Title

Deception in two-player zero-sum stochastic games : theory and application to warfare games

Permalink

https://escholarship.org/uc/item/9m2366jx

Author

Singh, Rajdeep

Publication Date

2006

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Deception in two-player zero-sum stochastic games: Theory and application to warfare games.

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Rajdeep Singh

Committee in charge:

Professor William M. McEneaney, Chair
Professor Robert R. Bitmead
Professor Patrick J. Fizsimmons
Professor J. William Helton
Professor Miroslav Krstić

2006

The dissertation of Rajdeep Singh is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2006

To the mysteries of mind and beyond.

TABLE OF CONTENTS

LIST OF SYMBOLS

$\mathbb{R}$	the real numbers
a, b, c etc.	scalars
$\mathbf{q}$	column matrix.
$\mathbf{A}, \mathbf{B}, \mathbf{C}$, etc.	matrices
$\mathbf{A}^T$	transpose of a matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of a nonsingular matrix $\mathbf{A}$
$\mathbf{A}^{-T}$	transpose of the inverse of A.
$\left(\begin{array}{c} n \\ k \end{array} \right)$	number of combinations choosing k out of n.
Blue	the player with partial state information.
Red	the omniscient player (with perfect state information).
$\mathbf{1_A(b)}$	indicator function; 1 if $b \in A$.
$\mathbf{U}, \mathbf{W}$	finite control sets for Blue and Red.
$\mathcal{X}$	finite set of discrete states
Y	finite set of observation.
$\mathbf{\Omega}$	sample space
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	random variables/vectors.
$\mathbf{T}$	terminal time for the game.
Λ, Θ	strategy sets for Blue and Red respectively.
λ, θ	a strategy for Blue and Red respectively (a mapping to open loop or state feedback control space).
u, w etc.	open loop controls for Blue and Red, i.e. $u \in U$.
$u_{[s,r)}$	the sequence $\{u_s, u_s + 1, ..., u_r - 1\}$.
$\vec{w}$	a state-feedback control for Red player, i.e. $\vec{w}_i \in W$.
$V_t(x)$	value corresponding to the state $x \in \mathcal{X}$.
$\mathcal{E}(x)$	terminal payoff for state $x \in \mathcal{X}$.
$\mathbf{E}\{V(\mathbf{X_T})\}$	mathematical expectation of $V(\mathbf{X_T})$.
$\mathbf{E}\{V(\mathbf{X_T}) \vert \mathbf{X_t} = x\}$	mathematical conditional expectation of $V(\mathbf{X})$ given $\mathbf{X_t} = x$.

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

I would like to express my gratitude for everyone who has been instrumental in shaping my life and career which today has led me to the culmination of my Thesis.

Foremost, I would like to thank my Advisor Professor William M. McEneaney for his consistent support and guidance in helping me with all aspects of graduate studies. His patience and commitment to my research was invaluable and instrumental in finishing this project. He was supportive and critical in all phases of my graduate studies. My experience as his student would equally help me in other professional and social endeavors of life. I am indebted to him for playing the role of my Advisor in true sense of the word.

I would also like to express my appreciation for my thesis committee members: Professors Miroslav Krstić, Robert R. Bitmead, J. William Helton, and Patrick J. Fitzsimmons. I had the opportunity to learn from all of them during various courses they instructed. I am also thankful to them for helping me in shaping the final form of my thesis and their valuable suggestions.

I sincerely appreciate the efforts of all my teachers and instructors for providing all the essential tools and the knowledge which will help me in successfully pursuing any future work. A special mention for the staff of the Mechanical Engineering and Mathematics departments for their technical and administrative support.

I would like to pay a special rememberance to my late grandparents Takhat Singh Ji and Mohinder Kaur Ji who loved me till their last breath. Though i never saw my maternal late grandfather, Prem Singh Ji, i have always felt his blessing for me and my family through my maternal grandmother Satwant Kaur Ji who continues to be an incessant source of love for us.

My Uncle, Dr. Hakam Singh Ji, has been a source of inspiration and support for my educational endeavor. I thank him and my Aunt Harbans Singh Ji for their blessings and love. I am grateful to my Uncle Harender J Singh Ji and Aunt Maninder Kaur Ji, for always being there for me and my family and my cousins Simran J Singh and Dishveen Kaur for their love and care. I would like to acknowledge the support and encouragement from my parents-in-law Amarpaul Singh Ji, Gurdeep Kaur Ji and my brother-in-law Gurpal Singh for keeping me in good spirits. My sister-in-law Dr. Jaswinder Kaur has been very loving and supportive and also helped me in editing this document.

I would also like to thank my elders, Dalip Singh Ji, Dharamveer Kaur Ji, Narenderjeet Singh Ji, Charanjeet Singh Ji, Inderjeet Singh Ji, Tejinder Pal Singh Ji, Jasbir Kaur Ji, and Rajinder Kaur Ji for their blessings. My dear cousins, Surjeet Singh, Kuljeet Kaur, Gurvinder Kaur, Gharpreet Singh, Manpreet Singh, and Harleen Kaur, are few names, of many, who have inspired me in some way or another. My close friends, Karanvir Singh, Sumit Mehrotra, and Dr. Vishal Sood, deserve a mention for their faith and confidence in me. My colleagues and friends, Dr. Sangho Ko, Chengjin, Jaspreet Singh, Matt, Charles, Jun Yan, Mahir, Sankar, Dr. Indrakanty Sastry and Justin made my stay at UCSD comfortable and memorable. This list can continue forever as I have met most wonderful people, made great friends and have been blessed with the most loving and supportive kin. I sincerely thank them as I have learned from each one of them and cherished the moments we have spent together.

Finally, with deep sense of humility, i am very excited to acknowledge the support and many sacrifice made by my father Manjeet Singh Ji and my mother Gurcharan Kaur Ji to help me fulfill all my dreams. I have learned the most from them about the importance of family and faith. I thank my brothers Harpreet Singh, Charanpreet Singh, Simran J Singh and Damanpreet Singh, who have given me very fond and happy memories of my childhood. My 'incredible' wife Charanpreet Kaur (Nonu) has been very patient, has helped me in staying calm and brought endless joy and love into my life. Life would be incomplete without my loving family. I dedicate this work with deep respect to my parents who are my eternal source of inspiration.

This dissertation includes the reprints of the following papers:

Chapter 2 is in part a reprint of:

Rajdeep Singh, William M. McEneaney - *Robustness to Deception*, Chapter 2.4 in the book "Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind", CRC Press, To appear.

Rajdeep Singh, William M. McEneaney - *Unmanned vehicle decision making under imperfect information in an adversarial environment*, AIAA Journal of Guidance Navigation and Control, in preparation.

Chapter 3 is in part a reprint of:

Rajdeep Singh, William M. McEneaney - *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, CRC press, To appear.

Rajdeep Singh, William M. McEneaney - *Unmanned vehicle decision making under imperfect information in an adversarial environment.* AIAA Journal of Guidance Navigation and Control, in preparation.

Rajdeep Singh, William M. McEneaney - *Unmanned Vehicle Operations under Imperfect Information in an Adversarial Environment*, AIAA Guidance, Navigation, and Control Conference and Exhibit, 2004.

Rajdeep Singh, William M. McEneaney - *Unmanned Vehicle Operations: Countering Imperfect Information in an Adversarial Environment*, AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit AIAA, 2004.

Chapter 4 is in part a reprint of:

Rajdeep Singh, William M. McEneaney - *Deception in Autonomous Vehicle Decision Making in an Adversarial Environment*, AIAA Guidance, Navigation, and Control Conference and Exhibit, 2005.

Rajdeep Singh, William M. McEneaney - *Deception-Enabled Control in Stochastic Games with Autonomous Vehicle Applications*, Sixth SIAM control Conference, 2005.

Rajdeep Singh - *Unmanned Vehicle Decision Making Under Imperfect Information in an Adversarial Environment II*, AIAA Guidance, Navigation, and Control Conference and Exhibit, 2005.

Rajdeep Singh, William M. McEneaney - *Exploitation of an Opponents Imperfect Information in a Stochastic Game with Autonomous Vehicle Application*, 43rd IEEE Conference on Decision and Control, 2004.

Chapter 5 is in part a reprint of:

Rajdeep Singh, William M. McEneaney - *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, CRC press, To appear.

The dissertation author was the primary author and the co-author listed in these publications directed and supervised the research.

VITA

1995-1999	B.E. Punjab Engineering College, Chandigarh, India.
1995-1999	University Scholarship for Undergraduate Studies.
1995-1999	Defense Scholarship for Undergraduate Studies.
1999	University Gold medal for Excellence in Undergraduate Studies.
1999–2000	Junior Scientist, Defence Research Development Organization, India.
2001–2004	Teaching Assistant, Department of Mechanical Engineering, University of California San Diego, USA.
2002-2005	Research Assistant, Department of Mechanical Engineering, University of California San Diego, USA.
2003	M.S. (Mechanical Engineering), University of California San Diego, USA.
2003–2005	Senior Teaching Assistant, Department of Mechanical Engineering, University of California San Diego, USA.
2004	Summer Intern, Tempest Technologies, Los Angeles, USA.
2004	C. Phil. (Mechanical Engineering), University of California San Diego, USA.
2004-2005	President - Graduate Student Representative Department of Mechanical Engineering, University of California San Diego, USA.
2005-2006	Intern, Senior Technology Specialist Information Assurance Group Orincon/Lockheed Martin, La Jolla, USA.
2006	Ph.D. (Mechanical Engineering), University of California San Diego, USA.

PUBLICATIONS

Journal Papers

1. Rajdeep Singh, William M. McEneaney
   Unmanned vehicle decision making under imperfect information in an adversarial environment.
   AIAA Guidance Navigation and Control, in preparation.

Conference Papers

1. Rajdeep Singh, William M. McEneaney
   Deception in Autonomous Vehicle Decision Making in an Adversarial Environment.
   AIAA Guidance, Navigation, and Control Conference and Exhibit
   San Francisco, California, Aug. 15-18, 2005.

2. Rajdeep Singh
   Unmanned Vehicle Decision Making Under Imperfect Information in an Adversarial Environment II
   AIAA Guidance, Navigation, and Control Conference and Exhibit
   San Francisco, California, Aug. 15-18, 2005.

3. Rajdeep Singh, William M. McEneaney
   Deception-Enabled Control in Stochastic Games with Autonomous Vehicle Applications
   SIAM Control Conference, New Orleans, Louisiana, July. 11-15, 2005.

4. Rajdeep Singh, William M. McEneaney
   Exploitation of an Opponents Imperfect Information in a Stochastic Game with Autonomous Vehicle Application
   43rd IEEE Conference on Decision and Control, Bahamas, Dec. 14-17, 2004.

5. Rajdeep Singh, William M. McEneaney
   Unmanned Vehicle Operations under Imperfect Information in an Adversarial Environment
   AIAA Guidance, Navigation, and Control Conference and Exhibit
   Providence, Rhode Island, Aug. 16-19, 2004.

6. Rajdeep Singh, William M. McEneaney
   Unmanned Vehicle Operations: Countering Imperfect Information in an Adversarial Environment
   AIAA 3rd ”Unmanned Unlimited” Technical Conference, Workshop and Exhibit
   Chicago, Illinois, Sep. 20-23, 2004.

Book Chapters

1. Rajdeep Singh, William M. McEneaney
   Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind, CRC Press, To appear.

FIELDS OF STUDY

Major Field: Engineering (Mechanical Engineering)

Studies in Control and Estimation.
Professors Robert R. Bitmead, Miroslav Krstić, William M. McEneaney, Robert E. Skelton

Studies in Fluid Mechanics.
Professors Juan Lasheras, Paul F. Linden

Studies in Numerical Methods.
Professors Thomas Bewley, Constantine Pozrikidis

Studies in Mathematics.
Professors Patrick Fitzsimmons, J. William Helton, William M. McEneaney, Linda Rothschild

ABSTRACT OF THE DISSERTATION

Deception in two-player zero-sum stochastic games:

Theory and application to warfare games.

by

Rajdeep Singh

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2006

Professor William M. McEneaney, Chair

In this work, two-player zero-sum stochastic games, under imperfect information, are investigated in the discrete-time/discrete-state case. We focus on the case where only one player, Blue, has incomplete or partial information and the other player, Red, has complete state information. In stochastic games with partial information the Information State is a function of a conditional probability distribution. In the problem form here, the payoff is only a function of the terminal state of the system, and the initial information state is a max-plus sum of max-plus delta functions. The Blue player can achieve robustness to the effect of Red's control on its observations. Using the recently established deception-robust theory, we demonstrate that the full state-feedback optimal control applied at the Maximum Likelihood State ('MLS') is not optimal for the Blue player in a partially-observed game and hence the Certainty Equivalence Principle does not hold. An automated deception-enabled control algorithm is derived for the Red player with an assumption that Red can model the Blue algorithm completely. An example game is used to demonstrate that even for the Red player, with complete state information, the optimal control is not the state-feedback optimal control. A future study of deception-enabled Red approach is proposed in the mixed strategy framework. Lastly, some modelling ideas are presented for Urban Warfare. The example cases considered in this study are simple enough to allow an intuitive understanding of optimal strategies, while complex enough to demonstrate real-world difficulties. The theory discussed here is more general than the specific application which has been presented owing to the critical nature of imperfect information and hence its utility in war games.

Chapter 1

Introduction

1.1 Motivation

In order to motivate the research done in this dissertation, we note that in recent years, computational aids have greatly stepped-up the pace of military operations. This pace will only increase further with increasing reliance on autonomous vehicles. This stepped-up pace of operations has, in turn, led to efforts in the development of decision aids appropriate to problems in this domain. A fundamental aspect of such problems is the presence of an adversary. Consequently, many of the efforts in this domain are making use of game theoretic methods (D.P. Bertsekas & Logan 1999, J.B. Cruz 2000, D. Ghose & Shamma 2000, Heise & Morse 2000, Jelinek & Godbole 2000), (McEneaney, Lauko & Fitzpatrick 2004), among many notable others. Further, a great number of these problems have stochastic as well as game-theoretic components. It is well-known that human decision makers have found deception and counter-deception to be extremely valuable tools. In fact, the imperfect information aspect of these problems is often a critical factor.

Stochastic games with imperfect information have also been studied more specifically in the pursuit-evasion type games, (P. Bernhard 1987) looks into a game where the evader (rabbit) makes no observations and the pursuer (hunter) has consequently no means to deceive the evader. In another pursuit-evasion type game with partial information for both players, (Olsder & Papavassilopoulos 1988), the evader doesn't control the observations. There, the pursuer can light up its current node location and two adjacent nodes, one on each side, on the movement grid along the circumference of a circle. If the evader falls in the lighted zone, the game is over. When the pursuer uses this control choice (called a searchlight) to search for the evader, it discloses its own position by default. This leaves no scope for deceiving the evader. In these games either the adversarial noise is missing or there is no observation process for the evader. Dynamic stochastic games, with imperfect information for only one player, who's observations have adversarial noise are even harder. In (Dinah Rosenberg & Vieille 2004), a different class of partial information is discussed with the payoff matrix being unknown to one player.

In the last ten years, there has been a substantial effort in the application of automated reasoning techniques to problems in the military Command and Control arena. In these efforts, it has become increasingly clear that the lack of perfect information and deception play critical roles in the development of useful Command and Control strategies. Of course, these play important roles in many other areas. However, it was a Command and Control application that happened to provide the impetus for the work to be described in this dissertation. Although these issues had arisen earlier, see for example (P. Bernhard 1987, Basar & Olsder 1982, Olsder & Papavassilopoulos 1988), but the substantial growth in computational power in the intervening years has allowed one to address these problems much more satisfactorily.

In (Swarup & Speyer 2004) theory for a system taking values in the continuum rather than in a discrete set is developed but it is restricted to linear systems. That restriction allows one to obtain certain elegant results which are not generalizable to the problem form considered here. There is a similarity in that the controls obtained there are specifically constructed to handle potential deception, but there are tremendous problem-complexity reductions which are induced by the linearity.

Control under partial information involves three components. The first consists of the accumulation of observational data up to the current moment and the construction of an abstract object which condenses this data into a form useful to the controller. The second consists in the determination of the effects of control choices on the expected (broadly defined) future costs. The third and last is the component which combines the output of the other two components in a way that yields the optimal (again broadly defined) choice of control at the current moment. The object which is obtained by the first component is generally referred to as the *information state.* It depends only on the past. The object obtained by the second component is generally referred to as the *value function,* and maps current states into future costs. Thus, the third component is combining these past and future objects to obtain the best decision in the present.

One very natural way to address the control problem under partial information (and here we also use the term control broadly to indicate also the decision process in a game problem), is to estimate the current state of the system, and then apply the optimal control that one has determined for that state. In linear, stochastic problems with quadratic cost measures (most notably, the linear/quadratic regulator), this does in fact yield the optimal control given the current available observational data. However, there are very few problems outside of that example for which this approach yields the optimal control decision. On the other hand, until recently the computations required to obtain the mathematically demonstrable optimal controller have been too excessive for real-time applications for most problems. Further, for problems that are not "too" nonlinear, the above heuristic approach has yielded an acceptable (and often quite good) controller.

Unfortunately, most real-world adversarial problems do not fit within the category of problems which are well-handled by that heuristic approach. Rather, the problems are often strongly nonlinear, and, importantly, have an opponent who may be attempting to cause you to make an incorrect decision through its influence on your observation process. Interestingly however, for a class of linear adversarial problems, one may find that a very low-complexity information state can still obtain the optimal control; see (Swarup & Speyer 2004). In such problems an information state which is simply a single state estimate cannot contain enough information to make an optimal control decision. One must be able to evaluate the alternatives based on the (past) data up to the moment and the (future) measures of the total cost. States which may "seem" unlikely based on the observations, but which pose large benefits for the opponent may be important in deciding which action to take.

The estimation of ground-truth in the presence of deception is quite difficult. At best, one might obtain a measure of the likelihood (loosely defined) that a deception is being employed. However, this does not imply that one cannot determine an effective course of action regardless of the lack of certainty with regard to whether the opponent is employing deception. How should one act if deception is suspected but cannot be ascertained? In this dissertation, we consider how one can reduce the susceptibility to deception, i.e., how one can choose one's actions so that the impact of a deception is minimized. We refer to this reduction in susceptibility to deception as deception-robustness. We also consider the flip side, where we could deceive our opponent (with partial information) when we have the perfect information, by use of our controls on the opponent's observations. We refer to automated controllers that use deception, when it is useful, as deception-enabled ('DE') controllers.

In order to study this problem properly, we must have some mathematical model of adversarial conflict. This model must encompass both the inputs of an intelligent adversary and the necessarily unpredictable outcomes of low-level actions in a conflict. Thus, the natural framework for study of this problem is that of stochastic games. We will be interested only in *dynamic* games (i.e. time-dependent systems), rather than static games. We will be discussing deceptions which critically rely on the lack of perfect observations of the system. Thus, we are led to the realm of stochastic games under imperfect information (also referred to in the literature as stochastic games under partial information). The bulk of the theory is independent of whether the system state takes values in a discrete space or in the continuum. However, in order to develop tractable algorithms, we concentrate on the case of systems with a finite number of discrete states. We will also assume that these systems operate in discrete-time. This implies that the underlying models of the state dynamics will be discrete-time Markov chains. We will suppose that there are two players in these games. Consequently, the transition probabilities for the Markov chains will be controlled by the actions of the two players. The player with partial information, (who we attempt to assist with a deception-robust or 'DR' control) is designated as Blue, and its opponent (with complete information) is designated as Red.

As noted above, the classes of deceptions we study here will rely on Blue's imperfect knowledge of the state of the system. This implies that we must model the Blue player, for which we are developing our deception-robust controller, as obtaining its information from an observation process (which will also be combined with some initial estimates). On the other hand, we choose not to model the Red player as also obtaining its information from its own observation process. One of the reasons for this simplification is the presence of serious mathematical roadblocks to solution of such games in the case where the information available to each of the players is different and where one player's knowledge does not necessarily completely subsume the other's. Further, the case where the opponent has perfect knowledge of the system is clearly the most demanding. If the opponent has partial and corrupted knowledge, then the achieved results will be more favorable than predicted. Lastly, we note that we will consider only zero-sum games. This is the case when the opponent is choosing its actions to maximize whatever criterion it is that we wish to minimize. One can make a compelling argument that the opponent will generally not have a diametrically opposed goal. However, if the opponent is choosing its actions based on a goal other than the diametrically opposed goal, then the expected outcome of the game (from our perspective) will be no worse than what we predict under the zero-sum assumption. We will also refer to this formulation as a minimax formulation since we will be minimizing the maximum (worst-case) expected outcome.

The theoretical underpinning of the methods developed and discussed in this dissertation have origins in a particular branch of game theory and nonlinear, risk-sensitive, stochastic control. (Fleming 1964, Friedman 1971, Elliott & Kalton 1972) developed the notion of value for dynamic games in the 1960's and early 1970's. In the 1980's, nonlinear H_∞ control was developed. Most importantly, a representation in terms of dynamic games was formed; see (Basar & Bernhard 1991). Soon after, in the early 1990's, risk-sensitive control was developed (subsuming stochastic control and robust/H_∞ control), and was found to have an equivalent representation as a stochastic, dynamic game; see (Fleming & McEneaney 1995, Fleming & McEneaney 1992*a*, Fleming & McEneaney 1992*b*, James 1992, Runolfsson 1993). These were first developed as state-feedback control. James et al. (James & Baras 1996, James & Yuliar 1995) extended this to observation-feedback. (Basar & Bernhard 1991) also obtained a similar result. In the robust/H_∞ limit approach to observation-feedback, the concept of information state as a worst-case cost over potential opponent inputs was a critical breakthrough

(Basar & Bernhard 1991, Helton & James 1999). The risk-sensitive approach in this dissertation is a direct descendent of the above referenced risk-sensitive control theory. The deception-robust approach significantly generalizes the above information state from a class of problems with only deterministic noise inputs to the stochastic game realm.

In (João Hespanha 2000), it is demonstrated that deception can be a useful strategy. In the game considered there, there is a single observational event followed by a single dynamical event. In this dissertation, we extend that to a multi-step problem. We also follow a slightly different (but related) approach to deal with the scenario of partial information. In particular, we provide a general mathematical framework from the viewpoint of the player who lacks perfect information. Also, in contrast to (João Hespanha 2000), which deals with mixed strategies for a given problem, we build a theoretical framework using pure strategies dependent on the information patterns of the players. This framework is applied to a problem which is similar in nature to the problem of (João Hespanha 2000) in that the events and states of the system take values in finite sets. One other notable variation in the application used here is that we add decoys and false alarm observations to the problem as well.

Deception-Robust theory has already been established from the Blue player's perspective in a partially observed game set up (McEneaney 2004). A natural motivation is to explore the merits of using a complex control algorithm like the deception-robust theory over the standard approach that uses the Certainty Equivalence Principle. The deception-enabled theory developed in this dissertation is to provide an impetus to similar work from the Red player's perspective. Naturally so, with military applications serving as a major motivation, we also venture into modelling some contemporary war game situation as an the application of the deception-robust and the deception-enabled technology developed in this work.

The overall structure and contributions of this research are as follows.

1.2 Dissertation Overview

In chapter 2, we concentrate on two-player zero-sum stochastic games in discrete space ($\mathcal{X}$) and discrete-time domain, where both players have complete information (full state-feedback). A terminal time game is formulated with the state transition following a markov-chain process

$$p_{ij}(u,w) = \Pr(X_{t+1} = j \mid X_t = i, u_t = u, w_t = w). \tag{1.1}$$

A strategy set is defined for each player and using a Basar-Olsder type, lower value definition

$$V_{\bar{t}}(x) = \sup_{\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}} \inf_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}[\mathcal{E}(X_T) \mid X_{\bar{t}} = x]$$

we derive the the dynamic programming equation (DPE)

$$V_t(x) = \max_{w \in W} \min_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1}) | X_t = x],\ \forall\ \bar{t} \leq t < T$$

where U and W are the finite control sets for Blue and Red respectively and $X_t \in \mathcal{X}$. The optimal controls are then obtained using the DPE for both the players. There is no observation process, so no deception is possible in this case. In section 2.3, a terminal time game example, called the Masked Attack Game (MAG), is formulated with appropriate definitions of U and W. We solve two cases, one where the state-transition is only controlled by the Blue player, and another where both players controls affect the state-transition as defined in (1.1). The optimal control are then obtained for the Blue and the Red player in both the cases. The example is set up such that a saddle point does exist, or in other words the minmax value is the same as the maxmin value ,

$$\max_{w \in W} \min_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1}) | X_t = x] = \min_{u \in U} \max_{w \in W} \mathbf{E}[V_{t+1}(X_{t+1}) | X_t = x],\ \forall\ \bar{t} \leq t < T.$$

The analysis of the MAG example concludes this chapter with insights into the additional complexity one might expect in the partially-observed game.

In Chapter 3, the observation process for the Blue player (the one with partial information) is included in the dynamics and various Blue approaches are outlined using a stochastic Red control modelling. The main approaches discussed include the certainty equivalent control (alternatively referred here as the Maximum Likelihood State or the 'MLS' control) and a heuristic based on the equivalence between risk-sensitive stochastic control and stochastic games called the risk-averse approach. Then the deception-robust theory is outlined from (McEneaney 2004) and a refined form of the information state, without maximization over the Red state feedback controls $\vec{w} \in W^n$, is defined (without any change in the final results). The information state is the maximal cost over the space of feasible conditional probability distributions on the state.

$$\mathcal{I}_t(q; u_{[0,t)}, y_{[0,t)}) \doteq \begin{cases} \sup_{q_0 \in Q_0^{q,u_{[0,t)}}} \mathcal{I}_0(q_0) & \text{if } q \in Q_t(u_{[0,t)}, y_{[0,t)}); \\ -\infty & \text{otherwise.} \end{cases} \tag{1.2}$$

The information state propagation and some robustness results given in (McEneaney 2004) are derived again using the definition of the information state given by (1.2). A robust control for the Blue player obtained as in (McEneaney 2004) is given here as the deception-robust control for the partially observed game.

$$u_t^m \doteq \operatorname*{argmin}_{u \in U} \left[\max_{q \in Q(\mathcal{X})} \{ \mathcal{I}_t(q) + L_t(q,u) \} \right]. \tag{1.3}$$

We then discuss the MAG example in the partially-observed game set-up in section 3.4. We present simulation results with different levels of information and compare the 'MLS' approach with the deception-robust approach to assert the robustness property of the controller given by (1.3). Then we present an overall comparison between the performance of the approaches outlined in section 3.2 to the deception-robust approach discussed in section 3.3. The focus is then primarily shifted to the deception-robust approach with comparative simulation results being presented, whenever appropriate, to highlight its advantages compared to the other approaches. Section 3.5.1 discusses the initialization of $\mathcal{I}$, where we assume that $\mathcal{I}_0$ takes the form of a max-plus sum of max-plus delta functions, where ϕ is a (finite) max-plus sum of max-plus delta functions if there exist $\{q_k\}_{k=1}^K$ such that

$$\phi(q) = \bigoplus_{k=1}^{K} \phi_k(q) = \max_k \phi_k(q). \tag{1.4}$$

Discussion on how to initialize the $\{q_k\}_{k=1}^K$ is followed by a study that highlights the value of any knowledge on the initial Red state or Red control on the performance of the

Blue player. The exponential growth of Q_t is shown to be reasonably contained using the simple methods, referred to here as pruning, in section 3.5.2. Finally, mismodelling studies for parameters that have an effect on the observation process of the Blue player, namely observation of stealthy entities and decoys/false alarms, are given in section 3.6 to illustrate the robustness properties of the control given by (1.3) and the sensitivity of the ‘MLS’ and the ’Risk-Averse’ approach.

Chapter 4 looks at deception-enabled controllers for the Red player under the strong assumption

$$\text{The Red player knows } (Q_0, \{y_r\}_{r=0}^{t}) \text{ and the Blue control algorithm.} \tag{A-RI}$$

The Red player maximizes the expected terminal cost to obtain the optimal control

$$\vec{w}^* \in \operatorname*{argmax}_{\vec{w} \in W^n} \mathbf{E}[W_{t+1}(X, q) | X_t = x, q_t = \tilde{q}]. \tag{1.5}$$

With appropriate modelling assumptions applied to the MAG example game, the simulation results using (1.5) are presented in section 4.2. Section 4.3 provides discussion on the obvious issues of mismodelling when assumption (A-RI) does not hold. In the same section, we discuss an example where Red does not employ any internal Blue model as a motivation for potential future research; studying deception-enabled Red controllers in the partially-observed set up with a mixed-strategy Red control.

Finally, in chapter 5, modelling issues related to simulating an urban warfare game are outlined. Discussion on path planning and attrition modelling (with emphasis on computing most of the data offline) is followed by the computation of an approximate (or heuristic) value and optimal controls for both players, in the state-feedback case. This chapter concludes with discussion on some results obtained using simulated gaming environment (automated control computations), representative of real-world behavior.

1.3 Contributions

The contributions of this dissertation can be summarized as follows:

(1) Complete-Information Game: State Feedback (Chapter 2)

- Derived the Dynamic Programming Equation (DPE) for two-player zero-sum stochastic game with complete-information state-feedback.
- Analysis of the MAG, with complete information, to elucidate the complexity one expects in the partially-observed stochastic games.

(2) Blue Approach in the Partially-Observed Game (Chapter 3)

- Refined definition of Information state, $\mathcal{I}_t$, in partially-observed games, with observation based information for Blue and complete information for Red. This definition is without maximization over the state feedback Red controls, $\vec{w} \in W^n$ (see 3.3).
- Derived information state propagation results with the new definition and robustness properties of the proposed minimizing control u^m (as given in equation (3.53)).
- Application of deception-robust theory (McEneaney 2004) to the MAG example and analysis to confirm the optimality of the proposed deception-robust controller given by (3.53) (sections 3.4 and 3.6).
- Analysis of initialization for the MAG example, where the information state takes the form of max-plus sum of max-plus delta functions or choosing $q \in \tilde{Q}_0^{\phi}$ (section 3.5.1). Analysis of Intel about Red initial state and control set W^n and the resulting performance advantage to the Blue player.
- Proposed simplistic and efficient pruning methods to contain the exponential growth of Q_t and provided analysis for the MAG example with discussion on error tolerance and computational speeds (section 3.5.2).
- Mismodelling analysis of the parameters that directly affect the information dissemination to the Blue player in the MAG example. Confirmation of the deception-robustness properties using simulation results and consequently the utility of deception-robust approach to war games (section 3.6).

(3) Red Approach in the Partially-Observed Game (Chapter 4)

- Derived an optimal (potentially, deception-enabled) control for the Red player with assumption of having complete knowledge about the Blue control computation (section 4.1).

- Confirmation that the control given by equation (1.5) (or (4.8)) is indeed optimal (and deception-enabled) for the MAG example and assertion of the sub-optimality of the Red state-feedback optimal control for the partially-observed game.
- Mismodelling analysis for the MAG example when Red's internal Blue control approach mismatches the actual Blue approach. Construction of an example to motivate research for Red deception-enabled control as a mixed strategy (without assuming any knowledge or modelling of Blue control algorithm).

(4) Urban Warfare Modelling (Chapter 5)

- Construction of a model for an urban warfare game (in state-feedback) and demonstration of real-world strategies like 'Feint' and 'Protect' in a simulated environment with the proposed modelling (section 5.0.2). This also serves an an underlying exercise to provide test-beds for any future application of deception-enabled and deception-robust theory to Urban Warfare.

Chapter 2

Complete-Information Game: State Feedback

We first formulate the state-feedback zero-sum game between two players where both players have complete information of the state of the system. Each player can be a single entity or a group with common interests and goals. The objectives can be problem specific but would be generally antagonistic and hence we utilize a single cost function which the Blue player is trying to minimize and the Red player is trying to maximize. Since the state is completely known to both the players, there is no initial cost (owing to obfuscating the information state of the player with partial information). We only have terminal cost (reflecting the accomplishment of goals of both players). There is no running cost as the terminal payoff will be an ensemble cost (alternatively called the payoff) of the loss or gain made by each player during the course of interactions between the initial and the final time. In particular, as an example one can think of a war game where an objective could be to take over a high-value strategic target. The cost incurred during the course of such action is generally reflected by the accomplishment of this goal at the end of the game.

