}(\mathbf{x})$, $\pi_{22}(\mathbf{x})$ are then computed as:

$$\begin{aligned}
\pi_{11}(\mathbf{x}) &= 1 - \pi_{12}(\mathbf{x}) \\
\pi_{21}(\mathbf{x}) &= \pi_{11}(\mathbf{x}) \\
\pi_{22}(\mathbf{x}) &= 1 - \pi_{21}(\mathbf{x})
\end{aligned} \tag{4.13}$$

The integration in (4.11) can be computed as follows. We consider π_{12} first, as the other cases can be easily solved from (4.13) once π_{12} is computed. From (4.5), (4.9), (4.11), and (4.12), we have

$$\begin{aligned}
p(q(k) = 2|q(k-1) = 1, \mathbf{Y}^{k-1}) &= \int_{\mathbb{R}^n} \pi_{12}(\mathbf{x}) p(\mathbf{x}(k-1) = \mathbf{x}|q(k-1) = 1, \mathbf{Y}^{k-1}) d\mathbf{x} \\
&= \int_{\mathbb{R}^n} \pi(\mathbf{x}) p(\mathbf{x}(k-1) = \mathbf{x}|q(k-1) = 1, \mathbf{Y}^{k-1}) d\mathbf{x} \\
&= \int_{\mathbb{R}^n} \eta \mathcal{N}_s(\mathbf{L}\mathbf{g}(\mathbf{x}); \boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathcal{N}_n(\mathbf{x}; \hat{\mathbf{x}}^i(k-1), \mathbf{P}^i(k-1)) d\mathbf{x}
\end{aligned} \tag{4.14}$$

There are two cases: 1) $\mathbf{g}$ is the identity function ($\mathbf{g}(\mathbf{x}) = \mathbf{x}$), or 2) $\mathbf{g}$ is not the identity function. For the first case, from the identity of Gaussian distribution [80], the multiplication of two Gaussian distributions can be rewritten as:

$$\mathcal{N}_s(\mathbf{L}\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathcal{N}_n(\mathbf{x}; \hat{\mathbf{x}}^i(k-1), \mathbf{P}^i(k-1)) = \kappa \mathcal{N}_n(\mathbf{x}; \boldsymbol{\lambda}, \boldsymbol{\Lambda}) \tag{4.15}$$

where

$$\begin{aligned}
\boldsymbol{\Lambda} &= \left(\mathbf{P}^i(k-1)^{-1} + \mathbf{L}^T \boldsymbol{\Sigma}^{-1} \mathbf{L}\right)^{-1} \\
\boldsymbol{\lambda} &= \boldsymbol{\Lambda} \left(\mathbf{P}^i(k-1)^{-1} \hat{\mathbf{x}}^i(k-1) + \mathbf{L}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right)
\end{aligned} \tag{4.16}$$

and κ is a constant given by

$$\begin{aligned}
\kappa = &\frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{s/2} |\mathbf{P}^i(k-1)|^{1/2} |\boldsymbol{\Sigma}|^{1/2}} \\
&\times \exp\left[-\frac{1}{2}(\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} + \hat{\mathbf{x}}^i(k-1)^T \mathbf{P}^i(k-1)^{-1} \hat{\mathbf{x}}^i(k-1) - \boldsymbol{\lambda}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\lambda})\right]
\end{aligned} \tag{4.17}$$

Using (4.15), Equation (4.14) can be evaluated in a closed-form as [26, 95]:

$$\int_{\mathbb{R}^n} \mathcal{N}_s(\mathbf{Lg}(\mathbf{x}); \boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathcal{N}_n(\mathbf{x}; \hat{\mathbf{x}}^i(k-1), \mathbf{P}^i(k-1)) d\mathbf{x} = \eta\kappa \tag{4.18}$$

When $\mathbf{g}$ is not the identity function, we assume that the distribution of $\mathbf{g}(\mathbf{x})$ is given by a Gaussian distribution as $\mathcal{N}_q(\mathbf{g}(\mathbf{x}); \boldsymbol{\mu}_\mathbf{g}, \boldsymbol{\Sigma}_\mathbf{g})$. Then, the mode transition probability (4.14) can be computed as:

$$\begin{aligned} p(q(k) = 2|q(k-1) = 1, \mathbf{Y}^{k-1}) &= \int_{\mathbb{R}^n} \pi_{12}(\mathbf{x}) p(\mathbf{x}(k-1) = \mathbf{x}|q(k-1) = 1, \mathbf{Y}^{k-1}) d\mathbf{x} \\ &= \int_{\mathbb{R}^q} \eta \mathcal{N}_s(\mathbf{Lg}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathcal{N}_q(\mathbf{g}; \boldsymbol{\mu}_\mathbf{g}, \boldsymbol{\Sigma}_\mathbf{g}) d\mathbf{g} \end{aligned} \tag{4.19}$$

where the multiplication of two Gaussian distributions in the right-hand side can be simplified using (4.15)∼(4.17). Then, similarly to (4.18), Equation (4.19) can be evaluated in a closed-form.

Based on the mixing probability, the mean and covariance that need to be propagated at time k for the EKF matched to the j-th model are obtained as:

$$\begin{aligned} \hat{\mathbf{x}}^{0j}(k-1) &= \sum_{i=1}^{2} m^{i|j}(k) \hat{\mathbf{x}}^i(k-1) \\ \mathbf{P}^{0j}(k-1) &= \sum_{i=1}^{2} m^{i|j}(k) \left\{ \mathbf{P}^i(k-1) \right. \\ &\quad \left. + \left[\hat{\mathbf{x}}^i(k-1) - \hat{\mathbf{x}}^{0j}(k-1)\right] \left[\hat{\mathbf{x}}^i(k-1) - \hat{\mathbf{x}}^{0j}(k-1)\right]^T \right\} \end{aligned} \tag{4.20}$$

Step 2: Model-conditioned estimation

For a given model j and the initial conditions obtained in (4.20), the model-conditioned prior distribution $p(\mathbf{x}(k)|q(k) = j, \mathbf{Y}^{k-1})$ is computed as (based on the Gaussian assumption of the EKF):

$$p(\mathbf{x}|q(k) = j, \mathbf{Y}^{k-1}) = \mathcal{N}_n(\mathbf{x}(k); \hat{\mathbf{x}}^j(k|k-1), \mathbf{P}^j(k|k-1)) \tag{4.21}$$

where

$$\begin{aligned} \hat{\mathbf{x}}^j(k|k-1) &= \mathbf{f}(\hat{\mathbf{x}}^{0j}(k-1)) \\ \mathbf{P}^j(k|k-1) &= \mathbf{F}(k-1)\mathbf{P}^{0j}(k-1)\mathbf{F}(k-1)^T + \mathbf{BQ}_j(k-1)\mathbf{B}^T \end{aligned} \tag{4.22}$$

and $\mathbf{F}(k-1) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big|_{\hat{\mathbf{x}}^{0j}(k-1)}$. Using the new measurement $\mathbf{y}(k)$ at time k, the model-conditioned posterior distribution $p(\mathbf{x}(k)|q(k)=j,\mathbf{Y}^k)$ is computed as:

$$p(\mathbf{x}(k)|q(k)=j,\mathbf{Y}^k) = \mathcal{N}_n(\mathbf{x}(k);\hat{\mathbf{x}}^j(k),\mathbf{P}^j(k)) \tag{4.23}$$

where

$$\begin{aligned}\hat{\mathbf{x}}^j(k) &= \hat{\mathbf{x}}^j(k|k-1) + \mathbf{K}(k)(\mathbf{y}(k) - \mathbf{h}(\hat{\mathbf{x}}^j(k|k-1)))\\ \mathbf{P}^j(k) &= (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}^j(k|k-1)\end{aligned} \tag{4.24}$$

and

$$\begin{gathered}\mathbf{K}(k) = \mathbf{P}^j(k|k-1)\mathbf{H}(k)^T(\mathbf{H}(k)\mathbf{P}^j(k|k-1)\mathbf{H}(k)^T + \mathbf{R}(k))^{-1}\\ \mathbf{H}(k) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\bigg|_{\hat{\mathbf{x}}^j(k|k-1)}\end{gathered} \tag{4.25}$$

Step 3: Mode probability update

Using Bayes' rule, the mode probability is updated as:

$$\begin{aligned}m^j(k) &= p(q(k)=j|\mathbf{Y}^k)\\ &= \frac{1}{c}p(\mathbf{y}(k)|q(k)=j,\mathbf{Y}^{k-1})p(q(k)=j|\mathbf{Y}^{k-1})\end{aligned} \tag{4.26}$$

where c is a normalizing constant, $p(\mathbf{y}(k)|q(k)=j,\mathbf{Y}^{k-1})$ is the model-conditioned likelihood function given by

$$p(\mathbf{y}(k)|q(k)=j,\mathbf{Y}^{k-1}) = \mathcal{N}_p(\mathbf{y}(k) - \mathbf{h}(\hat{\mathbf{x}}^j(k|k-1));\mathbf{0},\mathbf{S}_j(k)) \tag{4.27}$$

where

$$\mathbf{S}_j(k) = \mathbf{H}(k)\mathbf{P}^j(k|k-1)\mathbf{H}(k)^T + \mathbf{R}(k) \tag{4.28}$$

and $p(q(k)=j|\mathbf{Y}^{k-1})$ is the prior mode probability computed as:

$$p(q(k)=j|\mathbf{Y}^{k-1}) = \sum_{i=1}^{2}\gamma_{ij}(k-1)m^i(k-1) \tag{4.29}$$

Step 4: Output

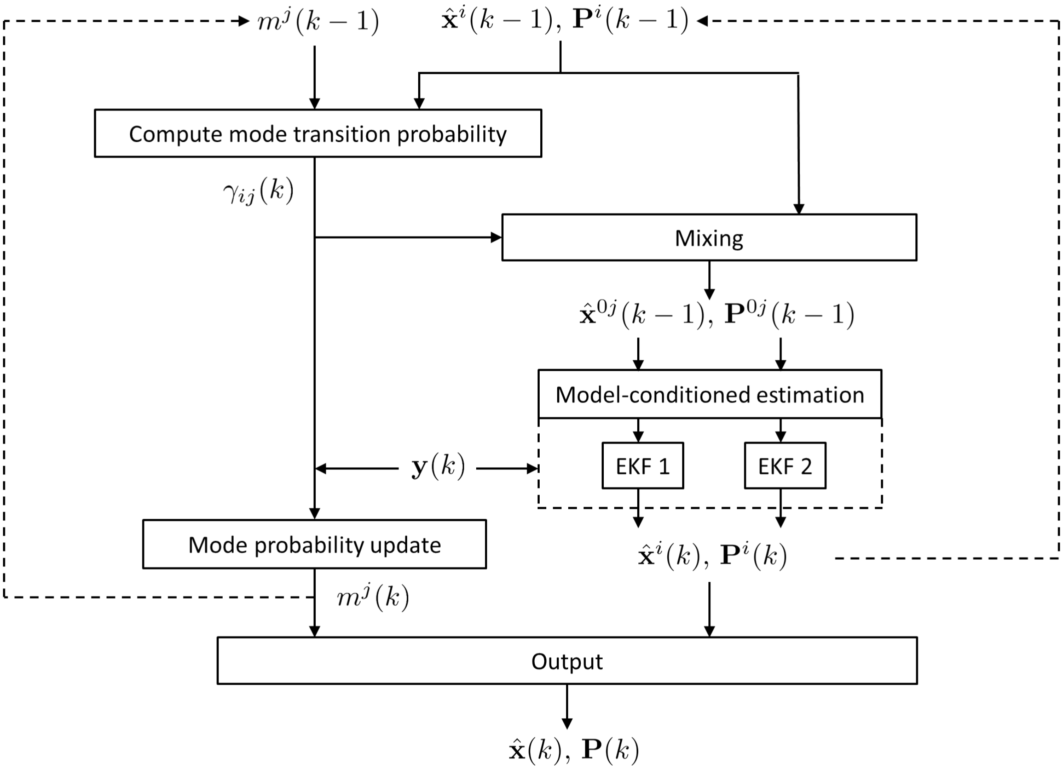

$m^j(k-1)$
$\hat{\mathbf{x}}^i(k-1)$, $\mathbf{P}^i(k-1)$
Compute mode transition probability
$\gamma_{ij}(k)$
Mixing
$\hat{\mathbf{x}}^{0j}(k-1)$, $\mathbf{P}^{0j}(k-1)$
Model-conditioned estimation
$\mathbf{y}(k)$
EKF 1
EKF 2
Mode probability update
$\hat{\mathbf{x}}^i(k)$, $\mathbf{P}^i(k)$
$m^j(k)$
Output
$\hat{\mathbf{x}}(k)$, $\mathbf{P}(k)$

Figure 4.2. Structure of the proposed algorithm (EKF denotes extended Kalman filter)

Using the model-conditioned state estimates $\hat{\mathbf{x}}^j(k)$ and covariances $\mathbf{P}^j(k)$, $j = 1, 2$, a single representative state estimate $\hat{\mathbf{x}}(k)$ and covariance $\mathbf{P}(k)$ are computed as weighted sums as:

$$\begin{aligned} \hat{\mathbf{x}}(k) &= \sum_{j=1}^{2} \hat{\mathbf{x}}^j(k) m^j(k) \\ \mathbf{P}(k) &= \sum_{j=1}^{2} \left\{ \mathbf{P}^j(k) + \left[\hat{\mathbf{x}}^j(k) - \hat{\mathbf{x}}(k)\right] \left[\hat{\mathbf{x}}^j(k) - \hat{\mathbf{x}}(k)\right]^T \right\} m^j(k) \end{aligned} \tag{4.30}$$

The overall structure of the proposed algorithm is shown in Figure 4.2.

4.4 Applications to the Tracking of Impulsively Maneuvering Spacecraft

In this section, the proposed algorithm is applied to two spacecraft tracking examples: 1) tracking of a geostationary satellite performing station keeping maneuvers and 2) tracking of a spacecraft performing orbital transfers. For each example, the equations of motion (4.1) are introduced, and an abrupt state jump model (4.5) is derived. The state-dependent adaptive estimation algorithm is then applied to estimate the position and velocity of the impulsively maneuvering spacecraft. The performance of the proposed algorithm is compared with the IMM algorithm and with two EKFs, each of which uses either the "No abrupt change model" only or the "Abrupt change model" only.

4.4.1 Example 1: Tracking of a Geostationary Satellite Performing Station Keeping Maneuvers

Since the geostationary satellite stays around a desired location which moves around the Earth in a circular orbit, the behavior of the geostationary satellite can be described using a relative motion with respect to the desired location. Define a state vector $\mathbf{x}$ as $\mathbf{x} := [x\ y\ z\ v_x\ v_y\ v_z]^T$, where x, y, and z represent the coordinates of the geostationary satellite's position with respect to the desired location in the local vertical and local horizontal (LVLH) frame (see Figure 4.3), and v_x, v_y, and v_z denote the velocities along each direction. Then, the relative motion of the geostationary

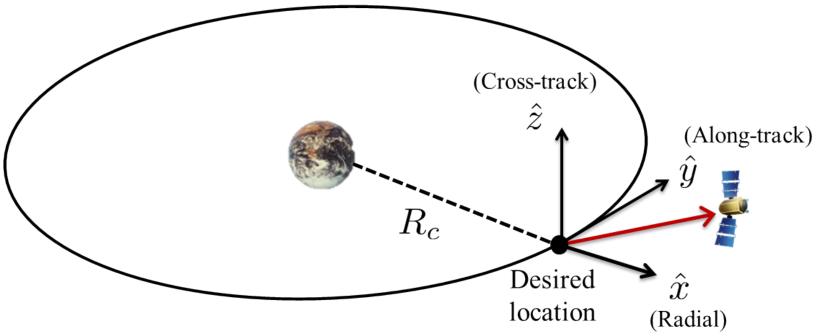


Figure 4.3. Relative motion in the local vertical and local horizontal (LVLH) frame

satellite performing impulsive maneuvers (i.e., abrupt state jumps) can be described as [96]:

$$\begin{aligned}\mathbf{x}(k+1) &= \begin{bmatrix} x(k+1) \\ y(k+1) \\ z(k+1) \\ v_x(k+1) \\ v_y(k+1) \\ v_z(k+1) \end{bmatrix} \\ &= \begin{bmatrix} x(k) + \Delta T v_x(k) \\ y(k) + \Delta T v_y(k) \\ z(k) + \Delta T v_z(k) \\ v_x(k) + \Delta T \left(2nv_y(k) + n^2 x(k) - \frac{\mu(R_c+x(k))}{[(R_c+x(k))^2+y(k)^2+z(k)^2]^{3/2}} + \frac{\mu}{R_c^2}\right) \\ v_y(k) + \Delta T \left(-2nv_x(k) + n^2 y(k) - \frac{\mu y(k)}{[(R_c+x(k))^2+y(k)^2+z(k)^2]^{3/2}}\right) \\ v_z(k) + \Delta T \left(-\frac{\mu z(k)}{[(R_c+x(k))^2+y(k)^2+z(k)^2]^{3/2}}\right) \end{bmatrix} \\ &\quad + \mathbf{w}(k) + \delta_{t,k+1} \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix} \mathbf{u}\end{aligned} \tag{4.31}$$

where $\mathbf{u} := [u_x \; u_y \; u_z]^T$; u_x, u_y, and u_z denote the control inputs along each direction; $\mathbf{w}(k) \in \mathbb{R}^6$ is a zero-mean white Gaussian noise with covariance $\mathbf{Q}(k)$; $n = \sqrt{\frac{\mu}{R_c^3}}$

is the mean orbital motion of the desired location; R_c is the orbital radius of the desired location and μ is the gravitational constant of the Earth; ΔT is the sampling time. We assume that the satellite is observed by a ground-based optical sensor that generates the angular measurements given as [97]:

$$\mathbf{y}(k) = \begin{bmatrix} \tan^{-1}\left(\frac{y(k)-y_{obs}(k)}{R_c+x(k)-x_{obs}(k)}\right) \\ \tan^{-1}\left(\frac{z(k)-z_{obs}(k)}{\sqrt{(R_c+x(k)-x_{obs}(k))^2+(y(k)-y_{obs}(k))^2}}\right) \end{bmatrix} + \mathbf{v}(k) \tag{4.32}$$

where $[x_{obs}(k)\ y_{obs}(k)\ z_{obs}(k)]^T$ is the position vector of the optical sensor with respect to the center of the Earth represented in the $\hat{x}-\hat{y}-\hat{z}$ frame, and $\mathbf{v}(k) \in \mathbb{R}^2$ is a zero-mean white Gaussian noise with covariance $\mathbf{R}(k)$. Due to orbital perturbations such as non-spherical earth gravity, solar radiation pressure, and third body gravity, the geostationary satellite drifts from the desired location, and thus it needs to regularly perform maneuvers to maintain its position within an allowed region centered on the desired location. Generally, the geostationary satellite performs two types of maneuvers: 1) North-South (NS) maneuvers for latitude (or inclination) correction and 2) East-West (EW) maneuvers for longitude correction. As an example, the proposed algorithm is applied to an NS maneuver case, where the desired location of the geostationary satellite is at 150 *deg* west longitude, and the corresponding jump condition is given as:

$$g(\mathbf{x}(k)) - inc_d = 0 \tag{4.33}$$

where $g : \mathbb{R}^6 \rightarrow \mathbb{R}$ is a mapping from the state $\mathbf{x}$ to the inclination of the geostationary satellite, and inc_d is the maximum allowed inclination error (ideally the inclination of the geostationary satellite remains at zero to ensure no latitude error). Note that inc_d corresponds to $\boldsymbol{\mu}$ in the general jump condition in (4.3) and is set as $inc_d = 0.2\ deg$ such that the corresponding latitude bound is 0.2 *deg* from the desired location. To account for the uncertainties in the jump conditions, the covariance ($\mathbf{\Sigma}$ in (4.5)) is set as $\mathbf{\Sigma} = 0.00001^2\ deg^2$. This value for $\mathbf{\Sigma}$ is chosen by assuming that the maneuver execution timing error is around $1 \sim 2$ sec, and considering the usual cross track drift rate (around 10 m/s) of a GEO satellite. The scaling parameter

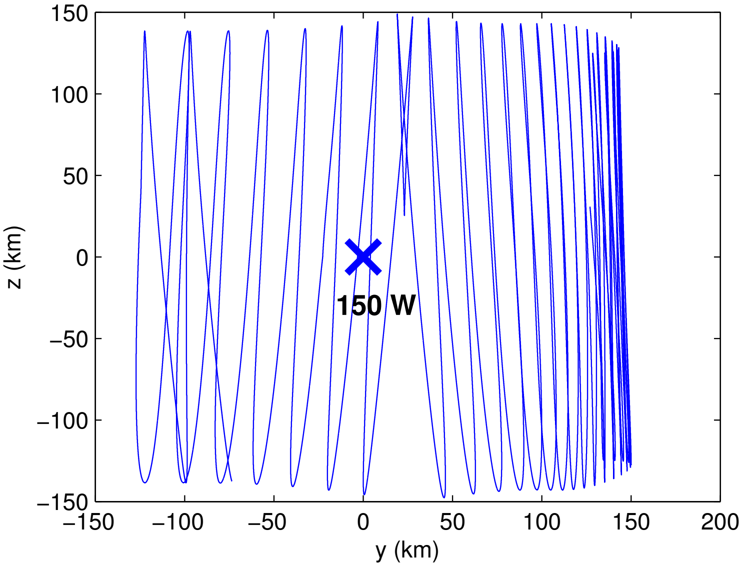


Figure 4.4. A true trajectory of the geostationary satellite in the $y-z$ plane of the LVLH frame (corresponding to longitude-latitude)

η is set to 1. Note that the proposed algorithm can also be applied to an EW maneuver case by setting the corresponding jump condition as $y(k) - y_d = 0$, where y_d is the maximum allowed along-track distance from the desired location (e.g., $y_d =$ 147 km for 0.2 deg longitude bound from the desired location). For simulation, the covariance of the process noise $\mathbf{Q}(k)$ of the true system (4.31) is set as $\mathbf{Q}(k) =$ $diag([10^{-2}\,m^2,\ 10^{-2}\,m^2,\ 10^{-2}\,m^2,\ 10^{-7}\,(m^2/sec^2),\ 10^{-7}\,(m^2/sec^2),\ 10^{-7}\,(m^2/sec^2)])$, where $diag([\cdot])$ denotes a square diagonal matrix with the elements of vector $[\cdot]$ on the main diagonal. The sampling rate ΔT is 10 sec (i.e., it is assumed that the measurement $\mathbf{y}(k)$ is obtained every 10 sec) and the covariance of the measurement noise $\mathbf{R}(k)$ is set as [98]:

$$\mathbf{R}(k) = \begin{bmatrix} 2^2\ arcsec^2 & 0 \\ 0 & 2^2\ arcsec^2 \end{bmatrix} \tag{4.34}$$

Figure 4.4 shows a true trajectory of the geostationary satellite, which is bounded around the desired location due to the station keeping maneuvers, and Figure 4.5

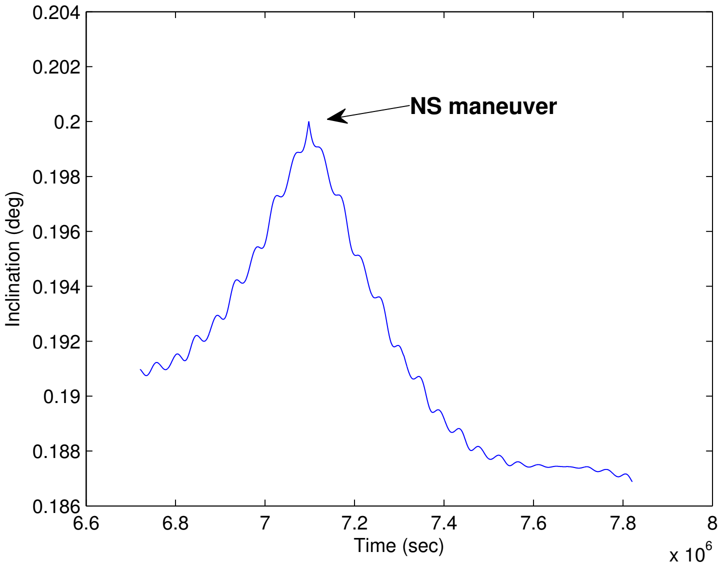
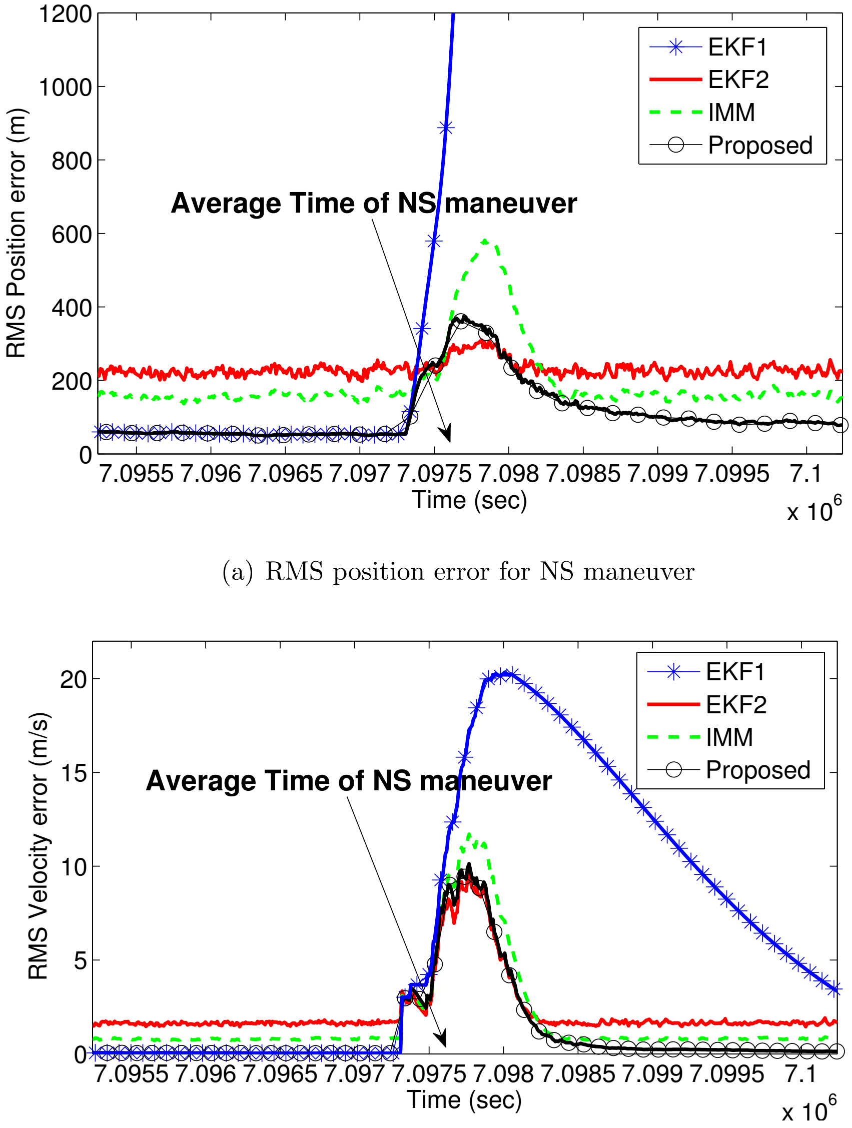


Figure 4.5. History of the inclination of the geostationary satellite around the moment of a NS station keeping maneuver

shows the history of the inclination at around time of a NS maneuver. Given the above simulated scenario, the proposed estimation algorithm is applied to estimate the state of the geostationary satellite. In the algorithm, the covariances of the process noises for both models $\mathbf{Q}_q(k)$ ($q = 1$: no abrupt change model and $q = 2$: abrupt change model) are designed as: $\mathbf{Q}_1(k) = \mathbf{Q}(k)$ and $\mathbf{Q}_2(k) = diag([100^2\ m^2,\ 100^2\ m^2,\ 100^2\ m^2,\ 1^2\ (m^2/sec^2),\ 1^2\ (m^2/sec^2),\ 1^2\ (m^2/sec^2)])$. Note that the values of $\mathbf{Q}_1(k)$ and $\mathbf{Q}_2(k)$ can be determined by examining the magnitude of the unmodeled external inputs. In this example, the magnitude of $\mathbf{Q}_1(k)$ can be chosen comparable to the magnitude of orbital perturbations such as the solar radiation pressure and third body gravity (e.g., comparable to the magnitude of $\mathbf{Q}(k)$), while the magnitude of $\mathbf{Q}_2(k)$ can be chosen using the knowledge of the bound on the magnitude of unknown impulsive maneuvers. For station keeping maneuvers of a GEO satellite, a normal range of the magnitude of impulsive maneuvers is estimated as a few m/s [99]. In this way, the values for $\mathbf{Q}_1(k)$ and $\mathbf{Q}_2(k)$ can be systematically chosen depending on the scenario considered. The simulation was performed with the initial state $\mathbf{x}(0)$ as:

$$\mathbf{x}(0) = [-2121\ m,\ 20985\ m,\ 94202\ m,\ -0.100\ m/s,\ 0.278\ m/s,\ -8.438\ m/s]^T$$

The two filters were initialized as $\hat{\mathbf{x}}^i(0) = \mathbf{x}(0)$ and $\mathbf{P}^i(0) = diag([100^2\ m^2,\ 100^2\ m^2,\ 100^2\ m^2,\ 10^2\ (m^2/sec^2),\ 10^2\ (m^2/sec^2),\ 10^2\ (m^2/sec^2)])$ for $i = 1, 2$. The performance of the proposed algorithm is compared with 1) a single EKF using only the no abrupt change model (denoted as 'EKF1'), 2) a single EKF using only the abrupt change model (denoted as 'EKF2'), and 3) IMM algorithm using both models with constant transition probabilities $\pi_{11} = 0.9$, $\pi_{12} = 0.1$, $\pi_{21} = 0.1$, and $\pi_{22} = 0.9$. Figure 4.6 shows the RMS state estimation errors (i.e., difference between the true state $\mathbf{x}(k)$ and the estimate $\hat{\mathbf{x}}(k)$) for 100 Monte Carlo simulations (since the along-track and cross-track motions are of interest in this example, the errors are computed for the $\hat{y}$ and $\hat{z}$ components). There are a few interesting observations from the results. First, when there is no impulsive maneuver, EKF1 performs best as it is designed using the no abrupt change model. On the other hand, EKF2 using the abrupt change model produces errors whose magnitudes are similar to those of the measurement noise. When the maneuver happens, EKF1 quickly diverges, causing significant estimation errors, since the no abrupt change model in EKF1 cannot account for the abrupt state jump due to the maneuver (in this example, the magnitude of the impulse is 21.376 m/s for the NS maneuver). On the other hand, both of the IMM and proposed algorithm using interacting multiple models produce smaller errors than EKF1 when the maneuver happens and than EKF2 when no maneuver happens. It is obvious that, the proposed algorithm not only performs similarly to EKF1 when there is no maneuver but also keeps the magnitude of errors as small as EKF2 does when the maneuver occurs. In Figure 4.7, the mode transition probabilities, γ_{12} and γ_{21}, of the proposed algorithm for a single run are shown. Unlike the IMM algorithm where the mode transition probabilities are fixed, those values are time-varying in the proposed algorithm as they account for the state-dependent mode transition. As shown in the figure (in this case, the satellite approaches its inclination bound at



(a) RMS position error for NS maneuver

(b) RMS velocity error for NS maneuver

Figure 4.6. Comparison of state estimation accuracy for Example 1 (100 Monte Carlo runs)

around $7,097,700$ *sec*), the mode transition probability γ_{12} (from "No abrupt change model" to "Abrupt change model") increases or decreases as the satellite approaches or moves away from the bound, respectively.

