INCOMPLETE INFORMATION PURSUIT-EVASION GAMES WITH APPLICATIONS TO SPACECRAFT RENDEZVOUS AND MISSILE DEFENSE

A Dissertation

by

KURT DALE AURES-CAVALIERI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	John E. Hurtado
Co-Chair of Committee,	John L. Junkins
Committee Members,	S. Rao Vadali
	Aniruddha Datta
Head of Department,	Rodney D. W. Bowersox

December 2014

Major Subject: Aerospace Engineering

ABSTRACT

Pursuit-evasion games reside at the intersection of game theory and optimal control theory. They are often referred to as differential games because the dynamics of the relative system are modeled by the pursuer and evader differential equations of motion. Pursuit-evasion games diverge from traditional optimal control problems due to the participation of multiple intelligent agents with conflicting goals. Individual goals of each agent are defined through multiple cost functions and determine how each player will behave throughout the game. The optimal performance of each player is dependent upon how much knowledge they have about themselves, their opponent, and the system. Complete information games represent the ideal case in which each player can truly play optimally because all pertinent information about the game is readily available to each player.

Player performance in a pursuit-evasion game greatly diminishes as information availability moves further from the ideal case and approaches the most realistic scenarios. Methods to maintain satisfactory performance in the presence of incomplete, imperfect, and uncertain information games is very desirable due to the application of optimal pursuit-evasion solutions to high-risk missions including spacecraft rendezvous and missile interception. Behavior learning techniques can be used to estimate the strategy of an opponent and augment the pursuit-evasion game into a one-sided optimal control problem. The application of behavior learning is identified in final-time-fixed, infinite-horizon, and final-time-free situations. A two-step dynamic inversion process is presented to fit systems with nonlinear kinematics

and dynamics into the behavior learning framework for continuous, linear-quadratic games. These techniques are applied to minimum-time, spacecraft reorientation, and missile interception examples to illustrate the advantage of these techniques in real-world applications when essential information is unavailable.

To my brother, Mikey.

ACKNOWLEDGEMENTS

The amount of effort exhausted over the past six years has been staggering. I could not have succeeded without the unparalleled support from my family, friends, and colleagues. I would first like to thank my parents, Margie and Tony, for believing in me from the start. I want to thank my brother, Mikey, for keeping me grounded yet always trying to one-up me. I would like to express my sincere gratitude to my better half, Brittany, for giving me the ability to stay focused and the courage to push through the final stages. Thank you.

I simply cannot thank my advisors, Drs. John E. Hurtado and John L. Junkins, enough for their guidance, their support, and the confidence they put in me throughout my graduate career. We have built such strong relationships while working on some of the most amazing projects. There was never a doubt in my mind about whether or not I made the best decision when joining their technical family. I will always cherish my time spent at Texas A&M and The Land, Air, and Space Robotics Laboratory.

I would also like to thank my committee members Drs. S. Rao Vadali and Aniruddha Datta for their valuable input to the work presented in this dissertation. Additionally, I want to thank the very patient staff members that kept me in line during my stay at Texas A&M including Lisa, Karen, Andrea, and Yolanda.

From The LASR Lab, I want to thank my colleagues Brent, Clark, Austin, and Dylan for their efforts over the past five years. We have been able to turn drawings on a whiteboard into impressive hardware realizations. Thanks for making the countless

late nights somewhat more bearable through an endless supply of loud music, cheap food, and Mountain Dew. I would like to thank my roommates, Brian, Alex, and Jimmy, for helping me relax when I would sometimes forget how to. I'm honored to have such great friends. Special thanks to Drs. Jeremy Davis, James Doebbler, Manoranjan Majji, and Julie Parish for all taking on mentoring roles in one way or another. I am forever grateful.

Finally, I would like to acknowledge that this work was supported by the Bradley Fellowship and Texas Space Grant Consortium Fellowship.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

TABLE Page

CHAPTER I

INTRODUCTION

Pursuit-evasion (PE) games reside at the intersection of game theory and optimal control theory. More specifically, these scenarios fall within the confines of differential games (DGs) because the dynamics of the relative system are modeled by the pursuer and evader differential equations of motion. Differential games diverge from traditional optimal control problems (OCPs) due to the participation of multiple intelligent agents with conflicting goals. Individual goals of each agent are defined through multiple cost functions - or objective functions, or performance indices - and determine how each player will behave throughout the game. Additionally, the dynamic constraint found in traditional OCPs is now being modified by the control input of multiple players instead of a single agent. The concept of DGs was widely publicized by Issacs when studying the homicidal chauffeur problem [1].

The optimal performance of each player is dependent upon how much knowledge they have about themselves, their opponent, and the system. Traditionally, PE games are investigated through computer-based simulations under the assumption of a complete information game. In a complete information game, each player has access to all relevant information pertaining to the game. This information includes the exact differential equations governing the motion of each agent, the control input of each agent, and the behavior model each agent has assumed. The behavior of each player includes their objective and strategy. Player objectives are defined by the form of their cost function and strategy is determined by the selection of gains found within

the cost function. Complete information games represent the ideal case in which each player can truly play optimally because all information about the game is readily available. The more realistic cases, however, are subject to incomplete information, imperfect information, and uncertain information.

Incomplete information games are those in which one or more key pieces of information about the system is not known precisely. In most incomplete information games, the behavior of an opponent is unknown. This could mean the evader is not aware of the pursuer's strategy or vice versa. Most often, neither player is aware of their opponent's strategy. When an opponent's strategy is unknown, a player is unable to predict how that opponent will behave and therefore must play more conservatively by assuming the opponent is attempting to achieve the exact opposite objective using the same strategy. For example, if the evader's behavior is unknown to the pursuer and the pursuer is attempting to minimize a particular cost function with specific gains, then the pursuer will assume the evader's behavior is dictated by maximizing the same cost function with the same gain selections. This conservative play is referred to as a zero-sum strategy because the pursuer is assuming the evader is playing the exact opposite game and therefore the sum of both optimal games is zero. Behavior learning methods for incomplete information games have been developed but rely on the assumption of a perfect information game [2].

Perfect information games assume the relative states are known exactly and control history of the opponent is available. This is convenient because it allows batch estimation methods to be used to determine the strategy of an opponent [2]. In real-world scenarios, however, relative state measurements are subject to measurement

noise and the control history of an opponent is unknown. Therefore, more advanced methods must be used to help a player play more effectively than the conservative zero-sum strategy. If the relative state measurements contain errors and the control history of an opponent is unknown, then the game is referred to as an imperfect information game.

By introducing modeling errors, it is possible to take the game further into the realistic realm. Traditionally, simulations assume perfect modeling of the differential equations which govern the motion of each player. The introduction of modeling errors produces a disconnect between how each player believes the relative system will evolve over time and how it actually evolves. This discontinuity has significant effects on the outcome of a PE game and must be considered in any real-world application. A game in which modeling errors are present is referred to as an uncertain information game. Currently, methods designed to overcome the issues associated with imperfect and uncertain information games are unavailable.

Pursuit-evasion games can take on any level of information availability. The scope of the work and results presented here will consider all types of information characteristics. Complete information games will be used as a baseline to aid in the evaluation of the presented methods. Incomplete information will be used to describe simulations which experience imperfect and incomplete information because measurement noise is a common factor in all realistic applications. Uncertain information will be used to describe simulations which experience the characteristics of imperfect, incomplete, and uncertain information. These definitions are summarized in Fig. I.1 and have been presented in this manner to allow for unambiguous

descriptions of the simulations. Of course, PE games can take on any combination of the information characteristics found in Fig. I.1. Pursuit-evasion games have a

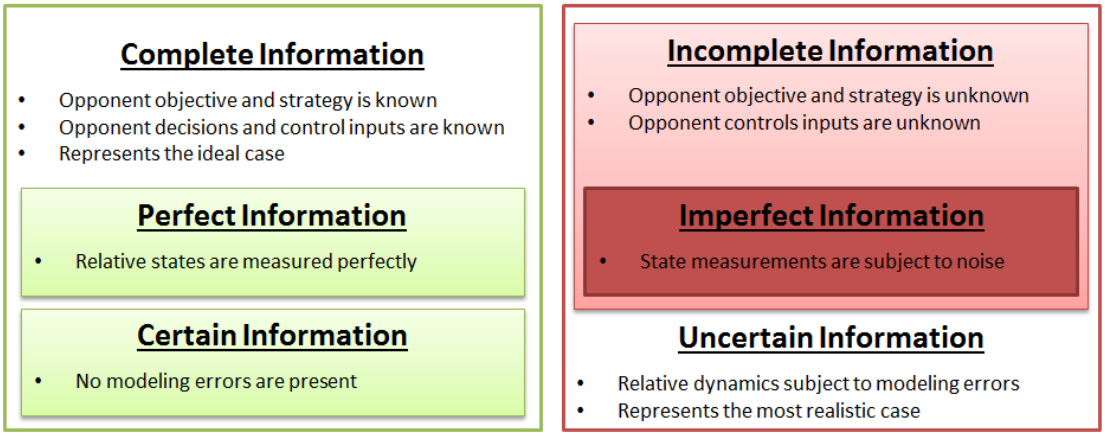

Figure I.1. Information Characteristics of Pursuit-Evasion Games

wide variety of aerospace applications. Most scenarios involving multiple vehicles with differing objectives that wish to behave optimally in some way can be cast into a PE framework. Spacecraft rendezvous missions, where a capture spacecraft is attempting to dock with an uncooperative or retired satellite, can be modeled as a PE problem. The missile interception problem in which a vehicle is required to intercept another before a destination is reached also fits into this framework. Aerial tracking of one or more ground vehicles has the basic characteristics of a PE-type problem along with other extensions of this work including tasking and spoofing methods. These examples may involve multiple pursuers or evaders and the concepts can be applied to any combination of vehicle types including aircraft, spacecraft, and ground vehicles. Because PE games can be made up of an infinity number of combinations

of pursuers and evaders, we will limit our discussion and focus on the one-pursuer, one-evader scenario.

One major obstacle associated with real-world optimal control problems is that the differential equations governing the dynamic constraints are commonly nonlinear. This can pose an issue because nonlinear optimal control problems require iterative, numerical solutions which add additional constraints on time and processing power, depending on the application. Linear pursuit-evasion games have been studied in depth and optimal control solutions to those types of problems are well known [3, 4]. Moreover, the need for feedback solutions will be stressed. Feedback solutions are essential to pursuit-evasion because players are not required to play a certain way. They could take on any strategy at any time and those decisions can be exploited by feedback control schemes. It would be ideal if nonlinear PE problems could be transformed into their linear counterparts and the familiar linear techniques could be applied. This allows the focus to remain on the behavior learning aspects associated with the incomplete, imperfect, and uncertain information games instead of iterative solutions. A useful method involves applying dynamic inversion to the system such that the PE game can be designed with a convenient linear system. Dynamic inversion can be a very useful tool for transforming complex control problems into more manageable, but still effective systems. This concept has been proven in simulation and on test flights of advanced weapon systems [5].

I.A. Motivation

To gain a better understanding of how a player's performance breaks down with the varying levels of information availability, we will examine a specific pursuit-evasion scenario and how the solution changes as information about the players and system is revoked.

Consider differentially driven vehicles in the horizontal plane where each player can control the magnitude of their forward velocity, v, and their turn rate, ω. The no-slip kinematic model of a single player is defined as [6]

$$\dot{x} = v\cos\theta \,, \tag{1.1}$$

$$\dot{y} = v\sin\theta \,, \tag{1.2}$$

$$\dot{\theta} = \omega \,, \tag{1.3}$$

where x and y represent the player position with respect to the inertial reference frame and θ represents the orientation of the body-fixed reference frame with respect to the inertial frame. The concept of flat dynamics can be applied to this system to allow for a linear dynamic representation [7].

For Player i, the desired state representation is given by

$$\boldsymbol{z}_i = [x_i,\, y_i,\, \dot{x}_i,\, \dot{y}_i]^T = [z_{i_1},\, z_{i_2},\, z_{i_3},\, z_{i_4}]^T \,, \tag{1.4}$$

$$\dot{\boldsymbol{z}}_i = [z_{i_3},\, z_{i_4},\, u_{i_1},\, u_{i_2}]^T \,. \tag{1.5}$$

Differentiation of Eqns. 1.1 and 1.2 lead to the definition of the true control inputs,

v_i and ω_i, in terms of the state vector, $\boldsymbol{z}$, and new controls, u_{i_1} and u_{i_2}.

$$v_i = \sqrt{z_{i_3}^2 + z_{i_4}^2}\,, \tag{1.6}$$

$$\omega_i = \frac{z_{i_3}u_{i_2} - z_{i_4}u_{i_1}}{z_{i_3}^2 + z_{i_4}^2}\,. \tag{1.7}$$

Note the denominator found in Eqn. 1.7 is equivalent to v_i^2 and must not be equal to zero for computation of the true controls. In practice, this is done by having non-zero initial conditions [8].

By defining $\boldsymbol{u}_i = [u_{i_1}, u_{i_2}]^T$, Eqn. 1.5 can be written in the vector-matrix form

$$\dot{\boldsymbol{z}}_i = A\boldsymbol{z}_i + B\boldsymbol{u}_i\,, \tag{1.8}$$

where

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{1.9}$$

Formulation of the relative state vector is given by

$$\boldsymbol{z} = \boldsymbol{z}_p - \boldsymbol{z}_e\,, \tag{1.10}$$

where subscripts p and e denote the state vector of the pursuer and evader, respectively. The motivational PE game is then defined by the zero-sum cost function

$$J = \frac{1}{2}\boldsymbol{z}_f^T S_f \boldsymbol{z}_f + \frac{1}{2}\int_{t_0}^{t_f} \left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p - \boldsymbol{u}_e^T R_e \boldsymbol{u}_e\right) \mathrm{d}t\,, \tag{1.11}$$

subject to

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u}_p - B\boldsymbol{u}_e\,. \tag{1.12}$$

These equations make up a linear-quadratic, final-time-fixed game where subscript f dictates the vector or matrix at the final time, $t_f = 30$. First, the results for the complete information case will be shown as a baseline. Throughout these simulations, the evader will be subject to an imperfect information game and assume a zero-sum safe strategy in an effort to illustrate how the performance of the pursuer degrades when its assumption of the evader's strategy is inaccurate. The true gain selection for both players is summarized by

$$S_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.009 & 0 & 0 & 0 \\ 0 & 0.009 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$R_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_e = \begin{bmatrix} 1.09 & 0 \\ 0 & 1.09 \end{bmatrix}. \tag{1.13}$$

Figures I.2 - I.4 show the results for the complete information, zero-sum pursuit-evasion game in which the pursuer and evader implement the same zero-sum cost function described in Eqn. 1.11. Players start near the origin with non-zero initial conditions. The aerial view of the players' trajectories are shown in Fig. I.2. Figure I.3 contains plots of the relative states along with the total relative displacement and speed while Fig. I.4 presents the cumulative cost and cost-to-go for each player. The total cost for the pursuer and the evader are 1.0167×10^2 for this example. Note that final cost, cumulative cost, and cost-to-go are identical for the pursuer and evader because of the complete information, zero-sum strategy implementation.

To illustrate the effects of incomplete information, the same simulation was run

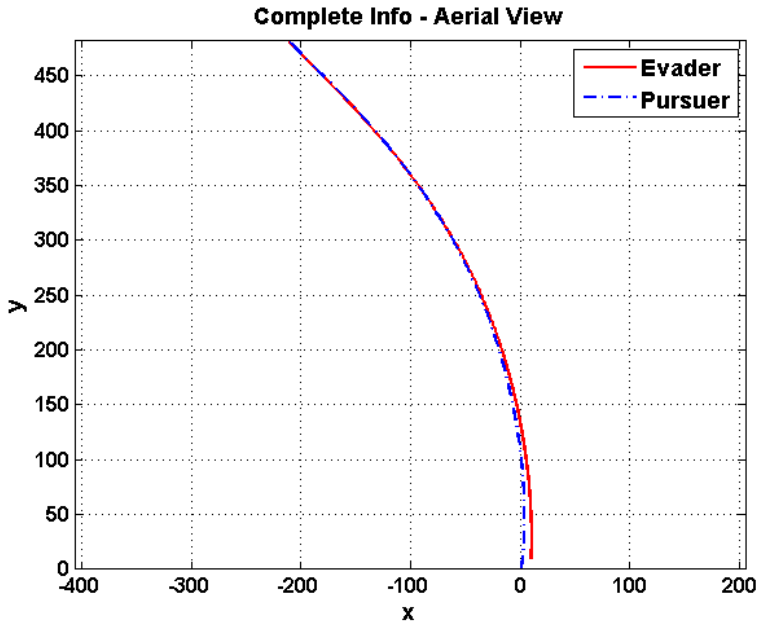

Figure I.2. Final-Time-Fixed, Complete Information Aerial View

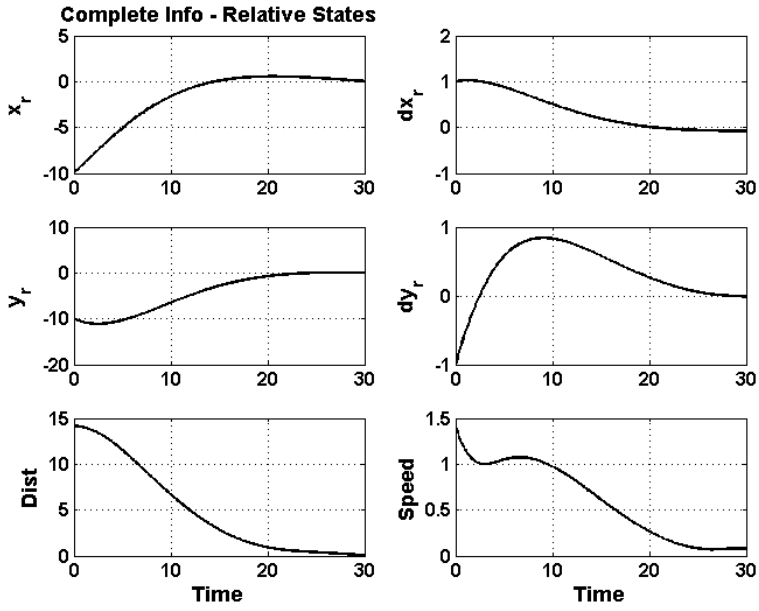

Figure I.3. Final-Time-Fixed, Complete Information Relative States

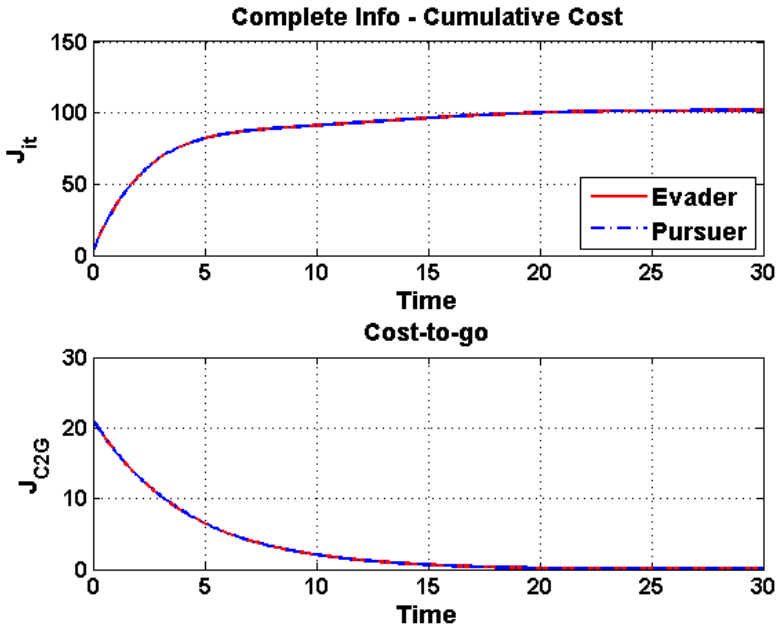

Figure I.4. Final-Time-Fixed, Complete Information Cost Analysis

but slightly different gains were assumed by the pursuer while those of the evader remained constant. The pursuer's gain selection for the incomplete information case is summarized by

$$S_{f_p} = \begin{bmatrix} 0.95 & 0 & 0 & 0 \\ 0 & 0.95 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_p = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{1.14}$$

and

$$R_{p_p} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_{e_p} = \begin{bmatrix} 1.1 & 0 \\ 0 & 1.1 \end{bmatrix}. \tag{1.15}$$

The results for the incomplete, imperfect information game using the same initial conditions are shown in Figs. I.5 - I.7. The aerial view of the players' trajectories

are shown in Fig. I.5. Figure I.6 contains plots of the relative states along with the total relative displacement and speed while Fig. I.7 presents the cumulative cost and cost-to-go for each player. The total cost of the pursuer is 1.9781×10^3 while that for the evader is 1.9833×10^3. The introduction of incomplete information affected both players as shown by the total cost. This is because the pursuer is now more interested in the intermediate states yet did not properly assume how much control the evader was willing to use. Therefore, the evader was able to maximize the performance index more so than in the complete information game while the pursuer was unable to minimize the desired performance index as well. Furthermore, the pursuer is unable to close in on the evader as it was able to in the complete information game. These results indicate how poor assumptions related to an opponent's strategy can decrease the performance of a player.

Finally, the effects of an incomplete, imperfect, uncertain information game are shown in Figs. I.8 - I.10. In addition to the gain errors in the previous example, the pursuer was also subject to modeling uncertainties. The relative dynamics are shown in Eqn. 1.12. Here, modeling uncertainty is defined as errors in the model matrix A. Therefore, the non-zero elements of the model matrix A assumed by the pursuer were modified such that

$$A = \begin{bmatrix} 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.9 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{1.16}$$

The true dynamic model assumed by the evader remained constant throughout all simulations.

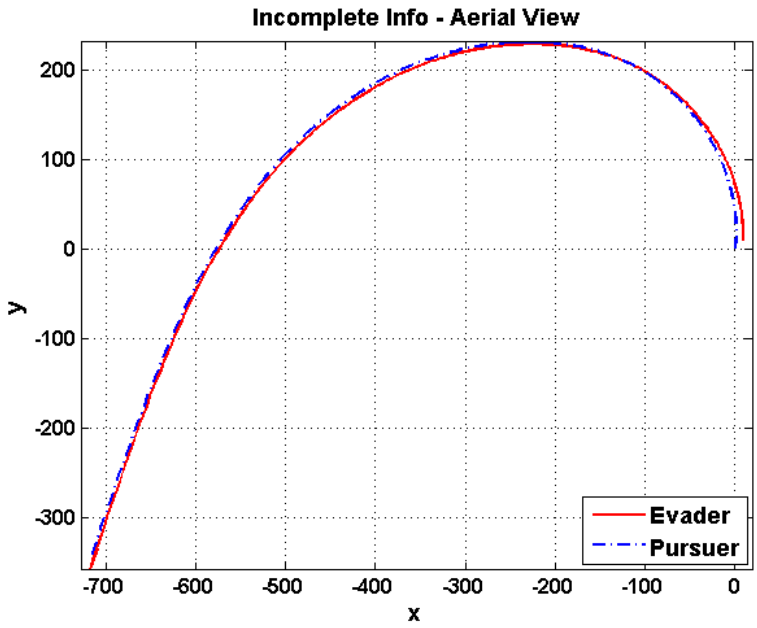

Figure I.5. Final-Time-Fixed, Incomplete Information Aerial View

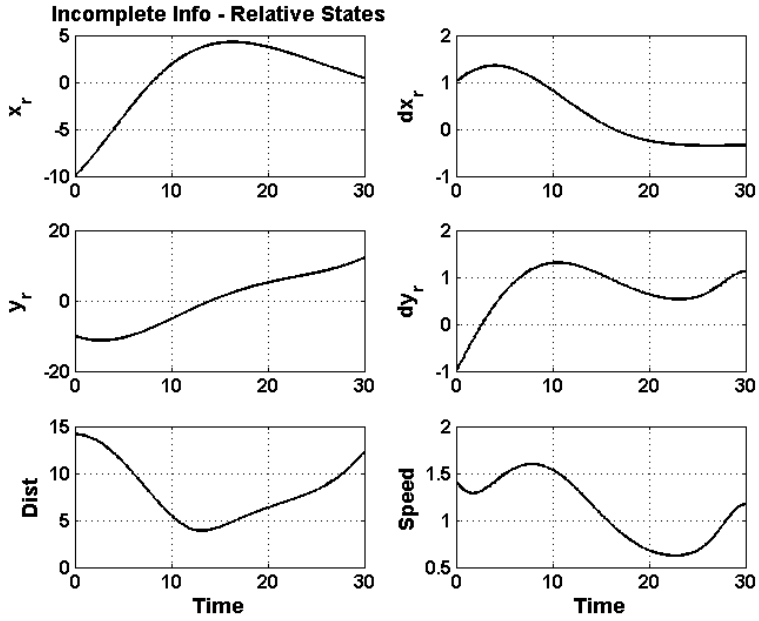

Figure I.6. Final-Time-Fixed, Incomplete Information Relative States

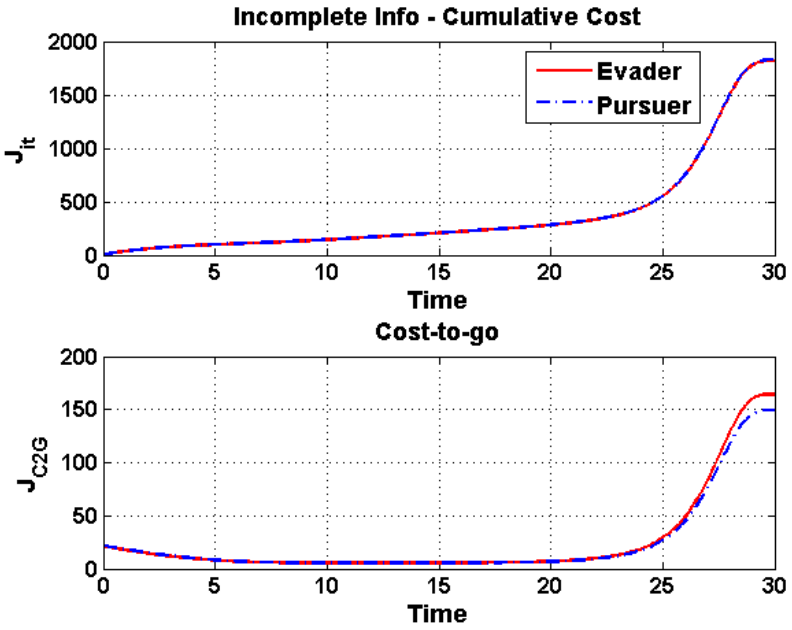

Figure I.7. Final-Time-Fixed, Incomplete Information Cost Analysis

Once the uncertain information is introduced to the game, the pursuer's performance greatly diminishes and the evader is able to completely evade capture as illustrated in Figs. I.8 and I.9. The cost analysis for the uncertain information game is shown in Fig. I.10. Total cost for the pursuer was 8.4537×10^4 while the evader cost was 8.6271×10^4. Again, the total cost for the pursuer increases more than an order of magnitude once uncertain information is introduced. As desired, the evader was able to obtain a larger total cost.

A summary of the cumulative cost experienced by the pursuer for all three final-time-fixed simulations is found in Fig. I.11. It becomes very clear how much performance can be lost as information pertaining to the game becomes less available. The goal of the work presented in the following dissertation aims to increase the performance of a player when faced with an incomplete information game. Behavior

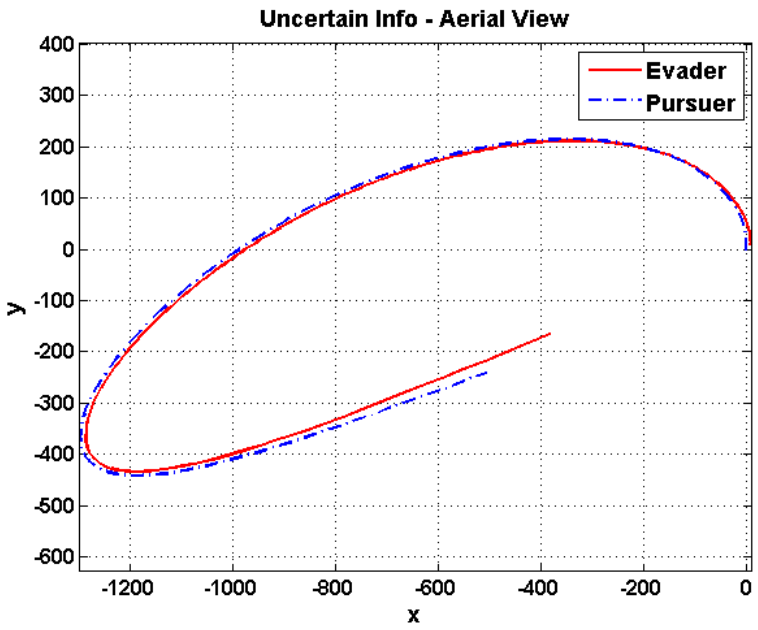

Figure I.8. Final-Time-Fixed, Uncertain Information Aerial View

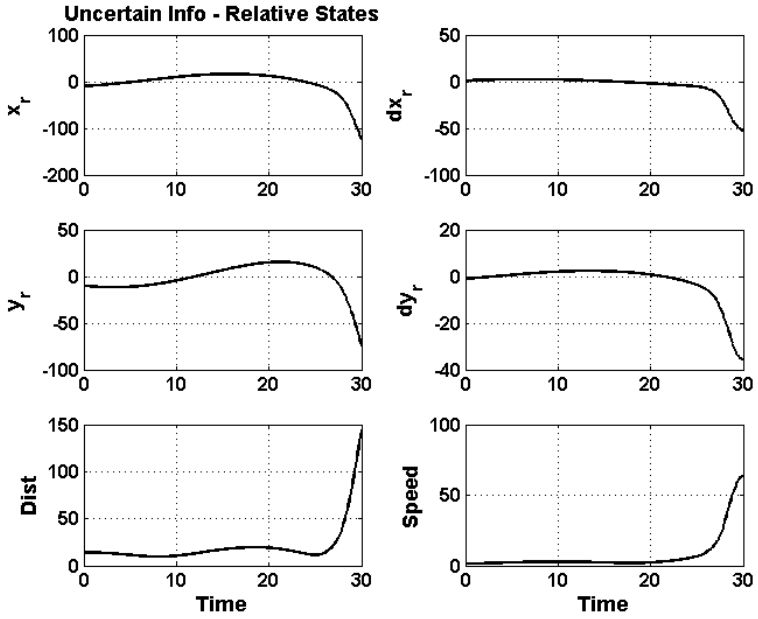

Figure I.9. Final-Time-Fixed, Uncertain Information Relative States

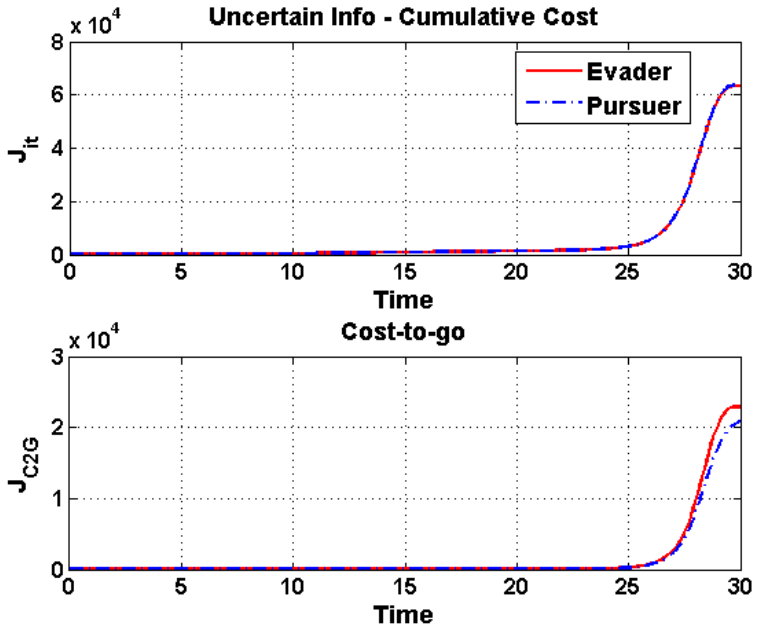

Figure I.10. Final-Time-Fixed, Uncertain Information Cost Analysis

learning techniques will be identified for different types of games defined by their final-time: final-time-fixed, infinite-horizon, and final-time-free. Once a behavior learning solution has been implemented and a player can predict their opponent's behavior for all time, it is then possible for a player to turn the pursuit-evasion problem into a one-sided optimal control problem with a time-varying model matrix A. It will be shown that when a player is subject to incomplete, imperfect, and uncertain information, these methods can be used to allow the player to gain a tactical advantage.

Final-time-fixed games are those in which the game occurs for a predetermined amount of time. Games of these nature are best suited for those where a pursuer must catch an evader in a fixed amount of time or the evader must not get caught within a fixed amount of time. In the aerospace realm, final-fixed-time games are

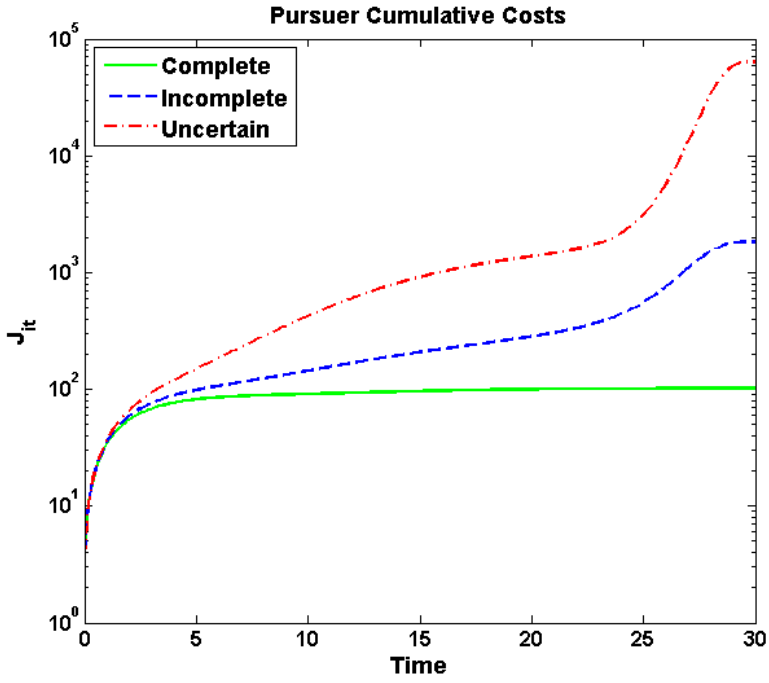

Figure I.11. Final-Time-Fixed, Cumulative Cost Comparison

best used to define missile interception problems. Often, an interceptor has a small window in which it must intercept the target of interest before a specific destination or altitude is reached.

Infinite-horizon games also occur for a predetermined amount of time. That time, however, is specified as infinity. That is, the pursuit evasion game goes on indefinitely. Spacecraft proximity operations type problems where one vehicle may need to get into position to inspect another takes on this form. Another example includes tracking of a ground vehicle by an aerial vehicle.

Final-time-free games do have have a specified amount of time to be played. In these games, time is a variable which can be minimized (maximized) by the pursuer (evader) in an effort to obtain the best performance. In some cases, if the relative

states or player control inputs are being considered, these games can be reduced to simply a minimum time problem with a final-state-fixed constraint. Minimum time problems can also be applied to missile interception in addition to orbit transfer games for spacecraft rendezvous.

I.B. Literature Review

Differential games were first introduced by Issacs in 1954 [9–12] while at the RAND Corporation. Over the following decade, DGs received a considerable amount of interest due to the emergence of a new optimal control topic, but also because of their obvious applications to warfare strategies including pursuit-evasion and the popular proportional navigation law for interception [13, 14]. It was not until the publication of Issacs book in 1964 that the true hurdles of DG theory application became clear [1]. The largest obstacle involved the very different perspectives that game theorists and control theorists approached the problem with. Issacs stressed the need for non-traditional feedback solutions [15] and the true limitations to the theory when an incomplete information scenario is introduced [1]. Feedback solutions play a critical role in pursuit-evasion because the decisions made by an opponent must be taken into account when a player desires a truly optimal solution. The inherent issue with a feedback scheme is the possible lack of information related to the player, the opponent, and the relative system.

Initially, zero-sum games in which a single performance index is maximized by one agent and minimized by another, were studied in depth [16]. Following the encouragement of Issacs towards a more realistic theory, Starr and Ho introduced

the nonzero-sum game which takes advantage of the Nash equilibrium solution [17]. Issacs ideas involving information availability were reverberated by Ho in his 1970 survey of DGs where he placed emphasis on information sets and presented a generalized framework to help mesh the different points of view from game theorists and control theorists [18]. By 1980, the discrepancies between theory and application were apparent as Shinar managed to find a way to reliably apply PE theory to air-to-air combat scenarios [19].

Work towards a theory with non-ideal state information was initiated by Rhodes in 1968 when he proposed the use of a separation theorem to deal with measurement noise in linear-quadratic games. Others built on his idea of stochastic games and laid a framework for variable information sets which are defined based on ideal, noisy, or no state measurements [20, 21]. More recently, a push towards multi-player teams in PE scenarios has been of particular interest with the emergence of unmanned air vehicles (UAVs) and their military applications. Search and state estimation strategies for teams of players have been presented by Li and Antoniades, respectively [22, 23].

Still, the little work that been done in the area of incomplete information PE games has focused on how to handle measurement noise and not on the actual estimation of opponent strategy. The community has focused on imperfect information games. Most recently, Satak has developed methods for behavior learning in DGs by separating the estimation issue from the game theory and applying Gaussian-least-squares-differential-correction (GLSDC) to opponent decision data to estimate the assumed strategy of an opponent [2, 8]. As we work towards the most realistic

PE scenarios, strategy estimation techniques are needed which act on relative state measurements and not the control input of an opponent because that decision data is most likely unavailable in the true incomplete information case. Additionally, methods to deal with uncertainties in the dynamic model are also of interest because of the possible lack of intelligence regarding an opponent, which is prevalent in military applications.

I.C. Outline

The following pages present the development and implementation of a behavior learning framework for linear and nonlinear PE systems which are subjected to incomplete, imperfect, and uncertain information scenarios. Behavior learning techniques for final-time-fixed, infinite-horizon, and final-time-free cases will be examined and compared to baseline results generated from the corresponding complete information case. Although these methods can be extended to team-based PE games, the scope of this work is limited to one-pursuer, one-evader scenarios and the terms "pursuer" and "Player p" will be used interchangeably along with the terms "evader" and "Player e". Throughout the chapters, special focus will be given to the performance of the pursuer who will be enabled with behavior learning while the evader will always assume a zero-sum conservative strategy.

Chapter II presents behavior learning for the final-time-fixed case which is the most traditional type of pursuit-evasion game found in the literature. The zero-sum solution to the PE game is derived and the incomplete information behavior learning method is introduced. An extension to this behavior learning method to

include the uncertain information case is also derived. It is then shown how a player can augment their strategy once a behavior learning solution has been obtained, such that the player enabled with behavior learning can simplify their solution into that of a one-sided optimal control problem. An extended Kalman filter version of the final-time-fixed behavior learning technique is developed and implemented for the motivational example provided in Section I.A. The results for the behavior learning filter applied to the incomplete and uncertain information cases is compared to the complete information case and the incomplete and uncertain information cases without behavior learning.

Behavior learning for infinite-horizon PE is the focus of Ch. III. An infinite-horizon zero-sum solution is derived and behavior learning methods for incomplete and uncertain information scenarios are introduced. Strategy augmentation is provided which a player implements after converging on a behavior learning solution. The extended Kalman filter is used to implement the infinite-horizon behavior learning method and apply it to the motivational example found in Section I.A. The results for the behavior learning filter applied to the incomplete and uncertain information cases is compared to the complete information case and the incomplete and uncertain information cases without behavior learning.

Chapter IV breaks down how behavior learning techniques can be applied to final-time-free PE games. Specifically, behavior learning for minimum-time PE is examined with a focus on state space trajectories. A minimum-time pursuit-evasion example is developed in Ch. V. The feedback solution is derived and behavior learning is applied to this scenario. A comparison of the assumed and true state

space trajectories are provided along with the results that arise from implementing a minimum-time behavior learning filter.