2.1 Problem Formulation and State Dynamics

Potential states of the system will be represented by $x \in \mathcal{X}$ where $\mathcal{X}$ is some finite set. Let time t take values in $\bar{\mathbf{T}}$, where $\bar{\mathbf{T}} = \{\bar{t}, \bar{t}+1, ., \ldots, T\}$ and where $\bar{t}$ is the initial time of the game. The state of the system at time t will be denoted by X_t. Each state x will be associated with a unit basis vector in $\mathbf{R}^n$, where $n \doteq (\#\mathcal{X})$. We suppose that the state evolves as a controlled Markov process. Let the probability that $X_{t+1} = j$, given $X_t = i$ with controls $u_t = u$ and $w_t = w$ be

$$p_{ij}(u, w) = \Pr(X_{t+1} = j | X_t = i, u_t = u, w_t = w) \tag{2.1}$$

and let the $n \times n$ matrix of the elements p_{ij} be denoted as $P(u, w)$ where $n \doteq \#\mathcal{X}$. The state X_t propagates as a Markov chain with probabilities given by (2.1). For the state-feedback case, each player control decision is based on the complete state information, $X_t \in \mathcal{X}$, at current time t. Let U and W be the finite sets of open loop Blue and Red controls respectively. Then the Blue player state-feedback control $\vec{u}_t \in U^n$ and the state-feedback Red controls $\vec{w}_t \in W^n$ (W^n being the outer product of W, n times). Let a conditional probability (in absence of observations) of the state at time t be denoted by $q_t \in Q(\mathcal{X})$. For given controls $\vec{u}_t, \vec{w}_t$, the probability distribution propagates according to

$$q_{t+1} = \tilde{P}^T(\vec{u}_t, \vec{w}_t) q_t \tag{2.2}$$

Note that though the mapping $\tilde{P}$ is into $Q(\mathcal{X})$, it is not necessarily onto. Also note that in the above propagation, appropriate components of $\vec{u}_t$ and $\vec{w}_t$ are used, i.e.

$$\tilde{P}_{ij}(\vec{u}_t, \vec{w}_t) = P_{ij}([\vec{u}_t]_i, [\vec{w}]_i). \tag{2.3}$$

2.2 Strategies, Value Function, Dynamic Programming Equation and Optimal State-Feedback Control

We now define the strategies for both players. The strategies for Red for the state-feedback case are defined as follows.

$$\Theta_{[\bar{t},T)} = \left\{ \theta_{[\bar{t},T)} : \mathcal{X}^{T-\bar{t}} \to W^{(T-\bar{t})}, \text{n.a} \right\} \tag{2.4}$$

Note that $\theta_{[\bar{t},T)}$ is n.a. (nonanticipative) if given any $t \in \bar{\mathbf{T}}$, ${X^1}_{[\bar{t},T)} \in \mathcal{X}^{T-\bar{t}}$, and ${X^2}_{[\bar{t},T)} \in \mathcal{X}^{T-\bar{t}}$ such that $X^1_r = X^2_r,\ \forall\ \bar{t} \leq r \leq t$, then

$$\theta_t[X^1_{[\bar{t},T)}] = \theta_t[X^2_{[\bar{t},T)}].$$

Note that in defining the strategies we use the fact that both the players have complete state knowledge at any given time. Thus, setting the domain of the strategy to include the state knowledge till current time t, the range is just the sequence of open loop controls (in the appropriate time domain). The set of strategies for Blue is defined similarly,

$$\Lambda_{[\bar{t},T)} = \Big\{ \lambda_{[\bar{t},T)} : \mathcal{X}^{T-\bar{t}} \times W^{T-\bar{t}} \rightarrow U^{(T-\bar{t})}, \text{n.a} \Big\}. \qquad (2.5)$$

where non-anticipativeness of $\lambda_{[\bar{t},T)}$ is defined similarly. In particular, $\lambda_{[\bar{t},T)}$ is n.a. (nonanticipative) if given any $t \in \bar{\mathbf{T}}$, $({X^1}_{[\bar{t},T)}, w^1_{[\bar{t},T)}) \in (\mathcal{X}^{T-\bar{t}} \times W^{T-\bar{t}})$, and $({X^2}_{[\bar{t},T)}, w^2_{[\bar{t},T)}) \in (\mathcal{X}^{T-\bar{t}} \times W^{T-\bar{t}})$ such that $X^1_r = X^2_r$ and $w^1_r = w^2_r$, $\forall\ \bar{t} \leq r \leq t$, then

$$\lambda_t[X^1_{[\bar{t},T)}, w^1_{[\bar{t},T)}] = \lambda_t[X^2_{[\bar{t},T)}, w^2_{[\bar{t},T)}]$$

Note that if the Red player chooses $w_{[\bar{t},T)}$ using a strategy $\theta_{[\bar{t},T)} \in \Theta_{\bar{t},T}$, then $\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}$ is n.a., if given any $t \in \bar{\mathbf{T}}$, $X^1_{[\bar{t},T)} \in \mathcal{X}^{T-\bar{t}}$, and $X^2_{[\bar{t},T)} \in \mathcal{X}^{T-\bar{t}}$ such that $X^1_r = X^2_r$, $\forall\ \bar{t} \leq r \leq t$, we have

$$\lambda_t[X^1_{[\bar{t},T)}, w^1_{[\bar{t},T)}] = \lambda_t[X^2_{[\bar{t},T)}, w^2_{[\bar{t},T)}]$$

where by non-anticipativeness of $\theta_{[\bar{t},T)}$, $\forall\ t \leq r \leq t$ one has:

$$w^1_r \doteq \theta_r(X^1_{[\bar{t},T)}) = \theta_r(X^2_{[\bar{t},T)}) \doteq w^2_r$$

It implies that the state process $X_{[\bar{t},T)}$ is sufficient (Red control history $w_{[\bar{t},T)}$ is not required) for an alternate definition of non-anticipativeness of $\lambda_{[\bar{t},T)}$. A simple plot to explicate this notion is given in (2.1).

The terminal cost (or payoff) is $\mathcal{E} : \mathcal{X} \rightarrow \mathbf{R}$; the cost or the payoff at terminal state $x \in \mathcal{X}$ is $\mathcal{E}(x)$. This motivates the Blue player to use strategies such that the terminal state X_T yields the lowest possible payoff and Red player will try to achieve the exact opposite.

For the state-feedback (complete-information) game, we define the lower value of the Basar-Olsder type (Basar & Bernhard 1991), and (Basar & Olsder 1982). Define

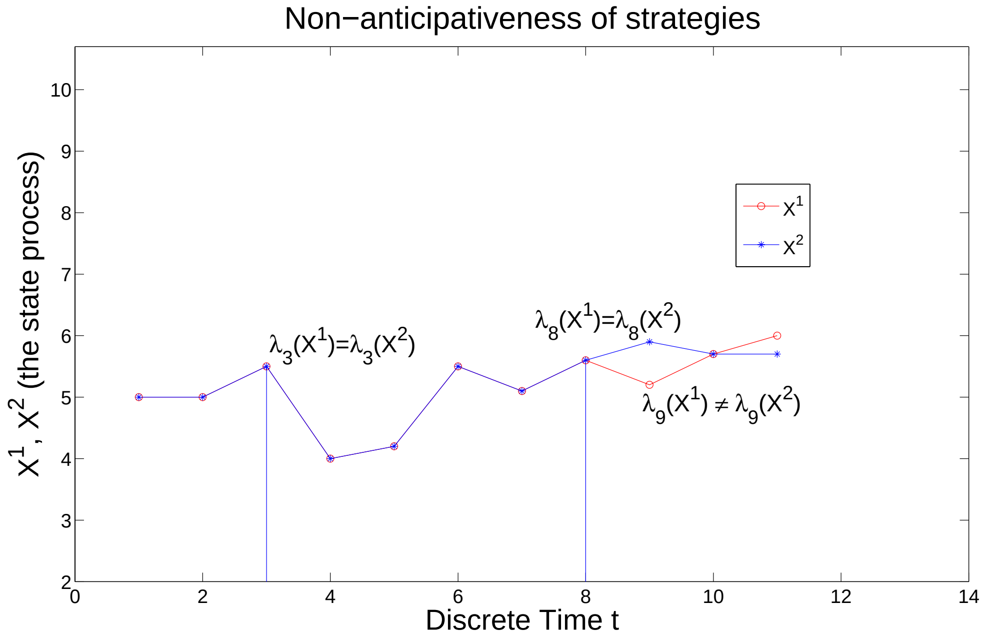


Figure 2.1: Non-anticipativeness of a strategy

the value function for state-feedback game (SFG) as :

$$V_{\bar{t}}(x) = \sup_{\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}} \inf_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}[\mathcal{E}(X_T) \,|\, X_{\bar{t}} = x] \tag{2.6}$$

The state-feedback value function at the terminal time is $V_T(x) = \mathcal{E}(x)$. Hence we obtain,

$$V_{\bar{t}}(x) = \sup_{\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}} \inf_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}[V_T(X_T) \,|\, X_{\bar{t}} = x] \tag{2.7}$$

Note that $\mathcal{X}$ and W are finite and $\theta_{[\bar{t},T)} : \mathcal{X}^{T-\bar{t}} \to W^{T-\bar{t}}$. So $\theta_{[\bar{t},T)}$ maps a finite number of state processes $n^{T-\bar{t}}$ to a finite number of control processes $N_w^{T-\bar{t}}$, where recall that $n \doteq \#\mathcal{X}$ and $N_w \doteq \#W$. Also, $\lambda_{[\bar{t},T)}$ maps a finite number of state process and Red control process combinations $(nN_w)^{T-\bar{t}}$ to a finite number of control processes $N_u^{T-\bar{t}}$, where $N_u \doteq \#U$. This allows us to use min for infimum and max for supremum and we get

$$V_{\bar{t}}(x) = \max_{\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}} \min_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}[\mathcal{E}(X_T) \,|\, X_{\bar{t}} = x] \tag{2.8}$$

Note that X represents the random state process propagated by $\theta_{[\bar{t},T)}$ and $\lambda_{[\bar{t},T)}$. We now present some basic results for deriving the DPE (Dynamic Programming Equation).

Lemma 2.2.1. *Let's fix $\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}$, and for any $\bar{t} < t < T$, let $\lambda_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}$, and $\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}$, where*

$$\Lambda_{[\bar{t},t)} = \Big\{ \lambda_{[\bar{t},t)} : \mathcal{X}^{t-\bar{t}} \times W^{t-\bar{t}} \to U^{t-\bar{t}}, n.a \Big\} \tag{2.9}$$

and

$$\Lambda_{[t,T)} = \Big\{ \lambda_{[t,T)} : \mathcal{X}^{T-t} \times W^{T-t} \to U^{T-t}, n.a \Big\}. \tag{2.10}$$

Define

$$\lambda^*{}_s = \begin{cases} \lambda_s & \mathit{if}\, s < t; \\ \overline{\lambda}_s & \mathit{if}\, s \geq\ t. \end{cases} \tag{2.11}$$

Then $\lambda^*{}_{[\bar{t},T)}$ is also a strategy; $\lambda^*{}_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}$. We will denote the construction in the above sense by $\lambda_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}$.

Proof. See Appendix (2.4.1). □

Corollary 2.2.1. *Let $\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}$. Let, $\tilde{\lambda}_s = \lambda_s$, if $s < t$, and for any given any $X_{[\bar{t},t)} \in \mathcal{X}^{t-\bar{t}}$ and $\bar{X}_{[t,T)} \in \mathcal{X}^{T-t}$, let $\overline{\lambda}_s^{X_{[\bar{t},t)}} \left[\bar{X}_{[t,T)}\right] \doteq \lambda_s \left[X_{[\bar{t},t)} \bigcup \bar{X}_{[t,T)}\right]$. Then $\tilde{\lambda} \in \Lambda_{[\bar{t},t)}$ and $\overline{\lambda}^{X_{[\bar{t},t)}} \in \Lambda_{[t,T)}$.*

Proof. See Appendix (2.4.2). □

Similar results also hold true for the Red player strategy $\theta_{[\bar{t},T)}$.

Lemma 2.2.2. *Fix $\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}$. Given any $x \in \mathcal{X}$ and $\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}$, let $F(x, \lambda_{[\bar{t},T)}) \doteq \mathbf{E}[\mathcal{E}(X_T) \,|\, X_{\bar{t}} = x]$, where the state process $X_.$ is propagated using $\lambda_{[\bar{t},T)}$ and $\theta_{[\bar{t},T)}$ and initial condition $X_{\bar{t}} = x$, then*

$$\min_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} F(x, \lambda_{[\bar{t},T)}) = \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \ \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} F(x, \tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \tag{2.12}$$

where on the right hand side the state process $X_.$ is propagated by $\tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}_{[\bar{t},T)}$ and $\theta_{[\bar{t},T)}$ with initial condition $X_{\bar{t}} = x$.

Proof. See Appendix (2.4.3). □

Lemma 2.2.3. *Fix any $\bar{t} < t < T$, $x \in \mathcal{X}$, Choose any $\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}$ and any $\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}$, then*

$$S(x, \theta_{[\bar{t},T)}, \tilde{\lambda}_{[\bar{t},t)}) = R(x, \theta_{[\bar{t},T)}, \tilde{\lambda}_{[\bar{t},t)}) \tag{2.13}$$

where S and R are defined as below

$$S \doteq \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} \mathbf{E}[\mathbf{E}[\mathcal{E}(\bar{X}_T)|\bar{X}_t = X_t]|\ X_{\bar{t}} = x] \tag{2.14}$$

$$R \doteq \mathbf{E}[\min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} \mathbf{E}[\mathcal{E}(\bar{X}_T)|\bar{X}_t = X_t]|\ X_{\bar{t}} = x] \tag{2.15}$$

The state process $X_{[\bar{t},t]}$ is propagated by $\theta_{[\bar{t},t)}$ and $\tilde{\lambda}_{[\bar{t},t)}$, with initial condition $X_{\bar{t}} = x$ and $\bar{X}_{[t,T]}$ is propagated by $\theta_{[t,T)}$ and $\overline{\lambda}_{[t,T)}$, with initial condition $\bar{X}_t \doteq X_t$. The arguments of S and R are dropped for space constraints. Note that $\theta_{[t,T)}$ has implicit dependence on the state process $X_{[\bar{t},t)}$ as defined in corollary 2.2.1.

Proof. See Appendix (2.4.4). □

We now prove the Dynamic Programming Principle. Let us define

$$N_{\bar{t}}(x, \theta_{[\bar{t},T)}) \doteq \min_{\lambda_{[\bar{t},T)}} \mathbf{E}[\mathcal{E}(X_T)\,|\,X_{\bar{t}} = x] \tag{2.16}$$

so that

$$V_{\bar{t}}(x) = \max_{\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}} N_{\bar{t}}(x, \theta_{[\bar{t},T)}) \doteq N_{\bar{t}}(x, \theta^o_{[\bar{t},T)}) \tag{2.17}$$

where

$$\theta^o_{[\bar{t},T)} \in \operatorname*{argmax}_{\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}} N_{\bar{t}}(x, \theta_{[\bar{t},T)}). \tag{2.18}$$

Lemma 2.2.4. *For any $\bar{t} \leq t < T$,*

$$V_{\bar{t}}(x) = N_{\bar{t}}(x, \theta^o_{[\bar{t},T)}) = \tilde{N}_{[\bar{t},t)}(x, \tilde{\theta}^o_{[\bar{t},t)}) \tag{2.19}$$

where

$$\tilde{N}_{[\bar{t},t)}(x, \theta_{[\bar{t},t)}) = \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \mathbf{E}[V_t(X_t)\,|\,X_{\bar{t}} = x] \tag{2.20}$$

and

$$\tilde{\theta}^o_{[\bar{t},t)} \in \operatorname*{argmax}_{\theta_{[\bar{t},t)}} \tilde{N}_{[\bar{t},t)}(x, \theta_{[\bar{t},t)}) \tag{2.21}$$

so that

$$\tilde{N}_{[\bar{t},t)}(x, \tilde{\theta}^o_{[\bar{t},t)}) = \max_{\theta_{[\bar{t},t)}} \min_{\tilde{\lambda}_{[\bar{t},t)}} \mathbf{E}[V_t(X_t)|\,X_{\bar{t}} = x] \tag{2.22}$$

Proof. See Appendix (2.4.5). □

Next we specialize this result and obtain the One-Step DPE.

Theorem 2.2.1. *For any $\bar{t} \leq t < T$, let $V_t(x)$ be as given by (2.19), then,*

$$V_t(x) = \max_{w \in W} \min_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = x]. \tag{2.23}$$

Proof. See Appendix 2.4.6. □

Note that (2.23), also gives the optimal controller $u^{*,w}$, and w^* for the complete-information state-feedback game) for each time $\bar{t} \leq t < T$. In particular:

$$w_t^* \in \operatorname*{argmax}_{w \in W} \min_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = x]. \tag{2.24}$$

and

$$u_t^*(w_t) \in \operatorname*{argmin}_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = x] \ , \forall w_t \in W. \tag{2.25}$$

In the max-min set-up, the Blue player has advantage being the inside player. For specific games (like in the upcoming example in the later sections) saddle-point existence may allow using max-min and min-max alternatively.

2.3 The Masked Attack Game (MAG)

The remainder of this chapter will be devoted primarily to application of the state-feedback theory to a seemingly simple example game. However, as we will see in section 3.4, once one introduces the partial information and deception components, determination of the best (or even nearly best) strategy becomes quite far from obvious. The following terminology (used in the remainder of this chapter) may require a little elaboration:

- Entity: controllable objects, e.g., tanks and unmanned aerial vehicles.
- Attrition: Damage caused by one of the sides to the entities belonging to another.
- C^2 : Command and Control, the process by which the opponents guide their entities in the battle.

- Intel: Information that one side obtains by observing the territory and entities of the other.
- Asset: An object of high value. (In this example, the term asset will only be used to designate certain stationary Blue objects — not the Red and Blue entities.)
- Decoy: An inexpensive imitation of an entity, without combat capability.
- Stealth: Use of camouflage or other means to avoid detection.
- UAV: Unmanned or Uninhabited Air Vehicle.
- UCAV: Unmanned or Uninhabited Combat Air Vehicle, capable of attacking the opponent entities.

We will assume without loss of generality that the initial time $\bar{t} = 0$. We refer to our example as the Masked Attack Game (MAG). A snapshot from the game is depicted in Figure 2.2. In the MAG example, the Red player is attempting to take (or, equivalently from the perspective of the game, destroy) a valuable Blue asset while Blue will attempt to interdict the Red advance(s). In the partially-observed game Red can use stealth and decoys to obscure the direction from which the attack will occur. However, in the complete-information case Blue has full-state knowledge, so the affect of using stealth is almost redundant (unless we allow for the attrition or damage caused by Blue on the Red entities to be dependent on stealth).

In particular, we consider an example where Red and Blue have only a handful of forces, and the attack(s) can come along only two routes. This MAG example is complex enough to demonstrate many of the issues that appear when applying this technology. At the same time, it is simple enough so that technical complications do not muddy the picture excessively.

Objective

In this game the Red player has four ground entities (say, tanks) and the Blue player has two UCAVs. The objective of the Red player is to capture the high-value Blue assets by moving at least one non-decoy Red entity to a Blue asset location by the terminal time, T. The Blue player uses the UCAVs to interdict and destroy the moving Red entities and prevent them from reaching the Blue assets by the terminal time. Winning and losing are measured in terms of the total cost (equivalently, the score or the payoff) at the pre-specified terminal time.

The payoff at terminal time is computed as follows: each Red surviving entity costs Blue 1 point and if Blue loses any (or both) of the high-value assets to Red (amounting to at least one non-decoy Red entity reaching the asset by the terminal time) it costs Blue 20 points. Suppose at time T, for a state $x \in \mathcal{X}$, we denote the total number of surviving Red entities on both routes as $x^{R,s}$. Recall that Red "takes" the assets by successfully moving (non-decoy) alive entities to their targets.

$$J(x) = \begin{cases} 20 + x^{R,s} & \text{if } x^{R,s} > 0; \\ 0 & \text{otherwise.} \end{cases} \tag{2.26}$$

So in the best case scenario, Blue achieves a payoff of 0 ($x^{R,s} = 0$) or in the worst case scenario a payoff of 24, ($20 + x^{R,s}$, where $x^{R,s} = 4$, at terminal time). There is no running cost. The running cost in an example such as this would refer to the cost of specific control processes used up to the current time t, for example, the cost of using the decoys, or/and the fuel cost when moving the UCAVs from one route to another.

Dynamics

Red entities move at the same speed independent of being stealthy or non-stealthy and independent of the route (uniform terrain). Red entities are not allowed to switch routes during the game and do not have any attrition capability against the Blue UCAVs (hence the dynamic update is dependent only on the Blue control). Blue UCAVs require at least two time steps to travel from one route to the other.

The simulation snapshot from the partial-information game in Figure 2.2, is taken between time steps 1 and 2 from the graphic for a MATLAB simulation that runs the example game. Blue's base is at the bottom of the figure. Red is depicted with a base at the top of the figure. The two rectangular shapes on either side of the Blue base represent the positions of two high-value assets belonging to Blue. The dashed lines are meant to indicate routes that the Red entities (depicted as tanks) could take

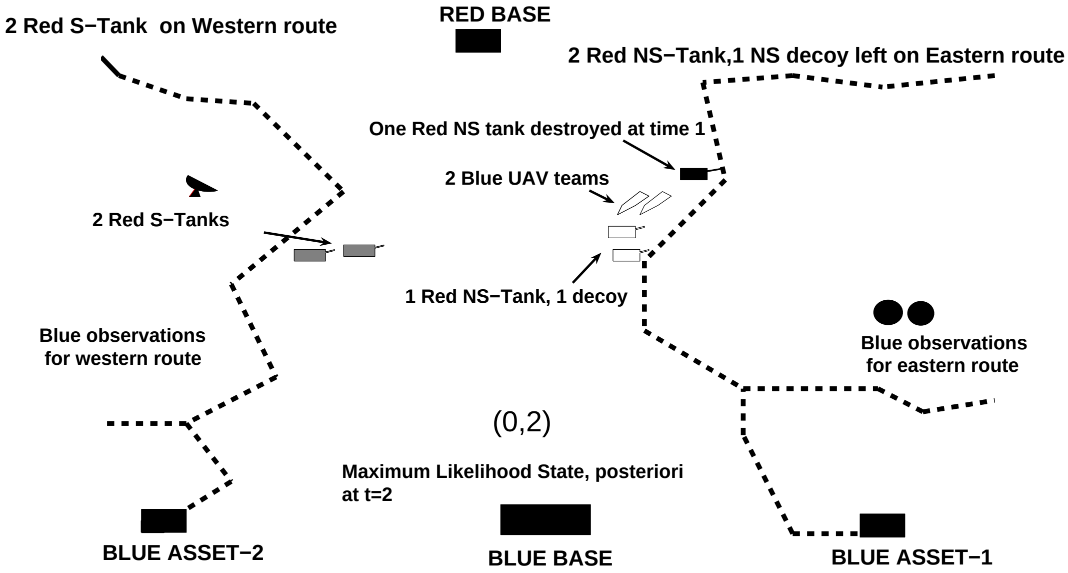

2 Red S−Tank on Western route
RED BASE
2 Red NS−Tank,1 NS decoy left on Eastern route
One Red NS tank destroyed at time 1
2 Blue UAV teams
2 Red S−Tanks
1 Red NS−Tank, 1 decoy
Blue observations for western route
Blue observations for eastern route
(0,2)
Maximum Likelihood State, posteriori at t=2
BLUE ASSET−2
BLUE BASE
BLUE ASSET−1

Figure 2.2: A snapshot from the Matlab simulation

toward each of the Blue high-value assets. The Red player is moving four ground entities from the Red base toward two Blue assets. These entities move along either an eastern route or a western route depending on which Blue asset they are attacking. (There may be Red entities moving along both routes, and it is assumed that there are simply two groups of Red entities, those moving along the eastern route and those moving along the western route.) At the time of the snapshot in Figure 2.2, Red was in the process of non-stealthily moving one tank and one decoy tank toward the eastern Blue asset, and two tanks (their gray color indicating stealthy movement) toward the western Blue asset. The black tank icon along the eastern route road indicates that at time 1 the tank on that route (which had been operating non-stealthily) was destroyed.

Obviously, Red entities may move stealthily or non-stealthily (with no affect on information in the state-feedback case). In partial-information game scenario, Red entities are detected more easily when they are non-stealthy. Blue has two attack UCAVs, which may be assigned to attack Red entities on either route, individually or in tandem. In the figure, the missile shaped icons moving along the eastern road indicate Blue

UCAVs which are currently attempting to intercept the Red entities moving along that route. Blue's UCAVs are typically expected to be more effective against the Red entities when they are moving in the non-stealthy mode (although for study purposes, we mainly include results where the effectiveness against stealthy and non-stealthy entities are identical once Blue has decided to attack them). Further, there is a fixed travel-time for the UCAVs to move from one route to the other. At each time step, Blue must decide how to assign its UCAVs, while Red decides which Red entities to make stealthy (and whether to employ a decoy in the partial-information game scenario). Red also initially decides how to partition its ground entities between the two routes; this partition remains in effect throughout the game. The healths of the Red entities will transition as a discrete-time Markov chain, where the transition probabilities depend on whether they are under attack by zero, one or two Blue UCAVs. We should note that Red "takes" an asset by successfully moving at least one non-decoy entity to the asset (while Blue UCAVs provide resistance by intercepting the Red entities). The state transition probabilities can be affected by the actions of both players. Note that in the partial-information game, the observation probabilities can also be affected by the controls of both players. The current Blue observations are also shown for both routes in Figure 2.2. In this snapshot, the annotations indicate that it so happens that Blue has detected both the Red entities on the eastern route (one of which is a decoy) and detected nothing on the western route. The 'MLS' estimate, i.e., the naive estimate, is also indicated. Blue can chose to apply the optimal state-feedback control corresponding to the 'MLS' estimate as the control for the partially-observed game. We will call this naive approach, employing the Certainty Equivalence Principle, as the 'MLS' approach. We use the above game discussion to first provide the state-feedback solution of the MAG example.

Controls

Let the state X_t be decomposed into the Red and the Blue components as (X_t^R, X_t^B). In the MAG example problem introduced in the previous section, Red entities move at a fixed rate, independent of stealth, and so Red entity positions are not included in the state. We define the Red state component X_t^R as the pair indicating the number of Red *surviving* on each route at any time t, $X_t^R = (r_t^1, r_t^2)$, with $r_t^1 <= r_0^1$ and $r_t^2 <= r_0^2$.

This definition of Red state component subsumes the health component of the state (only allowing two health states, 'OK' or 'Destroyed'). For the Blue UCAVS, the health component is redundant as we don't allow the Red teams to cause any attrition on the Blue player. We now define the open loop control sets, U and W. Let $U = [1, 2, 3, 4, 5, 6]$, then $u \in U$, where:

- u=1: Send both UCAVs to the western route.
- u=2: Send both UCAVs to the middle/neutral zone (no attack).
- u=3: Send both UCAVs to the eastern route.
- u=4: Send one UCAV to the western route and move one to the middle/neutral zone.
- u=5: Send one UCAV to the eastern route and move one to the middle/neutral zone.
- u=6: Send one UCAV to the eastern route and one to the western route.

The Blue state definition is simply derived from the Blue control options. Blue decision is to pick a route to send the Blue UCAVs for the next time step. The Blue state X_t^B is an index indicating the positions of the two UCAVs (similar to the Blue control definition) as defined above. Given $X_t = x$ (known to both players), the Blue state at the next time step is simply the Blue control at current time step, $X_{t+1}^B = u_t$, where $u_t \in U$, which implies that $X_t^B \in U$. Let

$$\mathcal{X}^{\mathcal{N}} \doteq \{x \in \mathcal{X} : x_{t+1}^B \neq u_t\}$$

be the set of states which do not correspond to the deterministic Blue state transition (given the Blue control u_t), then we have for $\bar{x} \in \mathcal{X}^{\mathcal{N}}$,

$$\Pr(X_{t+1} = \bar{x} \mid X_t = x, u_t, w_t) = 0.$$

We now define the Red control set W. Recall that Red initially partitions its forces along the two routes and this partition remains in effect throughout the game.

The set of Red controls W is given by

$$W = \{\bar{w}^1, \bar{w}^2, \bar{w}^3, \bar{w}^4\}$$

where individual controls $\bar{w}_i \in W$ have the following meaning:

- $\bar{w}^1 = (S, S)$: Red entities on both routes operate stealthily.
- $\bar{w}^2 = (S, N)$: Red entities on the western route operate stealthily, and those on the eastern route operate non-stealthily.
- $\bar{w}^3 = (N, S)$: Red entities on the western route operate non-stealthily, and those on the eastern route operate stealthily.
- $\bar{w}^4 = (N, N)$: Red entities on both routes operate non-stealthily.

Notation and Parameters

The main parameters employed in the simulation study to follow are:

- p_2^N: Probability of a Red entity being destroyed when attacked by both UCAVs and when Red is non-stealthy.
- p_1^N: Probability of a Red entity being destroyed when attacked by one UCAV and when Red is non-stealthy.
- p_2^S: Probability of a Red entity being destroyed when attacked by both UCAVs and when Red is stealthy.
- p_1^S: Probability of a Red entity being destroyed when attacked by one UCAV and when Red is stealthy.
- α^1: p_1^N/p_2^N or p_1^S/p_2^S, ratio of attrition caused by 1 UCAV relative to attrition caused by 2 UCAVs.
- $r_r{=}r_0^1/r_0^2$, ratio of the (asymmetrical) initial Red entity distribution on the two routes.

Attrition Model

The attrition model used in this example is a binomial model. We outline the model by denoting the attrition probability (or equivalently state transition probability) on each side as a function of Blue and Red control. At any time t, given a state $X_t = (X_t^R, X_t^B) = \{(i_1, i_2), i_3\})$, one can compute the optimal w and u using (2.24) and (2.25) respectively. For notational simplicity we define some subsets of U and W for constructing the attrition probability model. For example, U_L^2 denotes the set of control elements $u \in U$ which correspond to two UCAVs attacking the western (left) route.

- $U_L^2 = [1]$, $U_L^1 = [4, 6]$, and $U_L^0 = [2, 3, 5]$
- $U_R^2 = [3]$, $U_R^1 = [5, 6]$, and $U_R^0 = [1, 2, 4]$

Similarly for the Red player we define subsets of W (with Red controls as elements). For example, W_L^s denotes the set of control elements $w \in W$ which correspond to the Red entities on the left route being stealthy.

- $W_L^s = [\bar{w}^1, \bar{w}^2]$ and $W_L^n = [\bar{w}^3, \bar{w}^4]$
- $W_R^s = [\bar{w}^1, \bar{w}^3]$ and $W_R^n = [\bar{w}^2, \bar{w}^4]$

Given a Blue control u, the attrition probability for a non-stealthy Red on the left (right) side then becomes:

$$p_L^N(u) \doteq \mathbf{1}_{U_L^2}(u) p_2^N + \mathbf{1}_{U_L^1}(u) p_1^N$$

$$p_R^N(u) \doteq \mathbf{1}_{U_R^2}(u) p_2^N + \mathbf{1}_{U_R^1}(u) p_1^N$$

Similarly the attrition probability for a stealthy Red on the left (right) side given u is computed as:

$$p_L^S(u) \doteq \mathbf{1}_{U_L^2}(u) p_2^S + \mathbf{1}_{U_L^1}(u) p_1^S$$

$$p_R^S(u) \doteq \mathbf{1}_{U_R^2}(u) p_2^S + \mathbf{1}_{U_R^1}(u) p_1^S$$

Then the attrition probability for a Red on the left (right) side given u and w becomes:

$$p^L(u, w) = \mathbf{1}_{W_L^s}(w) p_L^S(u) + \mathbf{1}_{W_L^n}(w) p_L^N(u)$$

$$p^R(u,w) = \mathbf{1}_{W_R^s}(w)p_R^S(u) + \mathbf{1}_{W_R^n}(w)p_R^N(u).$$

Note that the Blue control u_t leads to X_t^B deterministically, so that

$$\mathrm{Pr}_{[X_t \to X_{t+1}]}(u,w) \doteq \mathrm{Pr}_{[X_t^R \to X_{t+1}^R]}(u,w).$$

Also the attrition caused by Blue on one route is independent of the attrition caused by Blue on the other route. Given $X_t^R = (i_1, i_2)$ and $X_{t+1}^R = (j_1, j_2)$ we get the following attrition (or state-transition) modelling :

$$\mathrm{Pr}_{[X_t^R \to X_{t+1}^R]}(u,w) \doteq \mathrm{Pr}_{[i_1 \to j_1, i_2 \to j_2]}(u,w) = \mathrm{Pr}_{[i_1 \to j_1]}(u,w)\mathrm{Pr}_{[i_2 \to j_2]}(u,w)$$

$$\mathrm{Pr}_{[X_t^R \to X_{t+1}^R]}(u,w) = \left[\begin{pmatrix} i_1 \\ j_1 \end{pmatrix} (p^L)^{i_1 - j_1}(1-p^L)^{j_1}\right]\left[\begin{pmatrix} i_2 \\ j_2 \end{pmatrix} (p^R)^{i_2 - j_2}(1-p^R)^{j_2}\right]$$

where we have used the attrition probabilities $p^L(u,w)$ and $p^R(u,w)$ without the control arguments for space constraints.

Notes on Strategy

We will use the following notation :

- A_D: When attrition is dependent on the stealthiness of Red entities (or $p_2^N > p_2^S$). Red control in this case is to choose an initial state X_0^R and controls w_t for $t \in [0,1,...,T-1]$.
- A_I: When attrition is independent on the stealthiness of Red entities (or $p_2^N = p_2^S$). Red control in this case is only to choose an initial state X_0^R.

For the state-feedback case it is worth noting that in this example, a saddle point solution does exist, i.e

$$V_t(x) \doteq \min_{u \in U} \max_{w \in W} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = x] = \max_{w \in W} \min_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = x]. \qquad (2.27)$$

Let's consider the A_I case first. Red can choose to set up an initial distribution of its total forces, r_0, which Blue is aware of in the complete-information case. The Red control option is redundant in this case, as there is no difference in choosing a side to be stealthy or not (in fact all possible control choices for Red ,$\bar{w} \in W$, are equivalent).

Both the min-max and max-min games (in the full state-feedback case) are reduced to a minimization problem for the Blue player i.e. :

$$\min_{u\in U}\max_{w\in W}\mathbf{E}[V_{t+1}(X_{t+1})|X_t=x]=\min_{u\in U}\mathbf{E}[V_{t+1}(X_{t+1})|X_t=x].$$

$$\max_{w\in W}\min_{u\in U}\mathbf{E}[V_{t+1}(X_{t+1})|X_t=x]=\min_{u\in U}\mathbf{E}[V_{t+1}(X_{t+1})|X_t=x].$$

For the A_I case all the control choices are equivalent. We will refer to any $\bar{w}^k \in W$ as the complete-information state-feedback optimal control and $\bar{w}^1$ will be the complete-information state-feedback deceptive control (will be appropriate when used in the partially-observed game set up).

Let's consider the A_D case now. Here again, Red can choose to set up an initial distribution of its total forces, r_0, which Blue is aware of in the complete-information game. The Red control option is trivial in this case as well (but not redundant). The expected number of surviving Red in this case depends on the attrition caused by Blue which depends on the stealth factor. Intuitively, the best option for the Red player is to turn the entities stealthy on the route under attack or both routes (irrespective of the min-max or the max-min case) since there is no running cost. We explicate this intuitive Red behavior next. For the max-min game this implies that since Red is moving first, it will always choose controls only based on the Red state at time t, X_t^R. Let's denote the set with Red states of the form $(0,b)$ as R^R, the set with Red states of the form $(a,0)$ as R^L, and the set with Red states of the form (a,b) as R^B The control choice for Red would be:

$$w_t^o=\begin{cases}\{\bar{w}^1,\bar{w}^3\} & \text{if } x\in R^R;\\ \{\bar{w}^1,\bar{w}^2\} & \text{if } x\in R^L;\\ \{\bar{w}^1\} & \text{if } x\in R^B;\end{cases} \tag{2.28}$$

Clearly the Red control choice $\bar{w}^1$ is optimal for all cases. So, in the max-min game for the example in this study we have the Red player choosing the stealthy option for both sides. This reduces the Blue optimal control computation to

$$u_t^o(w_t^o)=\operatorname*{argmin}_{u\in U}\mathbf{E}[V_{t+1}(X_{t+1})|X_t=x].$$

For the min-max game, since Red is moving second, it will choose controls based explicitly on the Red state at time t, X_t^R, and the Blue control u_t^o (hence implicitly

Table 2.1: State-Feedback Optimal Red control, minmax, A_D case

Set of Red states	Blue player optimal control u_t^*					
	1	2	3	4	5	6
R^L	W_L^s	W	W	W_L^s	W	W_L^s
R^R	W	W	W_R^s	W	W_R^s	W_R^s
R^B	W_L^s	W	W_R^s	W_L^s	W_R^s	$W_L^s \cap W_R^s$

dependent on the Blue components of the state also, X_t^B). The optimal control for the Blue player is

$$u_t^o = \underset{u \in U}{\operatorname{argmin}} \max_{w \in W} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = x].$$

The optimal Red choice $w_t^o(u_t^o)$ in this case would then be given by the Table 2.1. For example if $X_t^R = x$, where $x \in R^L$, then the optimal Red control is to turn the left side stealthy, $w_t^o \in W_L^s$, when it's under attack on that route, i.e, when $u \in \{1, 4, 6\}$. For $u \in \{2, 3, 5\}$ ($x \in R_L$), any of the $\bar{w}^k \in W$ is optimal.