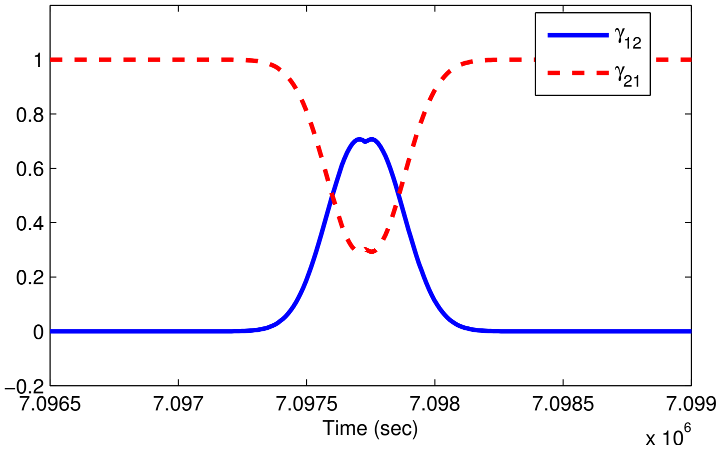


Figure 4.7. Mode transition probabilities γ_{12} and γ_{21} for a single run in Example 1

4.4.2 Example 2: Tracking of a Spacecraft Performing Orbital Transfers

In this example, we consider a spacecraft orbiting the Earth whose motion is represented in the inertial reference frame centered on the center of the Earth with the X-axis aligned with the vernal equinox line, the Z-axis aligned with the geographic North pole of the Earth, and the Y-axis completing a right-handed coordinate system (see Figure 4.8). Define a state vector $\mathbf{x}$ as $\mathbf{x} := [X \; Y \; Z \; v_X \; v_Y \; v_Z]^T$, where X, Y, and Z are the coordinates of the spacecraft's position in the reference frame, and v_X, v_Y, and v_Z are the velocities along each direction. The equations of the motion of the spacecraft performing impulsive orbital maneuvers are given as [96]:

$$\mathbf{x}(k+1) = \begin{bmatrix} X(k+1) \\ Y(k+1) \\ Z(k+1) \\ v_X(k+1) \\ v_Y(k+1) \\ v_Z(k+1) \end{bmatrix} = \begin{bmatrix} X(k) + \Delta T v_X(k) \\ Y(k) + \Delta T v_Y(k) \\ Z(k) + \Delta T v_Z(k) \\ v_X(k) - \frac{\mu}{R(k)^3} \Delta T X(k) \\ v_Y(k) - \frac{\mu}{R(k)^3} \Delta T Y(k) \\ v_Z(k) - \frac{\mu}{R(k)^3} \Delta T Z(k) \end{bmatrix} + \mathbf{w}(k) + \delta_{t,k+1} \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{I}_{3\times 3} \end{bmatrix} \mathbf{u} \tag{4.35}$$

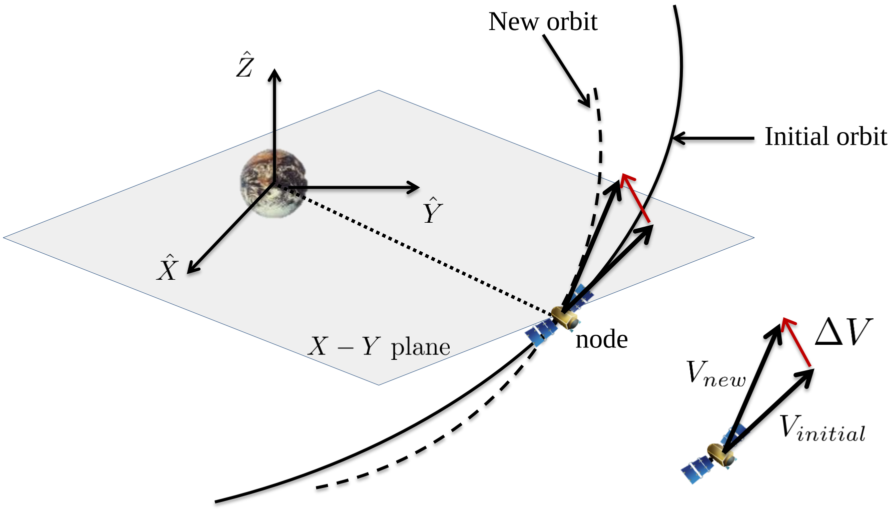


Figure 4.8. Impulsive maneuver at a node for a non-coplanar transfer (inclination change)

where $R(k) := \sqrt{X(k)^2 + Y(k)^2 + Z(k)^2}$; $\mathbf{u} := [u_X \; u_Y \; u_Z]^T$; u_X, u_Y, and u_Z denote the control inputs along each direction; and $\mathbf{w}(k) \in \mathbb{R}^6$ is a zero-mean white Gaussian noise with covariance $\mathbf{Q}(k)$. In this example, it is assumed that a ground-based radar is used to track the spacecraft and it generates the range and angle measurements as [100]:

$$\mathbf{y}(k) = \begin{bmatrix} \tan^{-1}\left(\frac{Y(k)-Y_{obs}(k)}{X(k)-X_{obs}(k)}\right) \\ \tan^{-1}\left(\frac{Z(k)-Z_{obs}(k)}{\sqrt{(X(k)-X_{obs}(k))^2+(Y(k)-Y_{obs}(k))^2}}\right) \\ \sqrt{(X(k)-X_{obs}(k))^2+(Y(k)-Y_{obs}(k))^2+(Z(k)-Z_{obs}(k))^2} \end{bmatrix} + \mathbf{v}(k) \tag{4.36}$$

where $[X_{obs}(k) \; Y_{obs}(k) \; Z_{obs}(k)]^T$ is the position of the radar with respect to the center of the Earth, and $\mathbf{v}(k) \in \mathbb{R}^3$ is a zero-mean white Gaussian noise with covariance $\mathbf{R}(k)$. In this example, we consider a non-coplanar transfer which changes the inclination of the spacecraft's orbit. For the inclination change, the impulsive maneuver usually occurs at the ascending or descending nodes of the initial orbit of the spacecraft as they are two common points in the initial orbit and new orbit after the maneuver (see Figure 4.8, where $V_{initial}$ and V_{new} denote the velocity vectors of the spacecraft right before and after the impulsive maneuver, respectively) [86]. Using this information, the jump condition is given by

$$Z(k) - Z_d = 0 \tag{4.37}$$

where $Z_d = 0$ corresponds to $\boldsymbol{\mu}$ in the general jump condition in (4.3). In the simulation, the spacecraft is assumed to move initially in a circular orbit of 7000 km around the center of the Earth with an inclination of 10 deg. The spacecraft changes its orbit to another circular orbit with the same orbital radius but with a new inclination of 30 deg. In this scenario, the magnitude of the maneuver is about 2.67 km/s. The uncertainties in the jump condition ($\mathbf{\Sigma}$ in (4.5)) is set as $\mathbf{\Sigma} = 10^2\ km^2$ and the scaling parameter η is set to 1. For simulation, the covariance of the process noise $\mathbf{Q}(k)$ of the true system (4.35) is set as $\mathbf{Q}(k) = diag([10^{-2}\ m^2,\ 10^{-2}\ m^2,\ 10^{-2}\ m^2,\ 10^{-8}\ (m^2/sec^2),\ 10^{-8}\ (m^2/sec^2),\ 10^{-8}\ (m^2/sec^2)])$. The sampling rate ΔT is 1 sec and the covariance of the measurement noise $\mathbf{R}(k)$ is set as [100]:

$$\mathbf{R}(k) = \begin{bmatrix} 0.01^2\ \deg^2 & 0 & 0 \\ 0 & 0.01^2\ \deg^2 & 0 \\ 0 & 0 & 20^2\ m^2 \end{bmatrix} \tag{4.38}$$

Figure 4.9 shows a true trajectory of the spacecraft around its ascending node where an impulsive maneuver occurs to change its inclination. Given the above simulated scenario, the proposed estimation algorithm is applied where the covariances of the process noises for both models are $\mathbf{Q}_1(k) = \mathbf{Q}(k)$ and $\mathbf{Q}_2(k) = diag([100^2\ m^2,\ 100^2\ m^2,\ 100^2\ m^2,\ 500^2\ (m^2/sec^2),\ 500^2\ (m^2/sec^2),\ 500^2\ (m^2/sec^2)])$. The simulation was performed with the initial state $\mathbf{x}(0)$ as:

$$\mathbf{x}(0) = [6696.35\,km,\ 1849.33\,km,\ -859.51\,km,\ -1.88\,km/s,\ 7.24\,km/s,\ 0.92\,km/s]^T$$

The two filters were initialized as $\hat{\mathbf{x}}^i(0) = \mathbf{x}(0)$ and $\mathbf{P}^i(0) = diag([100^2\ m^2,\ 100^2\ m^2,\ 100^2\ m^2,\ 10^2\ (m^2/sec^2),\ 10^2\ (m^2/sec^2),\ 10^2\ (m^2/sec^2)])$ for $i = 1, 2$. Figure 4.10 shows the estimation results obtained by the four different algorithms (EKF1, EKF2,

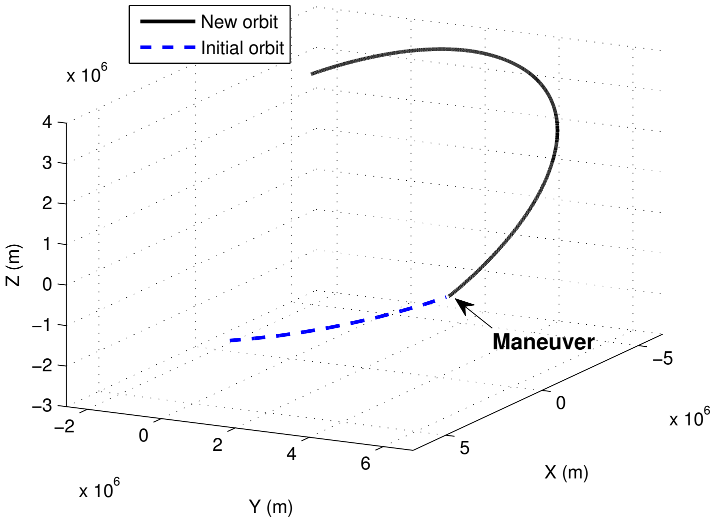
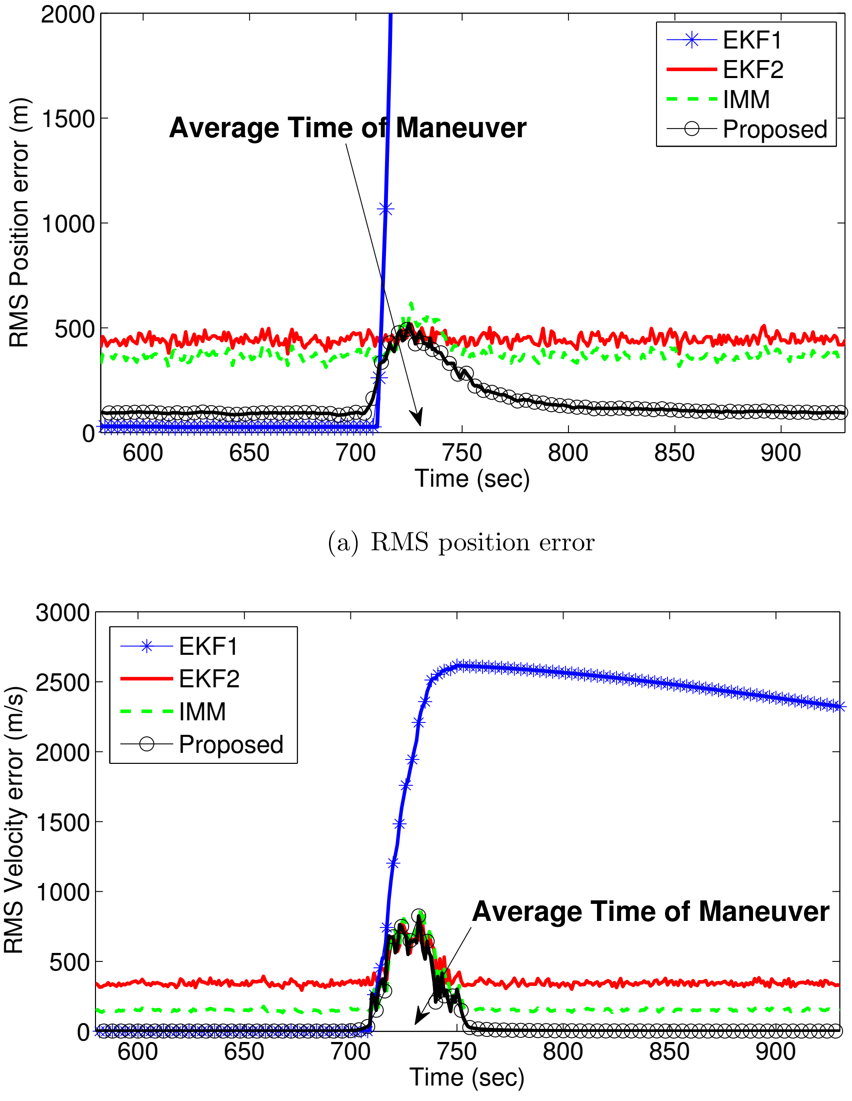


Figure 4.9. A true trajectory of the spacecraft around its ascending node

IMM, and the proposed algorithm), where the IMM algorithm uses the constant transition probabilities $\pi_{11} = 0.9$, $\pi_{12} = 0.1$, $\pi_{21} = 0.1$, and $\pi_{22} = 0.9$. Although EKF1 provides the best results when there is no maneuver, it incurs a significant estimation error when the actual maneuver happens. On the other hand, EKF2 gives estimation accuracy corresponding to the level of the measurement noise, which is not desirable. Similarly to the previous example, it is shown that the proposed algorithm outperforms all the other algorithms, providing the smallest estimation errors even when the orbital maneuver occurs. In Figure 4.11, the mode transition probabilities, γ_{12} and γ_{21}, of the proposed algorithm for a single run are shown (in this case, the spacecraft passes its ascending node around 730 *sec*). Similarly to the previous example, the mode transition probabilities are time-varying and its values are dependent on the state of the spacecraft.



(a) RMS position error

(b) RMS velocity error

Figure 4.10. Comparison of state estimation accuracy for Example 2 (100 Monte Carlo runs)

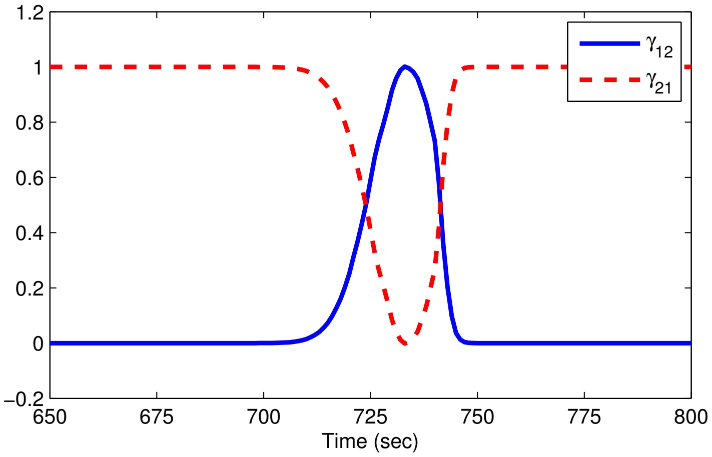


Figure 4.11. Mode transition probabilities γ_{12} and γ_{21} for a single run in Example 2

5. APPLICATION TO SPACE SITUATIONAL AWARENESS II: ANALYTICAL UNCERTAINTY PROPAGATION IN SATELLITE FORMATION FLYING

In this chapter, an important information inference problem in the SSA application, called the uncertainty propagation problem is discussed for satellite formation flying systems. The motivations and existing research for this problem are reviewed in Section 5.1. In Section 5.2, the uncertainty propagation problem is mathematically formulated for satellite relative motion near general elliptic orbits. In Section 5.3, an analytical closed-form solution to the uncertainty propagation problem is derived. Section 5.4 demonstrates the accuracy of the analytical solution with illustrative examples.

5.1 Background and Motivations

The surveillance and tracking of space objects are crucial tasks for space situational awareness with the objective of safe operation of space assets [81, 82]. Those tasks are challenging to perform since observations can be made only for a small subset of objects at a given time due to limited observational resources [40]. During the time period without measurement updates, the trajectory of an object needs to be predicted using an orbit propagation model (e.g., the simplified general perturbation model [101]) to maintain a desired level of situational awareness. However, the predicted trajectory is stochastic due to uncertainties in both the propagation model and initial conditions (i.e., knowledge of the object's state after the last measurement update) used for the prediction. Hence, it is important to keep track of statistical properties of the predicted trajectory such as the mean and covariance of the probability distribution of the object's state (known as uncertainty propagation).

The uncertainty propagation is crucial especially for satellite formation flying missions where multiple satellites are flying in a cluster to achieve certain mission objectives. There has been much interest in satellite formation flying, as it enables low-cost and high-efficient mission design such as synthetic apertures, space interferometry, and Earth mapping [102]. Many tasks of satellite formation flying, such as formation keeping [103] and formation reconfiguration [104], are performed based on the knowledge of the states of satellites within a cluster (or orbital debris moving through a cluster), which is usually updated by communication with ground control centers or between satellites. Since the availability of communication is limited due to the geometry between centers and satellites, signal loss, or signal delay, it is very important to monitor the uncertainties of neighboring satellites' states during the time intervals without communication. The uncertainty information is important as it can be used to compute the collision probability between satellites with the objective of collision avoidance [43, 44].

The uncertainty propagation problem has been posed in the literature to predict the probability density function (PDF) of the states of objects. Recently, Terejanu et al. [105] proposed a Gaussian-mixture-model approach to approximate the PDF by a finite sum of Gaussian density functions, where the weights of different components of a Gaussian-mixture model are determined by numerical optimization techniques. Similar approaches based on a Gaussian-mixture model were also developed in [42, 106, 107]. Fujimoto et al. [41] developed a state transition tensor based approach that approximates the solution to the Fokker-Plank equations associated to the two-body dynamics. These approaches are based on the nonlinear two-body dynamics (i.e., motion between a satellite and the Earth) and thus need to be modified to exploit features of the relative motion dynamics between satellites.

There are a few conditions to be considered in the uncertainty propagation for satellite formation flight. Most importantly, it should be noted that the uncertainty propagation is carried out without ground contacts, and thus necessary computation needs to be performed by onboard computers of each satellite within a cluster. Moreover, the computation should be fast enough in order to facilitate the collision probability calculation, which is crucial for initiating responsive collision avoidance maneuvers. These conditions imply that the algorithm for the uncertainty propagation should be computationally efficient so that satellite onboard computers having low computational capability can perform the necessary computations in a timely manner. To address this issue, analytical approaches have been proposed for the uncertainty propagation in satellite relative motion [108, 109]. In these approaches, closed-form solutions were developed based on the linearized relative motion dynamics and white Gaussian process noises, which analytically compute the mean and covariance of the PDF of the satellite's relative state without any numerical integration. These solutions, however, rely on an assumption of circular chief orbits, and thus use simplified relative dynamic models (e.g., the Clohessy-Wiltshire (CW) equations) that facilitate finding the analytical solutions. Hence, their accuracies significantly decrease when applied to more general cases where the chief orbit is elliptic, which limits the applicability of the solutions. In this sense, it is necessary to develop a new analytical solution that can account for satellite relative motions along general elliptic orbits. However, this involves the use of more complicated relative motion dynamics, and it makes the corresponding uncertainty propagation problem difficult to solve.

In this study, an analytical closed-form solution is developed for the uncertainty propagation in satellite relative motion near general elliptic orbits. The well-known Tschauner-Hempel (TH) equations, widely used to describe the relative motions of the deputy satellite with respect to the chief satellite moving in an elliptic orbit [110, 111], are used as a relative dynamic model. Based on the TH equations and the assumption of white Gaussian process noise, the uncertainty propagation problem is formulated to compute the mean and covariance of the PDF of the deputy's relative state. The evolution of the mean and covariance matrix is governed by a linear time-

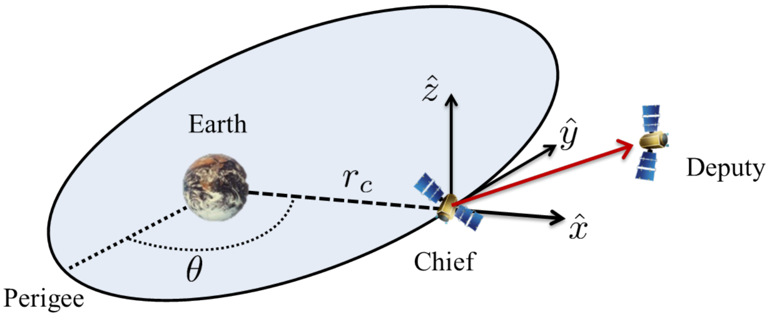


Figure 5.1. Relative motion in the local-vertical-local-horizontal (LVLH) frame

varying differential equation, whose solution requires the integration of the quadratic function of the inverse of the fundamental matrix associated to the TH equations. The difficulties in evaluating the integration are overcome by the introduction of an adjoint system to the TH equations and the binomial series expansion. From the analytical solution obtained, we can evaluate the uncertainty of the states of the deputy satellite at any time of interest without any numerical integration.

5.2 Problem Formulation

5.2.1 Relative Motion Dynamics

We consider the relative motion of a deputy satellite with respect to a chief satellite that moves around the Earth in an elliptic orbit. Under the assumption of linearized differential gravitational acceleration, the relative motion of the deputy can be described by [112]:

$$\begin{aligned}\ddot{x}(t) - 2\dot{\theta}(t)\dot{y}(t) - \ddot{\theta}(t)y(t) - \dot{\theta}(t)^2x(t) - \frac{2\mu x(t)}{r_c(t)^3} &= u_x(t)\\ \ddot{y}(t) + 2\dot{\theta}(t)\dot{x}(t) + \ddot{\theta}(t)x(t) - \dot{\theta}(t)^2y(t) + \frac{\mu y(t)}{r_c(t)^3} &= u_y(t)\\ \ddot{z}(t) + \frac{\mu z(t)}{r_c(t)^3} &= u_z(t)\end{aligned} \tag{5.1}$$

where $x(t)$, $y(t)$, and $z(t)$ are the coordinates of the deputy's position at time t with respect to the chief in the local-vertical-local-horizontal (LVLH) frame (see Figure 5.1); the dot $(\cdot)$ represents the differentiation with respect to time; $r_c(t)$ is the orbital radius of the chief from the center of the Earth; $u_x(t)$, $u_y(t)$, and $u_z(t)$ represent the external disturbances applied to the deputy; $\theta(t)$ refers to the true anomaly of the chief; and μ is the gravitational constant of the Earth. By defining the deputy's state vector $\mathbf{X}(t)$ and the disturbance vector $\mathbf{u}(t)$ as $\mathbf{X}(t) \equiv [x(t)\ y(t)\ z(t)\ \dot{x}(t)\ \dot{y}(t)\ \dot{z}(t)]^T \in \mathbb{R}^6$ and $\mathbf{u}(t) \equiv [u_x(t)\ u_y(t)\ u_z(t)]^T \in \mathbb{R}^3$, respectively, (5.1) can be represented in the state-space form as:

$$\dot{\mathbf{X}}(t) = \mathbf{A}(t)\mathbf{X}(t) + \mathbf{B}\mathbf{u}(t) \tag{5.2}$$

where

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{A}_1(t) & \mathbf{A}_2(t) \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix}$$
$$\mathbf{A}_1(t) = \begin{bmatrix} \dot{\theta}(t)^2 + \frac{2\mu}{r_c(t)^3} & \ddot{\theta}(t) & 0 \\ -\ddot{\theta}(t) & \dot{\theta}(t)^2 - \frac{\mu}{r_c(t)^3} & 0 \\ 0 & 0 & -\frac{\mu}{r_c(t)^3} \end{bmatrix}, \quad \mathbf{A}_2(t) = \begin{bmatrix} 0 & 2\dot{\theta}(t) & 0 \\ -2\dot{\theta}(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.3}$$

Define $\boldsymbol{\Upsilon}(t)$ and $\boldsymbol{\Gamma}(t, t_0) \equiv \boldsymbol{\Upsilon}(t)\boldsymbol{\Upsilon}(t_0)^{-1}$ as the fundamental matrix and state transition matrix associated to $\mathbf{A}(t)$, respectively. It is noted that (5.2) is linear in terms of $\mathbf{X}(t)$ and $\mathbf{u}(t)$, but the coefficients are functions of $r_c(t)$ and $\theta(t)$ that evolve by [113]

$$\begin{aligned} \ddot{r}_c(t) &= r_c(t)\dot{\theta}(t)^2 - \frac{\mu}{r_c(t)^2} \\ \ddot{\theta}(t) &= -\frac{2\dot{r}_c(t)\dot{\theta}(t)}{r_c(t)} \end{aligned} \tag{5.4}$$

So, to compute $\boldsymbol{\Gamma}(t, t_0)$, (5.4) needs to be integrated numerically, which would hamper finding a closed-form solution to the uncertainty propagation later in this paper. However, the equations can be converted to a simpler form by changing the independent variable from time t to the true anomaly θ and by introducing the following transformation [114]:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} \equiv \dot{\theta}^{1/2} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5.5}$$

This transformation was introduced by Tschauner and Hempel [114] to facilitate the development of an analytical solution to the satellite relative motion along general elliptic Earth orbits. With this transformation, (5.2) is now represented by another linear state space form called the Tschauner-Hempel (TH) equations as [114]:

$$\tilde{\mathbf{X}}'(\theta) = \tilde{\mathbf{A}}(\theta)\tilde{\mathbf{X}}(\theta) + \tilde{\mathbf{B}}(\theta)\mathbf{u}(\theta) \tag{5.6}$$

where ($'$) denotes derivative with respect to the true anomaly θ and

$$\begin{gathered} \tilde{\mathbf{X}}(\theta) \equiv [\tilde{x}(\theta) \;\; \tilde{y}(\theta) \;\; \tilde{z}(\theta) \;\; \tilde{x}'(\theta) \;\; \tilde{y}'(\theta) \;\; \tilde{z}'(\theta)]^T \\ \mathbf{u}(\theta) \equiv [u_x(\theta) \; u_y(\theta) \; u_z(\theta)]^T \\ \tilde{\mathbf{A}}(\theta) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \tilde{\mathbf{A}}_1(\theta) & \tilde{\mathbf{A}}_2 \end{bmatrix}, \quad \tilde{\mathbf{B}}(\theta) = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \frac{1}{\dot{\theta}(\theta)^{3/2}}\mathbf{I}_{3\times3} \end{bmatrix} \\ \tilde{\mathbf{A}}_1(\theta) = \begin{bmatrix} 3/\rho(\theta) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \tilde{\mathbf{A}}_2 = \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{gathered} \tag{5.7}$$

where $\rho(\theta) \equiv 1 + e\cos\theta$, e is the eccentricity of the chief's orbit, $\dot{\theta}(\theta) = \frac{\rho(\theta)^2\mu^2}{h^3}$ [86], and h is the angular momentum of the chief's orbit. The transformation matrix $\mathbf{T}(\theta)$ between $\mathbf{X}(t)$ and $\tilde{\mathbf{X}}(\theta)$ is then obtained as [115]:

$$\begin{gathered} \mathbf{X}(t(\theta)) = \mathbf{T}(\theta)\tilde{\mathbf{X}}(\theta) \\ \mathbf{T}(\theta) = \begin{bmatrix} \frac{h^{3/2}}{\rho(\theta)\mu}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \frac{\mu e \sin\theta}{h^{3/2}}\mathbf{I}_{3\times3} & \frac{\rho(\theta)\mu}{h^{3/2}}\mathbf{I}_{3\times3} \end{bmatrix} \end{gathered} \tag{5.8}$$

where the independent variables θ and t are interrelated through Kepler's equation (that is, there is an one-to-one map $\mathcal{G}$ such that $\theta = \mathcal{G}(t)$ and thus $t = \mathcal{G}^{-1}(\theta)$).