In an effort to allow nonlinear systems to fit within the linear-quadratic behavior learning framework, a two-step dynamic inversion process is presented in Chapter VI. Dynamic inversion allows a system with nonlinear kinematics and dynamics to take on the response specified by the user. Of course, caution must be exercised when forcing a nonlinear system to behave as a desired linear system.

Two key applications are given in Chapters VII and VIII. Chapter VII contains a spacecraft attitude reorientation PE scenario which takes on the form of an infinite-horizon game. The nonlinear model for each spacecraft is developed and dynamic inversion is applied to test the robustness of behavior learning when key assumptions made about the model are false. Results for the incomplete information behavior learning algorithm are compared against complete and incomplete information scenarios.

Chapter VIII contains a military application of behavior learning as the missile interception problem is studied. Following the model derivation, the realistic space made up of all states defining a single vehicle is transformed to a reduced space and a proportional-derivative (PD) controller is introduced to allow for proper control of the missile. The effectiveness of behavior learning is tested with the presence of dynamic inversion and the control manipulation. Results for complete, incomplete, and incomplete information with behavior learning are compared.

Finally, Chapter IX contains a summary of the results and conclusions regarding the development and implementation of behavior learning. Limitations and exten-

sions to the presented behavior learning framework are given.

CHAPTER II

FINAL-TIME-FIXED BEHAVIOR LEARNING

In pursuit-evasion games, it is natural for a player to assume a zero-sum safe strategy if their opponent's behavior is unknown. As shown in Section I.A, an incorrect zero-sum strategy assumption can be devastating to a player's performance, especially in the presence of imperfect and uncertain information scenarios. Fortunately, behavior learning techniques can be used to estimate the strategy of an opponent. Once an opponent's strategy is known, it is possible for a player to augment their performance index as necessary to account for the modeled opponent behavior which includes modifying the pursuit-evasion game into a one-sided optimal control problem.

Behavior learning techniques for final-time-fixed pursuit-evasion games were introduced by Satak [2]. These methods are based on the batch estimation technique GLSDC. The major drawback of these techniques are the assumption of a perfect information game in which the control input of the opponent is readily available for processing. As the discussion of pursuit-evasion moves further from the ideal scenario and approaches most realistic cases, it becomes apparent that the control input of an opponent will be unavailable in the most realistic pursuit-evasion games. In an effort to allow implementation of behavior learning in the most realistic pursuit-evasion scenarios, it is critical for the methods to be applicable to incomplete, imperfect, and uncertain information games.

This chapter identifies the form of behavior learning needed for final-time-fixed

pursuit-evasion games with varying levels of information availability. Once the form of behavior learning is identified, it is possible to utilize any one of several estimation techniques to estimate the parameters which capture the behavior of a given opponent. The methods presented here assume an incomplete and imperfect information scenario where the only measurements available to the player are the relative states associated with the pursuit-evasion model. Relative state measurements are subject to a zero-mean Gaussian noise distribution. It will also be shown that under certain conditions, these behavior learning methods can be extended to the incomplete, imperfect, and uncertain information scenarios.

The methods outlined in this chapter will follow many of the same game assumptions previously discussed. For consistency, we will always consider the pursuer to be enabled with behavior learning. The evader will continue to play using the same type of behavior used in the previous examples. Specifically, the evader will be subject to an incomplete information game and will assume a zero-sum safe strategy for the duration of the game. The evader will not be using any type of behavior learning and will always be subject to an imperfect and certain information game. That is, the evader's relative state measurements will be subject to a zero-mean Gaussian noise distribution and the evader's relative dynamic model will contain no inaccuracies. Of course, it is possible for either or both players to be enabled with behavior learning and be subject to varying levels of information availability. The assumptions made here are done in an effort to limit the scope of the discussion to study how well the developed techniques perform for a single intelligent player with behavior learning enabled against an opponent that assumes a safe strategy with

reliable modeling information possibly gathered from a reconnaissance mission.

When considering the perspective of the pursuer, we wish to enable the pursuer with a means to improve its total cost for the PE game. When faced with an incomplete information game, it will not be possible to achieve the same performance of an complete information game even with behavior learning enabled because the complete information game represents the ideal case. However, if the overall performance of the pursuer can be improved by a quantifiable means for a game with incomplete information, then the behavior learning method has been successful. Behavior learning aims to estimate the the strategy of the PE opponent using an assumed objective function. The strategy is defined as the gain selection the opponent uses to compute its control.

Pursuit-evasion games can be defined by an infinite number of cost function and dynamic system combinations. The focus of this discussion will be limited to continuous-time, linear-quadratic, PE games. These types of games are defined by quadratic cost functions subject to a continuous, linear, dynamic constraint. Optimal solutions to zero-sum games will be reviewed before identifying where player strategy manifests within the solution. It is possible to add additional constraints to these PE games. Final-state constraints can be used to require interception or rendezvous. Control constraints can also be implemented. Unfortunately, these additional constraints tend to rely on iterative solution methods such as multiple shooting which do not use feedback. As mentioned in Chapter I, feedback solutions are absolutely critical for pursuit-evasion applications and their importance will continue to play a primary role in this framework.

II.A. Pursuit-Evasion

In a traditional two-player PE scenario involving no external objectives, the pursuer is interested in driving some or all of the relative states between the two players, $\boldsymbol{z}$, to zero while conserving enough control input, $\boldsymbol{u}_p$, to do so. For an intercept problem, the pursuer is most concerned about the relative position. The relative velocity at intercept is not of interest because an intercept generally means destruction of both vehicles. In a rendezvous problem, the pursuer may want to drive the relative position and velocity to zero so neither vehicle is damaged if rendezvous occurs.

Simultaneously, the evader is attempting to maximize some or all of the relative states in an effort to prevent capture. The evader must also be cautious about the amount of control, $\boldsymbol{u}_e$, being spent. These characteristics allows the game to be formulated as an optimal control problem with the dynamic constraint being modified by two intelligent agents.

Zero-sum pursuit-evasion games are those in which a single cost function is used to define the entire game. The pursuer attempts to minimize the zero-sum cost function while the evader works to maximize it. These types of games have also been referred to as minimax games [24]. A zero-sum, final-time-fixed, LQ PE game is traditionally defined by a performance index of the form

$$J_{ZS_{fix}} = \phi\left(\boldsymbol{z}_f, t_f\right) + \int_{t_0}^{t_f} L\left(\boldsymbol{z}, \boldsymbol{u}_p, \boldsymbol{u}_e, t\right) \mathrm{dt}\,, \tag{2.1}$$

$$J_{ZS_{fix}} = \frac{1}{2}\boldsymbol{z}_f^T S_f \boldsymbol{z}_f + \frac{1}{2}\int_{t_0}^{t_f} \left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p - \boldsymbol{u}_e^T R_e \boldsymbol{u}_e\right) \mathrm{dt}\,, \tag{2.2}$$

subject to the linear dynamic constraint

$$\dot{\boldsymbol{z}} = \boldsymbol{f}\left(\boldsymbol{z}, \boldsymbol{u}_p, \boldsymbol{u}_e\right) = A\boldsymbol{z} + B\boldsymbol{u}_p - B\boldsymbol{u}_e\,, \tag{2.3}$$

where $\boldsymbol{z} \in \mathbb{R}^n$ and $\boldsymbol{u}_p, \boldsymbol{u}_e \in \mathbb{R}^m$. Matrices S_f, Q, R_p, and R_e represent the gains for the zero-sum game and subscript f is used to denote the state or gain at the final time t_f. Matrices S_f and Q are symmetric and positive semidefinite, while R_p and R_e are symmetric and positive definite. Here, the relative plant in Eqn. 2.3 and the weight matrices are assumed to be time-invariant.

As discussed in Ch. I and stressed by Issacs [15], we seek a closed-loop control solution. The Hamiltonian is defined by

$$H = L + \boldsymbol{\lambda}^T \boldsymbol{f}\,, \tag{2.4}$$

$$H = \frac{1}{2}\left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p - \boldsymbol{u}_e^T R_e \boldsymbol{u}_e\right) + \boldsymbol{\lambda}^T\left(A\boldsymbol{z} + B\boldsymbol{u}_p - B\boldsymbol{u}_e\right)\,. \tag{2.5}$$

Control solutions are given by the stationarity conditions

$$\frac{\partial H}{\partial \boldsymbol{u}_p} = 0 \qquad \rightarrow \qquad \boldsymbol{u}_p = -R_p^{-1} B^T \boldsymbol{\lambda}\,, \tag{2.6}$$

$$\frac{\partial H}{\partial \boldsymbol{u}_e} = 0 \qquad \rightarrow \qquad \boldsymbol{u}_e = -R_e^{-1} B^T \boldsymbol{\lambda}\,. \tag{2.7}$$

When Eqns. 2.6 and 2.7 are substituted into the state equation given by Eqn. 2.3, this yields

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} - BR_p^{-1}B^T\boldsymbol{\lambda} + BR_e^{-1}B^T\boldsymbol{\lambda}\,. \tag{2.8}$$

The costate equation is given by

$$\frac{\partial H}{\partial \boldsymbol{z}} = -\dot{\boldsymbol{\lambda}}^T \quad \rightarrow \quad \dot{\boldsymbol{\lambda}} = -Q\boldsymbol{z} - A^T\boldsymbol{\lambda}\,, \tag{2.9}$$

and optimal control theory [4], the terminal condition is given by

$$\boldsymbol{\lambda}(t_f) = \frac{\partial \phi}{\partial \boldsymbol{z}}_{t_f} = S_f \boldsymbol{z}_f\,. \tag{2.10}$$

The initial condition is given by initial states $\boldsymbol{z}_0$ and the two-point boundary-value problem can be solved using the sweep method [3]. Assuming the state and costate satisfy a linear relation that takes on the form of Eqn. 2.10 for all time $t \in [t_0, t_f]$, then

$$\boldsymbol{\lambda} = S\boldsymbol{z}\,. \tag{2.11}$$

If such an S can be found, then the assumption given in Eqn. 2.11 is valid.

The solution requires differentiating the costate.

$$\dot{\boldsymbol{\lambda}} = \dot{S}\boldsymbol{z} + S\dot{\boldsymbol{z}}\,. \tag{2.12}$$

By substituting Eqns. 2.9 and 2.8 into Eqn. 2.12 and rearranging, the differential equation for S becomes

$$\dot{S} = -Q - A^T S - SA + SBR_p^{-1}B^T S - SBR_e^{-1}B^T S\,. \tag{2.13}$$

Equation 2.13 takes on the form of a modified matrix Riccati equation used for PE. An effective gain R can be computed based on R_p and R_e to transform it into a standard matrix Riccati equation. If

$$R^{-1} = R_p^{-1} - R_e^{-1}\,, \tag{2.14}$$

then

$$\dot{S} = -Q - A^T S - SA + SBR^{-1}B^T S\,. \tag{2.15}$$

Using the backwards sweep method, S can now be solved for at every time t by starting at time t_f and integrating backwards through time using S_f as the terminal condition. The feedback control solutions become

$$\boldsymbol{u}_p = -K_p\boldsymbol{z}\,, \tag{2.16}$$

$$\boldsymbol{u}_e = -K_e\boldsymbol{z}\,, \tag{2.17}$$

with the time-varying Kalman gains K_p and K_e defined as

$$K_p = R_p^{-1} B^T S \,, \tag{2.18}$$

$$K_e = R_e^{-1} B^T S \,. \tag{2.19}$$

The single cost function found in Eqn. 2.2 is used for a complete information game. In an incomplete information game, a zero-sum safe strategy could be implemented by either player if they are not confident in their assumed opponent's objective and strategy. Each player would assume their own cost function of the form shown in Eqn. 2.2 which is done in an attempt to implement a conservative strategy. This strategy assumes an opponent is attempting to accomplish the exact opposite objective.

II.B. Incomplete Information Behavior Learning

Because behavior is made up of an objective and strategy, it is necessary to assume a form for the opponent's (or in this case, the evader's) objective function in order to estimate any type of strategy. If the evader is not enabled with behaving learning as well, it is reasonable to assume that the evader is implementing a zero-sum safe strategy whose solution is given by Eqn. 2.17. Before implementing a behavior learning technique, it is necessary to identify where the strategy manifests itself within the evader's optimal control solution.

The optimal control for the evader is dictated by the relative states, $\boldsymbol{z}$, and the Kalman gain which is given by Eqn. 2.19. From the solution derivation shown in Section II.A, it is clear that $K_e = K_e(B, S, R_e)$, and because the differential equation $\dot{S} = \dot{S}(A, B, S, Q, R_p, R_e)$ from Eqn. 2.13, it can be concluded that $K_e =$

$K_e(A, B, S, Q, R_p, R_e)$. In a certain information game, system matrices A and B are known exactly and the behavior learning objectives evolves into the estimation of $K_e = K_e(S, Q, R_p, R_e)$. Because the gains are relative and the optimal solution is indifferent to scaled gain selections, it is possible to consider the evader's selection of gains Q, R_e, and S relative to $R_p = \mathbf{1}$, where $\mathbf{1}$ is an identity matrix of the appropriate size. It follows that for an incomplete and imperfect information game where the evader assumes a zero-sum safe strategy, the entire strategy assumed by the evader is captured in the Kalman gain K_e and the objective of behavior learning has evolved into estimating $K_e = K_e(S, Q, R_e)$.

One issue that must still be addressed involves the fact that for the final-time-fixed case, the Kalman gain K_e is time-varying because of the influence of Eqn. 2.13. Depending on the nature of the game, reasonable assumptions can be made about the form of the gains S_f, Q, and R_e. For example, it is already known that these gain matrices are symmetric, which decreases the number of independent elements to be estimated. Additionally, these gain matrices are most often diagonal to reduce unwanted cross-coupling effects. If system matrices A and B are known perfectly, then their sparceness can also help determine which elements of the optimal gains influence K_e.

The final hurdle is the relationship between the estimated S and the chosen S_f of the evader. It should be noted what the pursuer must do after solving the behavior learning problem. It will be shown that if K_e can be computed for each instance in time, then the pursuer can augment their cost function such that the two-sided pursuit-evasion problem becomes a one-sided optimal control problem. The gain S_f

simply provides the terminal condition for the two-point boundary value problem and allows for calculation of S before the game starts. If the gains S, Q, and R_e can be estimated with $R_p = \mathbf{1}$, then S can be computed at each instance in time using Eqn. 2.13. The chosen S_f simply allows for the calculation of S by specifying S at t_f. Therefore, if S can be determined through another means at a given t, then its solution at each instance in time can be determined.

Recall from Eqn. 2.36 that after behavior learning occurs, the one-sided OCP that the pursuer switches to requires the computation of the gain K_e in a backwards sweep fashion. Thus, S must be known at every instance in time prior to implementing the modified K_p based on the one-sided OCP. This can be accomplished by using the estimated evader Q and R_e and propagating S forward in time. This means that the evader's entire strategy can be defined by the estimated values Q, R_e, and S along with Eqn. 2.15 when it is assumed that $R_p = \mathbf{1}$.

Sequential estimation techniques require differential state equations for the states being estimated. In an imperfect information game, the relative states of interest defined by $\boldsymbol{z}$ are measured through some means whether directly or by measuring the inertial movements of both players then computing the necessary relative states. Therefore, it is assumed the relative states are subject to a zero-mean Gaussian distribution and the estimation of $\boldsymbol{z}$ is also of interest.

It has been shown here that the entire strategy assumed by the evader is captured in the time-varying Kalman gain K_e. To implement the most efficient behavior learning techniques, it is essential to apply all knowledge about the system. Therefore, the details of K_e must be investigated. Consider the example with differentially

driven vehicles in the horizontal plane presented in Section I.A. The relative system is given by Eqn. 1.12.

If the evader is interested in the relative position only, weighs the x- and y-components equally, and also weighs the control inputs for x and y equally, then the gains will take on the form

$$S_f = \begin{bmatrix} s_f & 0 & 0 & 0 \\ 0 & s_f & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} q & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_e = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}. \tag{2.20}$$

With these selections and the help of Eqn. 2.13, the instantaneous S takes on the form

$$S = \begin{bmatrix} s_1 & 0 & s_3 & 0 \\ 0 & s_1 & 0 & s_3 \\ s_3 & 0 & s_2 & 0 \\ 0 & s_3 & 0 & s_2 \end{bmatrix}. \tag{2.21}$$

We now see for this particular problem, K_e as a function of S and R_e becomes

$$K_e = \begin{bmatrix} \frac{s_3}{r} & 0 & \frac{s_2}{r} & 0 \\ 0 & \frac{s_3}{r} & 0 & \frac{s_2}{r} \end{bmatrix}. \tag{2.22}$$

The way in which player strategy manifests within the control computation for the final-time-fixed scenario has been identified. Next, the state equations for the parameters must be identified to implement any of several estimation tools to carry out the parameter estimation. The objective is to estimate the relative states, $\boldsymbol{z}$, and the independent elements of S, Q, and R_e the evader has assumed using the

measured relative states and the known control input of the pursuer, $\boldsymbol{u}_p$.

The states to estimate are given by the vector

$$\boldsymbol{x} = [z_1,\, z_2,\, z_3,\, z_4,\, s_1,\, s_2,\, s_3,\, q,\, r]^T \,, \tag{2.23}$$

$$\boldsymbol{x} = [x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6,\, x_7,\, x_8,\, x_9]^T \,, \tag{2.24}$$

where states $x_1 - x_7$ are time-varying. Because these states are time-varying, it is necessary to implement a sequential estimator to identify these states. The state equations for $x_1 - x_4$ are given by Eqn. 2.3 and the state equations for $x_5 - x_7$ are given by the corresponding scalar elements of Eqn. 2.13.

At first glance, Eqn. 2.3 appears to be linear. However, after substitution of Eqns. 2.17 and 2.19, it becomes nonlinear in the states.

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u_p} + BR_e^{-1}B^T S\boldsymbol{z} \,. \tag{2.25}$$

Therefore, a nonlinear estimator is required. The states x_8 and x_9 are constant so the corresponding state equations are simply zero.

For this particular example, with $\boldsymbol{u}_p = [u_{p_1}, u_{p_2}]$, the corresponding state equations are summarized as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) \,, \tag{2.26}$$

where

$$\boldsymbol{f} = \begin{bmatrix} x_3 \\ x_4 \\ \frac{x_1 x_7}{x_9} + \frac{x_3 x_6}{x_9} + u_{p_1} \\ \frac{x_2 x_7}{x_9} + \frac{x_4 x_6}{x_9} + u_{p_2} \\ x_7^2 - \frac{x_7^2}{x_9} - x_8 \\ x_6^2 - \frac{x_6^2}{x_9} - 2x_7 \\ x_6 x_7 - x_5 - \frac{x_6 x_7}{x_9} \\ 0 \\ 0 \end{bmatrix} . \tag{2.27}$$

The measurements available for the behavior learning filter are

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) = \left[x_1\,, x_2\,, x_3\,, x_4\right]^T . \tag{2.28}$$

Equations 2.26 and 2.28 are in the standard form needed for a nonlinear filter such as the extended Kalman or unscented filters. Once a nonlinear estimator is employed and has converged on a solution for the estimates given by Eqn. 2.24, the pursuer can then compute the evader's Kalman gain at each instance in time using Eqns. 2.22 and 2.13. As the complexity of the relative dynamic system increases, so does the chance for observability issues to occur. Nonlinear observability can be computed using the Lie derivative [25]. If a linearized filter is implemented such as an extended Kalman filter, observability can be computed by checking the rank of the observability matrix O [25]. For this particular example, the system is observable. Observability must be treated on a case-by-case basis because each system model and associated strategy assumptions are different.

II.C. Uncertain Information Behavior Learning

The behavior learning framework presented in the previous section may be extended to the uncertain information case. In an uncertain information game, a player is subject to modeling errors present in the relative dynamic model. In Section I.A, this was defined to be errors in the model matrix A. The independent elements of A can be added to the state estimate vector described by Eqn. 2.27.

For the ongoing example, the true model matrix is given by

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \tag{2.29}$$

It is important that all available information about the system is applied. For this specific example, it will be assumed that A contains two independent elements defined by

$$A = \begin{bmatrix} 0 & 0 & a_1 & 0 \\ 0 & 0 & 0 & a_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \tag{2.30}$$

For the time-invariant case, the state equations describing these new parameters are simply $\dot{a}_1 = \dot{a}_2 = 0$. The behavior learning state estimate vector is then augmented such that

$$\boldsymbol{x} = [z_1,\, z_2,\, z_3,\, z_4,\, s_1,\, s_2,\, s_3,\, q,\, r,\, a_1,\, a_2]^T , \tag{2.31}$$

$$\boldsymbol{x} = [x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6,\, x_7,\, x_8,\, x_9,\, x_{10},\, x_{11}]^T , \tag{2.32}$$

and the state equation vector becomes

$$\boldsymbol{f} = \begin{bmatrix} x_3 x_{10} \\ x_4 x_{11} \\ \frac{x_1 x_7}{x_9} + \frac{x_3 x_6}{x_9} + u_{p_1} \\ \frac{x_2 x_7}{x_9} + \frac{x_4 x_6}{x_9} + u_{p_2} \\ x_7^2 - \frac{x_7^2}{x_9} - x_8 \\ x_6^2 - \frac{x_6^2}{x_9} - 2x_7 x_{10} \\ x_6 x_7 - x_5 x_{10} - \frac{x_6 x_7}{x_9} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} . \tag{2.33}$$

No additional measurements are available, therefore the form of Eqn. 2.28 remains the same. It is obvious that as model parameters are added to the state estimate vector, the possibly for the system to be unobservable become greater. For this particular example, the observability matrix O is of full rank. This form of the behavior learning algorithm will be implemented by the pursuer for the uncertain information case.

II.D. Augmented Strategy

If an agent is enabled with behavior learning, it is possible for that agent to turn the pursuit-evasion game into a one-sided optimal control problem. The goal of behavior learning is to give an agent a tactical advantage in an incomplete infor-

mation game. The objective is to estimate the opponent's strategy then use that information to play more effectively than simply assuming a zero-sum strategy. By taking on the pursuer's perspective, it becomes apparent that the pursuer is interested in estimating the evader's Kalman gain, K_e. Equation 2.19 reveals that for the final-time-fixed case, K_e is time-varying because of the influence of the solution of S. If the pursuer can properly estimate K_e such that it can be computed for any time t, then the new one-sided optimal control problem for the pursuer becomes

$$J_{P_{fix}} = \min \frac{1}{2}\boldsymbol{z}_f^T S_f \boldsymbol{z}_f + \frac{1}{2}\int_{t_0}^{t_f} \left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p\right) \mathrm{dt}\,, \tag{2.34}$$

subject to the modified system

$$\dot{\boldsymbol{z}} = \left(A + BK_e\right)\boldsymbol{z} + B\boldsymbol{u}_p\,. \tag{2.35}$$

Following the same solution derivation process for the zero-sum strategy, the Hamiltonian is formed and the stationarity condition coupled with the costate equation solution yields the same feedback control law given by Eqns. 2.16 and 2.18. Here, S is found using

$$\dot{S} = -Q - \left(A + BK_e\right)^T S - S\left(A + BK_e\right) + SBR_p^{-1}B^T S\,. \tag{2.36}$$

Note the S, Q, and R_p here are the pursuer's assumed gains and different from those which are estimated and those which are assumed by the evader. The estimated gains are used in the computation of K_e.

II.E. Extended Kalman Filter Implementation

One of the most pivotal contributions to optimal estimation came from Kalman in 1960 with the introduction of the Kalman filter for estimation of linear systems [26].

Because aerospace systems are most commonly nonlinear, extensions to his work were soon introduced [27, 28]. The nonlinear extension of the Kalman filter came to be known as the extended Kalman filter (EKF) and has been the de facto estimation technique for nonlinear systems, especially for navigation applications. Because of the nonlinearities in the the behavior learning problem shown in Eqn. 2.15, the EKF lends itself nicely to behavior learning applications.

By blindly applying an extended Kalman filter to the imperfect information motivational example problem in an attempt to estimate the relative states and gain matrices Q, R_e, and S, one would find themselves with 40 states to estimate. This is taking R_p as identity and scaling all gains with respect to R_p. By applying system knowledge and choosing the independent elements of these matrices to estimate, it is possible to reduce the number of state estimates to 9. Observability can become an issue if the system becomes too complex, the number of independent elements cannot be reduced, or if relative state measurements are unavailable. Observability should be treated on a case-by-case basis and can be computed for nonlinear systems using the Lie derivative [25]. However, due to the structure of the extended Kalman filter and how it is linearized using a first-order Taylor series expansion about the estimated state, linear observability methods can be applied as well.

A continuous-discrete extended Kalman filter takes on the form [29]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + G(t)\boldsymbol{w}(t), \quad \boldsymbol{w}(t) \sim N(\boldsymbol{0}, Q(t)), \tag{2.37}$$

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim N(\boldsymbol{0}, R_k), \tag{2.38}$$

where the states to be estimated are designated by the vector $\boldsymbol{x}$ and the discrete measurements are designated by the vector $\tilde{\boldsymbol{y}}_k$. The model process noise is denoted

by $\boldsymbol{w}(t)$ and $G(t)$ is the process noise distribution matrix. Measurement noise at each time-step is given by $\boldsymbol{v}_k$. Matrices $Q(t)$ and R_k make up the covariance for noise processes $\boldsymbol{w}(t)$ and $\boldsymbol{v}_k$, respectively. Note the vector function $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$ is fundamentally different from those used in the dynamic inversion process.

For the dynamic model, recall the relative system of the motivational example given by Eqn. 1.12 and consider the pursuer attempting to learn evader's behavior.

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u}_p - B\boldsymbol{u}_e \,, \tag{2.39}$$

where

$$\boldsymbol{z} = [x_r,\, y_r,\, \dot{x}_r,\, \dot{y}_r]^T \,, \qquad \boldsymbol{u}_p = [u_{p_1}, u_{p_2}]^T \,, \qquad \boldsymbol{u}_e = [u_{e_1}, u_{e_2}]^T \,, \tag{2.40}$$

and

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \,, \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \,. \tag{2.41}$$

The states x_r, y_r, $\dot{x}_r$, and $\dot{y}_r$ represent the relative position and velocity of the two players in the horizontal plane while $\boldsymbol{u}_i$ denotes the control input vector for player i. It is also assumed that all elements of $\boldsymbol{z}$ can be measured at each instance in time.

When considering the perspective of the pursuer, it is known what the computed $\boldsymbol{u}_p$ is for each instance in time. However, the evader's control input $\boldsymbol{u}_e$ is unknown. It is necessary to assume a form for the evader's control. Here, it is assumed the evader is using a zero-sum safe strategy without behavior learning. For a zero-sum

strategy game, the evader's control input takes on the form

$$\boldsymbol{u}_e = -R_e^{-1} B^T S_e \boldsymbol{z}\,, \tag{2.42}$$

where

$$\dot{S}_e = -Q_e - A^T S_e - S_e A + S_e B R_p^{-1} B^T S_e + S_e B R_e^{-1} B^T S_e\,. \tag{2.43}$$

Matrices S_e and Q_e are of size 4×4 while R_e and R_p are of size 2×2. Matrices S_e and Q_e are symmetric positive semidefinite while R_e is symmetric positive definite. Furthermore, it is a reasonable assumption that Q_e and R_e are diagonal. Using these characteristics with the known form of A and B, it is possible to estimate the unique, non-zero elements in Q_e, R_e, and S_e. Gain R_p is assumed to be identity. That is, the elements of Q_e, R_e, and S_e are normalized with respect to R_p.

Relating back to the EKF dynamic model, the state estimate $\boldsymbol{x}$ is now an 9×1 vector made up of the following elements:

$$\boldsymbol{z} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, S_e = \begin{bmatrix} x_5 & 0 & x_7 & 0 \\ 0 & x_5 & 0 & x_7 \\ x_7 & 0 & x_6 & 0 \\ 0 & x_7 & 0 & x_6 \end{bmatrix}, Q_e = \begin{bmatrix} x_8 & 0 & 0 & 0 \\ 0 & x_8 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, R_e = \begin{bmatrix} x_9 & 0 \\ 0 & x_9 \end{bmatrix}. \tag{2.44}$$

By substituting Eqn. 2.42 into Eqn. 2.39, the state equations for x_1 through x_4 are given by

$$\boldsymbol{f}_{1:4} = \dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u}_p - B R_e^{-1} B^T S_e \boldsymbol{z}\,, \tag{2.45}$$

while the state equations for x_5 through x_7 are given by the corresponding scalar equations found in Eqn. 2.43. Because the matrices Q_e and R_e are constant, the

state equations for their elements are simply $\boldsymbol{f}_{8:9} = \mathbf{0}$. This behavior learning form of the EKF is representative of a incomplete and imperfect information game.

The pursuer is equipped with the necessary sensors such that $\boldsymbol{z}$ can be measured. Mathematically, $\boldsymbol{h} = \boldsymbol{z}$. In practice, this could be done by measuring the relative states directly or by computing the pursuer's inertial states using strapdown inertial navigation then tracking the evader using a ground station with a communication link to the pursuer.

II.E.1. Incomplete Information Results

The incomplete information example shown in Section I.A was simulated again but this time with an EKF version of the behavior learning algorithm running. At $t = 5$, a new optimal control solution was computed by the pursuer using the augmented strategy defined in Section II.D. Figures II.1 - II.6 convey the results for the behavior learning case. By computing a new solution based the the behavior learning results, the pursuer was able to reduce its total cost to 1.0414×10^2 while the evader's cost was computed to be 1.0348×10^2. The pursuer's behavior learning algorithm was able to converge on estimates for the independent elements found in gain matrices Q, R_e, and S assumed by the evader as shown in Figs. II.4 and II.5. A comparison of the cost and cost-to-go for the complete, incomplete, and incomplete with behavior learning cases in Fig. II.6.

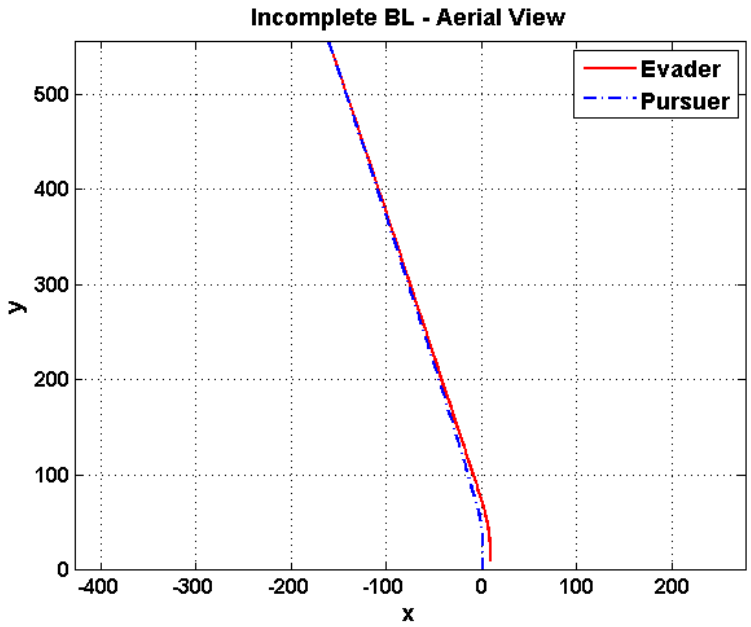

Figure II.1. Final-Time-Fixed, Incomplete Information with Behavior Learning Aerial View

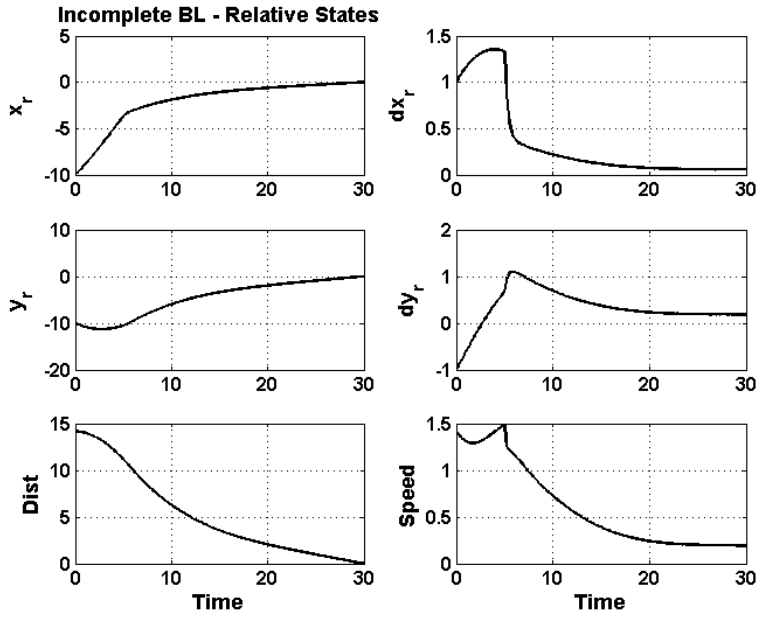

Figure II.2. Final-Time-Fixed, Incomplete Information with Behavior Learning Relative States

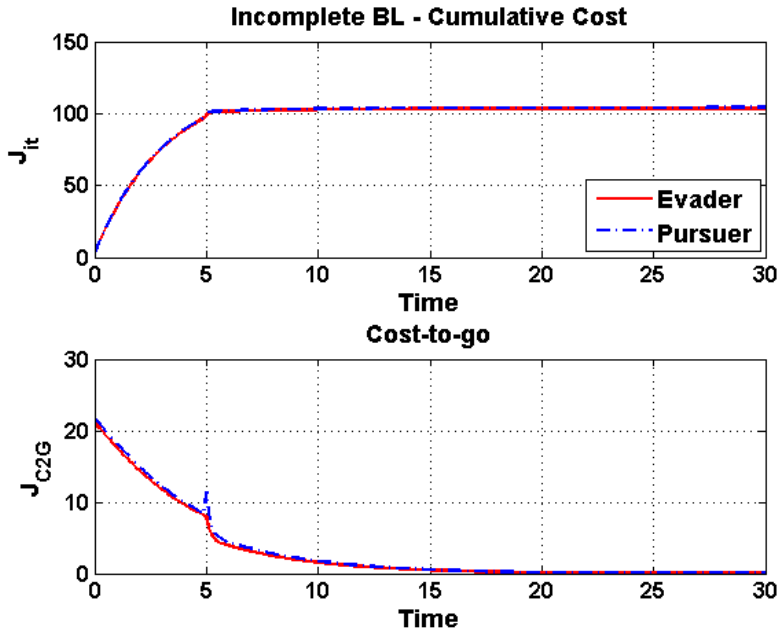

Figure II.3. Final-Time-Fixed, Incomplete Information with Behavior Learning Cost Analysis

II.E.2. Uncertain Information Behavior Learning Results

The incomplete information example shown in Section I.A was simulated again but this time with an EKF version of the behavior learning algorithm running. At $t = 5$, a new optimal control solution was computed by the pursuer using the augmented strategy defined in Section II.D. Figures II.7 - II.13 convey the results for the behavior learning case. By computing a new solution based the the behavior learning results, the pursuer was able to reduce its total cost to 1.6519×10^2 while the evader's cost was computed to be 1.6567×10^2. The pursuer's behavior learning algorithm was able to converge on estimates for the independent elements found in gain matrices Q, R_e, and S assumed by the evader as shown in Figs. II.10 and II.11. Moreover, the estimates of the independent elements found within model matrix A

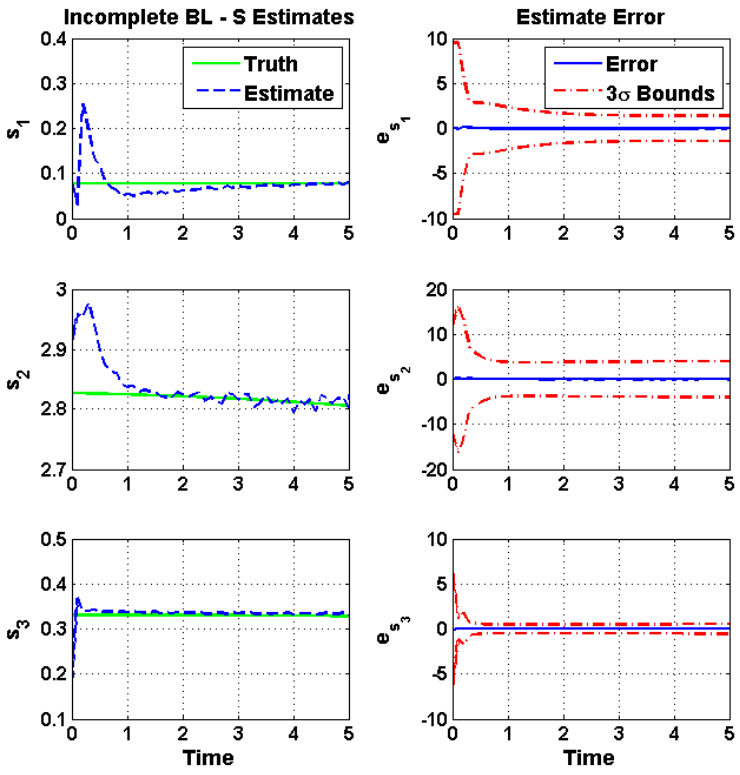

Figure II.4. Final-Time-Fixed, Incomplete Information with Behavior Learning S Estimates

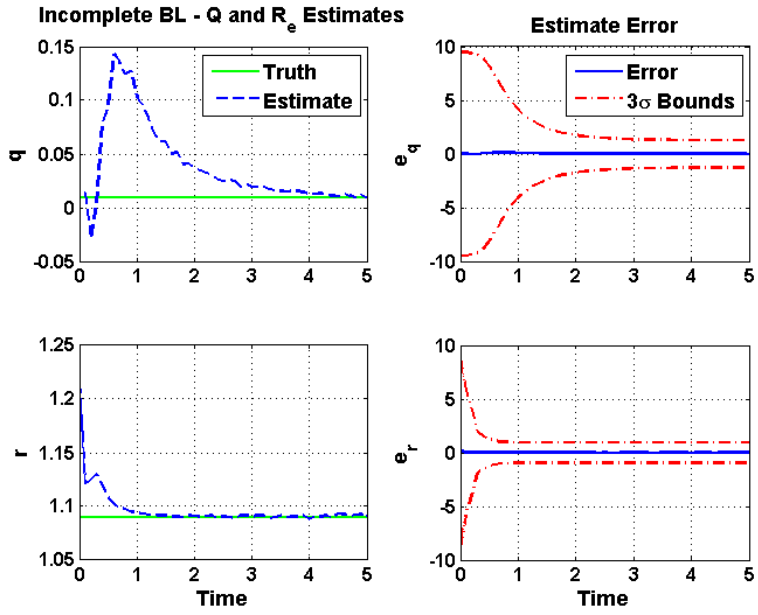

Figure II.5. Final-Time-Fixed, Incomplete Information with Behavior Learning Q and R Estimates

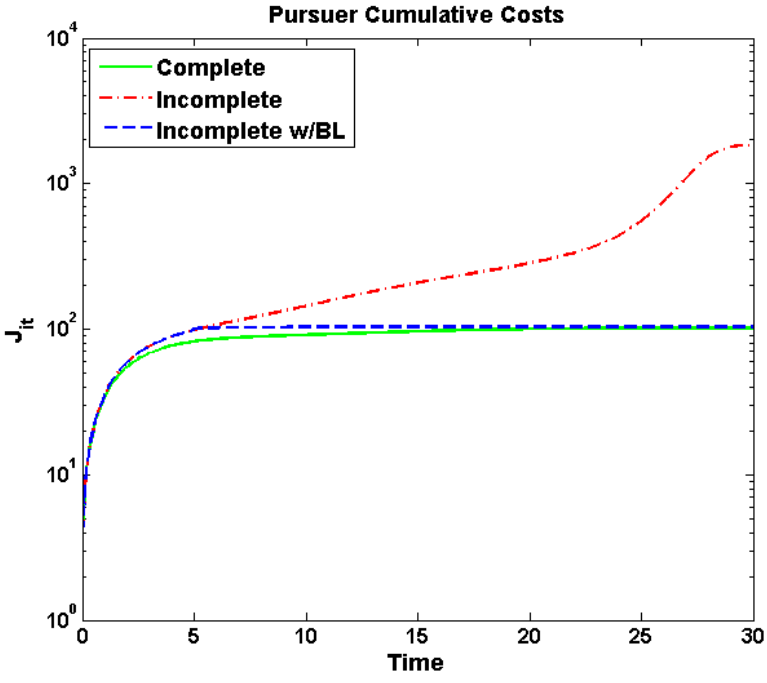

Figure II.6. Final-Time-Fixed, Incomplete Information Cumulative Cost Comparison

properly converged as evident from Fig. II.12. A comparison of the cost and cost-to-go for the complete information, incomplete information, and incomplete information with behavior learning cases in Fig. II.13.