So in this min-max game also, the Red player's optimal choice is to use the stealthy option for both sides (or the side with at least one surviving Red entity). Red will achieve the same expected payoff with any of the optimal choices, when multiple optimal Red controls are available. The min-max game (in the complete-information state-feedback case) is again reduced to a minimization problem for the Blue player, with Blue optimal control given by

$$u_t^o = \underset{u \in U}{\operatorname{argmin}} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = x]$$

with $w_t^o = \bar{w}^1$. For the A_D case, we will call $\bar{w}^1$ as the complete-information state-feedback optimal control.

Analysis of the State-Feedback case

Through the analysis of the MAG example, it will be clear that the state-feedback problem is fairly simple. Further, the optimal control choices in the state-feedback case help illuminate the partially-observed problem, which is the focus of our study. We now discuss some simulation results to elucidate aspects of dynamics of the MAG example and some natural differences from the partially observed scenario.

Table 2.2: Blue State-Feedback Control, Fast speed for Blue, MS2

Blue State X_t^B	Blue player control u_t					
	1	2	3	4	5	6
1	1	1	0	1	0	0
2	1	1	1	1	1	1
3	0	1	1	0	1	0
4	1	1	0	1	1	1
5	0	1	1	1	1	1
6	0	1	0	1	1	1

Table 2.3: Blue State-Feedback Control, Slow speed for Blue, MS3

Blue State X_t^B	Blue player control u_t					
	1	2	3	4	5	6
1	1	0	0	1	0	0
2	1	1	1	1	1	1
3	0	0	1	0	1	0
4	1	1	0	1	1	1
5	0	1	1	1	1	1
6	0	1	0	1	1	1

In our modelling, we allow the Blue UCAVs to move from one route to the other in either two or three time steps. We will use the notation $MS2$ for the former and $MS3$ for the later. Note that in these two cases described above the state-dependent allowable Blue controls will be different. For $MS2$ and $MS3$ the state dependent Blue control are shown in Tables 2.2 and 2.3. The tables should be understood as indicating that a control is allowed in a certain state by having a 1 in that state/control entry. In other words, the entry 1 at the (i, j) position in the above tables implies that if Blue is at state i at any time, movement to state j is a feasible control option for Blue.

The strategy and the payoff also depend on the terminal time, T. We would refer to model with $T = k$ by Tk. Combining these notations together, $MS2T5$ would refer to the model, where the Blue UCAVs can move from one route to the other in 2 time steps and the terminal time is 5 units (or 4 control actions allowed per UCAV). The first results compare $MS2T5$ to $MS3T5$ for exactly the same parameters (note that this will compare mean-sample payoff as a function of Blue speed here). This is shown in Figure 2.3. Clearly, the faster Blue (being able to go from one route to the other in less time) achieves a smaller mean-sample payoff.

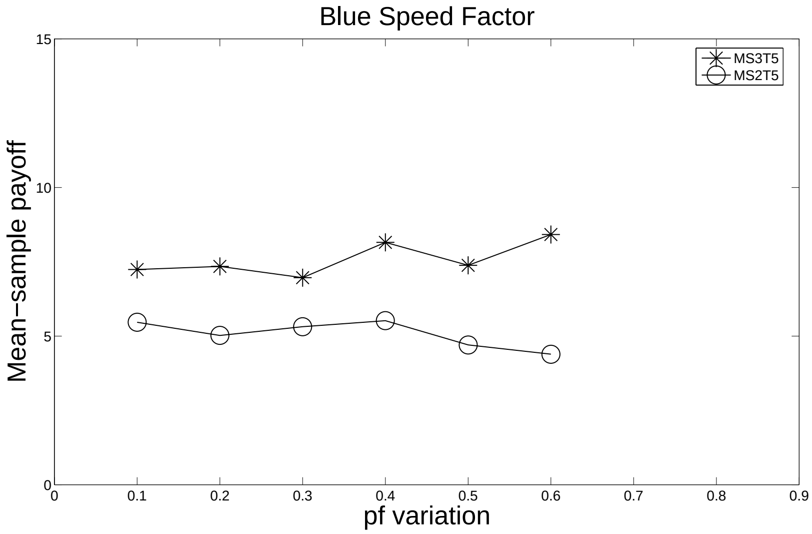


Figure 2.3: Payoff dependence on Blue speed, 'DR'

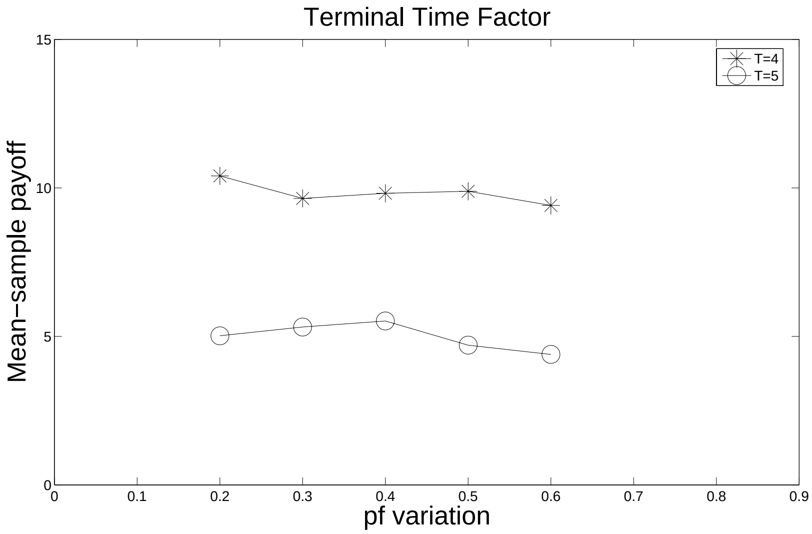


Figure 2.4: Payoff dependence on the terminal time, 'DR'

The second result gives the effect of having more control action for Blue. Obviously, a higher T (in $MS2T5$) gives Blue more control steps, and so on an average it is expected to destroy more Red entities, leading to a lower mean-sample payoff as shown in Figure 2.4.

In state-feedback games, with available control choices (sets U and W) known for both players, one can narrow down the set of feasible controls to a smaller set of controls that are sufficient to compute the optimal control sequences. We consider the case A_I from Blue's perspective. We choose $\alpha^1 \leq 0.5$, the attrition capability of one UCAV being less than half the attrition capability of two UCAVs. For nontrivial discussion, we choose the Red initial state distribution to be either $X_0^R = (1, 3)$ or $X_0^R = (2, 2)$. We discuss the case, $X_0^R = (1, 3)$. With the first Blue control with initial state $X_0^B = 2$, (or Blue UCAVs in the central/neutral zone) is to attack the eastern route with both UCAVs, $u_1^B = 3$. Further, the Blue control for subsequent steps depends on the outcome of the interaction at time 1 between the two UCAVs and the Red entities (3 in number) on the eastern route. Recall that the Blue control depends on the current state and the modelling constraints. For best results, the Blue player would like to have at least one shot at Red on both routes. With $u_0 = 3$, a maximum of 108 control sequences, of type $(u_0 = 3, u_1, u_2, u_3)$, are available for Blue in this set up. One can easily enumerate the the sequences $(3, u_1, u_2, u_3)$ such that the Blue player attacks the western route at least once and then reduce the available choices by by cross comparing sequences. For example, a reduced set of control sequences, of type $(u_0 = 3, u_1, u_2, u_3)$ is given below.

- (3,2,1,1), (3,2,2,1), (3,2,4,1), (3,2,5,6), (3,2,6,6).
- (3,5,2,1), (3,5,4,1), (3,5,4,6), (3,5,5,6), (3,5,6,6).
- (3,3,2,1), (3,3,5,6)

Some of these control sequences will achieve better results mainly due to removing redundant steps ($(3, 2, 1, 1)$ versus $(3, 2, 2, 1)$) and/or by noting the existence of more efficient attack sequences ($(3, 2, 6, 6)$ versus $(3, 2, 5, 6)$). Finally, only 5 sequences ($(3, 2, 1, 1)$, $(3, 3, 2, 1)$, $(3, 5, 4, 1)$, $(3, 5, 6, 6)$, and $(3, 3, 5, 6)$) need to be considered for evaluating the optimal strategy for Blue. For the parameter space in which we ran most of our simulation, $(3, 3, 2, 1)$ is the optimal Blue control.

The Blue control choice is thus fairly straight-forward for state-feedback game. Clearly with X_0^R unknown, such computational simplification is not available for the Blue player and finding optimal control sequence for the entire time horizon is not a simple task anymore. In adversarial environment, such similar reasoning is not sufficient or does not have the structure (due to lack of information) to yield a narrower set of control choices for computing the optimal control sequence. In fact, for the state-feedback case, for given attrition parameters, the optimal control sequences can also be found as a function of $X_0 = (a, b)$ and α_1. We conclude this part of the discussion by stating that unlike state-feedback, partially-observed games require a more complex algorithmic approach (like the deception-robust theory).

Another aspect of strategy or control formulation which is easy to analyze in the state-feedback game is the dependence of the Blue control strategy on α^1. For Red initial states (a, b), with $b > a$ and $b \neq 0$, $a \neq 0$ (as in the above example $(1, 3)$), the optimal Blue control u^o_{T-1} (with $X^B_{T-1} = 2$) changes from '3' to '6' for specific α^1 as a function of (a, b) and p_2^N. The switch between the two control options is determined by simply comparing the expected payoff between the Blue control choices $u_{T-1} = 3$ and $u_{T-1} = 6$. With $u_{T-1} = 3$, at least a Red entities survive and the expected payoff for the game is given by:

$$a + 20 + \sum_{k=0}^{b} \left[k\mathbf{E}[X_T^R = (a, k)] \right]$$

which can be written as

$$a + 20 + \sum_{k=0}^{b} \left[k \begin{pmatrix} b \\ k \end{pmatrix} (p_2^N)^{b-k} (1 - p_2^N)^k \right]$$

which is

$$a + 20 + \sum_{k=1}^{b} \left[k \begin{pmatrix} b \\ k \end{pmatrix} (p_2^N)^{b-k} (1 - p_2^N)^k \right]$$

Note that

$$k \begin{pmatrix} b \\ k \end{pmatrix} = b \begin{pmatrix} b-1 \\ k-1 \end{pmatrix}$$

Substituting the above identity, we get further simplification

$$a + 20 + \sum_{k=1}^{b} \left[b(1-p_2^N) \left(\begin{array}{c} b-1 \\ k-1 \end{array} \right) (p_2^N)^{b-k}(1-p_2^N)^{k-1} \right]$$

by re-indexing of the summation limits $k_1 = k-1$ we get,

$$a + 20 + b(1-p_2^N) \sum_{k_1=0}^{b-1} \left[\left(\begin{array}{c} b-1 \\ k_1 \end{array} \right) (p_2^N)^{b-1-k_1}(1-p_2^N)^{k_1} \right]$$

which using the binomial expansion finally gives

$$a + 20 + b(1-p_2^N)$$

Note that this result can also be achieved using the binomial mean for the expected Red teams surviving on the right route. Payoff achieved using $u_{T-1} = 6$ can be similarly computed, where we need to allow for all possibilities of k Red objects surviving at time T, i.e. $(1, k-1)$ or $(k-3, 3)$ dependent on the state X_{T-1}^R at time $T-1$.

$$\sum_{k=0}^{a} \left[(p_1^N)^{a+b-k}(1-p_1^N)^k (20+k) \left[\sum_{l_1}^{l_2} \left(\begin{array}{c} a \\ l \end{array} \right) \left(\begin{array}{c} b \\ k-l \end{array} \right) \right] \right]$$

where $l_1 = \max(0, k-b)$ and $l_2 = \min(k, a)$.

If $b >> a$, typically a higher α^1 is required for switching the control from 3 to 6, for a fixed p_2^N. Such change in strategy is not distinct in the partially-observed game because Blue control is a function of q_t which is dependent on the observation process. Also note that obviously for even $r_r = 1$, a higher α^1 is required for $(a+k, a+k)$ compared to (a, a). For the partially-observed game lets discuss the above example in the appropriate α^1 regime that admits sending one UCAV to each route as an optimal Blue strategy at time $T-1$. An observation, say $(0, 3)$, due to use of decoys by Red, may give a maximum likelihood state of $(0, r_4^2)$, leading to an optimal Blue control $u_t^o = 3$ (using the naive or the 'MLS' approach). However, since the true state is $(1, 3)$, this would be a sub-optimal control for Blue in the partially-observed game set-up. Simplified analytical expressions for computing the optimal Blue control are not available in the

Table 2.4: Comparing complete-information and partially-observed scenario, $(1, 3)$

p_2^N	0.6	0.65	0.7	0.75	0.8	0.85
State-Feedback	13.77	11.94	10.02	8.08	6.17	4.33
Partially-Observed	19.34	18.54	18.10	17.45	16.24	15.5

Table 2.5: Comparing complete-information and partially-observed scenario, $(2, 2)$

p_2^N	0.6	0.65	0.7	0.75	0.8	0.85
State-Feedback	15.75	14.14	12.38	10.48	8.48	6.40
Partially-Observed	18.42	17.44	16.46	15.24	12.42	11.12

partially-observed game owing to the unknown Red states and potentially adversarial noise.

We point another natural difference in the nature of state-feedback game to the partially-observed game scenario (using the MAG example) by comparing the mean-sample payoff when Red chooses a symmetric Red $(2, 2)$ initial state to the mean-sample payoff for an asymmetric Red initial state, $(1, 3)$ (in both full state-feedback and partial-information game). Obviously, given lack of true state information, Blue will achieve a higher mean-sample payoff compared to case where it has complete state information. This result are shown in Tables 2.4 and 2.5 by comparing the mean-sample payoff columns under full state-feedback and partially-observed cases in both tables.

Finally, note that results in these tables indicate that the symmetric distribution $(2, 2)$ is better for Red than $(1, 3)$ in the full state-feedback case. On the other hand, in the partially-observed game scenario, the asymmetric layout $(1, 3)$ (skewed with an added decoy on the non-stealthy eastern route) gives a higher mean-sample payoff and is a better initial layout for Red. These results are for the case A_I; note that when attrition is independent of stealth, the effect of the information level on the Blue control decisions is isolated to illustrate the importance of state information.

From Red perspective the case A_D is just minutely different. Red can still choose to make entities on each route stealthy or non-stealthy in this case. Typically attrition for stealthy entities will never be greater than that for non-stealthy entities. Consequently, the optimal Red control is to have the entities operate stealthily on the route under attack . Red, being the maximizer, is the inside player in the minimax, and makes its control decision after Blue makes its decision. Red can stay stealthy for the entire game as there is no running cost. The partially-observed game in the A_D case is again complex, as the Red player may use decoys to corrupt the observation process of Blue. Simply employing the state-feedback control for Red does not allow it the full potential to deceive Blue. We conclude with the above analysis that the partially-observed game (with adversarial noise) is a problem with a very complex structure.

2.4 Appendices

2.4.1 Proof of Lemma 2.2.1

Note that to prove $\lambda^*_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}$, we need to prove

(a). $\lambda^*_{[\bar{t},T)} \rightarrow U^{(T-\bar{t})}$.

(b). $\lambda^*_{[\bar{t},T)}$ is non-anticipative as defined in (2.5).

Since $\tilde{\lambda}_{[\bar{t},t)} \rightarrow U^{t-\bar{t}}$ and $\overline{\lambda}_{[t,T)} \rightarrow U^{T-t}$. Then given any $X_{[\bar{t},T]}$, by definition $\lambda^*_{[\bar{t},T)} : X_{\bar{t},T} \rightarrow (U^{t-\bar{t}} \bigcup U^{T-t}) \doteq U^{T-\bar{t}}$. Let $X^1_{\bar{t},T}$ and $X^2_{\bar{t},T}$ be two state processes such that $X^1_r = X^2_r$, $\forall\ r \leq s < T$. We divide the proof into two possibilities for proving non-anticipativeness. We first consider the case $s \leq t$. We assume that the Red control process is fixed by the non-anticipativeness of the θ process and the second arguments for λ is dropped in the following proof. We need to prove that $\lambda^*_s[X^1_{\bar{t},T}, .] = \lambda^*_s[X^2_{\bar{t},T}, .]$.

$$\lambda^*_s[X^1_{\bar{t},T}, .] = \lambda^*_s[X^1_{\bar{t},t} \bigcup X^1_{[t,T)}, .] = \lambda_s[X^1_{\bar{t},t}, .] = \lambda_s[X^2_{\bar{t},t}, .] = \lambda^*_s[X^2_{\bar{t},T}, .].$$

Where the second equality follows by the definition of $\lambda^*_{\bar{t},T}$ in (2.11), the third equality is a result of non-anticipativeness of $\lambda_{\bar{t},t}$ and the last equality is again by the definition of $\lambda^*_{\bar{t},T}$ in (2.11). For the second case $s \geq t$. We need to prove that $\lambda^*_s[X^1_{\bar{t},T}, .] = \lambda^*_s[X^2_{\bar{t},T}, .]$.

$$\lambda^*_s[X^1_{\bar{t},T}, .] = \lambda^*_s[X^1_{\bar{t},t} \bigcup X^1_{[t,T)}, .] = \overline{\lambda}_s[X^1_{[t,T)}, .] \tag{2.29}$$

with $X^1_{\bar{t},t} = X^2_{\bar{t},t}$. Similarly,

$$\lambda^*_s[X^2_{\bar{t},T}, .] = \lambda^*_s[X^2_{\bar{t},t} \bigcup X^2_{[t,T)}, .] = \overline{\lambda}_s[X^2_{[t,T)}, .] \tag{2.30}$$

with $X^2_{\bar{t},t} = X^1_{\bar{t},t}$. But $\overline{\lambda}_s[X^2_{[t,T)}, .] = \overline{\lambda}_s[X^1_{[t,T)}, .]$ by non-anticipativeness of $\overline{\lambda}_{[t,T)}$. Combining equations (2.29) and (2.30) completes the proof for the second case.

2.4.2 Proof of Corollary 2.2.1

The proof for $\tilde{\lambda} \in \Lambda_{[\bar{t},t)}$ is obvious by the definition and proof of Lemma 2.2.1. We only prove the n.a. of $\overline{\lambda}^{X_{[\bar{t},t)}} \in \Lambda_{[t,T)}$. Let $X^1_{[t,T)} \in \mathcal{X}^{T-t}$ and $X^2_{[t,T)} \in \mathcal{X}^{T-t}$, such that $X^1_r = X^2_r$, $\forall\ t \leq r \leq s$. Then by n.a. of $\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}$, one has

$$\lambda_s\left[X^{1,*}_{[\bar{t},T)}\right] = \lambda_s\left[X^{2,*}_{[\bar{t},T)}\right]$$

$\forall\ t \leq r \leq s$, where $X^{1,*} = \bar{X}_{[\bar{t},t)} \bigcup X^1_{[t,T)}$ and $X^{2,*} = \bar{X}_{[\bar{t},t)} \bigcup X^2_{[t,T)}$ for any $\bar{X}_{[\bar{t},t)} \in \mathcal{X}^{t-\bar{t}}$. Then by definition,

$$\overline{\lambda}_s^{\bar{X}_{[\bar{t},t)}} \left[X^1_{[t,T)}\right] \doteq \lambda_s \left[X^{1,*}_{[\bar{t},T)}\right]$$

and

$$\overline{\lambda}_s^{\bar{X}_{[\bar{t},t)}} \left[X^2_{[t,T)}\right] \doteq \lambda_s \left[X^{2,*}_{[\bar{t},T)}\right].$$

which completes the proof.

2.4.3 Proof of Lemma 2.2.2

We fix some $x \in \mathcal{X}$ and some $\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}$. We will hereon suppress the dependence on the first two arguments in (2.31), (2.32), and (2.34) for space constraints. Let

$$\lambda^o_{[\bar{t},T)} \in \underset{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}}{\operatorname{argmin}} F(x, \lambda_{[\bar{t},T)}). \tag{2.31}$$

So that the $X_\cdot$ process propagates using $\theta_{[\bar{t},T)}$ and $\lambda^o_{[\bar{t},T)}$ with initial condition $X_{\bar{t}} = x$. Also let

$$\tilde{\lambda}^o_{[\bar{t},t)} \in \underset{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}}{\operatorname{argmin}} \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} F(x, \tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \tag{2.32}$$

in which case the $X_\cdot$ process propagates using $\theta_{[\bar{t},T)}$ and $\tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}$ with initial condition $X_{\bar{t}} = x$. Finally for any given $\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}$ let

$$\overline{\lambda}^o_{[t,T)}(\tilde{\lambda}_{[\bar{t},t)}) \in \underset{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}}{\operatorname{argmin}} F(x, \tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \tag{2.33}$$

where now the $X_\cdot$ process propagates using $\theta_{[\bar{t},T)}$ and $\tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}^o_{[t,T)}(\tilde{\lambda}_{[\bar{t},t)})$ with initial condition $X_{\bar{t}} = x$. In particular

$$\overline{\lambda}^o_{[t,T)}(\tilde{\lambda}^o_{[\bar{t},t)}) \in \underset{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}}{\operatorname{argmin}} F(x, \tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \tag{2.34}$$

By (2.32) and (2.34), $\tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}^o_{[t,T)}(\tilde{\lambda}^o_{[\bar{t},t)}) \in \Lambda_{[\bar{t},T)}$. Then using the definition of $\lambda^o_{\bar{t},T}$ given in (2.31), one has

$$\min_{\lambda_{\bar{t},T} \in \Lambda_{\bar{t},T}} F(x, \lambda_{\bar{t},T}) \doteq F(x, \lambda^o_{\bar{t},T}) \leq F(x, \tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}^o_{[t,T)}(\tilde{\lambda}^o_{[\bar{t},t)})) \tag{2.35}$$

Using (2.35) and (2.34) one gets the first inequality

$$\min_{\lambda_{t,T} \in \Lambda_{t,T}} F(x, \lambda_{\bar{t},T}) \leq \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} F(x, \tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \tag{2.36}$$

For the reverse inequality, note again that using (2.32) and (2.34) we have

$$F(x, \tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}^o_{[t,T)}(\tilde{\lambda}^o_{[\bar{t},t)})) \doteq \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} F(x, \tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \tag{2.37}$$

By Corollary 2.2.1, $\lambda^o_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}$, which gives

$$F(x, \tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}^o_{[t,T)}(\tilde{\lambda}^o_{[\bar{t},t)})) \leq \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} F(x, \lambda^o_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)})$$

where on the right hand side, the $X_.$ process propagates using $\theta_{[\bar{t},T)}$ and $\lambda^o_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}$ with initial condition $X_{\bar{t}} = x$. Again using Corollary 2.2.1, $\lambda^o_{[t,T)} \in \Lambda_{[t,T)}$ (where the dependence of $\lambda^o_{[t,T)}$ on the state process $X_{[\bar{t},t)}$ is implicit in the definition of $\lambda^o_{[t,T)}$), which further gives

$$F(x, \tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}^o_{[t,T)}(\tilde{\lambda}^o_{[\bar{t},t)})) \leq F(x, \lambda^o_{[\bar{t},t)} \bigcup \lambda^o_{[t,T)})$$

note that $\lambda^o_{[\bar{t},T)} \doteq \lambda^o_{[\bar{t},t)} \bigcup \lambda^o_{[t,T)}$, so we have

$$F(x, \tilde{\lambda}^o_{[\bar{t},t)} \bigcup \overline{\lambda}^o_{[t,T)}(\tilde{\lambda}^o_{[\bar{t},t)})) \leq F(x, \lambda^o_{[\bar{t},t)} \bigcup \lambda^o_{[t,T)}) \doteq F(x, \lambda^o_{\bar{t},T})$$

Equations (2.31), (2.32), (2.34) with the above inequality yields

$$\min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} F(x, \tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \leq \min_{\lambda_{\bar{t},T} \in \Lambda_{\bar{t},T}} F(x, \lambda_{\bar{t},T}) \tag{2.38}$$

Finally equations (2.36) and (2.38) gives the required equality

$$\min_{\lambda_{\bar{t},T} \in \Lambda_{\bar{t},T}} F(x, \lambda_{\bar{t},T}) = \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} F(x, \tilde{\lambda}_{[\bar{t},t)} \bigcup \overline{\lambda}_{[t,T)}) \tag{2.39}$$

2.4.4 Proof of Lemma 2.2.3

One can rewrite S and R as follows:

$$S \doteq \min_{\lambda_{[t,T)}} \mathbf{E}_q\big\{\mathbf{E}[\mathcal{E}(\bar{X}_T)|\ \bar{X}_t = X_t]\big\} \tag{2.40}$$

$$R \doteq \mathbf{E}_q\big\{\min_{\lambda_{[t,T)}} \mathbf{E}[\mathcal{E}(\bar{X}_T)|\ \bar{X}_t = X_t]\big\} \tag{2.41}$$

where $X.$ propagates with the initial condition $X_{\bar{t}} = x$ using $\theta_{[\bar{t},t)}$ and $\tilde{\lambda}_{[\bar{t},t)}$ and $\bar{X}$ propagates with initial condition $\bar{X}_t = X_t$ and using $\theta_{[t,T)}$ and $\lambda_{[t,T)}$. The random variable $\bar{X}_t$ is distributed according to q, $\bar{X}_t \sim q$, and by $\mathbf{E}_q$ we mean the expectation with respect to the distribution q (or $\mathbf{E}_q \doteq \mathbf{E}_{\bar{X}_t \sim q}$). Let

$$\lambda^*_{[t,T)} \in \operatorname*{argmin}_{\lambda_{[t,T)}} \mathbf{E}_q\big\{\mathbf{E}[\mathcal{E}(\bar{X}_T)|\ \bar{X}_t = X_t]\big\}$$

and let $X^*_{[t,T)}$ be the process propagated by strategies $\lambda^*_{[t,T)}$ and $\theta_{[t,T)}$, then we get

$$S = \mathbf{E}_q\big\{\mathbf{E}[\mathcal{E}(X^*_T)|\ X^*_t = X_t]\big\}$$

and since $\lambda^*_{[t,T)} \in \Lambda_{[t,T)}$ we have the inequality

$$S \geq \mathbf{E}_q\big\{\min_{\lambda_{[t,T)}} \mathbf{E}[\mathcal{E}(\bar{X}_T)|\ \bar{X}_t = X_t]\big\} \doteq R$$

For the other direction, recall that $\lambda_{[t,T)}$ is dependent on the state process $X_{[t,T)}$ (non-anticipatively). Given any $z \in \mathcal{X}$, let $\overline{\lambda}^{*,z}$ be optimal, that is

$$\mathbf{E}\big[\mathcal{E}(\bar{X}^{z,*}_T)|\ \bar{X}^{z,*}_t = z\big] = \min_{\lambda_{[t,T)}} \mathcal{E}\big[(\bar{X}_T)|\ \bar{X}_t = z\big] \tag{2.42}$$

where $\bar{X}^{z,*}_{\cdot}$ is the state process corresponding to $\overline{\lambda}^{*,z}$ with initial condition $\bar{X}^{z,*}_t = z$. Now define $\lambda^*_{\cdot}$ as follows. For each sequence $\bar{X}_{[t,T]} \in \chi^{T-t+1}$ such that $\bar{X}_t = z$ let $\lambda^*_{[t,T)} = \lambda^{*,z}_{[t,T)}$. Note that this defines $\lambda^*_{[t,T)}$ uniquely for each process path. Given $\theta_{[t,T)}$, $\lambda^*_{[t,T)}$, and any initial $\bar{X}_t$, let $\bar{X}^*_{[t,T]}$ be the corresponding process. Then by (2.42) and definition of $\lambda^*_{\cdot}$, we get

$$\mathbf{E}_q\big\{\mathbf{E}\big[\mathcal{E}(\bar{X}^*_T)|\ \bar{X}^*_t = X_t\big]\big\} = \mathbf{E}_q\big\{\min_{\lambda_{[t,T)}\in\Lambda_{[t,T)}} \mathbf{E}\big[\mathcal{E}(\bar{X}_T)|\ \bar{X}_t = X_t\big]\big\} \doteq R \tag{2.43}$$

where $X_{\cdot}$ propagates (on both the sides) with the initial condition $X_{\bar{t}} = x$ using $\theta_{[\bar{t},t)}$ and $\tilde{\lambda}_{[\bar{t},t)}$. From time t onwards, on the left hand side $\bar{X}^*$ propagates with initial condition $\bar{X}^*_t = X_t$ and using $\theta_{[t,T)}$ and $\lambda^*_{[t,T)}$, whereas on the right hand side $\bar{X}$ propagates with initial condition $\bar{X}_t = X_t$ and using $\theta_{[t,T)}$ and $\lambda_{[t,T)}$. Since $\lambda^*_{[t,T)} \in \Lambda_{[t,T)}$, one immediately gets:

$$S \doteq \min_{\lambda_{[t,T)}} \mathbf{E}_q\big\{\mathbf{E}\big[\mathcal{E}(\bar{X}_T)|\ \bar{X}_t = X_t\big]\big\} \leq \mathbf{E}_q\big\{\mathbf{E}\big[\mathcal{E}(\bar{X}^*_T)|\ \bar{X}^*_t = X_t\big]\big\}$$

Then using (2.43), we get $S \leq R$, which completes the proof.

2.4.5 Proof of Lemma 2.2.4

The first equality is a restatement of (2.17), so we only need to prove the second equality in (2.19). From (2.20) one has

$$\tilde{N}_{[\bar{t},t)}(x, \tilde{\theta^o}_{[\bar{t},t)}) = \min_{\tilde{\lambda}_{[\bar{t},t)}\in\Lambda_{[\bar{t},t)}} \mathbf{E}\big[V_t(X_t)|\ X_{\bar{t}} = x\big] \tag{2.44}$$

where $X_{\cdot}$ propagates with initial condition $X_{\bar{t}} = x$ to X_t using $\tilde{\theta^o}_{[\bar{t},t)}$ and $\tilde{\lambda}_{[\bar{t},t)}$ using dynamics (1). By definition (as in (2.17))

$$V_t(x_1) = N_t(x_1, \overline{\theta}^{o,x_1}_{[t,T)}) \tag{2.45}$$

where

$$\overline{\theta}^{o,x_1}_{[t,T)} \in \operatorname*{argmax}_{\overline{\theta}_{[t,T)} \in \overline{\Theta}_{[t,T)}} N_t(x_1, \overline{\theta}_{[t,T)}) \tag{2.46}$$

For any process $\bar{X}_{[t,T]}$, let

$$\bar{\theta^*}_{[t,T)} = \overline{\theta}^{o,x_1}_{[t,T)} \text{ , if } \bar{X}_t = x_1. \tag{2.47}$$

Then by (2.47) and (2.46) and for any $z \in \mathcal{X}$

$$N_t(z, \bar{\theta^*}_{[t,T)}) = N_t(z, \overline{\theta}^{o,z}_{[t,T)}) \tag{2.48}$$

Using (2.16), and (2.48), we get

$$N_t(z, \bar{\theta^*}_{[t,T)}) = \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} \mathbf{E}\big[\mathcal{E}(\bar{X}_T) \,|\, \bar{X}_t = z\big] \tag{2.49}$$

where $\bar{X}_{[t,T]}$ propagates by (1) with controls $\overline{\lambda}_{[t,T)}$ and $\bar{\theta^*}_{[t,T)}$, and $\bar{X}_t = z$. Then using (2.45), and (2.48), and for any $z \in \mathcal{X}$ we get

$$V_t(z) = N_t(z, \bar{\theta^*}_{[t,T)}) \tag{2.50}$$

For any ω in the sample space, $\bar{X}_t(\omega) \in \mathcal{X}$ with probability 1. Note that since (2.50) is true $\forall z \in \mathcal{X}$, this gives

$$V_t(\bar{X}_t) = N_t(\bar{X}_t, \bar{\theta^*}_{[t,T)}) \tag{2.51}$$

Then substituting (2.51) in (2.44), we get

$$\tilde{N}_{[\bar{t},t)}(x, \tilde{\theta}^o_{[\bar{t},t)}) = \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \mathbf{E}\left[N_t(X_t, \bar{\theta}^*_{[t,T)}) \mid X_{\bar{t}} = x\right]$$

Further using (2.16) in the right side of the above equation, gives

$$\tilde{N}_{[\bar{t},t)}(x, \tilde{\theta}^o_{[\bar{t},t)}) = \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \mathbf{E}\big[\min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} \mathbf{E}\big\{\mathcal{E}(\bar{X}_T) \,|\, \bar{X}_t = X_t\big\} | X_{\bar{t}} = x\big]$$

where $X_{\cdot}$ propagates using initial condition $X_{\bar{t}} = x$ using $\tilde{\theta}^o_{[\bar{t},t)}$ and $\lambda_{[\bar{t},t)}$ and $\bar{X}_{\cdot}$ propagates with initial condition $\bar{X}_t = X_t$ using $\theta^*_{[t,T)}$ and $\lambda_{[t,T)}$. Then using Lemma 2.2.1, Lemma 2.2.2, and definition of Λ gives

$$\tilde{N}_{[\bar{t},t)}(x, \tilde{\theta}^o_{[\bar{t},t)}) = \min_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}\big[\mathbf{E}\big\{\mathcal{E}(\bar{X}_T) \,|\, \bar{X}_t = X_t\big\} \; | X_{\bar{t}} = x\big]$$

Further by conditional expectation (Resnick 1998) this yields

$$\tilde{N}_{[\bar{t},t)}(x,\tilde{\theta}^o_{[\bar{t},t)}) = \min_{\overline{\lambda}_{[\bar{t},T)}} \mathbf{E}\big[\mathcal{E}(X_T)\,|X_{\bar{t}} = x\big] \tag{2.52}$$

where $X_{[\bar{t},T]}$ propagates according to dynamics (2.1) with controls $\tilde{\theta}^o_{[\bar{t},t)} \cup \bar{\theta}^*{}_{[t,T)}$ and $\overline{\lambda}_{[\bar{t},T)}$. Note that $\tilde{\theta}^o_{[\bar{t},t)} \cup \bar{\theta}^*{}_{[t,T)} \in \Theta_{[\bar{t},T)}$, then using (2.52) implies

$$\tilde{N}_{[\bar{t},t)}(x,\tilde{\theta}^o_{[\bar{t},t)}) \leq \max_{\theta_{[\bar{t},T)} \in \Theta_{[\bar{t},T)}} \; \min_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}\big[\mathcal{E}(X_T) \mid X_{\bar{t}} = x\big].$$

The right hand side, by definition, is $V_{\bar{t}}(x)$, so we get the inequality

$$\tilde{N}_{[\bar{t},t)}(x,\tilde{\theta}^o_{[\bar{t},t]}) \leq V_{\bar{t}}(x) \tag{2.53}$$

For the other direction let us fix $x \in \mathcal{X}$. Then define

$$\tilde{\theta}^o_{[\bar{t},t)} \in \underset{\tilde{\theta}_{[\bar{t},t)} \in \Theta_{[\bar{t},t)}}{\operatorname{argmax}} \; \max_{\overline{\theta}_{[t,T)} \in \Theta_{[t,T)}} \; \min_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}\big[\mathcal{E}(X_T)\,|X_{\bar{t}} = x\big] \tag{2.54}$$

Then $X_.$ propagates till time t using $\tilde{\theta}^o_{[\bar{t},t)}$ and $\lambda_{[\bar{t},t)}$ with initial condition $X_{\bar{t}} = x$ (the dependence of the optimal θ^o on x is implicit here). From time t onwards, $X_.$ propagates using $\overline{\theta}_{[t,T)}$ and $\lambda_{[t,T)}$. Now for a fixed $x \in \mathcal{X}$ and $\tilde{\theta}^o_{[\bar{t},t)}$ given by (2.54), define

$$\overline{\theta}^o_{[t,T)} \in \underset{\theta_{[t,T)} \in \Theta_{[t,T)}}{\operatorname{argmax}} \; \min_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}\big[\mathcal{E}(X_T)\,|X_{\bar{t}} = x\big] \tag{2.55}$$

where now $X_.$ propagates till time t using $\tilde{\theta}_{[\bar{t},t)}$ and $\lambda_{[\bar{t},t)}$ from initial condition $X_{\bar{t}} = x$ and from time t onwards, $X_.$ propagates using $\overline{\theta}^o_{[t,T)}$ and $\lambda_{[t,T)}$ (the dependence of the optimal $\overline{\theta}^o$ on x and $\tilde{\theta}^o_{[\bar{t},t)}$ is implicit here). Then using definition of $V_{\bar{t}}(x)$ and (2.54) and (2.55)

$$V_{\bar{t}}(x) = \min_{\lambda_{[\bar{t},T)} \in \Lambda_{[\bar{t},T)}} \mathbf{E}\big[\mathcal{E}(X^*_T)\,|X^*_{\bar{t}} = x\big] \tag{2.56}$$

where $X^*_.$ propagates till time t using $\tilde{\theta}^o_{[\bar{t},t)}$ and $\lambda_{[\bar{t},t)}$ from initial condition $X^*_{\bar{t}} = x$ and from time t onwards, $X^*_.$ propagates using $\overline{\theta}^o_{[t,T)}$ and $\lambda_{[t,T)}$. Using conditional expectation (Resnick 1998) and Lemma 2.2.1, (2.56) becomes

$$V_{\bar{t}}(x) = \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \mathbf{E}\big[\min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} \mathbf{E}\big\{\mathcal{E}(\bar{X}^*_T) \mid \bar{X}^*_t = X^*_t\big\} \mid X^*_{\bar{t}} = x\big] \tag{2.57}$$

with appropriate propagation of X^*. and $\bar{X}^*_.$ in the appropriate time domain as in (2.56). Note that $\overline{\theta}^o_{[t,T)} \in \Theta_{[t,T)}$ (where this $\overline{\theta}^o$ is the specific one dependent on x and $\tilde{\theta}^o_{[\bar{t},t)}$) which gives

$$V_{\bar{t}}(x) \leq \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{\bar{t}},t)}} \mathbf{E}\Big[\max_{\overline{\theta}_{[t,T)} \in \Theta_{[t,T)}} \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} \mathbf{E}\big\{ \mathcal{E}(\bar{X}_T) \,|\, \bar{X}_t = X^*_t \big\} |\; X^*_{\bar{t}} = x \Big] \tag{2.58}$$

where $X^*_\cdot$ propagates till time t using $\tilde{\theta}^o_{[\bar{t},t)}$ and $\lambda_{[\bar{t},t)}$ from initial condition $X^*_{\bar{t}} = x$ and from time t onwards, $\bar{X}_\cdot$ propagates using $\overline{\theta}_{[t,T)}$ and $\lambda_{[t,T)}$ with initial condition $\bar{X}_t = X^*_t$. Further, using $\tilde{\theta}^o_{[\bar{t},t)} \in \Theta_{[\bar{t},t)}$ gives

$$V_{\bar{t}}(x) \leq \max_{\tilde{\theta}_{[\bar{t},t)} \in \Theta_{[\bar{t},t)}} \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \mathbf{E}\Big[\max_{\overline{\theta}_{[t,T)} \in \Theta_{[t,T)}} \min_{\overline{\lambda}_{[t,T)} \in \Lambda_{[t,T)}} \mathbf{E}\big\{ \mathcal{E}(\bar{X}_T) \,|\, \bar{X}_t = X_t \big\} |\; X_{\bar{t}} = x \Big] \tag{2.59}$$

where $X_\cdot$ propagates till time t using $\tilde{\theta}_{[\bar{t},t)}$ and $\lambda_{[\bar{t},t)}$ from initial condition $X_{\bar{t}} = x$ and from time t onwards, $\bar{X}_\cdot$ propagates using $\overline{\theta}_{[t,T)}$ and $\lambda_{[t,T)}$ with initial condition $\bar{X}_t = X_t$. By definition of V_t, (2.59) yields

$$V_{\bar{t}}(x) \leq \max_{\tilde{\theta}_{[\bar{\bar{t}},t)} \in \Theta_{[\bar{t},t)}} \min_{\tilde{\lambda}_{[\bar{t},t)} \in \Lambda_{[\bar{t},t)}} \mathbf{E}[V(X_t) | X_{\bar{t}} = x] \tag{2.60}$$

By (2.22), the right hand side of (2.60) is $\tilde{N}_{\bar{t},t}(x, \tilde{\theta}^o_{[\bar{t},t)})$, which completes the reverse direction.