Whenever θ and t appear together in the following equations throughout the paper, it should be understood that θ and t are related to each other by $\mathcal{G}$. Note that the transformed system (5.6) is much simpler than (5.2) in that its coefficients are now varying explicitly with the independent variable θ, which renders the integration of (5.4) unnecessary and thus facilitates the development of analytical solutions. In fact, for the system matrix $\tilde{\mathbf{A}}(\theta)$ in (5.6), the closed-form solution of fundamental matrix $\mathbf{\Phi}(\theta)$ is derived as [115]:

$$\mathbf{\Phi}(\theta) = \begin{bmatrix} 0 & -c & 0 & -s & 3esI-2 & 0 \\ 1 & s(1+1/\rho) & 0 & -c(1+1/\rho) & 3\rho^2 I & 0 \\ 0 & 0 & c/\rho & 0 & 0 & s/\rho \\ 0 & -c' & 0 & -s' & 3e(s'I + s/\rho^2) & 0 \\ 0 & 2c-e & 0 & 2s & 3(1-2esI) & 0 \\ 0 & 0 & -s/\rho & 0 & 0 & c/\rho \end{bmatrix} \equiv \begin{bmatrix} \phi_{3\times 6} \\ \phi'_{3\times 6} \end{bmatrix} \tag{5.9}$$

where $c \equiv \rho \cos\theta$, $s \equiv \rho \sin\theta$, and

$$I \equiv \int_{\theta_0}^{\theta} \frac{1}{\rho(\sigma)^2} d\sigma \tag{5.10}$$

Define $\mathbf{\Psi}(\theta, \theta_0) \equiv \mathbf{\Phi}(\theta)\mathbf{\Phi}(\theta_0)^{-1}$ as the state transition matrix associated to $\tilde{\mathbf{A}}(\theta)$. Then, from the transformation in (5.8), we have

$$\mathbf{\Gamma}(t(\theta), t_0) = \mathbf{T}(\theta)\mathbf{\Psi}(\theta, \theta_0)\mathbf{T}(\theta_0)^{-1} \tag{5.11}$$

where $t_0 = \mathcal{G}^{-1}(\theta_0)$. The TH equations and associated fundamental matrix will be used to derive analytical solutions to the uncertainty propagation between satellites flying in formation near elliptic orbits.

5.2.2 Uncertainty Propagation Problem

Assume that $\mathbf{u}(t)$ accounts for uncertainties exerted on the relative motion dynamics, (5.2), and is modeled as a white Gaussian noise process of zero mean and strength $\mathbf{U}$ ($\mathbf{U}$ is symmetric and positive semidefinite), i.e.,

$$\begin{aligned} \mathbb{E}\{\mathbf{u}(t)\} &= \mathbf{0} \\ \mathbb{E}\{\mathbf{u}(t)\mathbf{u}(t')^T\} &= \mathbf{U}\delta(t-t') \end{aligned} \tag{5.12}$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation of a random variable, and δ is the Dirac delta. In addition, assume that the initial state $\mathbf{X}(t_0)$ is independent of $\mathbf{u}(t)$ and has a Gaussian distribution with mean $\mathbf{X}_0$ and covariance $\mathbf{P}_0$ as:

$$\mathbf{X}(t_0) \sim \mathcal{N}(\mathbf{X}(t_0); \mathbf{X}_0, \mathbf{P}_0) \tag{5.13}$$

The assumption of the zero-mean white Gaussian noise process and initial Gaussian distribution is reasonable in the sense that the orbit determination techniques usually produce estimation results in terms of Gaussian distributions, and dominant stochastic disturbances such as aerodynamics and solar pressure are typically modeled as white Gaussian process noises [116]. The uncertainty propagation problem is then defined to compute the PDF of $\mathbf{X}(t)$ (now $\mathbf{X}(t)$ is a random vector) whose evolution is governed by (5.2) with the uncertainties in $\mathbf{u}(t)$ and $\mathbf{X}(t_0)$. In the stochastic setting, (5.2) can be represented as a linear stochastic differential equation as [117]:

$$d\mathbf{X}(t) = \mathbf{A}(t)\mathbf{X}(t)dt + \mathbf{B}d\boldsymbol{\beta}(t) \tag{5.14}$$

where $\boldsymbol{\beta}(t)$ is a vector-valued Brownian motion process associated with $\mathbf{u}(t)$, which has the following properties:

$$\begin{aligned} \mathbb{E}\left\{\boldsymbol{\beta}(t)\right\} &= \mathbf{0}, \quad \forall t > t_0 \\ \mathbb{E}\left\{[\boldsymbol{\beta}(t_2)-\boldsymbol{\beta}(t_1)][\boldsymbol{\beta}(t_2)-\boldsymbol{\beta}(t_1)]^T\right\} &= \int_{t_1}^{t_2} \mathbf{U}dt, \quad \forall t_1, t_2, \quad t_1 < t_2 \end{aligned} \tag{5.15}$$

The solution to (5.14) is the stochastic process $\mathbf{X}(t)$ given by

$$\mathbf{X}(t) = \boldsymbol{\Gamma}(t,t_0)\mathbf{X}(t_0) + \int_{t_0}^{t} \boldsymbol{\Gamma}(t,\tau)\mathbf{B}d\boldsymbol{\beta}(\tau) \tag{5.16}$$

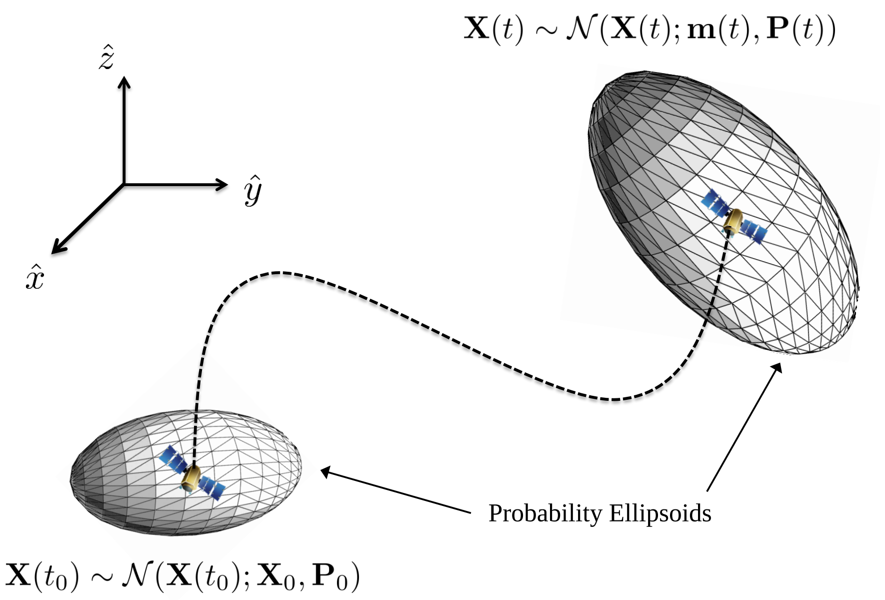

$\mathbf{X}(t) \sim \mathcal{N}(\mathbf{X}(t); \mathbf{m}(t), \mathbf{P}(t))$
$\hat{z}$
$\hat{y}$
$\hat{x}$
Probability Ellipsoids
$\mathbf{X}(t_0) \sim \mathcal{N}(\mathbf{X}(t_0); \mathbf{X}_0, \mathbf{P}_0)$

Figure 5.2. Evolution of the PDF of $\mathbf{X}(t)$ (only position is considered)

It is well-known that given the linear system with the zero mean white Gaussian noise process $\mathbf{u}(t)$ and the initial Gaussian distribution $\mathbf{X}(t_0)$, $\mathbf{X}(t)$ is a Gaussian random variable for any fixed t. That is, the statistical properties of $\mathbf{X}(t) \sim \mathcal{N}(\mathbf{X}(t); \mathbf{m}(t), \mathbf{P}(t))$ are characterized only by its mean $\mathbf{m}(t)$ and covariance $\mathbf{P}(t)$, which can be computed as [117]:

$$\begin{aligned} \mathbf{m}(t) &= \mathbf{\Gamma}(t, t_0)\mathbf{X}_0 \\ \mathbf{P}(t) &= \mathbf{\Gamma}(t, t_0)\mathbf{P}_0\mathbf{\Gamma}(t, t_0)^T + \int_{t_0}^{t} \mathbf{\Gamma}(t, \tau)\mathbf{B}\mathbf{U}\mathbf{B}^T\mathbf{\Gamma}(t, \tau)^T d\tau \end{aligned} \tag{5.17}$$

The overall framework of the uncertainty propagation problem is illustrated in Figure 5.2, where a probability ellipsoid refers to a boundary within which the state vector is contained with a specific probability. Its shape and size are determined by the state covariance matrix $\mathbf{P}(t)$ [118]. Note that, as discussed earlier, finding analytical expressions for $\mathbf{\Upsilon}(t)$ and $\mathbf{\Gamma}(t, \tau)$ is not straightforward, and thus it is challenging to find a closed-form solution to (5.17). To overcome this difficulty, we propose to use the transformed system (5.6) that is much simpler and preserves linearity, so that we can easily exploit the well-known characteristics of the linear Gaussian model. To do that, we need to convert the distributions of $\mathbf{u}(t)$ and $\mathbf{X}(t_0)$, which are defined in the time domain, into the θ domain to find the statistics of $\mathbf{u}(\theta)$ and $\tilde{\mathbf{X}}(\theta_0)$ as follows.

First, from the fact that a linear transformation of a Gaussian random vector has also a Gaussian distribution [79], the PDF of the transformed initial state vector $\tilde{\mathbf{X}}(\theta_0)$ can be computed as:

$$\tilde{\mathbf{X}}(\theta_0) \sim \mathcal{N}(\tilde{\mathbf{X}}(\theta_0); \tilde{\mathbf{X}}_0, \tilde{\mathbf{P}}_0) \tag{5.18}$$

where

$$\begin{aligned} \tilde{\mathbf{X}}_0 &= \mathbf{T}(\theta_0)^{-1}\mathbf{X}_0 \\ \tilde{\mathbf{P}}_0 &= \mathbf{T}(\theta_0)^{-1}\mathbf{P}_0(\mathbf{T}(\theta_0)^{-1})^T \end{aligned} \tag{5.19}$$

Second, the statistics of $\mathbf{u}(\theta)$ can be determined by examining the properties of its associated Brownian motion $\tilde{\boldsymbol{\beta}}(\theta)$, which is obtained by the transformation of $\boldsymbol{\beta}(t)$ to the θ domain. The properties of $\tilde{\boldsymbol{\beta}}(\theta)$ can be completely characterized by its diffusion $\tilde{\mathbf{U}}(\theta)$ such that

$$\begin{aligned} \mathbb{E}\left\{\tilde{\boldsymbol{\beta}}(\theta)\right\} &= \mathbf{0}, \quad \forall\theta > \theta_0 \\ \mathbb{E}\left\{[\tilde{\boldsymbol{\beta}}(\theta_2) - \tilde{\boldsymbol{\beta}}(\theta_1)][\tilde{\boldsymbol{\beta}}(\theta_2) - \tilde{\boldsymbol{\beta}}(\theta_1)]^T\right\} &= \int_{\theta_1}^{\theta_2} \tilde{\mathbf{U}}(\theta)d\theta, \quad \forall\theta_1, \theta_2, \quad \theta_1 < \theta_2 \end{aligned} \tag{5.20}$$

So, the task of finding the statistics of $\mathbf{u}(\theta)$ is equivalent to the determination of $\tilde{\mathbf{U}}(\theta)$. We consider a stochastic differential equation for the transformed system (5.6) as:

$$d\tilde{\mathbf{X}}(\theta) = \tilde{\mathbf{A}}(\theta)\tilde{\mathbf{X}}(\theta)d\theta + \tilde{\mathbf{B}}(\theta)d\tilde{\boldsymbol{\beta}}(\theta) \tag{5.21}$$

The solution to (5.21) is given by the stochastic process $\tilde{\mathbf{X}}(\theta)$ as:

$$\tilde{\mathbf{X}}(\theta) = \boldsymbol{\Psi}(\theta, \theta_0)\tilde{\mathbf{X}}(\theta_0) + \int_{\theta_0}^{\theta} \boldsymbol{\Psi}(\theta, \sigma)\tilde{\mathbf{B}}(\sigma)d\tilde{\boldsymbol{\beta}}(\sigma) \tag{5.22}$$

Similarly to (5.16), $\tilde{\mathbf{X}}(\theta)$ is a Gaussian random vector for any fixed θ, and its mean and covariance are computed as:

$$\begin{aligned} \tilde{\mathbf{m}}(\theta) &= \boldsymbol{\Psi}(\theta, \theta_0)\tilde{\mathbf{X}}_0 \\ \tilde{\mathbf{P}}(\theta) &= \boldsymbol{\Psi}(\theta, \theta_0)\tilde{\mathbf{P}}_0\boldsymbol{\Psi}(\theta, \theta_0)^T + \int_{\theta_0}^{\theta} \boldsymbol{\Psi}(\theta, \sigma)\tilde{\mathbf{B}}(\sigma)\tilde{\mathbf{U}}(\sigma)\tilde{\mathbf{B}}(\sigma)^T\boldsymbol{\Psi}(\theta, \sigma)^T d\sigma \end{aligned} \tag{5.23}$$

Note that the two random vectors $\mathbf{X}(t)$ and $\tilde{\mathbf{X}}(\theta)$ are related to each other through the linear transformation $\mathbf{T}(\theta)$ (5.8). That is, similarly to (5.19), we have the following identity:

$$\mathbf{P}(t(\theta)) = \mathbf{T}(\theta)\tilde{\mathbf{P}}(\theta)\mathbf{T}(\theta)^T \tag{5.24}$$

From the identity, we have

$$\begin{aligned}
\mathbf{\Gamma}(t,t_0)\mathbf{P}_0\mathbf{\Gamma}(t,t_0)^T + \int_{t_0}^{t} \mathbf{\Gamma}(t,\tau)\mathbf{B}\mathbf{U}\mathbf{B}^T\mathbf{\Gamma}(t,\tau)^T d\tau& \\
= \mathbf{T}(\theta)\mathbf{\Psi}(\theta,\theta_0)\tilde{\mathbf{P}}_0\mathbf{\Psi}(\theta,\theta_0)^T\mathbf{T}(\theta)^T& \\
+ \mathbf{T}(\theta)\left(\int_{\theta_0}^{\theta} \mathbf{\Psi}(\theta,\sigma)\tilde{\mathbf{B}}(\sigma)\tilde{\mathbf{U}}(\sigma)\tilde{\mathbf{B}}(\sigma)^T\mathbf{\Psi}(\theta,\sigma)^T d\sigma\right)\mathbf{T}(\theta)^T&
\end{aligned} \tag{5.25}$$

Note that in (5.25), we can easily see from (5.11) and (5.19) that the first terms in both sides are equivalent and thus canceled out. Then, (5.25) can be simplified as:

$$\begin{aligned}
&\int_{t_0}^{t} \mathbf{\Gamma}(t,\tau)\mathbf{B}\mathbf{U}\mathbf{B}^T\mathbf{\Gamma}(t,\tau)^T d\tau \\
&= \mathbf{T}(\theta)\mathbf{T}(\theta)^{-1}\left(\int_{\theta_0}^{\theta} \mathbf{\Gamma}(t(\theta),\tau(\sigma))\mathbf{T}(\sigma)\tilde{\mathbf{B}}(\sigma)\tilde{\mathbf{U}}(\sigma)\tilde{\mathbf{B}}(\sigma)^T\mathbf{T}(\sigma)^T\mathbf{\Gamma}(t(\theta),\tau(\sigma))^T d\sigma\right) \\
&\quad \times (\mathbf{T}(\theta)^{-1})^T\mathbf{T}(\theta)^T \\
&= \int_{\theta_0}^{\theta} \mathbf{\Gamma}(t(\theta),\tau(\sigma))\mathbf{T}(\sigma)\tilde{\mathbf{B}}(\sigma)\tilde{\mathbf{U}}(\sigma)\tilde{\mathbf{B}}(\sigma)^T\mathbf{T}(\sigma)^T\mathbf{\Gamma}(t(\theta),\tau(\sigma))^T d\sigma \\
&= \int_{t_0}^{t} \mathbf{\Gamma}(t,\tau)\frac{1}{\dot{\theta}}\mathbf{B}\tilde{\mathbf{U}}(\sigma(\tau))\frac{1}{\dot{\theta}}\mathbf{B}^T\mathbf{\Gamma}(t,\tau)^T\dot{\theta}d\tau \\
&= \int_{t_0}^{t} \mathbf{\Gamma}(t,\tau)\mathbf{B}\frac{1}{\dot{\theta}}\tilde{\mathbf{U}}(\sigma(\tau))\mathbf{B}^T\mathbf{\Gamma}(t,\tau)^T d\tau
\end{aligned} \tag{5.26}$$

where $\mathbf{T}(\sigma)\tilde{\mathbf{B}}(\sigma)$ in the third last line is computed to $\frac{1}{\dot{\theta}}\mathbf{B}$ in the second last line using (5.3), (5.7), and (5.8). Equation (5.26) implies that the diffusion $\tilde{\mathbf{U}}(\theta)$ can be computed as:

$$\tilde{\mathbf{U}}(\theta) = \dot{\theta}(\theta)\mathbf{U} \tag{5.27}$$

which then can be used to evaluate (5.23) to characterize the random vector $\tilde{\mathbf{X}}(\theta)$. The mean $\mathbf{m}(t)$ and covariance $\mathbf{P}(t)$ of the random vector $\mathbf{X}(t)$ are then obtained by

$$\begin{aligned} \mathbf{m}(t) &= \mathbf{T}(\theta(t))\tilde{\mathbf{m}}(\theta(t)) \\ \mathbf{P}(t) &= \mathbf{T}(\theta(t))\tilde{\mathbf{P}}(\theta(t))\mathbf{T}(\theta(t))^T \end{aligned} \tag{5.28}$$

That is, the uncertainty propagation problem in this paper becomes equivalent to computing $\tilde{\mathbf{m}}(\theta)$ and $\tilde{\mathbf{P}}(\theta)$ in (5.23), which requires the integration that is quadratic in terms of $\mathbf{\Psi}(\theta, \sigma)$. The next section will show how to derive an analytical solution to (5.23).

5.3 Analytical Solution to Uncertainty Propagation

5.3.1 Derivation of the State Transition Matrix $\mathbf{\Psi}$

To compute (5.23), we need to derive the state transition matrix $\mathbf{\Psi}(\theta, \sigma)$ in an analytical form. Note that the derivation of $\mathbf{\Psi}(\theta, \sigma)$ requires the calculation of the inverse of the fundamental matrix $\mathbf{\Phi}(\theta)^{-1}$, which is challenging to perform. To deal with this difficulty, we propose to introduce an adjoint system associated with the system matrix $\tilde{\mathbf{A}}(\theta)$ given by

$$\mathbf{\Lambda}'(\theta) = -\tilde{\mathbf{A}}(\theta)^T \mathbf{\Lambda}(\theta) \tag{5.29}$$

where $\mathbf{\Lambda}(\theta) \equiv [\lambda_1(\theta)\ \lambda_2(\theta)\ \lambda_3(\theta)\ \lambda_4(\theta)\ \lambda_5(\theta)\ \lambda_6(\theta)]^T \in \mathbb{R}^6$ is the adjoint state of $\tilde{\mathbf{X}}(\theta)$. Denote the fundamental matrix of the adjoint system (5.29) as $\mathbf{\Pi}(\theta)$ (i.e., $\mathbf{\Lambda}(\theta) = \mathbf{\Pi}(\theta)\mathbf{\Pi}(\theta_0)^{-1}\mathbf{\Lambda}(\theta_0)$). The following identity shows the relation between $\mathbf{\Pi}(\theta)$ and $\mathbf{\Phi}(\theta)$ [119]:

$$\begin{aligned} \frac{d}{d\theta}[\mathbf{\Pi}(\theta)^T\mathbf{\Phi}(\theta)] &= \mathbf{\Pi}(\theta)'^T\mathbf{\Phi}(\theta) + \mathbf{\Pi}(\theta)^T\mathbf{\Phi}(\theta)' \\ &= -[\tilde{\mathbf{A}}(\theta)^T\mathbf{\Pi}(\theta)]^T\mathbf{\Phi}(\theta) + \mathbf{\Pi}(\theta)^T\tilde{\mathbf{A}}(\theta)\mathbf{\Phi}(\theta) \\ &= \mathbf{0} \end{aligned} \tag{5.30}$$

Equation (5.30) indicates that the multiplication $\mathbf{\Pi}(\theta)^T\mathbf{\Phi}(\theta)$ is constant, and thus we have $\mathbf{\Pi}(\theta)^T\mathbf{\Phi}(\theta) = \mathbf{C}$ where $\mathbf{C}$ is a constant matrix. From this, the inverse of the fundamental matrix $\mathbf{\Phi}(\theta)^{-1}$ can be computed as follows:

$$\mathbf{\Phi}(\theta)^{-1} = \mathbf{C}^{-1}\mathbf{\Pi}(\theta)^T \tag{5.31}$$

It is noted that the task of computing the inverse of the fundamental matrix is now reduced to finding the fundamental matrix of the adjoint system $\mathbf{\Pi}(\theta)$ and computing the inverse of the constant matrix $\mathbf{C}$, which are much easier to perform. The fundamental matrix of the adjoint system $\mathbf{\Pi}(\theta)$ can be found in the literature as [120]:

$$\begin{aligned}\mathbf{\Pi}(\theta) &= \begin{bmatrix} \tilde{\mathbf{A}}_2\boldsymbol{\phi}(\theta) - \boldsymbol{\phi}'(\theta) \\ \boldsymbol{\phi}(\theta) \end{bmatrix} \\ &= \begin{bmatrix} 2 & 3s/\rho & 0 & -3c/\rho - e & 3I(3\rho - 1 + e^2) - 3es/\rho^2 & 0 \\ 0 & e & 0 & 0 & 1 & 0 \\ 0 & 0 & s/\rho & 0 & 0 & -c/\rho \\ 0 & -c & 0 & -s & 3esI - 2 & 0 \\ 1 & s(1+1/\rho) & 0 & -c(1+1/\rho) & 3\rho^2 I & 0 \\ 0 & 0 & c/\rho & 0 & 0 & s/\rho \end{bmatrix}\end{aligned} \tag{5.32}$$

Using $\mathbf{\Phi}(\theta)$ and $\mathbf{\Pi}(\theta)$ given in (5.9) and (5.32), respectively, the constant matrix $\mathbf{C} = \mathbf{\Pi}(\theta)^T\mathbf{\Phi}(\theta)$ and its inverse $\mathbf{C}^{-1}$ can be computed as:

$$\begin{aligned}
\mathbf{C} &= \mathbf{\Pi}(\theta)^T\mathbf{\Phi}(\theta)\\
&= \begin{bmatrix}\boldsymbol{\phi}(\theta)^T\tilde{\mathbf{A}}_2^T - \boldsymbol{\phi}'(\theta)^T & \boldsymbol{\phi}(\theta)^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\phi}(\theta)\\ \boldsymbol{\phi}'(\theta)\end{bmatrix}\\
&= \boldsymbol{\phi}(\theta)^T\tilde{\mathbf{A}}_2^T\boldsymbol{\phi}(\theta) - \boldsymbol{\phi}'(\theta)^T\boldsymbol{\phi}(\theta) + \boldsymbol{\phi}(\theta)^T\boldsymbol{\phi}'(\theta)\\
&= \begin{bmatrix}0 & -e & 0 & 0 & -1 & 0\\ e & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & -1 & 0 & 0 & -e & 0\\ 1 & 0 & 0 & e & 0 & 0\\ 0 & 0 & -1 & 0 & 0 & 0\end{bmatrix}
\end{aligned} \tag{5.33}$$

$$\mathbf{C}^{-1} = \frac{1}{1-e^2}\begin{bmatrix}0 & -e & 0 & 0 & 1 & 0\\ e & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & e^2-1\\ 0 & 1 & 0 & 0 & -e & 0\\ -1 & 0 & 0 & e & 0 & 0\\ 0 & 0 & 1-e^2 & 0 & 0 & 0\end{bmatrix} \tag{5.34}$$

Note that, given e, $\mathbf{C}$ in (5.33) is computed as a constant matrix. Using $\mathbf{C}^{-1}$ and $\mathbf{\Pi}(\theta)$, the inverse of the fundamental matrix $\mathbf{\Phi}(\theta)^{-1}$ can be obtained using (5.31), and thus we can analytically derive the state transition matrix $\mathbf{\Psi}(\theta, \sigma)$ without the explicit inversion of $\mathbf{\Phi}(\theta)$ [115, 121].