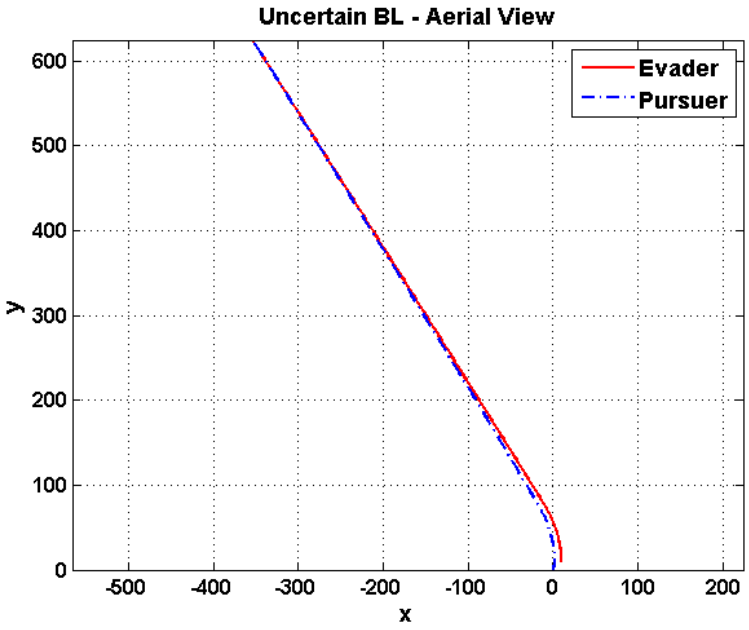

Figure II.7. Final-Time-Fixed, Uncertain Information with Behavior Learning Aerial View

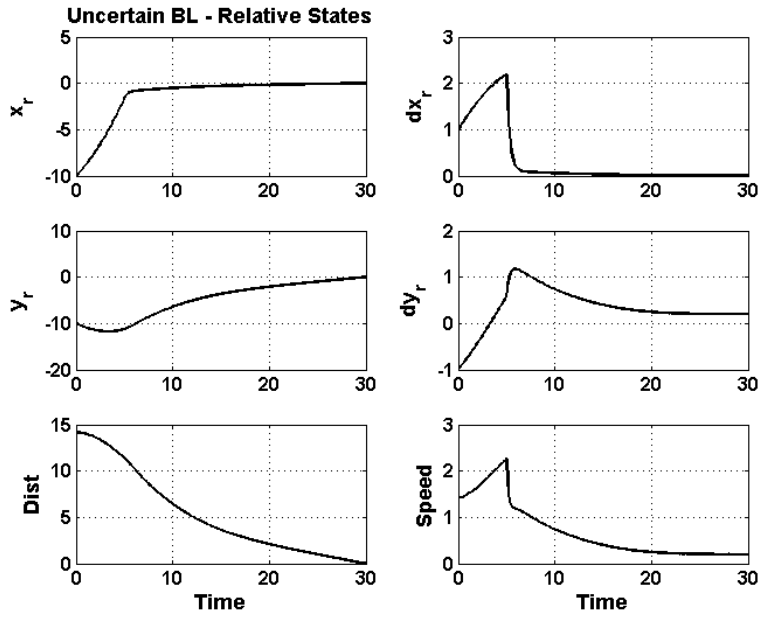

Figure II.8. Final-Time-Fixed, Uncertain Information with Behavior Learning Relative States

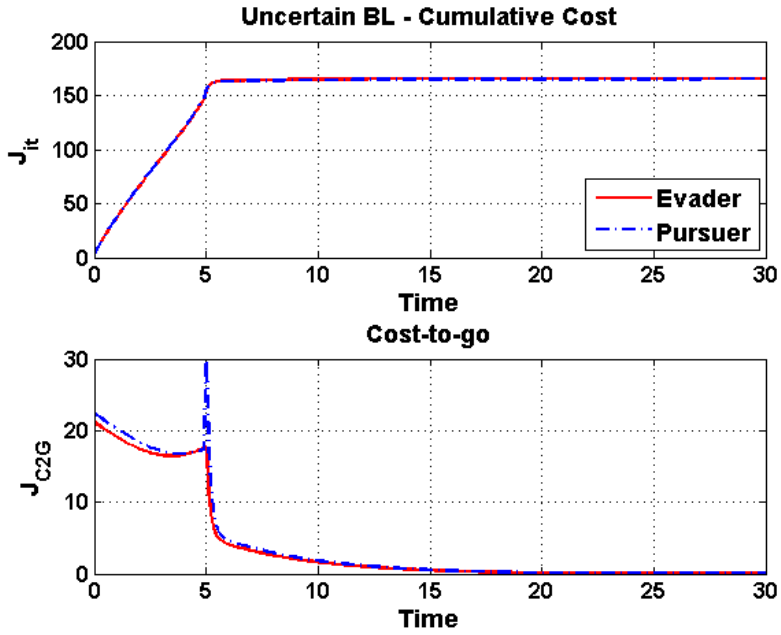

Figure II.9. Final-Time-Fixed, Uncertain Information with Behavior Learning Cost Analysis

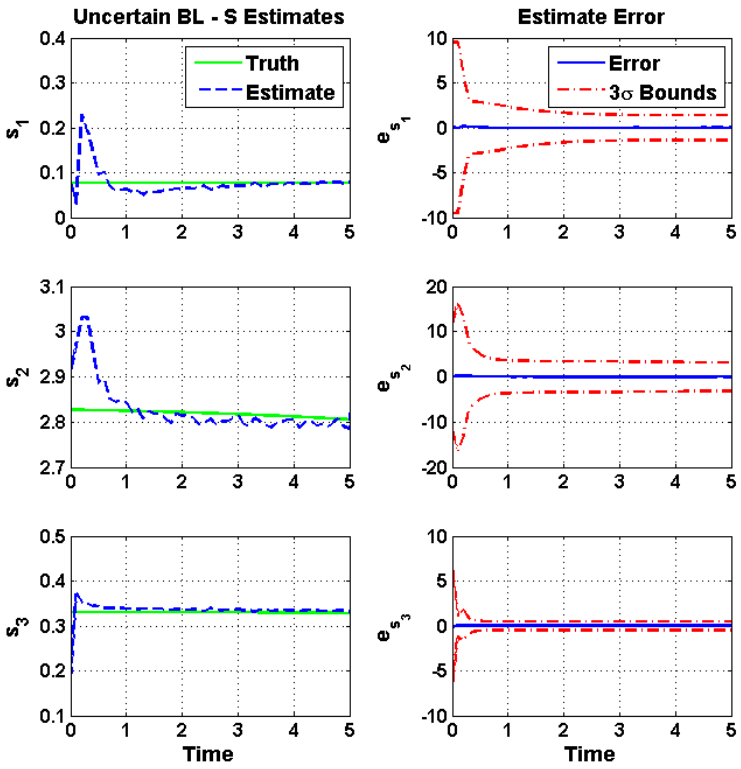

Figure II.10. Final-Time-Fixed, Uncertain Information with Behavior Learning S Estimates

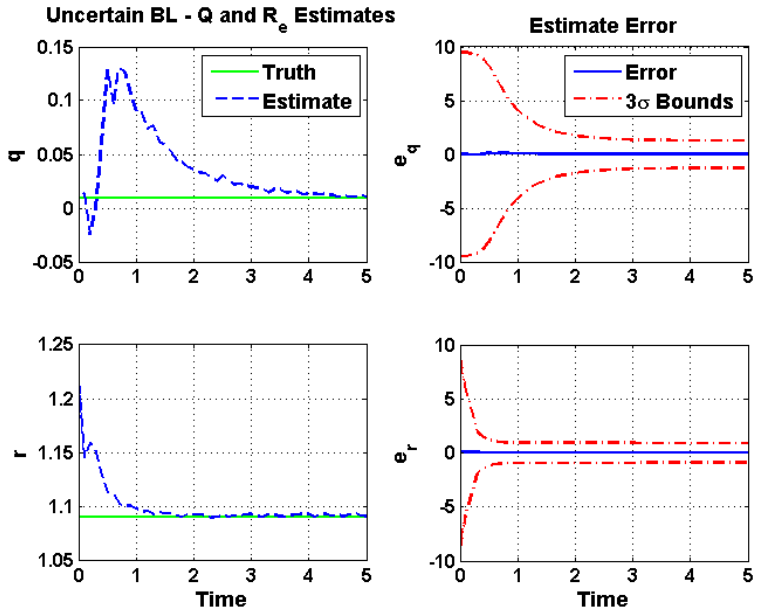

Figure II.11. Final-Time-Fixed, Uncertain Information with Behavior Learning Q and R Estimates

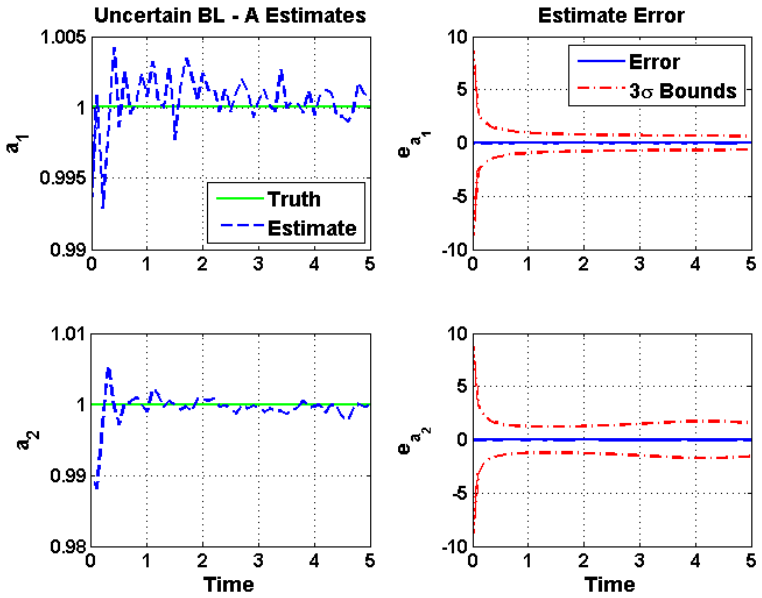

Figure II.12. Final-Time-Fixed, Uncertain Information with Behavior Learning A Estimates

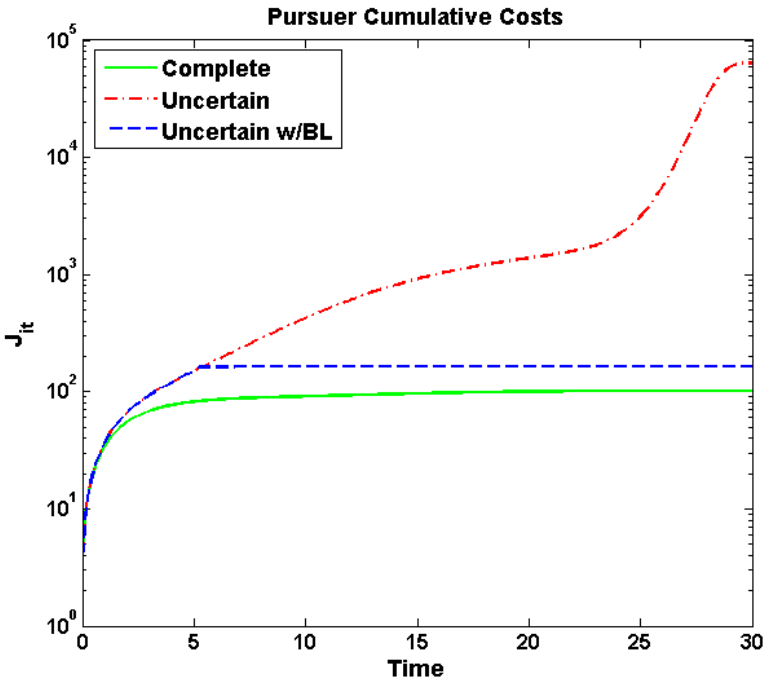

Figure II.13. Final-Time-Fixed, Uncertain Information Cumulative Cost Comparison

II.F. Summary

By identifying how player strategy manifests itself within the Kalman gain, it becomes possible to implement a behavior learning filter to estimate opponent strategy from an assumed behavior model and relative state measurements. Seemingly linear-quadratic games may require nonlinear estimation techniques due to the nonlinearities present in the Riccati equation and relative equations of motion once the form of the opponent's control input is substituted.

A cumulative cost comparison for all five cases of the final-time-fixed pursuit-evasion game is summarized in Fig. II.14. In both the incomplete and uncertain information cases, behavior learning was able to increase the pursuer's performance when a new solution was computed at $t = 5$. The pursuer's final cost summary is shown in Table II.1.

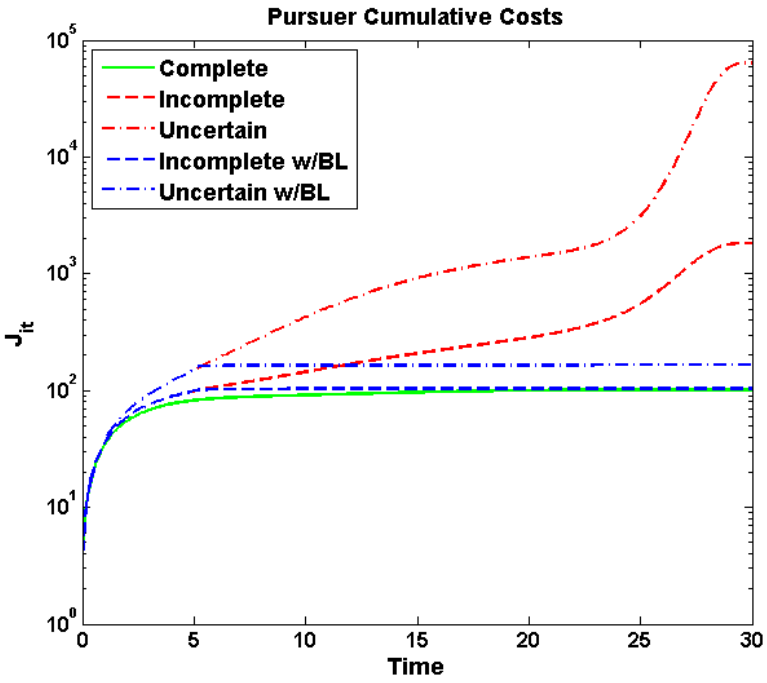

Figure II.14. Final-Time-Fixed, Cumulative Cost Comparison Summary

Table II.1. Planar Game Final-Time-Fixed Cost Summary

Information Type	**Pursuer Cost**
Complete	1.0167×10^2
Incomplete	1.9781×10^3
Uncertain	8.4537×10^4
Incomplete + BL	1.0414×10^2
Uncertain + BL	1.6567×10^2

CHAPTER III

INFINITE-HORIZON BEHAVIOR LEARNING

This chapter identifies the form of behavior learning needed for infinite-horizon pursuit-evasion games with varying levels of information availability. It is possible to implement any one of several estimation methods to estimate the parameters which capture the behavior of a given opponent once the form of behavior learning is identified. Many of the same assumptions made for the final-time-fixed case will also be exploited for infinite-horizon behavior learning. The methods presented here assume an incomplete and imperfect information scenario where the only measurements available to the player are the relative states which are subject to a zero-mean Gaussian noise distribution. Behavior learning for the incomplete information case will also be extended to encompass the uncertain information case.

Similar to the previous chapter, we will always consider the pursuer to be enabled with behavior learning. The evader will continue to be subject to an incomplete information game and will assume a zero-sum safe strategy for the duration of the game. The evader will not be using any type of behavior learning and will always be subject to an imperfect and certain information game.

III.A. Pursuit-Evasion

A zero-sum infinite-horizon LQ PE game is traditionally defined by a performance index of the form

$$J_{ZS_{inf}} = \int_{t_0}^{\infty} L\left(\boldsymbol{z}, \boldsymbol{u}_p, \boldsymbol{u}_e\right) \mathrm{dt}\,, \tag{3.1}$$

$$J_{ZS_{inf}} = \frac{1}{2}\int_{t_0}^{\infty} \left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p - \boldsymbol{u}_e^T R_e \boldsymbol{u}_e\right) \mathrm{dt}\,, \tag{3.2}$$

subject to the same dynamic constraint shown in Eqn. 2.3. Matrices Q, R_p, and R_e take on the same symmetry and definiteness assumptions found in the final-fixed-time case. The Hamiltonian for the infinite-horizon case takes on the same form defined by Eqn. 2.4 and subsequently, the stationarity conditions and costate equations yield the same results shown in Eqns. 2.6, 2.7, and 2.9.

As the final-time approaches infinity, the Riccati equation in Eqn. 2.15 can converge to a limiting solution $S(\infty)$. If $S(\infty)$ exists, then the optimal feedback control laws for the infinite-horizon case take on the form [4]

$$\boldsymbol{u}_p = -K_p \boldsymbol{z}\,, \tag{3.3}$$

$$\boldsymbol{u}_e = -K_e \boldsymbol{z}\,, \tag{3.4}$$

with the Kalman gains K_p and K_e defined as

$$K_p = R_p^{-1} B^T S\left(\infty\right)\,, \tag{3.5}$$

$$K_e = R_e^{-1} B^T S\left(\infty\right)\,. \tag{3.6}$$

The solution to $S(\infty)$ is given by the modified algebraic Riccati equation (ARE)

$$0 = Q + A^T S + SA - SBR_p^{-1}B^T S + SBR_e^{-1}B^T S\,, \tag{3.7}$$

which can be simplified to the standard ARE

$$0 = Q + A^T S + SA - SBR^{-1}B^T S\,, \tag{3.8}$$

using the effective control weight matrix relation

$$R^{-1} = R_p^{-1} - R_e^{-1}\,. \tag{3.9}$$

Equation 3.8 can be solved for S using the Schur method [30]. Note that if $\boldsymbol{z}$, $\boldsymbol{u}_p$, and $\boldsymbol{u}_e$ are scalar, then the solution to Eqn. 3.8 simply reduces to the quadratic equation.

III.A.1. Infinite-Horizon Example

A complete information example is provided along with incomplete and uncertain information examples to show how the performance of the pursuer degrades as information is revoked from the infinite-horizon pursuit-evasion game. The planar system described in Section I.A is used, only a new performance index of the form shown in Eqn. 3.2 is implemented. The complete information gain selection is summarized by

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_e = \begin{bmatrix} 1.09 & 0 \\ 0 & 1.09 \end{bmatrix}, \tag{3.10}$$

Figures III.1 - III.3 show the results for the complete information, zero-sum pursuit-evasion game in which the pursuer and evader implement the same zero-sum cost function described in Eqn. 3.2. Players start near the origin with the same non-zero initial conditions as those used for the final-fixed-time cases. The aerial view

of the players' trajectories are shown in Fig. III.1. Figure III.2 contains plots of the relative states along with the total relative displacement and speed while Fig. III.3 presents the cumulative cost and cost-to-go for each player. The total cost for the pursuer and the evader are 3.4449×10^6 for this example. Note that final cost, cumulative cost, and cost-to-go are identical for the pursuer and evader because of the complete information, zero-sum strategy implementation.

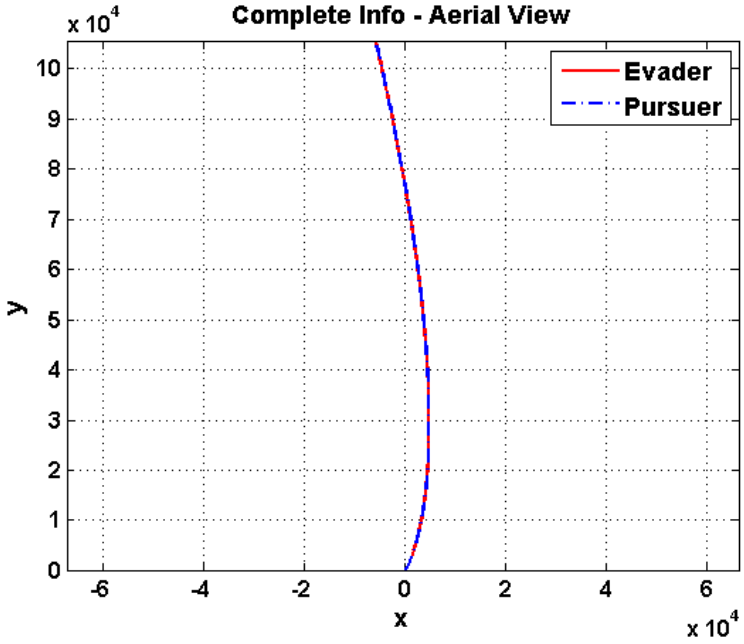

Figure III.1. Infinite-Horizon, Complete Information Aerial View

To illustrate the effects of incomplete information, the same simulation was run but slightly different gains were assumed by the pursuer while those of the evader remained constant. The pursuer's gain selection for the incomplete information case

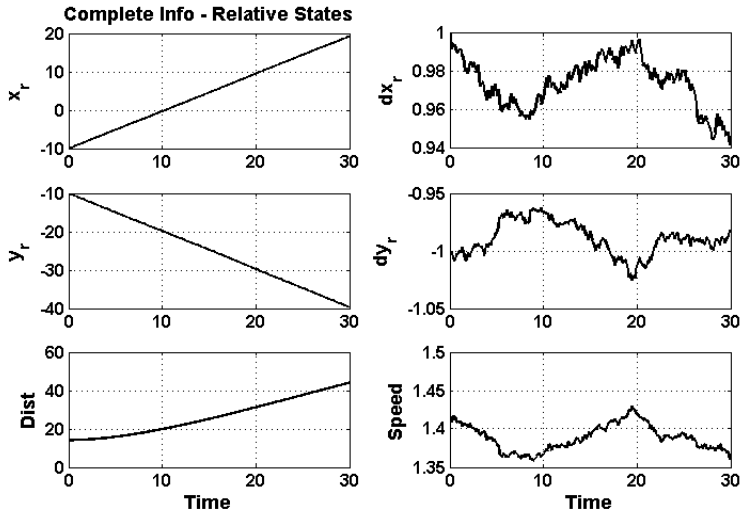

Figure III.2. Infinite-Horizon, Complete Information Relative States

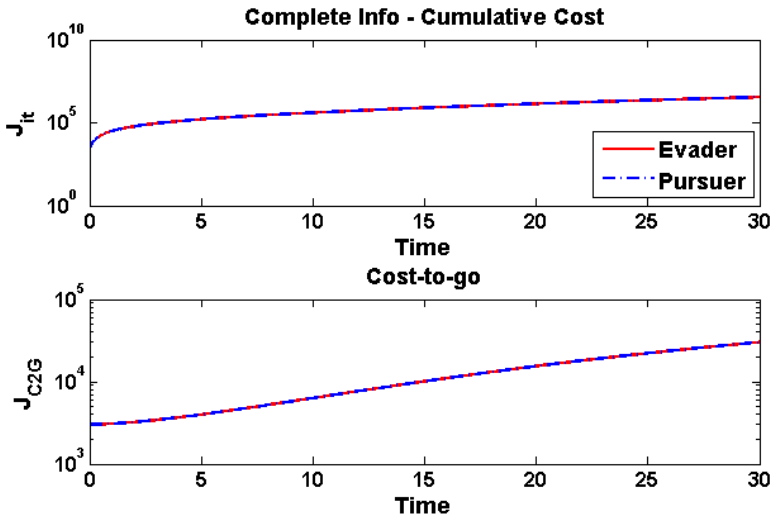

Figure III.3. Infinite-Horizon, Complete Information Cost Analysis

is summarized by

$$Q_p = \begin{bmatrix} 9.5 & 0 & 0 & 0 \\ 0 & 9.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R_{p_p} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_{e_p} = \begin{bmatrix} 1.1 & 0 \\ 0 & 1.1 \end{bmatrix}. \tag{3.11}$$

The results for the incomplete, imperfect information game using the same initial conditions are shown in Figs. III.4 - III.6. The aerial view of the players' trajectories are shown in Fig. III.4. Figure III.5 contains plots of the relative states along with the total relative displacement and speed while Fig. III.6 presents the cumulative cost and cost-to-go for each player. The total cost of the pursuer is 7.5409×10^{46} while that for the evader is 7.5037×10^{46}. The introduction of incomplete information affected both players as shown by the total cost and turns out to be completely devastating to the pursuer's performance. These results indicate how poor assumptions related to an opponent's strategy can decrease the performance of a player in the infinite-horizon case.

Finally, the effects of an incomplete, imperfect, uncertain information game are shown in Figs. III.7 - III.9. In addition to the gain errors in the previous example, the pursuer was also subject to modeling uncertainties. Here, modeling uncertainty is defined as errors in the model matrix A. Therefore, the non-zero elements of the

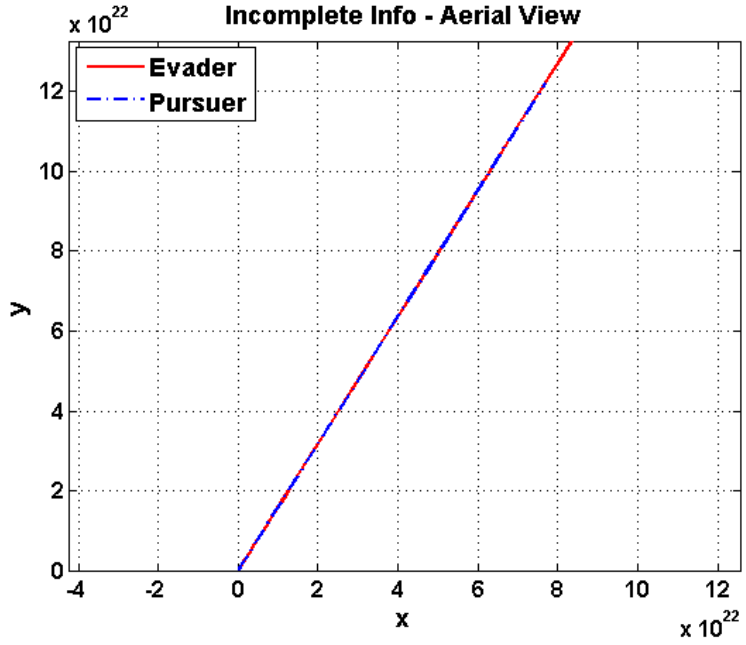

Figure III.4. Infinite-Horizon, Incomplete Information Aerial View

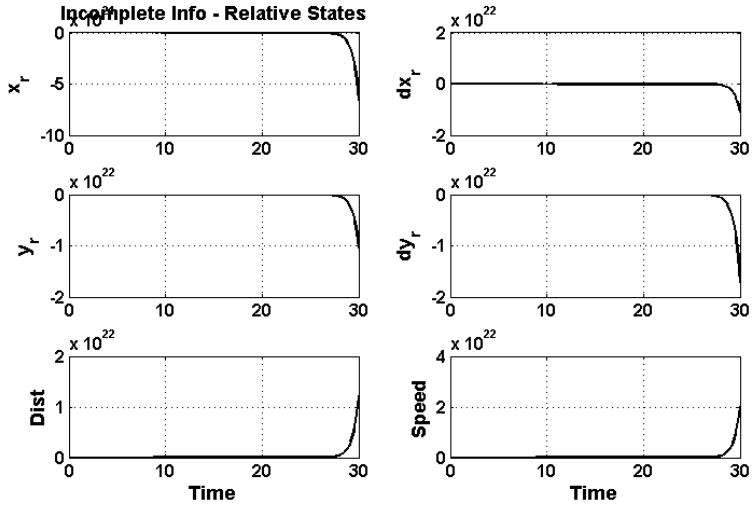

Figure III.5. Infinite-Horizon, Incomplete Information Relative States

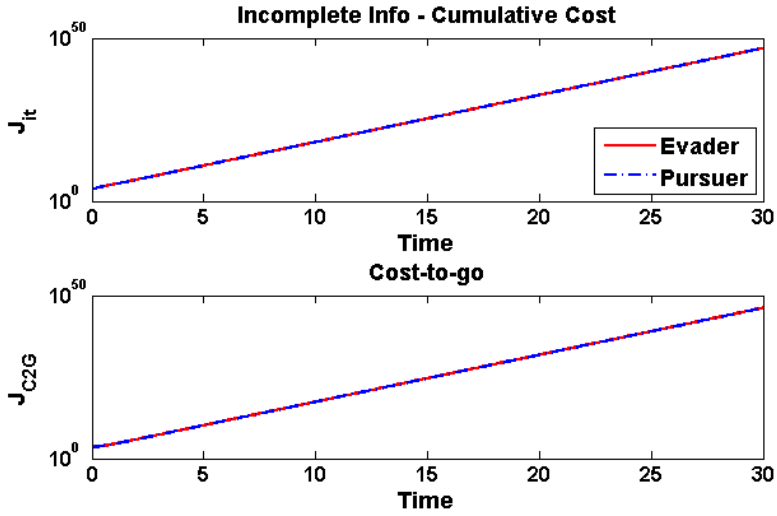

Figure III.6. Infinite-Horizon, Incomplete Information Cost Analysis

model matrix A assumed by the pursuer were modified such that

$$A = \begin{bmatrix} 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.9 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.12}$$

The true dynamic model assumed by the evader remained constant throughout all simulations.

Once the uncertain information is introduced to the game, the pursuer's performance diminishes even further as illustrated in Figs. III.7 and III.8. The cost analysis for the uncertain information game is shown in Fig. III.9. Total cost for the pursuer was 3.8193×10^{60} while the evader cost was 3.7995×10^{60}.

Figure III.10 illustrates how the pursuer's performance degrades as information is revoked from the infinite-horizon pursuit-evasion game.

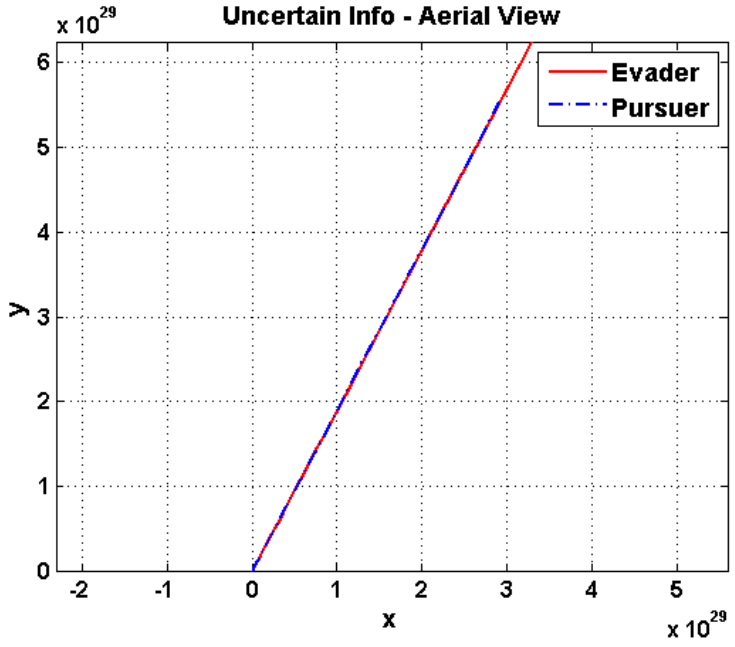

Figure III.7. Infinite-Horizon, Uncertain Information Aerial View

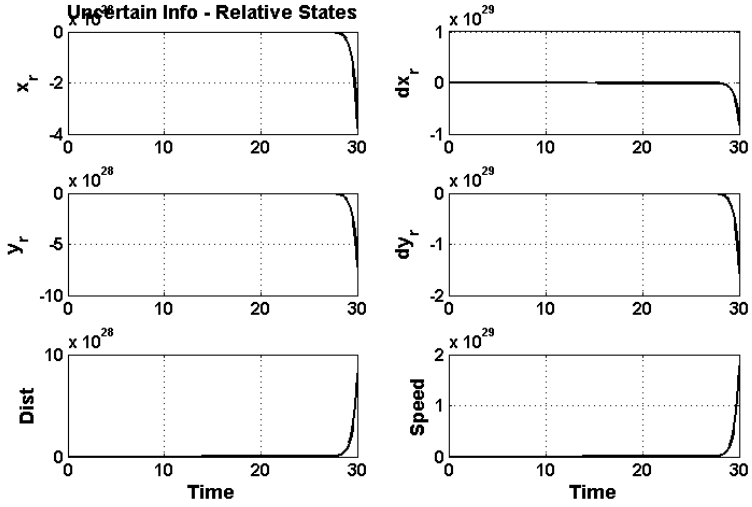

Figure III.8. Infinite-Horizon, Uncertain Information Relative States

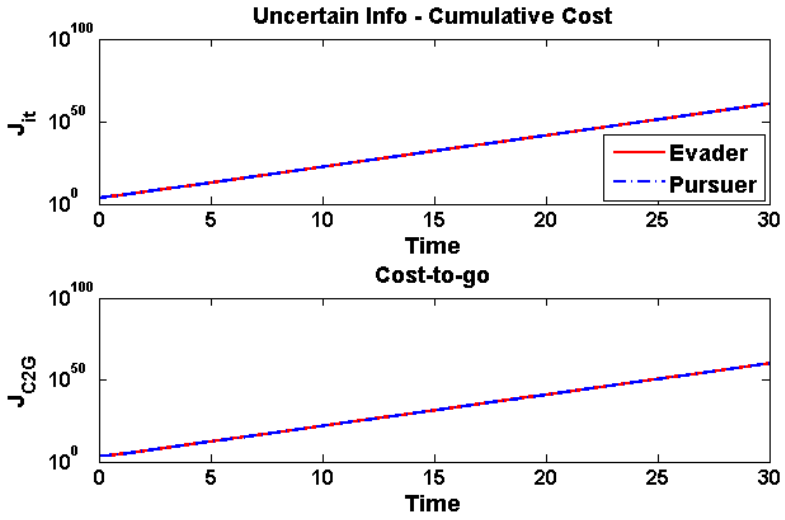

Figure III.9. Infinite-Horizon, Uncertain Information Cost Analysis

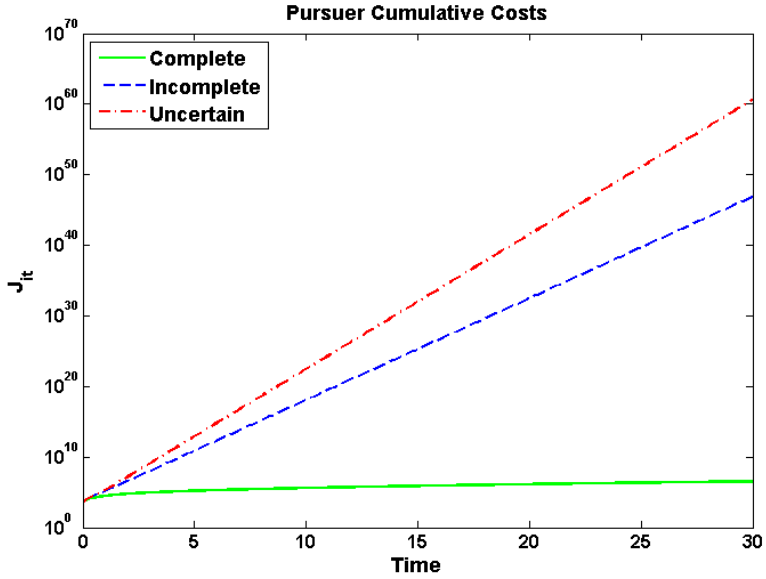

Figure III.10. Infinite-Horizon, Cumulative Cost Comparison

III.B. Incomplete Information Behavior Learning

From Eqn. 3.6 it is immediately apparent that the evader's Kalman gain is dependent on B, S, and R_e. Because S is the solution to the ARE found in Eqn. 3.7, the Kalman gain is once again $K_e = K_e(A, B, S, Q, R_p, R_e)$. Taking A and B to be known and scaling the gains with respect to $R_p = \mathbf{1}$, we can define K_e as a function of the unknowns $K_e = K_e(S, Q, R_e)$. It is important to apply all pertinent knowledge about the game to simplify the behavior learning task. Considering the same example problem of differentially driven vehicles in the horizontal plane, as described in Sect. II.A, the same form of the gain assumptions will be applied here.

With the assumed form of Q and R_e shown in Eqn. 2.20, solution of the the ARE in Eqn. 3.7 produces an S of the form

$$S = \begin{bmatrix} s_1 & 0 & s_3 & 0 \\ 0 & s_1 & 0 & s_3 \\ s_3 & 0 & s_2 & 0 \\ 0 & s_3 & 0 & s_2 \end{bmatrix}. \tag{3.13}$$

Substituting the form of S and R_e into Eqn. 3.6, the 2×4 Kalman gain used by the evader is

$$K_e = \begin{bmatrix} \frac{s_3}{r} & 0 & \frac{s_2}{r} & 0 \\ 0 & \frac{s_3}{r} & 0 & \frac{s_2}{r} \end{bmatrix}. \tag{3.14}$$

Recall that the form of S from the solution to the ARE is constant. This is important because it follows that with a constant R_e, the evader's Kalman gain is also a constant. The form for K_e for the differentially driven vehicles in the plane is encouraging because we have revealed for this particular example, the evader's

Kalman gain is made up of only two independent and constant elements. Therefore, we can conclude that the entire strategy for the evader taking part in an infinite-horizon pursuit-evasion game can be captured by the parameters k_1 and k_2 where

$$K_e = \begin{bmatrix} k_1 & 0 & k_2 & 0 \\ 0 & k_1 & 0 & k_2 \end{bmatrix} . \tag{3.15}$$

Note that the form of K_e in Eqn. 3.15 is not always the case. However, it should always be possible to reduce the number of independent elements found in K_e by applying knowledge to the system. For the case with four relative states and two control inputs, K_e will have at most eight unique elements and will always be constant for the infinite-horizon scenario.

For infinite-horizon behavior learning, the states to estimate are now defined by the vector

$$\boldsymbol{x} = [z_1,\, z_2,\, z_3,\, z_4,\, k_1,\, k_2] , \tag{3.16}$$

$$\boldsymbol{x} = [x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6] , \tag{3.17}$$

where states $x_1 - x_4$ are time-varying and whose state equations are given by

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u_p} + BK_e\boldsymbol{z} \tag{3.18}$$

The state equations needed for the nonlinear estimator are summarized by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) , \tag{3.19}$$

where

$$\boldsymbol{f} = \begin{bmatrix} x_3 \\ x_4 \\ x_1x_5 + x_3x_6 + u_{p_1} \\ x_5x_7 + x_4x_6 + u_{p_2} \\ 0 \\ 0 \end{bmatrix}, \tag{3.20}$$

and $\boldsymbol{u_p} = [u_{p_1}, u_{p_2}]$.

The measurements available are the relative states defined by

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{h}(\boldsymbol{x}_k) = [x_1\,, x_2\,, x_3\,, x_4]^T\,. \tag{3.21}$$

Equations 3.19 and 3.21 are in the standard form needed for a nonlinear filter. Once a nonlinear estimator is employed and has converged on a solution for the estimates given by Eqn. 3.17, the pursuer can then compute the evader's Kalman gain at each instance in time because it is fixed. The gain can be continuously monitored and the pursuer has the ability to modify the solution as necessary.

III.C. Uncertain Information Behavior Learning

The behavior learning framework presented in the previous section may be extended to the uncertain information case. In an uncertain information game, a player is subject to modeling errors present in the relative dynamic model. In Section I.A, this was defined to be errors in the model matrix A. The independent elements of A can be added to the state estimate vector described by Eqn. 3.20.

Much like the final-time-fixed case, it is essential that all available information

about the system is applied. Assuming the same form for A as provided in Section II.C, the new parameters to estimate become a_1 and a_2. For the time-invariant case, the state equation describing these new parameters are simply $\dot{a}_1 = \dot{a}_2 = 0$.

The behavior learning state estimate vector is then augmented such that

$$\boldsymbol{x} = [z_1,\, z_2,\, z_3,\, z_4,\, k_1,\, k_2,\, a_1,\, a_2]\ , \tag{3.22}$$

$$\boldsymbol{x} = [x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6,\, x_7,\, x_8]\ , \tag{3.23}$$

and the state equation vector becomes

$$\boldsymbol{f} = \begin{bmatrix} x_3 x_7 \\ x_4 x_8 \\ x_1 x_5 + x_3 x_6 + u_{p_1} \\ x_5 x_7 + x_4 x_6 + u_{p_2} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\ . \tag{3.24}$$

No additional measurements are available, therefore the form of Eqn. 3.21 remains the same. For this infinite-horizon, uncertain information example, the observability matrix O is of full rank. This form of the behavior learning algorithm will be implemented by the pursuer for the uncertain information case.