2.4.6 Proof of Theorem 2.2.1

Using Lemma 2.2.4 and choosing $\bar{t} \doteq s$ and $t \doteq s+1$

$$V_s(x) = \max_{\theta_s \in \Theta_s} \min_{\lambda_s \in \Lambda_s} \mathbf{E}[V_{s+1}(X_{s+1} |\; X_s = x)]$$

We note that for a given $X_s = x$, maximum over θ_s can be replaced by maximum over W (as $\theta_s : x \rightarrow W$) and similarly minimum over λ_s can be replaced by minimum over U, which gives the one-step DPE (replacing the dummy time variable s with t above):

$$V_t(x) = \max_{w \in W} \min_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1}) |\; X_t = x] \tag{2.61}$$

which completes the proof of Theorem 2.2.1.

This chapter is in part a reprint of the materials as is appears in,

Rajdeep Singh, William M. McEneaney - *Robustness to Deception*, Chapter 2.4 in the book “Adversarial Reasoning: Computational Approaches to Reading the Opponent’s Mind”, CRC Press, To appear.

Rajdeep Singh, William M. McEneaney - *Unmanned vehicle decision making under imperfect information in an adversarial environment*, AIAA Journal of Guidance Navigation and Control, in preparation.

The dissertation author was the primary author and the co-author listed in these publications directed and supervised the research.

Chapter 3

Blue Approach in the Partially-Observed Game

The partially-observed game where the Blue player has only some or no information about the state of the system will be now discussed. The Red player has complete state knowledge. Note the dynamics of the state transition is still modelled as a discrete-time controlled markov chain process (affected by the controls of both players). The information set for the Blue player, q, in this game is built on the observations that the Blue player will make, which noticeably, can also be controlled by the Red player. Blue decision process will based on the belief that the true state at time t, X_t, is distributed as per q_t. The Blue player also needs to estimate or (guess based on some intel) the initial state of the system, q_0. The non-random control action due to presence of an adversary (Red) which can affect the Blue player's control (by affecting Blue's observations) makes this problem interesting and challenging. We refer to deception as the use of controls by the Red player that may allow the Red player to steer the state which is favored by Red, or in other words, which leads to maximizing the terminal payoff. Note that such deception controls purposely introduce non-random noise into Blue observation process due to which the Blue may chose a sub-optimal control action or strategy (dependent on the estimates driven by observations). Since, the Blue control action is based on state estimates, the utility of automated Red controllers which employ deception when useful, is an equally important and interesting problem. Some Red controls (more than others) may be able to corrupt Blue observations and possibly achieve a higher payoff. We first focus on the Blue player's need to implement an approach which allows for some robustness to the potential deception that the Red player may employ. We look at some of the standard approaches for the Blue player followed by a recently proposed theory (with some revised results) that formulates the game problem using the concept of information state and gives a deception-robust Blue control.

3.1 Partially-Observed Game Formulation

Recall that the Red player will know the state perfectly. We will assume that there is an observation process for Blue which can be controlled by both players. Let the observation process be $y.$, with $y_t \in Y$. Given $X_t = i$, and controls $u_t = u$, $w_t = \vec{w}_i$

$$\tilde{R}_i^{\bar{y}}(u, \vec{w}) = \Pr(y_t = \bar{y} \mid X_t = i) \tag{3.1}$$

denotes the probability that observation $y_t = \bar{y}$, where $\vec{w} \in W^n$ is the Red state-feedback control. We take Y to be a finite set for consistency (but that does not appear to be required for the results to follow). We will assume the observation process does not depend on the Blue control. Then using (3.1), given $X_t = i$ and $w_t = \vec{w}_i$, the probability that $y_t = \bar{y}$ becomes

$$\tilde{R}_i^{\bar{y}}(\vec{w}) = \Pr(\mathbf{y}_t = \bar{y} \; | X_t = i). \tag{3.2}$$

With the observation process included, we now discuss the propagation of the distribution q_t. We first discuss the case where Blue models the (unseen) Red control actions as a stochastic process and propagates forward a conditional probability representing its lack of knowledge of the state of the system.

3.2 Blue Approaches Using a Stochastic Modelling of Red Control

Blue can propagate a single distribution (initialized to $q_0 \in Q(\mathcal{X})$) conditioned on the above observation process. The posteriori $\hat{q}$ is then obtained using the Bayesian update and (3.2),

$$\left[\hat{q}^{\vec{w}}_{t+1}\right]_i = \frac{\tilde{R}^{\bar{y}}_i(\vec{w})\,[q_t]_i}{\sum_{k\in\mathcal{X}} \tilde{R}^{\bar{y}}_k(\vec{w})\,[q_t]_k} \tag{3.3}$$

However Blue doesn't know the current Red control so one get's the average posteriori by

$$\widehat{q}_{t+1} = \sum_{\vec{w}\in\vec{W}^n} \hat{q}^{\vec{w}}_{t+1} p^B_{\vec{w}}. \tag{3.4}$$

where $p^B_{\vec{w}}$ gives the probability with which the Blue player assumes the Red control to be $\vec{w}$; Blue models the Red control to be distributed according to $p^B_{\vec{w}}$. Obviously, Red may not be using such simplistic control approach. The dynamical update given the current $u_t = u$ is then given by

$$q_{t+1} = \sum_{\vec{w}\in W^n} \left[\widetilde{P}^T(u,\vec{w})\hat{q}^{\vec{w}}_t\right] p^B_{\vec{w}} \tag{3.5}$$

where we note that the $\hat{q}$ is as given by (3.3), since it is inside the summation and we want to use the same $\vec{w}$ in the observation and the dynamic update before taking the expectation using $p^B_{\vec{w}}$. Also note that in the arguments of $\tilde{P}$, we now have the open loop Blue player control as the Blue player does not have the state process information, $X_{\cdot}$. Hence $\tilde{P}_{ij}(u,\vec{w}) = P_{ij}(u,\vec{w}_i)$, where $\vec{w}_i$ is the ith component of the Red state feedback control $\vec{w}$ applied to the state i and P_{ij} definition is as given by (2.1).

In this chapter we study the Blue player's control decision viewpoint by fixing some hand-crafted strategy for the Red player. The use of decoys is an obvious control action for Red in the partially-observed scenario. In this hand-crafted strategy, Red moves the route with less entities in a stealthy mode and moves the bigger group of entities on the other route in a non-stealthy mode with an added decoy (to further exaggerate the asymmetry in Blue's estimation of the true state $X_{\cdot}$). We will refer to this strategy as the 'RG' or the Red-game strategy. Note that we allow the Red player to have complete access to the appended state (X_t, q_t) for making its control decision (considering that as a worst-case scenario for Blue). In the section on deception-enabled control for the Red player, we will see that this hand-crafted strategy is optimal for Red to use deception for its advantage, whenever possible. The order of action/operation for the game at any time t is as follows.

- Observation process: The random variable $\mathbf{y}_t$ is distributed by (3.2), where $\mathbf{y}_t : \Omega \rightarrow \mathbf{Y}$. For example, given $\mathbf{X}_t(\omega) = i$ and $w_t = \vec{w}_i$, $\mathbf{y}_t(\omega) = \bar{y}$, for some $\bar{y} \in Y$.
- Blue control decision: Blue can propagate a single distribution process $q_\cdot$ (using a stochastic model of Red control, $p^B_{\vec{w}}$). Then, Blue can either apply the state-feedback optimal control at some estimate of the state or use a 'HB' recursion based control approach. Blue can propagate $q_\cdot$ process from q_0 using the observation update (3.4) and dynamic update (3.5) and apply one of the following approaches (to be discussed in section 3.2):
  - The 'MLS' or Naïve approach.
  - Risk-sensitive approach ('RS').
  - Heuristic Blue ('HB').
- Finally one uses the control u (obtained from one of the previous listed items) in the state transition (2.1) and and the dynamic update (3.5).

Recall that Blue control at time t, u_t, is based on q_t (with Blue assuming that the true state X_t, is distributed as per q_t). We first discuss the controls based on a single distribution process $q_\cdot$ carried forward by the Blue player (starting at some $q_0 \in Q(\mathcal{X})$) using the updates (3.4) and (3.5). We will use the following notation throughout:

$$\Gamma\left[a\right] = \underset{i \in \mathcal{X}}{\operatorname{argmax}}(a_i) \tag{3.6}$$

and

$$\gamma\left[a\right] = \underset{i \in \mathcal{X}}{\operatorname{argmin}}(a_i) \tag{3.7}$$

In linear control systems with quadratic cost criteria, the control obtained through the *separation principle* is optimal. That is, the optimal control is obtained from the state-feedback control applied at the state given by

$$\overline{x} = \Gamma\left[q_t\right]. \tag{3.8}$$

where Γ is defined by (3.6). Note that the argmax computation could lead to a set of multiple states achieving the maximum and the equality in the above equation implies that we choose one of those states. This would be the first and the most 'naïve' approach where using equations (3.8), (2.25), and (2.24), allows us to compute optimal controller

$$\tilde{u}_t^*(\tilde{w}_t^*) \in \operatorname*{argmin}_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1}) | X_t = \overline{x}]. \tag{3.9}$$

where

$$w_t^* \in \operatorname*{argmax}_{w \in W} \min_{u \in U} \mathbf{E}[V_{t+1}(X_{t+1}) | X_t = \overline{x}].$$

We will refer to this as the 'MLS' approach and the control given by (3.9) as the 'MLS' Blue control or the u^{MLS} control. Note that using the above approach with the min-max definition of value may lead to different optimal controls for Red or (and) Blue (if no saddle point exists). Of course, the unseen Red controls will very likely not be randomly chosen. In order to safeguard itself against those possibilities which are most dangerous, Blue needs to somehow emphasize those possible states when deciding what action to take. In this section, we consider one approach to that method, which we will refer to as the risk-sensitive controller (for Blue). This method will combine the likelihood of each possible state with the dangerousness of that state in order to obtain a state estimate. This estimate will be averse to risk in the sense that it will tend toward those states which are likely to lead to undesirable outcomes from the Blue perspective. This approach will employ a heuristic that is based on an equivalence between risk-sensitive stochastic control and stochastic games. Proving such an equivalence is technically challenging. This equivalence has been obtained for some problem classes, but is not proven for our problem class. Nonetheless, we will apply the resulting theory (assuming equivalence) to our problem.

A different principle, the *Certainty Equivalence Principle,* is appropriate in robust control. We have applied a generalization of the controller that would emanate from this latter principle. This generalization allows us to tune the relative importance between the likelihood of possible states and the risk of misestimation of the state. Let us motivate the proposed approach in a little more detail. In deterministic games under partial information, the Certainty Equivalence Principle indicates that one should use the state-feedback optimal control corresponding to the state given by

$$\overline{\overline{x}} = \Gamma \left[\mathbb{I}_t + V_t \right]$$

where $\mathbb{I}$ is the information state and V is the value function (Fleming & Soner 1992) (assuming uniqueness of the argmax of course). In this problem class, the information state is essentially the worst case cost-so-far, and the value is the minimax cost-to-come. So, heuristically, this is roughly equivalent to taking the worst-case possibility for total cost from initial time to terminal time. (See, for instance, James et al. (James & Baras 1996, Helton & James 1999, James & Yuliar 1995), and McEneaney (McEneaney 1999*a*, McEneaney 1999*b*).) We now discuss the mathematics which lead to the heuristic for the algorithm described below.

The deterministic information state is very similar to the *log* of the observation-conditioned probability density in stochastic formulations for terminal/exit cost problems. (In fact, this is exactly true for a class of linear/quadratic problems.) In the stochastic linear/quadratic problem formulation, the information state at any time, t, is characterized as a Gaussian distribution, say

$$p_t(x) = k(t)\exp\Big\{-\tfrac{1}{2}(x-\bar{x}(t))^T C^{-1}(t)(x-\bar{x}(t))\Big\}.$$

In the deterministic game formulation, the information state at any time, t, is characterized as a quadratic cost, say

$$\mathcal{I}_t(x) = -\tfrac{1}{2}(x-\hat{x}(t))^T Q(t)(x-\hat{x}(t)) + r(t).$$

Interestingly, Q and C^{-1} satisfy the same Riccati equation (or, equivalently, Q^{-1} and C satisfy the same Riccati equation). $\overline{x}$ and $\overline{\overline{x}}$ satisfy identical equations as well. Therefore, $\mathcal{I}_t(x) = \log[p_t(x)] +$ "time-dependent constant" (McEneaney 1999*b*, Fleming 1997).

This motivates the algorithm proposed. This algorithm is the following: apply state-feedback control at

$$\bar{x}^* = \Gamma\left[\log \widehat{q}_t + \kappa V_t\right] \tag{3.10}$$

where $\widehat{q}$ is the probability distribution based on the conditional distribution for Blue given by (3.4) and V is state-feedback stochastic game value function (c.f. (Fleming & Soner 1992)). Here, $\kappa \in [0,\infty)$ is a measure of risk aversion. Note that $\kappa = 0$ implies that one is employing a 'MLS' estimate in the state- feedback control (for the game), i.e.

$$\Gamma\left[\log \widehat{q}_t\right] = \Gamma\left[\widehat{q}_t\right].$$

Note also (at least in linear-quadratic case where $\log[\widehat{q}_t]_i = \mathbb{I}_t(i)$ modulo a constant), $\kappa = 1$ corresponds to the deterministic game Certainty Equivalence Principle (Helton & James 1999, James & Baras 1996), i.e. $\text{argmax}\{\mathbb{I}_t(i) + V_t(i)\}$. As $\kappa \to \infty$, this converges to an approach which always assumes the worst possible state for the system when choosing a control – regardless of observations. In equation (3.10), if the evaluation of the two components $\log[\widehat{q}_t]$ and V are approximately in some overlapping (or similar) range , then a reasonable estimate for the risk-sensitive approach is ensured otherwise the argmax computation could be driven by either one. One can scale one of the components in the following way to get same range of numbers for evaluation of both the components. Let $q^M = \max_{i\in\mathcal{X}}\{\log[\widehat{q}_t]_i\}$ and $q^m = \min_{i\in\mathcal{X}}\{\log[\widehat{q}_t]_i\}$. Similarly define $V^M = \max_{i\in\mathcal{X}} V_i$ and $V^m = \min_{i\in\mathcal{X}} V_i$. Now one can use an alternate (modified) form of (3.10) computation as

$$\bar{x}^* = \Gamma\left[\log\widehat{q}_t + K\kappa V_t\right] = \Gamma\left[\frac{\log\widehat{q}_t}{K} + \kappa V_t\right] \tag{3.11}$$

where $K = \frac{q^M - q_m}{V^M - V^m}$. The control is then computed using equations (3.11), (2.25), and (2.24) as

$$\tilde{u}_t^*(\tilde{w}_t^*) \in \underset{u\in U}{\text{argmin}}\, \mathbf{E}[V_{t+1}(X_{t+1})|X_t = \bar{x}^*]. \tag{3.12}$$

where

$$w_t^* \in \underset{w\in W}{\text{argmax}}\, \min_{u\in U} \mathbf{E}[V_{t+1}(X_{t+1})|X_t = \bar{x}^*].$$

The control given by (3.12) will be the 'RS' Blue control, u^{RS}.

Another heuristic based on using the state-feedback value function is now outlined. Given $q_t = q$, and controls $u_t = u$ and $w_t = \vec{w}$, let us define

$$J(q,u,\vec{w}) = \sum_{i\in\mathcal{X}}\sum_{j\in\mathcal{X}} \tilde{P}_{ij}(u,\vec{w})V_{t+1}(j)\,[\hat{q}_t]_i \tag{3.13}$$

where $\hat{q}$ is given by (3.3). Then, taking expectation with respect to the Red state-feedback controls being distributed as per $p^B_{\vec{w}}$, one gets

$$\bar{J}(q,u) = \sum_{\vec{w}\in W^n} J(q,u,\vec{w})p^B_{\vec{w}}$$

Then let's define

$$u_t^{*,q} \in \underset{u\in U}{\text{argmin}}\, \bar{J}(q,u) \tag{3.14}$$

to be the best Blue control based on this computation. This heuristic controller given by (3.14) will be referred to as the 'HB' Blue control, u^{DP}.

Now we turn to the case where Blue can propagate forward multiple distributions, and does not use a stochastic Red control. Instead Blue propagates each distribution forward based on all possible Red control, ($\vec{w} \in W^n$). In this manner Blue defines an information state and propagates it forward in time. Blue control is shown to be based on a payoff that takes into account the information state and the best cost Blue can achieve in the future given that information state. This Blue approach is now discussed in the following section.

3.3 Deception-Robust Theory

In a deterministic game under imperfect information, the information state for Blue is a function of the state, and it represents the minimal cost to the opposing player (maximal cost from the point of view of Blue) for the state to be x at current time t given the observations up to the current time. Alternatively, in a stochastic control problem under imperfect information, the information state is simply the probability that $X_t = x$ conditioned on the observations up to the current time t. Here however, Red can affect the observation process, so one must consider the cost to Red to produce a possibly misleading conditional probability distribution. Thus, it is natural to propose an information state for Blue as $I_t : Q(\mathcal{X}) \to R$ where $Q(\mathcal{X})$ is the space of probability distributions over state space $\mathcal{X}$; $Q(\mathcal{X})$ is the simplex in the first octant of R^n defined by the unit basis vectors. For example with $\#(\mathcal{X}) = 3$, one has a simplex in R^3 as shown in Figure 3.1. For simplicity of presentation, we henceforth refer to I_t as an information state, although the basis for this designation does not appear until Section 3.3.3. We let the initial information state be $I_0(.) = \phi(.)$. Here, ϕ represents the initial cost to obtain and/or obfuscate initial state information. The case where this information cannot be affected by the players may be represented by a maxplus delta function (a form which, as we will show soon, allow for tractability). ϕ_k is a max-plus delta function over $Q(\mathcal{X})$

if there exists $q_k \in Q(\mathcal{X})$ such that

$$\phi_k(q) = \begin{cases} 0 & \text{if } q = q_k \\ -\infty & \text{otherwise.} \end{cases} \tag{3.15}$$

The problem will still be finite-dimensional for initial information states taking the form of finite max-plus sums of max-plus delta functions. ϕ is a (finite) max-plus sum of max-plus delta functions if there exist $\{q_k\}_{k=1}^{K}$ such that

$$\phi(q) = \bigoplus_{k=1}^{K} \phi_k(q) = \max_k \phi_k(q). \tag{3.16}$$

This will be the case we will concentrate on here. Let the set of max plus delta function used in the definition of ϕ be denoted by $\tilde{Q}_0^{\phi} \subset Q(\mathcal{X})$,

$$\tilde{Q}_0^{\phi} = \{q \in Q(\mathcal{X}) : \phi(q) = 0\}. \tag{3.17}$$

Clearly note that $\phi(q) = 0 \iff \phi_k(q) = 0$ for at least some k. Clearly for a finite K and projecting all future paths using every possible $\vec{w} \in W^n$, till time t, the maximum number of state trajectories will be $K[\#W^n]^t$. Thus, with the max-plus sum of delta functions, the information state propagation leads to only finitely many distributions. We first discuss the information state propagation for any initial form of $\mathcal{I}_0$.

3.3.1 Information State Propagation

Since w_t is not known by Blue, it will be necessary to keep track of a set of feasible conditional probabilities at time t, Q_t. Note that for t prior to the current time, u_t being Blue's control is known by Blue. We do not allow either player to know the control history of their opponent. Let $w_{[0,t)} = \{w_0, w_1, ., ., w_{t-1}\}$, where each $w_r \in W$ denotes a sequence of controls for Red. Then, if the controls for Red were independent of the true state, x, one would have

$$Q_t(u_{[0,t)}) = \big\{q \in Q(\mathcal{X}) : \exists w_{[0,t)} \in W^t \text{ such that } q_0 \in Q(\mathcal{X}) \text{ where } \\ q_0 \text{ is given by backward propagation (3.19) with } q_t = q\,\big\} \tag{3.18}$$

where

$$q_{r-1} = \widetilde{P}^{-T}(u_{r-1}, w_{r-1})q_r. \tag{3.19}$$

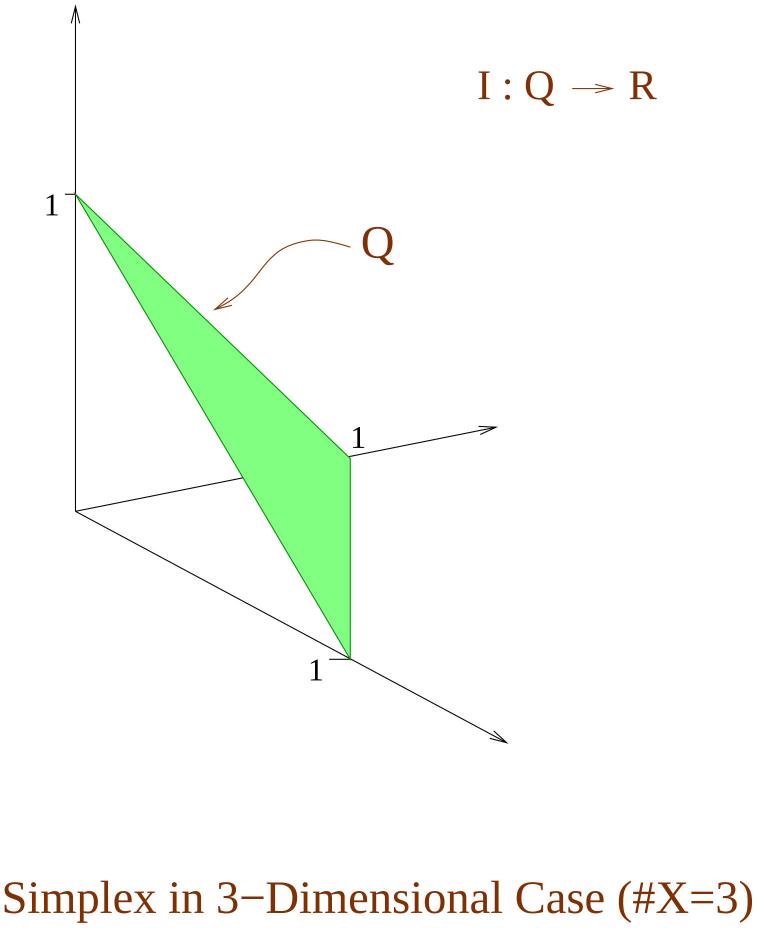


Figure 3.1: Simplex in a 3D

We will assume the existence of P^{-T} in the standard sense throughout. Now let $\vec{w}_{[0,t)} = \{\vec{w}_0, \vec{w}_1, \ldots, \vec{w}_{t-1}\}$ where each $\vec{w}_r \in W^n$ denotes a vector of state-dependent controls for Red. One now sees that (in the absence of an observation process) the feasible set at time t should be given by

$$Q_t(u_{[0,t)}) = \big\{q \in Q(\mathcal{X}) : \exists \vec{w}_{[0,t)} \in [W^n]^t \text{ such that } q_0 \in Q(\mathcal{X}) \text{ where}$$
$$q_0 \text{ is given by backward propagation (3.21) with } q_t = q\,\big\} \tag{3.20}$$

where

$$q_{r-1} = \widetilde{P}^{-T}(u_{r-1}, \vec{w}_{r-1}) q_r. \tag{3.21}$$

Recall that the observations occur at each time step just before the dynamics, and we continue to denote the a priori by q_t and the a posteriori by $\widehat{q}_t$. Also, using $\tilde{R}(\bar{y}, u_t, \vec{w}) \doteq R^{\bar{y}}(u, \vec{w})$, where $R^{\bar{y}}(u, \vec{w})$ is defined in (3.1), we define

$$\widehat{q}_t = \left(\tfrac{1}{\widetilde{R}'(\overline{y}, u_t, \vec{w}) q_t} \right) D(\overline{y}, u_t, \vec{w}) q_t. \tag{3.22}$$

The possible set of posteriori distributions, $\widehat{\mathcal{Q}}_t$ is the set of all $\widehat{q}_t$ given by (3.22) for some $q_t \in \mathcal{Q}_t$. We suppose that D is full rank; i.e. that $\widetilde{R}_i \neq 0$ for all i. Inverting this, one finds with a little work that each component $q_{ti} = [1/(\sum_i \widetilde{R}_i^{-1} \hat{q}_{ti})] \widetilde{R}_i^{-1} \hat{q}_{ti}$.

With the addition of the observation process, the feasible set now becomes

$$\begin{aligned} Q_t(u_{[0,t)}, y_{[0,t)}) = \big\{ q \in Q(\mathcal{X}) : \exists \vec{w}_{[0,t)} \in [W^n]^t \text{ such that } q_0 \in Q(\mathcal{X}) \\ \text{where } q_0 \text{ is given by backward propagation (3.24)} \\ \text{with } q_t = q \, \big\} \end{aligned} \tag{3.23}$$

where

$$\begin{aligned} q_{r-1} &= \hat{G}^{-1}(q_r, u_{r-1}, \vec{w}_{r-1}, y_{r-1}) \\ &\doteq \tfrac{1}{\widehat{R}'(y_{r-1}, u_{r-1}, \vec{w}_{r-1}) q_r} D^{-1}(y_{r-1}, u_{r-1}, \vec{w}_{r-1}) \widetilde{P}^{-T}(u_{r-1}, \vec{w}_{r-1}) q_r \end{aligned} \tag{3.24}$$

where $\widehat{R}_i(y_{r-1}, u_{r-1}, \vec{w}_{r-1}) \doteq 1/[\widetilde{R}_i(y_{r-1}, u_{r-1}, \vec{w}_{r-1})]$.

The information state definition for $t > 0$ becomes

$$\mathcal{I}_t(q; u_{[0,t)}, y_{[0,t)}) \doteq \begin{cases} \sup_{q_0 \in Q_0^{q, u_{[0,t)}}} \mathcal{I}_0(q_0) & \text{if } q \in Q_t(u_{[0,t)}, y_{[0,t)}); \\ -\infty & \text{otherwise.} \end{cases} \tag{3.25}$$

where for some $Q_0 \subset Q(\mathcal{X})$ (given by the choice of $\mathcal{I}_0$),

$$Q_0^{q, u_{[0,t)}} \doteq \{ \widetilde{q} \in Q_0 : \exists \vec{w}_{[0,t)} \in [W^n]^t \text{ such that } q_0 = \widetilde{q}, \text{ using (3.24) and initial } q_t = q \}.$$

Note that we will often suppress the dependence of Q_t on $u_{[0,t)}, y_{[0,t)}$. We note that the information state above is a refined form of the definition in (McEneaney 2004)

$$\mathcal{I}_t(q; u_{[0,t)}, y_{[0,t)}) \doteq \begin{cases} \sup_{q_0 \in Q_0^{q,u_{[0,t)}}} \max_{\vec{w}_{[0,t)} \in [W^n]^t} \mathcal{I}_0(q_0) & \text{if } q \in Q_t(u_{[0,t)}, y_{[0,t)}); \\ -\infty & \text{otherwise.} \end{cases}$$

We remove the max over the Red control history in the refined form and reproduce the same results as obtained in (McEneaney 2004). The following results hold with $\mathcal{I}_t$ given by (3.25)

Lemma 3.3.1. *(McEneaney 2004)*

$$Q_{t+1} = \left\{ q \in Q(\mathcal{X}) : \exists q_t \in Q_t,\ \vec{w} \in W^n \ \textit{such that}\ q = G(y_t, u_t, \vec{w})[q_t] \right\} \tag{3.26}$$

where

$$G(y, u, \vec{w})[q] \doteq \hat{G}(y, u, \vec{w}, q) = \widetilde{P}^T(u, \vec{w}) \frac{1}{\widetilde{R}^T(y,u,\vec{w})q} D(\widetilde{R}(y, u, \vec{w}))q.$$

Lemma 3.3.2. *(McEneaney 2004)*

$$Q_t \neq \emptyset, \ \forall\ t \in \bar{T} \tag{3.27}$$

Lemma 3.3.3. *(McEneaney 2004)*

$$\mathcal{I}_{t+1}(q) = \begin{cases} \max_{\vec{w} \in W_t^q} \max_{\widehat{q} \in S_t^{\vec{w},q}} \mathcal{I}_t(\widehat{q}) & \textit{if}\ W_t^q \neq \emptyset; \\ -\infty & \textit{otherwise} \end{cases} \tag{3.28}$$

where

$$\begin{aligned} S_t^{\vec{w},q} &= S_t^{\vec{w},q}(u_{[0,t]}, y_{[0,t]}) = \{\widehat{q} \in Q_t : q = G(y_t, u_t, \vec{w})[\widehat{q}]\} \\ W_t^q &= \{\vec{w} \in W^n : S_t^{\vec{w},q} \neq \emptyset\}. \end{aligned} \tag{3.29}$$

In fact, one can also have an equivalent result to Lemma 3.3.3 in the following form using (3.25):

Theorem 3.3.1.

$$\mathcal{I}_{t+1}(q) = \begin{cases} \max_{\widehat{q} \in S_t^q} \mathcal{I}_t(\widehat{q}) & \textit{if}\ S_t^q \neq \emptyset; \\ -\infty & \textit{otherwise} \end{cases} \tag{3.30}$$

where

$$S_t^q = S_t^q(u_{[0,t]}, y_{[0,t]}) = \{\widehat{q} \in Q_t : \, q = G(y_t, u_t, \vec{w})[\widehat{q}] \\ \textit{for some } \vec{w} \in W^n\} \tag{3.31}$$

Proof. See Appendix (3.7.1). □

Note that the information state at time t maps conditional probability distributions (conditioned on the observation process) to costs ($\in \mathbf{R} \cup \{-\infty\}$). It indicates the maximal cost (optimal from Red perspective) to generate conditional distribution q, in a Bayesian estimator, given the Blue observations up to time t. Though the above discussion is true for any form of $\mathcal{I}_0$, for the case we concentrate on here, the initial information state, $I_0 = \phi$ takes the form of a max-plus delta function (3.15). This corresponds to the situation where Red controls do not affect the initialization. For each (known) u_0 and (unknown) $\vec{w}_0$, the dynamics and observation propagation discussed above takes q_0 into some q_1. The set of all possible q_1's which may be generated by feasible $\vec{w}_0$'s is Q_1 (as indicated mathematically above). Note that the size of Q_1 is no larger than the size of W^n. Further,

$$I_1(q) = \begin{cases} 0 & \text{if } q \in Q_1; \\ -\infty & \text{otherwise.} \end{cases}$$

This defines the propagation of the information state forward in time by one time-step for this particular class of initial information state. Further for the case where the information is initialized as a sum of max-plus delta functions, for each (known) u_0 and (unknown) $\vec{w}_0$, the dynamics and observation propagation discussed above takes $q_0^k \in \tilde{Q}_0^\phi$ into some q_1^k. The set of all possible q_1^k's which may be generated by feasible $\vec{w}_0$'s is $\tilde{Q}_1^\phi$. Note that the size of $\tilde{Q}_1^\phi$ is no larger than the size of KW^n. Further,

$$I_1(q) = \begin{cases} 0 & \text{if } q \in \tilde{Q}_1^\phi; \\ -\infty & \text{otherwise.} \end{cases}$$

We rewrite the feasible set for this specific form as

$$\tilde{Q}_t(u_{[0,t)}, y_{[0,t)}) = \big\{ q \in Q(\mathcal{X}) : \, \exists \vec{w}_{[0,t)} \in [W^n]^t \text{ such that } q_0 \in \tilde{Q}_0^\phi \\ \text{where } q_0 \text{ is given by backward propagation (3.24)} \\ \text{with } q_t = q \,\big\} \tag{3.32}$$

$$\tag{3.33}$$

The information state definition becomes

$$\mathcal{I}_t(q; u_{[0,t)}, y_{[0,t)}) \doteq \begin{cases} \sup_{q_0 \in \tilde{Q}_0^{q,u_{[0,t)}}} \mathcal{I}_0(q_0) & \text{if } q \in \tilde{Q}_t(u_{[0,t)}, y_{[0,t)}); \\ -\infty & \text{otherwise.} \end{cases}$$

for $t > 0$

where

$$\tilde{Q}_0^{q,u_{[0,t)}} \doteq \{\widetilde{q} \in \tilde{Q}_0^{\phi} : \exists \vec{w}_{[0,t)} \in [W^n]^t \text{ such that } q_0 = \widetilde{q} \text{ given}$$
$$q_t = q \text{ and backward propagation (3.24)}\}.$$

Lemma 3.3.4. *If ϕ is a max-plus delta function as given by (3.15), then $I_t(q) : Q(\mathcal{X}) \to \{-\infty, 0\}$ is a max-plus sum of at most $(\# W^n)^t$ max-plus delta functions.*

Proof. The proof is obvious from the above discussion. □

Lemma 3.3.5. *If ϕ is a max-plus sum of max plus delta function as given by (3.16), then $I_t(q) : Q(\mathcal{X}) \to \{-\infty, 0\}$ is a max-plus sum of at most $K(\# W^n)^t$ max-plus delta functions.*

Proof. The proof is obvious from the above discussion. □

3.3.2 State Feedback Value Function

We now turn to the state-feedback value function. The full state of the system is now described by the true state taking values $x \in \mathcal{X}$ and Blue's conditional probability process taking values $q \in Q(\mathcal{X})$. We denote the terminal payoff for the game as $\mathcal{E} : \mathcal{X} \to \mathbf{R}$ (where of course this does not depend on the internal conditional probability process of Blue). Thus the state-feedback value function at the terminal time is

$$V_T(x, q) = \mathcal{E}(x). \tag{3.34}$$

One issue that arises is the information available to Red. One option would be to assume that it knows only the actual true state, x. However, with full knowledge of the state and observations, Red could also construct the conditional probability, q. This second model is more conservative in terms of construction of the Blue control, and this model will be used here.