5.3.2 Evaluation of the Covariance $\tilde{\mathbf{P}}(\theta)$

Given the state transition matrix $\mathbf{\Psi}(\theta, \sigma)$, the evaluation of $\tilde{\mathbf{m}}(\theta)$ in (5.23) can be easily performed, and thus we focus on the computation of $\tilde{\mathbf{P}}(\theta)$, which involves the integration of a quadratic form of $\mathbf{\Psi}(\theta,\sigma)$. Using (5.31), $\tilde{\mathbf{P}}(\theta)$ in (5.23) can be evaluated as:

$$\begin{aligned}
\tilde{\mathbf{P}}(\theta) &= \mathbf{\Psi}(\theta,\theta_0)\tilde{\mathbf{P}}_0\mathbf{\Psi}(\theta,\theta_0)^T + \int_{\theta_0}^{\theta}\mathbf{\Psi}(\theta,\sigma)\tilde{\mathbf{B}}(\sigma)\tilde{\mathbf{U}}(\sigma)\tilde{\mathbf{B}}(\sigma)^T\mathbf{\Psi}(\theta,\sigma)^T d\sigma \\
&= \mathbf{\Phi}(\theta)\mathbf{\Phi}(\theta_0)^{-1}\tilde{\mathbf{P}}_0(\mathbf{\Phi}(\theta)\mathbf{\Phi}(\theta_0)^{-1})^T \\
&\quad + \mathbf{\Phi}(\theta)\left[\int_{\theta_0}^{\theta}\mathbf{\Phi}(\sigma)^{-1}\tilde{\mathbf{B}}(\sigma)\tilde{\mathbf{U}}(\sigma)\tilde{\mathbf{B}}(\sigma)^T(\mathbf{\Phi}(\sigma)^{-1})^T d\sigma\right]\mathbf{\Phi}(\theta)^T \\
&= \mathbf{\Phi}(\theta)\mathbf{\Phi}(\theta_0)^{-1}\tilde{\mathbf{P}}_0(\mathbf{\Phi}(\theta)\mathbf{\Phi}(\theta_0)^{-1})^T \\
&\quad + \mathbf{\Phi}(\theta)\mathbf{C}^{-1}\left[\int_{\theta_0}^{\theta}\mathbf{\Pi}(\sigma)^T\tilde{\mathbf{B}}(\sigma)\tilde{\mathbf{U}}(\sigma)\tilde{\mathbf{B}}(\sigma)^T\mathbf{\Pi}(\sigma)d\sigma\right](\mathbf{C}^{-1})^T\mathbf{\Phi}(\theta)^T \\
&= \mathbf{\Phi}(\theta)\mathbf{\Phi}(\theta_0)^{-1}\tilde{\mathbf{P}}_0(\mathbf{\Phi}(\theta)\mathbf{\Phi}(\theta_0)^{-1})^T \\
&\quad + \mathbf{\Phi}(\theta)\mathbf{C}^{-1}\left[\int_{\theta_0}^{\theta}\frac{1}{\dot{\theta}(\sigma)^3}\boldsymbol{\phi}(\sigma)^T\tilde{\mathbf{U}}(\sigma)\boldsymbol{\phi}(\sigma)d\sigma\right](\mathbf{C}^{-1})^T\mathbf{\Phi}(\theta)^T
\end{aligned} \tag{5.35}$$

Since the first term in the right-hand side of (5.35) can be easily evaluated using $\tilde{\mathbf{P}}_0$, $\mathbf{\Phi}(\theta)$ and $\mathbf{\Phi}(\theta)^{-1}$, the only remaining difficulty is to compute the integral in the second term, which can be rewritten using (5.27) as:

$$\begin{aligned}
\int_{\theta_0}^{\theta}\frac{1}{\dot{\theta}(\sigma)^3}\boldsymbol{\phi}(\sigma)^T\tilde{\mathbf{U}}(\sigma)\boldsymbol{\phi}(\sigma)d\sigma &= \int_{\theta_0}^{\theta}\frac{1}{\dot{\theta}(\sigma)^2}\boldsymbol{\phi}(\sigma)^T\mathbf{U}\boldsymbol{\phi}(\sigma)d\sigma \\
&= \frac{h^6}{\mu^4}\int_{\theta_0}^{\theta}\frac{1}{\rho(\sigma)^4}\boldsymbol{\phi}(\sigma)^T\mathbf{U}\boldsymbol{\phi}(\sigma)d\sigma
\end{aligned} \tag{5.36}$$

It is noted that the direct integration of (5.36) is cumbersome due to the presence of $1/\rho(\sigma)^4$ and I (in $\boldsymbol{\phi}(\sigma)$, see (5.10)) in the integrand. This difficulty can be dealt with by introducing the change of variable from the true anomaly θ to the eccentric anomaly E, which can be performed using the following identities [86]:

$$\begin{aligned}
d\theta &= \frac{\sqrt{1-e^2}}{1-e\cos E}dE \\
\cos\theta &= \frac{\cos E - e}{1-e\cos E} \\
\sin\theta &= \frac{\sqrt{1-e^2}\sin E}{1-e\cos E} \\
\rho(\theta) = 1 + e\cos\theta &= \frac{1-e^2}{1-e\cos E} \\
I = (1-e^2)^{-3/2}&(E - e\sin E - E_0 + e\sin E_0)
\end{aligned} \tag{5.37}$$

where E_0 is the initial eccentric anomaly corresponding to θ_0. Define a matrix-valued function $\mathbf{K}(E)$ as the integrand after the change of variable using (5.37) such that

$$\int_{\theta_0}^{\theta} \frac{1}{\rho(\sigma)^4} \boldsymbol{\phi}(\sigma)^T \mathbf{U} \boldsymbol{\phi}(\sigma) d\sigma = \int_{E_0}^{E} \mathbf{K}(\eta) d\eta \tag{5.38}$$

It can be easily found that almost all elements of $\mathbf{K}(E)$ are composed of polynomials of $\cos E$, $\sin E$, and E, which can be analytically integrated (the elements of $\mathbf{K}(E)$ are presented in the Appendix). However, a few of the elements have the term of $(1 - e\cos E)^{-1}$, which hampers the analytical integration. To deal with this difficulty, we propose to use the binomial series expansion as:

$$(1 - e\cos E)^{-1} = 1 + e\cos E + e^2\cos^2 E + e^3\cos^3 E + \cdots \tag{5.39}$$

Since $0 \leq e < 1$ and thus $|e\cos E| < 1$, the series in (5.39) converges absolutely. For the integration, we use N terms of the series, where N can be chosen based on the desired accuracy. Since $(1 - e\cos E)^{-1}$ is now represented as a polynomial of $\cos E$, all the elements in $\mathbf{K}(E)$ can be analytically integrated, resulting in a closed-form solution. Once the closed-form solution is obtained, the distribution of $\tilde{\mathbf{X}}(\theta)$ (and thus $\mathbf{X}(t)$) at any θ (also t) of interest can be analytically described without any numerical integration, which reduces computational loads significantly. The overall procedure for the analytical uncertainty propagation is summarized in Table 5.1.

5.4 Numerical Simulation

In this section, the accuracy of the analytical solution is demonstrated with illustrative numerical examples. To evaluate its accuracy, we first analytically compute $\mathbf{m}(t)$ and $\mathbf{P}(t)$ using the analytical solution (we will denote them as $\mathbf{m}^a(t)$ and $\mathbf{P}^a(t)$), and compare them to the distribution computed from Monte Carlo simulation. We also compare $\mathbf{m}^a(t)$ and $\mathbf{P}^a(t)$ with those computed through the direct numerical integration of (5.23) (we will denote them as $\mathbf{m}^n(t)$ and $\mathbf{P}^n(t)$).

In the simulation, the semi-major axis of the chief's orbit is assumed to be $7,000$ km, and the gravitational constant is given as $\mu = 398600.4418$ km^3/sec^2. Different

Table 5.1. Procedure to find analytical solution to the uncertainty propagation problem

1.Input

1) The distribution $\mathbf{X}(t_0) \sim \mathcal{N}(\mathbf{X}(t_0); \mathbf{X}_0, \mathbf{P}_0)$ at the initial time t_0.

2) Time t_f for which the distribution will be computed.

2.UncertaintyPropagation

A. Compute the true anomalies θ_0 and θ_f corresponding to t_0 and t_f (using Kepler's equation), respectively.

B. Transform $\mathbf{X}(t_0) \sim \mathcal{N}(\mathbf{X}(t_0); \mathbf{X}_0, \mathbf{P}_0)$ to $\tilde{\mathbf{X}}(\theta_0) \sim \mathcal{N}(\tilde{\mathbf{X}}(\theta_0); \tilde{\mathbf{X}}_0, \tilde{\mathbf{P}}_0)$ using (5.19).

C. Compute the closed-form of $\boldsymbol{\Psi}(\theta, \sigma)$ using the results in Section 5.3.1.

D. Evaluate the integral $\int_{\theta_0}^{\theta_f} \frac{1}{\rho(\sigma)^4} \boldsymbol{\phi}(\sigma)^T \mathbf{U} \boldsymbol{\phi}(\sigma) d\sigma$ in (5.36).

a. Compute E_0 and E_f corresponding to θ_0 and θ_f, respectively.

b. Evaluate the integral $\int_{E_0}^{E_f} \mathbf{K}(\eta) d\eta$ in (5.38).

E. Compute $\tilde{\mathbf{m}}(\theta_f)$ using (5.23).

F. Compute $\tilde{\mathbf{P}}(\theta_f)$ using (5.35).

G. Transform $\tilde{\mathbf{m}}(\theta_f)$ and $\tilde{\mathbf{P}}(\theta_f)$ to $\mathbf{m}(t_f)$ and $\mathbf{P}(t_f)$, respectively using (5.28).

3.Output

The distribution of $\mathbf{X}(t_f)$ at time t_f, $\mathbf{X}(t_f) \sim \mathcal{N}(\mathbf{X}(t_f); \mathbf{m}(t_f), \mathbf{P}(t_f))$.

values for the eccentricity of the chief's orbit e are considered for the simulation. For the initial distribution $\mathbf{X}(t_0) \sim \mathcal{N}(\mathbf{X}(t_0); \mathbf{X}_0, \mathbf{P}_0)$, we have

$$\mathbf{X}_0 = [20 \text{ m}, 15 \text{ m}, 20 \text{ m}, 0.1 \text{ m/s}, 0.1 \text{ m/s}, 0.1 \text{ m/s}]^T \tag{5.40}$$

$$\mathbf{P}_0 = \begin{bmatrix} 10^{-3}\ \text{m}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-3}\ \text{m}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-3}\ \text{m}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-5}\ \text{m}^2/\text{s}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-5}\ \text{m}^2/\text{s}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-5}\ \text{m}^2/\text{s}^2 \end{bmatrix} \tag{5.41}$$

The strength $\mathbf{U}$ of $\mathbf{u}(t)$ is set as [108]:

$$\mathbf{U} = \begin{bmatrix} 10^{-5} & 0 & 0 \\ 0 & 10^{-5} & 0 \\ 0 & 0 & 10^{-5} \end{bmatrix} (\text{m}/\text{s}^2)^2 \tag{5.42}$$

In the analysis, we focus on the uncertainties in the position of the deputy (x, y, z), which are important as they are directly related to collision probability computation. For the purpose of visualization, we present the results either in the x-y plane or y-z plane. To obtain statistical information tailored for each plane, we partition $\mathbf{P}^a(t)$ to obtain covariance matrices $\mathbf{P}^a_{xy}(t)$ and $\mathbf{P}^a_{yz}(t)$ containing information only for x and y, or y and z, respectively (see Figure 5.3). Uncertainties in the velocity of the deputy can also be analyzed in the similar way.

From the eigenvectors and eigenvalues of $\mathbf{P}^a_{xy}(t)$ and $\mathbf{P}^a_{yz}(t)$, we can determine the principle directions and the magnitude of the covariances along these directions, from which probability ellipsoids can be drawn. Figure 5.4 shows the simulation results when $e = 0.2$. In the figure, the probability ellipsoids computed from $\mathbf{P}^a_{xy}(t)$ and $\mathbf{P}^a_{yz}(t)$ (obtained using $N = 5$ expansion terms) are shown with blue lines, which contain 80% of the distribution, for every 500 seconds after the initial time $t_0 = 0$ sec. For comparison, the probability ellipsoids computed from the analytical solutions using the CW equation [108] are also presented with green dashed lines. Note that, given the covariance matrix, the probability ellipsoid can be computed for any number of percent. In this example, we choose 80% for the appropriate visualization. It is clear that the proposed analytical solution (blue line) can accurately predict the probability

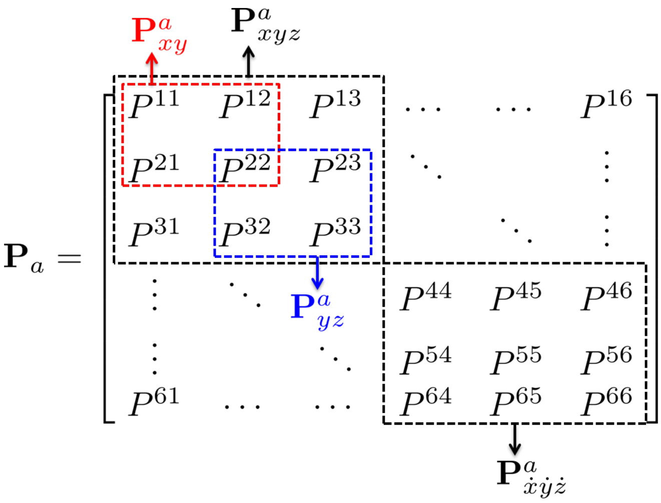

$\mathbf{P}^a_{xy}$
$\mathbf{P}^a_{xyz}$
$\mathbf{P}_a =$
P^{11} P^{12} P^{13} $\cdots$ $\cdots$ P^{16}
P^{21} P^{22} P^{23}
P^{31} P^{32} P^{33}
$\mathbf{P}^a_{yz}$
P^{44} P^{45} P^{46}
P^{54} P^{55} P^{56}
P^{61} $\cdots$ $\cdots$ P^{64} P^{65} P^{66}
$\mathbf{P}^a_{\dot{x}\dot{y}\dot{z}}$

Figure 5.3. Partition of $\mathbf{P}^a$ into $\mathbf{P}^a_{xy}$, $\mathbf{P}^a_{yz}$, $\mathbf{P}^a_{xyz}$, and $\mathbf{P}^a_{\dot{x}\dot{y}\dot{z}}$, where P^{ij} is the element in the ith row and jth column of $\mathbf{P}^a$

distribution of the states (red dots) obtained from Monte Carlo simulation. On the other hand, the solutions using the CW equations fail to capture the distribution. This is because the CW equations assume a circular chief orbit, and thus when applied to the case where the chief orbit is elliptic, their accuracy significantly decreases.

Although the comparison to Monte Carlo simulation provides a qualitative way to check the accuracy of the analytical solution, it is also necessary to examine the accuracy quantitatively. In this sense, we compare $\mathbf{P}^a(t)$ (obtained from the analytical solution) directly to $\mathbf{P}^n(t)$ (obtained from the numerical integration of (5.23)). For the appropriate comparison, we first partition $\mathbf{P}^a(t)$ and $\mathbf{P}^n(t)$ into $\mathbf{P}^a_{xyz}(t)$ and $\mathbf{P}^a_{\dot{x}\dot{y}\dot{z}}(t)$, and $\mathbf{P}^n_{xyz}(t)$ and $\mathbf{P}^n_{\dot{x}\dot{y}\dot{z}}(t)$ containing statistical information on the position and velocity, respectively (see Figure 5.3). We then use the Frobenius norm to measure a distance $d(\cdot,\cdot)$ between the two arbitrary matrices with the same size, $\mathbf{O} = \{o^{ij}\} \in \mathbb{R}^{s\times l}$ and $\mathbf{Q} = \{q^{ij}\} \in \mathbb{R}^{s\times l}$ as:

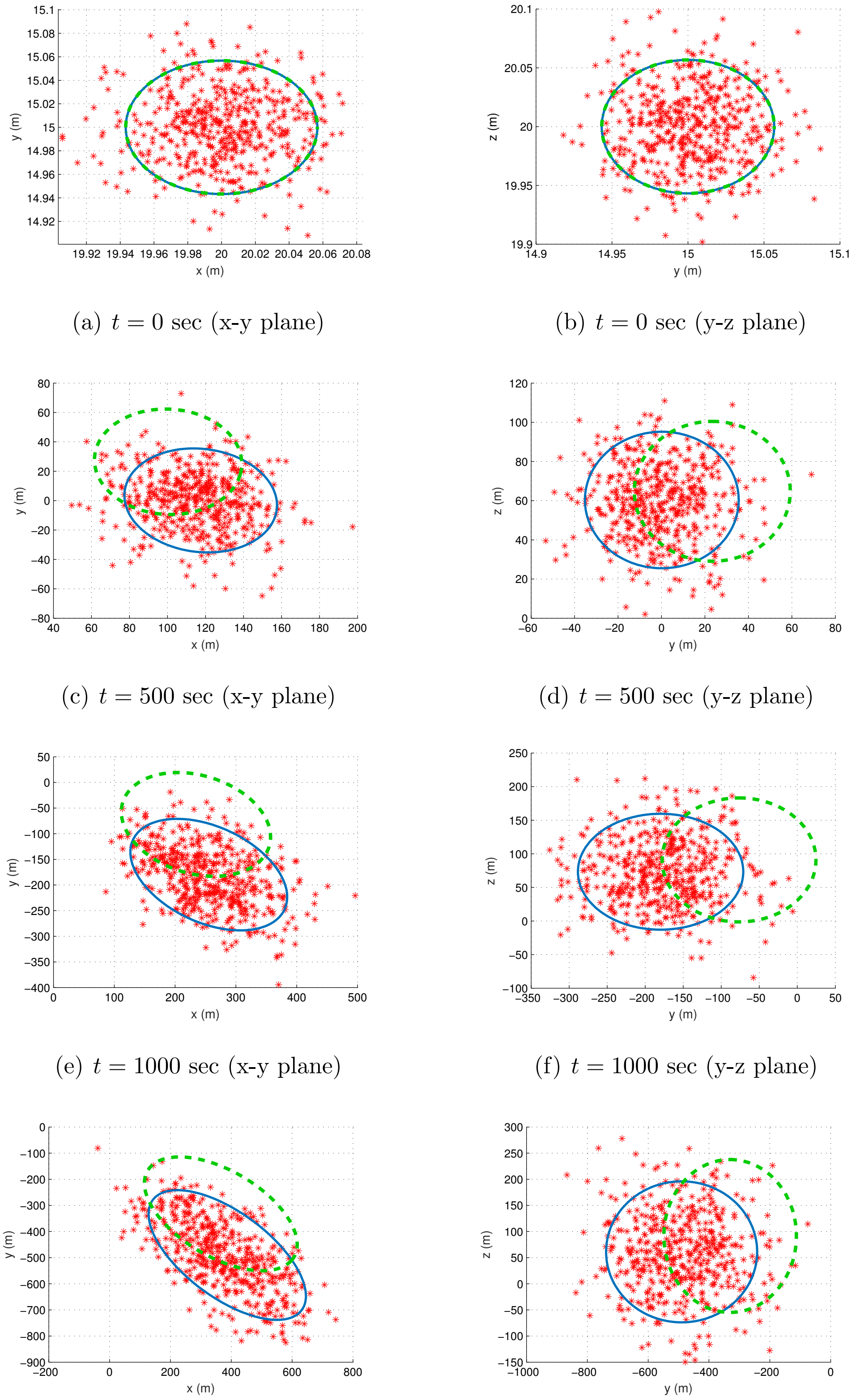
$$d(\mathbf{O},\mathbf{Q}) = \sqrt{\sum_{i=1}^{s}\sum_{j=1}^{l}(o^{ij} - q^{ij})^2} \tag{5.43}$$

Table 5.2 shows the distances $d(\mathbf{P}^a_{xyz}(t), \mathbf{P}^n_{xyz}(t))$ and $d(\mathbf{P}^a_{\dot{x}\dot{y}\dot{z}}(t), \mathbf{P}^n_{\dot{x}\dot{y}\dot{z}}(t))$, where $\mathbf{P}^a(t)$ and $\mathbf{P}^n(t)$ are computed using several values for e and N, with the same initial conditions $\mathbf{X}_0$ and $\mathbf{P}_0$ in (5.40) and (5.41) and $\mathbf{U}$ in (5.42). The distances are evaluated at $t = 1,500$ sec. From the table, it is found that the analytical solution converges to the numerical solution as it uses more terms in the binomial series expansion. In addition, it is observed that more terms in the binomial series are needed to attain a certain level of accuracy as e increases. This is reasonable since the magnitude of higher-order terms (i.e., the magnitude of truncation error) in the binomial series is proportional to the magnitude of e. Note that when $e = 0$, the proposed analytical solution becomes exact (i.e., no approximation), and thus the measured distance (i.e., the difference between the analytical solution and numerical solution) is attributed only to inherent numerical errors caused by numerical integration. This implies that the accuracy obtained for the case when $e = 0$ in Table 5.2 can be used as reference for determining how good a given approximation is.

As shown in Figure 5.4 and Table 5.2, it is demonstrated that the analytical solution developed in this paper is highly accurate and its accuracy can be effectively managed by the number of the binomial series N. Therefore, this analytical solution provides an accurate tool for characterizing the uncertainties in the satellite relative motion without the computational cost caused by numerical integration.

Table 5.2. Comparison between $\mathbf{P}^a$ and $\mathbf{P}^n$ using distance measure d

		$d_1 \equiv d(\mathbf{P}^a_{xyz}, \mathbf{P}^n_{xyz})$ (m^2) and $d_2 \equiv d(\mathbf{P}^a_{\dot{x}\dot{y}\dot{z}}, \mathbf{P}^n_{\dot{x}\dot{y}\dot{z}})$ (m^2/s^2)						
		$N=2$	$N=3$	$N=4$	$N=5$	$N=6$	$N=7$	$N=8$
$e=0$	d_1	7.66×10^{-4}	7.66×10^{-4}	7.66×10^{-4}	7.66×10^{-4}	7.66×10^{-4}	7.66×10^{-4}	7.66×10^{-4}
	d_2	7.89×10^{-10}	7.89×10^{-10}	7.89×10^{-10}	7.89×10^{-10}	7.89×10^{-10}	7.89×10^{-10}	7.89×10^{-10}
$e=0.1$	d_1	499.94	37.42	3.02	0.25	0.02	1.48×10^{-3}	7.87×10^{-4}
	d_2	2.65×10^{-4}	2.27×10^{-5}	2.01×10^{-6}	1.83×10^{-7}	1.72×10^{-8}	2.05×10^{-9}	9.12×10^{-10}
$e=0.2$	d_1	1402.01	199.36	31.83	5.28	0.91	0.16	0.03
	d_2	9.54×10^{-4}	1.62×10^{-4}	2.88×10^{-5}	5.22×10^{-6}	9.63×10^{-7}	1.80×10^{-7}	3.42×10^{-8}
$e=0.3$	d_1	2288.99	443.91	108.06	26.43	6.80	1.79	0.48
	d_2	1.87×10^{-3}	4.69×10^{-4}	1.24×10^{-4}	3.37×10^{-5}	9.28×10^{-6}	2.58×10^{-6}	7.27×10^{-7}
$e=0.4$	d_1	3222.34	722.20	253.44	81.02	28.47	10.04	3.64
	d_2	2.80×10^{-3}	9.07×10^{-4}	3.21×10^{-4}	1.15×10^{-4}	4.21×10^{-5}	1.56×10^{-5}	5.82×10^{-6}



(a) $t = 0$ sec (x-y plane)

(b) $t = 0$ sec (y-z plane)

(c) $t = 500$ sec (x-y plane)

(d) $t = 500$ sec (y-z plane)

(e) $t = 1000$ sec (x-y plane)

(f) $t = 1000$ sec (y-z plane)

(g) $t = 1500$ sec (x-y plane)

(h) $t = 1500$ sec (y-z plane)

Figure 5.4. Evolution of the probability ellipsoid (blue line: the proposed analytical solution, green dashed line: the analytical solution using the CW equations, and red dots: Monte Carlo simulation using 500 samples)

6. APPLICATION TO PILOT-AUTOMATION INTERACTION ISSUE DETECTION

In this chapter, we develop information inference algorithms for pilot-automation integrated systems to identify safe-critical issues due to malicious interaction between the pilot and automation. In Section 6.1, the motivations of this research are presented. Section 6.2 proposes intent-based behavior models for the pilot and automation. Based on the models, a real-time pilot-automation interaction issue detection algorithm is developed in Section 6.3. In Section 6.4, the proposed algorithm is demonstrated with two illustrative pilot-automation interaction issue examples.

[122–143]

6.1 Background and Motivations

The operation of aircraft has become increasingly automated through the development of advanced autopilot systems [46]. The control inputs (signals) necessary to manage the aircraft's behaviors, which were manually generated by pilots in the past, are now produced mainly by automation through various flight modes in the autopilot systems. The tasks of the pilots have become more supervisory such as to decide which flight mode is appropriate for the aircraft's desired motion, and to engage a flight mode with necessary information (e.g., target altitude, target heading, target airspeed, etc.). Although these automated systems have enabled accurate and efficient operations of the aircraft, its complexity has caused a new safety concern called 'human-automation mode confusion'. The mode confusion (also called the lack of mode awareness or automation surprise [122]) happens when the pilot becomes confused about the current and future status (i.e., operating flight mode) of the automation and interact with it incorrectly. As a consequence, the automation behaves inconsistently with the pilot's goal and fails to produce the desired aircraft's motion as intended by the pilot. It is found that many incidents and accidents reported in the NASA's Aviation Safety Reporting System (ASRS) have been caused by the mode confusion problem [47,123]. Because the mode confusion can cause unexpected changes in the aircraft's altitude and airspeed, which should be maintained within a safe operating range, it is important to identify the mode confusion in a timely manner to prevent undesirable accidents/incidents. Thus, real-time mode confusion detection is a crucial problem for safety monitoring of the aircraft.

In the literature, it is found that most mode confusion detection algorithms have been designed to work off-line. Many of them have been developed in the context of 1) verifying a given automation system, or 2) identifying anomalies in the recorded flight data, which are not appropriate for real-time safety monitoring. Firstly, formal verification approaches have been used for the off-line verification of a given automation system in order to identify any faulty logic or elements of the system that can cause the mode confusion to the pilot [124–128]. The verification process requires examining every possible operating scenario, which is computationally expensive for on-line applications. The other limitation of these approaches is that, they require the accurate knowledge on pilot's decision making behavior (i.e., the sophisticated mental model of pilots), which are usually not available or hard to obtain. Also, since they use formal language [129,130] defined on the discrete states (e.g., automation logic), it is hard to incorporate the continuous states (i.e., continuous motion of an aircraft such as altitude change, speed change, heading change, etc.) in their model. Since the consequences of the mode confusion are usually reflected in the continuous states (e.g., undesired altitude change), the continuous states should be exploited to increase the effectiveness in detecting the mode confusion.

Secondly, in the anomaly detection approaches, the mode confusion is regarded as an anomaly in the flight data which rarely occurs compared to the normal aircraft operation. At a raw data level, anomalies are defined as a set of data points that are different from the majority of the given data set, where the data points belonging to the majority are assumed as the consequences of the normal operation [131–135]. Therefore, the aircraft's anomalous behaviors due to the mode confusion are reflected as anomalies in the data set recorded by the onboard flight data recorder. Several data mining and machine learning techniques have been used to identify anomalies in the recorded flight data such as clustering [136], classification [137], and regression [138]. However, anomalies identified by these algorithms can be attributed not only to the mode confusion but also to other factors such as the aircraft's mechanical failure or noises in the data set. Hence, an additional step to differentiate the mode confusion-related anomalies from the others is necessary, which is usually done by domain experts. In addition, these anomaly detection approaches can be applied only when enough amount of flight data is collected and thus, these approaches are not appropriate for the real-time detection of the mode confusion.