III.D. Augemented Strategy

When an agent is enabled with behavior learning in an infinite-horizon scenario, it is possible for that agent to turn the pursuit-evasion game into a one-sided optimal

control problem, just like the final-time-fixed scenario. The objective is to estimate the opponent's strategy then use that information to play more effectively by predicting the opponent's behavior with the behavior learning solution. By taking on the pursuer's perspective, it becomes apparent that the pursuer is interested in estimating the evader's constant Kalman gain, K_e. Once K_e is known, then the new one-sided optimal control problem for the pursuer becomes

$$J_{P_{inf}} = \min \frac{1}{2} \int_{t_0}^{\infty} \left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p \right) \mathrm{dt} \,, \tag{3.25}$$

subject to the modified system

$$\dot{\boldsymbol{z}} = \left(A + BK_e \right) \boldsymbol{z} + B\boldsymbol{u}_p \,. \tag{3.26}$$

Following the same augmentation process as the final-time-fixed case, the Hamiltonian is formed and the stationarity condition coupled with the costate equation solution yields the same feedback control law given by Eqns. 2.16 and 2.18. For the infinite-horizon framework, S is found using

$$0 = Q + \left(A + BK_e \right)^T S + S \left(A + BK_e \right) - SBR_p^{-1} B^T S \,. \tag{3.27}$$

which still takes on the form of an ARE. Note the Q, and R_p here are the pursuer's assumed gains and different from those which are estimated by the pursuer and assumed by the evader.

III.E. Implementation

A behavior learning filter for an infinite-horizon scenario reduces to a much simpler implementation than that employed for the final-time-fixed case. This is

because the independent elements of Kalman gain matrix are constant. If enough knowledge can be applied to the system, it was shown in Sect. III.B that the number of independent Kalman gain elements can be reduced to two.

Using the same extended Kalman filter framework reviewed in Sect. III.B, an incomplete information version of this filter for the infinite-horizon case can be implemented using Eqns. 3.20 and 3.21. Similarly, the uncertain information behavior learning filter uses Eqns. 3.24 and 3.21.

III.E.1. Incomplete Information Results

The incomplete information example shown in Section III.A was simulated again but this time with an EKF version of the behavior learning algorithm running. At $t = 1$, a new optimal control solution was computed by the pursuer using the augmented strategy defined in Section III.D. Figures III.11 - III.15 convey the results for the incomplete information behavior learning case. By computing a new solution based the the behavior learning results, the pursuer was able to reduce its total cost to 2.3957×10^5 while the evader's cost was computed to be 2.4772×10^5, compared to the non-behavior learning values of 7.5409×10^{46} and 7.5037×10^{46} for the purser and evader, respectively.

The pursuer's behavior learning algorithm provided effective gain estimates describing the influence of k_1 and k_2 on the system shown in Fig. III.14. A comparison of the cost and cost-to-go for the complete, incomplete, and incomplete with behavior learning cases in Fig. III.15.

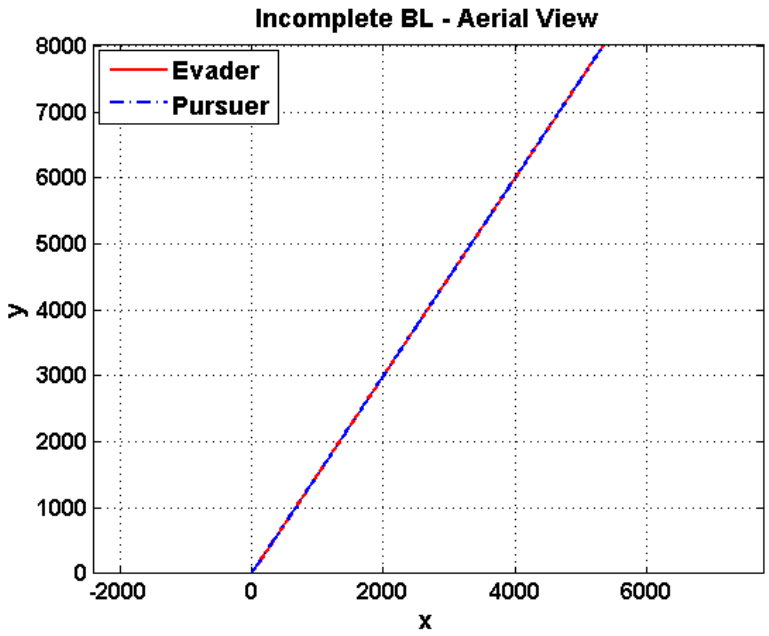

Figure III.11. Infinite-Horizon, Incomplete Information with Behavior Learning Aerial View

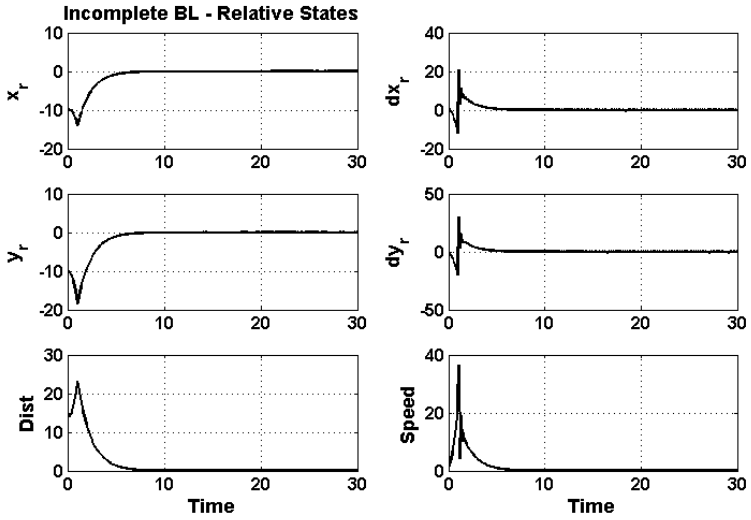

Figure III.12. Infinite-Horizon, Incomplete Information with Behavior Learning Relative States

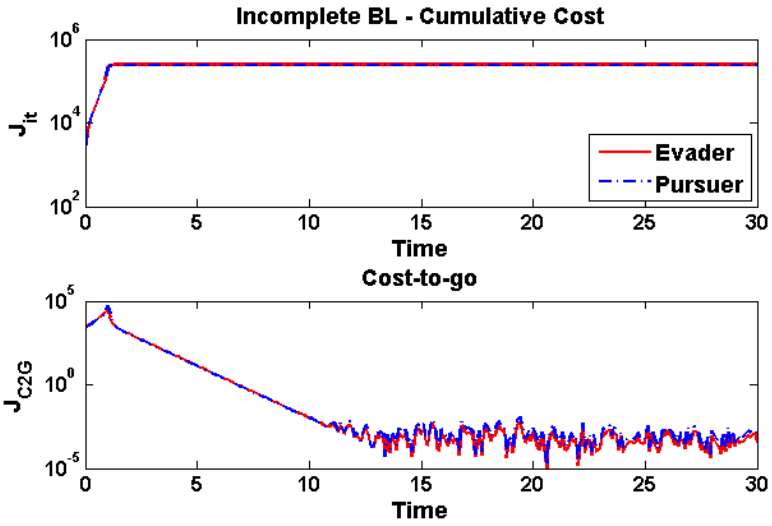

Figure III.13. Infinite-Horizon, Incomplete Information with Behavior Learning Cost Analysis

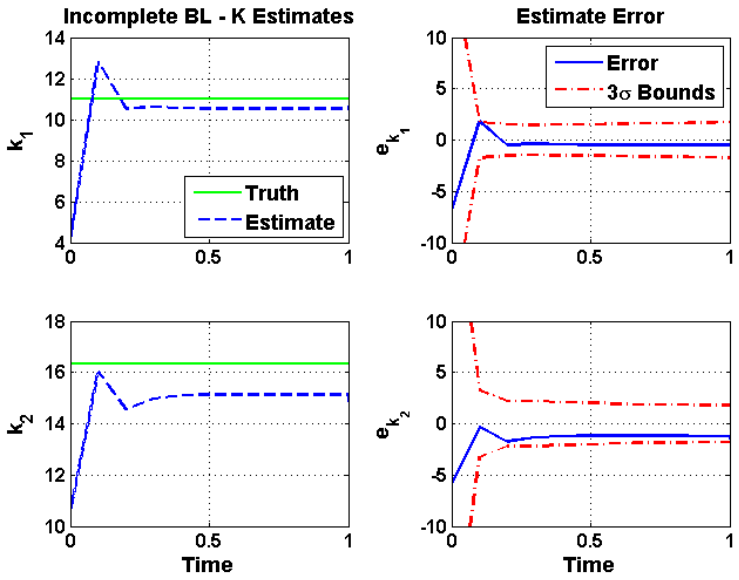

Figure III.14. Infinite-Horizon, Incomplete Information with Behavior Learning K Estimates

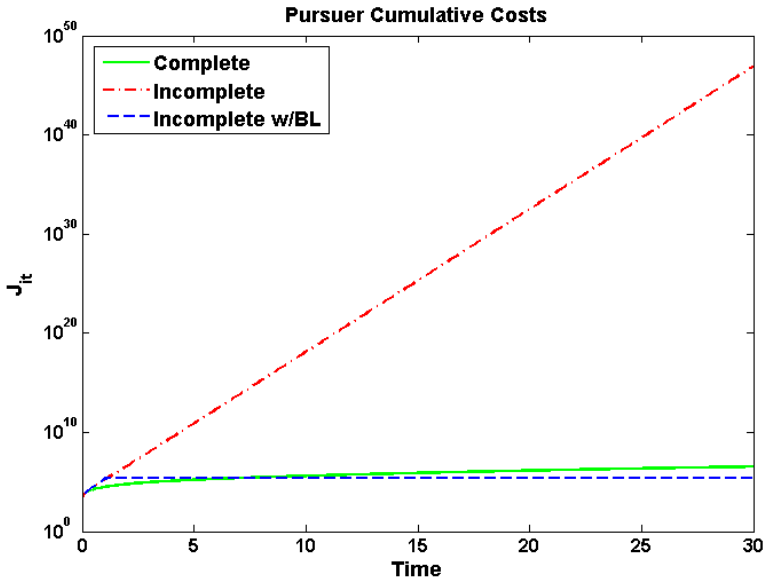

Figure III.15. Infinite-Horizon, Incomplete Information Cumulative Cost Comparison

III.E.2. Uncertain Information Results

The uncertain information example shown in Section I.A was simulated again but this time with an EKF version of the behavior learning algorithm running. At $t = 2$, a new optimal control solution was computed by the pursuer using the augmented strategy defined in Section III.D. Figures III.16 - III.21 convey the results for the behavior learning case. By computing a new solution based the the behavior learning results, the pursuer was able to reduce its total cost to 2.4221×10^7 while the evader's cost was computed to be 2.4946×10^7 compared to the non-behavior learning values of 3.8193×10^{60} and 3.7995×10^{60} for the pursuer and evader, respectively.

The pursuer's behavior learning algorithm was able to provide effective estimates for k_1 and k_2 as shown in Fig. III.19. Moreover, the estimates of the independent elements found within model matrix A provided reasonable estimates as evident from

Fig. III.20. A comparison of the cost and cost-to-go for the complete information, incomplete information, and incomplete information with behavior learning cases in Fig. III.21.

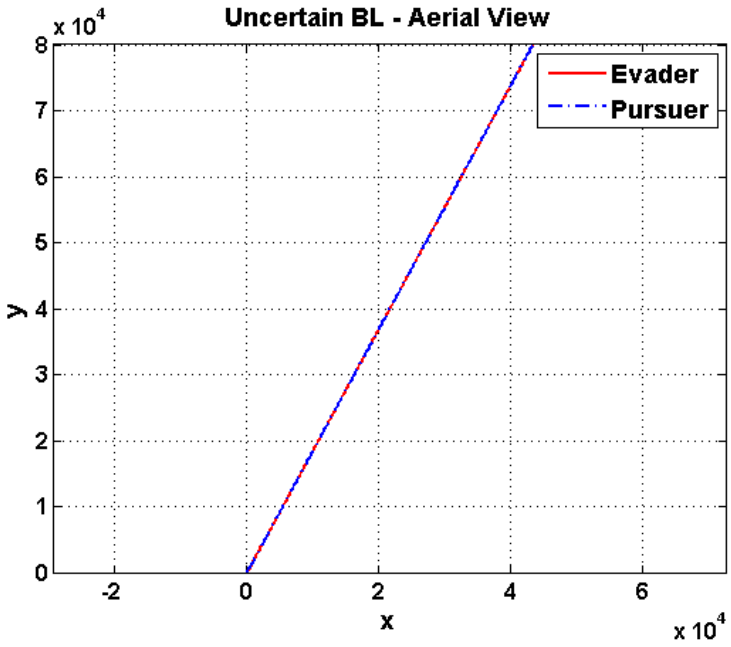

Figure III.16. Infinite-Horizon, Uncertain Information with Behavior Learning Aerial View

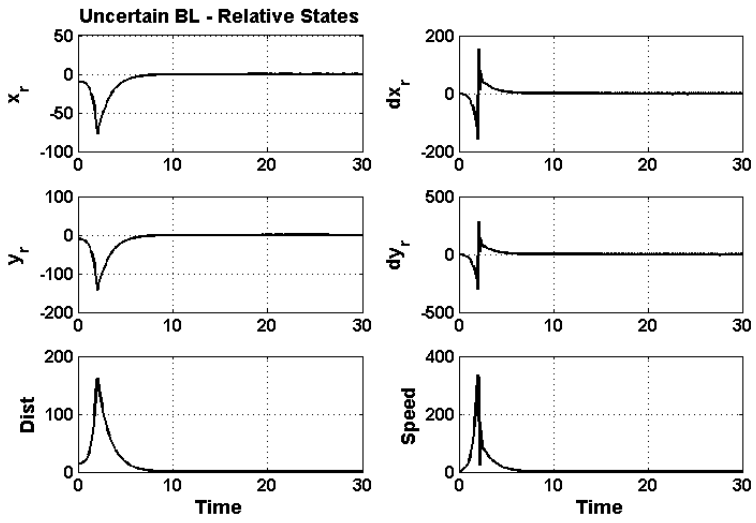

Figure III.17. Infinite-Horizon, Uncertain Information with Behavior Learning Relative States

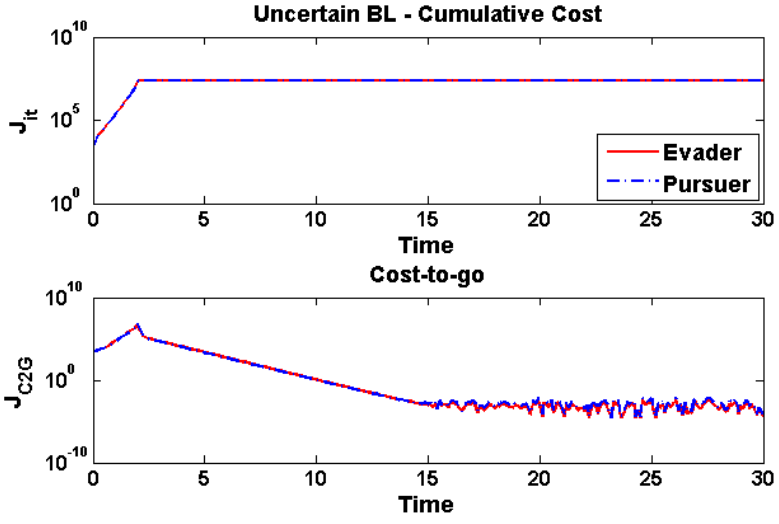

Figure III.18. Infinite-Horizon, Uncertain Information with Behavior Learning Cost Analysis

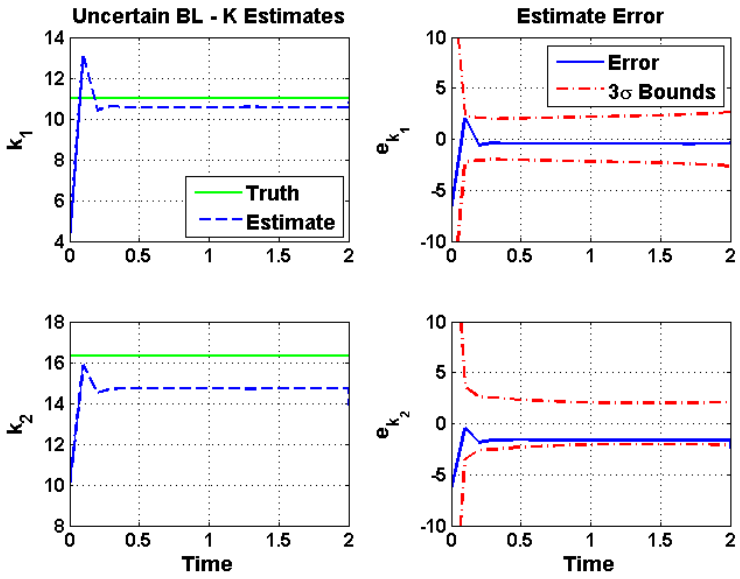

Figure III.19. Infinite-Horizon, Uncertain Information with Behavior Learning K Estimates

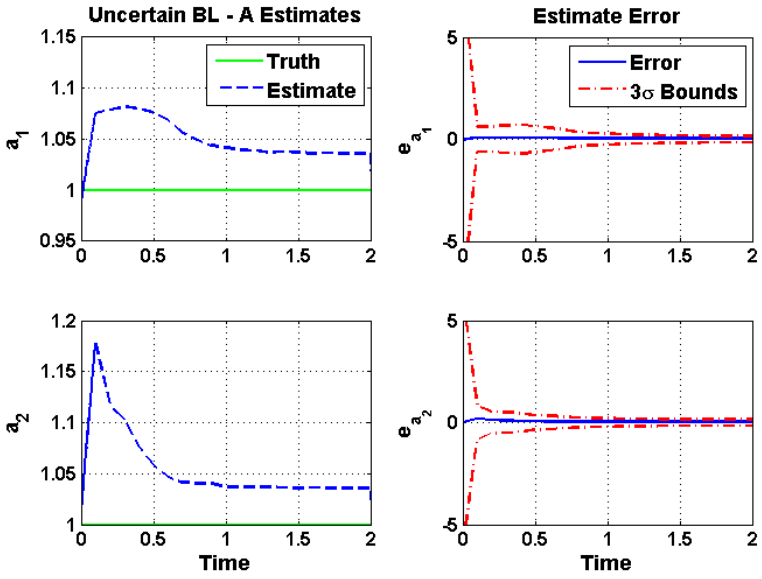

Figure III.20. Infinite-Horizon, Uncertain Information with Behavior Learning A Estimates

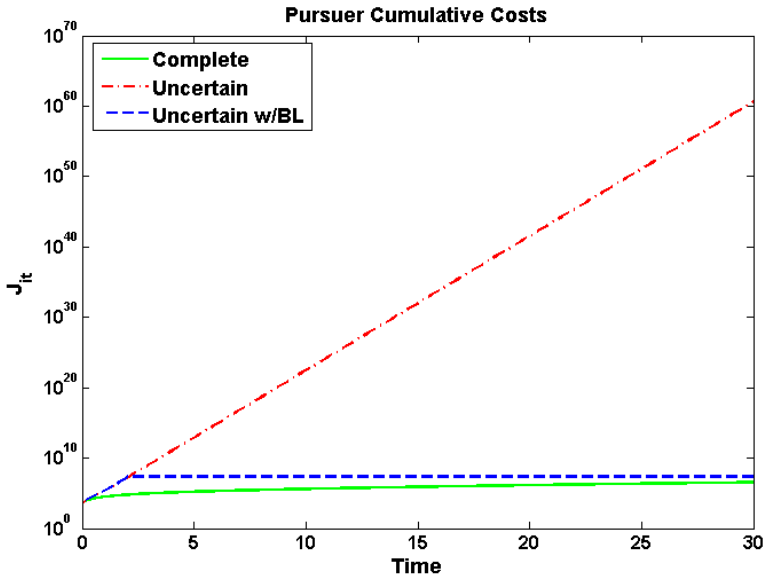

Figure III.21. Final-Time-Fixed, Uncertain Information Cumulative Cost Comparison

III.F. Summary

Figure III.22 shows a cumulative cost comparison for all five cases of the infinite-horizon pursuit-evasion game. In both the incomplete and uncertain information cases, behavior learning was able to increase the pursuer's performance when a new solution was computed. Note the time of strategy augmentation for the pursuer occurs at $t = 1$ for the incomplete information case and $t = 2$ for the uncertain information case. The pursuer's final cost summary is shown in Table III.1.

Infinite-horizon behavior learning can be computationally more efficient due to the fact that the unique elements found within the Kalman gain matrix are constant which simplifies things from a behavior estimation perspective. However, it was shown that the the behavior learning filter converged on effective gains for k_1 and k_2 that were not necessarily the true gains. Still, this confirms that the behavior

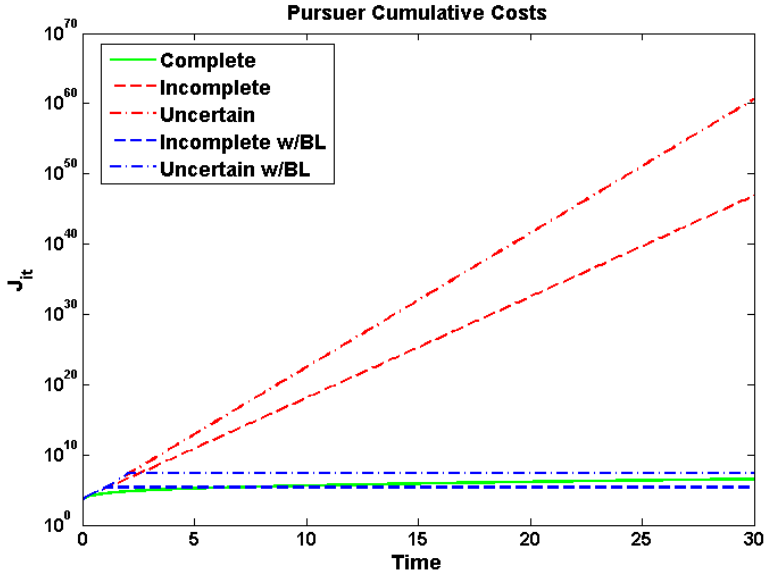

Figure III.22. Infinite-Horizon, Cumulative Cost Comparison Summary

Table III.1. Planar Game Infinite-Horizon Cost Summary

Information Type	Pursuer Cost
Complete	3.4449×10^{6}
Incomplete	7.5409×10^{46}
Uncertain	3.8193×10^{60}
Incomplete + BL	2.3957×10^{5}
Uncertain + BL	2.4221×10^{7}

learning framework can be effective even if the true behavior of the opponent is not perfectly modeled. All a player needs to do is develop a model that is representative of their opponent's behavior. If a model can be converged upon, then the player has the opportunity to play more effectively than simply using a zero-sum safe strategy.

CHAPTER IV

FINAL-TIME-FREE BEHAVIOR LEARNING

Pursuit-evasion games of the final-time-free nature are not often studied. Depending on the performance index, these types of scenarios most often reduce to minimum-time games. One major hurdle issue with minimum-time pursuit-evasion games is that a final-state constraint is used to completely define the problem. These final state constraints usually define interception or rendezvous and can only be valid if the pursuer is guaranteed to capture the evader. That is, the pursuer must be more agile than the evader. This chapter will develop a minimum-time pursuit-evasion game and discuss how behavior learning can be used to help the pursuer compute his control for an incomplete information game.

IV.A. Minimum-Time Pursuit-Evasion

Consider a scalar, minimum-time pursuit-evasion game defined by the performance index

$$J_{min} = \int_{t_0}^{t_f} \mathrm{dt}\,, \tag{4.1}$$

and the state equation

$$\ddot{x} = u_p - u_e\,, \tag{4.2}$$

where x represents the relative state between the pursuer and the evader. Each player is subject to control constraints which are defined by

$$|u_p| \leq 1\,, \tag{4.3}$$

$$|u_e| \leq K_e\,. \tag{4.4}$$

By defining the relative state and its rate as $x = z_1$ and $\dot{x} = z_2$, the solution must satisfy the final state constraint

$$\boldsymbol{z}_f = \begin{bmatrix} z_{1_f} \\ \\ z_{2_f} \end{bmatrix} = \mathbf{0}\,, \tag{4.5}$$

where the subscript f denotes the state at the final time, t_f.

The pursuer's goal is to minimize the amount of time it takes to satisfy the final-state constraint while the evader wants to maximize the amount of time necessary. Because of these similar but opposite objectives, this can be defined as a zero-sum, minimum-time pursuit-evasion game. Both players will exert their maximum allowable control due to the minimum-time nature of the problem and because there is no weighting present on any control variables in the performance index. In order for the final-state constraint to be satisfied and for a solution to exist, the pursuer must have more control authority or be more agile than the evader. Mathematically, $K_e < 1$. The solution will be of the bang-bang type that is typical of minimum-time problems. The PE solution requires the switching function to be found [3].

The Hamiltonian can be written as

$$H = 1 + \lambda_1 \dot{z}_1 + \lambda_2 \dot{z}_2 = 1 + \lambda_1 z_2 + \lambda_2 u_p - \lambda_2 u_e\,. \tag{4.6}$$

By inspection of Eqn. 4.6, if the goal is for the pursuer (evader) to minimize (maximize) the Hamiltonian, then the switching function for both players is dependent

upon the sign associated with λ_2. We can conclude

$$\text{if } \lambda_2 < 0 \qquad \text{then} \qquad u_p = 1,\, u_e = K_e\,, \tag{4.7}$$

$$\text{if } \lambda_2 > 0 \qquad \text{then} \qquad u_p = -1,\, u_e = -K_e\,. \tag{4.8}$$

From the costate equations, it follows

$$\frac{\partial H}{\partial z_1} = -\dot{\lambda}_1 \qquad \rightarrow \qquad \dot{\lambda}_1 = 0\,, \tag{4.9}$$

$$\frac{\partial H}{\partial z_2} = -\dot{\lambda}_2 \qquad \rightarrow \qquad \dot{\lambda}_2 = -\lambda_1\,. \tag{4.10}$$

The ideal solution trajectory lies on intersecting parabolas and the optimal trajectory for a game of this nature utilizes two parabolas. The first parabola depends on the initial conditions and the second parabola always intersects the final-state constraint. For this particular example, the final state constraint is represented by the state space origin. The shape of these parabolas are defined by the total control input which is defined as

$$w = u_p - u_e = 1 - K_e\,. \tag{4.11}$$

The parabola equations can be solved for by manipulation of the state equations given by $\dot{z_1} = z_2$ and $\dot{z_2} = w$.

$$\frac{\mathrm{d}z_1}{\mathrm{d}z_2} = \frac{\mathrm{d}z_1/\mathrm{dt}}{\mathrm{d}z_2/\mathrm{dt}} = \frac{\dot{z_1}}{\dot{z_2}} = \frac{z_2}{w}\,. \tag{4.12}$$

Therefore,

$$w\mathrm{d}z_1 = z_2\mathrm{d}z_2\,, \tag{4.13}$$

which can be integrated on both sides to produce

$$\int_{t_0}^{t_f} w\mathrm{d}z_1 \;=\; \int_{t_0}^{t_f} z_2\mathrm{d}z_2\,, \tag{4.14}$$

$$w\left(z_{1_f} - z_{1_0}\right) \;=\; \frac{1}{2}\left(z_{2_f}^2 - z_{2_0}^2\right)\,. \tag{4.15}$$

By imposing the final-state constraint given by Eqn. 4.5, Eqn. 4.15 becomes the trajectory parabola equation given by

$$wz_{1_0} = \frac{z_{2_0}^2}{2}\,. \tag{4.16}$$

Recall the control switching functions given by Eqns. 4.7 and 4.8. For these cases, w_{min} and w_{max} can be defined.

$$w_{max} = w\left(\lambda_2 < 0\right) = 1 - K_e\,, \tag{4.17}$$

$$w_{min} = w\left(\lambda_2 > 0\right) = -1 + K_e\,. \tag{4.18}$$

The parabolas can be plotted in the state space of z_1 and z_2. When w_{max} is used with Eqn. 4.16, the coefficient on z_{1_0} is positive and the parabola vertex is located on the left side of the trajectory. A negative coefficent is present on z_{1_0} when w_{min} is used and the vertex is located on the right side of the parabola. Because the shape of these trajectory parabolas are dictated by w and u_p is fixed for this example, the trajectory becomes a function for the selection of K_e by the evader. Figures IV.1, IV.2, and IV.3 illustrate these shape differences for K_e selections of 0, 0.25, and 0.75, respectively. Note that when $K_e = 0$, the trajectory becomes that of a simple bang-bang minimum-time solution for a single agent because $K_e = 0$ represents no evader input [3].

The path for w_{min} parabolas follow a downward trajectory while the w_{max} parabolas follow an upward trajectory. Depending upon the initial conditions, the game trajectory would begin a particular parabola and follow it until it met the parabola that intersects the origin. At that instance in time, the control switches and the game path takes the appropriate trajectory to the origin.

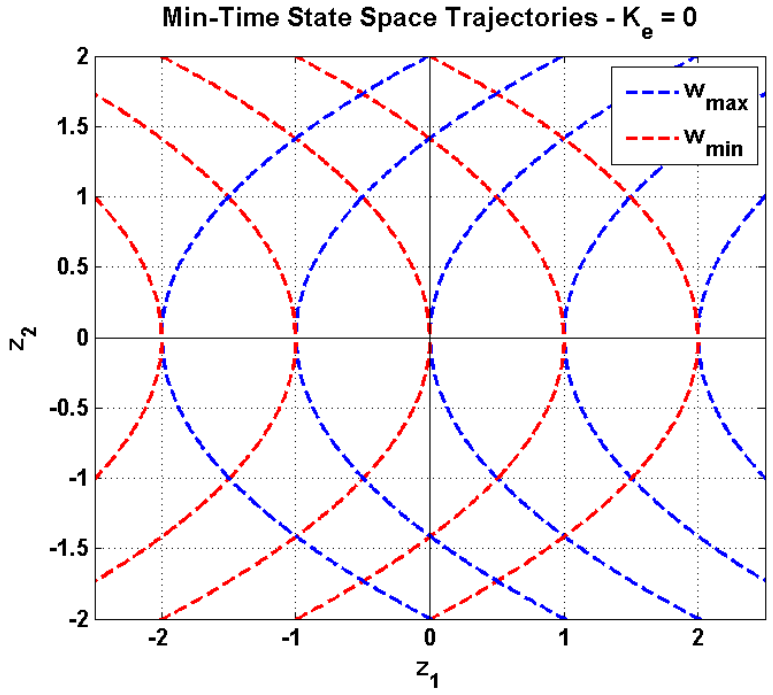

Figure IV.1. Minimum-Time PE Trajectories with $K_e = 0$

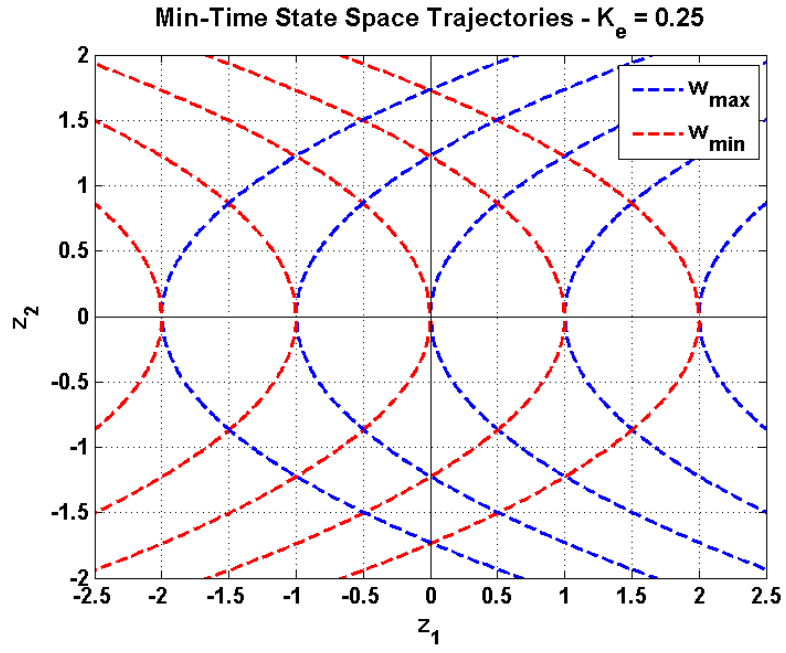

Figure IV.2. Minimum-Time PE Trajectories with $K_e = 0.25$

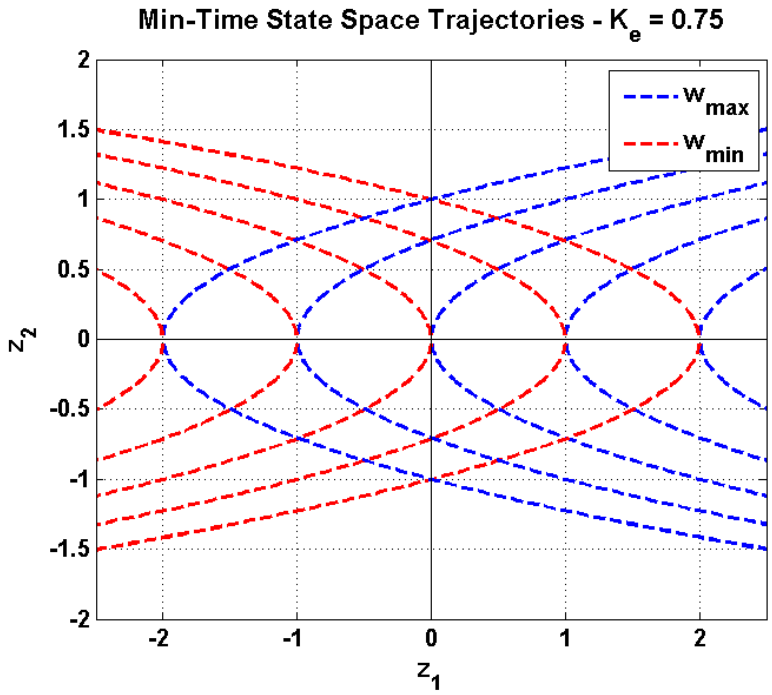

Figure IV.3. Minimum-Time PE Trajectories with $K_e = 0.75$

IV.B. Incomplete Information Behavior Learning

The optimal PE solutions begin on one trajectory parabola and continue on it until that parabola intersects with one which leads to the origin. It is at this intersection where the switching function λ_2 crosses zero and w_{max} switches to w_{min} or vice versa. If the switching function is not properly evaluated, then the trajectory must remain on the new parabola until it intersects one which leads to the origin again. Based on the trajectories shown in Fig. IV.1 - IV.3, it becomes clear that the selection of K_e is essential to where the intersections occur and therefore necessary to know in order for the pursuer to switch at the appropriate time.

An incomplete information minimum-time game is one in which a player is unaware of the their opponent's strategy. For the minimum-time case, it will be as-

sumed that the evader plays a complete information game and can properly evaluate the switching function. In this minimum-time example, behavior learning aims to estimate the value of K_e which defines the control constraint imposed on the evader. If the pursuer can estimate K_e, then the pursuer has the ability to switch control schemes at the correct time to switch to the proper trajectory and arrive at the final-state constraints.

For minimum-time behavior learning, the states to estimate are defined by the vector

$$\boldsymbol{x} = \begin{bmatrix} z_1 \\ z_2 \\ K_e \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} , \tag{4.19}$$

where states x_1 and x_2 are time-varying and whose state equations are given by

$$\dot{z}_1 = z_2 , \tag{4.20}$$

$$\dot{z}_2 = u_p - K_e . \tag{4.21}$$

The state equations needed for the estimator are summarized by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) , \tag{4.22}$$

where

$$\boldsymbol{f} = \begin{bmatrix} x_2 \\ u_p - x_3 \\ 0 \end{bmatrix} . \tag{4.23}$$

Relative states are available for measurement and are defined by

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) = \left[x_1 , x_2\right]^T . \tag{4.24}$$

Equations 4.22 and 4.24 are in the standard form needed for an estimator. For this particular example, a linear estimator can be used because the states appear linearly in $\boldsymbol{f}$. In the event the evader decides to play non-optimally, i.e. uses a value for K_e that is less than the originally specified K_e, it would be ideal for the pursuer to be aware of this because it would alter the optimal trajectory. Therefore, it is in the best interest of the pursuer to continuously monitor the value used for K_e and use a sequential linear estimator for processing. A full estimator would allow the state measurements to be smoothed that are brought about by imperfect information and provide an estimate of K_e needed to deal with the incomplete information.

IV.C. Summary

Behavior learning plays an important role in the minimum-time pursuit-evasion scenario. Because of the switching nature of the state space trajectory, behavior learning is used to estimate the gain K associated with an opponent which dictates when the optimal trajectory switch occurs. When faced with an incomplete information minimum-time game, a player runs the risk of improperly evaluating the switching function and missing the optimal time at which to switch trajectory parabolas.

In the presented framework, the choice of the gain K_e determines the agility or control authority associated with the evader which in turn dictates the shape of the state space trajectory. Depending on the complexity of the relative state equations, the behavior learning filter could be implemented with a linear estimator.

CHAPTER V

A MINIMUM-TIME EXAMPLE

Minimum-time optimal control problems are a specific type of final-time-free problems in which the total time is the only factor considered in the performance index. Problems of the minimum-time nature are ill posed without additional constraints because the solution would simply be the selection of a control input which is infinite. Therefore, limits on the magnitude of the control input are specified along with a final-state-constraint. The final-state-constraint is used to specify an end condition so an interesting solution exists.

Three major hurdles exist in the implementation of PE games of the minimum-time type. The first is that a final-state-constraint must be imposed to completely define the problem, yet to do this, the pursuer must be able to catch the evader. That is, the problem can only be properly defined if the evader is guaranteed to be captured. Second, the control switching that occurs from the limits imposed by on the control magnitude are undesirable, especially when more than one intelligent agent is making decisions. The simple minimum-time problem that once had a bang-bang solution can now undergo constant switching. Finally, the ongoing desire for feedback solutions becomes more difficult to fulfill as minimum-time solutions are generally open loop.

Minimum-time pursuit-evasion problems have the potential to be applicable to several types of aerospace related scenarios. The most popular military application would be the missile interception problem which is traditionally solved using a

final-time-fixed approach. Other examples include minimum-time orbit transfer for spacecraft rendezvous and asset allocation for a team of UAVs tasked with tracking multiple targets. Because of the potential implications behavior-learning could have on minimum-time solutions to these types of scenarios, the simple case still has merit. This chapter will present an academic minimum-time problem based on the principles proposed in Ch. V.

V.A. Model

Consider an agent which undergoes rectilinear motion and is influenced by acceleration-level control,

$$\ddot{x}_i = u_i \,. \tag{5.1}$$

If the state vector is defined as

$$\boldsymbol{z}_i = [x,\, \dot{x}] \;, \tag{5.2}$$

then the vector-matrix form of the agent's equations of motion can be written as

$$\dot{\boldsymbol{z}} = A\boldsymbol{z}_i + Bu_i \,, \tag{5.3}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \tag{5.4}$$

If a pursuing and evading agent both behave according to Eqn. 5.3 and the relative state vector is defined as

$$\boldsymbol{z} = \boldsymbol{z}_p - \boldsymbol{z}_e \,, \tag{5.5}$$

then the relative model can be written as

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + Bu_p - Bu_e \,. \tag{5.6}$$

V.B. Minimum-Time Pursuit-Evasion

The final-time-free, zero-sum pursuit-evasion game is defined by the performance index

$$J = \int_{t_0}^{t_f} \mathrm{dt} \,, \tag{5.7}$$

and subject to the dynamic constraint given by Eqn. 5.6. The initial conditions are given by

$$\boldsymbol{z}_0 = \begin{bmatrix} z_{1_0} \\ \\ z_{2_0} \end{bmatrix} \,. \tag{5.8}$$

and the final state constraints are specified as

$$\boldsymbol{z}_f = \begin{bmatrix} z_{1_f} \\ \\ z_{2_f} \end{bmatrix} = \begin{bmatrix} 0 \\ \\ 0 \end{bmatrix} \,. \tag{5.9}$$

Additionally, the control input for each player is subject to the constraints

$$|u_p| \;\leq\; 1 \,, \tag{5.10}$$

$$|u_e| \;\leq\; K_e \,. \tag{5.11}$$

The goal is to drive the relative position and velocity between the two players to zero while satisfying the control constraints. The pursuer aims to do this in minimum-time while the pursuer wishes to maximize Eqn. 5.7 and therefore do this in maximum-time. In order for a solution to exist, K_e must satisfy $K_e < 1$, otherwise, the final-state-constraint cannot be satisfied. Note that this formulation is the same

as that posed in Section IV.A. The solution is given by Eqns. 4.7 and 4.8 while the costate equations are shown in Eqns. 4.9 and 4.10.