The state of the state-feedback game at time t is (X_t, q_t). Blue will have access only to the probability distributions up to the current time, while Red will have access to the true state as well.

We define the strategies for Blue as follows. We continue to use the convention that interval subscripts indicate sequences; for instance, $u_{[\bar{t},r]} = \{u_r\}_{r=\bar{t}}^{r}$. Since Blue has access only to probability distributions, the set of strategies for Blue over time interval $[\bar{t}, T-1]$ is

$$\overline{\Lambda}_{[\bar{t},T-1]} = \Big\{ \overline{\lambda}_{[\bar{t},T-1]} : Q^{T-\bar{t}} \to U^{T-\bar{t}}, \text{ nonanticipative in } q. \Big\}. \tag{3.35}$$

Note that $\overline{\lambda}_{[\bar{t},T-1]}$ is nonanticipative in $q.$ if given any $t \in \{\bar{t}, \bar{t}+1, \dots, T-1\}$ and any $q_{[\bar{t},T-1]}, \tilde{q}_{[\bar{t},T-1]} \in Q^{T-\bar{t}}$ such that $q_r = \tilde{q}_r$ for all $r \le t$, then $\overline{\lambda}_t[q_{[\bar{t},T-1]}] = \overline{\lambda}_t[\tilde{q}_{[\bar{t},T-1]}]$. Further, note that $\overline{\lambda}_t$ is independent of x. More specifically, if the true state $X_t \neq \widehat{X}_t$, but $q_r = \tilde{q}_r$ for all $r \le t$, then one still has $\overline{\lambda}_t[q_{[\bar{t},T-1]}] = \overline{\lambda}_t[\tilde{q}_{[\bar{t},T-1]}]$. For notational simplicity, let $\overline{\lambda}_t \equiv \overline{\lambda}_{[t,T]}$. For reasons of robustness, we will be interested in an upper value (giving advantage to Red). Consequently, the strategy set for Red is naturally

$$\overline{\Theta}_{[\bar{t},T-1]} = \Big\{ \overline{\theta}_{[\bar{t},T-1]} : \mathcal{X}^{T-\bar{t}-1} \times Q^{T-\bar{t}} \to W^{n(T-\bar{t})}, \text{ nonanticipative in } X., q. \Big\}.$$

Note that the dependence of $\overline{\theta}_t$ on the current state, X_t, is implicit in the fact that $\vec{w}$ is a vector of length n where component i represents the control w to be played if the current state is $X_t = i$. The strategy set $\overline{\Theta}$ corresponds to the closed-loop perfect state (CLPS) information pattern (Basar & Bernhard 1991, Basar & Olsder 1982), while $\overline{\lambda}$ is similar to CLPS but with the x-portion of the state unobserved.

Since Blue knows only the $q.$ process, the best that could be achieved from Blue's perspective would be

$$V_{\bar{t}}^1(q) = \inf_{\overline{\lambda}_{[\bar{t},T-1]} \in \overline{\Lambda}_{[\bar{t},T-1]}} \sup_{\overline{\theta}_{[\bar{t},T-1]} \in \overline{\Theta}_{[\bar{t},T-1]}} \mathbf{E}_q \Big\{ \mathbf{E}[\mathcal{E}(X_T) \,|\, X_{\bar{t}} = X] \Big\} \tag{3.36}$$

where $\mathbf{E}_q$ represents expectation over X with $P(X = i) = q_i$ for all $i \in \mathcal{X}$, and the dynamics are propagated with strategies $\overline{\lambda}$ and $\overline{\theta}$. Since the above formulation is slightly nonstandard, some equivalent formulations follow.

Lemma 3.3.6. *(McEneaney 2004) The optimal Blue value, $V^1_{\bar t}$, satisfies*

$$V^1_{\bar t}(q) = \inf_{\overline{\lambda}_{[\bar t,T-1]}\in\overline{\Lambda}_{[\bar t,T-1]}} \; \max_{\vec w_{[\bar t,T-1]}\in W^{n(T-\bar t)}} \mathbf{E}_q\Big\{\mathbf{E}[\mathcal{E}(X_T)\,|\,X_{\bar t}=X]\Big\} \tag{3.37}$$

$$= \inf_{\overline{\lambda}_{[\bar t,T-1]}\in\overline{\Lambda}_{[\bar t,T-1]}} \mathbf{E}_q\Big\{ \max_{\vec w_{[\bar t,T-1]}\in W^{n(T-\bar t)}} \mathbf{E}[\mathcal{E}(X_T)\,|\,X_{\bar t}=X]\Big\} \tag{3.38}$$

$$= \inf_{\overline{\lambda}_{[\bar t,T-1]}\in\overline{\Lambda}_{[\bar t,T-1]}} \mathbf{E}_q\Big\{ \max_{\overline{\theta}_{[\bar t,T-1]}\in\overline{\Theta}_{[\bar t,T-1]}} \mathbf{E}[\mathcal{E}(X_T)\,|\,X_{\bar t}=X]\Big\}. \tag{3.39}$$

Define

$$M_{\bar t}(x,q,\overline{\lambda}_{[\bar t,T-1]}) \doteq \max_{\overline{\theta}_{[\bar t,T-1]}\in\overline{\Theta}_{[\bar t,T-1]}} \mathbf{E}[V_T(X_T,q_T)\,|\,X_{\bar t}=x] \tag{3.40}$$

so that

$$V^1_{\bar t}(q) = \inf_{\overline{\lambda}_{[\bar t,T-1]}\in\overline{\Lambda}_{[\bar t,T-1]}} \mathbf{E}_q\big\{M_{\bar t}(X,q,\overline{\lambda}_{[\bar t,T-1]})\big\}. \tag{3.41}$$

Noting the fact that U is finite, one sees that there exists an optimal $\overline{\lambda}^0_{[\bar t,T-1]}$ (see (McEneaney 2004)) given by

$$\overline{\lambda}^0_{[\bar t,T-1]} = \operatorname*{argmin}_{\overline{\lambda}_{[\bar t,T-1]}\in\overline{\Lambda}_{[\bar t,T-1]}} \mathbf{E}_q\big\{M_{\bar t}(X,q,\overline{\lambda}_{[\bar t,T-1]})\big\}. \tag{3.42}$$

We also define

$$V_{\bar t}(x,q) = M_{\bar t}(x,q,\overline{\lambda}^0_{[\bar t,T-1]}),$$

which one might interpret as a Red value function, but that will not be pursued here.

Now that the state-feedback value has been defined, one needs to show how it can be obtained by backward dynamic programming propagation. Let $V^d_t(x,q)$ be the function obtained by the following backward dynamic programming iteration. (Note that the d superscript notation does not indicate an index for a set, but is instead intended to denote the function obtained by this backward dynamic programming iteration.) It must be shown that $V^d_t(\cdot,\cdot) = V_t(\cdot,\cdot)$. Let $V^d_T(x,q) = \mathcal{E}(x)$ for all $x\in\mathcal{X}$ and $q\in Q(\mathcal{X})$. We now suppose that one has $V^d_{t+1}(\cdot,\cdot)$, and demonstrate how one obtains $V^d_t(\cdot,\cdot)$.

1. First, let the vector-valued function $\vec M_t$ be given component-wise by

$$[\vec M_t]_x(q,u) = \max_{\vec w\in W^n}\Big[\sum_{j\in\mathcal{X}} \widetilde{P}_{xj}(u,\vec w)V^d_{t+1}(j,q'(q,u,\vec w))\Big] \tag{3.43}$$

where

$$q'(q,u,\vec{w}) = \widetilde{P}^T(u,\vec{w})q \tag{3.44}$$

and the optimal $\vec{w}$ is

$$\vec{w}_t^0 = \vec{w}_t^0(x,q,u) = \operatorname*{argmax}_{\vec{w}\in W^n}\Big\{\sum_{j\in\mathcal{X}} \widetilde{P}_{xj}(u,\vec{w})V_{t+1}^d(j,q'(q,u,\vec{w}))\Big\}. \tag{3.45}$$

2. Then define L_t as

$$L_t(q,u) = q^T\vec{M}_t(q,u), \tag{3.46}$$

and note that the optimal u is

$$u_t^0 = u_t^0(q) = \operatorname*{argmin}_{u\in U} L_t(q,u) = \operatorname*{argmin}_{u\in U} q^T\vec{M}_t(q,u). \tag{3.47}$$

3. With this, one obtains the next iterate from

$$V_t^d(x,q) = \sum_{j\in\mathcal{X}} \widetilde{P}_{xj}(u_t^0,\vec{w}_t^0)V_{t+1}^d(j,q'(q,u_t^0,\vec{w}_t^0)) = [\vec{M}_t]_x(q,u_t^0) \tag{3.48}$$

and the corresponding best achievable expected result from the Blue perspective is

$$V_t^{d,1}(q) = q^T\vec{M}_t(q,u_t^0). \tag{3.49}$$

Consequently, for each $t\in\{0,1,\ldots,T\}$ and each $x\in\mathcal{X}$, $V_t^d(x,\cdot)$ is a piecewise constant function over simplex $Q(\mathcal{X})$. (Once we obtain $V_t \equiv V_t^d$, this will obviously imply the corresponding piecewise constancy of the state-feedback value function V_t.) Due to this piecewise constant nature, propagation is relatively straight-forward (more specifically, it is finite-dimensional in contradistinction to the general case). However, this is slightly less critical than the propagation issue for the information state, since the state-feedback value may be pre-computed, while the information state must be propagated in real-time.

We now show that in fact, $V_t \equiv V_t^d$ for all $t\in[0,T]$. By definition, $V_T^d(x,q) = \mathcal{E}(x) = V_T(x,q)$ for all $x\in\mathcal{X}$ and $q\in Q(\mathcal{X})$. The next step in proving the equivalence is to prove that V_t satisfies the dynamic programming principle (DPP). For the problem considered here, the DPP takes the form of the following theorem.

Theorem 3.3.2. *(McEneaney 2004) Let $0\le t<r\le T$. Then*

$$V_t(x,q) = M_t(x,q,\overline{\lambda}^0_{[t,T-1]}) = \widetilde{M}_{[t,r)}(x,q,\widetilde{\overline{\lambda}}^0_{[t,r-1]}) \tag{3.50}$$

where

$$\widetilde{M}_{[t,r)}(x, q, \overline{\lambda}_{[t,r-1]}) = \max_{\overline{\theta}_{[t,r-1]} \in \overline{\Theta}_{[t,r-1]}} \mathbf{E}\big[V_r(X_r, q_r) \,|\, X_t = x\big] \tag{3.51}$$

$q_t = q$, *and*

$$\widetilde{\overline{\lambda}}^{0}_{[t,r-1]} = \operatorname*{argmin}_{\overline{\lambda}_{[t,r-1]} \in \overline{\Lambda}_{[t,r-1]}} \mathbf{E}_q \left\{ \widetilde{M}_{[t,r)}(X, q, \overline{\lambda}_{[t,r-1]}) \right\}. \tag{3.52}$$

Using this DPP (Theorem 3.3.2), one can inductively obtain the equivalence of our defined value and the output of the DPP iteration (3.43)–(3.49).

Theorem 3.3.3. *(McEneaney 2004)*

$$V_t = V_t^d \qquad \forall\, t \in [0, T]$$

and of course

$$V_t^1 = V_t^{d,1} \qquad \forall\, t \in [0, T].$$

Again, this validates the DPP iteration (3.43)–(3.49) as a means for computing the state-feedback value function, V_t.

3.3.3 Robustness

The remaining component of the computation of the control at each time instant is now discussed. The control computation for such games is typically performed via the use of the Certainty Equivalence Principle (Basar & Bernhard 1991, Helton & James 1999)). When the Certainty Equivalence Principle holds, the information state and state-feedback value function can be combined to obtain the "optimal" controls which can be shown to be robust in a sense to be discussed below. The chief gain is that this allows one to compute a state-feedback controller ahead of time, and then only propagate the information state "estimator" forward in time rather than computing the control as a function of the information state in real time. Otherwise, the computational cost would be prohibitive.

To simplify notation, note that by (3.46), (3.43) and Theorem 3.3.3 for any u,

$$L_t(q, u) = \mathbf{E}_q \left[\max_{\vec{w} \in W^n} \sum_{j \in \mathcal{X}} \widetilde{P}_{Xj}(u, \vec{w}) V_{t+1}(j, q'(q, u, \vec{w})) \right]$$

where the notation $q'(q, u, \vec{w})$ is defined in (3.44). Let us hypothesize that the optimal control for Blue is

$$u_t^m \doteq \operatorname*{argmin}_{u \in U} \left[\max_{q \in Q(\mathcal{X})} \{ \mathcal{I}_t(q) + L_t(q, u) \} \right]. \tag{3.53}$$

We will assume that u^m is a strict minimizer. Note that u^m is a *strict minimizer* of a function $f(u)$ if $f(u^m) < f(u)$ for all $u \neq u^m$. One has obvious robust game inequalities such as the following:

Lemma 3.3.7. *(McEneaney 2004) Suppose u_t^m is a strict minimizer. Then, given any $\tilde{u}_t \neq u_t^m$, $\exists$ q^1, $\vec{w^1}$ and $\epsilon > 0$ such that*

$$\begin{aligned} &\{\mathcal{I}_t(q^1) + \mathbf{E}_{q^1}[\sum_{j\in\mathcal{X}} \widetilde{P}_{xj}(\tilde{u}_t, \vec{w^1})V_{t+1}(j, q^{'}(q^1, \tilde{u}_t, \vec{w^1}))]\} \\ &> \max_{q\in Q(\mathcal{X})}\{\mathcal{I}_t(q) + \mathbf{E}_q \max_{\vec{w}\in W^n}[\sum_{j\in\mathcal{X}} \widetilde{P}_{xj}(u_t^m, \vec{w})V_{t+1}(j, q'(q, u_t^m, \vec{w}))]\} + \epsilon \end{aligned} \tag{3.54}$$

Lemma 3.3.8. *Let $\mathcal{I}_0$,$u_{[0,t)}$ and $y_{[0,t)}$ be given. Suppose u_t^m is a strict minimizer. Then, given any $\tilde{u}_t \neq u_t^m$, $\exists$ $q_0^1 \in Q(\mathcal{X})$, $\vec{w}_t^1 \in [W^n]^t$ and $\epsilon > 0$ such that*

$$\begin{aligned} &\mathbf{E}_{q_t^{''}}\{\mathcal{I}_0(q_0^1) + [\sum_{j\in\mathcal{X}} \widetilde{P}_{Xj}(\tilde{u}_t, \vec{w}_t^1)V_{t+1}(j, q^{'}(q_t^{''}, \tilde{u}_t, \vec{w}_t^1))]\} - \epsilon \\ &> \max_{q_0\in Q(\mathcal{X})} \max_{\vec{w}_{[0,t]}\in[W^n]^{t+1}} \mathbf{E}_{q_t^{'}}\{\mathcal{I}_0(q_0) + \sum_{j\in\mathcal{X}} \widetilde{P}_{Xj}(u_t^m, \vec{w}_t)V_{t+1}(j, q'(q_t^{'}, u_t^m, \vec{w}_t))\} \end{aligned} \tag{3.55}$$

Proof. See Appendix (3.7.2). □

In order to prove robustness with respect to the value of the game and not just $\{\mathcal{I}_t(.) + L_t(.)\}$, one must first define an imperfect observation value function in terms of the worst-case expected cost (from the Blue point of view). In order to make this section more readable, we will begin by writing down this value function, and then describe the terms within it rather than vice-versa. For technical reasons, it appears best to work with the following value function. This value at any time $\bar{t}$ is

$$Z_{\bar{t}} \doteq \sup_{q_{\bar{t}}\in Q_t} \inf_{\lambda_{[\bar{t},T-1]}\in\Lambda_{[\bar{t},T-1]}} \sup_{\theta_{[\bar{t},T-1]}\in\theta_{[\bar{t},T-1]}} \Big[I_{\bar{t}}(q_{\bar{t}}) + \mathbf{E}_{q_{\bar{t}}}\left\{\mathbf{E}[\mathcal{E}(X_T)|\, X_{\bar{t}} = X]\right\}\Big]. \tag{3.56}$$

$$= \sup_{q_{\bar{t}}\in Q_t} \Big[I_{\bar{t}}(q_{\bar{t}}) + \inf_{\lambda_{[\bar{t},T-1]}\in\Lambda_{[\bar{t},T-1]}} \sup_{\theta_{[\bar{t},T-1]}\in\theta_{[\bar{t},T-1]}} \mathbf{E}_{q_{\bar{t}}}\left\{\mathbf{E}[\mathcal{E}(X_T)|\, X_{\bar{t}} = X]\right\}\Big]. \tag{3.57}$$

The expectation uses the (Blue) assumption that the distribution of $X_{\bar{t}}$ is $q_{\bar{t}}$ for each $q_{\bar{t}} \in Q_{\bar{t}}$ and is taken not only over $X_{\bar{t}}$ but also over all observation and dynamic noise from time $\bar{t}$ to terminal time T. Note that this is not a full upper value in that the supremum over $q_{\bar{t}}$ occurs outside the infimum over Blue controls $\lambda_{[\bar{t},T-1]}$. The strategy set for Blue is

$$\Lambda_{[\bar{t},T-1]} = \Big\{ \lambda_{[\bar{t},T-1]} : Y^{T-\bar{t}} \to U^{T-\bar{t}}, \text{ nonanticipative in } y_{\cdot-1} \Big\}$$

where "nonanticipative in $y_{\cdot-1}$" is defined as follows. A strategy, $\lambda_{[\bar{t},T-1]}$ is nonanticipative in $y_{\cdot-1}$ if given any $t \in (\bar{t}, T-1]$ and any sequences $y_\cdot, \tilde{y}_\cdot$ such that $y_r = \tilde{y}_r$ for all $r \in [\bar{t}, t-1]$, one has $\lambda_t[y] = \lambda_t[\tilde{y}]$. Note that since the infimum over $\lambda_{[\bar{t},T-1]}$ in (3.56) occurs inside the supremum over $q_{\bar{t}}$, the "optimal" choice of λ may depend on $q_{\bar{t}}$. Also note that the "optimal" choice of $\lambda_{[\bar{t},T-1]}$ may depend on $I_{\bar{t}}(\cdot)$. The strategy set for Red (neglecting $q_{\bar{t}}$ as a Red control) is naturally

$$\Theta_{[\bar{t},T-1]} = \Big\{ \theta_{[\bar{t},T-1]} : Y^{T-\bar{t}} \to W^{n(T-\bar{t})}, \text{ nonanticipative in } y_{\cdot-1} \Big\}. \tag{3.58}$$

Also, $Q_t = Q_t(u_{[0,t)}, y_{[0,t)})$ as given in (3.23). Since the supremum over $\theta_{[\bar{t},T-1]}$ is inside the infimum, and the $\vec{w}_{[\bar{t},T-1]}$ process is a feedback on the state, then as in Lemma 3.3.6, one can replace the supremum over $\theta_{[\bar{t},T-1]} \in \Theta_{[\bar{t},T-1]}$ with a maximum over $\vec{w}_{[\bar{t},T-1]} \in W^{n(T-\bar{t})}$, and so

$$Z_{\bar{t}} = \sup_{q_{\bar{t}} \in Q_t} \; \inf_{\lambda_{[\bar{t},T-1]} \in \Lambda_{[\bar{t},T-1]}} \; \max_{\vec{w}_{[\bar{t},T-1]} \in W^{n(T-\bar{t})}} \left[I_{\bar{t}}(q_{\bar{t}}) + \mathbf{E}_{q_{\bar{t}}} \left\{ \mathcal{E}(X_T) \right\} \right]. \tag{3.59}$$

The first step in obtaining the robustness result is to show that the value, $Z_{\bar{t}}$ has the following representation

Theorem 3.3.4. *(McEneaney 2004)*

$$Z_{\bar{t}} = \sup_{q_{\bar{t}} \in Q_{\bar{t}}} \left[I_{\bar{t}}(q_{\bar{t}}) + V^1_{\bar{t}}(q_{\bar{t}}) \right] \qquad \forall \, \bar{t} \in [0, T]. \tag{3.60}$$

In other words, the game value $Z_{\bar{t}}$ is the supremum of the sum of the information state, $I_{\bar{t}}$, and the optimal expected state-feedback value, $V^1_{\bar{t}}$, from Blue's perspective. It is interesting to note that in the max-plus algebra (Cuninghame-Green 1979), (3.60) takes the form

$$Z_{\bar{t}} = \int^{\oplus}_{Q_{\bar{t}}} V^1_{\bar{t}}(q) \otimes I_{\bar{t}}(q) \, dq \tag{3.61}$$

where $\int^{\oplus}_{A}$ indicates max-plus integration over set A. In other words, (3.61) is the max-plus expectation of $V^1_{\bar{t}}$ with respect to max-plus probability $I_{\bar{t}}$ (see (Akian 1999), (Fleming 2004) for example).

Using Theorem 3.3.4, one can show (McEneaney 2004) that

$$Z_{\bar{t}} = \sup_{q_{\bar{t}} \in Q_t} \Big[I_{\bar{t}}(q_{\bar{t}}) + \min_{u \in U} L_{\bar{t}}(q_{\bar{t}}, u) \Big] = \sup_{q_{\bar{t}} \in Q_t} \min_{u \in U} \Big[I_{\bar{t}}(q_{\bar{t}}) + L_{\bar{t}}(q_{\bar{t}}, u) \Big]. \tag{3.62}$$

In order to obtain the Robustness/Certainty Equivalence result to follow, it is sufficient to make the following Saddle Point Assumption. We assume that

$$\sup_{q_{\bar{t}} \in Q_t} \min_{u \in U} \Big[I_{\bar{t}}(q_{\bar{t}}) + L_{\bar{t}}(q_{\bar{t}}, u) \Big] = \min_{u \in U} \sup_{q_{\bar{t}} \in Q_t} \Big[I_{\bar{t}}(q_{\bar{t}}) + L_{\bar{t}}(q_{\bar{t}}, u) \Big] \qquad \forall\, \bar{t} \in [0, T]. \tag{$A5.1$}$$

With Assumption (A5.1), (3.62) becomes

$$Z_{\bar{t}} = \min_{u \in U} \sup_{q_{\bar{t}} \in Q_t} \Big[I_{\bar{t}}(q_{\bar{t}}) + L_{\bar{t}}(q_{\bar{t}}, u) \Big]. \tag{3.63}$$

Finally, after some work , one obtains the robustness result:

Theorem 3.3.5. *(McEneaney 2004) Let $\bar{t} \in \{0, T-1\}$. Let $\mathcal{I}_0$, $u_{[0,\bar{t}-1]}$ and $y_{[0,\bar{t}-1]}$ be given. Let the Blue control choice, $u^m_{\bar{t}}$, given by (3.53) be a strict minimizer. Suppose Saddle Point Assumption $(A5.1)$ holds. Then, given any Blue strategy, $\lambda_{[\bar{t},T-1]}$ such that $\lambda_{\bar{t}}[y.] \neq u^m_{\bar{t}}$, there exists $\varepsilon > 0$, $q^\varepsilon_{\bar{t}}$ and $\vec{w}^\varepsilon_{[\bar{t},T-1]}$ such that*

$$\sup_{q \in Q_{\bar{t}}} \{ \mathcal{I}_{\bar{t}}(q) + L_{\bar{t}}(q, u^m_{\bar{t}}) \} = Z_{\bar{t}} \le \mathcal{I}_{\bar{t}}(q^\varepsilon_{\bar{t}}) + \mathbf{E}_{q^\varepsilon_{\bar{t}}} \Big\{ \mathbf{E}[\mathcal{E}(X^\varepsilon_T) \,|\, X^\varepsilon_{\bar{t}} = X] \Big\} - \varepsilon \tag{3.64}$$

where X^ε denotes the process propagated with control strategies $\lambda_{[\bar{t},T-1]}$ and $\vec{w}^\varepsilon_{[\bar{t},T-1]}$.

Remark 3.3.1. *Theorem 3.3.5 also serves as a basis for referring to $\mathcal{I}_t$ as an information state – at least in the case where Assumption $(A5.1)$ holds.*

3.4 Partially-Observed Game: MAG Revisited

We introduce some additional parameters for the extending the modelling of the example discussed in the state-feedback case.

- p^n: Probability of observing a non-stealthy Red entity.
- p^s: Probability of observing a stealthy Red entity.
- pf: Probability of a observing a Red decoy.

- p_s: Probability of all Red entities on a side being stealthy. This is a model parameter used internally by the Blue controller when Blue uses the naïve approach.

Recall that for the MAG example in the complete-information state-feedback case, the optimal control for Red was dependent on the two cases, A_D and A_I. In particular, for the case A_I, optimal Red control in the complete-information state-feedback case was to use any $\bar{w}^k \in W$. For the A_D case, Red entities on both routes were made stealthy with the optimal Red choice, $\bar{w}^1 \in W$. The results to follow in this chapter and chapter 4 will be for the A_I case and we assume hereon that the Red player will use the 'RG' strategy for its control computation (otherwise one will use a random red control which is not expected from an intelligent adversary). The following study will be focused on comparing the deception-robustness Blue approach compared to the standard existing approaches. We will focus on two main approaches namely the 'MLS' and the deception-robust for the major part of the remaining discussion.

Note that the Red player may deceive Blue by the use of stealth and decoys when appropriate. Recall that Red can start at one of the following initial states, X_0^R: $(0,4)$, $(1,3)$, $(2,2)$, $(3,1)$, and $(4,0)$. The cases $(0,4)$ and $(4,0)$ are trivial from structure and $(1,3)$ is axially symmetric to $(3,1)$. As shown in the state-feedback game section $(1,3)$ gives higher mean-sample payoff in the partially-observed games, so we discuss simulation results for this Red initial state (unknown to Blue). Note that the initial Red state also affects Blue's observations (and consequently the information state or the observation conditioned distributions). The 'RG' strategy forms a formidable Red opponent. The low probability of detecting the single Red entity on the western route relative to the likely observations on the eastern route would tend to lure a naïve Blue controller into thinking that all the Red forces are along the eastern route, and to only apply UAVs there in order to have the highest probability of stopping all the forces on this route. This is indeed the optimal Blue choice when using a Bayesian Filter (or the naïve 'MLS' approach), but the deception-robust approach does not fall for this deception and provides resistance to the western Red entities as well. Note that use of stealth is equivalent to low values of p^s. In particular we allow the observation probability for a Red decoy to be the same as for a non-stealthy Red (non-decoy). With Red RG approach, we return to Blue's control mechanism and the affect of p^s, p_s, and pf mismodelling on

Blue's expected payoff. Recall that Blue is not aware of the true initial Red state. The objective for the rest of the chapter is to explore the merits of using a complex control algorithm like deception-robust over the 'MLS' approach.

Assumptions and Approximations

The computation of $V_t(x,q)$ as outlined in (3.43)-(3.49) requires that one already has pre-computed $V_{t+1}(\bar{x},\bar{q})$, $\forall$ $(\bar{x},\bar{q}) \in \mathcal{X} \times Q_{t+1}$. In the one step recursion from terminal time T to $T-1$, since $V_T(x,q)$ only depends on x, one can obtain the approximation to the value at time $T-1$ by the iterative scheme outlined in (3.43)-(3.49). For the example problem under consideration $\#\mathcal{X} = 13$, but Q_t is propagated in real time and one cannot pre-compute $V_{t+1}(\bar{x},\bar{q})$ for all $(\bar{x},\bar{q}) \in \mathcal{X} \times Q_t$. One way to circumvent this is to approximate each $q \in Q_t$ by some $\tilde{q} \in \tilde{Q}$, where $\tilde{Q}$ is some predefined set of distributions (on which one can pre-compute $V(x,\tilde{q})$, for every $(x,\tilde{q}) \in \mathcal{X} \times \tilde{Q}$). One can also approximate the future cost $V_{t+1}(x,q)$ by $V_{t+1}(x)$ and then repeat the computation outlined in (3.43)-(3.46).

1.

$$[\vec{M}^a_t]_x(q,u) = \max_{\vec{w}\in W^n}\Big[\sum_{j\in\mathcal{X}} \widetilde{P}_{xj}(u,\vec{w})V_{t+1}(j)\Big] \tag{3.65}$$

and the optimal Red control in this internal Blue computation is

$$\vec{w}^a_t = \vec{w}^a_t(x,q,u) = \operatorname*{argmax}_{\vec{w}\in W^n}\Big\{\sum_{j\in\mathcal{X}} \widetilde{P}_{xj}(u,\vec{w})V_{t+1}(j)\Big\}. \tag{3.66}$$

2. Then define L^a_t as

$$L^a_t(q,u) = q^T \vec{M}^a_t(q,u), \tag{3.67}$$

and the 'approximate' Blue control u^a can then be computed as a function of q

$$u^a_t = \operatorname*{argmin}_{u\in U}\left[\max_{q\in Q_t}\{\mathcal{I}_t(q) + L^a_t(q,u)\}\right] \tag{3.68}$$

Recall that we are initializing $\mathcal{I}_0$ as the max-plus sum of max plus delta functions, so we only compute controls based on $q \in Q_t$ (and hence $\mathcal{I}_t(q) = 0$ and it drops out of the computation, since $V_t(.) > 0$ for all t). Finally

$$u^a_t = \operatorname*{argmin}_{u\in U}\left[\max_{q\in Q_t} L^a_t(q,u)\right] = \operatorname*{argmin}_{u\in U}\left[\max_{q\in Q_t} q^T \vec{M}^a_t(q,u)\right] \tag{3.69}$$

Clearly with this approximation the Blue control obtained will not have the robustness properties derived in section 3.3.3. However, one expects that using the control by (3.69) Blue will not achieve a better payoff than when using the actual robust control given by (3.53), i.e. $u^a \notin \operatorname{argmin}_{u \in U} \left[\max_{q \in Q(\mathcal{X})} \{L_t(q,u)\} \right]$. The results to follow will indicate that even with Blue using the control given by (3.69), one obtains a much lower payoff than when Blue uses any other approach outlined in the section 3.2. That clearly indicates that Blue will do no worse compared to other techniques when Blue uses the robust control given by (3.53).

Observation Model

For simplicity of exposition and to extend the masked attack problem, from complete-information game to the partial-information game in the simplest manner, we will define W^n as a natural extension of the set W definition in the state-feedback section (with the addition of using a decoy as a control). One could allow for the possible state-dependent Red control choices, $\vec{w}$, in a manner where each Red entity can be turned stealthy individually based on X_t. However, as we will see shortly, a definition of W^n that only allows the Red player to turn all entities stealthy or non-stealthy on a given route for all time (with the choice of using a decoy on each route) is sufficient to capture the affect of partial information on Blue performance and sub-optimal control decision. Let

$$\widehat{W_I} = \{\hat{w}^1, \hat{w}^2, \hat{w}^3, \hat{w}^4\} \tag{3.70}$$

where $\hat{w}^i \in \widehat{W_I}$ are defined as an extension of the corresponding $\bar{w}^i \in \overline{W}$ control (given in section 2.3) with an addition of a non-stealthy decoy on each route. For example, $\hat{w}^1$ is the extension of the Red control $\bar{w}^1$ (where Red entities operate stealthily on both routes), with a non-stealthy decoy on each side. Now define

$$W_I^n = \{\vec{w} : \vec{w}_i = w, \forall\ i \in \mathcal{X}, \text{ for the some } w \in \widehat{W_I}\} \tag{3.71}$$

Also, let

$$\widetilde{W} = \{\tilde{w}^1, \tilde{w}^2, \tilde{w}^3, \tilde{w}^4\} \tag{3.72}$$

where $\tilde{w}^i \in \widetilde{W}$ are defined as an extension of the corresponding $\bar{w}^i$ control with an addition of a non-stealthy decoy on the eastern route. For example, $\tilde{w}^2$ is the extension of the Red control $\bar{w}^2$ (where Red entities operate stealthily on the western route and non-stealthily on the eastern route), with a non-stealthy decoy on the eastern route. The Red 'RG' strategy uses $\tilde{w}^2 \in \widetilde{W}$. Also define

$$W^n = \{\vec{w} : \vec{w}_i = w, \forall\ i \in \mathcal{X}, \text{ for the some } w \in \widetilde{W}\} \tag{3.73}$$

It is very obvious by construction that W^n and W_I^n are only consisting of open loop controls for the Red player. Let us assume that the Red player will always choose to use a decoy on the eastern route only; the actual Red control belongs to $\widetilde{W}$. Also, if one chooses $pf = 0$, then one has $\widehat{W}_I = \widetilde{W}$ and $W^n = W_I^n$.