It is noted that when the mode confusion happens, the automation's behavior and the pilot's objective are not consistent [48, 139]. So, in order to detect the mode confusion, the goal of the automation based on which the aircraft's actual behavior is generated needs to be compared with that of the pilot. To represent the automation's and the pilot's goals, we propose to use a concept of 'intent set' whose elements ('intent') abstractly describe the behavior of the automation and the pilot (e.g., 'climb', 'descent', 'constant altitude', etc.). By using an intent set, the complex behavior of the pilot and the automation can be succinctly represented in terms of 'intent', and therefore, can be effectively compared to detect the mode confusion. However, the intent is not directly available, and not known ahead of time, since neither the flight management system nor flight data recording system provides explicit intent information. Therefore, the intent needs to be inferred from the available information such as measurements (e.g., the pilot's input, the aircraft's flight modes, etc.) and the automation's and the pilot's models describing how the measurements are generated, which is challenging. To deal with the necessary steps for the intent inference explained above, we propose a hybrid system and a discrete event system as the models for the automation and the pilot, respectively, and based

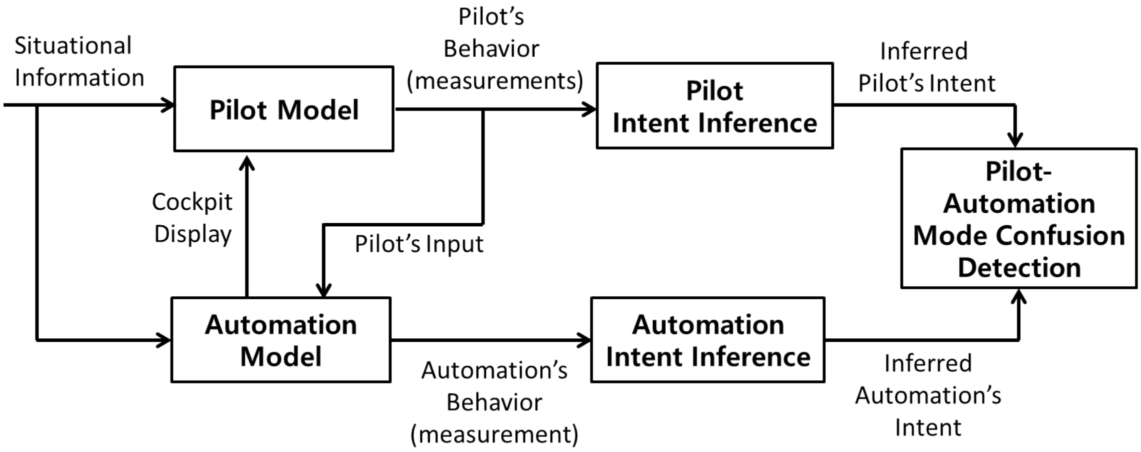

Situational Information
Pilot Model
Pilot's Behavior (measurements)
Pilot Intent Inference
Inferred Pilot's Intent
Pilot-Automation Mode Confusion Detection
Cockpit Display
Pilot's Input
Automation Model
Automation Intent Inference
Automation's Behavior (measurement)
Inferred Automation's Intent

Figure 6.1. Overall framework for pilot-automation mode confusion detection

on those models, develop an intent inference algorithm. The inferred intents of the automation and the pilot are then monitored in real-time and compared to identify any inconsistency which indicates the occurrence of the mode confusion. The overall framework of the proposed algorithm is presented in Figure 6.1.

6.2 Intent-Based Pilot and Automation Behavior Modeling

The underlying idea of our proposed algorithm is that the mode confusion is reflected in the flight measurement data in which the actual behavior of the automation is targeted towards a goal which is different from that of the pilot. This implies that the mode confusion can be detected by comparing the goal (intent) of the pilot to that of the automation. To this end, we first construct an intent set consisting of finite flight intents that can abstractly describe the behaviors of the automation and the pilot. We then model the pilot and the automation to characterize the generation of the measurement data under their specific flight intents. Using the models and the intent sets, the intents of the pilot and the automation can be inferred by estimating for what intents the observed measurement data is generated.

Table 6.1. Construction of dimension-wise intent sets

Dimension	**Intents**
Vertical, i_V	$i_V \in$ {Climb, Descend, Constant Altitude}
Lateral, i_L	$i_L \in$ {Turn Left, Turn Right, Constant Heading}
Speed, i_S	$i_S \in$ {Accelerate, Decelerate, Constant Speed}

6.2.1 Design of Intent Set

Based on our survey on the mode confusion incidents and accidents, we have found that the mode confusion can be represented as a conflict between the intents of the automation and the pilot at a tactical level of the aircraft's behavior (e.g., altitude change, heading change, and speed change) [48, 140]. In general, the automation system consists of vertical flight modes (governing vertical motion), lateral flight modes (governing lateral motion), and auto-throttle flight modes (governing airspeed) which the pilot can control via the Mode Control Panel (MCP). Therefore, it is reasonable to assume that the pilot has intents along the three dimensions (vertical, lateral, and speed) and issues commands accordingly to control the flight modes in each dimension. Motivated by this, an intent set along each dimension is constructed as presented in Table 6.1. In Table 6.1, i_V, i_L, and i_S denote vertical flight intent, lateral flight intent, and speed flight intent, respectively. The overall intent (denoted by I) is then represented as a 3-tuple of the dimension-wise intents as $I = (i_V, i_L, i_S) \in \mathcal{I}$, where $\mathcal{I} = \{I^1, I^2, \dots, I^{27}\}$ is the total intent set whose elements are combinations of the three dimension-wise intents (e.g., $I^1 = (\text{Climb}, \text{Turn Left}, \text{Accelerate})$,..., $I^{27} = (\text{Constant Altitude}, \text{Constant Heading}, \text{Constant Speed})$). By representing the automation's and the pilot's intents using the same intent set $\mathcal{I}$, the consistency between the complex behaviors of the automation and the pilot can be easily monitored.

6.2.2 Modeling of the Automation: Hybrid System

Hybrid systems refer to dynamical systems which contain interacting continuous states and discrete states (or modes) [25, 26]. The dynamics of the automation (aircraft) can be represented as a hybrid system since the aircraft's behavior consists of both the logical behavior (discrete transitions between flight modes (discrete states)) and the physical behavior (the aircraft's continuous motion (continuous state) corresponding to a specific flight mode) [32,144]. Let us define the continuous state, the sensor measurements vector, and the discrete state at a given time k as $x(k) \in \mathbf{X} \subset \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$, and $q(k) \in \mathcal{Q} = \{1, 2, \ldots, n_q\}$, respectively. For each $q(k)$, the continuous state dynamics and the measurement equation are given by

$$\begin{aligned} x(k+1) &= A_{q(k)}x(k) + w_{q(k)}(k) \\ y(k) &= C_{q(k)}x(k) + u_{q(k)}(k) \end{aligned} \tag{6.1}$$

where $w_{q(k)}(k)$ and $u_{q(k)}(k)$ are zero-mean white Gaussian noises with covariance $Q_{q(k)}$ and $R_{q(k)}$, respectively. The discrete state (flight mode) transitions are governed by the mode transition function $\gamma : \mathcal{Q} \times \mathbf{X} \times \mathbf{Z_c} \times \mathbf{Z_d} \to \mathcal{Q}$ as:

$$q(k+1) = \gamma(q(k), x(k), z_c(k), z_d(k)) \tag{6.2}$$

where $z_c(k) \in \mathbf{Z_c} \subset \mathbb{R}^{n_{z_c}}$ and $z_d(k) \in \mathbf{Z_d} = \{1, 2, \ldots, n_{z_d}\}$ are the pilot's continuous (e.g., MCP target value setting) and discrete control inputs (e.g., mode switch engagement, i.e., each element in $\mathbf{Z_d}$ represents a specific switch configuration), respectively. The mode transition function γ is mathematically defined based on a set of guard conditions $G(i, j) \subset \mathbf{X} \times \mathbf{Z_c} \times \mathbf{Z_d}$, $\forall i, j \in \mathcal{Q}$ as:

$$\gamma(i, x, z_c, z_d) = j \quad \text{if } [x^T \; z_c^T \; z_d]^T \in G(i, j) \tag{6.3}$$

These guard conditions can be formulated based on a given automation logic (i.e., autopilot) to describe the flight mode transitions. The flight mode transitions can be classified as: 1) forced transition, and 2) autonomous transition. The forced transition is triggered when the pilot engages a specific flight mode via an appropriate mode switch setting (e.g., the Altitude Hold mode is switched to the Vertical Speed (V/S) mode when the pilot pushes the V/S mode button with a higher target altitude setting). In this forced transition case, the guard condition is given by

$$G(i,j) = \{[x^T \ z_c^T \ z_d]^T | \\ z_d \in \{\text{mode switch configuration triggering the transition from mode } i \text{ to mode } j\}\} \tag{6.4}$$

The autonomous transition happens without the pilot's external input when the continuous state x satisfies certain conditions. For example, the aircraft climbing to a target altitude in the V/S mode automatically transitions from the V/S mode to the Altitude Capture mode when the current altitude approaches the target altitude in order to prepare a smooth level-off. In the autonomous transition case, the guard condition is generally given by

$$G(i,j) = \{[x^T \ z_c^T \ z_d]^T | L_{x,ij}x + L_{\theta,ij}\theta \leq 0\} \tag{6.5}$$

where $L_{x,ij} \in \mathbb{R}^{l \times n}$ and $L_{\theta,ij} \in \mathbb{R}^{l \times n_{z_c}}$ are constant matrices characterizing the autonomous transition, and $\theta \in \mathbb{R}^{n_{z_c}}$ is a random vector with probability density function (pdf):

$$p(\theta) = \mathcal{N}(\theta; \bar{\theta}, \Sigma_\theta) \tag{6.6}$$

which describes the uncertainties of the guard condition due to noise in the sensor measurements (where $\mathcal{N}$ represents the Gaussian distribution with the mean $\bar{\theta}$ and the covariance Σ_θ). For the above example of the altitude capturing, the guard condition can be represented as $|H_f - h(k)| \leq \delta$, where H_f is the target altitude (i.e., pilot's continuous input), $h(k)$ is the actual altitude of the aircraft (i.e., continuous state), and δ is a parameter denoting the distance within which the capture happens. This condition can be written as:

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} h(k) + \begin{bmatrix} 1 \\ -1 \end{bmatrix} (H_f - \delta) \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.7}$$

The illustration of the hybrid system model for the automation is given in Figure 6.2 with three flight modes. The behavior of the automation described by $x(k)$ and $q(k)$

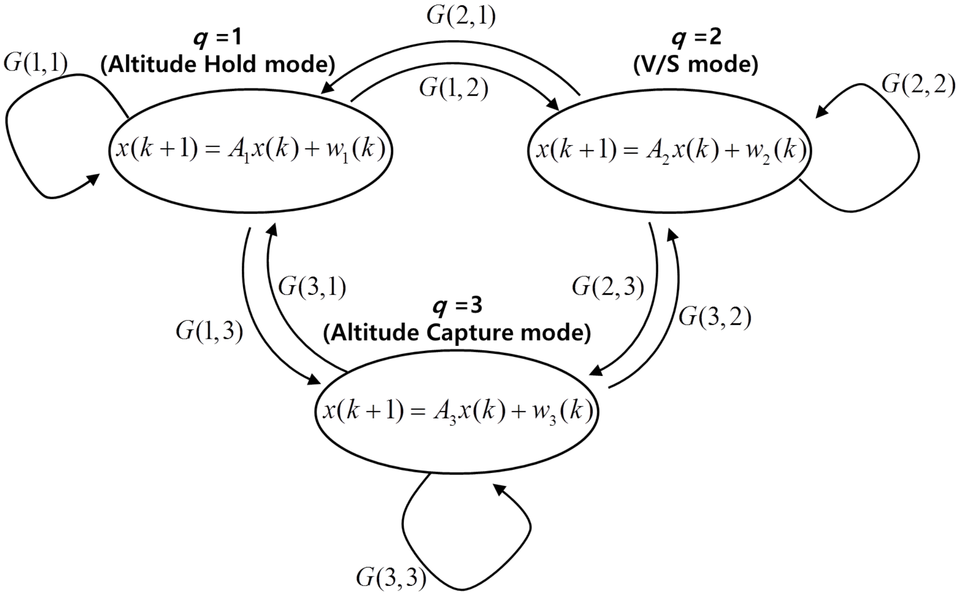


Figure 6.2. Illustration of the automation as hybrid system model

is then abstracted to the automation's flight intent $I_a(k) \in \mathcal{I}$ through the following intent mapping $\mu : \mathbf{X} \times \mathcal{Q} \rightarrow \mathcal{I}$:

$$I_a(k) = \mu(x(k), q(k)) \tag{6.8}$$

For example, when the aircraft is operating in the V/S mode with a positive altitude change rate, the corresponding intent of the automation is 'Climb'. Using this intent representation, we can succinctly describe the aircraft's current motion as one of the flight intents in the intent set $\mathcal{I}$.

6.2.3 Modeling of the Pilot: Discrete Event System

When it comes to the pilot model, it should be noted that the pilot decides the desired flight behavior (flight intent) and issues commands (output) to the automation (e.g., to make the aircraft 'climb', the pilot either noses up using the yoke or sets a higher target altitude value using the MCP) to make it happen. This implies that the behavior of the pilot could be effectively described by discrete transitions between his/her flight intents [140, 141]. In this sense, a discrete event system whose states correspond to the intent of the pilot $I_p \in \mathcal{I}$ is used to model the behavior of the pilot as:

$$I_p(k) = g(I_p(k-1), x(k), q(k), z_c(k), z_d(k)) \tag{6.9}$$

In the above equation, the transition function g implies that the current intent of the pilot is determined by the previous intent $I_p(k-1)$, current continuous and discrete states of the aircraft $x(k)$, $q(k)$, and the pilot's inputs $z_c(k)$ and $z_d(k)$. For example, assume that the intent of the pilot at time $k-1$ is $I_p(k-1) =$ (Constant Altitude, Constant Heading, Constant Speed). When the pilot sets the higher target altitude value on the MCP and pushes the V/S mode switch, the intent of the pilot is changed to $I_p(k) =$ (Climb, Constant Heading, Constant Speed).

6.3 Real-Time Mode Confusion Detection

Our proposed algorithm detects the mode confusion by identifying a mismatch between the inferred intents of the pilot and the automation. The first step is to infer the pilot's and the automation's intents which are not directly available from the measurements. The goal of intent inference is to compute $\hat{I}_p(k)$ and $\hat{I}_a(k)$ (which are estimates for the actual intents, $I_p(k)$ and $I_a(k)$) using the noise-corrupted sensor measurements $y(k)$ and the pilot's control inputs, $z_c(k)$ and $z_d(k)$. In this paper, it is assumed that the avionics information, such as the flight modes, is not directly available to the algorithm, and thus the continuous and the discrete states, $x(k)$ and $q(k)$, of the automation should be first estimated for intent inference (note that the proposed framework can also handle the case where all the avionics information is accessible, which is a special case of the problem considered). This leads to the hybrid state estimation problem where the probability density functions of both the continuous and the discrete states are computed using the sequence of noisy measurements y. In this paper, we propose to use the state-dependent transition hybrid estimation (SDTHE) algorithm [25, 26] to solve the hybrid estimation problem. Unlike the IMM algorithm [72] which assumes constant mode transition probabilities (i.e., it assumes the flight mode transitions happen with fixed probabilities regardless of the values of the continuous states), the SDTHE can deal with continuous-state-dependent mode transitions (i.e., autonomous flight mode transitions depending on the conditions that the continuous states need to satisfy).

6.3.1 Hybrid State Estimation using the SDTHE Algorithm

Let $\mathbf{Y}(k) \equiv \{y(1), y(2), \ldots, y(k)\}$ denote the set of measurements up to time k. The hybrid estimation problem is then defined as to compute both the continuous state pdf $p(x(k)|\mathbf{Y}(k))$ and the mode probability $p(q(k)|\mathbf{Y}(k))$. From the total probability theorem, $p(x(k)|\mathbf{Y}(k))$ is computed by

$$p(x(k)|\mathbf{Y}(k)) = \sum_{i=1}^{n_q} p(x(k)|q(k) = i, \mathbf{Y}(k))p(q(k) = i|\mathbf{Y}(k)) \tag{6.10}$$

Define the discrete state probability $m^i(k)$ as $m^i(k) \equiv p(q(k) = i|\mathbf{Y}(k))$, for $i = 1, 2, \ldots, n_q$. The estimates of the hybrid states are then obtained as:

$$\begin{aligned} \hat{x}(k) :=& E[x(k)|\mathbf{Y}(k)] \\ \hat{q}(k) :=& \underset{i}{\operatorname{argmax}}\ m^i(k) \end{aligned} \tag{6.11}$$

where $E[\cdot|\cdot]$ is the (conditional) expectation of a random variable.

In the SDTHE algorithm, a bank of n_q Kalman filters [117], each of which is matched to the continuous dynamics of the individual flight mode, are used. Assume that, from the last iteration at time $k-1$, the mode probabilities $m^i(k-1)$, $i = 1, 2, \ldots, n_q$ are computed and the mode conditioned continuous pdfs are obtained as:

$$p(x(k-1)|q(k-1) = i, \mathbf{Y}(k-1)) = \mathcal{N}(x(k-1); \hat{x}^i(k-1), P^i(k-1)) \tag{6.12}$$

for $i = 1, 2, \ldots, n_q$, where $\mathcal{N}$ denotes the Gaussian distribution with the mean $\hat{x}^i(k-1)$ and the covariance $P^i(k-1)$ which are computed from the i-th Kalman filter at time $k-1$. The goal of the estimation is to compute $p(x(k)|q(k) = i, \mathbf{Y}(k))$ and $m^i(k)$ for all modes $i \in \mathcal{Q}$ using the new measurement $y(k)$ generated at time k.

Step 1: Mixing

The mixing probability $m^{i|j}(k)$ is computed as:

$$\begin{aligned} m^{i|j}(k) &= p(q(k-1)=i|q(k)=j,\mathbf{Y}(k-1)) \\ &= \frac{p(q(k)=j|q(k-1)=i,\mathbf{Y}(k-1))p(q(k-1)=i|\mathbf{Y}(k-1))}{p(q(k)=j|\mathbf{Y}(k-1))} \\ &= \frac{\lambda_{ij}(k-1)m^i(k-1)}{\Sigma_{l=1}^{n_q}\lambda_{lj}m^l(k-1)} \end{aligned} \tag{6.13}$$

where $\lambda_{ij}(k-1) := p(q(k)=j|q(k-1)=i,\mathbf{Y}(k-1))$ is denoted as the mode transition probability and computed as [142]:

$$\begin{aligned} \lambda_{ij}(k-1) &= \int_{\mathbb{R}^n} p(q(k)=j|q(k-1)=i,x(k-1)=x,\mathbf{Y}(k-1)) \\ &\quad \times p(x(k-1)=x|q(k-1)=i,\mathbf{Y}(k-1))dx \\ &= \int_{\mathbb{R}^n} \Phi_l(L_{\theta,ij}z_c + L_{x,ij}x, L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T)\mathcal{N}(x;\hat{x}^i(k-1),P^i(k-1))dx \\ &= \Phi_l(L_{\theta,ij}z_c + L_{x,ij}\hat{x}^i(k-1), L_{\theta,ij}\Sigma_\theta L_{\theta,ij}^T + L_{x,ij}P^i(k-1)L_{x,ij}^T) \end{aligned} \tag{6.14}$$

In (6.14), $\Phi_l(\mu,\Sigma)$ is the l-dimensional Gaussian cumulative density function for $y \sim \mathcal{N}(\mu,\Sigma)$ defined as:

$$\Phi_l(\mu,\Sigma) := Pr(y \leq 0) \tag{6.15}$$

Based on the mixing probability, the initial conditions for the Kalman filter matched to mode j are obtained as:

$$\begin{aligned} \hat{x}^{0j}(k-1) &= \Sigma_{i=1}^{n_q} m^{i|j}(k)\hat{x}^i(k-1) \\ P^{0j}(k-1) &= \Sigma_{i=1}^{n_q} m^{i|j}(k)\{P^i(k-1) \\ &\qquad + [\hat{x}^i(k-1)-\hat{x}^{0j}(k-1)][\hat{x}^i(k-1)-\hat{x}^{0j}(k-1)]^T\} \end{aligned} \tag{6.16}$$

Step 2: Mode-conditioned estimation

For given mode j and the initial conditions computed in (6.16), Kalman filter j computes its own posterior $p(x(k)|q(k)=j,\mathbf{Y}(k))$ as:

$$p(x(k)|q(k)=j,\mathbf{Y}(k)) = \mathcal{N}(x(k);\hat{x}^j(k),P^j(k)) \tag{6.17}$$

where

$$\begin{aligned}
\hat{x}^j(k) &= A_j \hat{x}^{0j}(k-1) + K_j(k)\nu_j(k) \\
P^j(k) &= [I - K_j(k)C_j]P^j(k|k-1) \\
P^j(k|k-1) &= A_j P^{0j}(k-1)A_j^T + Q_j \\
K_j(k) &= P^j(k|k-1)C_j^T S_j^{-1}(k) \\
S_j(k) &= C_j P^j(k|k-1)C_j^T + R_j \\
\nu_j(k) &= y(k) - C_j A_j \hat{x}^{0j}(k-1)
\end{aligned} \tag{6.18}$$

Step 3: Mode probability update

The mode probability is updated using Bayes' rule as:

$$\begin{aligned}
m^j(k) &= p(q(k) = j|\mathbf{Y}(k)) \\
&= \frac{1}{c} p(y(k)|q(k) = j, \mathbf{Y}(k-1))p(q(k) = j|\mathbf{Y}(k-1))
\end{aligned} \tag{6.19}$$

where c is a normalizing constant; $p(y(k)|q(k) = j, \mathbf{Y}(k-1))$ is the mode-conditioned likelihood function given by

$$p(y(k)|q(k) = j, \mathbf{Y}(k-1)) = \mathcal{N}(\nu_j(k); 0, S_j(k)) \tag{6.20}$$

and $p(q(k) = j|\mathbf{Y}(k-1))$ is the prior mode probability computed as:

$$p(q(k) = j|\mathbf{Y}(k-1)) = \Sigma_{i=1}^{n_q} \lambda_{ij}(k-1)m^i(k-1) \tag{6.21}$$

Step 4: Output

The continuous state estimate $\hat{x}(k)$ and its covariance $P(k)$ are obtained as a weighted sum of the mode-conditioned state estimates and the covariances as:

$$\begin{aligned}
\hat{x}(k) &= \Sigma_{j=1}^{n_q} \hat{x}^j(k)m^j(k) \\
P(k) &= \Sigma_{j=1}^{n_q} \{P^j(k) + [\hat{x}^j(k) - \hat{x}(k)][\hat{x}^j(k) - \hat{x}(k)]^T\}m^j(k)
\end{aligned} \tag{6.22}$$

The discrete state estimate $\hat{q}(k)$ can be computed with (6.11).

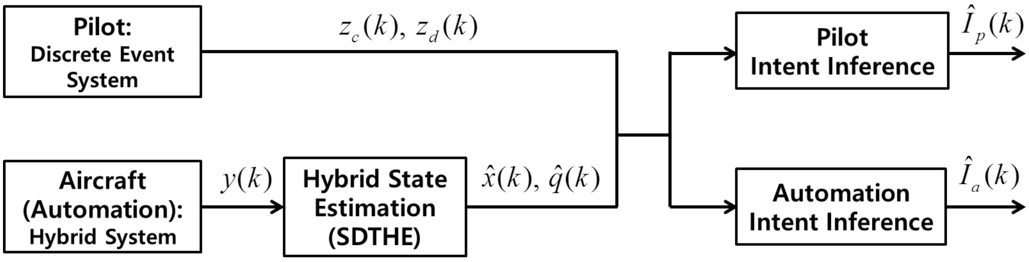

Pilot:
Discrete Event System
$z_c(k)$, $z_d(k)$
Pilot Intent Inference
$\hat{I}_p(k)$
Aircraft (Automation): Hybrid System
$y(k)$
Hybrid State Estimation (SDTHE)
$\hat{x}(k)$, $\hat{q}(k)$
Automation Intent Inference
$\hat{I}_a(k)$

Figure 6.3. Intent inference-based mode confusion detection algorithm

6.3.2 Intent Inference and Mode Confusion Detection

Once the continuous and the discrete states estimates, $\hat{x}(k)$ and $\hat{q}(k)$ are obtained using (6.11) and (6.22), the intents of the pilot and the automation can be inferred as $\hat{I}_p(k)$ and $\hat{I}_a(k)$, respectively, by

$$\begin{aligned} \hat{I}_a(k) &= \mu(\hat{x}(k), \hat{q}(k)) \\ \hat{I}_p(k) &= g(\hat{I}_p(k-1), \hat{x}(k), \hat{q}(k), z_c(k), z_d(k)) \end{aligned} \tag{6.23}$$

The mode confusion can then be detected by identifying mismatches between the inferred intents of the pilot and the automation as:

$$\text{The occurrence of mode confusion at time } k = \begin{cases} \text{yes}, & \text{if } \hat{I}_p(k) \neq \hat{I}_a(k) \\ \text{no}, & \text{if } \hat{I}_p(k) = \hat{I}_a(k) \end{cases} \tag{6.24}$$

Note that the comparison can be performed at each time step k, so that the mode confusion can be monitored in real-time. In Figure 6.3, the proposed mode confusion detection algorithm based on intent inference is summarized.

6.4 Demonstration of the Proposed Pilot-Automation Mode Confusion Detection Algorithm

In this section, we demonstrate the proposed algorithm with two illustrative mode confusion examples. For each example, we first describe the real scenario of the mode confusion, and then, test the proposed algorithm to demonstrate its performance.

6.4.1 Mode Confusion Example 1: "kill the capture"

Mode Confusion Incident Scenario

One of the well-known mode confusion examples is "kill the capture" [124]. There are two main vertical flight modes related to this incident: the Altitude Hold mode where a specified altitude value is maintained, and the Vertical Speed (V/S) mode. The logical behavior that describes transitions among the vertical flight modes of an autopilot is illustrated in Figure 6.4. In Figure 6.4, the MCP target altitude refers to the target altitude value chosen by the pilot through the MCP. Note that there are autonomous transitions between flight modes (e.g., the V/S mode to the Altitude Capture mode and the Altitude Capture mode to the Altitude Hold mode). Also, note that there are two distinct V/S modes, one of which moves toward the MCP target altitude (will be denoted by V/S target mode) and the other freely climbs or descends without a target altitude (will be denoted by V/S free mode). The most interesting feature is the presence of the capture start altitude which is determined according to the MCP target altitude. When the current altitude of the aircraft (while its flight mode is V/S target mode) reaches the capture start altitude, the Altitude Capture mode is automatically triggered and the aircraft starts to capture the target altitude. The confusion occurs when the pilot changes the MCP target altitude value while the autopilot is in the Altitude Capture mode. If the MCP target altitude is reset to a value ahead of the capture start altitude, the autopilot will transit to the V/S target mode with the new MCP target value. However, if the MCP target value is reset to a value less than the capture start altitude, the newly set altitude will be ignored, and the autopilot will autonomously enter the V/S free mode. The transition to the V/S free mode makes the aircraft fail to capture the newly set altitude intended by the pilot.

Consider the following real incident scenario (ASRS report #113722) [124]: "On climb to 27,000 feet and leaving 26,500 feet, Memphis Center gave us a clearance to descend to 24,000 feet. The aircraft had gone to "Altitude Capture" mode when

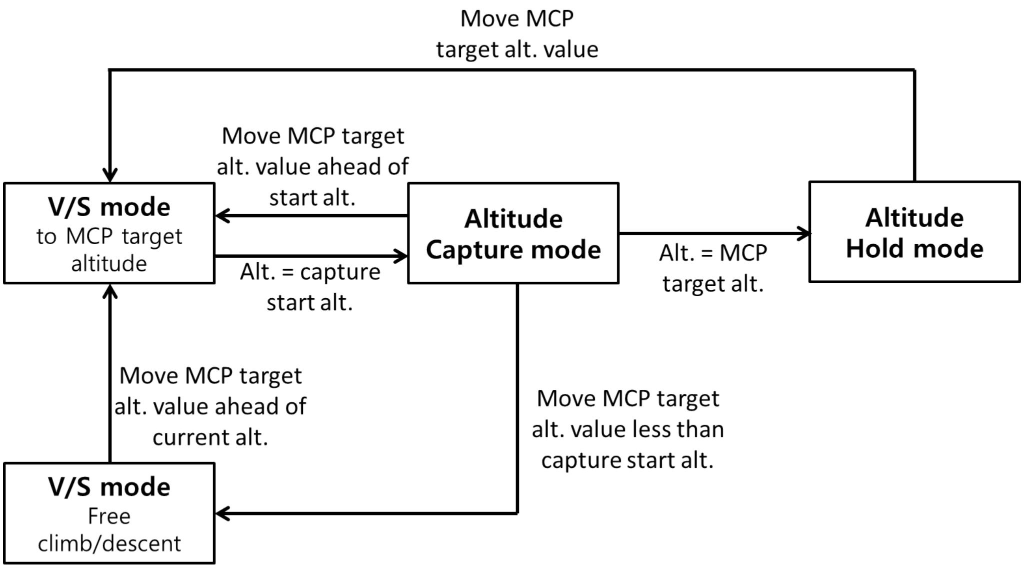


Figure 6.4. Logical behavior between vertical flight modes related to "kill the capture" example

the first officer selected 24,000 feet on the MCP altitude setting. This disarmed the altitude capture and the aircraft continued to climb at approximately 300 feet-per-minute. There was no altitude warning and this "altitude bust" went unnoticed by myself and the first officer, due to the slight rate-of-climb. At 28,500, Memphis Center asked our altitude and I replied 28,500 and started an immediate descent to 24,000 feet". In this scenario, the new target altitude (24,000 feet) was set during the Altitude Capture mode toward 27,000 feet. However, the vertical mode switched from the Capture mode to the V/S free mode, because the new target altitude was set less than the capture start altitude. This resulted in the unconstrained climb of the aircraft (see Figure 6.5).