The costate λ_2 is a function of time that can be written as

$$\lambda_2 = \lambda_1 (t - t_0) + \lambda_{2_0} = \lambda_1 (t_f - t) + \lambda_{2_f} . \tag{5.12}$$

Note from Eqn. 5.12 that the switching function is a linear function of time and can therefore change sign once, at most.

Recall Isaacs' push for feedback solutions to pursuit-evasion games. By defining $w = u_p - u_e$, we can manipulate the switching function based on the sign of λ_2 into a feedback switching function. If

$$\dot{z}_2 = w , \tag{5.13}$$

then it follows

$$z_2 = w (t - t_0) + z_{2_0} = w (t - t_f) + z_{2_f} . \tag{5.14}$$

The terminal conditions are defined as $z_{1_f} = z_{2_f} = 0$ therefore Eqn. 5.14 reduces to

$$z_2 = w (t - t_f) . \tag{5.15}$$

Integration of z_2 yields

$$z_1 = \frac{w (t - t_f)^2}{2} = \frac{z_2^2}{2w} , \tag{5.16}$$

and solving for w in terms of z_1 and z_2 gives

$$w = \frac{z_2^2}{2z_1} . \tag{5.17}$$

Equation 5.17 can be rewritten as

$$\operatorname{sgn}(w) \operatorname{abs}(w) = \frac{z_2^2}{2z_1} . \tag{5.18}$$

Note that Eqn. 5.17 is consistent with the parabola equation given by Eqn. 4.16. The parabola plots shown in Figs. IV.1 - IV.3 can be used with the help of the parabola equation to form the feedback control solution in terms of z_1 and z_2. From the switching curve it can be concluded that above the curve, $w = w_{min}$, and below the curve, $w = w_{max}$, because $\text{sgn}(w_{min}) = -1$ and $\text{sgn}(w_{max}) = +1$. Together with Eqn. 5.18, the feedback solution is given by

$$\begin{aligned} w = w_{max} \quad & \text{if} \quad \left[z_2^2 \text{sgn}(z_2) < -2z_1 \text{abs}(w_{max})\right] \\ & \text{or} \quad \left[z_2^2 \text{sgn}(z_2) = -2z_1 \text{abs}(w_{max}),\ z_1 > 0\right], \\ w = w_{min} \quad & \text{if} \quad \left[z_2^2 \text{sgn}(z_2) > -2z_1 \text{abs}(w_{min})\right] \\ & \text{or} \quad \left[z_2^2 \text{sgn}(z_2) = -2z_1 \text{abs}(w_{min}),\ z_1 < 0\right], \end{aligned} \tag{5.19}$$

where

$$w_{max} = 1 - Ke, \tag{5.20}$$

$$w_{min} = -1 + Ke. \tag{5.21}$$

V.C. Behavior Learning

For the type of game presented, it was revealed in Chapter IV that behavior learning can be used by the pursuer to estimate the evader's selection of K_e. For this minimum-time pursuit evasion game, we will explore the effects of behavior learning with slightly different assumptions on the evader's game play. The evader will always play a complete, perfect, and certain information game. That is, the evader will know precisely the pursuer's control bounds, $u_p = \pm 1$. Additionally, the evader's measurements of the relative states z_1 and z_2 will free of measurement noise. There

are no uncertainties present in the evader's relative model. These characteristics of the evader will remain fixed for all simulations. For the imperfect information case, the evader may switch too early or too late which can cause oscillating behavior especially when it occurs near the true switch time. Although this would be a more realistic situation, it distracts the reader from the usefulness of behavior learning in the minimum-time case.

The evaluation of w given by Eqn. 5.19 dictates which value for u_p is selected by the pursuer. The value of K_e plays an important role in this determination which is done independently by the pursuer and the evader. The switching function is used so the pursuer can jump to the proper trajectory to the state-space origin at the correct time. If this switching function is not properly evaluated, then the pursuer will not be able to switch at the correct time to drive the relative states to the origin in minimum-time.

When the pursuer is subject to an incomplete, imperfect, certain information game, the states to be estimated become those given by Eqn. 4.19. The state equations are given by Eqn. 4.23 and the measurements of the relative states are summarized in Eqn. 4.24. Because the state equations and measurements are both linear functions of the states given by Eqn. 4.19, a linear estimator can be used to estimate the time varying parameters z_1 and z_2 along with the constant control bound of the evader given by K_e. In the event that the evader decides to play non-optimally, i.e. use $|u_e| < K_e$, then it is in the best interest of the pursuer to continuously monitor the estimate $x_3 = K_e$ so the switching function can always be properly evaluated. Therefore, the pursuer wishes to filter the states z_1 and z_2 while estimating the pa-

rameter K_e. For these reasons, a standard Kalman filter is used for behavior learning in this example.

V.D. Simulation

The initial conditions for each player were chosen to be

$$z_{p_0} = \begin{bmatrix} 2.5 \\ \\ 1 \end{bmatrix}, \qquad \text{and} \qquad z_{e_0} = \begin{bmatrix} 1 \\ \\ 0.5 \end{bmatrix}, \tag{5.22}$$

which produce the relative initial conditions

$$z_0 = \begin{bmatrix} 1.5 \\ \\ 0.5 \end{bmatrix}. \tag{5.23}$$

Three sets of simulation results are presented for the minimum-time behavior learning example. The complete information case is used as a baseline to show how the game should play out in the ideal scenario. The incomplete information case is used to show the consequences of the pursuer not being able to properly evaluate the switching function. Finally, an incomplete information example with the pursuer enabled with behavior learning is shown to illustrate the usefulness of these methods for the minimum-time example.

For the incomplete information examples, the pursuer is also subject to imperfect information. In incomplete information simulations, the pursuer's relative state measurements are subject to a zero-mean Gaussian noise distribution with a standard deviation of $\sigma = 0.001$.

V.D.1. Complete Information

The results for the complete information simulation are shown in Figs. V.1 - V.3. This example used a true value of $K_e = 0.2$ which was known by both players. The state space trajectory is shown in Fig. V.1 along with the optimal trajectories that were generated using $K_e = 0.2$. The game trajectory after the switching point which leads to the origin resides slightly lower than the actual optimal trajectory. In theory, this does not occur because the exact switching time can be computed. However, in practice, simulation are forced to rely on discrete time. As the simulation timestep approaches zero, the game trajectory converges on the optimal trajectory.

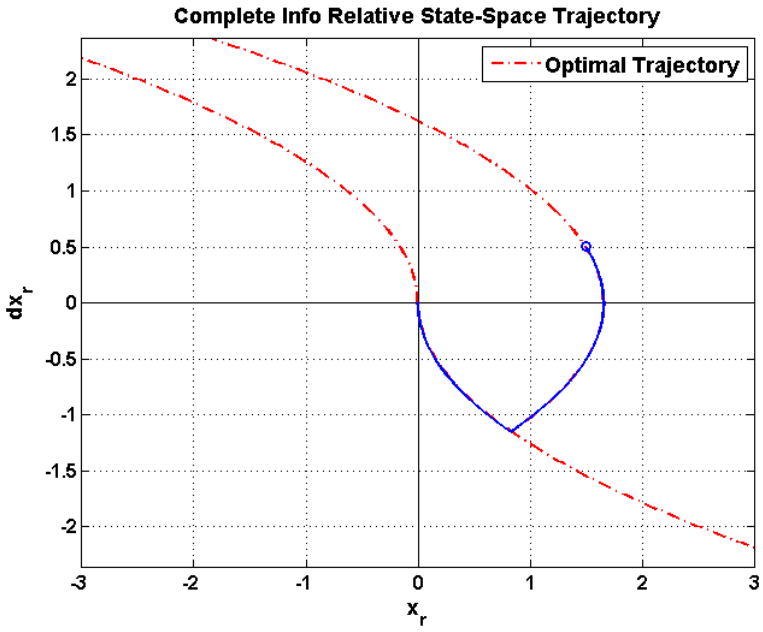

Figure V.1. Complete Information State Space Trajectory

Figure V.2 contains the pursuer, evader, and relative states. Note that at the final time $t_f = 3.503$ seconds, the relative states go to zero. The pursuer control, u_p, evader control, u_e, and total control, w, are illustrated in Fig. V.3. The pursuer and evader are both able to properly evaluate the switching function and therefore switch in unison.

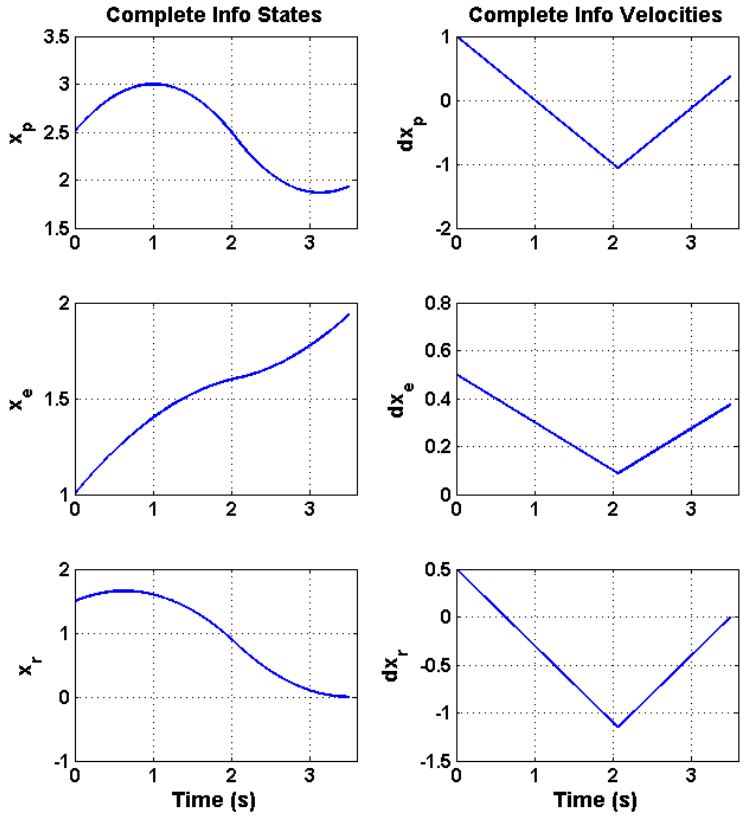

Figure V.2. Complete Information States

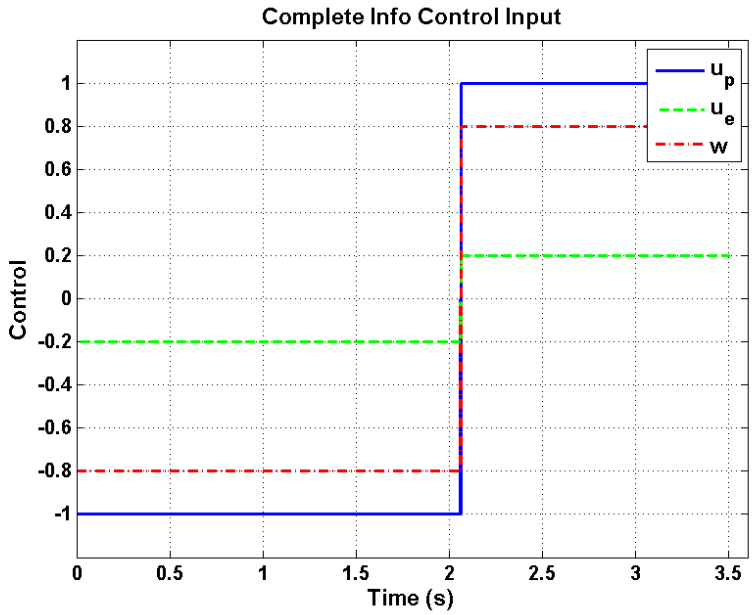

Figure V.3. Complete Information Control Input

V.D.2. Incomplete Information

For the incomplete information case, the pursuer continued to assume $K_e = 0.2$ but the evader's control constraint was actually $K_e = 0.4$. The results for these assumptions are found in Figs. V.4 - V.6. The state space trajectory is shown in Fig. V.4 along with the optimal trajectory assumed by the pursuer using $K_e = 0.2$ and the actual optimal trajectory given by $K_e = 0.4$. Figure V.5 contains the states given by the incomplete information case and the control input is shown in Fig. V.6.

Figures V.4 and V.6 show that the pursuer assumes it is on the wrong trajectory given by $K_e = 0.2$ and switches after the evader causing the trajectory to deviate from the optimal trajectory. Because the pursuer switches too late, the resulting trajectory is parallel to that of the optimal trajectory and the switching function must be evaluated again in order to get to the proper trajectory to the origin. This cycle

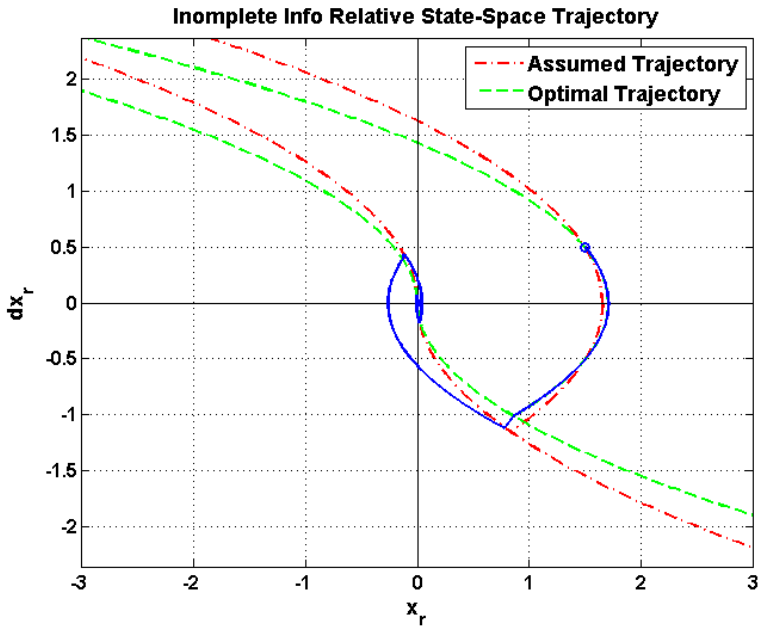

Figure V.4. Incomplete Information State Space Trajectory

continues until the system finally converges. The presence of incomplete information raises the final time from $t_f = 3.503$ to $t_f = 6.820$.

V.D.3. Incomplete Information with Behavior Learning

The incomplete information case was simulated again with the pursuer using a Kalman filter for behavior learning. These results are shown in Figs. V.7 - V.10 using the same selections for K_e as in the incomplete information simulation. With the use behavior learning, the evader is able to estimate the true value of K_e using its own assumption as the initial guess as illustrated in Fig. V.10. Even though $\dot{K}_e = 0$ is exact, a considerable amount of process noise was needed on that parameter to ensure a short transient time after the control switch. This is desirable in case the

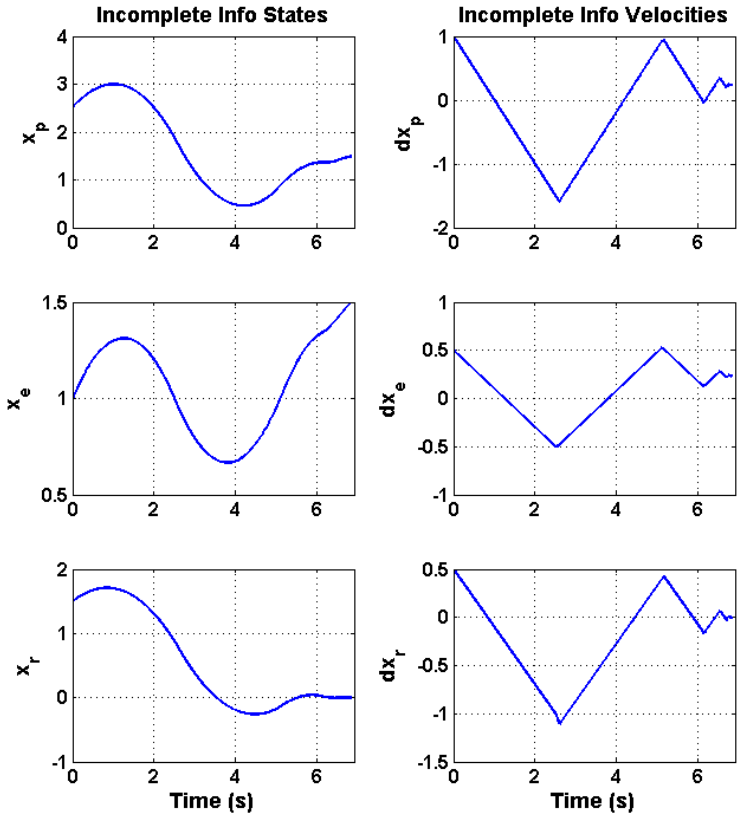

Figure V.5. Incomplete Information States

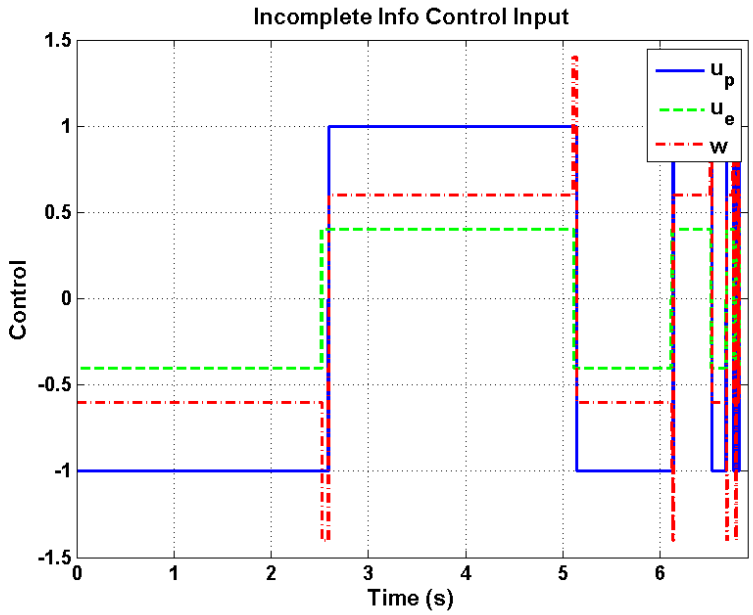

Figure V.6. Incomplete Information Control Input

evader decides to play with a non-optimal selection of K_e.

Figure V.7 confirms that the switching function was properly evaluated by both players at the optimal trajectory was taken. The final time with behavior learning implemented was $t_f = 4.209$. The discrepancy between this final time and that of the complete information case is because the complete information case uses a true K_e of 0.2 while the incomplete version uses $K_e = 0.4$. This was done to show the trajectory that the pursuer expected.

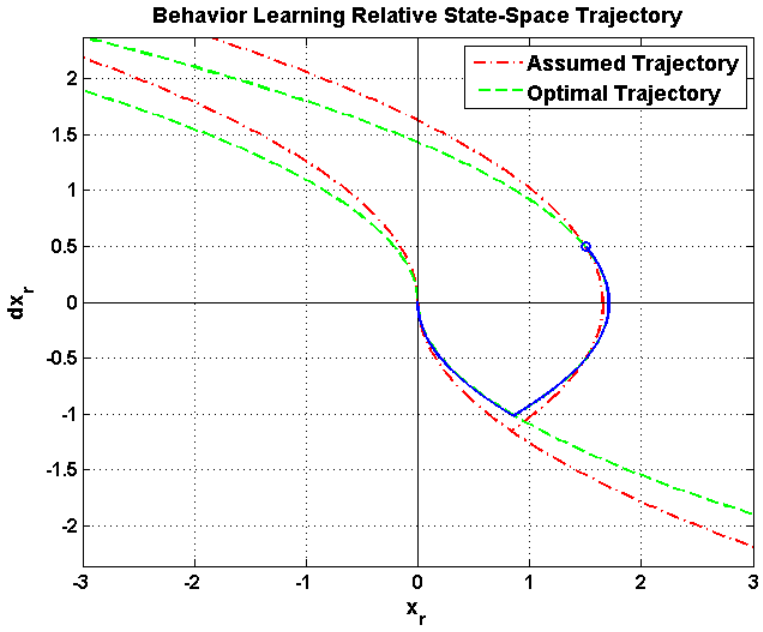

Figure V.7. Incomplete Information with Behavior Learning State Space Trajectory

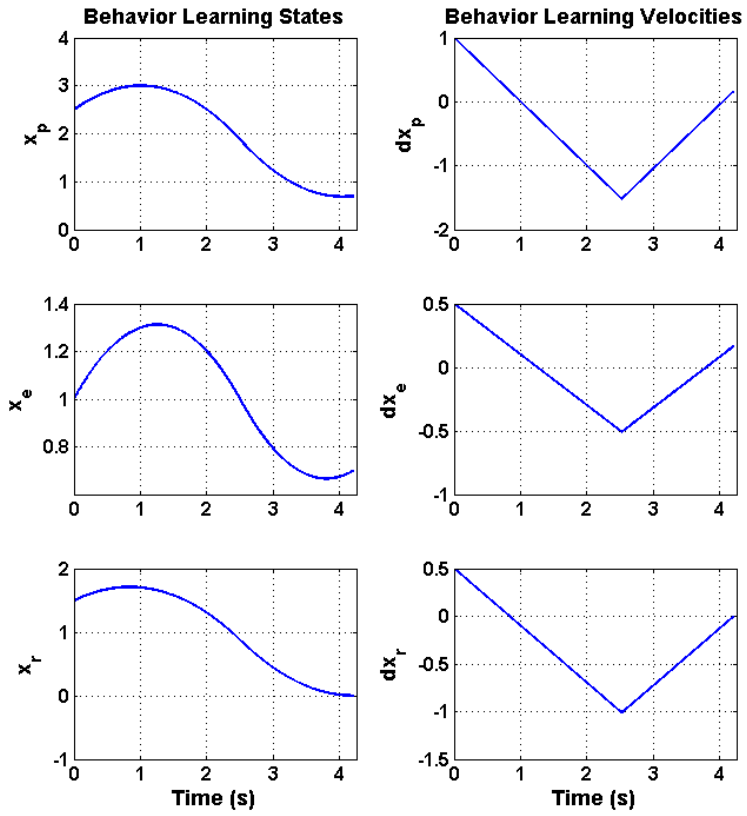

Figure V.8. Incomplete Information with Behavior Learning States

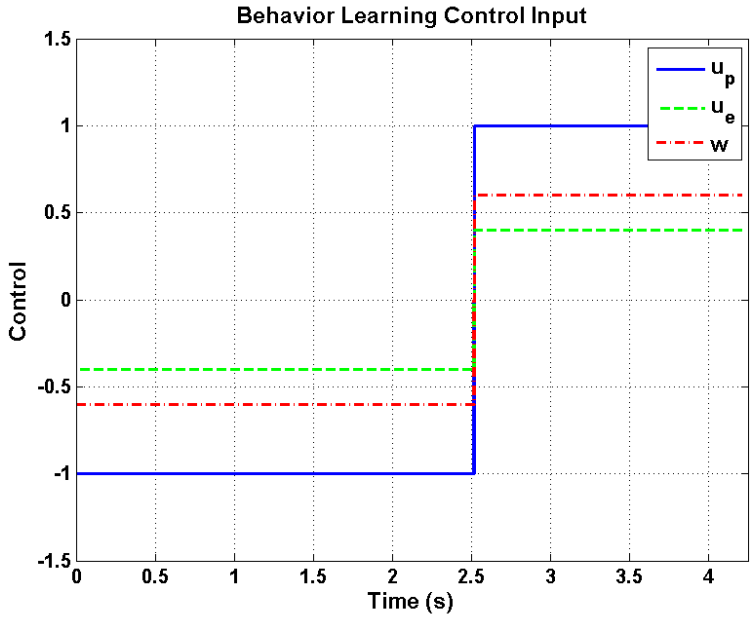

Figure V.9. Incomplete Information with Behavior Learning Control Input

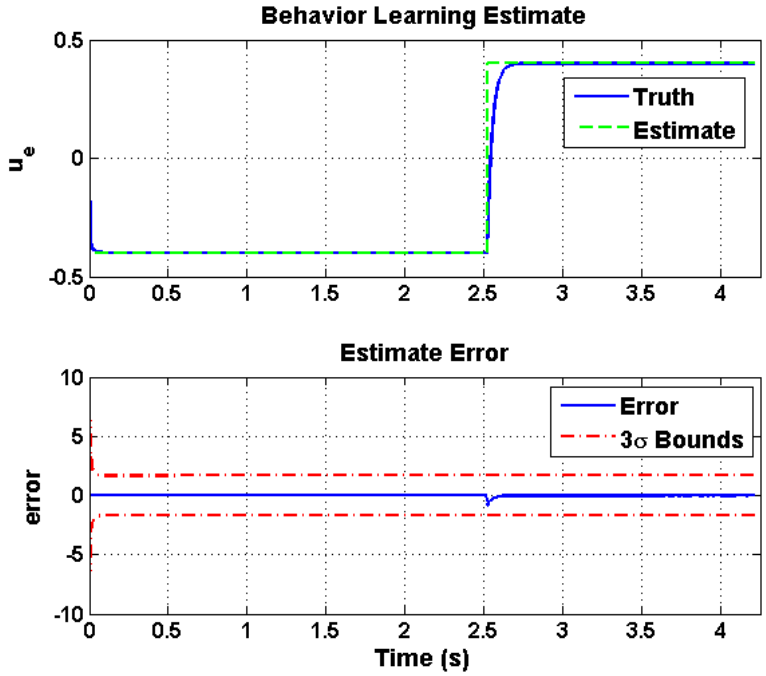

Figure V.10. Incomplete Information with Behavior Learning Estimate

V.E. Summary

A comparison of the complete, incomplete, and behavior learning minimum-time trajectories are shown in Fig. V.11. When a player incorrectly assumes their opponent's control input, it becomes impossible to properly evaluate the switching function and the evolution of the state space path is forced to take a sub-optimal trajectory. The process noise associated with the gain K_e in the behavior learning estimator was intentionally set high in order to accommodate the switching nature of the control. The final cost summary is shown in Table V.1. Although the

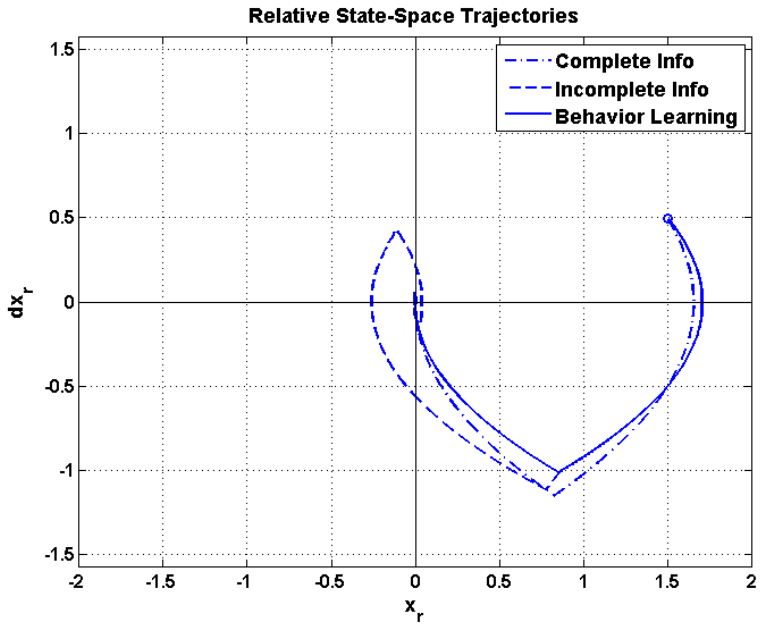

Figure V.11. Minimum-Time State Space Trajectories

minimum-time problem presented here is a purely academic example, it is apparent

Table V.1. Minimum-Time Game Cost Summary

Information Type	**Pursuer Cost**
Complete	3.503
Incomplete	6.820
Incomplete + BL	4.209

how behavior learning can play an important role in final-time-free PE games which are subject to incomplete information. In practice, additional issues arise such as missing the proper switching time due to the chosen discrete time step size. As the time step between measurements increases, the time between when the switch should occur and when the switch is properly evaluated increases. As the update rate increases, this issue becomes less prevalent.

One other concern includes improper evaluation of the switching function based on the imperfect nature of state measurements subject to noise. When zero-mean white noise is added to the measurements, a player could think the switch should occur then re-evaluate such that the switch should not occur at the next time step. As the magnitude of the additive noise increases, so does the probability that this phenomenon will occur. This issue has a cascading effect when behavior learning is enabled because a player estimates their opponent's behavior based on the relative states which are driven by the control input. When the opponent continues to incorrectly evaluate the switching function, it could have negative effects on the player enabled with behavior learning. Because filtering of the relative states is a

byproduct of the behavior learning framework presented, a single player can reduce the possibility of this happening for their own switching function computation by implementing a behavior learning filter.

CHAPTER VI

DYNAMIC INVERSION

As outlined in the previous chapters, feedback solutions are critical for PE scenarios. The need for closed-loop control has shaped the discussion around games that are linear-quadratic in nature. Before continuing, an important detail must be addressed concerning real-world dynamic systems. In the aerospace industry, dynamic systems are most commonly nonlinear which can be problematic when approaching a PE game which requires a feedback solution. Therefore, a reliable method is needed to help transform common nonlinear systems, including Euler's rotational equations of motion or a those describing a vehicle in flight, into their linear counterpart such that they fit into the framework of a linear-quadratic pursuit-evasion game.

Over the years, techniques have been developed to aid in the control of vehicles whose motion is described by nonlinear differential equations. The most common method involves linearization by taking partial derivatives of the nonlinear equations about equilibrium points. If a vehicle can stay within a certain motion envelope, these techniques can hold true. Linearization can be performed both analytically [31] or experimentally [32]. The drawback to this method, however, is that the states that are of interest in a pursuit-evasion game are often not the same states that make up the linearized model. For example, the linearized model for an aircraft in flight is based upon the Euler angles that make up the aircraft's attitude, and for good reason. The stability of an air vehicle is very important in order for it to maintain desired, stable, flight characteristics. However, most often the position of a vehicle

is of interest in a PE scenario.

Another tool used to provide a linear transformation for a naturally nonlinear system is dynamic inversion [33]. The method of dynamic inversion allows the engineer to select a desired, linear system response and subsequently compute the necessary control input for the nonlinear system to achieve the desired response. This method has been shown to be useful in the control of highly maneuverable vehicles [34], its stability and robustness verified [35–37], and it has even been used on flight tests of advanced military applications [5].

VI.A. Model

The method of dynamic inversion can be very useful when applied to generalized nonlinear equations of motion. Consider the nonlinear differential equations of motion for a single player which take on the form

$$\boldsymbol{q} = \left[\boldsymbol{r}^T,\ \boldsymbol{s}^T\right]^T , \tag{6.1}$$

$$\dot{\boldsymbol{r}} = H\left(\boldsymbol{r}\right)\boldsymbol{s} , \tag{6.2}$$

$$\dot{\boldsymbol{s}} = \boldsymbol{f}\left(\boldsymbol{r},\boldsymbol{s}\right) + G\left(\boldsymbol{s}\right)\boldsymbol{v} , \tag{6.3}$$

where $\boldsymbol{r},\boldsymbol{s},\boldsymbol{f} \in \mathbb{R}^n$, $\boldsymbol{v} \in \mathbb{R}^m$, $H \in \mathbb{R}^{n\times n}$, and $G \in \mathbb{R}^{n\times m}$.

Vector $\boldsymbol{q}$ is the state vector with $\boldsymbol{r}$ and $\boldsymbol{s}$ representing the position and velocity level variables, respectively. Vector $\boldsymbol{v}$ signifies the control input. The kinematics and dynamics are defined by Eqn. 6.2 and Eqn. 6.3, respectively, while vector and matrix functions $\boldsymbol{f}(\boldsymbol{r},\boldsymbol{s})$, $G(\boldsymbol{s})$, and $H(\boldsymbol{r})$ may or may not be nonlinear. This particular class of systems is affine in the controls meaning the the control input appears linearly in the nonlinear state differential equations. This form is common among aerospace sys-

tems and therefore the following discussion will be limited to control-affine nonlinear systems.

Nonlinear optimal control problems, including solutions to PE games, become significantly more difficult when the optimal solution must adhere to a nonlinear dynamic constraint. Analytical, closed-form feedback solutions to the optimal control problem are desired. Therefore, it is of particular interest to find useful transformations to map the relationships found in Eqns. 6.2 and 6.3 to suitable linear versions. To accomplish this, a two-step dynamic inversion process is used.

VI.B. Method

Dynamic inversion is applicable to n^{th}-order nonlinear systems that are control-affine like those found in Eqn. 6.3 [5]. Consider a desired linear dynamic model, defined as $\dot{\boldsymbol{s}}_{des}$, which can be prescribed later for whichever system we choose. To force the nonlinear system to follow the dynamics of the desired linear model, set the right hand side of Eqn. 6.3 equal to the desired dynamic model and solve for the control input vector.

$$\begin{aligned} \dot{\boldsymbol{s}}_{des} &= \boldsymbol{f} + G\boldsymbol{v}\,, \\ \boldsymbol{v} &= G^{-1}\left[\dot{\boldsymbol{s}}_{des} - \boldsymbol{f}\right]\,. \end{aligned} \tag{6.4}$$

Equation 6.4 is used to compute the control input vector $\boldsymbol{v}$ required to make the nonlinear dynamics in Eqn. 6.3 behave as those prescribed by $\dot{\boldsymbol{s}}_{des}$. Note that the matrix function G must be invertible. If G is square then $m = n$ and the number of control inputs for the original system are equal to the number of position coordinates. Inversion is possible if it is of full rank. A minimum norm solution can also be used

if $m > n$ which is representative of an over-actuated system. This makes sense because if we wish to drive the relative position states to zero, we need at least that many control inputs to have enough control authority to maneuver the system. Still, the possibility for nonlinear kinematics exists as given by Eqn. 6.2. The concept of dynamic inversion can be applied a second time to this system.

Again, consider the desired dynamics $\boldsymbol{w}$ that are yet to be specified. By setting this desired behavior equal to the time derivative of the right hand side of Eqn. 6.2 and substituting the desired $\dot{\boldsymbol{s}}_{des}$, it is possible to compute the consistent $\dot{\boldsymbol{s}}_{des}$ based on the specified $\boldsymbol{w}$.

$$\begin{aligned} \boldsymbol{w} &= \ddot{\boldsymbol{r}}\,, \\ \boldsymbol{w} &= \dot{H}\boldsymbol{s} + H\dot{\boldsymbol{s}}\,, \\ \boldsymbol{w} &= \dot{H}\boldsymbol{s} + H\dot{\boldsymbol{s}}_{des}\,, \\ \dot{\boldsymbol{s}}_{des} &= H^{-1}\left[\boldsymbol{w} - \dot{H}\boldsymbol{s}\right]\,. \end{aligned} \tag{6.5}$$

Applying two-step dynamic inversion to the specific class of systems described by Eqns. 6.2 and 6.3 yields a convenient double integrator problem. First, $\boldsymbol{w}$ is chosen and is used for the pursuit-evasion optimal control solution. Then, Eqn. 6.5 is used to compute the consistent $\dot{\boldsymbol{s}}_{des}$ for the double integrator framework. Finally, Eqn. 6.4 is used to compute the actual input to the original system. Each variable is evaluated using the current states at the current timestep. Note that the actual system still behaves according to Eqn. 6.3 which is used to simulate the evolution of motion in computer-based applications. This two-step process rids the system of the complications associated with nonlinear kinematics and allows for a double-integrator form for the equations of motion. It is important to note that for the

common kinematic relationship $\dot{\boldsymbol{r}} = \boldsymbol{s}$, H takes on the form of an identity matrix then $\dot{\boldsymbol{s}}_{des}$ is simply equal to $\boldsymbol{w}$.

Upon implementation, dynamic inversion creates a main outer control loop and a dynamic inversion inner control loop which usual runs at a faster rate. The main outer control is computed using the desired dynamics defined by $\boldsymbol{w}$, like that from a pursuit-evasion game, then those results are used to compute the necessary input $\boldsymbol{v}$ for the inner control loop. Dynamic inversion has been proven to be very useful when the execution rate is fast enough to deal with the system nonlinearities [5]. It is not uncommon for the inner control loop to run at a rate an order of magnitude greater than the outer control loop. The desired dynamics, $\boldsymbol{w}$, would remain constant during the extra computational steps of the inner control loop and updated at its own rate as necessary.

The dynamic inversion process is used to provide a linear transformation for a single vehicle or player. To form the relative equations of motion needed for the PE dynamic constraint, it is necessary to perform two-step dynamic inversion for each system.

VI.C. Relative Model

If the original system for Player i is nonlinear, the desired linear representation could be selected as

$$\boldsymbol{w}_i = \boldsymbol{u}_i = \begin{bmatrix} u_{i_1} \\ \\ u_{i_2} \end{bmatrix}, \tag{6.6}$$

which represents a system with acceleration level control under the influence of no external forces.

For a point mass system moving in the horizontal plane, the linear representation of the state vector and its time derivative are then written as

$$\boldsymbol{z}_i = [x_i, y_i, \dot{x}_i, \dot{y}_i]^T = [z_{i_1}, z_{i_2}, z_{i_3}, z_{i_4}]^T \,, \tag{6.7}$$

$$\dot{\boldsymbol{z}}_i = [z_{i_3}, z_{i_4}, u_{i_1}, u_{i_2}]^T \,. \tag{6.8}$$

In vector-matrix form, Eqn. 6.8 becomes

$$\dot{\boldsymbol{z}}_i = A\boldsymbol{z}_i + B\boldsymbol{u}_i \,, \tag{6.9}$$

where

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \,, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \,. \tag{6.10}$$

When the pursuer and evader are both modeled by Eqn. 6.9 and therefore have the same system matrices A and B, the relative equations of motion can be formed as

$$\boldsymbol{z}_r = \boldsymbol{z}_p - \boldsymbol{z}_e \,, \tag{6.11}$$

$$\dot{\boldsymbol{z}}_r = \dot{\boldsymbol{z}}_p - \dot{\boldsymbol{z}}_e = A\boldsymbol{z}_p + B\boldsymbol{u}_p - A\boldsymbol{z}_e - B\boldsymbol{u}_e \,, \tag{6.12}$$

$$\dot{\boldsymbol{z}}_r = A\left[\boldsymbol{z}_p - \boldsymbol{z}_e\right] + B\boldsymbol{u}_p - B\boldsymbol{u}_e \,, \tag{6.13}$$

$$\dot{\boldsymbol{z}}_r = A\boldsymbol{z}_r + B\boldsymbol{u}_p - B\boldsymbol{u}_e \,, \tag{6.14}$$

where subscripts p and e denote states and control inputs of the pursuer and evader, respectively. The relative differential equations found in Eqn. 6.14 define the dynamic constraint for a PE game between Player p and Player e. Because dynamic inversion

is used to produce a linear constraint, familiar feedback solutions to linear optimal control problems such as the linear-quadratic regulator can be applied.

Note that it is not always the case that the relative equations of motion take on the exact form shown in Eqn. 6.14. For example, a spacecraft reorientation PE game will be studied later which takes into account the relative attitude and attitude rates. Special attention must be paid to systems whose relative states cannot be computed using a simple difference relation like that in Eqn. 6.11. The relative attitude and the associated rates must be computed consistently using the necessary form of the attitude influence matrix [38].

VI.D. Summary

Although the robustness of dynamic inversion has been verified in certain applications [36], it must be used with caution and its robustness examined on a case-by-case basis. Depending on the selection of the desired linear dynamics and the feasibility of those dynamics by the true nonlinear system, there exists the possibility that what is being requested of the system is unobtainable. In the most extreme cases, the choice for the desired dynamics may produce undesirable system characteristics which may include control saturation or system instability.

Control saturation is brought about by large magnitudes being required in order for the nonlinear system to behave like the desired system. Control saturation can become catastrophic for nonlinear systems that are already inherently unstable. These large magnitudes may also produce system oscillations and can lead to system instability.