Note that if we will allow the Blue player to have access to W_I^n, then Blue has imperfect information on the Red control set. However access to W^n will give perfect information on the Red control set to the Blue player. Recall that Blue uses the W^n knowledge to assume a stochastic model for the Red control by choosing some $p_{\vec{w}}^B$ or in the deception-robust approach it uses W^n to propagate $q_.$ along each $\vec{w} \in W^n$. Clearly the lack of information on W^n, the Red state-feedback control set, will adversely affect Blue's performance. At this point we present a result to demonstrate the sensitivity of the 'MLS' approach compared to the deception-robust approach with different levels of information imperfection. Without providing any details on the parameters used in various information levels at this point, we only make brief comments on the extreme levels of imperfection. In the least imperfect scenario, the Blue player will have access to the correct Red control set W^n (and where Red is only using $w \in W$) and all the simulation parameters are perfectly known by the Blue player. In the most imperfect scenario there is mismodelling of simulation parameters by Blue. In fact, the Blue player only has access to the imperfect Red control set W_I^n whereas the Red player is actually using $\tilde{w} \in W^n$ (more specifically $\tilde{w}^2$). These levels are shown in are the extreme left and right of Figure 3.2.

Note that the 'MLS' approach ('*' curve) gives a higher mean-sample payoff compared to the 'DR' approach ('o' curve). From the lowest imperfect information level on the left to the most imperfect information level on the right, the 'MLS' approach gives a very slowly increasing mean-sample payoff with increasing imperfection levels, or slightly worse performance for Blue. On the other hand the 'DR' approach is quite robust

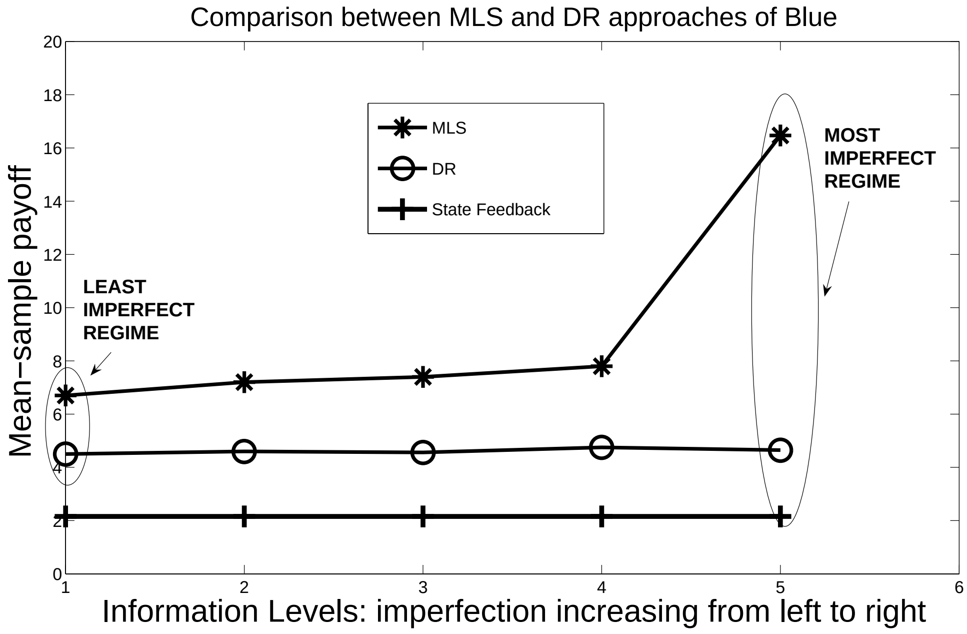


Figure 3.2: Comparing the sensitivity of ‘MLS’ approach to the ‘DR’ approach

to the levels of imperfect information and hence the advantage of using ‘DR’ approach is clearly improving as one moves towards the right extreme (with most imperfect levels of information) where there is a almost a 75% drop in the mean-sample payoff from the ‘MLS’ approach. In the least imperfect scenario there is a still a significant advantage of about 30% using ’DR” over the ‘MLS’ approach. In fact note that the ‘DR’ approach in the most imperfect information scenario achieves a lower mean-sample payoff compared to the mean-sample payoff using the ‘MLS’ approach even in the least imperfect information scenario. We will be doing most of our studies in the most imperfect information regime (unless stated otherwise). We makes two quick notes before constructing the observation model. Firstly, when Blue uses a stochastic model of Red and Red is using a fixed hand-crafted strategy like ‘RG’, there is inherent mismodelling owing to the choice of $p^B_{\vec{w}}$. Secondly, in the most imperfect information regime as defined above, we have another inherent mismodelling due to W^n_I being used by Blue instead of W^n. However our comparisons of the Blue approaches will have the same underlying modelling assumptions, so the results will be indicative of the sensitivity of each approach under partially-observed scenario.

We now construct an observation model by denoting the observation probability on each side as a function of $\bar{w} \in W$ only (neglecting the presence of decoys or $pf = 0$). This will only capture the affect of using stealth as a Red control affecting Blue's observation process. Using the notation defined earlier, the observation probability for the left side can be defined as

$$p_o^L(w) \doteq \mathbf{1}_{W_L^s}(w)p^s + \mathbf{1}_{W_L^n}(w)p^n. \tag{3.74}$$

Similarly the observation probability for the right side be defined as

$$p_o^R(w) \doteq \mathbf{1}_{W_R^s}(w)p^s + \mathbf{1}_{W_R^n}(w)p^n \tag{3.75}$$

Defining y in terms of the components on the western and the eastern route as $y = (y_1, y_2)$, we assume that the observation on each side is independent of the other; the random measurement one one side, y_1, is independent of the random measurement on the other side, y_2 (but obviously dependent on the Red control). Further with no dependence on u, the observations have no dependence on X^B (by the dynamics of the masked attack example). Then, given $X_t = (X^R, X^B)$, with $X^R = (i_1, i_2)$, the observation probability as a function of Red control w (in the absence of decoys) is given by:

$$\Pr(y|X^R)[w] = \Pr((y_1, y_2)|(i_1, i_2))[w] = [\Pr(y_1|i_1)]\,[w]\,[\Pr(y_2|i_2)]\,[w].$$

The terms on the each route are computed using (3.74) and (3.75), for example,

$$\Pr(y_1|i_1)[w] = \begin{pmatrix} i_1 \\ y_1 \end{pmatrix} (p_o^L)^{y_1}(1 - p_o^L)^{i_1 - y_1}. \tag{3.76}$$

gives the observation probability on the western route. Then using (3.76) (and similar definition of $\Pr(y_2|i_2)[w]$) one gets

$$\Pr(y|X^R)[w] = \begin{pmatrix} i_2 \\ y_2 \end{pmatrix} \begin{pmatrix} i_2 \\ y_2 \end{pmatrix} (p_o^L)^{y_1}(p_o^R)^{y_2}(1 - p_o^L)^{i_2 - y_2}(1 - p_o^R)^{i_2 - y_2}$$

The above equation is now extended to include presence of decoys on each side ($pf \neq 0$). We now choose $\vec{w} \in W_I^n$ which by (3.71) is equivalent to choosing $\hat{w} \in \widehat{W}_I$. Also, by definition $\hat{w} \in \widehat{W}_I$, Red employs a decoy on both routes as an extension of $\bar{w} \in W$.

Table 3.1: Comparing different Blue approaches

pf parameter	Mean-sample payoff			
	'MLS'	'HB'	'RS'	'DR'
0.3	16.24	8.62	13.86	4.86
0.4	15.88	8.84	14.12	5.12
0.5	16.44	9.24	14.32	4.68
0.6	15.96	8.58	14.48	4.76
0.7	15.82	9.38	13.18	5.08
0.8	16.06	9.36	13.04	5.48

Then, the observation probabilities using decoys on each route are derived from their corresponding observation probabilities without using decoy as below:

$$\Pr(y_k|X_k^R)[\hat{w}] = \begin{cases} \{\Pr(y_k - 1|X_k^R)[w]\}pf + \{\Pr(y_k|X_k^R)[w]\}(1-pf) & \text{if } y_k > 0; \\ \{\Pr(y_k|X_k^R)[w]\}(1-pf) & \text{otherwise.} \end{cases} \tag{3.77}$$

Then using $k = 1$ and $k = 2$ in (3.77) gives the observation probability in presence of decoys on the western and the eastern route respectively.

3.5 Comparing Different Blue Approaches

We present some results for the A_I case with some of the parameter values as follows, $p_2^N = p_2^S = 0.85$, $p_1^N = p_1^S = 0.425$, $p^s = 0.2$, $p^n = 0.8$, $pf_T = 0.8$, and $\kappa = 0.6$. Note that we set all the simulation and Blue parameters to be the same except the false alarm parameter, i.e., $pf_T \neq pf_B$. We vary the parameter pf_B and the simulation results using each approach as a function of pf_B are given in Table 3.1.

There are several interesting things that these results indicate. Firstly, the mean-sample payoff for each approach is not very sensitive to the variation in pf_B. Note that we are in the most imperfect regime; very low probability of detecting the stealthy Red entities ($p^s = 0.2$) and the Blue player using the set W_I^n for its control computation. The mean-sample payoff is determined by the Blue control sequence alone since we are in the A_I case. The results indicate that there is no change in the Blue player's strategy using any of these approaches for the specifically chosen parameter regime. The 'DR' approach achieves the minimum mean-sample payoff and is the optimal Blue control.

The main difficulty with using the 'RS' approach is a choice of κ, which may be obtained for a given problem after possibly lot of simulations and trial and error. For the example studied in this chapter we varied κ between 0 and 10 to demonstrate the nature of the risk-sensitive approach in general. Firstly, for the case $\kappa = 0$, we have the risk-sensitive approach equivalent to the naïve approach; apply the state-feedback control at the 'MLS' estimate. As κ increases we expect the approach to achieve a lower mean-sample payoff for Blue, since it is taking into account the expected future cost as a risk-sensitive measure. Note however that in the adversarial environment the effect of the Red player's control on the Blue player's observations has more complex consequences than that of random noise. The risk-averse approach gets the best mean-sample payoff for Blue with κ between 0.5 and 0.6 (note again that this choice will be problem specific). As κ increases beyond this point, the mean-sample payoff begins increasing, and has a horizontal asymptote which corresponds to a Blue controller which ignores all the observations and assumes the worst-case possible Red configuration. The 'RS' approach is a useful approach if one has a structured methodology to find the optimal κ.

Referring to Table 3.1, we also note that the 'MLS' approach is the worst approach for Blue. Surprisingly, the heuristic 'HB' approach gives a lower mean-sample payoff compared to the 'MLS' and the 'RS' methods. Their is some structural similarity between the 'DR' and the 'RS' methods which might provide some explanation for this result. The optimal control computation using the 'HB' approach given by (3.13)-(3.14) computes the expected payoff using the state-feedback value V_{t+1} and $p^B_{\vec{w}}$ (like the term $L^a(q, u)$ except that here their is no maximization over $\vec{w} \in W^n$). Our approximation of the 'DR' approach computes a very similar expected cost but maximizes over the Red state-feedback control space instead of using $p^B_{\vec{w}}$. Of course, finally the max over $q_t \in Q$ is computed to give the approximation of the robust controller, see (3.68). The computation using 'HB' approach clearly does not reason for the robustness at all, unlike the 'DR' approach. There is no proven theory behind the heuristic 'HB' approach and it seems to be 'loosely' inspired by similarity in structure to the 'DR' approach. Hence, we will explore the difference in performance between the simplest control approach (the 'MLS' approach) and the most complex one (the 'DR' approach); we will focus mainly on the 'MLS' and the 'DR' approach for the simulation results to appear in the following discussions.

3.5.1 Initializing the Controller

Instead of taking a vector form of q, here we represent q as a matrix to reflect that the Red player state is composed of number of Red entities on two routes, i.e. $X^R = (r^1, r^2)$. The Blue player will model the possibility of a Red decoy on each route with the knowledge that there are at most 4 Red teams (and at least 1, otherwise there is no game). The size of q in the matrix representation is (5×5) (allowing for $X^R = (0, a)$ or $X^R = (b, 0)$). So the entry $q_{i,j}$ would give the probability for the state $X^R = (i-1, j-1)$ (obviously $q_{i,j} = 0$, if $i + j > 5$). A uniform distribution q_0^U would then be of the form ($p_1^c = \frac{1}{14}$):

$$q_0^U = \begin{bmatrix} 0 & p_1^c & p_1^c & p_1^c & p_1^c \\ p_1^c & p_1^c & p_1^c & p_1^c & 0 \\ p_1^c & p_1^c & p_1^c & 0 & 0 \\ p_1^c & p_1^c & 0 & 0 & 0 \\ p_1^c & 0 & 0 & 0 & 0 \end{bmatrix}$$

or if Blue uses non-zero mass only for the Red states such that total number of Red entities is at least two, then ($p_1^c = \frac{1}{14}$):

$$\bar{q}_0^U = \begin{bmatrix} 0 & 0 & p_1^c & p_1^c & p_1^c \\ 0 & p_1^c & p_1^c & p_1^c & 0 \\ p_1^c & p_1^c & p_1^c & 0 & 0 \\ p_1^c & p_1^c & 0 & 0 & 0 \\ p_1^c & 0 & 0 & 0 & 0 \end{bmatrix}$$

Blue could also choose a distribution with higher mass corresponding to a higher number of Red teams as q_0^{NU}:

$$q_0^{NU} = \begin{bmatrix} 0 & p_1^c & p_2^c & p_3^c & p_4^c \\ p_1^c & p_2^c & p_3^c & p_4^c & 0 \\ p_2^c & p_3^c & p_4^c & 0 & 0 \\ p_3^c & p_4^c & 0 & 0 & 0 \\ p_4^c & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.78}$$

One set of constants in equation (3.78) can be assigned the following values, $p_1^c = \frac{1}{20}$, $p_2^c = \frac{1}{15}$, $p_3^c = \frac{3}{40}$, and $p_4^c = \frac{2}{25}$. This would assign a mass of 0.1, 0.2, 0.3 and 0.4 mass corresponding to total Red entities ($r_1 + r_2$) being 1, 2, 3, and 4 respectively. Recall that we will be mainly considering ϕ which are sums of max-plus delta functions (these can alternatively be represented by sets of distributions). Since an observation at any state with 0 mass will lead to some computational issues and ill-defined propagation, we can replace the 0's with reasonably small ϵ. We will refer to these type of distributions as $q_{i,j}^{G,0}$, a delta function distributions corresponding to the state $X^R = (i-1, j-1)$.

$$q_{2,4}^G = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1-14\epsilon & 0 \\ \epsilon & \epsilon & \epsilon & 0 & 0 \\ \epsilon & \epsilon & 0 & 0 & 0 \\ \epsilon & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.79}$$

The decision on what is the best set of initial distributions depends on several factors, of which, information on Red's initial state plays a critical role.

Recall that we consider only the case where the initial information state in the deception-robust controller, $\mathcal{I}_0 = \phi$, is zero on finitely many $q_k \in \tilde{Q}_0^\phi$ (in the case of max-plus sum of max-plus delta functions) and $-\infty$ elsewhere. Choosing the proper ϕ at the outset is important in ensuring good future behavior of the controller. This issue is a significant generalization over the analogous issue for standard methods where one simply needs to pick a reasonable initial probability distribution (covariance for example). Here, in the deception-robust case, one could imagine that very poor initial information might be represented by a ϕ which is zero only on the uniform distribution since the uniform distribution represents a total lack of knowledge. Alternatively, one might also represent this lack of knowledge by letting ϕ be zero (or a sufficiently high mass) on every distribution which is one at a single state and zero (or a small ϵ) elsewhere. These two possibilities represent radically different concepts for the reason for our lack of initial information of the opponent state. The first approach corresponds to a world-view wherein all of our lack of knowledge is due to unknown random variables – as though the initial state was random. The second approach corresponds to a world-view where one imagines the opponent carefully choosing its initial state with no randomness about the actual initial state at all. Thus, we see that the initialization issue is a good deal more complex in the deception-robust controller than it is in more typical approaches. We will present some data indicating how this decision should be made.

We introduce the following notation: parameter p indexed by T (p_T) corresponds to the parameter in the simulator (or true parameter) and if indexed by B (p_B) corresponds to a parameter in Blue modelling of truth (used in state estimation and control computation). Recall that there is the inherent decoy mismodelling because Blue is using the Red control set W_I^n. We refer to the decoy modelling (or mismodelling) with the following notation:

- Type $[pf_1]$: Blue using W_I^n and $pf_{\mathrm{T}} = pf_B$.

- Type $[pf_2]$: Blue using W_I^n and $pf_{\mathrm{T}} \neq pf_B$.

Choice of the Blue initial distributions requires more analysis. If the Red player uses an intelligent strategy (non-probabilistic) for initial-state layout (on the two routes) but Blue models Red initial state layout probabilistically (or vice versa), how does such mismodelling affect Blue's performance? We discuss some simulation results to answer these mismodelling issues. We will need the following terminology:

- BD-RD: Blue uses distribution based modelling of Red initial-state layout, actual Red initial-state layout is also distribution based (naïve).

- BD-RG: Blue uses distributions based modelling of Red initial-state layout, actual Red initial-state layout is non-distribution based (intelligent, say as in RG).

- BG-RG: Blue uses max-plus sum of max-plus delta functions (with $q_{i,j}^G$ distributions) initialization to model Red initial-state layout, actual Red initial-state layout is non-distribution based (intelligent, say as in RG).

- BG-RD: Blue uses max-plus sum of max-plus delta functions (with $q_{i,j}^G$ distributions) initialization to model Red initial-state layout, Red initial-state layout is distribution based (naïve).

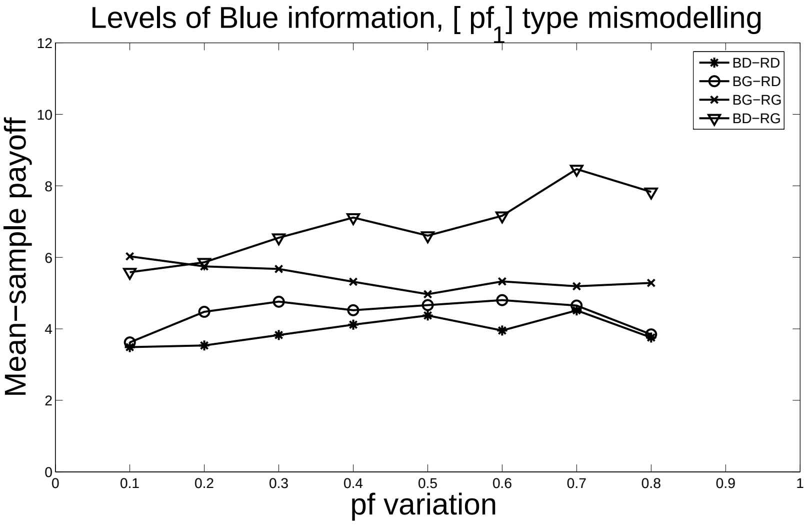


Figure 3.3: What is a good initial distribution for Blue, $[pf_1]$ mismodelling, 'DR'

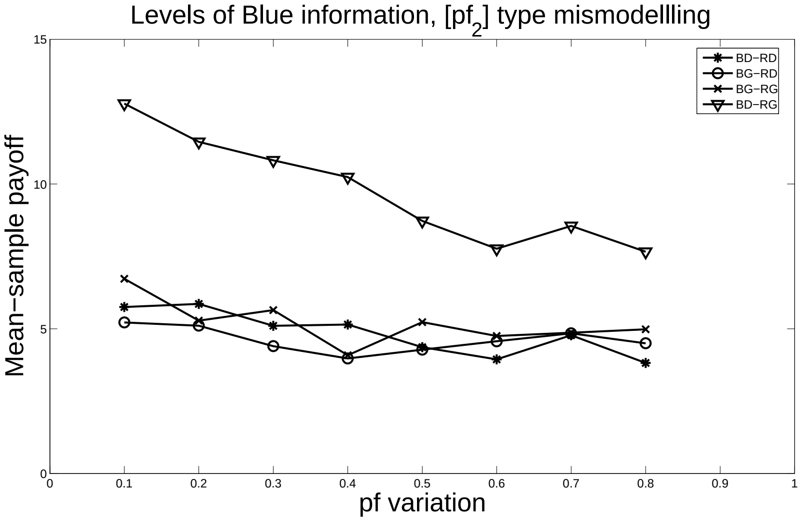


Figure 3.4: What is a good initial distribution for Blue, $[pf_2]$ mismodelling, 'DR'

In Figure 3.3, there is Type $[pf_1]$ mismodelling of false alarms whereas Figure 3.4 has Type $[pf_2]$ mismodelling of false alarms. In both these figures we have four set of data plots. In Type $[pf_1]$ mismodelling results, it is clear by comparing '*' curve with 'o' curve that one has a lower mean-sample payoff using a single probabilistic distribution as per which Red is naïvely choosing its initial state in comparison to the mean-sample payoff when using max-plus sum of max-plus delta functions ($q^G_{i,j}$ type distributions). The latter Blue approach is not lacking too much in performance and one can attribute this performance difference due to Blue's mismodelling (even with a relatively intelligent modelling) of the actual Red initial-state layout strategy. Similar results can be deduced by comparing 'x' with '∇' (or the inverted triangle) for the second case. More intuitively, lower mean-sample payoff is achieved (better for Blue), when it models the (true) intelligent Red initial-state layout appropriately, using the max-plus sum of max-plus delta functions ($q^G_{i,j}$ type distributions) as initial distributions compared to modelling Red initial-state layout probabilistically. If Blue mismodels in this case then the difference is proportionately higher indicating that in the presence of mismodelling, Blue tends to suffer more drastically when using a naïve model for an intelligent Red initial-state layout than when using an intelligent model strategy for a naïve Red initial-state layout. The results for Type $[pf_2]$ mismodelling in Figure 3.4 lead to very similar conclusions. As an aside, one may also note that a higher mean-sample payoff is achieved (better for Red) using an intelligent strategy (involving deception). This can be confirmed by comparing the plots: '∇' vs '*' and 'x' vs 'o'. Blue on the other hand will generally achieve better results by using max-plus sum of max-plus delta functions (of type $q^G_{i,j}$ at various potential states). Note that these initial distributions type are harder to prune as we will see in the pruning section 3.5.2.

We now ascertain that quality of information is more important than quantity of information. Some interesting results obtained by differentiating the quality of information contained in the initial distributions are presented here. In the first scenario, Blue models the unknown Red initial state with 3 initial distributions (one of which is a delta function at the true Red state $(1,3)$, i.e. $q^G_{2,4}$). More distributions (of type, $q^G_{i,j}$) are then added and simulation runs are repeated with the increasing number of initial distributions. The results for this scenario are given in Table 3.2.

There is no significant change in the mean-sample payoff if we keep adding more spiky distributions (to the original 3 distributions). We now compare this to the second

Table 3.2: Value of Intel in choosing the initial distributions: 'DR'

$\#Q_0$	Payoff with Intel	Payoff without Intel
3	4.84	7.24
5	4.76	6.82
7	4.92	5.56
9	5.02	5.24
11	4.88	5.02

scenario results given in Table 3.2, which are obtained with same parameters but with no initial distribution of type $q^G_{i,j}$. We use relatively flatter initial distributions of kind q_0^U, $\bar{q}_0^U$ or q_0^{NU} for the second scenario. As we increase the number of such distributions, one obtains a lower mean-sample payoff for Blue, and it approaches the mean-sample payoff that Blue achieves in the first scenario. This leads to the conclusion that with quality of initial information (the knowledge that Red is using an initial layout as per some intelligent strategy (like RG) one can use fewer number of initial distributions and the computational growth factor can be lowered by some factor (linearly).

Affect of Specific Knowledge or Intel About the Red Control

We now show that the 'DR' approach will work no worse (and potentially better) when Blue has some 'specific' information about Red control choices. In this example the Red player is using the RG strategy. Let's assume that Blue knows using some intel that Red will only use $w \in W^* \ (\subset W)$. Then note immediately that $\#W^*$ is M^* which is less than M, the dimension of W. Clearly one has a slower growth rate per time step by a factor of M/M^*. Correspondingly one has $\underline{\mathbf{Q}}_{\mathbf{t}} \subset \mathbf{Q_t}$, where $\underline{\mathbf{Q}}_{\mathbf{t}}$ is new set of feasible distributions at time t. Then for any $u \in U$, one has

$$\max_{q \in \underline{\mathbf{Q}}_{\mathbf{t}}} \{\mathcal{I}_t(q) + L_t(q,u)\} \leq \max_{q \in \mathbf{Q_t}} \{\mathcal{I}_t(q) + L_t(q,u)\}$$

which gives

$$\min_{u \in U} \left[\max_{q \in \underline{\mathbf{Q}}_{\mathbf{t}}} \{\mathcal{I}_t(q) + L_t(q,u)\} \right] \leq \min_{u \in U} \left[\max_{q \in \mathbf{Q_t}} \{\mathcal{I}_t(q) + L_t(q,u)\} \right].$$

This implies that Blue achieves a payoff which is no worse than using a larger set of distributions and has potential for at least saving a lot of computational time. For technical reasons we will assume that no pruning is used in this analysis. We provide a brief comparison between the 'DR' and the 'MLS' approach using result from a simulation study (for similar intel/knowledge about the Red control set). Note that Blue uses a probabilistic model for Red control strategy in the naïve approach. Let $p_{\bar{w}_k}$ be the probability of using $\bar{w}_k \in W$ Red control in Blue update algorithm. We enumerate the intel level numerically by an increasing number representing improving knowledge level of Red control set:

- 1: $p_{\bar{w}^1} = 0$, $p_{\bar{w}^2} = \frac{1}{3}$, $p_{\bar{w}^3} = \frac{1}{3}$, $p_{\bar{w}^4} = \frac{1}{3}$.
- 2: $p_{\bar{w}^1} = 0$, $p_{\bar{w}^2} = p_{\bar{w}^3} = 0.45$, $p_{\bar{w}^4} = 0.1$
- 3: $p_{\bar{w}^1} = 0$, $p_{\bar{w}^2} = p_{\bar{w}^3} = 0.5$, $p_{\bar{w}^4} = 0$
- 4: $p_{\bar{w}^1} = 0.03$, $p_{\bar{w}^2} = 0.91$, $p_{\bar{w}^3} = 0.03$, $p_{\bar{w}^4} = 0.03$

For example, Blue may know that Red is using an asymmetrical control, and hence set $p_{\bar{w}^1} = 0$ and $p_{\bar{w}^4} = 0$. With such modelling of Red control one can again expect Blue to do no worse than having no information about actual Red control, even while using W_I^n. As seen in Figure 3.5, for the present study example such information does not gain any significant advantage for Blue using the naïve approach and W_I^n. More specifically, such information was not able to change the Blue control strategy. When the Blue player computed its control with W^n, a change in Blue control gave Blue almost 20% improvement in the mean-sample payoff. The higher mean-sample payoff given by the 'o' curve corresponds to Blue knowing W_I^n. Note that this change happens with the knowledge that $p_{\bar{w}^2} = p_{\bar{w}^3} = 0.5$, thus implying that substantial intel on Red control and the correct control set W^n makes the standard approach work reasonably well for Blue in this example (even though the mean-sample payoff from 'MLS' still does not match the mean-sample payoff when Blue is using the 'DR' approach).

3.5.2 Pruning or Reduction in the Size of Q_t

We refer to the reduction in the size of the set of potential states as pruning. From Blue's perspective in the partially-observed case, this refers to reduction in the size of the set of feasible observation-conditioned distributions, Q_t. Pruning techniques

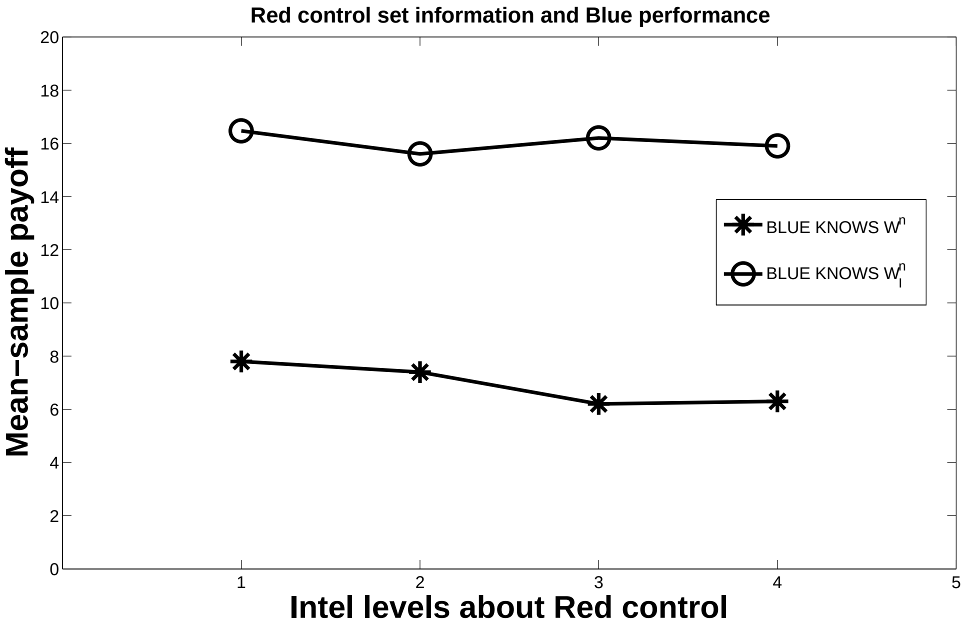


Figure 3.5: Naíve Blue and Intel

for Blue directly affect computational speeds. Choosing the initial distributions q_0^k (for modelling unknown Red initial state layout) and using some intel about the Red control is closely linked to the computational speed of pruning and Blue performance. This forms an important component of the study. As outlined in the theory, a set of initial distributions is carried forward by Blue using each Red control $\vec{w} \in W^n$, to obtain the set of feasible distributions at each time. Since, in the 'DR' approach, an argmax needs to be computed over such a set of feasible distributions, we need to prune these distributions sufficiently to allow computational feasibility and also stay within reasonable error tolerance at the same time. Growth of the number of these distributions is linear in the number of initial distributions and exponential in the dimension of the finite Red control set. This provides motivation to study the affects of pruning using various initial distributions (and in different numbers) and any effect of a more specific knowledge of Red control set on the performance of the Blue player or the pruning speeds. One may note that the pruning can be done after the observation update or after the dynamic update or one may even use pruning after each of these updates. If one uses some pruning method before the observation update, the most critical thing is to keep a track of the Red state-feedback control associated with a particular $q \in Q_t$. This is important since we assumed that the same Red control will be used for the observation and the dynamic update.

Note that if n^w is the size of W, and one has n^{iq} initial distributions, then the set of feasible distribution Q_t at time t has size $n^{iq} * (n^w)^t$. This growth is exponential and needs to be checked to allow practical implementation of the 'DR' approach. In practice only a few distributions will be useful. In computation involving argmax over $q \in Q_t$, a smaller size of Q_t will make faster computational speeds possible. We first outline the pruning algorithm that is used in the simulation results to follow. At time t one can obtain a set $\underline{Q}_t \subset Q_t$ as follows. Let's initialize $\bar{Q}_t \doteq Q_t$ and $\underline{Q}_t = \emptyset$. For some $q \in \bar{Q}_t$, let

$$Q_t^{\mu,q} = \{q^k \in \bar{Q}_t : \mathbb{L}_1(q^k, q) \leq \mu\}$$

where μ is the pruning error tolerance and $\mathbb{L}_1$ is the L1 norm. Note that $q \in Q^{\mu,q}$. We reset $\bar{Q}_t = \bar{Q}_t \setminus \{Q^{\mu,q}\}$ and $\underline{Q}_t = \underline{Q}_t \bigcup \{q\}$ and keep repeating the above operations till $(\#\bar{Q}_t) = 0$. The resulting $\underline{Q}_t$ gives us the pruned set of distributions. We refer to this pruning method as 'N-1' and if we use the L2 norm instead, we call it the 'N-2' pruning.

Another pruning technique can use the spikiness of the information in the distributions. Let

$$Q_t^{\Gamma c} = \{q \in Q_t, \text{ such that } \Gamma[q] \neq \Gamma[q^k], \forall\ q^k \in Q_t, q^k \neq q\}$$

Let's initialize $\bar{Q}_t \doteq Q_t \setminus Q_t^{\Gamma c}$ and $\underline{Q}_t = \emptyset$, then for some $q \in \bar{Q}_t$ such that $q \notin \underline{Q}_t$, let

$$\bar{Q}_t^{\mu,q} = \{q^k \in \bar{Q}_t : \Gamma[q] = \Gamma[q^k] \text{ and } 0 < ([q^k]_{\Gamma[q]} - [q]_{\Gamma[q]}) \leq \mu, q^k \neq q\}$$

Reset $\bar{Q}_t = \bar{Q}_t \setminus \{\bar{Q}^{\mu,q}\}$ and $\underline{Q}_t = \underline{Q}_t \bigcup \{q\}$. We stop repeating the above operation when there is no $q \in \bar{Q}_t$ such that $q \notin \underline{Q}_t$. Then the pruned set of distributions is given by $\bar{Q}_t \bigcup Q_t^{\Gamma c}$. This pruning technique will be called the 'N-3' pruning method.

We used N-1' and 'N-2' pruning techniques in our simulations because of their simple application and the fact that their was no significant difference in performance using any of the above, see Table 3.3. There was a slight advantage in speed owing to using 'N-2'; the set $\underline{Q}_t$ was on an average 20% smaller in size when using 'N-2' as compared to when using the 'N-1' technique.