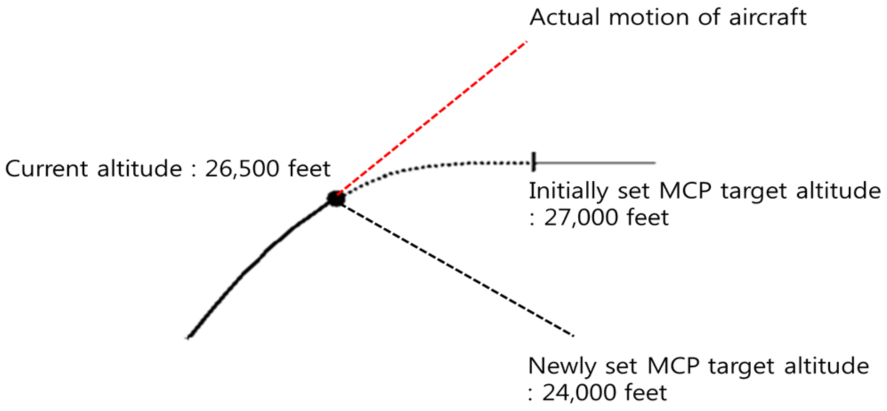

Actual motion of aircraft
Current altitude : 26,500 feet
Initially set MCP target altitude
: 27,000 feet
Newly set MCP target altitude
: 24,000 feet

Figure 6.5. Schematic of “kill the capture” example

Intent Set

In this example, the intent in the vertical dimension is of interest, because only the aircraft’s vertical motion is related to this example. In this sense, the overall intent I is represented only by i_v, and the intent set $\mathcal{I}$ is defined as:

$$I \in \mathcal{I} = \{\text{Climb, Descend, Constant Altitude}\} \tag{6.25}$$

Automation Model

The automation’s behavior depicted in Figure 6.4 is modeled as a hybrid system as follows. First, the continuous state $x(k)$ is defined as $x(k) := [h(k)\ \dot{h}(k)]^T \in \mathbf{X} \subset \mathbb{R}^2$, where h and $\dot{h}$ are the altitude and the altitude rate of the aircraft, respectively. The discrete state $q(k) \in \mathcal{Q} = \{1, 2, 3\}$ is defined as $q = 1$: Altitude Hold mode; $q = 2$: V/S mode (note that the two V/S modes denoted by the V/S target and the V/S free have the same continuous motion (i.e., either climb or descend), so that they are represented by one discrete mode); and $q = 3$: Altitude Capture mode. For each flight mode, the continuous state dynamics and the measurement equation are given by

1) $q = 1$: Altitude Hold mode

$$\begin{bmatrix} h(k+1) \\ \dot{h}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} h(k) \\ \dot{h}(k) \end{bmatrix} + w_1(k) \tag{6.26}$$

2) $q = 2$: V/S mode

$$\begin{bmatrix} h(k+1) \\ \dot{h}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h(k) \\ \dot{h}(k) \end{bmatrix} + w_2(k) \tag{6.27}$$

3) $q = 3$: Altitude Capture mode

$$\begin{bmatrix} h(k+1) \\ \dot{h}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ \beta^2 T_s & 1 \end{bmatrix} \begin{bmatrix} h(k) \\ \dot{h}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -\beta^2 T_s H_f \end{bmatrix} + w_3(k) \tag{6.28}$$

where β is the capture rate [143], H_f is the target altitude for the capture, T_s is the sampling rate, and w_1, w_2, and w_3 are zero mean white Gaussian noises for each dynamics that represent uncertainties in the aircraft's motion. For all the flight modes $q = 1, 2, 3$, the measurement vector $y(k) = [y_1(k)\ y_2(k)]^T \in \mathbb{R}^2$ is given by

$$y(k) = I_{2\times2} x(k) + u(k) \tag{6.29}$$

where $I_{2\times2}$ denotes the 2×2 identity matrix and $u(k)$ is the zero-mean white Gaussian noise. The continuous control input by the pilot $z_c \in \mathbf{Z_c} \subset \mathbb{R}$ is $z_c =$ 'the MCP target altitude set by the pilot, H_f', and the discrete control input $z_d \in \mathbf{Z_d} = \{1, 2\}$ is defined as $z_d = 1$: 'engage V/S mode switch', and $z_d = 2$: 'disengage V/S mode switch'. The logical behavior between the flight modes is then described by the mode transition function γ whose guard conditions $G(i,j)$, $\forall i,j \in \{1,2,3\}$ are defined as:

$$
\begin{aligned}
G(1,1) &= \mathbf{X} \times \mathbf{Z_c} \times \mathbf{Z_d} \backslash \left(G(1,2) \cup G(1,3)\right) \\
G(1,2) &= \{\left[[h\ \dot{h}]\ H_f\ z_d\right]^T | h \neq H_f,\ z_d = 1\} \\
G(1,3) &= \emptyset \\
\\
G(2,1) &= \emptyset \\
G(2,2) &= \mathbf{X} \times \mathbf{Z_c} \times \mathbf{Z_d} \backslash \left(G(2,1) \cup G(2,3)\right) \\
G(2,3) &= \{\left[[h\ \dot{h}]\ H_f\ z_d\right]^T | sgn(\dot{h})(h - H_c) \geq 0\} \\
\\
G(3,1) &= \{\left[[h\ \dot{h}]\ H_f\ z_d\right]^T | -h + H_f = 0\} \\
G(3,2) &= \{\left[[h\ \dot{h}]\ H_f\ z_d\right]^T \mid |h - H_f| \geq S\} \\
G(3,3) &= \mathbf{X} \times \mathbf{Z_c} \times \mathbf{Z_d} \backslash \left(G(3,1) \cup G(3,2)\right)
\end{aligned}
\tag{6.30}
$$

where H_c is the capture start altitude computed by $H_c = H_f - sgn(\dot{h})S$, given H_f and a design parameter S; $\emptyset$ denotes the empty set; and sgn is the sign function. For example, the guard condition $G(1,2)$ describes that only when the pilot engages the V/S mode switch ($z_d = 1$) and inputs a new target altitude H_f which is different from the current altitude h, the flight mode transitions from the Altitude Hold mode ($q = 1$) to the V/S mode ($q = 2$).

The continuous and the discrete behaviors of the automation are then abstracted to the flight intent $I_a \in \mathcal{I}$ by the following intent mapping $\mu(x,q) : \mathbf{X} \times \mathcal{Q} \rightarrow \mathcal{I}$:

$$
\mu(x,q) = \begin{cases} \text{Climb}, & \text{if } q = 2 \text{ or } 3, \text{ and } \dot{h} > 0 \\ \text{Descend}, & \text{if } q = 2 \text{ or } 3, \text{ and } \dot{h} < 0 \\ \text{Constant Altitude}, & \text{if } q = 1 \end{cases}
\tag{6.31}
$$

Pilot Model

The behavior of the pilot is represented by a discrete event system whose state is the intent of the pilot $I_p \in \mathcal{I}$. The transitions between the pilot's intents are determined by the transition function g as:

1) when $I_p(k) = \text{Climb}$

$$I_p(k+1) = \begin{cases} \text{Climb}, & \text{otherwise} \\ \text{Descend}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) < h(k+1), \\ & \text{and } z_d(k+1) = 1 \\ \text{Constant Altitude}, & \text{if } z_c(k+1) = h(k+1) \end{cases} \tag{6.32}$$

2) when $I_p(k) = \text{Descend}$

$$I_p(k+1) = \begin{cases} \text{Climb}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) > h(k+1), \\ & \text{and } z_d(k+1) = 1 \\ \text{Descend}, & \text{otherwise} \\ \text{Constant Altitude}, & \text{if } z_c(k+1) = h(k+1) \end{cases} \tag{6.33}$$

3) when $I_p(k) = \text{Constant Altitude}$

$$I_p(k+1) = \begin{cases} \text{Climb}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) > h(k+1), \\ & \text{and } z_d(k+1) = 1 \\ \text{Descend}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) < h(k+1), \\ & \text{and } z_d(k+1) = 1 \\ \text{Constant Altitude}, & \text{otherwise} \end{cases} \tag{6.34}$$

This pilot model characterizes the pilot's behavior (i.e., the generation of the pilot's inputs, z_c and z_d) in terms of their flight intents (in this example, the vertical flight intents).

Using the pilot and the automation models, the actual trajectory of the "kill the capture" example can be simulated. Figure 6.6 shows the histories of the MCP target

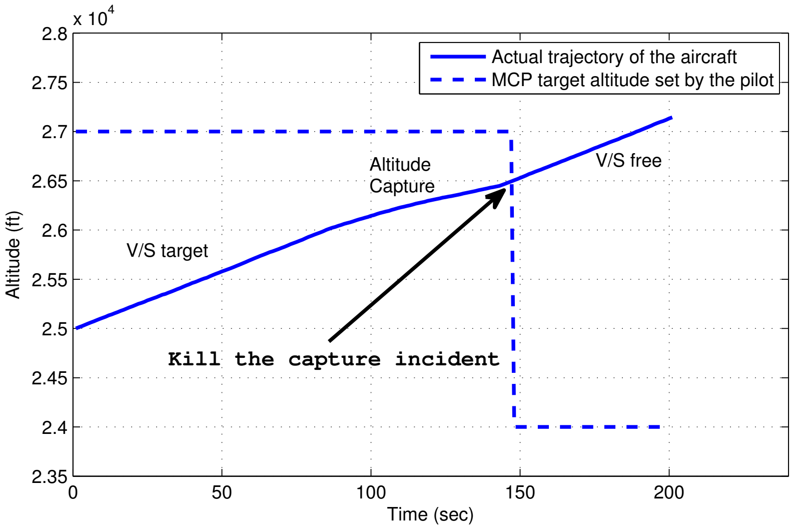


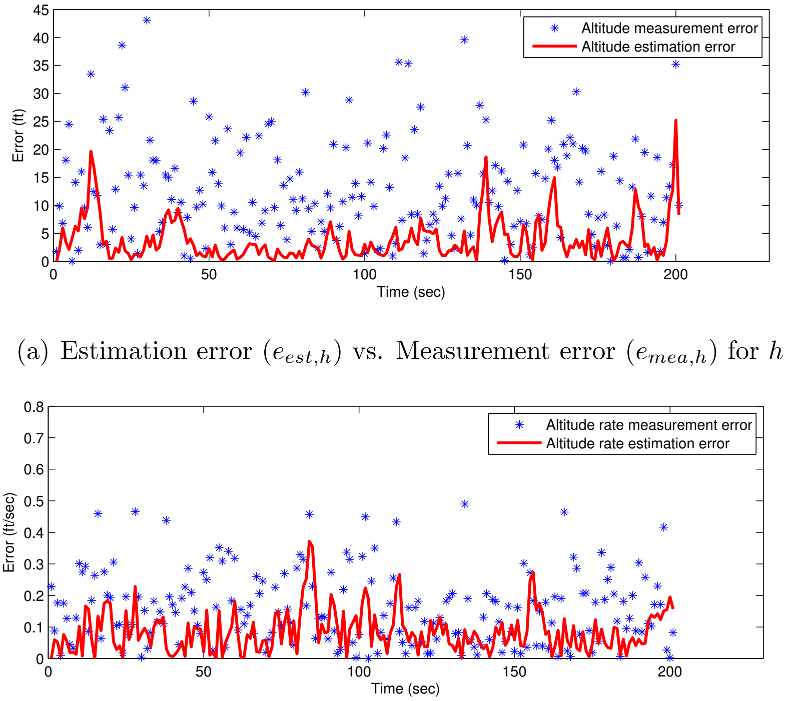
Figure 6.6. Simulated "kill the capture" incident scenario

altitude set by the pilot and the aircraft's actual altitude. In this simulation, the mode confusion incident occurs at 148 sec as the pilot newly sets a lower MCP target altitude (24,000 ft) while the automation is in the Altitude Capture mode.

Real-Time Mode Confusion Detection based on Intent Inference

The proposed mode confusion detection algorithm first infers the pilot's and the automation's intents using the measurements such as $y(k)$, $z_c(k)$, and $z_d(k)$. With noisy measurements, the hybrid states of the automation (aircraft) are estimated using the SDTHE. Figure 6.7 shows the estimation accuracy of the continuous states for a single run (the Monte Carlo simulation results are summarized in Table 6.2), where the altitude estimation error $e_{est,h}$ and the altitude rate estimation error $e_{est,\dot{h}}$ are defined as:

$$\begin{aligned} e_{est,h}(k) &\equiv |h(k) - \hat{h}(k)| \\ e_{est,\dot{h}}(k) &\equiv |\dot{h}(k) - \hat{\dot{h}}(k)| \end{aligned} \tag{6.35}$$



(a) Estimation error ($e_{est,h}$) vs. Measurement error ($e_{mea,h}$) for h

(b) Estimation error ($e_{est,\dot{h}}$) vs. Measurement error ($e_{mea,\dot{h}}$) for $\dot{h}$

Figure 6.7. Estimation accuracy of the continuous state estimates (h and $\dot{h}$)

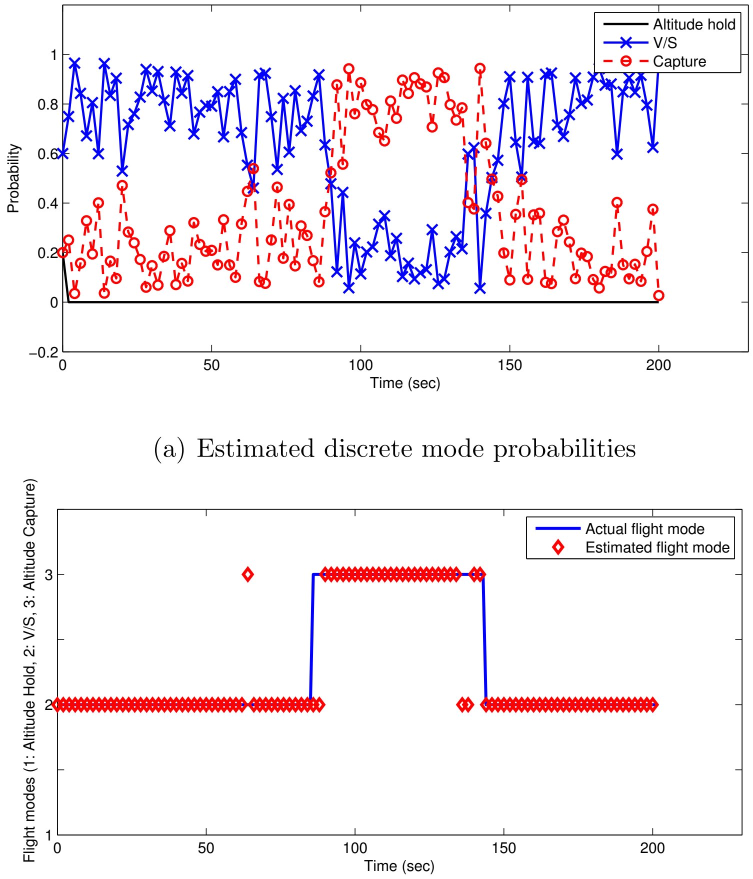
and the altitude measurement error $e_{mea,h}$ and the altitude rate measurement error $e_{mea,\dot{h}}$ are defined as:

$$\begin{aligned} e_{mea,h}(k) &\equiv |h(k) - y_1(k)| \\ e_{mea,\dot{h}}(k) &\equiv |\dot{h}(k) - y_2(k)| \end{aligned} \tag{6.36}$$

As shown in Figure 6.7 and Table 6.2, the estimation errors are smaller than the measurement errors for both the altitude and the altitude rate, which indicates the effectiveness of the SDTHE algorithm in filtering out the noise in the sensor measurement data. In Figure 6.8, the estimated flight mode probability $m^j(k)$, $j = 1, 2, 3$, and the corresponding estimated flight mode $\hat{q}(k)$ (obtained using (6.11)) are presented for a single run (the Monte Carlo simulation results are summarized in Table 6.2). Figure 6.8 shows that we can accurately estimate the actual flight mode with little delay which is caused by the uncertainties both in the dynamics and the

Table 6.2. Comparison of estimation error and measurement error for Example 1 (100 Monte Carlo runs)

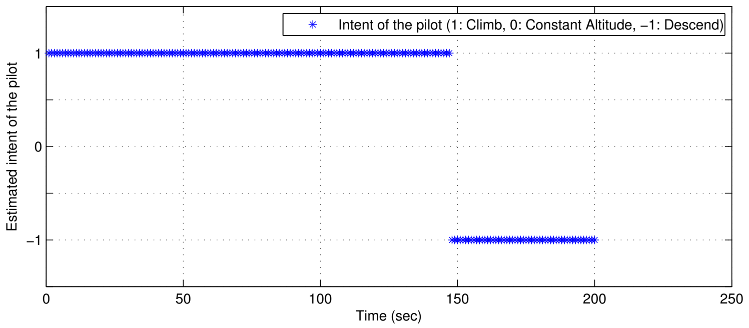
		Estimation error	Measurement error
RMS altitude error (ft)	Peak	10.26	19.31
	Average	5.68	14.94
RMS altitude rate error (ft/sec)	Peak	0.1560	0.2291
	Average	0.1050	0.1977
Average number of discrete state estimation errors $q(k) \neq \hat{q}(k)$		14.33 (7.16% of a simulation timespan (200 sec))	.



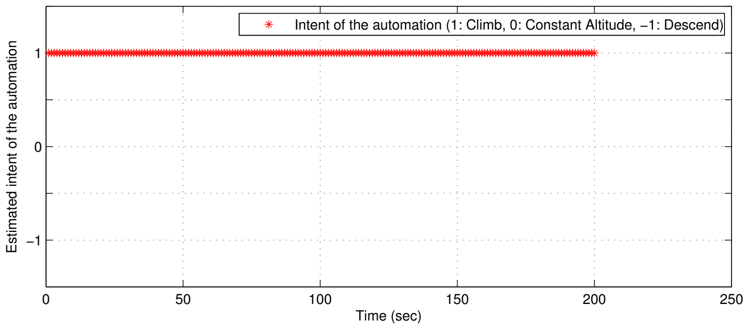
(a) Estimated discrete mode probabilities

(b) Actual flight mode (q) vs. Estimated flight mode ($\hat{q}$)

Figure 6.8. Estimation of discrete flight mode using hybrid estimation for Example 1



(a) Inferred intent $\hat{I}_p$ of the pilot



(b) Inferred intent $\hat{I}_a$ of the automation)

Figure 6.9. Inferred intents of the pilot and the automation for Example 1

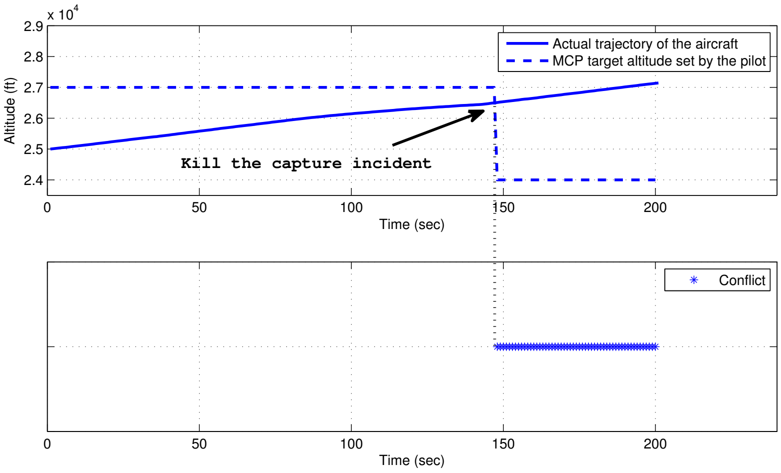


Figure 6.10. Intent conflict detection for Example 1

sensor measurements. Also, it is noted that the mode probability for the Altitude Hold mode decreases to near zero, which is consistent with the fact that the Altitude Hold mode was never engaged in this scenario.

Once $\hat{h}(k)$, $\dot{\hat{h}}(k)$, and $\hat{q}(k)$ are computed, the next step is to infer the intents of the pilot (using (6.32), (6.33), and (6.34)) and the automation (using (6.31)) using those estimates. Figure 6.9 represents the inferred intents of the pilot and the automation where the values $1, 0, -1$ in the vertical axis denotes Climb, Constant Altitude, and Descend intents, respectively. Finally, the mode confusion is detected by identifying the conflicts between the inferred intents of the pilot and the automation as shown in Figure 6.10. In the figure, when the pilot changes the MCP target altitude (i.e., intent change), the automation keeps climbing (i.e., no intent change), and therefore, their intents do not match with each other. The “kill the capture” mode confusion, which is reflected as a mismatch in the inferred intents, is accurately detected by the proposed algorithm. This timely identification of the mode confusion can prevent the aircraft’s undesirable behavior (as presented in the real scenario), and therefore, reduce the risk of accidents or incidents, enhancing the aviation safety.

6.4.2 Mode Confusion Example 2: “airspeed reset problem in the vertical navigation (VNAV) mode”

Mode Confusion Incident Scenario

In this example, the mode confusion is related to the combined behavior of both the vertical and the auto-throttle flight modes. When the pilot assigns a specific target airspeed during a certain vertical mode, the speed (SPD) auto-throttle mode is engaged to maintain the target airspeed. When the VNAV mode (in this vertical mode, the aircraft automatically follows the predefined vertical trajectory) is engaged, the airspeed target value is reset to the value computed by the Flight Management System (FMS) to achieve the most economical maneuver. If the previous target airspeed is assigned by an Air Traffic Controller (ATC) (and set into the MCP by the pilot) and required to be constant unless a new speed clearance is given, the autonomous change of the target speed value at the beginning of the VNAV mode can cause a conflict with respect to the ATC speed restriction. Consider the following real incident scenario [123]: “The aircraft was climbing to 11,000 feet per ATC instructions. During the climb (at about 10,500 feet), ATC instructed the crew to reduce speed to 240 knots. The first officer entered the speed, specified by ATC, via the MCP as a new reference parameter. As the aircraft neared 11,000 feet, it started the level-off maneuver through the “Altitude Capture” mode. Once at 11,000 feet, the “Altitude Capture” mode was disengaged and the “Altitude Hold” mode was engaged automatically. During and after the maneuver, the speed was kept at 240 knots. Shortly after, ATC instructed the crew to climb to 14,000 feet. The first officer reached up to the MCP and engaged the “Vertical Navigation” mode in order to initiate the climb. However, instead of climbing at a speed of 240 knots (as was still required by ATC), the aircraft speed defaulted to 300 knots (computed by the FMS). The crew violated the ATC speed restriction because they assumed that the “Vertical Navigation” mode would “remember” the speed reference parameter entered previously into the MCP. However, the “Vertical Navigation” mode defaulted to the economy speed (about 300 knots in this case)”. In this scenario, the target speed was set to 240 knots as required by ATC during the early stage of the flight. When the pilot started a new vertical maneuver using the VNAV mode, the target speed was automatically reset to 300 knots which was computed by the FMS. This resulted in the violation of ATC speed restriction.

Intent Set

Because the aircraft's vertical and speed motions are of interest in this example, the overall intent I is represented as a 2-tuple of i_v and i_s. The intent set $\mathcal{I}$ is then defined as:

$$\begin{aligned} I \in \mathcal{I} = \{&(\text{Climb, Accelerate}), (\text{Climb, Decelerate}), (\text{Climb, Constant Speed}), \\ &\ldots, (\text{Descend, Decelerate}), (\text{Descend, Constant Speed})\} \end{aligned} \tag{6.37}$$

Automation Model

Because the automation's vertical behavior is presented in detail in the previous example, only the behavior in the speed dimension is described in this example. The continuous state $x(k)$ is defined as $x(k) := [v(k)\ \dot{v}(k)]^T \in \mathbf{X} \subset \mathbb{R}^2$, where v and $\dot{v}$ are the airspeed and the airspeed rate of the aircraft, respectively. The discrete state $q(k) \in \mathcal{Q} = \{1, 2\}$ is defined such that $q = 1$: Speed Change mode, and $q = 2$: Speed Hold mode. For each mode, the continuous state dynamics are given by

1) $q = 1$: Speed Change mode

$$\begin{bmatrix} v(k+1) \\ \dot{v}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ \dot{v}(k) \end{bmatrix} + w_1(k) \tag{6.38}$$

2) $q = 2$: Speed Hold mode

$$\begin{bmatrix} v(k+1) \\ \dot{v}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ \dot{v}(k) \end{bmatrix} + w_2(k) \tag{6.39}$$

where T_s is the sampling rate, and w_1 and w_2 are zero mean white Gaussian noises. For all the modes $q = 1, 2$, the measurement vector $y(k) = [y_1(k)\ y_2(k)]^T \in \mathbb{R}^2$ is given by (6.29). The continuous control input $z_c \in \mathbf{Z_c}$ by the pilot is $z_c =$ 'the MCP target airspeed, V_t', and the discrete control input $z_d \in \mathbf{Z_d}$ is defined as $z_d = 1$: 'engage VNAV mode switch', and $z_d = 2$: 'disengage VNAV mode switch'.