It is essential to cautiously select the desired linear response $\boldsymbol{w}$. Most often, selections which yield a damped linear response perform much better than those selection which do not. Nevertheless, some agility must be sacrificed when selecting a particular stable response. For example, instead of a model matrix A, $A*$ could be selected for a more obtainable desired system.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k & 0 & -c & 0 \\ 0 & -k & 0 & -c \end{bmatrix}, \tag{6.15}$$

where k and c are positive.

The method of two-step dynamic inversion for nonlinear systems presented can be extremely useful when applied to pursuit-evasion games if its application is exercised with caution. The selection of $\boldsymbol{w}$ is critical for complex systems in which stability could become an issue.

CHAPTER VII

APPLICATIONS: SPACECRAFT PROXIMITY OPERATIONS

Recently, automated spacecraft proximity operations has become a major area of interest among government and private entities. Several factors have been the major driving force behind these research interests including automated removal of space debris, servicing of damaged spacecraft, re-purposing of out-of-date satellites by means of on-orbit disassembly and assembly, resupply of the International Space Station by private companies, and reconnaissance missions involving uncooperative spacecraft. Many of these factors can be adapted to fit the framework of a pursuit-evasion scenario. Spacecraft differential games have been developed [39] and solutions to the optimal guidance laws presented [40]. However, these pursuit-evasion results are limited to final-time-fixed and restricted to the complete information case.

Consider a space-based reconnaissance mission involving two spacecraft which are modeled as rigid bodies and whose attitude is defined through three degrees-of-freedom. The spacecraft are non-cooperative and the objective for the pursuer spacecraft is to match the attitude and angular velocities of the second evader spacecraft. That is, the pursuer spacecraft wants to drive the relative attitude error to zero while simultaneously driving the relative angular velocities to zero. The pursuer wishes to accomplish this task for an undetermined amount of time making a game of this nature best suited for an infinite-horizon PE scenario.

VII.A. Model

First, consider the attitude kinematics and dynamics of each vehicle individually. The attitude can be represented by several choices of attitude coordinates including Euler angles, Euler parameters, the classic Rodrigues parameters (CRPs), modified Rodrigues parameters (MRPs), or the principle vector and angle. The selection of attitude coordinates is important because they can make the problem more or less convenient with the implementation of dynamic inversion.

For this problem, the CRPs are used for two reasons. The first is that they are a minimum attitude representation meaning three parameters are used to define the three degrees-of-freedom associated with the attitude. This is important because the attitude influence matrix which relates the angular velocity to the attitude rates is square [38]. A square matrix is necessary for a valid implementation of dynamic inversion. It is possible for a four parameter set to be chosen such as the Euler parameters but this requires the kinematic constraint to be appended to the attitude influence matrix. For these reasons, a minimum attitude set is desired.

The second reason the CRPs are chosen in favor of the MRPs is that minimum attitude sets are susceptible to singularities. The MRPs have an orientation singularity when the principal angle $\phi = \pm 2\pi$. This is undesirable because once the relative attitude is formed, this corresponds to when the pursuer matches the evader's attitude exactly, which is the goal. Because the game is defined as infinite-horizon, the pursuer intends to hold the desired relative attitude at zero infinitely. The CRPs contain an orientation singularity when $\phi = \pm\pi$. When considering the relative attitude, This singularity corresponds to when the two vehicles have the absolute most

attitude error. This can be avoided if the pursuer has more control authority than the evader and the initial conditions are defined such that the relative $\phi \neq \pm\pi$.

Another solution to these issues would be to use a shadow set of coordinates such as the shadow MRPs. These attitude coordinates always avoid the orientation singularity by computing the attitude one of two ways. The drawback of this selection is it would introduce inconsistencies in the attitude description and could negatively affect the feedback solutions to the PE game.

By defining the attitude and angular velocity vectors for player i as

$$\boldsymbol{q}_i = \left[q_{i_1},\, q_{i_2},\, q_{i_3}\right]^T , \tag{7.1}$$

$$\boldsymbol{\omega}_i = \left[\omega_{i_1},\, \omega_{i_2},\, \omega_{i_3}\right]^T , \tag{7.2}$$

the attitude kinematics for the CRPs are defined for a single vehicle as

$$\dot{\boldsymbol{q}}_i = H_i \boldsymbol{\omega}_i \,, \tag{7.3}$$

where $\boldsymbol{q}_i$ represents the CRPs and $\boldsymbol{\omega}_i$ represents the body angular velocities with respect to the inertial reference frame, resolved in the body-fixed reference frame. The attitude influence matrix, H_i, is a function of the attitude coordinates and takes on the form

$$H_i = \frac{1}{2}\begin{bmatrix} 1+q_{i_1}^2 & q_{i_1}q_{i_2}-q_{i_3} & q_{i_1}q_{i_3}+q_{i_2} \\ q_{i_1}q_{i_2}+q_{i_3} & 1+q_{i_2}^2 & q_{i_2}q_{i_3}-q_{i_1} \\ q_{i_1}q_{i_3}-q_{i_2} & q_{i_2}q_{i_3}+q_{i_1} & 1+q_{i_3}^2 \end{bmatrix} . \tag{7.4}$$

If the moment of inertia tensor and the torque input vector are of the form

$$I_i = \begin{bmatrix} I_{i_1} & 0 & 0 \\ 0 & I_{i_2} & 0 \\ 0 & 0 & I_{i_3} \end{bmatrix} , \qquad \text{and} \qquad \boldsymbol{\ell}_i = \begin{bmatrix} \ell_{i_1} \\ \ell_{i_2} \\ \ell_{i_3} \end{bmatrix} , \tag{7.5}$$

respectively, then Euler's rotational equations of motion for a rotating rigid body provide the attitude dynamics which are given by

$$\dot{\omega}_{i_1} = \frac{1}{I_{i_1}} \left(I_{i_2} - I_{i_3}\right) \omega_{i_2} \omega_{i_3} + \frac{1}{I_{i_1}} \ell_{i_1} \,, \tag{7.6}$$

$$\dot{\omega}_{i_2} = \frac{1}{I_{i_2}} \left(I_{i_3} - I_{i_1}\right) \omega_{i_1} \omega_{i_3} + \frac{1}{I_{i_2}} \ell_{i_2} \,, \tag{7.7}$$

$$\dot{\omega}_{i_3} = \frac{1}{I_{i_3}} \left(I_{i_1} - I_{i_2}\right) \omega_{i_1} \omega_{i_2} + \frac{1}{I_{i_3}} \ell_{i_3} \,. \tag{7.8}$$

Both the attitude kinematics and rotational dynamics are nonlinear for a single vehicle. Therefore, we wish to impose dynamic inversion in an effort to put the equations of motion into a form that can be used for the implementation of a pursuit-evasion game with feedback solutions and behavior learning elements. The dynamics can be rewritten as

$$\dot{\boldsymbol{\omega}}_i = \boldsymbol{f}_i + G_i \boldsymbol{v}_i \,, \tag{7.9}$$

where

$$\boldsymbol{f}_i = \begin{bmatrix} \frac{1}{I_{i_1}} \left(I_{i_2} - I_{i_3}\right) \omega_{i_2} \omega_{i_3} \\ \frac{1}{I_{i_2}} \left(I_{i_3} - I_{i_1}\right) \omega_{i_1} \omega_{i_3} \\ \frac{1}{I_{i_3}} \left(I_{i_1} - I_{i_2}\right) \omega_{i_1} \omega_{i_2} \end{bmatrix} \,, \qquad G_i = \begin{bmatrix} \frac{1}{I_{i_1}} & 0 & 0 \\ 0 & \frac{1}{I_{i_2}} & 0 \\ 0 & 0 & \frac{1}{I_{i_3}} \end{bmatrix} \,, \tag{7.10}$$

and

$$\boldsymbol{v}_i = \left[\ell_{i_1},\, \ell_{i_2},\, \ell_{i_3}\right]^T \,. \tag{7.11}$$

Equations 7.3 and 7.9 now take on the same form as the model used for dynamic inversion in Eqns. 6.2 and 6.3. By applying two-step dynamic inversion, the necessary consistent angular velocity vector, $\dot{\boldsymbol{\omega}}_{i_c}$, is defined as

$$\dot{\boldsymbol{\omega}}_{i_c} = H_i^{-1} \left[\ddot{\boldsymbol{q}}_{i_{desired}} - \dot{H}_i \boldsymbol{\omega}_i\right] \,, \tag{7.12}$$

and the true control input can be computed using

$$\boldsymbol{v}_i = G_i^{-1}\left[\dot{\boldsymbol{\omega}}_{i_c} - \boldsymbol{f}_i\right] . \tag{7.13}$$

In Eqns. 7.12 and 7.13, $\boldsymbol{\omega}_i$ and $\boldsymbol{f}_i$ are evaluated using the current angular velocity, G_i is constant, H_i is evaluated using the current attitude, and $\dot{H}_i$ is evaluated at the current attitude and attitude rates using

$$\dot{H}_i = \frac{1}{2}\left(\dot{q}_{i_1}\begin{bmatrix} 2q_{i_1} & q_{i_2} & q_{i_3} \\ q_{i_2} & 0 & -1 \\ q_{i_3} & 1 & 0 \end{bmatrix} + \dot{q}_{i_2}\begin{bmatrix} 0 & q_{i_1} & 1 \\ q_{i_1} & 2q_{i_2} & q_{i_3} \\ -1 & q_{i_3} & 0 \end{bmatrix} + \dot{q}_{i_3}\begin{bmatrix} 0 & -1 & q_{i_1} \\ 1 & 0 & q_{i_2} \\ q_{i_1} & q_{i_2} & 2q_{i_3} \end{bmatrix} \right) . \tag{7.14}$$

By imposing this two-step process to remove the kinematic and dynamic nonlinearities, we are free to choose $\ddot{\boldsymbol{q}}_{i_{desired}}$ as we see fit. Note that through this dynamic inversion process, the attitude rates are used in place of the angular velocities and the desired dynamics are imposed on the the coordinates directly. This is done in the pursuit-evasion control computation and the dynamic inversion relations given by Eqns. 7.12 and 7.13 are used to compute the actual required controls for each system based on the desired attitude dynamics.

Direct control of each attitude coordinate is possible by defining

$$\ddot{\boldsymbol{q}}_{i_{desired}} = \boldsymbol{u}_i = \left[u_{i_1},\, u_{i_2},\, u_{i_3}\right]^T . \tag{7.15}$$

If player i's state vector is

$$\boldsymbol{z}_i = \left[q_{i_1},\, q_{i_2},\, q_{i_3},\, \dot{q}_{i_1},\, \dot{q}_{i_2},\, \dot{q}_{i_3}\right]^T , \tag{7.16}$$

then the vector-matrix form for single player EoMs become

$$\dot{\boldsymbol{z}}_i = A\boldsymbol{z}_i + B\boldsymbol{u}_i , \tag{7.17}$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{7.18}$$

Two-step dynamic inversion is imposed on both players to allow for a linear formulation of the attitude dynamics. The attitude and dynamics of each individual vehicle is defined with respect to the inertial reference frame. For the PE game, the relative attitude equations must be formed. During implementation, special consideration must be given to the relative attitude when also keeping track of the inertial attitude of each player.

Even with the implementation of dynamic inversion on each vehicle individually, the relative attitude kinematics and dynamics still exhibit nonlinearities due to the relationship between the body angular velocities and the attitude coordinate rates. To push the limits of behavior learning, an invalid assumption will be made pertaining to the relative attitude model. It will be assumed that the relative attitude dynamics behave linearly, which is not the case. The goal here is to show that even if the player's assumed model of the relative system is wrong, behavior learning can still be effective by producing some solution that can be used to model the system. If a player can converge on a model for their opponent's behavior, even if is not necessarily the correct model, a player should have the ability to perform better than by simply employing a zero-sum safe strategy.

The relative attitude coordinates of the pursuer with respect to the evader are defined as

$$\boldsymbol{q} = [q_1,\, q_2,\, q_3]^T \,. \tag{7.19}$$

The invalid assumption is that the acceleration of the attitude coordinates can be written linearly in terms of the pursuer and evader inputs as

$$\ddot{\boldsymbol{q}} = \boldsymbol{u}_p - \boldsymbol{u}_e \,. \tag{7.20}$$

Following this assumption, writing the relative state vector as

$$\boldsymbol{z} = [q_1,\, q_2,\, q_3,\, \dot{q}_1,\, \dot{q}_2,\, \dot{q}_3]^T \,, \tag{7.21}$$

the relative attitude dynamics can be expressed as

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u}_p - B\boldsymbol{u}_e \,, \tag{7.22}$$

where A and B are defined in Eqn. 7.18.

VII.B. Pursuit-Evasion Game

The infinite-horizon spacecraft reorientation pursuit-evasion game is defined by the zero-sum performance index

$$J_{SR} = \frac{1}{2}\int_{t_0}^{\infty} \left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p - \boldsymbol{u}_e^T R_e \boldsymbol{u}_e\right) \mathrm{dt} \,, \tag{7.23}$$

subject to the linear dynamic constraint defined by Eqn. 7.22. The optimal solutions are given by

$$\boldsymbol{u}_p = -R_p^{-1} B^T S \boldsymbol{z} \,, \tag{7.24}$$

$$\boldsymbol{u}_e = -R_e^{-1} B^T S \boldsymbol{z} \,, \tag{7.25}$$

where S is the solution to the ARE

$$0 = Q + A^T S + SA + SB\left(R_e^{-1} - R_p^{-1}\right) B^T S\,. \tag{7.26}$$

VII.C. Behavior Learning

Behavior learning will attempt to estimate the evader's Kalman gain which was shown to be constant for the infinite-horizon case. This gain is a 3×6 matrix with at most 18 independent gains to estimate. The evader's Kalman gain is defined by

$$K_e = R_e^{-1} B^T S\,. \tag{7.27}$$

With the form of A and B known and given by Eqn. 7.18, and the reasonable assumptions that:

- the evader implements a zero-sum safe strategy,
- the evader is not capable of bahavior learning,
- the evader weighs each of the relative coordinates equally,
- the evader weighs each of the relative coordinate rates equally,
- the evader weighs each of the control inputs equally, and
- the evader does not weigh any cross-coupling terms

the form of K_e can be reduced to

$$K_e = \begin{bmatrix} k_1 & 0 & 0 & k_2 & 0 & 0 \\ 0 & k_1 & 0 & 0 & k_2 & 0 \\ 0 & 0 & k_1 & 0 & 0 & k_2 \end{bmatrix}. \tag{7.28}$$

The behavior learning algorithm becomes an estimator for the relative attitude coordinates and coordinate rates given by $\boldsymbol{z}$, and the two independent elements of K_e. These states are summarized by the estimate vector

$$\boldsymbol{x} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} , \tag{7.29}$$

where states $x_1 - x_6$ are time-varying and whose state equations are given by

$$\dot{\boldsymbol{z}} = (A + BK_e)\, \boldsymbol{z} + B\boldsymbol{u_p}\,. \tag{7.30}$$

The state equations needed for the nonlinear estimator are summarized by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) , \tag{7.31}$$

where

$$\boldsymbol{f} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ x_1 x_7 + x_4 x_8 + u_{p_1} \\ x_2 x_7 + x_5 x_8 + u_{p_2} \\ x_3 x_7 + x_6 x_8 + u_{p_3} \\ 0 \\ 0 \end{bmatrix}, \tag{7.32}$$

and $\boldsymbol{u_p} = [u_{p_1}, u_{p_2}, u_{p_3}]^T$. The measurements available are the relative states defined by

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{h}(\boldsymbol{x}_k) = [x_1\,, x_2\,, x_3\,, x_4\,, x_5\,, x_6]^T\ . \tag{7.33}$$

Equations 7.31 and 7.33 are in the standard form needed for a nonlinear filter. Any of several nonlinear estimation techniques can be used to filter the relative states and estimate the strategy parameters k_1 and k_2. These estimates can be continuously monitored and the pursuer can then employ a one-sided optimal control solution.

The one-sided optimal control solution is given by Eqns. 2.16 and 2.18. If the pursuer notices a significant change in the evader's gains k_1 or k_2, Eqn 3.27 must be used to compute a new solution for S before this information can be accounted for.

VII.D. Simulation

To develop a baseline case for comparison purposes, a complete information case is simulated along with incomplete information and behavior learning enabled cases.

The complete information gain selections are summarized by

$$Q_e = \begin{bmatrix} 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{7.34}$$

$$R_{p_e} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R_{e_e} = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 15 \end{bmatrix}. \tag{7.35}$$

For the complete information case, each player assumes a zero-sum strategy using the gains defined in Eqns. 7.34 and 7.35.

The initial conditions for all cases were chosen to be

$$\boldsymbol{q}_{p_0} = [0.07,\, 0.02,\, 0.1]^T, \tag{7.36}$$

$$\boldsymbol{q}_{e_0} = [0.25,\, 0.5,\, 0.33]^T, \tag{7.37}$$

with the initial angular velocities of

$$\boldsymbol{w}_{p_0} = [-0.05,\, 0,\, 0.07]^T \text{ rad/s}, \tag{7.38}$$

$$\boldsymbol{w}_{e_0} = [0.15,\, -0.1,\, 0]^T \text{ rad/s}. \tag{7.39}$$

These initial conditions produce a relative attitude defined by the CRP vector

$$\boldsymbol{q}_{r_0} = [-0.1201,\, -0.2593,\, -0.1533]^T, \tag{7.40}$$

and an initial relative angular velocity of

$$\boldsymbol{\omega}_{r_0} = [-0.20,\, 0.10,\, 0.07]^T \text{ rad/s}. \tag{7.41}$$

The spacecraft moment of inertia tensors define equivalent near-axisymmetric rigid bodies.

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2.25 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \tag{7.42}$$

Imperfect information dictates that the relative state measurements are subject to a zero-mean Gaussian noise distribution. A standard deviation of $\sigma = 0.005$ was used on all relative state measurements which include the attitude coordinates and coordinate rates. In practice, the relative attitude could be measured using a stereo or laser-based imager. The relative attitude can be computed and given in the appropriate attitude coordinates. Aided with gyros measuring inertial angular velocity, a relative angular velocity could be estimated. Using the attitude influence matrix, the relative coordinate rates could also be provided for computation of the pursuit-evasion feedback control solution.

VII.D.1. Complete Information

The simulation results for the complete information case are shown in Figs. VII.1 - VII.5. Total relative attitude error is computed using

$$e_{att} = cos^{-1}\left[\frac{tr\left(R_t R_e^T\right) - 1}{2}\right], \tag{7.43}$$

where $tr(*)$ is the trace of $*$, R_t is the desired attitude matrix which is identity, and R_e is the attitude matrix formed from the relative CRPs, $\boldsymbol{q}$, using

$$R_e = [1] - \frac{2}{1+q^2}\left[q^\times\right] + \frac{2}{1+q^2}\left[q^\times\right]\left[q^\times\right] \;, \quad q^2 = q_1^2 + q_2^2 + q_3^2 \,. \tag{7.44}$$

The relative, pursuer, and evader states for the 300 second simulation are illustrated in Figs. VII.1, VII.2, and VII.3, respectively. The control input which consists of those requested by the PE solution and the actual applied torques from dynamic inversion are shown in Fig. VII.4. The cumulative cost and cost-to-go are found in Fig. VII.5.

After 300 seconds the relative state attitude error is 0.9472 degrees with a relative angular velocity of 0.0015 deg/s. No adverse effects result in the control input computation from the imposed two-step dynamic inversion as evident in Fig. VII.4. The cumulative cost and cost-to-go for the pursuer and evader are equivalent as expected from the complete information, zero-sum game. The total cost for the pursuer and evader was computed to be 4.9093 while the cost-to-go was $0.4233 \times 10^{-}3$.

VII.D.2. Incomplete Information

For the incomplete information case, the pursuer's gains remained constant and those assumed by the evader were altered. The evader's gains for the incomplete

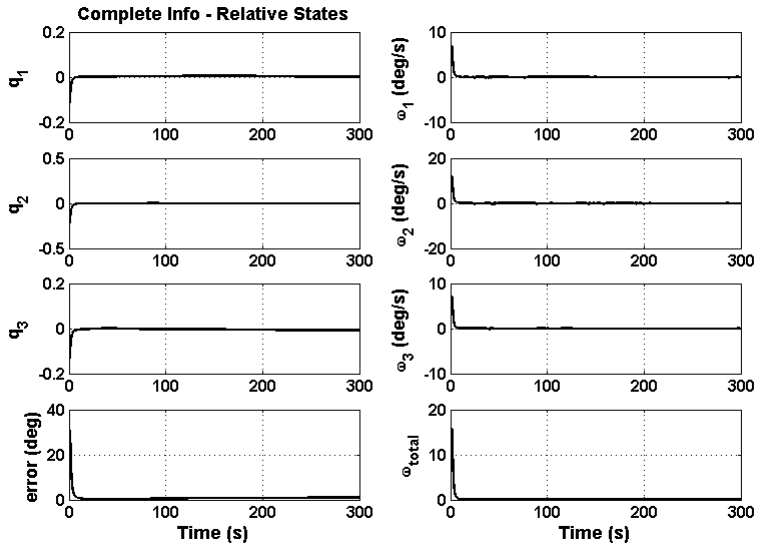

Figure VII.1. Complete Information Relative States

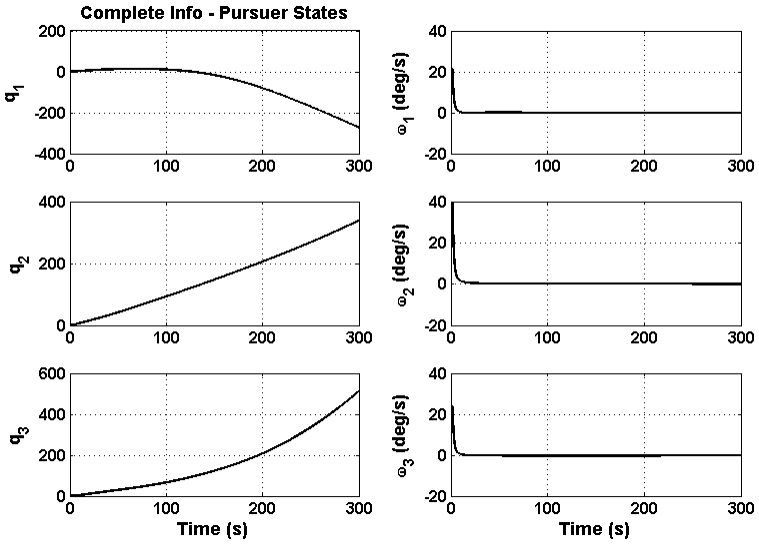

Figure VII.2. Complete Information Pursuer States

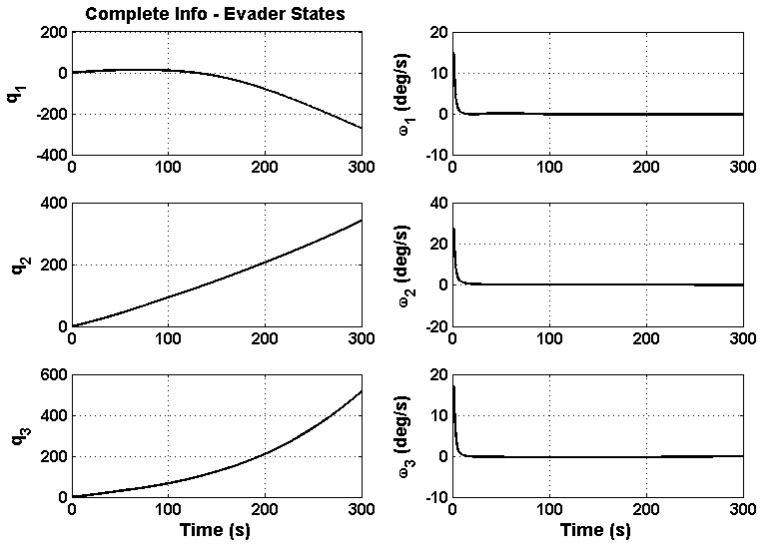

Figure VII.3. Complete Information Evader States

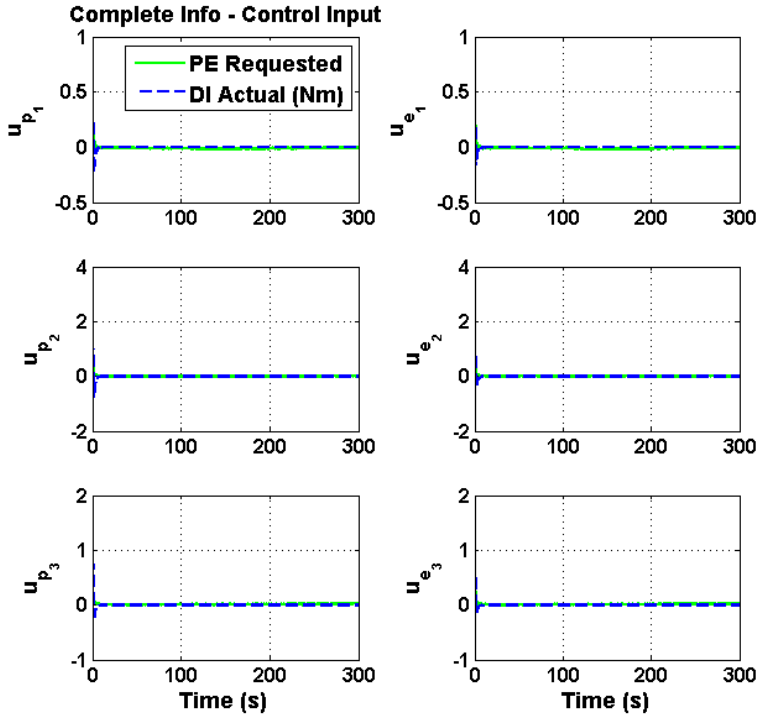

Figure VII.4. Complete Information Control Input

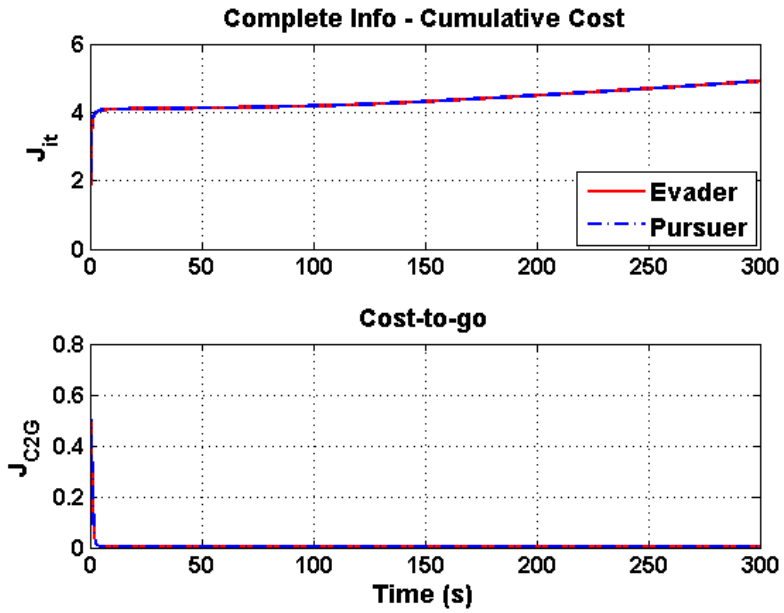

Figure VII.5. Complete Information Relative Cost

information case are summarized by

$$Q_e = \begin{bmatrix} 6.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{7.45}$$

$$R_{p_e} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R_{e_e} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}. \tag{7.46}$$

The relative, pursuer, and evader states for the 300 second simulation are illustrated in Figs. VII.6, VII.7, and VII.8, respectively. The control input which consists

of those requested by the PE solution and the actual applied torques from dynamic inversion are shown in Fig. VII.9. The cumulative cost and cost-to-go are found in Fig. VII.10.

After 300 seconds the relative state attitude error is 19.0259 degrees and with a relative angular velocity of 0.0031 deg/s. Again, no adverse effects result in the control input computation from the imposed two-step dynamic inversion as evident in Fig. VII.9, verifying the selection of the desired system for this particular example. The total cost for the pursuer was computed to be 3.6394×10^2 while that of the evader was 4.6851×10^2. The cost-to-go at the end of 300 seconds was 0.1739 and 0.1638 for the pursuer and evader, respectively.

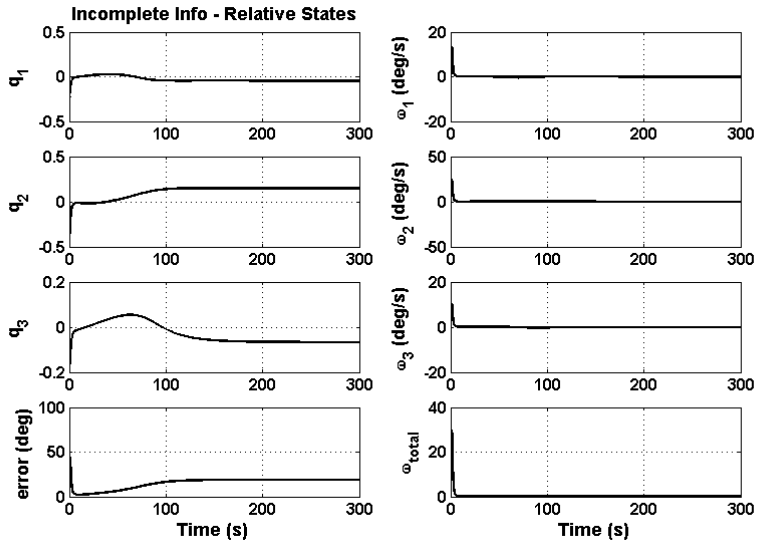

Figure VII.6. Incomplete Information Relative States

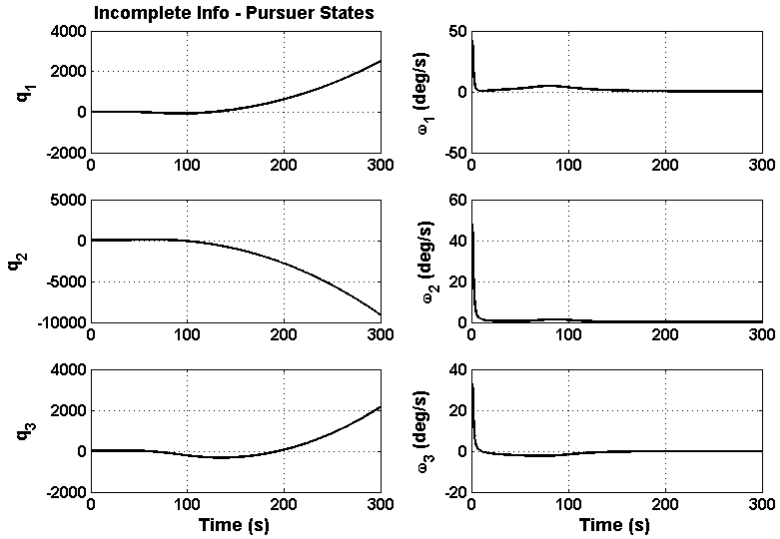

Figure VII.7. Incomplete Information Pursuer States

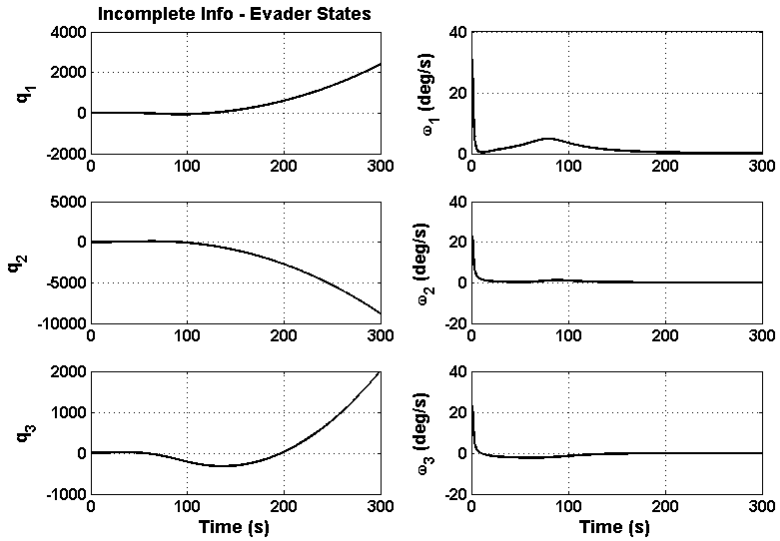

Figure VII.8. Incomplete Information Evader States

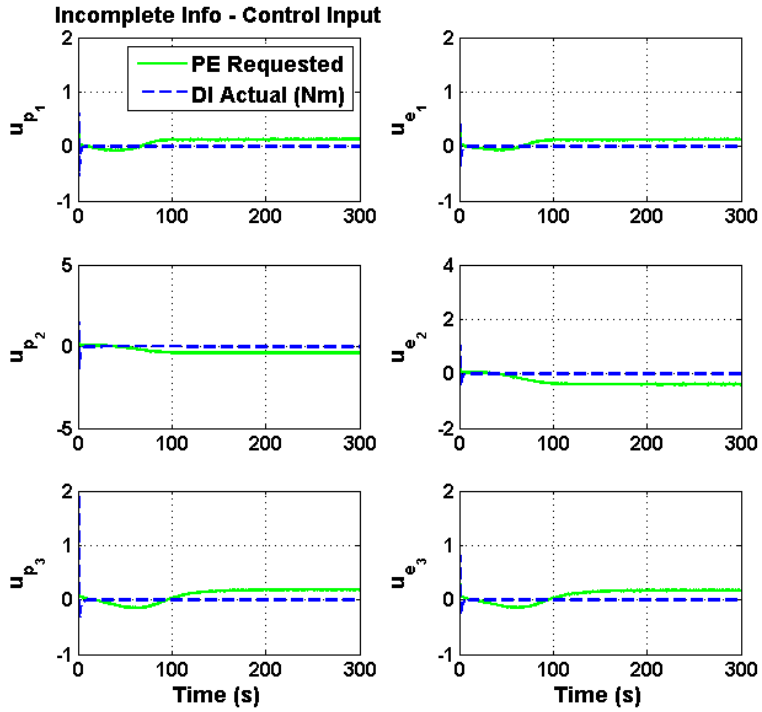

Figure VII.9. Incomplete Information Control Input

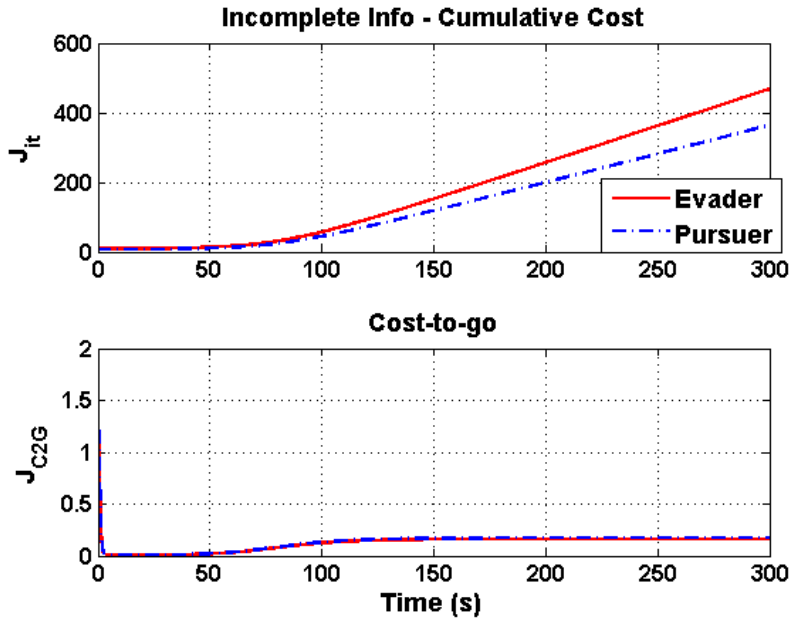

Figure VII.10. Incomplete Information Relative Cost

VII.D.3. Incomplete Information with Behavior Learning EKF

The objective of behavior learning for the infinite-horizon spacecraft reorientation problem involves reducing the final cost-to-go such that the slope of the cumulative cost decreases when compared to the incomplete information case. Additionally, the final attitude error and relative angular velocity at the 300 second mark should also be reduced. Even though the behavior learning filter was used for the entire game, the solution was recomputed a single time at $t = 10$ seconds.

The relative, pursuer, and evader states for the 300 second simulation are illustrated in Figs. VII.11, VII.12, and VII.13, respectively. The control input which consists of those requested by the PE solution and the actual applied torques from dynamic inversion are shown in Fig. VII.14. The cumulative cost and cost-to-go are found in Fig. VII.15 while the effective estimates are shown in Fig. VII.16.

After 300 seconds the relative state attitude error is 15.6157 degrees and with a relative angular velocity of 0.0021 deg/s. No adverse effects result in the control input computation from the imposed two-step dynamic inversion as evident in Fig. VII.9. The total cost for the pursuer was computed to be 7.2333×10^1 while that of the evader was 4.0062×10^2. The cost-to-go at the end of 300 seconds was 0.1608 and 0.1096 for the pursuer and evader, respectively.