Table 3.3: Comparing Blue performance using different pruning techniques

pf	Payoff with N-1	Payoff with N-2	Payoff with N-3
0.3	4.30	4.36	4.26
0.4	4.54	4.46	4.38
0.5	4.42	4.55	4.62
0.6	4.38	4.42	4.50
0.7	4.58	4.52	4.48

Table 3.4: Payoff based on error tolerance μ

μ	Payoff with $[q_{ij}^G]_0$	Payoff with q_0^U or q^{NU}
0.0001	4.30	4.76
0.001	4.54	5.26
0.05	4.42	6.38
0.1	4.38	7.44
0.2	4.36	7.92

In the following simulation results Red strategy RG and $MS2T5$ model are used. The affect of μ on computational speed is expected to be very simple, a higher μ would generally prune more distributions in comparison to a lower μ. Thus with higher μ simulation speeds are faster compared to lower values of μ due to slow growth of Q_t. However this intuitive relationship also depends on another factor, the actual distribution set also. Since Q_t is obtained from Q_0, the choice of initial distributions will also affect the pruning speed. When distributions of type $q_{i,j}^G$(at a different state) are used, less distributions are pruned and simulation speed slows down considerably. However the mean-sample payoff in this case is not very sensitive to the error tolerance, μ, as one can see in Table 3.4. This may allow for a potential higher error tolerance, μ, as admissible. Initial distributions that are more flat lead to faster pruning and hence better speeds. But with more flat distributions (q_0^U, $\bar{q}_0^U$, and q_0^{NU}) the mean-sample payoff is more sensitive to the μ. Consequently such distributions may need a tighter error tolerance, μ, to achieve the desired performance specifications. The results for this case are also shown in Table 3.4. Note that the error tolerance, μ, can be readjusted with time if some learning rate type of information is available to lower or increase the pruning level.

3.6 Mismodelling: Blue Deception-Robust Algorithm

In this section we again assume the $X_0^R = (1,3)$ and present simulation results for the case A_I (attrition level is not based on stealth).

Mismodelling p_2^N

We start with a simulation result where the attrition parameter $[p_2^N]_T \neq [p_2^N]_B$. For a constant α_1, this also gives $[p_1^N]_T \neq [p_1^N]_B$, $[p_2^S]_T \neq [p_2^S]_B$, and $[p_1^S]_T \neq [p_1^S]_B$. Let $[p_2^N]_B = k[p_2^N]_T$, where we allow $0.5 \leq k < 0.9$. The results in Figure 3.6 indicate that the mean-sample payoff doesn't change initially (indicating no change in Blue control) as k increases (or the mismatch between $[p_2^N]_T$ and $[p_2^N]_B$ is reduced). However, with k close to 0.9 for $[p_2^N]_T = 0.675$ (or 0.8 for $[p_2^N]_T = 0.875$), the mean-sample payoff drops significantly, indicating a change in Blue control. Note that this change happens for a relatively smaller k with a higher $[p_2^N]_T$. From the top plot ($pf = 0$) to the bottom plot ($pf \neq 0$), one can also see that with exclusion of decoys the mean-sample payoff is lower compared to the mean-sample payoff with inclusion of decoys (indicating the scope of deceptive Red controls and Blue performance in presence of higher levels of imperfection).

Mismodelling p^s

We set $pf_B = 0$ and $pf_T = 0$ to study the sole affect of probability of observing stealthy Red entities. Some other parameter values : $p^n = 0.8$ and $p_s = 0.5$ were chosen for the following results. Note that p_s is not used in the 'DR' approach. The 'RG' strategy of Red allows for very little (almost negligible) influence of p_s on 'MLS' performance. In absence of false alarms, and for a fixed p_s, the 'MLS' Blue control computation depends on observations (here affected by varying p_T^s and p_B^s, and a function of p_s). For the 'DR' approach control computation depends on observations (here affected by varying p_T^s and p_B^s, not a function of p_s). In Figure 3.7 it is clear that the 'DR' approach is robust to mismodelling of p^s (or almost independent of p_T^s for a fixed p_B^s), which is a very good result. The 'MLS' approach however fairs poorly as p_T^s decreases for a fixed p_B^s or stealthy Red entities become harder to detect, as seen in Figure 3.8. This essentially

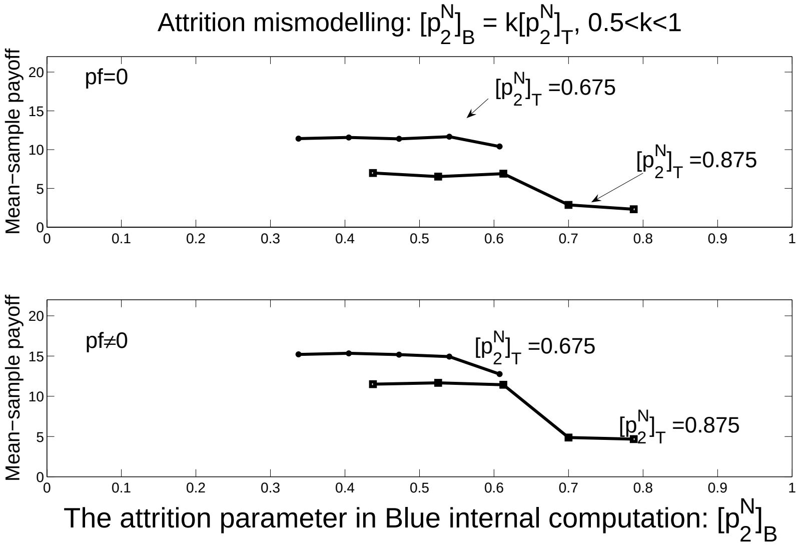


Figure 3.6: Attrition mismodelling: mismatch of p_2^N, $[p_2^N]_T \neq [p_2^N]_B$

translates into 'MLS' becoming more susceptible to deception.

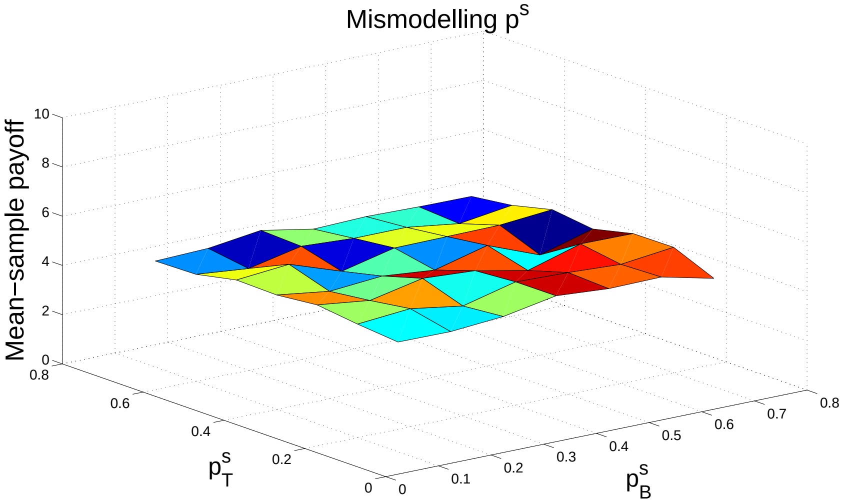


Figure 3.7: p^s mismodelling, 'DR'

For a fixed $p^s{}_T$ as p^s_B is varied, not much change is noticeable in the mean-sample

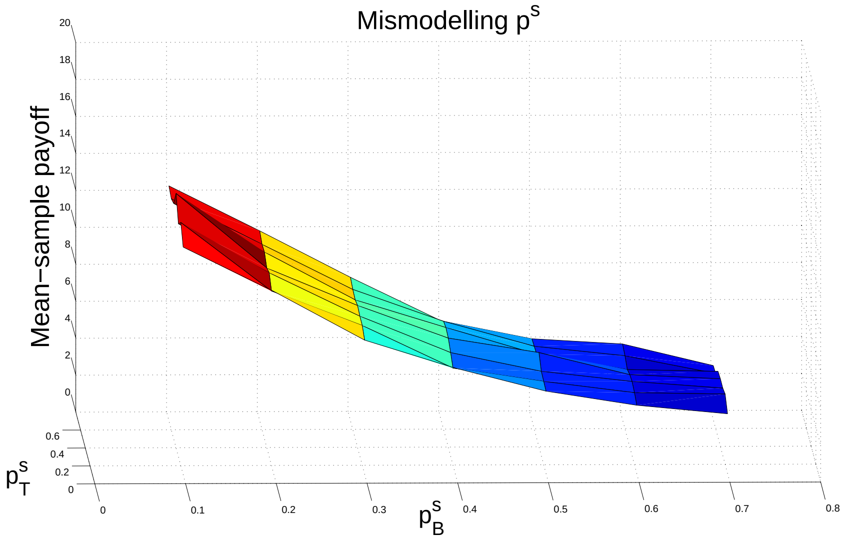


Figure 3.8: p^s mismodelling, 'MLS'

payoff. Note here that the observation process is exogenous and independent of Blue state or control. One potential argument is that with decoy and the skewed asymmetric Red layout the maximum likelihood state is seen to occur at states corresponding to (a, b), where $b > a$. Using 'MLS' approach Blue is committed to the eastern side at the first time step and with successive skewed observations leaning towards east (with a flashing decoy on the eastern route and the western route Red entities staying stealthy), Blue does not gain anything by modelling the p^s_T correctly for this scenario. The frequency of observing the stealthy Red on the western route is smaller than the frequency of observing the decoy even when Blue models p^s_T perfectly (i.e. $p^s_B = p^s_T$). This argument is clearly not generalizable as one would expect better performance with perfect modelling or a lower mean-sample payoff.

Mismodelling pf

With the notation defined earlier we discuss results for both cases, $[pf_1]$ and $[pf_2]$. Also note that we set $p^n = p^s$ and $p^n_T = p^n_B$ to avoid coupling affects of mismodelling due to the affect of stealth. Besides noticing the mismodelling trends one may note that in all the results so far 'DR' approach has been better for Blue than 'MLS' (lower expected payoff, Blue being the minimizer). In $[pf_1]$, Red can only use the stealth factor as additional deception (which we have not allowed by setting $p^n = p^s$) then one may note that 'MLS' Blue approach does better in this parameter regime compared to completely imperfect case where its mean-sample payoff (though not very sensitive to pf_T) is way higher than the mean-sample payoff corresponding to the 'DR' approach. 'MLS' approach gives even higher mean-sample payoff in the second case (very imperfect scenario) compared to their corresponding values for the $[pf_1]$ case. In Figures 3.9 and 3.10, one can see that using 'MLS' approach (as pf increases) Blue performance gets affected negatively (mean-sample payoff increases) whereas the 'DR' approach is still robust to this variation.

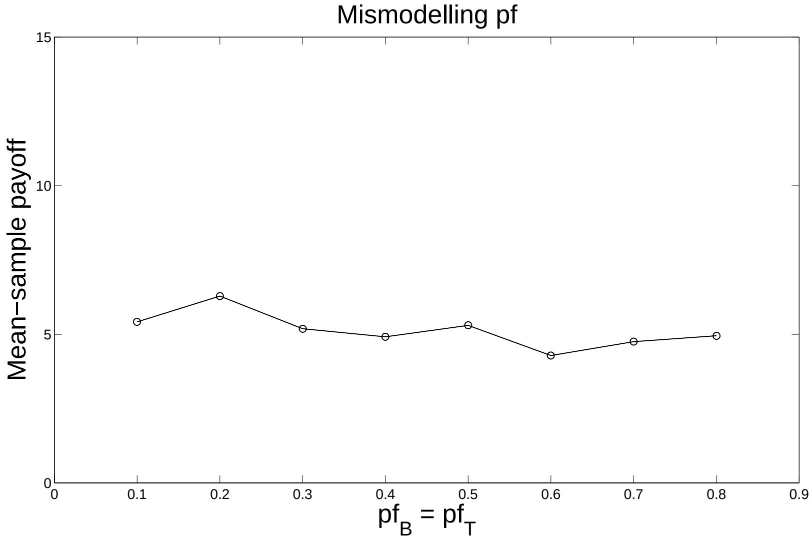


Figure 3.9: Mismodelling pf, $[pf_1]$, 'DR'

Figures 3.11 and 3.12 show that using 'MLS' or 'DR' approach (as pf increases) Blue performance doesn't get affected negatively. However, from $[pf_1]$ to $[pf_2]$ their is a substantial jump in the mean-sample payoff using 'MLS' whereas the 'DR' approach is very robust.

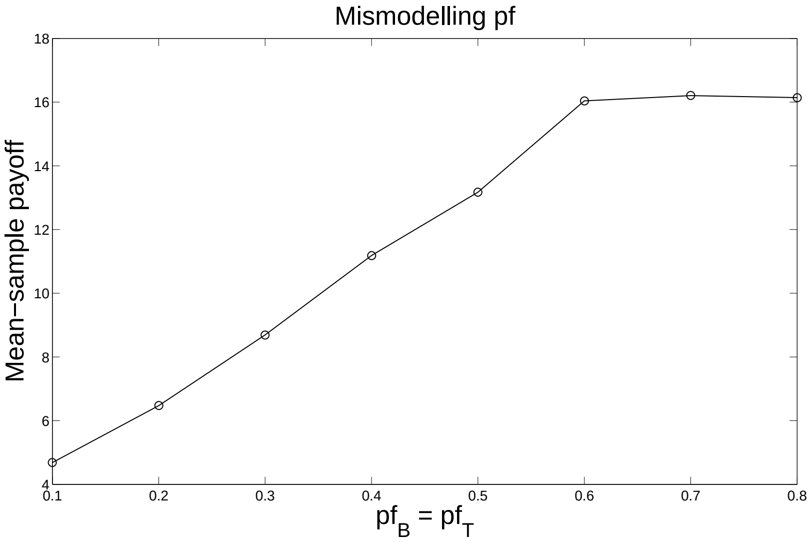


Figure 3.10: Mismodelling pf, $[pf_1]$ 'MLS'

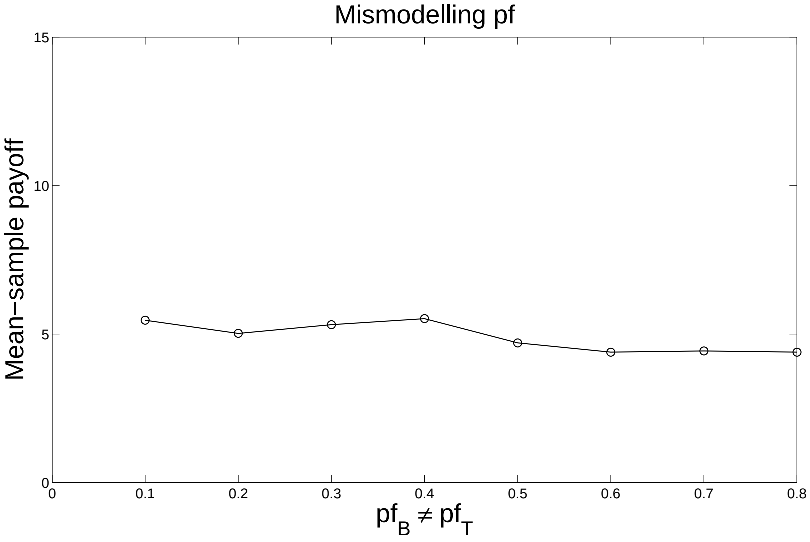


Figure 3.11: Mismodelling pf, $[pf_2]$, 'DR'

Variation of p_s

Recall that there is this second inherent mismodelling in 'MLS' approach as Red control is using the 'RG' strategy and Blue uses p_s to model Red control/strategy

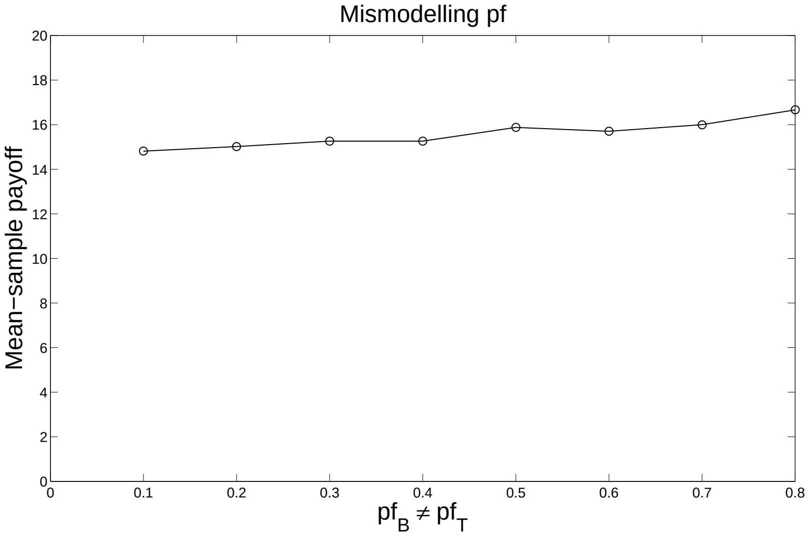


Figure 3.12: Mismodelling pf, $[pf_2]$, 'MLS'

Table 3.5: Mismodelling p_s, with $(pf_B \neq pf_T)$

pf	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Mean-sample payoff	16.24	15.82	16.14	15.96	15.94	16.08	15.74

(Blue doesn't know $X_.$ and $w_.$). We discuss two cases here also. First we set $pf = 0$ in this simulation to find p_s mismodelling affects in absence of false alarms and in the second case we set $pf_B \neq pf_T$ to find affect of mismodelling p_s in presence of false alarms with the $[pf_2]$ mismodelling. Results for the 'MLS' approach with variation in p_s are shown in Table 3.5. One can note that p_s variation doesn't change the mean-sample payoff substantially in either case. In fact this result is again supported by the argument of the previous section for this problem but may not hold true generally for all problems.

3.7 Appendices

3.7.1 Proof of Theorem 3.3.1

Proof. Note that the proof is similar to the proof in (McEneaney 2004), if we note that:

1. $S_t^q = \bigcup_{\vec{w} \in W_t^q} S_t^{\vec{w},q}$

2. $S_t^q \neq \emptyset \iff W_t^q \neq \emptyset$

The first item in the above list is by clear definition of the terms. Note that $S_t^q \neq \emptyset \iff \{S_t^{\vec{w},q} \neq \emptyset$ for some $\vec{w} \in W_t^q\}$. Which by definition of W_t^q gives $S_t^q \neq \emptyset \iff W_t^q \neq \emptyset$. We now provide the proof for the information state propagation given by the form (3.25). We first prove that $\mathcal{I}_{t+1}(q) \geq \max_{\hat{q} \in S_t^q} \mathcal{I}_t(\hat{q})$. The case $S_t^q = \emptyset$ is trivial. So we assume $S_t^q \neq \emptyset$, which implies $\exists\ \tilde{q}_t \in Q_t$ such that

$$q = G(y_t, u_r, \vec{\tilde{w}})[\tilde{q}_t] \tag{3.80}$$

for some $\vec{\tilde{w}} \in W^n$. Now using finiteness of W and $\tilde{q}_t \in Q_t$, $\exists \vec{\tilde{w}}_{[0,t)}$ and by the definition of Q_t, $\exists \hat{q}_0$ such that

$$\mathcal{I}_t(\tilde{q}_t) = \mathcal{I}_0(\hat{q}_0) \tag{3.81}$$

where

$$\tilde{q}_t = [\prod_{r=0}^{t-1} G(y_r, u_r, \vec{\tilde{w}}_r)][\hat{q}_0]$$

If one defines

$$\vec{w}_r = \begin{cases} \vec{\tilde{w}}_r & \text{if } r \leq t-1; \\ \vec{\tilde{w}}_r & \text{if } r = t. \end{cases} \tag{3.82}$$

then $\vec{w}_{[0,t]} \in [W^n]^t$. and

$$q = [\prod_{r=0}^{t} G(y_r, u_r, \vec{w}_r)][\hat{q}_0],$$

so that $q \in Q_{t+1}$. Then, using definition of $\mathcal{I}$, one gets

$$\mathcal{I}_{t+1}(q) \geq \mathcal{I}_0(\hat{q}_0). \tag{3.83}$$

Combining (3.81) and (3.83) gives

$$\mathcal{I}_{t+1}(q) \geq \max_{\hat{q} \in S_t^q} \mathcal{I}_t(\hat{q}). \tag{3.84}$$

Now we prove the reverse direction $\mathcal{I}_{t+1}(q) \leq \max_{\hat{q} \in S_t^q} \mathcal{I}_t(\hat{q})$. The case when $S_t^q = \emptyset$ is again trivial, so consider the case $\mathcal{I}_{t+1}(q) \neq -\infty$; otherwise there is nothing to prove. By finiteness of W there exists an optimal $\vec{\tilde{w}}$ and corresponding $q_0 \in Q(\mathcal{X})$ given by

$$q = [\prod_{r=0}^{t} G(y_r, u_r, \vec{\tilde{w}}_r)][q_0]$$

such that

$$\mathcal{I}_0(q_0) = \mathcal{I}_{t+1}(q). \tag{3.85}$$

Then

$$q_t = [\prod_{r=0}^{t-1} G(y_r, u_r, \vec{\tilde{w}}_r)][q_0] \in S_t^q.$$

Note that $q_t \in S_t^q \Rightarrow q_t \in Q_t$ and using definition of $\mathcal{I}_t$,

$$\mathcal{I}_t(q_t) \geq \mathcal{I}_0(q_0). \tag{3.86}$$

Combining (3.85) and (3.86) yields

$$\mathcal{I}_{t+1}(q) \leq \mathcal{I}_t(q_t) \leq \max_{\hat{q} \in S_t^q} \mathcal{I}_t(\hat{q}). \tag{3.87}$$

□

3.7.2 Proof of Lemma 3.3.8

Proof. It is sufficient to show that the left and the right hand sides of (3.54) are equivalent to the left and the right hand sides of (3.55). We first prove the equivalence of the right hand sides. Note that using the facts that $Q_t \neq \emptyset$ and if $q \notin Q_t$ then $\mathcal{I}(q) = -\infty$, the outer maximization can be rewritten as

$$= \max_{q \in Q_t} \{\mathcal{I}_t(q) + \mathbf{E}_q \max_{\vec{w} \in W^n} [\sum_{j \in \mathcal{X}} \widetilde{P}_{xj}(u_t^m, \vec{w}) V_{t+1}(j, q'(q, u_t^m, \vec{w}))]\}$$

Using the definition of $\mathcal{I}(.)$, one gets

$$\max_{q \in Q_t} \{ \max_{q_0 \in Q_0^{q, u_{[0,t)}}} \mathcal{I}_0(q_0) + \mathbf{E}_q \max_{\vec{w} \in W^n} [\sum_{j \in \mathcal{X}} \widetilde{P}_{xj}(u_t^m, \vec{w}) V_{t+1}(j, q'(q, u_t^m, \vec{w}))]\}.$$

Since the inside expectation over the future cost doesn't depend on q_0, one can move the inner $\max_{q_0 \in Q_0^{q, u_{[0,t)}}}$ as follows:

$$= \max_{q \in Q_t} \max_{q_0 \in Q_0^{q, u_{[0,t)}}} \{\mathcal{I}_0(q_0) + \mathbf{E}_q \max_{\vec{w} \in W^n} [\sum_{j \in \mathcal{X}} \widetilde{P}_{xj}(u_t^m, \vec{w}) V_{t+1}(j, q'(q, u_t^m, \vec{w}))]\}.$$

Note that for each $\tilde{q}_0 \in Q_0$ and some $\vec{w}_{[0,t)} \in [W^n]^t$, $\exists\, q = q_t^{'}$ $(\in Q(\mathcal{X}))$ such that $\tilde{q}_0 \rightarrow q_t^{'}$ using $\vec{w}_{[0,t)}$ and hence $\tilde{q}_0 \in Q_0^{q, u_{[0,t)}}$.

Conversely, for any $q \in Q_t$ and a corresponding $q_0 \in Q^{q,u_{[0,t)}}$, $q_0 \in Q_0$ and $q_0 \to q$ using some $\vec{w}_{[0,t)} \in [W^n]^t$. which then gives

$$= \max_{q_0 \in Q(\mathcal{X})} \max_{\vec{w}_{[0,t)} \in [W^n]^t} \{\mathcal{I}_0(q_0) + \mathbf{E}_{q_t^{'}} \max_{\vec{w} \in W^n} [\sum_{j \in \mathcal{X}} \widetilde{P}_{xj}(u_t^m, \vec{w}) V_{t+1}(j, q'(q_t^{'}, u_t^m, \vec{w}))]\}.$$

where $q_t^{'}$ is given by propagation (3.24) with initial q_0, controls $u_{[0,t)}$ and $\vec{w}_{[0,t)}$, and observations $y_{[0,t)}$, and noting that $\mathcal{I}_t$ is deterministic (given $y_{[0,t)}$)

$$= \max_{q_0 \in Q(\mathcal{X})} \max_{\vec{w}_{[0,t)} \in [W^n]^t} \mathbf{E}_{q_t^{'}} \{\mathcal{I}_0(q_0) + \max_{\vec{w} \in W^n} [\sum_{j \in \mathcal{X}} \widetilde{P}_{xj}(u_t^m, \vec{w}) V_{t+1}(j, q'(q_t^{'}, u_t^m, \vec{w}))]\}.$$

and since W^n consists of state-feedback controls

$$= \max_{q_0 \in Q(\mathcal{X})} \max_{\vec{w}_{[0,t]} \in [W^n]^{t+1}} \mathbf{E}_{q_t^{'}} \{\mathcal{I}_0(q_0) + [\sum_{j \in \mathcal{X}} \widetilde{P}_{xj}(u_t^m, \vec{w}) V_{t+1}(j, q'(q_t^{'}, u_t^m, \vec{w}))]\}.$$

which is the desired equivalence for the right sides.

For the left side equivalence, note that by definition of $\mathcal{I}_t(.)$, $\exists\ q_0^1 \in Q(\mathcal{X})$ and $\vec{w}_{[0,t)} \in [W^n]^t$ such that $q_0^1 \to q_t^{''}$ using $\vec{w}_{[0,t)}$ ($q_t^1 = q_t^{''} \in Q_t$) such that

$$\mathcal{I}_t(q_t^{''}) = \mathcal{I}_0(q_0^1).$$

Then utilizing that $\mathcal{I}(.)$ is a deterministic function of given $y_{[0,t)}$, and controls $u_{[0,t)}$ and $\vec{w}_{[0,t)}$ one gets the required result

$$\begin{aligned} & \mathbf{E}_{q_t^{''}} \{\mathcal{I}_0(q_0^1) + [\sum_{j \in \mathcal{X}} \widetilde{P}_{Xj}(\tilde{u}_t, \vec{w}_t^1) V_{t+1}(j, q^{'}(q_t^{''}, \tilde{u}_t, \vec{w}_t^1))]\} \\ = \quad & \{\mathcal{I}_t(q^1) + \mathbf{E}_{q^1}[\sum_{j \in \mathcal{X}} \widetilde{P}_{xj}(\tilde{u}_t, \vec{w^1}) V_{t+1}(j, q^{'}(q^1, \tilde{u}_t, \vec{w^1}))]\} \end{aligned} \tag{3.88}$$

□

This chapter is in part a reprint of the materials as is appears in,

Rajdeep Singh, William M. McEneaney - *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, CRC press, To appear.

Rajdeep Singh, William M. McEneaney - *Unmanned vehicle decision making under imperfect information in an adversarial environment.* AIAA Journal of Guidance Navigation and Control, in preparation.

Rajdeep Singh, William M. McEneaney - *Unmanned Vehicle Operations under Imperfect Information in an Adversarial Environment*, AIAA Guidance, Navigation, and Control Conference and Exhibit, 2004.

Rajdeep Singh, William M. McEneaney - *Unmanned Vehicle Operations: Countering Imperfect Information in an Adversarial Environment*, AIAA 3rd ”Unmanned Unlimited” Technical Conference, Workshop and Exhibit AIAA, 2004.

The dissertation author was the primary author and the co-author listed in these publications directed and supervised the research.

Chapter 4

Red Approach in the Partially-Observed Game

In this chapter we will develop some automated control algorithm for the Red player which will employ deceptive controls, if useful. This control will be deception-enabled control because it employs deception when it is profitable to do so. In this set up as before, Red will be maximizing and Blue will be minimizing the same cost criterion. Recall that the Blue player has no access to the actual state process, $X_{\cdot}$, but only knows the $q_{\cdot}$ process. Again, the Blue player has no access to the Red control history as well. In the Deception-Robust approach, we allowed the Red player to have access to the $q_{\cdot}$ process (which Red could construct if it had access to the observation process $y_{\cdot}$). This led to the appended state $(X_{\cdot}, q_{\cdot})$ for the Red player. This construction led to the worst case scenario for Blue, giving the deception-robust theoretical formulation. We could restrict the information set of the Red player to only state process $X_{\cdot}$, and that of Blue to only the observation process $y_{\cdot}$, which will lead us to information patterns that are not nested, since neither players information set subsumes the others. Instead, from the Red player's viewpoint we set up best case scenario, where Red has complete knowledge of not only the observation process $y_{\cdot}$, but also access to Blue's initial state estimate q_0, and the approach Blue is using to compute its control $u_{\cdot}$. As before the Red decision will still be based on the appended (X_t, q_t).

4.1 Red Algorithm Approach Using an Internal Blue Control Model

We reiterate the strong modelling assumption discussed above:

The Red player knows $(Q_0, \{y_r\}_{r=0}^{t})$and the Blue control algorithm. (A-RI)

Instead of solving the problem where the information sets are not nested, we explore the case where Red has maximum information (as given by (A-RI)) and find out if deception is feasible for our example problem. Clearly, if deception is not useful under these relaxed conditions, then one doesn't expect deception to be useful in the scenario where the information patterns are not nested (since Red has neither any information on the observation process nor any knowledge of the Blue control approach). Though the results for a particular example cannot be generalized they will shed some preliminary light into the construction and utility of automated deception control algorithm.

So we now solve the problem using the assumption (A-RI). Recall that the observation process is dependent on the actual Red control; the observation process takes place only after Red chooses a control. And given a q_t, the Blue control computation happens after the observation happens. Thus, under assumption (A-RI), and for a given q_t, the Red player needs to generate a virtual Blue control $u_t^{v,y}$ at each time t, as a function of $y \in Y$. Note that this is the internal control computations that Red does by mimicking the Blue algorithm and we assume that the parameters of the actual Blue algorithm are modelled perfectly by Red's internal Blue algorithm. Once Red uses this virtual control in deciding its optimal control (the algorithm for this will be defined soon), the actual observation y_t happens, and the actual Blue control $u_t^{y_t}$ is computed by the Blue player based on that observation and distribution q_t. If there is no mismatch of parameters used in the actual Blue algorithm and the internal mimicking of this algorithm in the Red virtual Blue computation, then one has $u_t^{y_t} = u_t^{v,y_t}$. Clearly this should form the most ideal information set for Red to be able to exercise deception, if possible. Note also that if there is some parameter mismatch between the actual Blue algorithm and Red's internal Blue algorithm, then one might expect $u_t^{y_t} \neq u_t^{v,y_t}$. Some simulation results will be presented to study mismodelling, specifically the mismatching of the algorithms (without any parameter mismodelling).

Recall that Red is the maximizer and in the domain with assumption (A-RI), one would expect Red to achieve no lower payoff than when it has no information on the actual observations, y_t, or Blue algorithm (or mismodels the internal parameters of the Blue algorithm) or a combination of all these factors. Since Red can generate the virtual Blue control, Red can now solve an optimal control problem, where Red is trying to maximize the cost function. There are reasonable possibilities for Blue algorithm modelling (on the part of Red); from naïve to a little more generic. Note that Red computes the virtual Blue control as a function of $\hat{q}_r$, $\forall\ r \leq \bar{\mathbf{T}}$ (obtained using $\{y_s\}_{s=0}^{r-1}$ and q_0, and some stochastic Red control model, $p^B_{\vec{w}}$). Note that the Blue (internal or actual) computation, has access to the current observation y_t in its decision process. Given initial q_0, the Red player can internally propagate $q_.$ by using an estimation based controller and stochastic Red control modelling as in section 3.2. We will denote this virtual control as $u^{v,s}_r$, where 's' stands for stochastic model of Red control. In which case $u^{v,s}_.$ may be computed using (3.12), (3.9), or (3.14). Similarly, if Red computes the virtual Blue control using the 'DR' approach (as in section 3.3), Red will compute the virtual Blue control, say $u^{v,d}_r$, as a function of Q_r (or $\hat{Q}_r$, in particular, the set of pruned posteriori distributions), $\forall\ r \leq \bar{\mathbf{T}}$ (obtained using $\{y_s\}_{s=0}^{r-1}$ and projecting each $q_0 \in Q_0$ along each possible Red control trajectory, $\{w_s\}_{s=0}^{r-1} \in W^r$). In Red's internal Blue control computation, some initial Q_0 and $\mathcal{I}_0$ are propagated for each possible observation process $y_.$, to Q_t, $\mathcal{I}_t(q)$ using the robust control given by (3.53), $u^{v,d}_. = u^m_.$.

Note that the Red player is mimicking the actual Blue algorithm and all these computations are done internally by Red in exactly the same manner as the actual Blue player would compute its control u_t, (based on the random observation at time t). If assumption (A-RI) holds, then based on its knowledge of the actual Blue algorithm Red internally computes the virtual Blue control, and propagates the state $q_.$ (internally) with :

$$\begin{cases} u^{v,s}_t \text{ and } p^B_{\vec{w}} \text{ using (3.4) and (3.5)} & \text{if Blue uses a stochastic Red control model} \\ u^{v,d}_t \text{ and } \overline{\theta}_.(X_., q_.) \ \text{ using (4.2)} & \text{if Blue uses the 'DR' approach} \end{cases} \tag{4.1}$$

where

$$\widehat{q}_t = \left(\tfrac{1}{\widetilde{R}'(\overline{y}, u_t, \vec{w}) q_t}\right) D(\overline{y}, u_t, \vec{w}) q_t \tag{4.2}$$

with appropriate definitions of D and $\tilde{R}$ from section 3.3 and $y_t = \bar{y}$.

We will refer to the virtual Blue control that Red computes internally as $u^{*,v}$, where

$$u_t^{*,v} = \begin{cases} u_t^{v,s} & \text{if Red's internal Blue computation uses a stochastic Red model} \\ u_t^{v,d} & \text{if Red's internal Blue computation is driven by 'DR' approach} \end{cases} \tag{4.3}$$

We now define the strategy set for Red in terms of the appended state $(X_., q_.)$ as follows:

$$\overline{\Theta}_{[t,T)} = \left\{ \overline{\theta}_{[t,T)} : \mathcal{X}^{(T-t)} \times Q^{(T-t)} \rightarrow [W^n]^{(T-t)}, \text{n.a} \right\}. \tag{4.4}$$

where $X_.$ propagates as a Markov chain with probabilities given by (2.1) (with $\vec{w}_. = \overline{\theta}_.(X_., q_.)$ and the internal Blue control given by (4.3). The $q_.$ process is propagated by (4.1).

Recall that there is no running cost for this problem. The terminal cost (or payoff) is $\mathcal{E} : \mathcal{X} \rightarrow \mathbf{R}$; the cost of terminal state (X_T, q_T) is $\mathcal{E}(X_T)$. Given any initial state $(x, \tilde{q}) \in \mathcal{X} \times Q(\mathcal{X})$, Red will be maximizing the following cost function for this optimal problem:

$$W_t(x, \tilde{q}) = \max_{\overline{\theta}_{[t,T)}} \mathbf{E}[W_T(X_T, q_T) |\ X_t = x, q_t = \tilde{q}] \tag{4.5}$$

where the cost function at the terminal time is: $W_T(X_T, q_T) = \mathcal{E}(X_T)$ and where $q_.$ is propagated using (4.1) and the $X_.$ process is propagated using (2.1) with same controls as used for propagating the $q_.$ process.