The mode transition is then described by defining the guard conditions $G(i,j), \forall i,j \in \{1,2\}$ as:

$$\begin{aligned} G(1,1) &= \mathbf{X} \times \mathbf{Z_c} \times \mathbf{Z_d} \backslash G(1,2) \\ G(1,2) &= \{[v\ \dot{v}]\ V_t\ z_d]^T | v - V_t = 0\} \end{aligned} \tag{6.40}$$

$$\begin{aligned} G(2,1) &= \{[v\ \dot{v}]\ V_t\ z_d]^T | v \neq V_t,\ z_d = 1\} \\ G(2,2) &= \mathbf{X} \times \mathbf{Z_c} \times \mathbf{Z_d} \backslash G(2,1) \end{aligned}$$

The automation's flight intent $I_a \in \mathcal{I}$ is then determined by the following intent mapping $\mu(x,q) : \mathbf{X} \times \mathcal{Q} \to \mathcal{I}$ (because the intent mapping for the vertical motion is presented in the previous example, only the intent mapping for the speed intent, i_s, is described in this example):

$$\mu(x,q) = \begin{cases} \text{Accelerate}, & \text{if } q = 1 \text{ and } \dot{v} > 0 \\ \text{Decelerate}, & \text{if } q = 1 \text{ and } \dot{v} < 0 \\ \text{Constant Speed}, & \text{if } q = 2 \end{cases} \tag{6.41}$$

Pilot Model

The transitions of the pilot's intents are determined by defining the transition function g as:

1) when $I_p(k) =$Accelerate

$$I_p(k+1) = \begin{cases} \text{Accelerate}, & \text{otherwise} \\ \text{Decelerate}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) < v(k+1) \\ \text{Constant Speed}, & \text{if } z_c(k+1) = v(k+1) \end{cases} \tag{6.42}$$

2) when $I_p(k) =$Decelerate

$$I_p(k+1) = \begin{cases} \text{Accelerate}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) > v(k+1) \\ \text{Decelerate}, & \text{otherwise} \\ \text{Constant Speed}, & \text{if } z_c(k+1) = v(k+1) \end{cases} \tag{6.43}$$

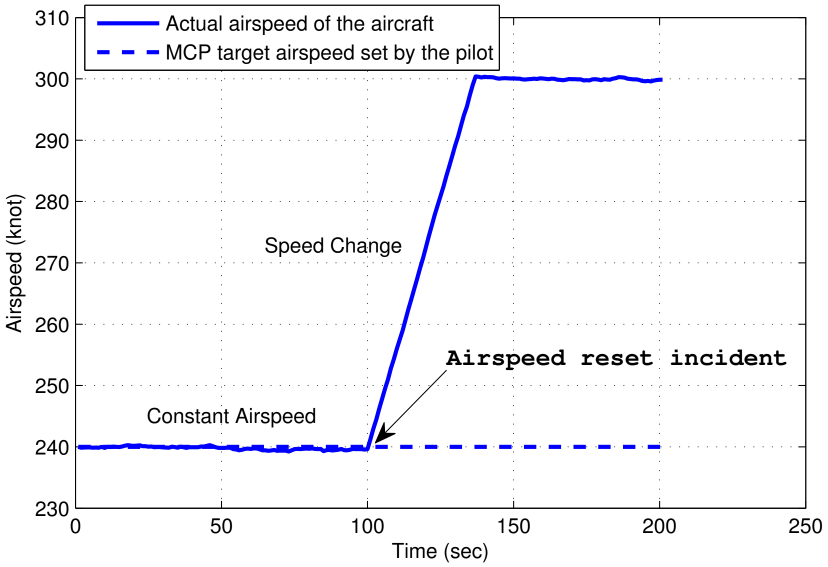


Figure 6.11. Simulated airspeed reset incident scenario

3) when $I_p(k)$ =Constant Speed

$$I_p(k+1) = \begin{cases} \text{Accelerate}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) > v(k+1) \\ \text{Decelerate}, & \text{if } z_c(k+1) \neq z_c(k),\ z_c(k+1) < v(k+1) \\ \text{Constant Speed}, & \text{otherwise} \end{cases} \tag{6.44}$$

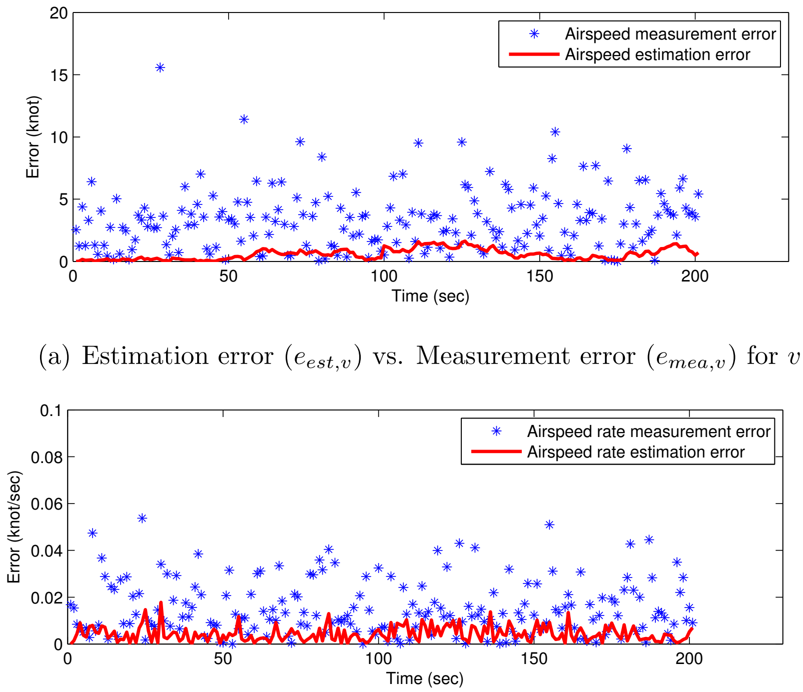
Using the pilot and the automation models, the real incident scenario is simulated as shown in Figure 6.11 where the MCP target airspeed set by the pilot and the aircraft's actual airspeed are presented. In this simulation, the mode confusion incident occurs at 100 sec when the pilot engages the VNAV mode while not changing the MCP target airspeed value.

Real-Time Mode Confusion Detection based on Intent Inference

First, using SDTHE, the hybrid states are estimated with the noisy measurements $y(k)$. The estimation accuracy for a single run is shown in Figure 6.12 (the Monte

Table 6.3. Comparison of estimation error and measurement error for Example 2 (100 Monte Carlo runs)

		Estimation error	Measurement error
RMS altitude error (knot)	Peak	1.44	2.36
	Average	1.18	2.00
RMS altitude rate error (knot/sec)	Peak	0.0091	0.0118
	Average	0.0051	0.0099
Average number of discrete state estimation errors $q(k) \neq \hat{q}(k)$		14.56 (7.28% of a simulation timespan (200 sec))	·
False conflict duration (sec)	Peak	5	·
	Average	1.31	·



(a) Estimation error ($e_{est,v}$) vs. Measurement error ($e_{mea,v}$) for v

(b) Estimation error ($e_{est,\dot{v}}$) vs. Measurement error ($e_{mea,\dot{v}}$) for $\dot{v}$

Figure 6.12. Estimation accuracy of the continuous state estimates (v and $\dot{v}$)

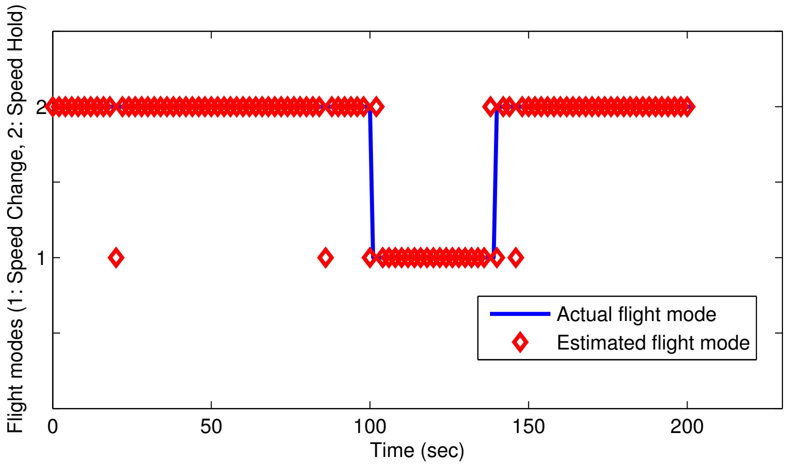


Figure 6.13. Estimation of discrete flight mode using hybrid estimation for Example 2

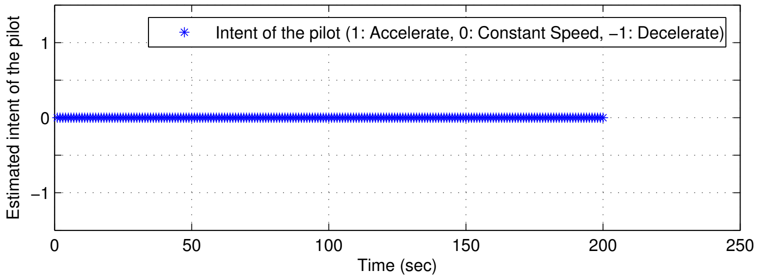
Carlo simulation results are summarized in Table 6.3), where the airspeed estimation error $e_{est,v}$ and the airspeed rate estimation error $e_{est,\dot{v}}$ are defined as:

$$\begin{aligned} e_{est,v} &\equiv |v(k) - \hat{v}(k)| \\ e_{est,\dot{v}} &\equiv |\hat{v}(k) - \hat{\dot{v}}(k)| \end{aligned} \tag{6.45}$$

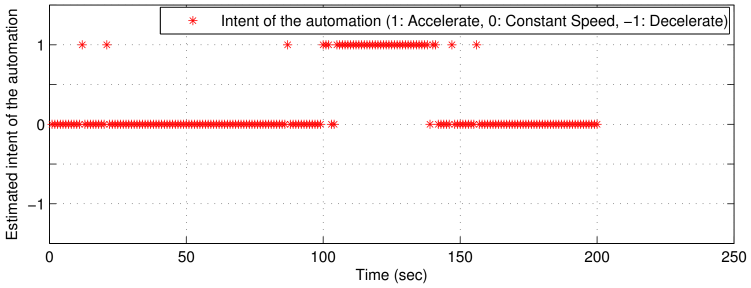
and the airspeed measurement error $e_{mea,v}$ and the airspeed rate measurement error $e_{mea,\dot{v}}$ are defined as:

$$\begin{aligned} e_{mea,v} &\equiv |v(k) - y_1(k)| \\ e_{mea,\hat{v}} &\equiv |\dot{v}(k) - y_2(k)| \end{aligned} \tag{6.46}$$

The estimated mode and the actual mode are presented in Figure 6.13 (the Monte Carlo simulation results are summarized in Table 6.3). With $\hat{v}(k)$, $\hat{\dot{v}}(k)$, and $\hat{q}(k)$, the intents of the pilot and the automation are estimated as in Figure 6.14, where the values $1, 0, -1$ in the vertical axis denotes Accelerate, Constant Speed, and Decelerate intent, respectively. In Figure 6.15, the estimated intents of the pilot and the automation are compared to each other and the mismatches between the intents are detected. Unlike the previous mode confusion example, due to severe noise in the airspeed measurement data, there are errors in estimating the intent of the automation (Figure 6.12(b)) and therefore, the false mismatches are identified as shown in Figure 6.15. However, the duration of the false conflicts are less than few seconds (the distribution of the durations of the false conflicts are computed through the Monte Carlo simulation, and the results are presented in Table 6.3), and therefore, they can be filtered out by setting an appropriate threshold. As shown in Figure 6.15, the speed-related mode confusion which happens at 100 sec is accurately detected as a long conflict between the intents.



(a) Inferred intent $\hat{I}_p$ of the pilot



(b) Inferred intent $\hat{I}_a$ of the automation)

Figure 6.14. Inferred intents of the pilot and the automation for Example 2

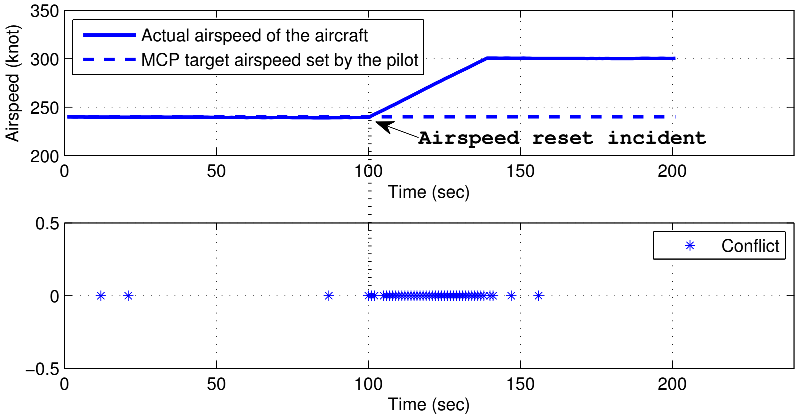


Figure 6.15. Intent conflict detection for Example 2

7. SUMMARY

In this thesis, we have discussed challenges in information inference for Cyber-Physical Systems (CPSs) and developed new inference algorithms that effectively overcome the challenges. The developed algorithms have been applied to information inference problems of various CPSs such as air traffic control systems, space surveillance systems, and pilot-automation integrated systems.

we have considered the event-based state estimation problem for CPSs where measurements are available only when some events happen. First, a general framework for the event-based state estimation problem has been developed in an attempt to derive general filtering equations that can be applied to various event-based estimation problems. Then, a numerical algorithm which is based on the Markov chain approximation method has been proposed to solve the filtering equations. Unlike the existing methods which perform the measurement update only when new measurement data arrives, the proposed approach systematically uses the information generated between the two consecutive measurement arrivals. The improvement of the estimation accuracy obtained by the proposed algorithm has been demonstrated using an illustrative state estimation example of bistable nonlinear system.

The event-based state estimation framework has been extended to CPSs that have the complex hybrid system structure. An event-based hybrid state estimation algorithm has been proposed that utilizes the idea of the interacting multiple model (IMM) approach to reduce the exponentially growing computational complexity, and the pseudo-measurement generation based approximation method for evaluating the multivariate integral required in the computation of the likelihood function. While the conventional IMM approach cannot solve the multivariate integral nor incorporate the implicit information that the measurements of sensors which do not send their data to the estimator should remain in a certain range, the proposed algorithm systematically exploits them to improve the estimation accuracy. The improvement of the estimation accuracy obtained by the proposed algorithm has been demonstrated by an aircraft tracking problem in the air traffic control application.

State estimation problem for CPSs whose states are subject to abrupt changes has been discussed with its application to space situational awareness (SSA). As an example, the tracking problem of impulsively maneuvering spacecraft has been considered. To account for the unknown magnitude and time of occurrence of impulsive maneuvers, a new state-dependent adaptive estimation algorithm has been proposed. To deal with abrupt changes in the spacecraft's motion due to the impulsive maneuvers, two dynamical models have been developed, each of which accounts for the spacecraft's motion with or without the maneuvers. In addition, to incorporate the useful information that the maneuvers occur when the state of the spacecraft satisfies certain conditions, a state-dependent transition probability has been derived and used to blend the two dynamical models. The proposed algorithm has been demonstrated with two illustrative satellite tracking problems.

Another important information inference problem in the SSA application, called the uncertainty propagation problem has been discussed for satellite formation flying systems. An analytical closed-form solution has been developed for uncertainty propagation in the satellite relative motion near general elliptic orbits. To deal with difficulties in finding the analytical solution, we have proposed to use an adjoint system associated to the Tschauner-Hempel equations and the binomial series expansion. Since the analytical solution does not require any numerical integration, it allows satellite onboard computers having low computational capability to perform necessary computations efficiently. Similarly, the analytical solution also benefits the calculation of the collision probability that must be carried out in a timely manner. Finally, the developed analytical solution can play an important role as the foundation based on which a solution for uncertainty propagation using more complex relative dynamic models can be developed.

Information inference algorithms for pilot-automation integrated systems have been developed to identify safe-critical issues due to malicious interaction between the pilot and automation. The behaviors of the pilot-automation system have been characterized using the hybrid system and discrete event system modeling approaches. The anomalous interaction issues happen when the goals of the pilot and that of the automation are different. To represent the goals of the pilot and the automation, the concept of intents has been introduced. The anomalous interaction of the combined system is then identified by comparing the inferred intents of the automation and pilot. Through illustrative real mode confusion examples, it has been demonstrated that the proposed algorithm can accurately detect pilot-automation interaction issues in a timely manner. Therefore, the developed interaction issue detection algorithm can enhance the safety of aircraft operation.

LIST OF REFERENCES

LIST OF REFERENCES

[1] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta. Ensuring safety, security, and sustainability of mission-critical cyber-physical systems. *Proceedings of the IEEE*, 100(1):283–299, 2012.

[2] A. B. Sharma, F. Ivancic, A. Niculescu-Mizil, H. Chen, and G. Jiang. Modeling and analytics for cyber-physical systems in the age of big data. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):74–77, 2014.

[3] Concept of operations for the next generation air transportation system. Technical report, Joint Planning and Development Office, Washington, D. C., 2011.

[4] W. Liu and I. Hwang. Probabilistic trajectory prediction and conflict detection. *Journal of Guidance, Control, and Dynamics*, 34(6):1779–1789, 2011.

[5] A. Ribeiro and G. B. Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks-part I: Gaussian case. *IEEE Transactions on Signal Processing*, 54(3):1131–1143, 2006.

[6] A. Ribeiro and G. B. Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks-part II: unknown probability density function. *IEEE Transactions on Signal Processing*, 54(7):2784–2796, 2006.

[7] P. G. Otanez, J. R. Moyne, and D. M. Tilbury. Using deadbands to reduce communication in networked control systems. In *Proceedings of the 2002 American Control Conference*, pages 3015–3020, Anchorage, AK, May 2002.

[8] J. K. Yook, D. M. Tilbury, and N. R. Soparkar. Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *IEEE Transactions on Contorl Systems Technology*, 10(4):503–518, 2002.

[9] C. Peng and T. C. Yang. Event-triggered communication and H_∞ control co-design for networked control systems. *Automatica*, 49(5):1326–1332, 2013.

[10] C. Peng, Q. L. Han, and D. Yue. To transmit or not to transmit: A discrete event-triggered communication scheme for networked takagi-sugeno fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 21(1):164–170, 2013.

[11] J. Lygeros, C. Tomlin, and S. Sastry. Hybrid systems: Modeling, analysis and control. preprint, 2008.

[12] J. Hu, J. Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. In *Hybrid Systems: Computation and Control,* volume 1790 of the series Lecture Notes in Computer Science, pages 160–173. Springer Verlag, 2000.

[13] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, 2001.

[14] P. Sanyal and C. N. Shen. Bayes' decision rule for rapid detection and adaptive estimation scheme with space applications. *IEEE Transactions on Automatic Control*, 19(3):228–231, 1974.

[15] S. Lee and I. Hwang. State-dependent adaptive estimation for impulsively maneuvering spacecraft tracking. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, January 2016.

[16] A. Willsky and H. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, 21(1):108–112, 1976.

[17] P. F. Weston and J. P. Norton. Detection and estimation of abrupt changes in input or state. *International Journal of Control*, 67(5):699–712, 1997.

[18] R. Durrett. *Stochastic Calculus: A Practical Introduction.* CRC Press, Boca Raton, FL, 1996.

[19] J. Hu, M. Prandini, and S. Sastry. Probabilistic safety analysis in three dimensional aircraft flight. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 5335–5340, Maui, HI, December 2003.

[20] X. D. Koutsoukos and D. Riley. Computational methods for verification of stochastic hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(2):385–396, 2008.

[21] S. Lee, W. Liu, and I. Hwang. Event-based state estimation algorithm using Markov chain approximation. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 6998–7003, Firenze, Italia, December 2013.

[22] S. Lee and I. Hwang. Event-based state estimation for stochastic hybrid systems. *IET Control Theory and Applications*, 9(13):1973–1981, 2015.

[23] J. P. Hespanha. A model for stochastic hybrid systems with application to communication networks. *Nonlinear Analysis: Theory, Methods, and Applications*, 62(8):1353–1383, 2005.

[24] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.

[25] C. E. Seah and I. Hwang. Stochastic linear hybrid systems: Modeling, estimation, and application in air traffic control. *IEEE Transactions on Control Systems Technology*, 17(3):563–575, 2009.

[26] C. E. Seah and I. Hwang. State estimation for stochastic linear hybrid systems with continuous-state-dependent transitions: An IMM approach. *IEEE Transactions on Aerospace and Electronic Systems*, 45(1):376–392, 2009.

[27] J. Lee, S. Lee, and I. Hwang. Hybrid system modeling and estimation for estimated time of arrival prediction in terminal airspace. *Journal of Guidance, Control, and Dynamics*, 2015. Accepted.

[28] H. A. P. Blom and E. A. Bloem. Exact Bayesian and particle filtering of stochastic hybrid systems. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):55–70, 2007.

[29] D. Lubey, D. Scheeres, and R. Erwin. Maneuver detection and reconstruction of stationkeeping spacecraft at GEO using the optimal control-based estimator. In *Proceedings of the 1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles*, pages 216–221, Seville, Spain, June 2015.

[30] Q. M. Zheng and Y. J. Jhao. Probabilistic approach to trajectory conformance monitoring. *Journal of Guidance, Control, and Dynamics*, 35(6):1888–1898, 2012.

[31] I. Lymperopoulos and J. Lygeros. Improved multi-aircraft ground trajectory prediction for air traffic control. *Journal of Guidance, Control, and Dynamics*, 33(2):347–362, 2010.

[32] J. L. Yepes, I. Hwang, and M. Rotea. New algorithms for aircraft intent inference and trajectory prediction. *Journal of Guidance, Control, and Dynamics*, 30(2):370–382, 2007.

[33] V. H. Nguyen and Y. S. Suh. Improving estimation performance in networked control systems applying the send-on-delta transmission method. *Sensors*, 7(10):2128–2138, 2007.

[34] S. Lee and I. Hwang. Interacting multiple model estimation for spacecraft maneuver detection and characterization. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Kissimmee, FL, January 2015.

[35] J. Sharma. Space-based visible space surveillance performance. *Journal of Guidance, Control, and Dynamics*, 23(1):153–158, 2000.

[36] K. J. DeMars, I. I. Hussein, C. Frueh, M. K. Jah, and R. Erwin. Multiple-object space surveillance tracking using finite-set statistics. *Journal of Guidance, Control, and Dynamics*, 38(Special Issue in Honor of Richard Battin):1741–1756, 2015.

[37] R. P. Patera. Space event detection method. *Journal of Spacecraft and Rockets*, 45(3):554–559, 2008.

[38] J. Huang, W.-D. Hu, Q. Xin, and X.-Y. Du. An object correlation and maneuver detection approach for space surveillance. *Research in Astronomy and Astrophysics*, 12(10):1402–1416, 2012.

[39] S. Lee, J. Lee, and I. Hwang. Maneuvering spacecraft tracking via state-dependent adaptive estimation. *Journal of Guidance, Control, and Dynamics*, 2016. Accepted.

[40] R. Erwin, P. Albuquerque, S. K. Jayaweera, and I. Hussein. Dynamic sensor tasking for space situational awareness. In *Proceedings of the 2010 American Control Conference*, pages 1153–1158, Baltimore, MD, June 2010.

[41] K. Fujimoto and D. J. Scheeres. Analytical nonlinear propagation of uncertainty in the two-body problem. *Journal of Guidance, Control, and Dynamics*, 35(2):497–509, 2012.

[42] K. J. DeMars, R. H. Bishop, and M. K. Jah. Entropy-based approach for uncertainty propagation of nonlinear dynamical systems. *Journal of Guidance, Control, and Dynamics*, 36(4):1047–1057, 2013.

[43] R. P. Patera. Space vehicle conflict-avoidance analysis. *Journal of Guidance, Control, and Dynamics*, 30(2):492–498, 2007.

[44] K. J. DeMars, Y. Cheng, and M. K. Jah. Collision probability with Gaussian mixture orbit uncertainty. *Journal of Guidance, Control, and Dynamics*, 37(3):979–985, 2014.

[45] S. Lee, H. Lyu, and I. Hwang. Analytical uncertainty propagation for satellite relative motion along elliptic orbits. *Journal of Guidance, Control, and Dynamics*, 2016. Accepted.

[46] C. E. Billings. Human-centered aircraft automation: A concept and guidelines. Technical report, NASA TM-103885, 1991.

[47] J. Croft. Automation paralysis: Airspeed eroded, despite six eyes on the Asiana 777 flight deck. *Aviation Week and Space Technology*, 175(24):22–24, 2013.

[48] S. Lee and I. Hwang. Intent inference-based flight-deck human-automation mode-confusion detection. *Journal of Aerospace Information Systems*, 12(8):503–518, 2015.

[49] A. Pawlowski, J. L. Guzman, F. Rodriguez, M. Berenguel, J. Sanchez, and S. Dormido. The influence of event-based sampling techniques on data transmission and control performance. In *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, pages 1–8, Mallorca, September 2009.

[50] Z. Wang, E. Bulut, and B. K. Szymanski. Distributed energy efficient target tracking with binary sensor networks. *ACM Transaction On Sensor Networks*, 6(4):ID 32, 2010.

[51] S. Lee, W. Liu, and I. Hwang. Event-triggered filtering with application to target tracking in binary sensor networks. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 769–774, Maui, HI, December 2012.

[52] L. Li, M. Lemmon, and X. Wang. Event-triggered state estimation in vector linear processes. In *Proceedings of the 2010 American Control Conference*, pages 2138–2143, Baltimore, MD, June 2010.

[53] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, 2004.

[54] Y. K. Xu and X. R. Cao. Lebesgue-sampling-based optimal control problems with time aggregation. *IEEE Transactions on Automatic Control*, 56(5):1097–1109, 2011.

[55] J. Wu, Q. S. Jia, K. H. Johansson, and L. Shi. Event-based sensor data scheduling: Trade-off between communication rate and estimation quality. *IEEE Transactions on Automatic Control*, 58(4):1041–1046, 2013.

[56] J. Sijs and M. Lazar. On event based state estimation. In *Hybrid Systems: Computation and Control,* volume 5469 of the series Lecture Notes in Computer Science, pages 336–350. Springer Verlag, 2009.

[57] J. Sijs and M. Lazar. Event based state estimation with time synchronous updates. *IEEE Transactions on Automatic Control*, 57(10):2650–2655, 2012.

[58] M. G. Cea and G. C. Goodwin. Event based sampling in non-linear filtering. *Control Engineering Practice*, 20(10):963–971, 2012.

[59] A. Capponi, I. Fatkullin, and L. Shi. Stochastic filtering for diffusion processes with level crossings. *IEEE Transactions on Automatic Control*, 56(9):2201–2206, 2011.

[60] S. Marano, V. Matta, and P. Willett. Some approaches to quantization for distributed estimation with data association. *IEEE Transactions on Signal Processing*, 53(3):885–895, 2005.

[61] K. J. Astrom and B. M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 2011–2016, Las Vegas, NV, December 2002.

[62] X. D. Koutsoukos. Estimation of hybrid systems using discrete sensors. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 155–160, Maui, HI, December 2003.

[63] C. W. Gardiner. *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences.* Springer Verlag, New York, NY, 2004.

[64] K. Kastella and C. Kreucher. Multiple model nonlinear filtering for low signal ground target applications. *IEEE Transactions on Aerospace and Electronic Systems*, 41(2):549–564, 2005.

[65] K. Kastella and A. Zatezalo. Nonlinear filtering for tracking low-elevation targets in multipath. In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, pages 452–459, Orlando, FL, September 1998.

[66] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations.* Springer Verlag, New York, NY, 2003.

[67] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47, 1996.

[68] W. Liu and I. Hwang. On hybrid state estimation for stochastic hybrid systems. *IEEE Transactions on Automatic Control*, 59(10):2615–2628, 2014.

[69] N. Berglund and B. Gentz. Beyond the Fokker-Plank equation: pathwise control of noisy bistable systems. *Journal of Physics A: Mathematical and General*, 35(9):2057–2091, 2002.

[70] S. Ditlevsen and O. Ditlevsen. Parameter estimation from observations of first-passage times of the Ornstein-Uhlenbeck process and the Feller process. *Probabilistic Engineering Mechanics*, 23(2-3):170–179, 2008.

[71] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[72] X.-R. Li and Y. Bar-Shalom. Design of an interacting multiple model algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*, 1(3):186–194, 1993.

[73] D. Riley, K. Riley, and X. Koutsoukos. Reachability analysis of stochastic hybrid systems: A biodiesel production system. *European Journal of Control*, 16(6):609–623, 2010.

[74] M. K. Ghosh, A. Arapostathis, and S. I. Marcus. Optimal control of switching diffusions with application to flexible manufacturing systems. *SIAM Journal on Control and Optimization*, 31(5):1183–1204, 1993.

[75] P. D. Hanlon and P. S. Maybeck. Multiple-model adaptive estimation using a residual correlation Kalman filter bank. *IEEE Transactions on Aerospace and Electronic Systems*, 36(2):393–406, 2000.

[76] R. J. Elliott, F. Dufour, and W. P. Malcolm. State and mode estimation for discrete-time jump Markov systems. *SIAM Journal on Control and Optimization*, 44(3):1081–1104, 2005.

[77] R. Cools. Advances in multidimensional integration. *Journal of Computational and Applied Mathematics*, 149(1):1–12, 2002.

[78] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, pages 1–49, 1998.

[79] R. V. Hogg, J. McKean, and A. T. Craig. *Introduction to Mathematical Statistics*. Pearson, 2012.

[80] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[81] Keeping the space environment safe for civil and commercial users. House of Representative, Subcommittee on Space and Aeronautics, Committee on Science and Technology, Washington, D. C., 2009.

[82] M. A. Baird. Maintaining space situational awareness and tracking it to the next level. *Air and Space Power Journal*, 27(5):50–72, 2013.

[83] T. Kelecy and M. Jah. Detection and orbit determination of a satellite executing low-thrust maneuvers. *Acta Astronautica*, 66(5-6):798–809, 2010.

[84] M. Holzinger, D. Scheeres, and K. T. Alfriend. Object correlation, maneuver detection, and characterization using control-distance metrics. *Journal of Guidance, Control, and Dynamics*, 35(4):1312–1325, 2012.