Even though the gain estimates did not converge on the true gains used by the evader, the pursuer was still able to improve its performance using behavior learning. The reason for non-convergence is because of the invalid assumption made regarding the linear attitude dynamics found in Eqn. 7.20. Despite the improperly modeled relative system, the pursuer is still able to estimate values for k_1 and k_2 that are

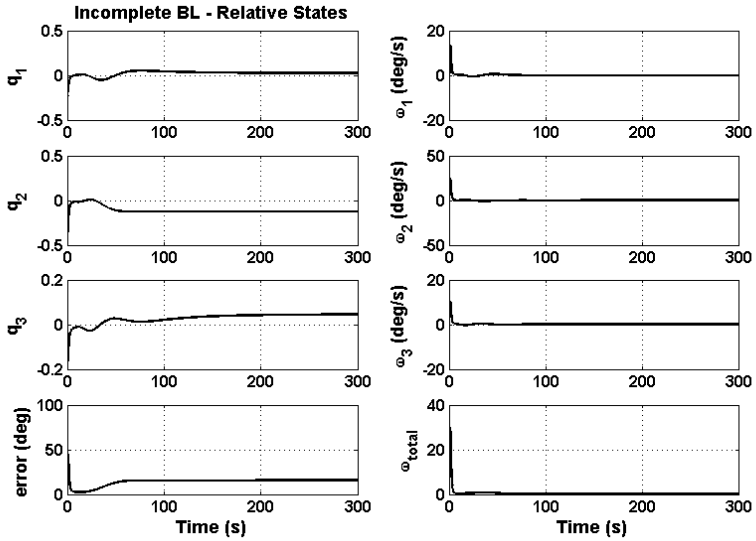

Figure VII.11. Behavior Learning Relative States

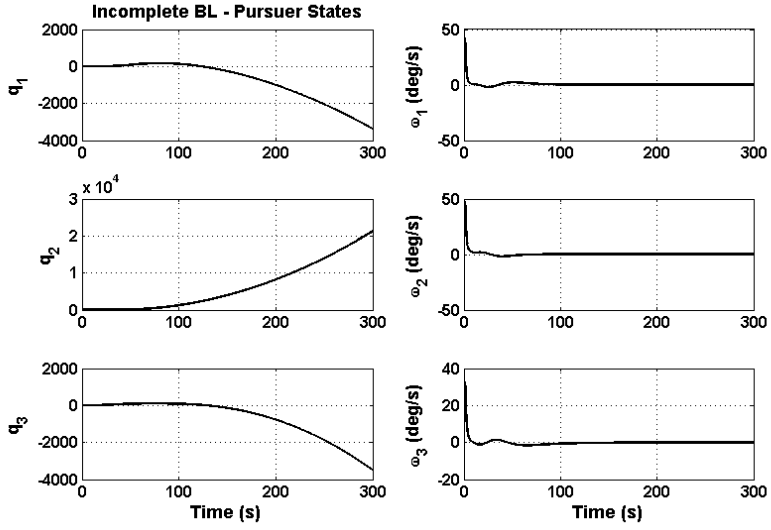

Figure VII.12. Behavior Learning Pursuer States

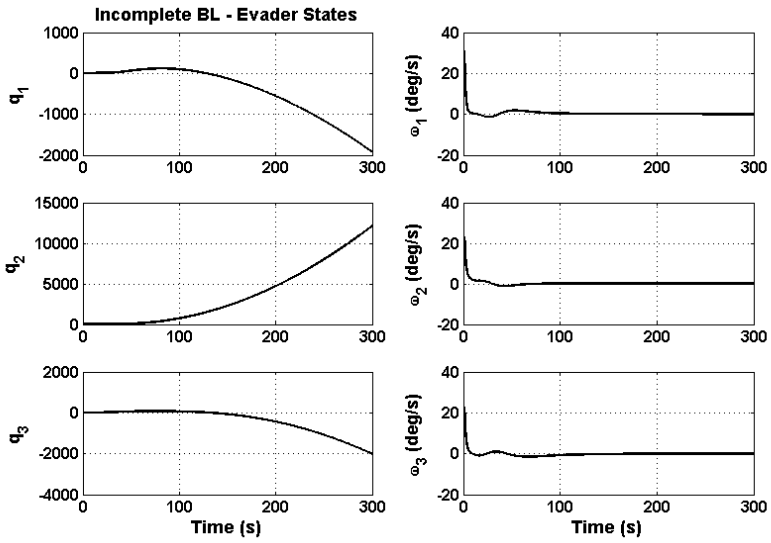

Figure VII.13. Behavior Learning Evader States

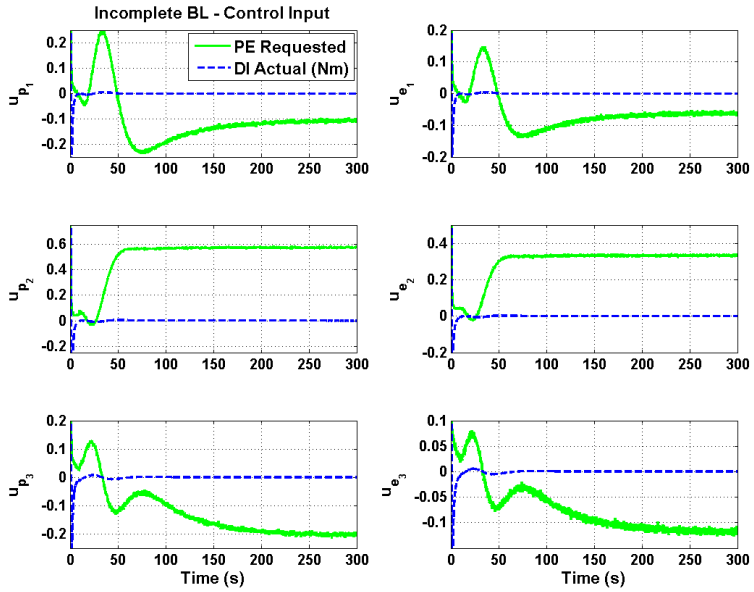

Figure VII.14. Behavior Learning Control Input

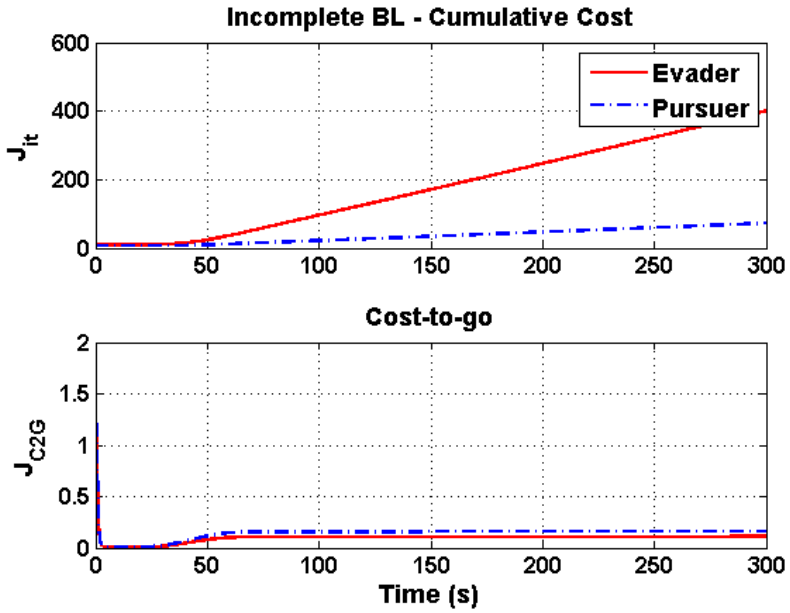

Figure VII.15. Behavior Learning Relative Cost

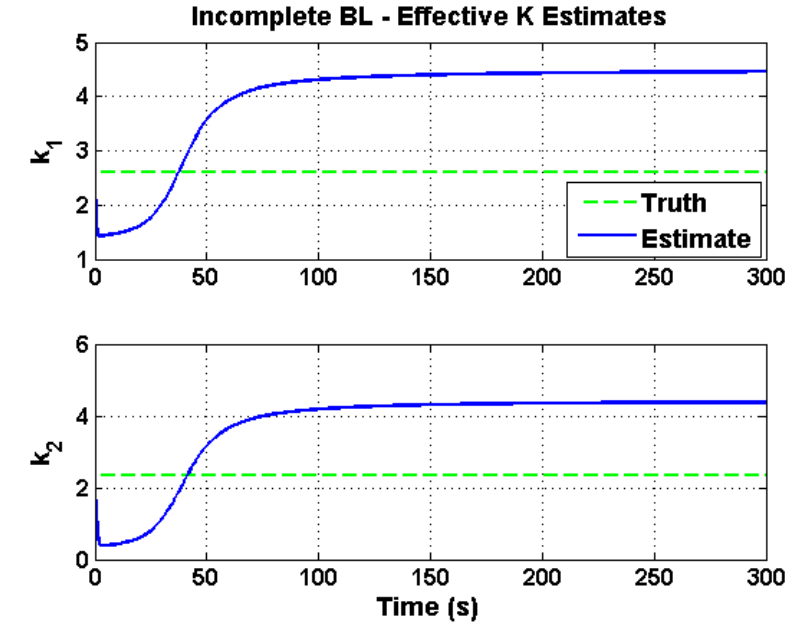

Figure VII.16. Effective Evader Gain Estimates

consistent with the measured values for the relative states and the assumed model. This gives the pursuer some indication of how the evader is behaving and is used to help the pursuer perform better than if a zero-sum strategy was implemented for the entire game.

VII.E. Summary

Cumulative cost and cost-to-go comparisons for each of the three cases are shown in Figs. VII.17 and VII.18. The pursuer is able to increase its performance for the incomplete information scenario with the help of behavior learning. With the effective estimates shown in Fig. VII.16, it can be concluded that behavior learning algorithms will not always converge on the true solution but their results can still be effective, much like the principles found in adaptive control [41]. The pursuer's final cost summary is shown in Table VII.1.

In the presence of severe modeling deficiencies, behavior learning aided the pursuer in playing more effectively. The transient of the gain estimates shown in Fig. VII.16 suggest that continuously augmenting the pursuer's control solution could be more effective than a single recomputation at the ten second mark. However, due to the nature of infinite horizon games, the evader's feedback gain K_e remains fixed. If the pursuer were to continuously augment its solution, the evader would continuously respond to the relative states which would actually decrease the pursuer's performance when compared to the incomplete information game with no behavior learning enabled. Therefore, when subjected to an infinite horizon game, it is of the best interest of the pursuer to converge on a behavior solution as quickly as possible

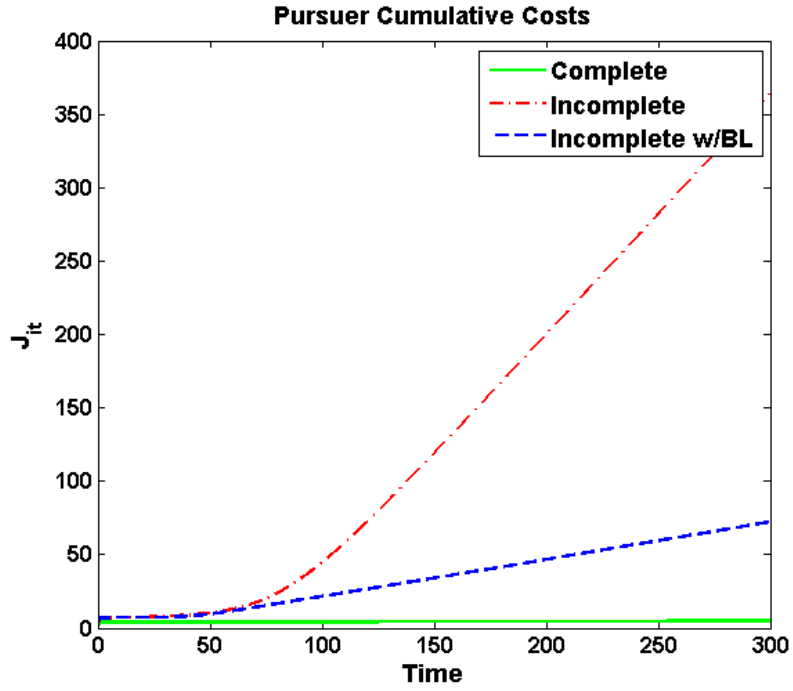

Figure VII.17. Pursuer Cumulative Cost Comparison

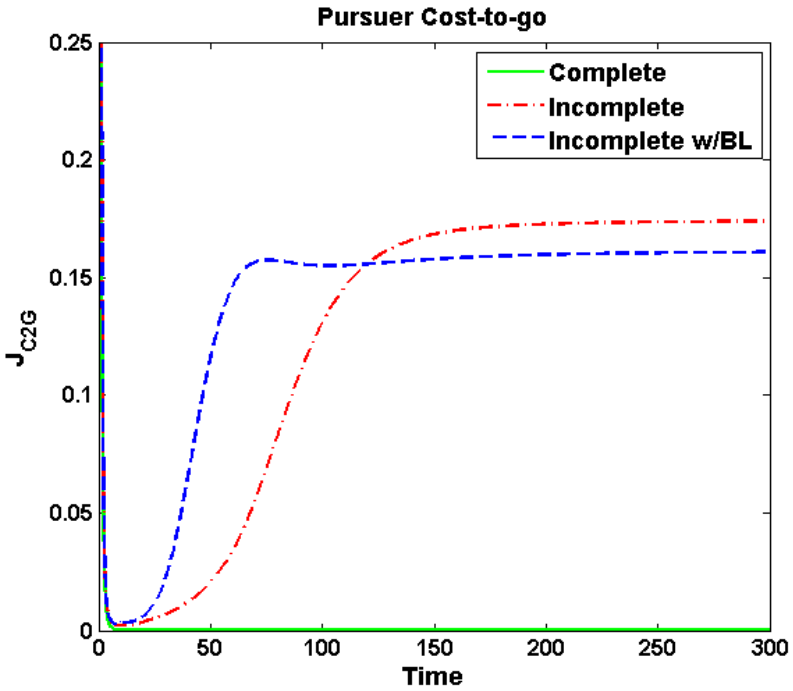

Figure VII.18. Pursuer Cost-To-Go Comparison

Table VII.1. Spacecraft Reorientation Cost Summary

Information Type	**Pursuer Cost**
Complete	4.9093×10^{0}
Incomplete	3.6394×10^{2}
Incomplete + BL	7.2333×10^{1}

then perform a control solution augmentation once.

CHAPTER VIII

APPLICATIONS: MISSILE INTERCEPTION

The interception problem has been one of the most popular military applications of differential games since the mid 1960's [14]. Interception can be applied to a variety of aircraft and missile scenarios including air-to-air combat of fighter jets, missile guidance for aircraft interception, defensive maneuvering of an aircraft to prevent a missile strike, and interception of an intercontinental ballistic missile (ICBM) by a missile defense system. Because of the expensive and high-risk testing that must be done in the development of systems that can achieve such goals, applicable pursuit-evasion solutions to the intercept problem are of high national interest.

With the help of differential game theory, several optimal missile guidance laws have been developed which take advantage of a pursuit-evasion framework [42–45]. These guidance laws have been developed with both offensive and defensive strategies in mind. Guidance laws that take into account the bounded control nature of missile systems have also been studied [46], including applications of time-varying systems [47]. For the interception of ICBMs upon reentry, linearized and multiple model techniques have also been developed [48, 49]. More recently, cooperative solutions have been of particular interest with focus on defensive strategies for missile and aircraft teams [50–52].

Due to the important application of pursuit-evasion strategies to military defense strategies, the need for these techniques to work under incomplete information scenarios remains essential. The possible lack of intelligence associated with an oppo-

nent continues to be an ongoing issue. Therefore, the application of behavior learning techniques to the missile interception problem can be very helpful in reducing the risk associated with development, testing, and final execution of the pursuit-evasion based guidance laws.

Consider an ICBM at some altitude that has been detected by a missile defense system. The flight of ICBMs can be separated into three phases: boost or ascent phase, midcourse phase, and terminal phase. During the boost phase, the missile aims to reach the exoatmosphere while achieving a specified flight envelope. During midcourse, maneuvering via aerodynamic forces and moments is unavailable due to air density. Following the midcourse phase, reentry occurs and the ICBM follows a ballistic trajectory. A diagram illustrating these phases of flight are shown in Fig. VIII.1.

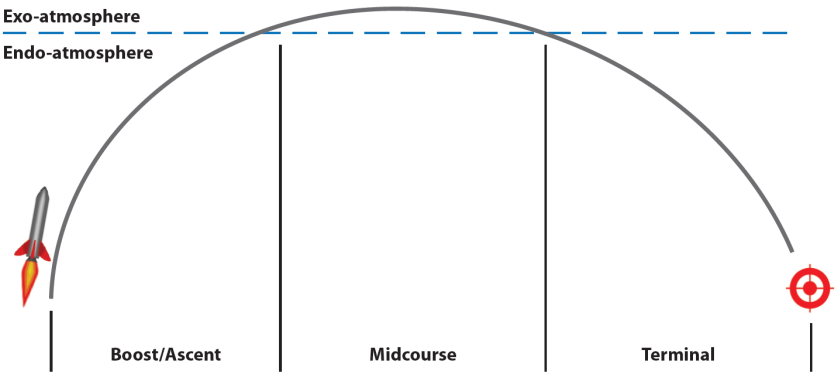

Figure VIII.1. Flight Phases of an ICBM

Upon the detection of an enemy ICBM launch via reconnaissance satellites, a second interceptor is launched to engage the threat and eliminate it. The goal of

the interceptor is come within a specified range of the target such that the onboard warhead's effective kill radius can destroy the evading missile within a fixed amount of time. Beyond that time, it could be possible for the evading missile system to deploy other defensive measures, separate another stage creating confusion of the specified target for the pursuer, or reach the exoatmosphere such that the aerodynamic forces and moments used to control the interceptor are ineffective. This scenario has a direct application to the ICBM interception problem which is an ongoing concern of military defense agencies. Figure VIII.2 shows a desired trajectory for the interceptor.

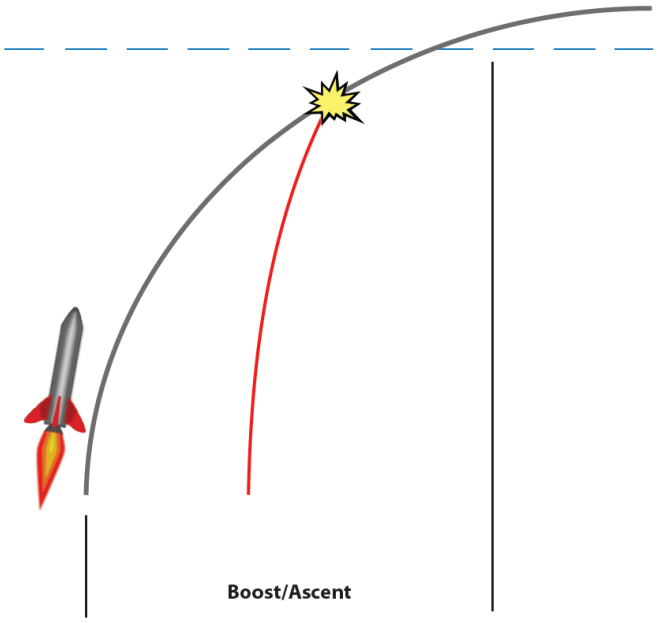

Figure VIII.2. Desired Interception of an ICBM

The following chapter develops a pursuit-evasion missile interception scenario, illustrates how an incomplete information game can affect the performance of the

interceptor, and shows how behavior learning can be used to carry out a successful mission in the presence of incomplete information.

VIII.A. Model

It will be assumed that the pursuing interceptor's launch site lies directly below the trajectory taken by the evading ICBM and neither the ICBM nor the interceptor experience any lateral motion. By doing this, the missile interception problem can be modeled in the vertical plane. The following additional assumptions are made about both vehicles:

- Each vehicle is equipped with a variable thruster which must remain positive.
- Each vehicle is equipped with control surfaces producing a total aerodynamic moment.
- Each vehicle maintains a constant mass throughout the game.
- The moment of inertia can be modeled by a constant density cylinder.
- Each vehicle experiences drag effects with a constant drag coefficient of 0.7.
- Drag is a function of speed and air density, which is a function of altitude.
- Drag acts through the center of pressure and opposite of the velocity vector.
- The earth is flat and acceleration due to gravity is constant.

The forces and moments which govern the motion of the missile in flight are shown in Fig. VIII.3. The inertial reference frame is denoted by n_1 and n_2 while the body-fixed reference frame is denoted by b_1 and b_2.

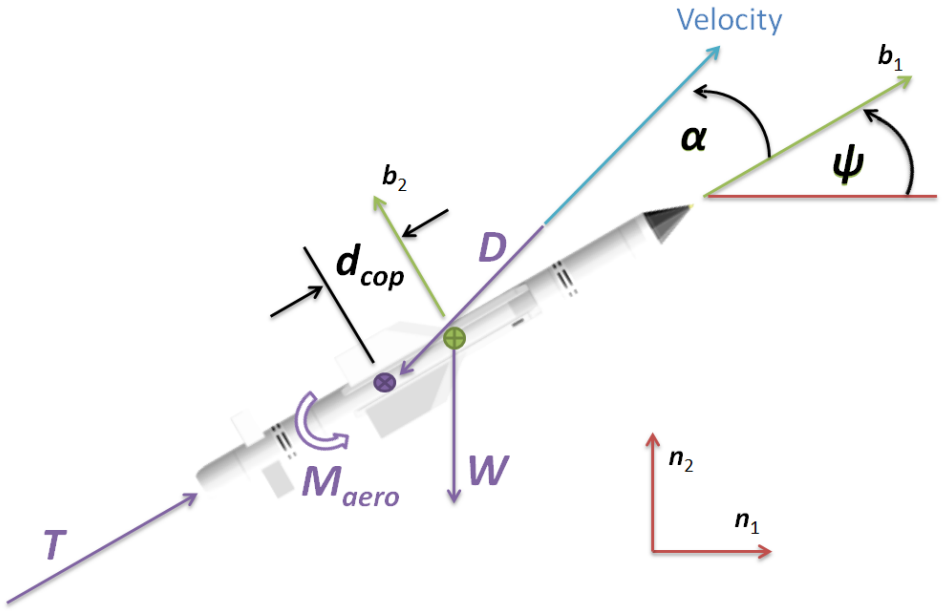

Figure VIII.3. Missile Model

The equations of motion for a single vehicle with respect to the inertial reference frame take on the form

$$m\ddot{x} = -D\cos(\alpha+\psi) + T\cos(\psi)\ , \tag{8.1}$$

$$m\ddot{y} = -W - D\sin(\alpha+\psi) + T\sin(\psi)\ , \tag{8.2}$$

$$I\ddot{\psi} = d_{cop}D\sin(\alpha) + M_{aero}\ , \tag{8.3}$$

where x and y describe the inertial position of the vehicle and ψ describes the orientation of the vehicle with respect to the horizon. Angle ψ represents the angle of attack which is defined as the angle between the velocity vector and a body-fixed axis running from the center of mass out through the missile nose cone. Forces W and D are those from weight and drag, respectively, while d_{cop} is the distance from the center of mass to the center of pressure. The controls are given by the thrust force, T, and the aerodynamic moments M_{aero}. The vehicle's mass properties are

made up of mass m and moment of inertia I.

The objective of the pursuit-evasion game will rely on relative position, therefore Eqns. 8.1 and 8.2 are of primary interest. Unfortunately, only one control, T, is present in these equations. It is possible to take advantage of Eqn. 8.3 and use the control M_{aero} to track a reference value for ψ. By doing this, ψ can now be used as an additional control within Eqns. 8.1 and 8.2 which provides two control inputs for two degrees-of-freedom. Because the two-step dynamic inversion framework is affine in the controls, it is necessary to define inputs $v_1 = T\cos(\psi)$ and $v_2 = T\sin(\psi)$.

For player i, equations 8.1 and 8.2 can be rewritten as

$$\dot{\boldsymbol{s}}_i = \boldsymbol{f}_i + G_i \boldsymbol{v}_i \,, \tag{8.4}$$

where

$$\boldsymbol{s}_i = [x_i,\, y_i]^T \,, \qquad \boldsymbol{v} = [v_{i_1},\, v_{i_2}]^T \,, \tag{8.5}$$

and

$$\boldsymbol{f}_i = \begin{bmatrix} \frac{1}{m_i}\left(-D_i \cos\left(\alpha_i + \psi_i\right)\right) \\ \frac{1}{m_i} - W_i - D_i \sin\left(\alpha_i + \psi_i\right) \end{bmatrix} \,, \qquad G_i = \begin{bmatrix} \frac{1}{m_i} & 0 \\ 0 & \frac{1}{m_i} \end{bmatrix} \,. \tag{8.6}$$

Equation 8.4 is in the form necessary for dynamic. At first glance, vector function $\boldsymbol{f}_i$ seems to be linear in the states, which can be true depending on how one wishes to consider the drag force D_i. Drag is calculated using

$$D_i = \frac{1}{2}\rho_i v_i^2 C_{d_i} A_{c_i} \,, \tag{8.7}$$

where ρ_i is the air density which is computed as a function of altitude y_i, and v_i is the airspeed which is a function of $\dot{x}_i$ and $\dot{y}_i$. The drag coefficient and cross-sectional area are denoted by C_{d_i} and A_{c_i}, respectively. Therefore, it could be argued that D_i

is a nonlinear function of the states y_i, $\dot{x}_i$, and $\dot{y}_i$, making $\boldsymbol{f}_i$ a nonlinear function of the states. If $\boldsymbol{f}_i$ is considered to simply be a time-varying vector of parameters, then it can also be modeled as a disturbance. For either case, implementing dynamic inversion on the system will allow us to write the relative system in the familiar form $\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u}_p - B\boldsymbol{u}_p$.

By requesting that the acceleration level variables $\ddot{x}_i$ and $\ddot{y}_i$ be directly controlled by u_{i_1} and u_{i_2}, respectively, it follows

$$\boldsymbol{v}_i = G_i^{-1} \left[\boldsymbol{w}_i - \boldsymbol{f}_i \right] , \tag{8.8}$$

where

$$\boldsymbol{w}_i = \left[u_{i_1},\, u_{i_2} \right]^T . \tag{8.9}$$

Vector $\boldsymbol{w}_i$ is provided from the pursuit-evasion optimal control solution, then Eqn. 8.8 is used to compute the necessary $\boldsymbol{v}_i$ required to force the system to follow the dynamics imposed by $\boldsymbol{w}_i$. Finally, the necessary T_i and ψ_{*_i} is computed using

$$\psi_{*_i} = \tan^{-1}\left(\frac{v_{i_2}}{v_{i_1}} \right) , \tag{8.10}$$

$$T_i = v_{i_1} \cos\left(\psi_i\right) + v_{i_2} \sin\left(\psi_i\right) . \tag{8.11}$$

By implementing a proportional-derivative (PD) or similar controller on ψ_{*_i} using Eqn. 8.3, a required value for M_{aero_i} can be computed. This framework allows the states of interest, x_i and y_i, to be controlled using the available controls T_i and M_{aero_i}. One item to address is whether ψ_i or ψ_{*_i} should be used for the computation of T_i in Eqn. 8.11. For consistency, the current value of ψ_i was used over the requested ψ_{*_i} based on the verified assumption that the controller on ψ_i can settle and track

ψ_{*_i} with the appropriate performance. The act of considering multiple degrees-of-freedom and reducing them to the minimum number of states needed for the PE game is referred to as transforming the realistic space to the reduced space [1].

For this system, dynamic inversion is used on a subset of the available states based on the goals of the players. If this process is performed for each player, the pursuit-evasion dynamics for Player i become

$$\dot{\boldsymbol{z}}_i = A\boldsymbol{z}_i + B\boldsymbol{u}_i \,, \tag{8.12}$$

where

$$\boldsymbol{z}_i = [x_i,\, y_i,\, \dot{x}_i,\, \dot{y}_i]^T \,, \qquad \boldsymbol{u}_i = [u_{i_1},\, u_{i_2}]^T \,, \tag{8.13}$$

and

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k & 0 & -c & 0 \\ 0 & -k & 0 & -c \end{bmatrix} \,, \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \,, \tag{8.14}$$

with k and c being positive constant. This particular selection of A allows the system to experience damped oscillation if so desired. When $k = c = 0$, then Eqn. 8.12 reduces to the same single player model used in the previous chapters.

By defining the relative states as

$$\boldsymbol{z} = \boldsymbol{z}_p - \boldsymbol{z}_e \,, \tag{8.15}$$

the relative system equations of motion become

$$\dot{\boldsymbol{z}} = A\boldsymbol{z} + B\boldsymbol{u}_p + B\boldsymbol{u}_e \,. \tag{8.16}$$

It is now possible to employ a linear-quadratic pursuit-evasion game for the two missiles.

VIII.B. Pursuit-Evasion Game

The final-time-fixed missile interception pursuit-evasion game is defined by the zero-sum performance index

$$J_{MI} = \frac{1}{2}\boldsymbol{z}_f^T S_f \boldsymbol{z}_f + \frac{1}{2}\int_{t_0}^{t_f} \left(\boldsymbol{z}^T Q \boldsymbol{z} + \boldsymbol{u}_p^T R_p \boldsymbol{u}_p - \boldsymbol{u}_e^T R_e \boldsymbol{u}_e\right) \mathrm{dt}\,, \tag{8.17}$$

subject to the linear dynamic constraint defined by Eqn. 8.16. The optimal solutions are given by

$$\boldsymbol{u}_p = -R_p^{-1} B^T S \boldsymbol{z}\,, \tag{8.18}$$

$$\boldsymbol{u}_e = -R_e^{-1} B^T S \boldsymbol{z}\,, \tag{8.19}$$

where S is the solution to the differential Riccati equation

$$\dot{S} = -Q - A^T S - SA - SB\left(R_e^{-1} - R_p^{-1}\right) B^T S\,. \tag{8.20}$$

VIII.C. Behavior Learning

Behavior learning, which is enabled by the pursuer, will attempt to estimate a model for the evader's strategy and therefore a means to predict the evader's behavior for all time. This behavior is captured in the opponent's Kalman gain K_e which takes on the form shown in Eqn. 8.21.

$$K_e = R_e^{-1} B^T S\,. \tag{8.21}$$

Because of the final-time-fixed nature of the game, K_e is time-varying. With A and B known, all reasonable assumptions about the system should be applied in an effort to reduce the number of states that need to be estimated to find a solution for the

opponent's behavior. The following assumptions are made by the pursuer's behavior learning algorithm:

- The evader implements a zero-sum safe strategy.
- The evader is not capable of behavior learning.
- The evader is not concerned with the relative states during game play.
- The evader weighs each of the relative position states at t_f equally.
- The evader is not concerned with the relative velocity states at t_f.
- The evader weighs each of the control inputs equally.
- The evader does not weigh any cross-coupling terms.

Under these assumptions, the evader's Kalman gain gain takes on the form

$$K_e = \begin{bmatrix} \frac{s_3}{r} & 0 & \frac{s_2}{r} & 0 \\ 0 & \frac{s_3}{r} & 0 & \frac{s_2}{r} \end{bmatrix}, \tag{8.22}$$

where

$$S_f = \begin{bmatrix} s_f & 0 & 0 & 0 \\ 0 & s_f & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad S = \begin{bmatrix} s_1 & 0 & s_3 & 0 \\ 0 & s_1 & 0 & s_3 \\ s_3 & 0 & s_2 & 0 \\ 0 & s_3 & 0 & s_2 \end{bmatrix}, \quad R_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_e = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}. \tag{8.23}$$

and $Q = \mathbf{0}$. Variables s_2 and s_3 found within K_e are time-varying and subject to

$$\dot{S} = -A^T S - SA - SB\left(R_e^{-1} - R_p^{-1}\right) B^T S\,. \tag{8.24}$$

The behavior learning algorithm becomes an estimator for the relative state vector $\boldsymbol{z}$, the three independent elements of S, and r. These states are summarized by the estimate vector

$$\boldsymbol{x} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ s_1 \\ s_2 \\ s_3 \\ r \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} . \tag{8.25}$$

The state equations for $x_1 - x_4$ are given by

$$\dot{\boldsymbol{z}} = (A + BK_e)\,\boldsymbol{z} + B\boldsymbol{u_p}\,, \tag{8.26}$$

while those for $x_5 - x_7$ are given by the scalar counterparts found in Eqn. 8.24. The state equation for x_8 is zero.

The state equations needed for the nonlinear estimator are summarized by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right)\,, \tag{8.27}$$

where

$$\boldsymbol{f} = \begin{bmatrix} x_3 \\ x_4 \\ u_{p_1} - x_3\left(c - \frac{x_6}{x_8}\right) - x_1\left(k - \frac{x_7}{x_8}\right) \\ u_{p_2} - x_4\left(c - \frac{x_6}{x_8}\right) - x_2\left(k - \frac{x_7}{x_8}\right) \\ 2kx_7 - x_7^2\left(\frac{1}{x_8} - 1\right) \\ x_6^2\left(1 - \frac{1}{x_8}\right) + 2cx_6 - 2x_7 \\ cx_7 - x_5 + kx_6 - x_6x_7\left(\frac{1}{x_8} - 1\right) \\ 0 \end{bmatrix}, \tag{8.28}$$

and $\boldsymbol{u_p} = [u_{p_1}, u_{p_2}]^T$. The measurements available are the relative states defined by

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) = \left[x_1\,, x_2\,, x_3\,, x_4\right]^T\,. \tag{8.29}$$

Equations 8.27 and 8.29 are in the standard form needed for a nonlinear filter. Any of several nonlinear estimation techniques can be used to filter the relative states and estimate the strategy parameters s_1, s_2, s_3 and r. By estimating these strategy parameters, the pursuer can then propagate these states forward in time to arrive at the estimated S_f. Because propagating the Riccati equation forward in time can lead to instability, caution must be exercised. The results shown were obtained by propagating the Riccati equation forward in time, then using the computed S_f to propagate backwards in time again in order to arrive at the best S for all time to define the behavior of the evader. If desired, these estimates can be continuously monitored and the pursuer may recompute its solution as necessary. The one-sided optimal control solution when K_e is known for all time is given in Section II.D.

Additional constraints also exist on the matrices S and R_e. From the optimal control theory these solution were derived from, S must be symmetric positive semidefinite for all time and R_e must be positive definite for all time. These constraints can be imposed by computing the nearest symmetric positive definite matrix

VIII.D. Simulation

To develop a baseline case for comparison purposes, a complete information case is simulated along with incomplete information and behavior learning enabled cases. The complete information gain selections are summarized by

$$S_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_e = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}. \tag{8.30}$$

For the complete information case, each player assumes a zero-sum strategy using the gains defined in Eqn. 8.30.

The initial conditions for all cases were chosen to be

$$\boldsymbol{z}_{i_0} = \begin{bmatrix} x_{i_0}[m] \\ y_{i_0}[m] \\ \psi_{i_0}[rad] \\ \dot{x}_{i_0}\left[\frac{m}{s}\right] \\ \dot{y}_{i_0}\left[\frac{m}{s}\right] \\ \dot{\psi}_{i_0}\left[\frac{rad}{s}\right] \end{bmatrix}, \qquad \boldsymbol{z}_{p_0} = \begin{bmatrix} 1000 \\ 0 \\ \frac{\pi}{2} \\ 100\cos\left(\frac{\pi}{2}\right) \\ 100\sin\left(\frac{\pi}{2}\right) \\ 0 \end{bmatrix}, \qquad \boldsymbol{z}_{e_0} = \begin{bmatrix} 0 \\ 3000 \\ \frac{\pi}{12} \\ 150\cos\left(\frac{\pi}{12}\right) \\ 150\cos\left(\frac{\pi}{12}\right) \\ 0 \end{bmatrix}. \tag{8.31}$$

The additional properties used to define the missile vehicles and environment are given in Table VIII.1.

Table VIII.1. Missile Mass Properties

Property	**Value**	**Units**
Mass	100	kg
Length	0.5	m
Radius	0.05	m
Drag Coefficient	0.7	N/A
Acceleration due to Gravity	9.81	m/s^2

Imperfect information dictates that the relative state measurements are subject to a zero-mean Gaussian noise distribution. A standard deviation of $\sigma = 0.03$ was used for the relative position measurements and $\sigma = 0.1$ was used for the relative velocity measurements. In practice, the relative position could be measured directly from relative sensors onboard the missile such at heat signature, radar, or laser based sensors. Based on the difference of these measurements between time steps and with the help of an inertial measurement unit, the relative velocities could be computed. It is also possible for a player to use a GPS-aided inertial navigation system to measure its own inertial states and have the inertial states of the opponent provided via communication link with a satellite- or ground-based tracking system. The relative measurements could then be formed from both sets of inertial states.

VIII.D.1. Complete Information

The complete information scenario was simulated in an effort to gain an understanding of the baseline performance. For each case, a final-time of 30 seconds was used along with a time step of 0.1 seconds. Results for this case are shown in Figs. VIII.4 - VIII.6. Figure VIII.4 gives a view of the vertical plane the game takes place in. The relative states for the 30 second game are summarized in Fig. VIII.5. Cumulative cost and cost-to-go plots are shown in Fig. VIII.6. The final cost for both players was calculated to be 1.3686×10^3.

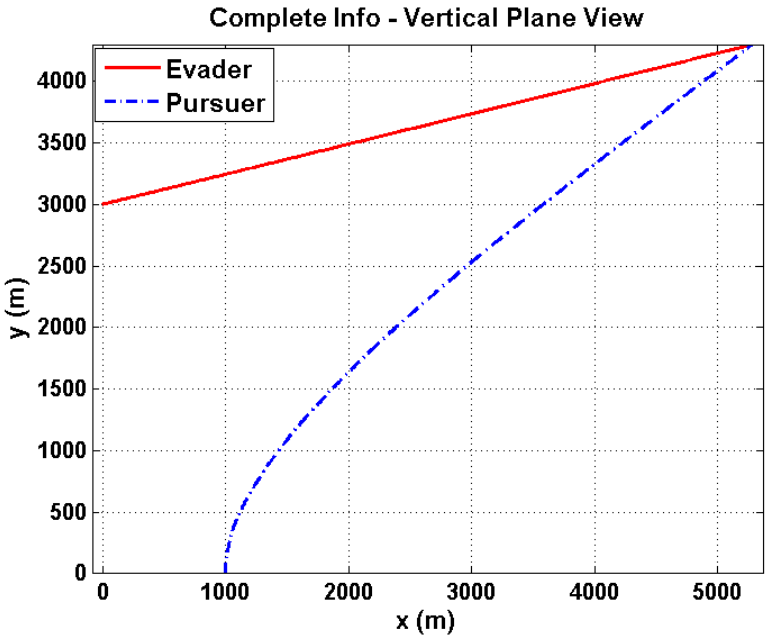

Figure VIII.4. Complete Information Vertical Plane View

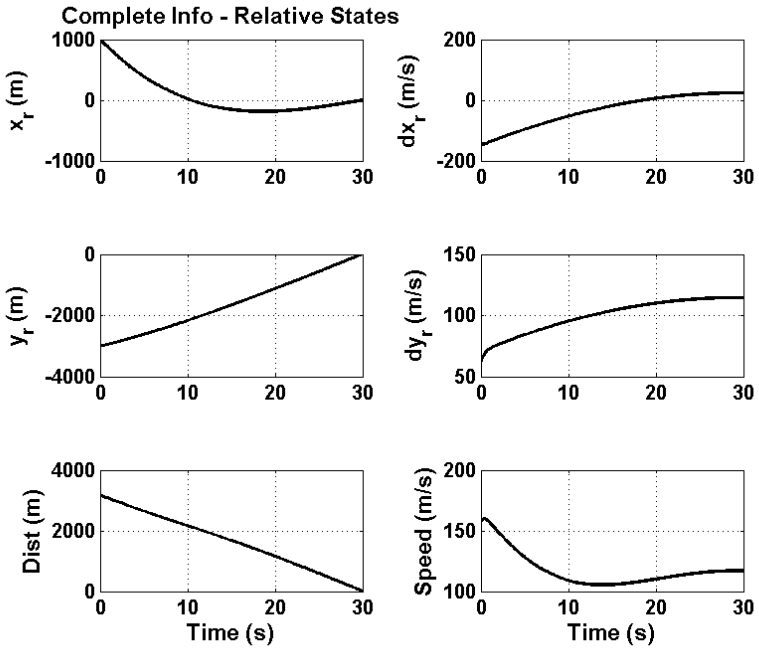

Figure VIII.5. Complete Information Relative States

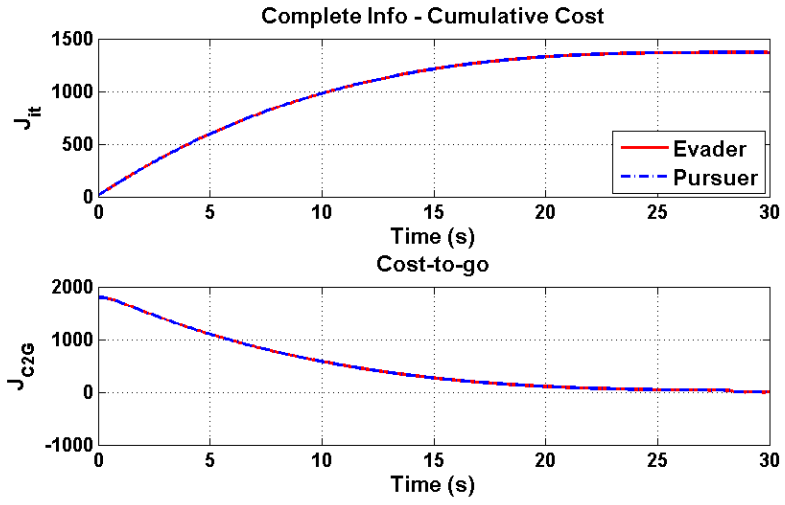

Figure VIII.6. Complete Information Cost Analysis

VIII.D.2. Incomplete Information

For, the incomplete information case, the pursuer assumed the same gains from the complete information case while the evader decided on a different gain selection. These gains are summarized by

$$S_{f_e} = \begin{bmatrix} 1.5 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R_{p_e} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_{e_e} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}. \tag{8.32}$$

The results for the incomplete information scenario are shown in Figs. VIII.7 - VIII.9. Figure VIII.7 gives a view of the vertical plane the game takes place in. The relative states for the 30 second game are summarized in Fig. VIII.8. Cumulative cost and cost-to-go plots are shown in Fig. VIII.9. As expected, the total cost for both the pursuer and evader increased with the gain assumptions associated with the incomplete information scenario. The pursuer's total cost was computed at 1.8537×10^3 while that of the evader was 1.7561×10^3.

VIII.D.3. Incomplete Information with Behavior Learning

The incomplete information scenario was simulated again with the pursuer enabled with a behavior learning algorithm. Behavior learning was used to compute and implement a one-sided optimal control solution at $t = 3$ seconds. Results for this case are illustrated in Figs. VIII.10 - VIII.18.