[85] S. Lemmens and H. Krag. Two-line-elements-based maneuver detection methods for satellites in low earth orbit. *Journal of Guidance, Control, and Dynamics*, 37(3):860–868, 2014.

[86] D. A. Vallado. *Fundamentals of Astrodynamics and Application*. Microcosm Press, 2013.

[87] B. Teixeira, M. Santillo, R. Erwin, and D. Bernstein. Spacecraft tracking using sampled-data Kalman filters. *IEEE Control Systems Magazine*, 28(4):78–94, 2008.

[88] R. Erwin, M. Santillo, and D. Bernstein. Spacecraft trajectory estimation using a sampled-data extended Kalman filter with range-only measurements. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3144–3149, San Diego, CA, December 2006.

[89] J. Cloutier, C.-F. Lin, and C. Yang. Enhanced variable dimension filter for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 29(3):786–797, 1993.

[90] G. Goff, D. Showalter, J. Black, and J. Beck. Parameter requirements for noncooperative satellite maneuver reconstruction using adaptive filters. *Journal of Guidance, Control, and Dynamics*, 38(3):361–374, 2015.

[91] G. Goff, J. Black, and J. Beck. Orbit estimation of a continuously thrusting spacecraft using variable dimension filters. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Kissimmee, FL, January 2015.

[92] R. L. Sackheim. Overview of united states space propulsion technology and associated space transportation systems. *Journal of Propulsion and Power*, 22(6):1310–1333, 2006.

[93] W. Wu, P. Cao, and Z. D. Pan. Maneuver target tracking based on Kalman IMM algorithm. *Advances in Electrical Engineering and Automation*, 139:479–483, 2012.

[94] D. Laneuville and Y. Bar-Shalom. Maneuvering target tracking: A Gaussian mixture based IMM estimator. In *Proceedings of the 2012 IEEE Aerospace Conference*, pages 1–12, Big Sky, MT, March 2012.

[95] C. E. Seah and I. Hwang. Terminal-area aircraft tracking using hybrid estimation. *Journal of Guidance, Control, and Dynamics*, 32(3):836–849, 2009.

[96] J. E. Prussing and B. A. Conway. *Orbital Mechanics*. Oxford Univ. Press, 1993.

[97] B. J. Hirsch. Maneuver estimation model for geostationary orbit determination. M.S. Thesis, Department of Aeronautics and Astronautics, Air Force Institute of Technology, 2006.

[98] J. Tombasco. Orbit estimation of geosynchronous objects via ground-based and space-based optical tracking. PhD Thesis, Department of Aerospace Engineering Sciences, University of Colorado, 2011.

[99] L. Beroual, D. Benatia, and N. Hedjazi. High thrust station keeping maneuvers for geostationary satellites. *International Journal of u-and e- Service, Science and Technology*, 8(1):401–414, 2015.

[100] M. L. Stone and G. P. Banner. Radars for the detection and tracking of ballistic missiles, satellites, and planets. *Lincoln Laboratory Journal*, 12(2):217–244, 2000.

[101] F. R. Hoots, P. W. Schumacher, and R. A. Glover. History of analytical orbit modeling in the u.s. space surveillance system. *Journal of Guidance, Control, and Dynamics*, 27(2):174–185, 2004.

[102] K. T. Alfriend, L. S. Breger, P. Gurfil, S. R. Vadali, and J. P. How. *Spacecraft Formation Flying: Dynamics, Control, and Navigation*. Elsevier, 2010.

[103] R. Armellin, M. Massari, and A. E. Finzi. Optimal formation flying reconfiguration and station keeping maneuvers using low thrust propulsion. In *Proceedings of the 18th International Symposium on Space Flight Dynamics*, pages 429–434, Munich, Germany, October 2004.

[104] S. Lee and S.-Y. Park. Approximate analytical solutions to optimal reconfiguration problems in perturbed satellite relative motion. *Journal of Guidance, Control, and Dynamics*, 34(4):1097–1110, 2011.

[105] G. Terejanu, P. Singla, T. Singh, and P. D. Scott. Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 31(6):1623–1633, 2008.

[106] K. Vishwajeet, P. Singla, and M. K. Jah. Nonlinear uncertainty propagation for perturbed two-body orbits. *Journal of Guidance, Control, and Dynamics*, 37(5):1415–1425, 2014.

[107] J. Horwood, N. Aragon, and A. Poore. Gaussian sum filters for space surveillance: Theory and simulations. *Journal of Guidance, Control, and Dynamics*, 34(6):1839–1851, 2011.

[108] G. L. Slater, S. M. Byram, and T. W. Williams. Collision avoidance for satellites in formation flight. *Journal of Guidance, Control, and Dynamics*, 29(5):1140–1146, 2006.

[109] S. Lee and I. Hwang. Analytical solutions to uncertainty propagation in satellite relative motion. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Boston, MA, August 2013.

[110] T. E. Carter. Optimal power-limited rendezvous for linearized equations of motion. *Journal of Guidance, Control, and Dynamics*, 17(5):1082–1086, 1994.

[111] M. Okasha and B. Newman. Guidance, navigation and control for satellite proximity operations using Tshauner-Hempel equations. *The Journal of the Astronautical Sciences*, 60(1):109–136, 2013.

[112] J. P. de Veries. Elliptic elements in terms of small increments of position and velocity components. *AIAA Journal*, 1(11):2626–2629, 1963.

[113] R. H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series, American Institute of Aeronautics and Astronautics, Inc., 1999.

[114] J. Tschauner and P. Hempel. Rendezvous zu ein min in elliptischer bahn umlaufenden ziel. *Astronautical Acta*, 11(2):104–109, 1965.

[115] K. Yamanaka and F. Ankersen. New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of Guidance, Control, and Dynamics*, 25(1):60–66, 2002.

[116] M. E. Campbell. Collision monitoring within satellite clusters. *IEEE Transactions on Control System Technology*, 13(1):42–55, 2005.

[117] P. S. Maybeck. *Stochastic Models, Estimation, and Control.* Academic Press, 1979.

[118] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control.* Taylor and Francis, New York, 1975.

[119] T. E. Carter. State transition matrices for terminal rendezvous studies: Brief survey and new example. *Journal of Guidance, Control, and Dynamics*, 21(1):148–155, 1998.

[120] H.-C. Cho, S.-Y. Park, and K.-H. Choi. Application of analytic solution in relative motion to spacecraft formation flying in elliptic orbit. *Journal of Astronomy and Space Science*, 25(3):255–266, 2008.

[121] A. J. Sinclair, R. E. Sherrill, and T. A. Lovell. Geometric interpretation of the Tschauner-Hempel solutions for satellite relative motion. *Advances in Space Research*, 55(9):2268–2279, 2015.

[122] N. B. Sarter and D. D. Woods. How in the world did we ever get into that mode? mode error and awareness in supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):5–19, 1995.

[123] A. Degani, M. Shafto, and A. Kirlik. Modes in automated cockpits: Problems, data analysis and a modelling framework. In *Proceedings of the 36th Israel Annual Conference on Aerospace Sciences*, pages 258–266, Haifa, Israel, February 1996.

[124] A. Degani, M. Heymann, G. Meyer, and M. Shafto. Some formal aspects of human-automation interaction. Technical report, NASA TM-209600, 2000.

[125] J. Rushby, J. Crow, and E. Palmer. An automated method to detect potential mode confusions. In *Proceedings of the 18th Digital Avionics Systems Conference*, pages 4.b.2.–1–14.b.2–6, St. Louis, MO, November 1999.

[126] E. J. Bass, K. M. Feigh, E. Gunter, and J. Rushby. Formal modeling and analysis for interactive hybrid systems. In *Proceedings of the 4th International Workshop on Formal Methods for Interactive Systems*, 2011.

[127] A. Andre and A. Degani. Do you know what mode you're in? an analysis of mode error in everyday things. *Human-automation interaction: Research and practice*, pages 19–28, 1997.

[128] J. Crow, D. Javaux, and J. Rushby. Models and mechanized methods that integrate human factors into automation design. In *Proceedings of International Conference on Human-Computer Interaction in Aeronautics*, pages 163–168, Toulouse, France, September 2000.

[129] S. Miller and J. N. Potts. Detecting mode confusion through formal modeling and analysis. Technical report, NASA CR-208971, 1999.

[130] R. Butler, S. Miller, J. Potts, and V. Carreno. A formal methods approach to the analysis of mode confusion. In *Proceedings of the 17th Digital Avionics Systems Conference*, Bellevue, WA, November 1998.

[131] S. Budalakoti, A. N. Srivastave, and M. E. Otey. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on System, Man, and Cybernetics*, 39(1):101–113, 2009.

[132] S. Budalakoti, A. N. Srivastave, and R. Akella. Discovering atypical flights in sequences of discrete flight parameters. In *Proceedings of the 2006 IEEE Aerospace Conference*, pages 1–8, Big Sky, MT, March 2006.

[133] D. L. Iverson. Inductive system health monitoring. In *Proceedings of the 2004 International Conference on Artificial Intelligence*, Las Vegas, NV, June 2004.

[134] S. Das, L. Li, A. N. Srivastava, and R. J. Hansman. Comparison of algorithms for anomaly detection in flight recorder data of airline operations. In *Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations Conference*, Indianapolis, IN, September 2012.

[135] B. Matthews, S. Das, K. Bhaduri, K. Das, R. Martin, and N. Oza. Discovering anomalous aviation safety events using scalable data mining algorithms. *Journal of Aerospace Information System*, 10(10):467–475, 2013.

[136] L. Li, M. Gariel, R. J. Hansman, and R. Palacios. Anomaly detection in onboard-recorded flight data using cluster analysis. In *Proceedings of the 30th Digital Avionics Systems Conference*, pages 4A4–1–4A4–11, Seattle, WA, October 2011.

[137] S. Das, B. Matthews, A. N. Srivastava, and N. Oza. Multiple kernel learning for heterogeneous anomaly detection: Algorithm and aviation safety case study. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 47–56, Washington, D. C., July 2010.

[138] E. Chu, D. Gorinevsky, and S. P. Boyd. Detecting aircraft performance anomalies from cruise flight data. In *Proceedings of the 2010 AIAA Infotech@Aerospace*, Atlanta, GA, April 2010.

[139] J. S. Nandiganahalli, S. Lee, I. Hwang, and B.-J. Yang. User interface validation using mode confusion detection. In *Proceedings of AIAA Modeling and Simulation Technologies Conference*, Atlanta, GA, June 2014.

[140] Y. Zhao, C. Haissig, and M. J. Hoffman. Analysis of pilot intent parameters in air traffic management. In *Proceedings of the 1998 American Control Conference*, pages 1789–1792, Philadelphia, PA, June 1998.

[141] R. Barhydt and A. W. Warren. Newly enacted intent changes to ads-b masps: Emphasis on operations, compatibility, and integrity. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, August 2002.

[142] C. E. Seah, A. Aligawesa, and I. Hwang. Algorithm for conformance monitoring in air traffic control. *Journal of Guidance, Control, and Dynamics*, 33(2):500–509, 2010.

[143] M. Peters and M. A. Konyak. The engineering analysis and design of the aircraft dynamics model for the faa target generation facility. Technical report, FAA 99162-01, 2012.

[144] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.

[145] H. J. Kushner and P. G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time.* Springer Verlag, London, U.K., 1992.

APPENDICES

A. Design of Transition Probabilities π in Chapter 2

How to choose the transition probabilities of the Markov chain is a design problem [145]. In this appendix, an example of how to design the appropriate π in (2.22) is presented. For simplicity, we consider a two-dimensional case of (2.1) (i.e., $n = 2$) whose state is $\mathbf{X} = [X_1 \ X_2]^T \in \mathbb{R}^2$ and whose evolution is governed by the following SDE:

$$dX_i = a_i(\mathbf{X}, t)dt + b_i(\mathbf{X}, t)dW_i \quad \text{for } i = 1, 2 \tag{A.1}$$

where a_1, a_2, b_1, and b_2 are scalar functions, and $\mathbf{W} = [W_1 \ W_2]^T$ is the standard two-dimensional Brownian motion. Let $\{\mathbf{Q}_j\}$ be the Markov chain to approximate $\mathbf{X}(t)$ and denote the Markov chain's state by $\mathbf{q} = [q_1 \ q_2]^T \in \mathbb{Z}^2$. Define

$$\begin{aligned}
\pi_1^+ :&= \pi(q_1 + 1, q_2) = \frac{1}{C_1} \exp\left(\frac{\epsilon_1 a_1(\mathbf{q})}{(b_1(\mathbf{q}))^2}\right) \\
\pi_1^0 :&= \pi(q_1, q_2) = \frac{1}{C_1} \left(\frac{2}{\lambda(b_1(\mathbf{q}))^2} - 2\right) \\
\pi_1^- :&= \pi(q_1 - 1, q_2) = \frac{1}{C_1} \exp\left(-\frac{\epsilon_1 a_1(\mathbf{q})}{(b_1(\mathbf{q}))^2}\right) \\
\pi_2^+ :&= \pi(q_1, q_2 + 1) = \frac{1}{C_2} \exp\left(\frac{\epsilon_2 a_2(\mathbf{q})}{(b_2(\mathbf{q}))^2}\right) \\
\pi_2^0 :&= \pi(q_1, q_2) = \frac{1}{C_2} \left(\frac{2}{\lambda(b_2(\mathbf{q}))^2} - 2\right) \\
\pi_2^- :&= \pi(q_1, q_2 - 1) = \frac{1}{C_2} \exp\left(-\frac{\epsilon_2 a_2(\mathbf{q})}{(b_2(\mathbf{q}))^2}\right)
\end{aligned} \tag{A.2}$$

where C_1 and C_2 are normalizing constants such that $\pi_1^+ + \pi_1^0 + \pi_1^- = 1$ and $\pi_2^+ + \pi_2^0 + \pi_2^- = 1$. Then, the transition probabilities π can be obtained by

$$\begin{aligned}
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [1\ 1]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^+ \pi_2^+ \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [0\ 1]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^0 \pi_2^+ \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [-1\ 1]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^- \pi_2^+ \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [1\ 0]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^+ \pi_2^0 \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [0\ 0]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^0 \pi_2^0 \qquad \text{(A.3)} \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [-1\ 0]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^- \pi_2^0 \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [1\ \ -1]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^+ \pi_2^- \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [0\ \ -1]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^0 \pi_2^- \\
&\Pr\{\mathbf{Q}_{j+1} = \mathbf{q} + [-1\ \ -1]^T | \mathbf{Q}_j = \mathbf{q}\} = \pi_1^- \pi_2^-
\end{aligned}$$

The transition probabilities designed above satisfy the convergence conditions of Theorem 2.3.1.

B. Proof of Theorem 3.3.1 in Chapter 3

To prove Theorem 3.3.1, let us consider the following definitions and theorem which are related to quasi-Monte Carlo approximation.

Definition B.0.1 (Discrepancy [78]) *Consider the s-dimensional unit cube $[0,1]^s$ in $\mathbb{R}^s$. The Discrepancy D_N of a set of N points $y_1, y_2, \dots, y_N \in [0,1]^s$ is defined as:*

$$D_N = \sup_{J \subset [0,1]^s} \left| \frac{\textit{number of points in } J}{N} - \textit{volume}(J) \right| \tag{B.1}$$

where J is a rectangular subspace in $[0,1]^s$ with sides parallel to the coordinate axes and with one vertex at 0.

Definition B.0.2 (Hardy and Krause Total Variation [78]) *Let f is a smooth function on $\mathbb{R}^s$. For all $k \leq s$ and all sets of k integers such that $1 \leq i_1 < i_2 < \cdots < i_k \leq s$, define the quantity*

$$V^k(f; i_1, \dots, i_k) = \int_{[0,1]^k} \left| \frac{\partial^k f}{\partial x_{i_1} \dots \partial x_{i_k}} \right|_{x_j = 1, j \neq i_1, \dots, i_k} dx_{i_1} \cdots dx_{i_k} \tag{B.2}$$

Then, the variation of f on $[0,1]^s$ in the sense of Hardy and Krause is defined as:

$$V(f) = \sum_{k=1}^{s} \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq s} V^{(k)}(f; i_1, \dots, i_k) \tag{B.3}$$

Theorem B.0.1 (Koksma-Hlawka Inequality [77]) *If f has bounded variation $V(f)$ on $[0,1]^s$ in the sense of Hardy and Krause, then, for any $y_1, y_2, \dots, y_N \in [0,1]^s$,*

$$\left| \int_{[0,1]^s} f(x) dx - \frac{1}{N} \sum_{n=1}^{N} f(y_n) \right| \leq V(f) D_N \tag{B.4}$$

Theorem B.0.1 indicates that when the deterministic points have a lower discrepancy, the corresponding approximation error for the integral of f is bounded by a smaller value. Therefore, in order to prove Theorem 3.3.1, we need to show that the designed grid points (i.e., pseudo-measurements) in (3.17)-(3.20) have a discrepancy which is inversely proportional to the sizes of N_i, $i = 1, 2, \ldots, n_{S(k)}$. Without loss of generality, assume that the domain of integral (3.15) is scaled to $[0,1]^{n_{S(k)}}$ with a proper transformation, and thus, the grid size ϵ_l in (3.17) is now $\frac{1}{N_l}$ for $l = 1, 2, \ldots, n_{S(k)}$. Then, the following lemma computes the discrepancy of the designed grid points.

Lemma B.0.1 *The discrepancy of the set of the grid points* $\left\{y^d_{pseudo}\right\}$ *designed in (3.17)-(3.20) is*

$$D_{N_{tot}} = \frac{\prod_{j=1}^{n_{S(k)}}(N_j - 1) - \prod_{j=1}^{n_{S(k)}} N_j}{\prod_{j=1}^{n_{S(k)}} N_j} \tag{B.5}$$

Proof For a given grid point $d := [d_1, d_2, \ldots, d_{n_{S(k)}}]^T \in \mathbb{N}^{n_{S(k)}}$ such that $1 \le d_j \le N_j - 1$ for $j = 1, 2, \ldots, n_{S(k)}$, consider a rectangle J^0 in $[0,1]^{n_{S(k)}}$ (with sides parallel to the coordinate axes and with one vertex at 0) whose length along each axis is $\frac{1}{N_j} \times d_j$. So, the rectangle J^0 contains $\prod_{j=1}^{n_{S(k)}} d_j$ grid points. Define J' as the rectangle whose volume is largest among rectangles only containing the same grid points as J^0. Therefore, the length of J' along each axis is less than $\frac{1}{N_j} \times (d_j + 1)$. Now consider an arbitrary rectangle J such that $J^0 \subseteq J \subseteq J'$. Then, we have

$$0 \le \left| \frac{\text{number of points in } J}{N_{tot}} - \text{volume}(J) \right| < \left| \frac{\prod_{j=1}^{n_{S(k)}} d_j}{N_{tot}} - \frac{\prod_{j=1}^{n_{S(k)}} (d_j + 1)}{N_{tot}} \right| \tag{B.6}$$

This is true for all the grid points, and the bound (the rightmost term in Eq. (B.6)) has the largest value when $d_j = N_j - 1$ for $j = 1, 2, \ldots, n_{S(k)}$. Therefore, the discrepancy of the set of the grid points is

$$D_{N_{tot}} = \frac{\prod_{j=1}^{n_{S(k)}}(N_j - 1) - \prod_{j=1}^{n_{S(k)}} N_j}{\prod_{j=1}^{n_{S(k)}} N_j} \tag{B.7}$$

which is decreasing as the sizes of N_i, $i = 1, 2, \ldots, n_{S(k)}$ are increasing. ■

Using Theorem B.0.1 and Lemma B.0.1, we have

$$\begin{aligned} \epsilon &\le V(p(y(k)|x(k), q(k) = j, \mathcal{I}^{k-1}))D_{N_{tot}} \\ &= V(p(y(k)|x(k), q(k) = j, \mathcal{I}^{k-1})) \left(\frac{\prod_{i=1}^{n_{S(k)}}(N_i - 1) - \prod_{i=1}^{n_{S(k)}} N_i}{\prod_{i=1}^{n_{S(k)}} N_i} \right) \end{aligned} \tag{B.8}$$

Because the integrand $p(y(k)|x(k), q(k) = j, \mathcal{I}^{k-1})$ is a Gaussian function, its variation $V(p(y(k)|x(k), q(k) = j, \mathcal{I}^{k-1}))$ is finite. In addition, we have

$$\lim_{N_i \to \infty, i=1,\ldots,n_{S(k)}} \left(\frac{\prod_{i=1}^{n_{S(k)}} (N_i - 1) - \prod_{i=1}^{n_{S(k)}} N_i}{\prod_{i=1}^{n_{S(k)}} N_i} \right) = 0 \tag{B.9}$$

Therefore, the approximation error ϵ converges to zero as $N_i \to \infty$ $(i = 1, 2, \ldots, n_{S(k)})$.

C. Explicit Expression for $\mathbf{K}(E)$ in Chapter 5

In this appendix, the elements of $\mathbf{K}(E)$ are explicitly presented. First, note that $\mathbf{U}$ is symmetric and thus can be defined by

$$\mathbf{U} \equiv \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{12} & U_{22} & U_{23} \\ U_{13} & U_{23} & U_{33} \end{bmatrix}$$

Then, using (5.9), (5.37), and (5.38), each element K_{ij} in the ith row and jth column of $\mathbf{K}(E)$ can be obtained as:

$$K_{11} = \zeta_1 U_{22},\ K_{12} = K_{21} = -\zeta_2 U_{12} + (\zeta_3 + \zeta_4) U_{22},\ K_{13} = K_{31} = \zeta_5 U_{23}$$

$$K_{14} = K_{41} = -\zeta_3 U_{12} + (-\zeta_2 - \zeta_5) U_{22},\ K_{15} = K_{51} = (3e\zeta_6 - 2\zeta_1) U_{12} + 3\zeta_7 U_{22}$$

$$K_{16} = K_{61} = \zeta_4 U_{23},\ K_{22} = \zeta_8 U_{11} + (-2\zeta_9 - 2\zeta_{10}) U_{12} + (\zeta_{11} + 2\zeta_{12} + \zeta_{13}) U_{22}$$

$$K_{23} = K_{32} = -\zeta_{14} U_{13} + (\zeta_{10} + \zeta_{15}) U_{23}$$

$$K_{24} = K_{42} = \zeta_9 U_{11} + (\zeta_8 - \zeta_{12} - \zeta_{11} + \zeta_{14}) U_{12} + (-\zeta_9 - 2\zeta_{10} - \zeta_{15}) U_{22}$$

$$\begin{aligned} K_{25} = K_{52} = (2\zeta_2 - 3e\zeta_{16}) U_{11} + (3e\zeta_{17} - 2\zeta_4 - 3\zeta_{18} - 2\zeta_3 + 3e\zeta_{19}) U_{12} \\ +3(\zeta_{20} + \zeta_{21}) U_{22} \end{aligned}$$

$$K_{26} = K_{62} = -\zeta_{10} U_{13} + (\zeta_{12} + \zeta_{13}) U_{23},\ K_{33} = \zeta_{22} U_{33},$$

$$K_{34} = K_{43} = -\zeta_{10} U_{13} + (-\zeta_{14} - \zeta_{22}) U_{23}$$

$$K_{35} = K_{53} = (3e\zeta_{23} - 2\zeta_5) U_{13} + 3\zeta_{24} U_{23},\ K_{36} = K_{63} = \zeta_{15} U_{33}$$

$$K_{44} = \zeta_{11} U_{11} + (2\zeta_9 + 2\zeta_{10}) U_{12} + (\zeta_8 + 2\zeta_{14} + \zeta_{22}) U_{22}$$

$$K_{45} = K_{54} = (2\zeta_3 - 3e\zeta_{17})U_{11} + (2\zeta_2 + 2\zeta_5 - 3\zeta_{20} - 3e\zeta_{16} - 3e\zeta_{23})U_{12}$$

$$+(-3\zeta_{18} - 3\zeta_{24})U_{22}$$

$$K_{46} = K_{64} = -\zeta_{12}U_{13} + (-\zeta_{10} - \zeta_{15})U_{23}$$

$$K_{55} = (4\zeta_1 - 12e\zeta_6 + 9e^2\zeta_{25})U_{11} + (18e\zeta_{26} - 12\zeta_7)U_{12} + 9\zeta_{27}U_{22}$$

$$K_{56} = K_{65} = (3e\zeta_{19} - 2\zeta_4)U_{13} + 3\zeta_{21}U_{23}, \ K_{66} = \zeta_{13}U_{33}$$

where

$$\zeta_1 = \frac{-(e\cos E - 1)^3}{(1-e^2)^{7/2}}, \ \zeta_2 = \frac{(e - \cos E)(e\cos E - 1)}{(1-e^2)^{5/2}},$$

$$\zeta_3 = \frac{-\sin E(e\cos E - 1)}{(1-e^2)^2} \ \zeta_4 = \frac{\sin E(e\cos E - 1)^2}{(1-e^2)^3},$$

$$\zeta_5 = \frac{-(e - \cos E)(e\cos E - 1)^2}{(1-e^2)^{7/2}}$$

$$\zeta_6 = \frac{\sin E(e\cos E - 1)(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^{7/2}},$$

$$\zeta_7 = \frac{(e\cos E - 1)(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^3}$$

$$\zeta_8 = \frac{-(e - \cos E)^2}{(1-e^2)^{3/2}(e\cos E - 1)}, \ \zeta_9 = \frac{\sin E(e - \cos E)}{(1-e^2)(e\cos E - 1)},$$

$$\zeta_{10} = \frac{-\sin E(e - \cos E)}{(1-e^2)^2} \ \zeta_{11} = \frac{\cos^2 E - 1}{(1-e^2)^{1/2}(e\cos E - 1)},$$

$$\zeta_{12} = \frac{\sin^2 E}{(1-e^2)^{3/2}}, \ \zeta_{13} = \frac{(\cos^2 E - 1)(e\cos E - 1)}{(1-e^2)^{5/2}}$$

$$\zeta_{14} = \frac{(e - \cos E)^2}{(1-e^2)^{5/2}}, \ \zeta_{15} = \frac{\sin E(e - \cos E)(e\cos E - 1)}{(1-e^2)^3}$$

$$\zeta_{16} = \frac{-\sin E(e - \cos E)(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^{5/2}(e\cos E - 1)}$$

$$\zeta_{17} = \frac{\sin^2 E(E_0 - E) - e\sin^2 E(\sin E_0 - \sin E)}{(1-e^2)^2(e\cos E - 1)},$$

$$\zeta_{18} = \frac{-(e - \cos E)(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^2(e\cos E - 1)}$$

$$\zeta_{19} = \frac{-\sin^2 E(E_0 - E) + e\sin^2 E(\sin E_0 - \sin E)}{(1-e^2)^3},$$

$$\zeta_{20} = \frac{\sin E(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^{3/2}(e\cos E - 1)}$$

$$\zeta_{21} = \frac{-\sin E(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^{5/2}},$$

$$\zeta_{22} = \frac{-(e-\cos E)^2(e\cos E - 1)}{(1-e^2)^{7/2}}$$

$$\zeta_{23} = \frac{\sin E(e-\cos E)(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^{7/2}},$$

$$\zeta_{24} = \frac{(e-\cos E)(E_0 - E - e\sin E_0 + e\sin E)}{(1-e^2)^{3}}$$

$$\zeta_{25} = \frac{-\sin^2 E(E_0 - E - e\sin E_0 + e\sin E)^2}{(1-e^2)^{7/2}(e\cos E - 1)},$$

$$\zeta_{26} = \frac{-\sin E(E_0 - E - e\sin E_0 + e\sin E)^2}{(1-e^2)^{3}(e\cos E - 1)}$$

$$\zeta_{27} = \frac{-(E_0 - E - e\sin E_0 + e\sin E)^2}{(1-e^2)^{5/2}(e\cos E - 1)}$$

VITA

VITA

Sangjin Lee received the B.S. (summa cum laude) and M.S degrees in Astronomy from Yonsei University, Seoul, Korea, in 2008 and 2010, respectively. His research during the M.S degree was specialized in spaceflight dynamics and control, and its application to satellite formation flying. He begun to pursue a doctorate degree in Aeronautics and Astronautics at Purdue University in 2011. His research focuses on control and information inference of Cyber-Physical Systems with various aerospace applications including air transportation systems, integrated human-automation systems, and networked unmanned aircraft systems.