Figure VIII.10 gives a view of the vertical plane the game takes place in. Behavior learning estimates for the first five seconds of the game are are shown

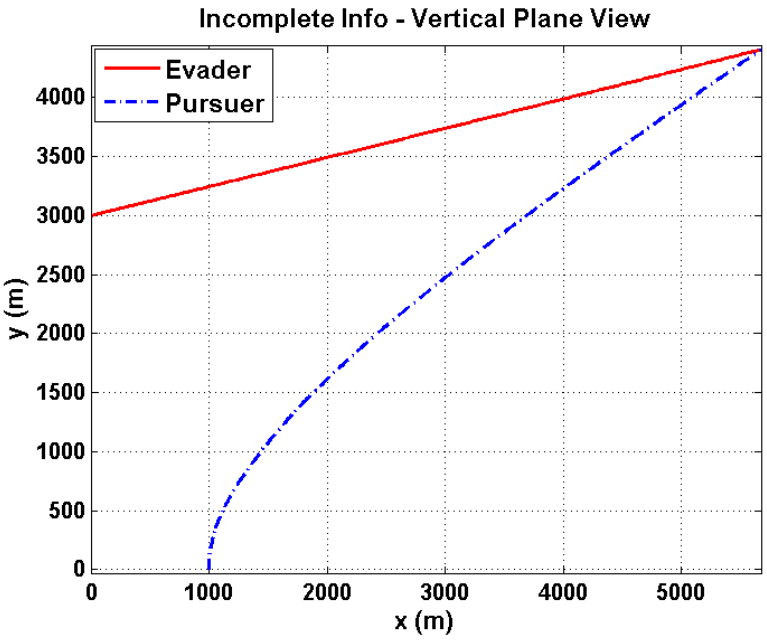

Figure VIII.7. Incomplete Information Vertical Plane View

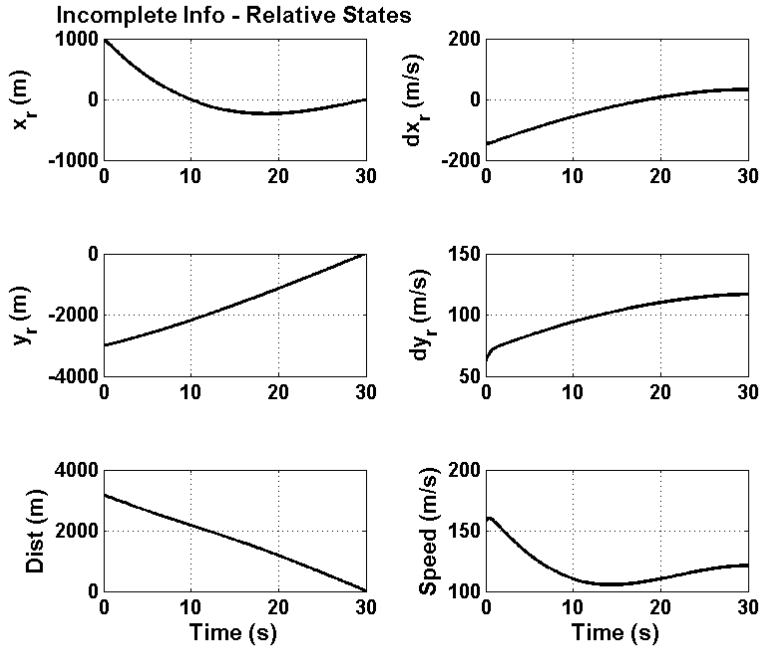

Figure VIII.8. Incomplete Information Relative States

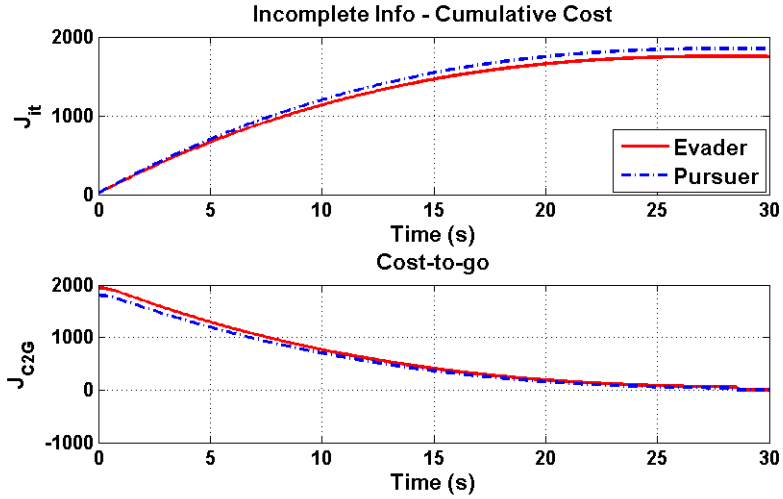

Figure VIII.9. Incomplete Information Cost Analysis

in Figs. VIII.11 and VIII.12. The relative states are summarized in Fig. VIII.13 while the inertial states for the pursuer and the evader can be seen Figs. VIII.14 and VIII.15, respectively.

Approximately one second passes before the S estimates can properly converge on the true values. In addition to the transients associated with the EKF, this is also brought about by the response in orientation tracking by the two vehicles. The response time for the commanded ψ for each vehicle is shown in Figs. VIII.14 and VIII.15. A PD controller was implemented to exploit ψ as a control input for the PE game. Large gain selections of $K_{\psi_p} = 25$ and $K_{\psi_e} = 10$ were chosen to achieve the quick but damped response. These PD gain selections were used for both vehicles. A satisfactory response in ψ is essential for the behavior learning algorithm to properly estimate the strategy gains.

The implementation of the new control solution is evident at the 3 second mark in Fig. VIII.14 when the pursuer's requested ψ makes a drastic switch once the

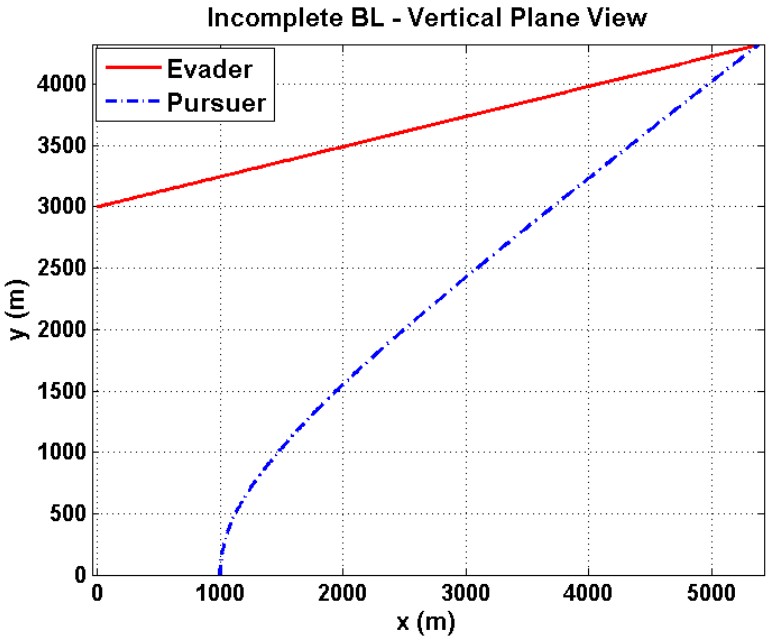

Figure VIII.10. Incomplete Information with Behavior Learning Vertical Plane View

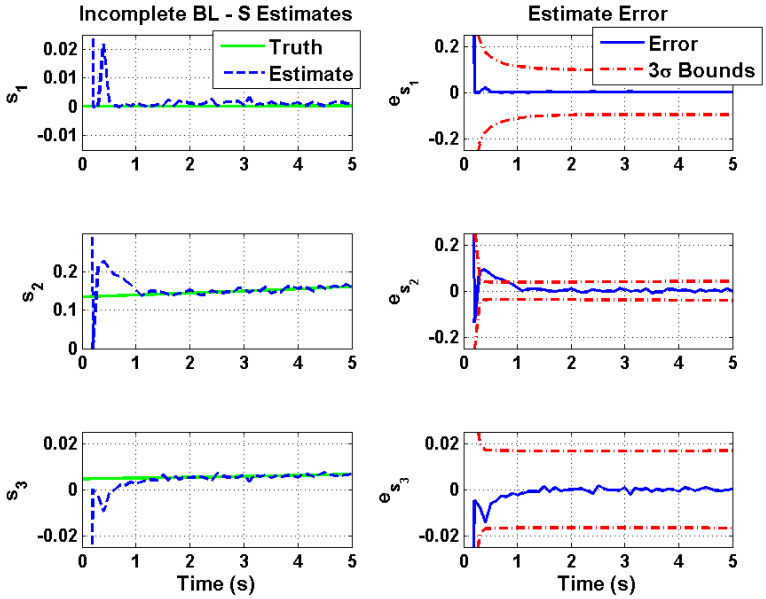

Figure VIII.11. Incomplete Information with Behavior Learning S Estimates

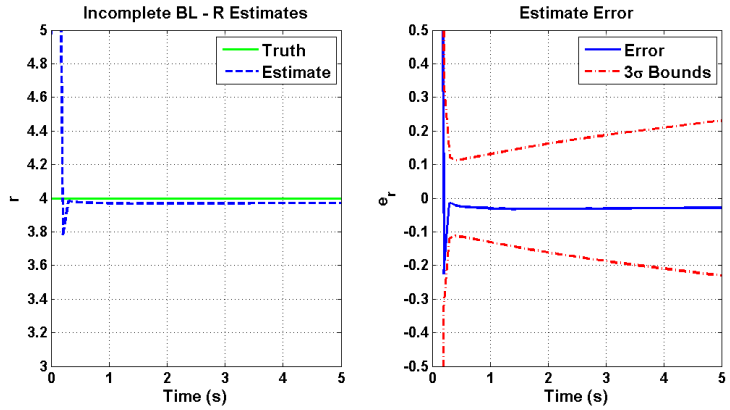

Figure VIII.12. Incomplete Information with Behavior Learning R Estimate

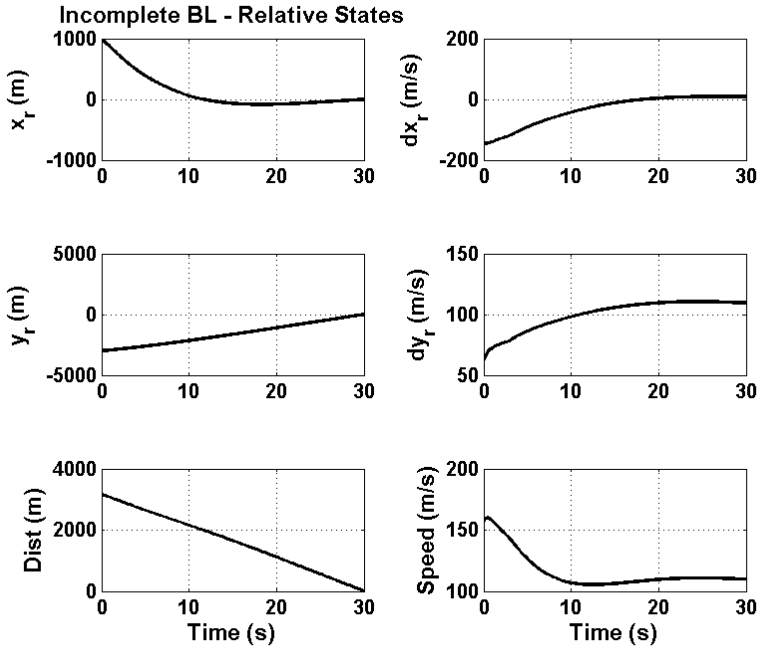

Figure VIII.13. Incomplete Information with Behavior Learning Relative States

opponent's behavior can be predicted.

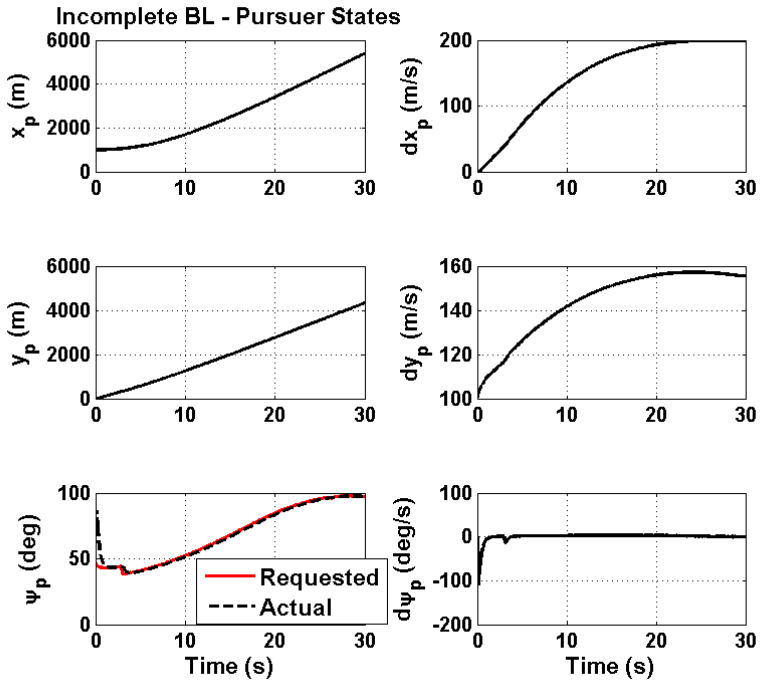

Figure VIII.14. Incomplete Information with Behavior Learning Pursuer States

The actual and requested input by the dynamic inversion process is shown in Fig. VIII.16. This discrepancy is also a product of the response time in the ψ-tracking. Larger errors in the requested and actual input are shown at the beginning of the game when each vehicle becomes aware of their opponent. The true missile control inputs are illustrated in Fig. VIII.17. The thrust input for the pursuer peaks at approximately 2.4 kN and trails off to approximately 800 N. This is representative of an engine equipped with a maximum thrust of 2.5 kN which can be throttled down

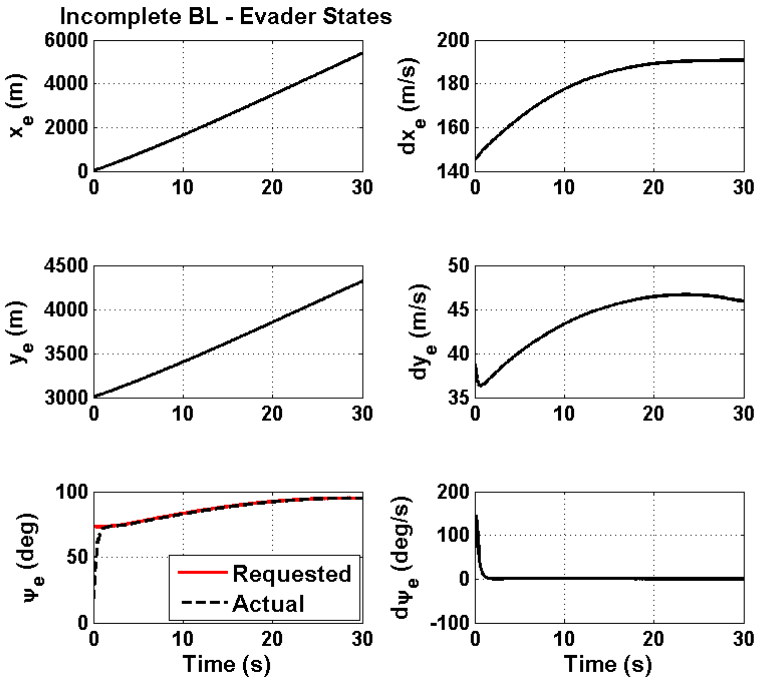

Figure VIII.15. Incomplete Information with Behavior Learning Evader States

to approximately 30%. The evader's thrust characteristics would require an engine size with a maximum thrust value of 1.2 kN with throttling capabilities down to 75%. Recently, Space Exploration Technologies Corporation has included the capability of throttling down to 70% in their Merlin 1D rocket engines [53]. Although anti-missile systems are generally equipped with solid rocket boosters unlike the Merlin 1D engines, newer examples of interceptor missile have adopted liquid rocket designs specifically for their throttling capabilities.

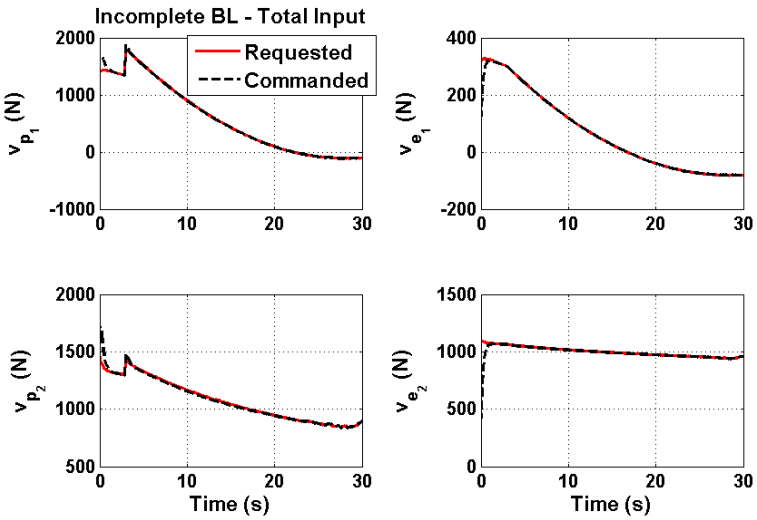

Figure VIII.16. Incomplete Information with Behavior Learning Dynamic Inversion Input

Cumulative cost and cost-to-go plots are shown in Fig. VIII.18. The total cost for the pursuer was decreased as a result of the behavior learning algorithm, but it was still higher than that associated with the complete information scenario. The pursuer's total cost was computed at 1.3745×10^3 while that of the evader was

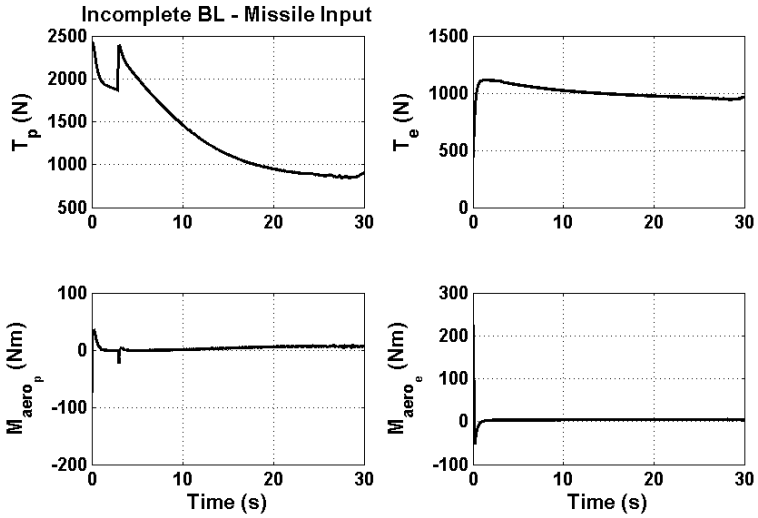

Figure VIII.17. Incomplete Information with Behavior Learning Missile Input

1.4984×10^3.

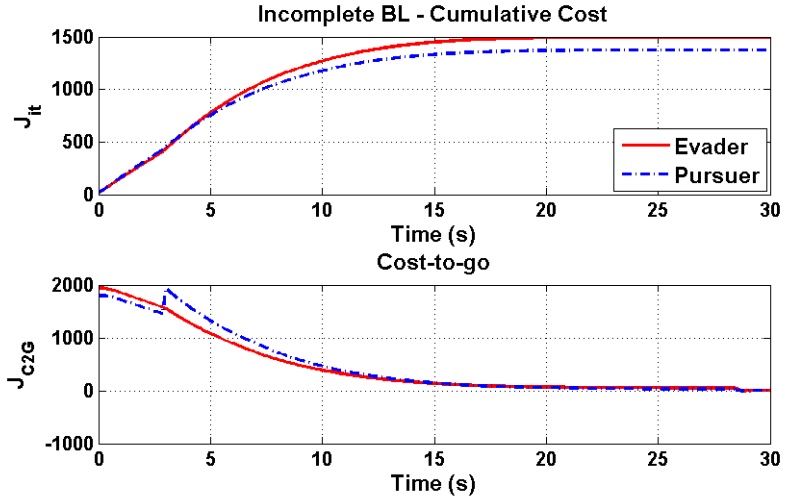

Figure VIII.18. Incomplete Information with Behavior Learning Cost Analysis

VIII.E. Summary

A comparison of the pursuer's cumulative cost for each of the three simulation found in this chapter are shown in Fig. VIII.19. With the introduction of behavior learning, the pursuer is able to predict how the evader will respond and modify the two-sided pursuit-evasion problem into a one-sided optimal control problem. The advantage of this method when in the presence of incomplete information is summarized by the cumulative cost comparison. With behavior learning, the pursuer's final cost approaches that of the complete information case. Is it clear that behavior learning can be extremely useful when applied to final-time-fixed interception problems and robust enough to provide a solution when dynamic inversion and alternate control methods are necessary for implementation. The pursuer's final cost summary is shown in Table VIII.2. The primary limitation of final-time-fixed behavior learning is due to the unstable nature of the Riccati equation when propagated forward

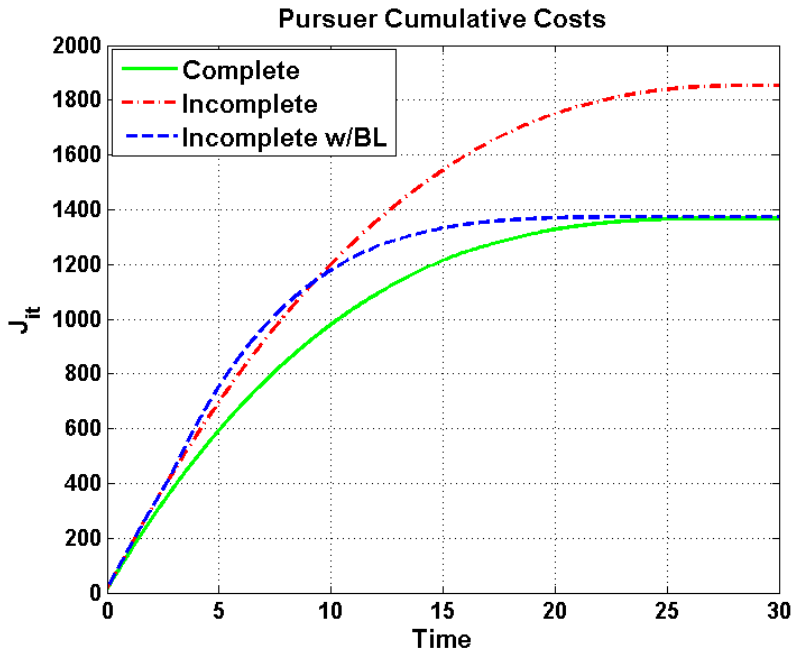

Figure VIII.19. Pursuer Cumulative Cost Comparison

in time. Therefore, it is essential for the behavior learning algorithm to converge on proper strategy estimates before using those estimates to propagate forward in time to compute S_f. A few additional steps are necessary to make this algorithm robust enough for repeatable execution. The estimates related to the Q and R gain matrices that are used for the new solution computation should be selected based on their covariance. That is, throughout the first few seconds of the game, those values with the smallest associated covariance value should be selected. The selected estimates of S are also of critical importance. After the covariance converges, a set of estimates for each parameter should be taken into consideration over the period of one to three seconds. Because the Riccati equation experiences near-linear behavior at the beginning of the game, the effective estimate for each of these parameters can be taken as the mean of the values from the desired period. These mean values are

Table VIII.2. Missile Interception Cost Summary

Information Type	**Pursuer Cost**
Complete	1.3686×10^3
Incomplete	1.8537×10^3
Incomplete + BL	1.3745×10^3

then used to determine an effective S at a time at the midpoint of the sample period. This process was implemented to achieve robustness and repeatability.

One possible solution to the Riccati forward propagation issue is the implementation of the inverse Riccati equation. Its use, however, does not necessarily eliminate all forward propagation issues. The inverse Riccati equation is useful for certain gain selections, but in many cases, can produce a solutions for the independent elements of S which exhibit additional dynamic characteristics which are undesirable. Its use should also be exercised with caution. With that, it would be more convenient if a single forward propagator could be used reliably and handle all types of gain selections because the initialization of an opponent's strategy is simply a guess. An opponent's strategy could be defined by any number of gain combinations and for this reason, the implemented forward Riccati propagation with additional statistical analysis proved to be the most effective. The inverse Riccati equation could be used in addition to the method presented here and the best solution could be implemented for the control solution augmentation.

CHAPTER IX

CONCLUSION

This dissertation presented behavior learning frameworks for final-time-fixed, infinite-horizon, and final-time-free pursuit-evasion games which are applicable to the incomplete, imperfect, and uncertain information scenarios. The developed methods focus on continuous-time pursuit-evasion games whose cost functions are quadratic in nature and whose relative dynamic systems are defined by linear differential equations. A two-step dynamic inversion technique was introduced to allow these behavior learning methods to be extended to nonlinear, control-affine dynamic systems. The two-step process outlined allows systems that are nonlinear in the kinematics and dynamics to take on the form of a linear double-integrator.

Two key aerospace applications were shown which invoked the behavior learning and dynamic inversion methods presented. Spacecraft rendezvous and missile interception are problems of current national interest that could benefit greatly from the implementation of behavior learning techniques in parallel with optimal pursuit-evasion solutions. It was shown that although a behavior learning algorithm may not always converge on the exact behavior of an evader, it remains an effective way to give a player a tactical advantage over simply implementing a zero-sum strategy. Behavior learning provides a model to an opponent's strategy that can be used to predict their behavior and allow a player to turn a pursuit-evasion game into a one-sided optimal control problem.

An example pertaining to the minimum-time case was also studied. Although

the minimum-time formulation can become increasingly complex when multiple degrees-of-freedom are considered, the simple scalar example has merit due to the obvious application of minimum-time solutions and behavior learning to the missile interception and associated warfare problems.

IX.A. Chapter Summary

A motivational example was presented in Ch. I which illustrated how a player's performance diminishes as key information about the game and system is revoked. Chapter II identified the behavior learning aspects of incomplete information final-fixed-time games and extended those concepts to the uncertain information case. The role of behavior learning in infinite-horizon PE games was examined in Ch. III and techniques for the incomplete and uncertain information scenarios were presented.

Insight to behavior learning for final-time-free games was given in Ch. IV and the role of behavior learning was identified for minimum-time games. Chapter V presented a minimum-time behavior learning example and the utility for behavior learning for the minimum-time case. A two-step dynamic inversion method was presented in Ch. VI to allow for the use of these behavior learning methods for nonlinear, control-affine dynamic systems.

Chapter VII presented a spacecraft reorientation example which took on the form of an infinite-horizon pursuit-evasion game. Dynamic inversion was used to allow the nonlinear system to fit within the behavior learning framework that was developed. In the presence of significant modeling deficiencies, it was shown that although the behavior learning algorithms may not converge on the true solution

defining an opponent's strategy, it remains effective at modeling behavior and giving a player a tactical advantage. It was learned that behavior learning for the infinite-horizon case provides the best performance when used to implement a control augmentation at a single, early time during the PE game.

A missile interception example was provided in Ch. VIII which utilized a final-time-fixed pursuit-evasion game. Dynamic inversion was implemented to a system that could be deemed to be nonlinear or linear with a disturbance. The orientation state, ψ, was used as a control variable to allow for the system to fit within the control-affine framework of dynamic inversion as the relative states were transformed from the realistic space to a reduced space. Behavior learning was applied to the incomplete information case and was effective at reducing the total cost for the pursuer. Details for robust and repeatable implementation were outlined to account for the unstable nature of the Riccati equation when propagated forward in time.

IX.B. Limitations

A few limitations became apparent during the development of the behavior learning framework. One potential issue with any any filter-type implementation is observability. Observability should always be treated on a case-by-case basis. If observability becomes a concern, a simplified model can be used to obtain a representative model of the opponent's behavior. To do this, assumptions must be made about the opponent's behavior and all relevant knowledge about the system must be applied including gain matrix properties such as positive definiteness and diagonalness. Additionally, multiple model approaches can be used in conjunction with

different combinations of assumptions to produce multiple behavior learning solutions. A blended solution or the solution with the best statistical properties can then be used to augment a player's strategy.

It is important to exercise caution when applying two-step dynamic inversion. Based on the feasibility of the selection of desired dynamics, large control magnitudes can be experienced. It is also possible for system oscillations to occur if the natural response of the nonlinear and desired linear systems significantly disagree. Therefore, it is important that the desired linear dynamics are carefully selected based on the nonlinear system of interest. Oscillation and damping terms can be added to the desired system in effort to achieve an acceptable response.

As previously mentioned, infinite-horizon behavior learning can be exceptionally difficult to implement because the feedback gain K is fix for a player until they decide to augment their solution. Because of this fixed gain, if behavior learning is used multiple times throughout the game, the pursuer runs the risk of continuously driving the opponent further and further away while exhausting control input. The nature of infinite-horizon pursuit-evasion games dictates that behavior learning is best done at a single instance in time, as soon as a solution for the opponent's behavior is obtained. By doing so, the pursuer is able to quickly augment their strategy without sacrificing an excessive amount of control and driving the evader farther away at the same time. Infinite-horizon pursuit-evasion is not a type of differential game that is often examined giving these observations merit as the study of behavior learning evolves.

Finally, the forward propagation of the Riccati equation can cause instabilities.

In the final-time-fixed case where the forward propagation of the Riccati equation is necessary, it is essential to select valid gains for solution recomputation. Additional data processing proved useful in order to arrive at an acceptable solution that could be propagated forward in time without difficulty. One possible solution to the Riccati forward propagation issue is the implementation of the inverse Riccati equation.

The inverse Riccati equation may prove to be useful in some instances where instabilities are observed. Its use, however, does not necessarily eliminate all forward propagation issues. The inverse Riccati equation is useful for certain gain selections, but in many cases, can produce solutions for the independent elements of S which exhibit additional dynamic characteristics which are undesirable. Its use should also be exercised with caution. With that, it would be more convenient if a single forward propagator could be used reliably and handle all types of gain selections because the initial value of an opponent's strategy is simply a guess. An opponent's strategy could be defined by any number of gain combinations and for this reason, the implemented forward Riccati propagation with additional statistical analysis proved to be the most effective. The inverse Riccati equation could be used in addition to the method presented here and the best solution could be implemented for the control solution augmentation.

IX.C. Extensions

The behavior learning framework presented focused on the perspective of the pursuer. In each example, the pursuer was enabled with behavior learning such that it was able to gain a tactical advantage over the evader. Even though the evader con-

tinued to invoke a zero-sum safe strategy, there is nothing preventing the evader from also implementing its own behavior learning algorithm. Furthermore, both players can be enabled with behavior learning in an effort to study the effects of continuously evolving opponent models. The assumptions made about an opponent may need to be modified to account for an intelligent challenger with the same behavior learning abilities. Depending on the accuracy of the model and the frequency of opponent strategy augmentation, player performance could increase or decrease drastically. Behavior learning can be implemented for offensive or defensive purposes.

Multiple model solutions would allow opponent behavior to be modeled by a bank of possible objective functions. This could include external objectives - such as hitting a specified target - and could aid in determining an asset that an opponent is attempting to conquer. Different assumptions defining the strategy gains could be used to help determine what the opponent is most interested in. The behavior learning solution may be defined by a blended solution or it may become a function of the solution with the most appealing statistical characteristics such as the model with the lowest associated covariance values.

Behavior learning is highly applicable to pursuit-evasion teams or teams of vehicles. Specifically, teams of unmanned aerial vehicles who may be working cooperatively but who want to maintain minimum distances throughout a trajectory. If each teammate's behavior can be properly modeled then a more comprehensive forward propagation can be used to predict vehicle interaction throughout a flight path. This has direct applications to path planning for commercial aviation and has the potential to have significant impact once fully autonomous flight systems are adopted.

With pilot-in-the-loop applications, greater safety measures can be defined around high traffic areas such as airports during the take-off and landing processes.

Minimum-time applications are a rich topic and an extension of the major principles from the final-time-fixed and infinite-horizon scenarios can be applied further. This can become a complicated issue due to the presence of constraints and the desire for feedback solutions. Minimum-time is most applicable to military applications where a threat must be recognized and intercepted immediately so secondary defenses can be utilized if the initial line of defenses are unsuccessful at exterminating the threat.

Behavior learning has proven to be a powerful tool for pursuit-evasion games. It can be used to enhance the performance of a player in the presence of incomplete, imperfect, and uncertain information. Although some characteristics of the framework need to be exercised with care, the methods presented have a broad impact on aerospace applications of high national interest including spacecraft rendezvous and missile defense.

REFERENCES

[1] Issacs, R., *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*, John Wiley & Sons, Inc., New York, NY, 1965.

[2] Satak, N., *Behavior Learning in Differential Games and Reorientation Maneuvers*, Ph.D. thesis, Texas A&M University, 2013.

[3] Bryson Jr., A. and Ho, Y., *Applied Optimal Control: Optimization, Estimation, and Control*, Taylor & Francis, New York, NY, 1975.

[4] Lewis, F., *Optimal Control*, John Wiley & Sons, Inc., New York, NY, 1986.

[5] Watts, A., "Control of a High Beta Maneuvering Reentry Vehicle Using Dynamic Inversion," Tech. Rep. SAND2005-0498, Sandia National Laboratories, Albuquerque, NM, May 2005.

[6] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Thrun, S., *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA, 2005.

[7] Tang, C., "Differential Flatness-based Kinematic and Dynamic Control of a Differentially Driven Wheeled Mobile Robot," *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Guilin, China, December 2009, pp. 2267–2272.

[8] Satak, N., Cavalieri, K., Moody, C., Siddarth, A., Hurtado, J., and Sharma, R., "Experimental Investigations of Trajectory Guidance and Control for Differential Games," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, AK, July 31 - August 4 2011, pp. AAS–11–639.

[9] Issacs, R., "Differential Games I," Tech. Rep. RM-1391, RAND Corporation, 1954.

[10] Issacs, R., "Differential Games II," Tech. Rep. RM-1399, RAND Corporation, 1954.

[11] Issacs, R., "Differential Games III," Tech. Rep. RM-1411, RAND Corporation, 1954.

[12] Issacs, R., "Differential Games IV," Tech. Rep. RM-1468, RAND Corporation, 1955.

[13] Berger, J., *Pursuit-Evasion Differential Games*, Ph.D. thesis, Washington University, 1968.

[14] Ho, Y., Bryson, A., and Baron, S., "Differential Games and Optimal Pursuit-Evasion Strategies," Tech. Rep. 457, Office of Naval Research, November 1964.

[15] Isaacs, R., "Differential Games: Their Scope, Nature, and Future," *Journal of Optimization Theory and Applications*, Vol. 3, No. 5, 1969, pp. 283–295.

[16] Berkovitz, L. and W.H., F., "On Differential Games with Integral Payoff," *Annals of Methematics*, Vol. 1, No. 39, 1957, pp. 413–435, Princeton University.

[17] Starr, A. and Ho, Y., "Nonzero-sum Differential Games," *Journal of Optimization Theory and Applications*, Vol. 3, No. 3, 1969, pp. 184–206.

[18] Ho, Y., "Differential Games, Dynamic Optimization, and Generalized Control Theory," *Journal of Optimization Theory and Applications*, Vol. 6, No. 3, 1970, pp. 179–209.

[19] Shinar, J., "Solution Techniques for Realistic Pursuit-Evasion Games," Tech. Rep. AFWAL-TR-81-1114, Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, OH, September 1980.

[20] Bagchi, A. and Olsder, G., "Linear-Quadratic Stochastic Pursuit-Evasion Games," *Applied Mathematics and Optimization*, Vol. 7, 1981, pp. 95–123.

[21] Kumkov, S. and Patsko, V., "Optimal Strategies in a Pursuit Problem with Incomplete Information," *Journal of Applied Mathematics and Mechanics*, Vol. 59, No. 1, 1995, pp. 75–85.

[22] Li, D., Cruz Jr., J., and Schumacher, C., "Stochastic multi-player pursuit-evasion differential games," *International Journal of Robust and Nonlinear Control*, Vol. 18, 2008, pp. 218–247.

[23] Antoniades, A., Kim, H., and Sastry, S., "Pursuit-Evasion Strategies for Teams of Multiple Agents with Incomplete Information," *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, December 2003, pp. 756–761.

[24] Ho, Y., "On The Minimax Principle and Zero Sum Stochastic Differential Games," Tech. Rep. 630, Office of Naval Research, April 1972.

[25] Hermann, R. and Krener, A., "Nonlinear Controllability and Observability," *IEEE Transactions on Automatic Control*, Vol. 22, No. 5, 1977.

[26] Kalman, R., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, 1960, pp. 35–45.

[27] Smith, G., Schmidt, S., and McGee, L., "Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity On Board a Circumlunar Vehicle," Tech. Rep. R-135, NASA, 1962.

[28] McElhoe, B., "An Assessment of the Navigation and Course Corrections for a Manned Flyby of Mars or Venus," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-2, No. 4, 1966, pp. 613–623.

[29] Crassidis, J. and Junkins, J., *Optimal Estimation of Dynamic Systems*, CRC Press, New York, NY, 2012.

[30] Laub, A., "A Schur Method for Solving Algebraic Riccati Equations," *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 6, 1979, pp. 913–921.

[31] Marcos, A. and Balas, G., "Development of Linear-Parameter-Varying Models for Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 2004, pp. 218–228.

[32] Brunton, S., Rowley, C., and Williams, D., "Linear Unsteady Aerodynamics Models from Wind Tunnel Measurements," *AIAA Fluid Dynamics Conference and Exhibit*, Honolulu, HI, June 2011.

[33] Snell, S., Enns, D., and Garrard Jr., W., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 976–984.

[34] Reiner, J., Balas, G., and Garrard, W., "Robust Dynamic Inversion for Control of Highly Maneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 18–24.

[35] Schumacher, C. and Khargonekar, P., "Stability Analysis of a Missile Control System with a Dynamic Inversion Controller," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 3, 1998, pp. 508–515.

[36] McFarland, M. and Hoque, S., "Robustness of a Nonlinear Missile Autopilot Designed Using Dynamic Inversion," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August 14-17 2000, pp. AIAA–2000–3970.

[37] Chen, W., "Nonlinear Disturbance Observer-Enhanced Dynamic Inversion Control of Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, 2003, pp. 161–166.

[38] Hurtado, J., *Kinematic and Kinetic Principles*, Lulu Press, Morrisville, NC, 2012.

[39] Wong, R., "Some Aerospace Differential Games," *Journal of Spacecraft*, Vol. 4, No. 11, 1967, pp. 1460–1465.

[40] Menon, P., Calise, A., and Leung, S., "Guidance Laws for Spacecraft Pursuit-Evasion and Rendezvous," *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, August 15-17 1988, pp. 688–697.

[41] Hurtado, J., *Elements of Spacecraft Control*, Lulu Press, Morrisville, NC, 2009.

[42] Gutman, S., "On Optimal Guidance for Homing Missiles," *Journal of Guidance and Control*, Vol. 2, No. 4, 1979, pp. 296–300.

[43] Shinar, J. and Shima, T., "A Game Theoretical Interceptor Guidance Law for Ballistic Missile Defense," *Proceedings of the IEEE Conference on Decision and Control*, Kobe, Japan, December 1996, pp. 2780–2785.

[44] Shinar, J. and Shima, T., "Nonorthodox Guidance Law Development Approach for Intercepting Maneuvering Targets," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 658–666.

[45] Turetsky, V. and Shinar, J., "Missile guidance laws based on pursuit-evasion game formulations," *Automatica*, Vol. 39, 2003, pp. 607–618.

[46] Shinar, J. and Gutman, S., "Three-Dimensional Optimal Pursuit and Evasion with Bounded Controls," *IEEE Transactions on Automatic Control*, Vol. 25, No. 3, 1980, pp. 492–496.

[47] Shima, T. and Shinar, J., "Time-Varying Linear Pursuit-Evasion Games Models with Bounded Controls," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, 2002, pp. 425–432.

[48] Shinar, J., Shima, T., and Kebke, A., "On the Validity of Linearized Analysis in the Interception of Reentry Vehicles," *Proceedings of the AIAA*, 1998, pp. 1050–1060.

[49] Shima, T., Oshman, Y., and Shinar, J., "Efficient Multiple Model Adaptive Estimation in Ballistic Missile Interception," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 667–675.

[50] Shaferman, V. and Shima, T., "Cooperative Multiple-Model Adaptive Guidance for an Aircraft Defending Missile," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, 2010, pp. 1801–1813.

[51] Perelman, A., Shima, T., and Rusnak, I., "Cooperative Differential Games Strategies for Active Aircraft Protection from a Homing Missile," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 3, 2011, pp. 761–773.

[52] Shima, T., "Cooperative Evasion and Pursuit of Aircraft Protection," Tech. rep., Air Force Office of Scientific Research, 2012.

[53] Norris, G., *SpaceX Unveils Plans To Be Worlds Top Rocket Maker*, Aviation Week, August 2011.