Theorem 4.1.1. *Given (4.5), (4.4) and $r : t \leq r < T$*

$$W_t(x, \tilde{q}) = \max_{\overline{\theta}_{[t,r)}} \mathbf{E}[W_r(X_r, q_r) | X_t = x, q_t = \tilde{q}]. \tag{4.6}$$

where $X_.$ propagates by (2.1) (with $\vec{w}_. = \overline{\theta}_.(X_., q_.)$ and internal Blue control given by (4.3)) and the $q_.$ process is propagated by (4.1) .

Proof. See Appendix 4.4.1. □

Since the result is true for any r, in particular it is true for $r = t+1$. Note that in this special case (for a fixed $x \in \mathcal{X}$ and $\tilde{q} \in \ \bar{Q}_t$), the definition of the strategy set implies that the max over $\overline{\theta}_{\bar{t}} \in \overline{\Theta}_{\bar{t}}$ is equivalent to the max over $\vec{w} \in W^n$, which gives the one step backwards recursion

$$V_t(x, \tilde{q}) = \max_{\vec{w} \in W^n} \mathbf{E}[V_{t+1}(X, q) | X_t = x, q_t = \tilde{q}] \tag{4.7}$$

Finally we get the Red optimal controller from (4.7),

$$\vec{w}^* \in \operatorname*{argmax}_{\vec{w} \in W^n} \mathbf{E}[V_{t+1}(X, q) | X_t = x, q_t = \tilde{q}]. \tag{4.8}$$

We will use this control computation to determine if the Red player is able to take advantage using (A-RI) and employ deception, whenever it is useful.

4.2 Red Approach Using an Internal Blue Model: MAG Revisited

We now return to our example problem and illustrate that the Red player does indeed use deceptive control by using (4.8). As before, we will only allow the Blue player to know the set W_I^n (as defined in (3.71)). Also note that since the Blue player only knows W_I^n, the Red player assumes same information for its internal virtual Blue control computation. We allow Red to choose its control from the re-defined set

$$W^n = \{\vec{w} : \vec{w}_i = w^*, \forall \ i \in \mathcal{X}, \text{ for some } w^* \in \widetilde{W} \bigcup W \bigcup \underline{W}\} \tag{4.9}$$

where $\widetilde{W}$ is defined in section 3.3, W is defined in section 2.3, and $\underline{W}$ is the set of Red controls $\underline{w}^k$, such that $\underline{w}^k$ is an extension of $\bar{w}^k \in W$, with a decoy added on the western route. In this case, as an extension of $\bar{w}^1$ (using stealth on both routes), $\underline{w}^1$ will correspond to using stealth on both routes and a non-stealthy decoy added to the western route. The Red player may choose controls that do not use a decoy. If it does choose to use a decoy it is allowed to use decoys only on one route (note that $X_0 = (1, 3)$ is our fixed initial Red state). Clearly, with the new definition of W^n given in (4.9) above, the Red player has choices ranging from the state-feedback optimal control, $\bar{w}^1 \in W$, to potentially deceptive ones, $\tilde{w}^i \in \widetilde{W} \bigcup \underline{W}$. Recall that the 'RG' strategy uses, $w^o = \tilde{w}^2 (\in \widetilde{W})$, a decoy on the eastern route with three non-stealthy Red entities, and moving the Red entity on the western route stealthily. To obtain the result that the deceptive control outperforms the state-feedback control, one can simply run the simulation by fixing, say 'RG' as a potential deceptive Red strategy and compare it with the simulation runs by fixing the Red state-feedback optimal control as a strategy. Note that we now allow the Red player to choose a time dependent control given by (4.8) (unlike the 'DR' case where the Red strategy was not time-dependent). Figure 4.1 shows some results using assumption (A-RI), where the Blue player is using the 'MLS' approach, which is also the Red internal Blue algorithm (where 'MLS' approach is defined in section 3.2). Recall that this implies, $u_t^{y_t,v} = u_t^{y_t}$. The optimal Red control (for time $t = 1$) is $\tilde{w}_2$, and the subsequent controls are dependent on the random outcome of the observations and the dynamic process. We first discuss the results given in Figure 4.1 and then comment on the optimal Red control for time $t > 1$.

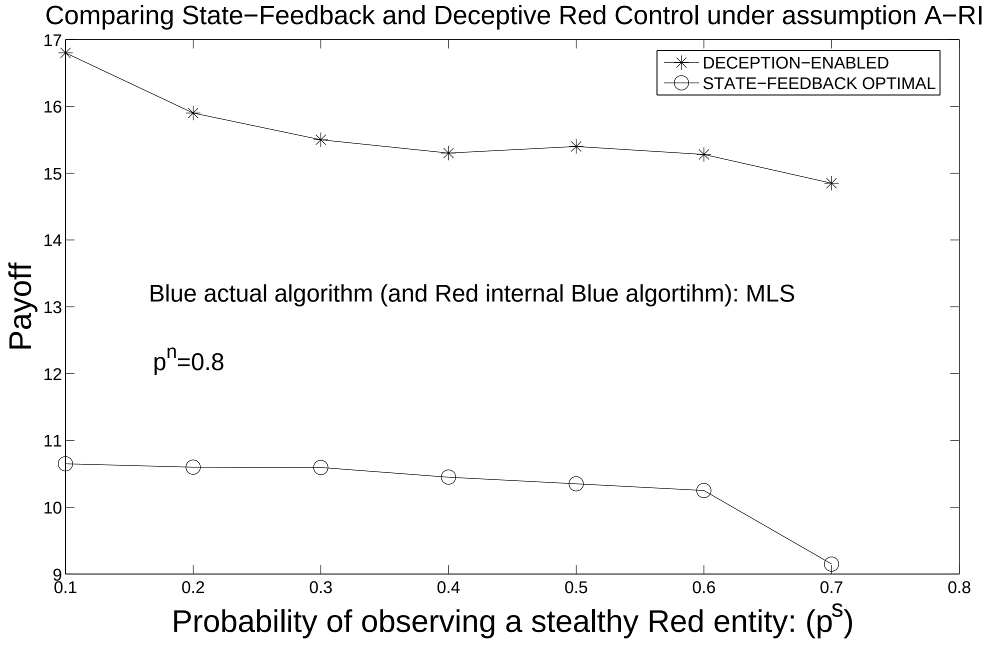


Figure 4.1: Red using 'DE' control vs. Red using state-feedback optimal control

It is clear from the data that the deceptive approach works better for the Red player than using the state-feedback optimal control ($\bar{w}^1$ or turning both sides stealthy). There is up to 60% improvement in the mean-sample payoff by using the deceptive control given by (4.8). As the probability to observe a stealthy object (p2) increases, the mean-sample payoff decreases for both choices of Red control indicating the role of stealth in exercising deception. These results presented in Figure 4.1 are with 2000 monte-carlo simulation runs; a larger number of iterations would capture this trend even more nicely. However it is clear that the deceptive controller for Red outperforms the state-feedback optimal Red controller. It is worth noting that the state-feedback optimal Red control is not optimal here, even when the Red player has perfect state knowledge. We conclude that the state-feedback optimal control is sub-optimal for the MAG example, under partial information set up, when the Red player is using assumption (A-RI).

The optimal Red control for time $t > 1$ gives a very unusual but seemingly intelligent Red strategy. After the first step dynamic interaction, if the random outcome of the interaction between the two UCAVs and Red entities on the eastern route leads to very low Red attrition (say 0 or 1 Red team is destroyed on the eastern route), then the Red player uses a control $\underline{w}^2 \in \underline{W}$. It uses a decoy on the western route and turns the single Red entity non-stealthy on that route. Also, all the remaining Red entities on the eastern route are now turned stealthy. If the observation at the next time leads the Blue player to assign a sufficiently high mass on state $(2, 0)$ or $(1, 0)$, the Blue player optimal 'MLS' control is to start moving both the UCAVs to the western route, i.e. $u_2 = 2$. Then as Blue moves towards the western route, the Red player is seen to exercise two control possibilities (depending on q_3). It can continue playing the same control as at time $t = 2$, $w_3 = \underline{w}^2$, and Blue sends both UCAVs to the western zone (in which case the UCAVs cannot return to intercept the remaining Red entities on the eastern route), leading to a payoff $20 + r_T^2$, where $r_T^2 = r_2^2$ (since no attrition happened after time step 2 on the eastern route). In the second possibility, Red can also choose to reverse the control at time $t = 3$ to $w_3 = \tilde{w}^2$ (again depending on q_3). If the random observation at $t = 3$ is still favorable to Red (leading Blue to again reassign the mass to a maximum likelihood state corresponding to most or all remaining Red entities on the eastern route), then the Blue player will choose to either send a single UCAV to each route or return both the UCAVs to the eastern route. In particular, it is often seen that the automated controller attempts to cause Blue to vacillate, wasting time in transit between the two routes for the above application problem. This sometime leads the Blue player to spend some time without intercepting the Red entities on either route or/and eventually neglect one side completely. This complex behavior demonstrates the Red intent to deceive Blue and gives a higher mean-sample payoff than the mean-sample payoff when using the state-feedback Red optimal control w_2.

This problem formulation has the obvious disadvantage of being susceptible to mismodelling by Red of the Blue control algorithm. The space of opponent controllers may be huge (even for our small scale problem, with appropriately redefining W) and so one would hope to avoid reliance on models of opponent controllers. We now present some mismodelling results where the assumption (A-RI) does not hold because the Red player has imperfect knowledge of the actual Blue algorithm (though it still has access to the correct observation process). In particular, the Red player is using an internal Blue approach different from the approach that the Blue player actually employs. The Red player may be using a smarter internal Blue algorithm or a much simpler/naïve internal Blue computation compared to the actual Blue algorithm. Clearly in this case $u_t^{y_t,v} = u_t^{y_t}$ may no longer be true. The Red internal Blue approach can be either the 'HB' or the 'MLS' approach defined in section 3.2, whereas we also allow the Blue player to use the 'DR' approach defined in section 3.3. Red may choose not to use any internal Blue computations and simply use the state-feedback optimal control. We will use R_a to mean that Red is using the control approach 'a', where 'a' could be the 'HB', 'MLS' or the 'State-feedback optimal' control approach. Recall, that in the A_I case (section 2.3), the state-feedback optimal Red control is to randomly chose some $\bar{w}^k \in W$. Clearly in partial information one may expect the control $\bar{w}^1 = (S, S)$ to be more useful for Red (which was the optimal control for the A_D case in the state-feedback set up). Simulation results show a small advantage for the Red player when using $\bar{w}^1 = (S, S)$ than when randomly using some $\bar{w}^k \in W$, so we use $\bar{w}^1$ is the state-feedback optimal control in the following results.

We first discuss the case where the Blue player chooses the 'MLS' approach. In Figure 4.2, note that even when (A-RI) holds or when Red is using the correct Blue approach ('MLS') in its computations, the Red player does not achieve a higher payoff than when using the incorrect Blue approach ('HB'). As pf increases, the payoff using the state-feedback optimal control stays the lowest but it's not too low compared to the payoff using Red control given by (4.8). Two more things can be noticed from the results in this figure. Firstly, as pf increases the Red player achieves a higher mean-sample payoff. Secondly, the state-feedback optimal Red control is almost as good as the controls given by (4.8) for high values of pf. Clearly in the parameter regime to the left of the plot (or lower pf values), the decoy addition is a valuable deceptive control for Red than using the state-feedback optimal control. With higher values of pf, the effect of pf variation does not yield as much advantage with the deceptive control given by (4.8) as with the state feedback optimal Red. This happens because the Blue player assumes W_I^n to be the Red control set. When Red uses the state-feedback optimal control and pf increases, this causes a worse mismodelling scenario for Blue than when Red is using a decoy on at least one side.

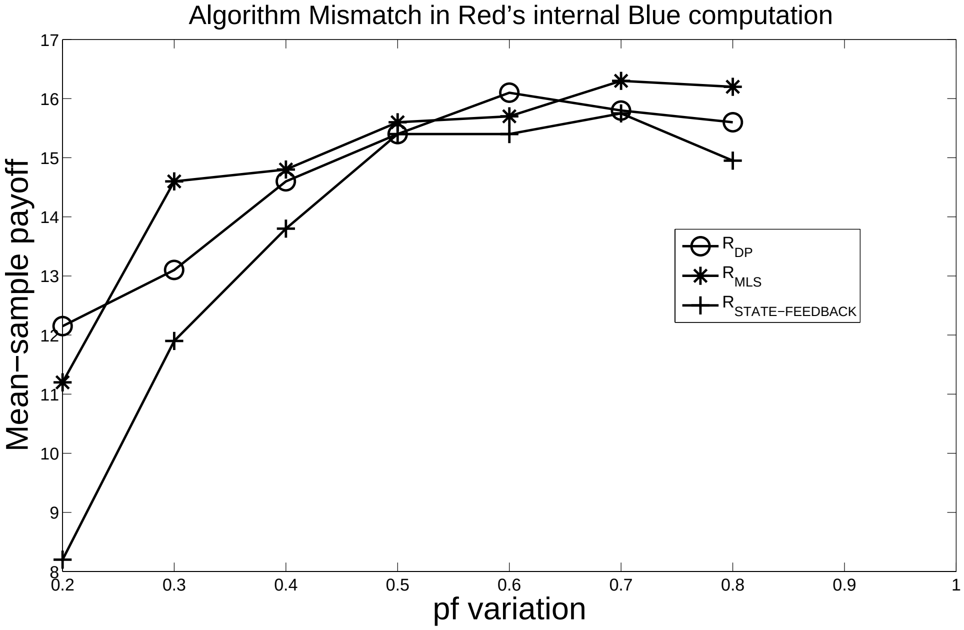


Figure 4.2: Red internal Blue algorithm mismatch: Actual Blue 'MLS'

Recall, that the 'HB' approach has been shown to yield a better payoff for the Blue player in the MAG example. For the case where the Blue player chooses the 'HB' approach, the results in Figure 4.3 give more intuitive results than when Blue uses the 'MLS' approach. Using the correct Blue approach, 'HB', in its internal computations,

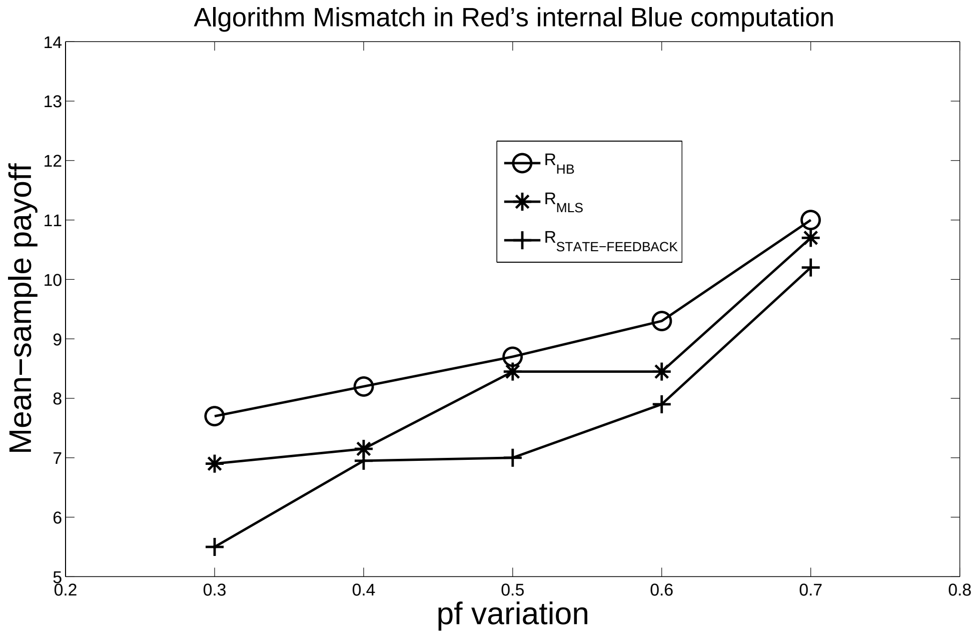


Figure 4.3: Red internal Blue algorithm mismatch: Actual Blue 'HB'

the Red player does achieve a higher mean-sample payoff than using the incorrect Blue approach, 'MLS'. The mean-sample payoff using the state-feedback optimal Red control is again the lowest. Moreover, in such mismodelling scenarios, the Red player's payoff is affected more adversely when it uses a less smart/naïve internal Blue approach ('MLS' vs 'HB') than the actual Blue approach compared to the scenario when Red uses a smarter internal Blue approach when the Blue player is actually using a naïve approach.

One can extend this study to include the mismodelling case $R_{DR}B_a$, where Blue is using the 'a' approach which is not the same as the Red internal 'DR' approach. Recall that in this case, the Red player has to propagate Q_t for every potential observation $y \in Y$, because Red control decision w_t happens before the actual observation y_t. This propagation is computationally burdensome and is not expected to yield any adverse controls for Red as indicated by the mismodelling results above.

Another interesting and surprising result is obtained when we try and force the Blue player to do exactly the opposite of what the Red player's internal computations suggest (the virtual optimal Blue control). This result is for the case where the Blue player is using the 'MLS' approach. The results in Figure 4.4 indicate that the Blue player does worse than using the 'MLS' approach instead of causing any serious problems for the Red player (since the assumption (A-RI) does not hold). Also, the Red state-feedback optimal control does better than the Red control using (4.8). The results obviously are limited to the example in hand and do not reflect any general behavior that can be extended to other problems.

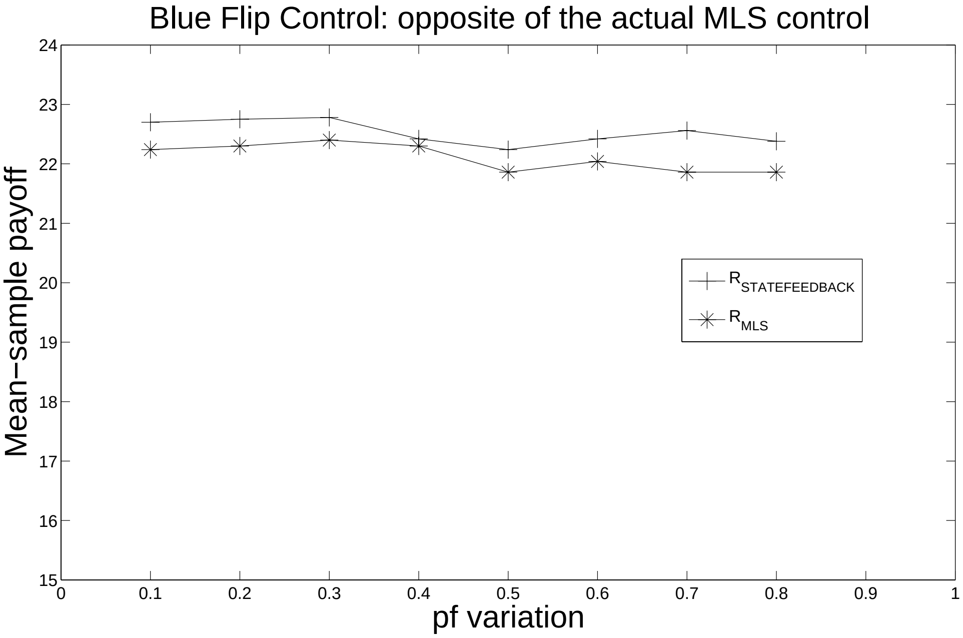


Figure 4.4: Blue control exact opposite of Red's internal virtual Blue computation

4.3 Red Approach Using No Internal Blue Control Model: Brief Insight

The main result of this study is that the automated controller given by (4.8) is optimal for Red under the assumption (A-RI) and Red does employ deception-enabled control choices, when useful. Also, the state-feedback optimal control is sub-optimal for the partially-observed case under the assumption (A-RI). The two cases for this example also indicate that when (A-RI) does not hold (owing to mismatch of algorithms in Red's internal computation and the actual Blue algorithm), the Red player still does better compared to the state-feedback optimal Red control but the optimal control is more dependent on the parameter regime and the actual Blue algorithm.

A simple, one-step example is now discussed to ascertain that it is possible that when Red does not use any model for the Blue player control computation, it may still be optimal to use a controller which is not state-feedback optimal, but which uses deception. In particular, it motivates the pursuit of optimal Red control in the form of a mixed-strategy and demonstrates that such pursuit in not vacuous (though it will not be a part of this dissertation).

The example problem consists of only a single observation followed by one-step dynamics yielding the payoff. The example is as follows. Red has two objects, A and B. The Red objects may be arranged with A on the left and B on the right or vice-versa. Blue will observe, very imperfectly, which side has which object. Red can affect this observation with a control. If Red plays T, then Blue's observation will be correct with probability one, and if Red plays F, Blue's observation will be wrong with probability one. Less extreme probabilities can be used; it simplifies the computation to make the probabilities trivial. The arrangement of the objects is (A, B) or (B, A), with equal probability. First, Red chooses whether to falsify the Blue observation or not; this is the Red control decision. Then, after making the observation, Blue can attack either the left or the right. If Blue attacks the side with object A and wins, then Blue receives -10 points. If Blue attacks the side with object B and wins, then Blue receives -1 point. No points are awarded if Blue loses. Blue is trying to minimize the payoff.

The Red control (T or F) affects the probability of Blue winning. The probabilities of Blue winning against A are 0.7 if Red plays T and 0.8 if Red plays F. The probabilities of Blue winning against B are 0.6 if Red plays T and 0.9 if Red plays F. We consider a maximin value. Note that the state-feedback optimal Red control is T, as with full information, Blue will always attack A, and Blue's expected payoff is -7 when Red plays T and -8 when Red plays F.

In this partially-observed problem, one must consider a mixed Red strategy. We also allow Blue a mixed strategy, although consideration of only deterministic Blue controls (as functions of the observation) is sufficient for optimality. Let the probability that Red plays T be p, and the probability that Blue attacks the side where A *is observed* be q.

As the problem is axially symmetric (i.e., left versus right), the computations of are significantly reduced. If Red plays T, the expected payoff is $J^T(q) = -0.7(10)q - 0.6(1)(1-q)$. If Red plays F, the expected payoff is $J^F(q) = -0.9(1)q - 0.8(10)(1-q)$. Thus the optimal Red control is

$$p^o = \operatorname*{argmax}_{p\in[0,1]}\Big\{ \min_{q\in[0,1]} \big[\, p(-7q - 0.6(1-q)) \\ +(1-p)(-0.9q - 8(1-q))\big]\Big\}.$$

The minimizing value of q always occurs at either $q = 0$ or $q = 1$ (and so Blue would do as well without mixed controls of course). One easily finds that

$$p^o = \operatorname*{argmax}_{p\in[0,1]} \begin{cases} -6.1p - 0.9 & \text{if} p \geq (7.1/13.5) \\ 7.4p - 8 & \text{if} p < (7.1/13.5). \end{cases}.$$

Consequently the optimal Red control is to play T with probability $p^o = (7.1/13.5)$. The state-feedback optimal control for Red is T, but the partial information optimal control for Red plays T only with probability $(7.1/13.5)$. Thus, even without knowledge of the Blue controller, the optimal Red control may choose to falsify the observation in spite of the fact that this control produces a worse expected payoff in the state-feedback case. Also note that the optimal Red mixed strategy did *not* depend on the actual state, only on the distribution of left and right (50% each). One may allow Blue to know the current Red mixed strategy (whose choice *could* depend on the true state but might not), but not the true state itself (outside of potential dependence of the Red strategy on the Red state). This earlier example indicates that this can indeed be a class where deception is fruitful.

4.4 Appendices

4.4.1 Proof of Theorem 4.1.1

Proof. Using (4.5) with $t = r$, and substituting it in (4.6) we equivalently need to show

$$\max_{\overline{\theta}_{[t,T)}} \mathbf{E}[W_T(X_T, q_T)|\ X_t = x, q_t = \tilde{q}] \doteq \max_{\overline{\theta}_{[t,r)}} \mathbf{E}[\max_{\overline{\theta}_{[r,T)}} G(X_r, q_r)|X_t = x, q_t = \tilde{q}] \quad (4.10)$$

where

$$G(X_r, q_r) = \mathbf{E}[W_T(\bar{X}_T, \bar{q}_T)|\bar{X}_r = X_r, \bar{q}_r = q_r]$$

where $\bar{X}_.$ propagates using $\overline{\theta}_{[r,T)}(\bar{X}_., \bar{q}_.)$ and the internal Blue control given by (4.3) and the $\bar{q}_.$ process is propagated by (4.1), with the initial condition $(\bar{X}_r, \bar{q}_r) = (X_r, q_r)$. It will be implicit in subsequent discussion that till time r, the state process $X_.$ is propagated using $\overline{\theta}_{[t,r)}(X_., q_.)$ and the internal Blue control given by (4.3) and the $q_.$ process is propagated by (4.1), with initial condition $(X_t, q_t) = (x, \tilde{q})$. Fix any $\overline{\theta}_{[t,r)} \in \overline{\Theta}_{[t,r)}$ and define:

$$L(x, \tilde{q}) = \mathbf{E}[\max_{\overline{\theta}_{[r,T)}} G(X_r, q_r)|X_t = x, q_t = \tilde{q}] \tag{4.11}$$

$$R(x, \tilde{q}) = \max_{\overline{\theta}_{[r,T)}} \mathbf{E}[G(X_r, q_r)|X_t = x, q_t = \tilde{q}] \tag{4.12}$$

Note that if we prove that $L(x, \tilde{q}) = R(x, \tilde{q})$, then taking max over $\overline{\theta}_{[t,r)}$, applying Lemma (2.2.2) to the appended state (X, q) , and use of conditional expectation would complete the proof of Theorem 4.1.1. We first prove the inequality i.e

$$L(x, \tilde{q}) \geq R(x, \tilde{q})$$

Fix $\overline{\theta}_{[t,r)} \in \overline{\Theta}_{[t,r)}$ and let

$$\overline{\theta}^*_{[r,T)} \in \operatorname*{argmax}_{\overline{\theta}_{[r,T)} \in \overline{\Theta}_{[r,T)}} \mathbf{E}[G(X_r, q_r)|X_t = x, q_t = \tilde{q}] \tag{4.13}$$

which gives

$$R(x, \tilde{q}) = \mathbf{E}[G(X_r, q_r)|X_t = x, q_t = \tilde{q}] \tag{4.14}$$

where $\bar{X}_.$ propagates using $\overline{\theta}^*_{[r,T)}(\bar{X}_., \bar{q}_.)$ and the internal Blue control given by (4.3) and the $\bar{q}_.$ process is propagated by (4.1) with the initial condition $(\bar{X}_r, \bar{q}_r) = (X_r, q_r)$. Since $\overline{\theta}^*_{[r,T)}(\bar{X}_., \bar{q}_.) \in \overline{\Theta}_{[r,T)}$, we get the obvious inequality

$$R(x, \tilde{q}) \leq \mathbf{E}[\max_{\overline{\theta}_{rT}} G(X_r, q_r)|X_t = x, q_t = \tilde{q}] \doteq L(x, \bar{q}) \tag{4.15}$$

where $\bar{X}_.$ propagates using $\overline{\theta}_{[r,T)}(\bar{X}_., \bar{q}_.)$ and the internal Blue control given by (4.3) and the $\bar{q}_.$ process is propagated by (4.1) with the initial condition $(\bar{X}_r, \bar{q}_r) = (X_r, q_r)$. The reverse inequality can be proved very similarly to the proof of the reverse inequality of Lemma 2.2.4, with the state now being $(X_., q_.)$ and appropriate propagation of the individual state components $X_.$ and $q_.$. □

This chapter is in part a reprint of the materials as is appears in,

Rajdeep Singh, William M. McEneaney - *Deception in Autonomous Vehicle Decision Making in an Adversarial Environment*, AIAA Guidance, Navigation, and Control Conference and Exhibit, 2005.

Rajdeep Singh, William M. McEneaney - *Deception-Enabled Control in Stochastic Games with Autonomous Vehicle Applications*, Sixth SIAM control Conference, 2005.

Rajdeep Singh - *Unmanned Vehicle Decision Making Under Imperfect Information in an Adversarial Environment II*, AIAA Guidance, Navigation, and Control Conference and Exhibit, 2005.

Rajdeep Singh, William M. McEneaney - *Exploitation of an Opponents Imperfect Information in a Stochastic Game with Autonomous Vehicle Application*, 43rd IEEE Conference on Decision and Control, 2004.

The dissertation author was the primary author and the co-author listed in these publications directed and supervised the research.

Chapter 5

Urban Warfare Modelling

We now discuss the main steps one goes through in implementation of this technology to large scale problems as of an Urban Warfare Combat. The main goal will be to generate automated controls (reflecting realistic behaviors) on part of both players. First, one must clearly define the (finite) state space. That is, one must develop a model of all possible (physical) states of the system, and this must consist of a finite number of states. In many warfare problems, a state space is defined as the set of all possible entity positions and healths (where "health" is broadly defined, but where one can typically index this by a few numbers). For example, consider a game with n_B Blue entities and n_R Red entities, and in which each of these entities may occupy a position on the "board" where there are, say, n_L positions, and we indicate the set of these positions as $\mathcal{L}$. Let the Blue and Red entity positions be $L_1^B, L_2^B, \dots L_{n_B}^B$ and $L_1^R, L_2^R, \dots L_{n_R}^R$, respectively. Suppose each of these entities has one of four health states, say in $\mathcal{H} = \{\text{destroyed, damaged, needs maintainence, OK}\}$. These health states may be denoted as $H_1^B, H_2^B, \dots H_{n_B}^B$ and $H_1^R, H_2^R, \dots H_{n_R}^R$. A state, $x \in \mathcal{X}$ then corresponds to a vector

$$x = \{L_1^B, \dots L_{n_B}^B, L_1^R, \dots L_{n_R}^R, H_1^B, \dots H_{n_B}^B, H_1^R, \dots H_{n_R}^R\} \in \mathcal{L}^{N^T} \times \mathcal{H}^{N^T}$$

where the superscripts on $\mathcal{L}$ and $\mathcal{H}$ indicate outer product and $N^T = n_B + n_R$. For $n_B = 6$, $n_R = 8$, and $n_L = 1000$, $\mathcal{X}$ is comprised of 4000^{14} states. In a military game such as this with a state space such as that indicated above, the possible controls for any entity might be to move from $l_{31} \in \mathcal{L}$ to adjacent location $l_{171} \in \mathcal{L}$, or say fire at position l_{22}. They may also be more general such as “lay low”, or return fire if fired upon (”tight”). We suppose that the controls for each entity take values in finite sets U_0 and W_0 (containing controls such as those above). Then the control sets for all of Blue and all of Red would be $U = U_0^6$ and $W = W_0^8$, respectively. The allowable controls may be state-dependent and we will discuss more about that in the specific modelling of a small example to follow.

Next, one must determine the transition probabilities for moving from state x to state $\bar{x}$ given controls $u \in U$ and $w \in W$, $P_{x,\bar{x}}(u, w)$. For nontrivial games, these will not be enumerated for each possibility, but instead, will be built up from probabilities of outcomes of individual entity actions. In our example, there would be a probability for Blue Entity 2 to go from health state OK to health state damaged, given that its control is to move from location l_1 to l_{80} while being fired upon by Red Entity 3 at position l_7 while this Red entity is itself under fire from another Blue entity. This defines the dynamics. We have the full information modelling in which we do not have any observation process. Without loss of generality, we will have the defending player as the minimizer and the attacking player as the maximizer. Also, we will now have an exit set game instead of a terminal time game. It can be shown that an exit set game can be arbitrarily well-approximated by a terminal time game. Intuitively, one can assign the terminal time to be the some reasonably high number and when the state enters the exit set, say X^E, trivial dynamics follows for the rest of the time. Note that we will also include running cost in this example (motivated by simulation results demanding some motivation for the attacking team to move towards the target).

5.0.1 Urban Warfare Game Example

The layout of the example graph is as depicted in Figure 5.1. Each numbered ’dot’ on the graph corresponds to a location or a node with coordinates n_x, n_y, and n_z. There are a total of 33 nodes distributed over several levels of the buildings and on the streets, so for this example $\sharp\mathcal{L} = 33$. We have three Blue and three Red teams, $n^B = 3$, $n^R = 3$, and $n^T = 6$. We will only allow three health states, say in $\mathcal{H} = \{\text{destroyed (3), damaged (2), OK (1)}\}$. Then for this example, state $x \in \mathcal{L}^6 \times \mathcal{H}^6$ i.e., a

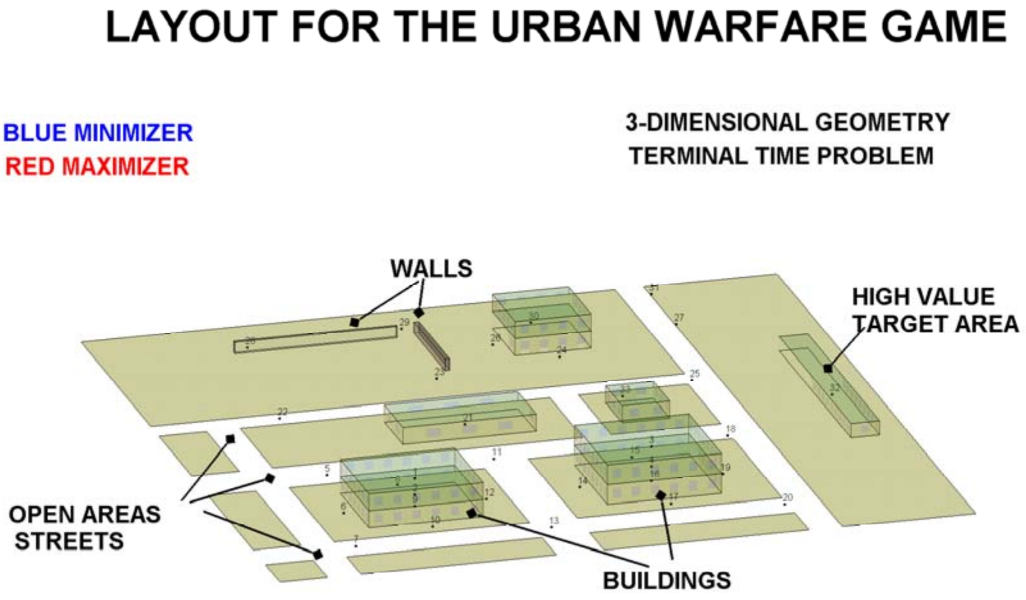


Figure 5.1: Urban Warfare simulation layout

state vector is a 12 dimensional vector, An example of state would be $x \in \mathcal{L}^6 \times \mathcal{H}^6$,

$$x \doteq (5, 8, 1, 28, 25, 21, 1, 1, 2, 1, 1, 1)$$

For x as given above, there are three Blue teams at nodes 5 (completely healthy), 8 (completely healthy), and 1 (damaged) respectively. All the Red teams are healthy and located at 28, 25, and 21. Let $x = (x^B, x^R)$ be the decomposition of the state into state components of each player. Further decomposition of each player's state component into location and health components gives, $x^B = (x^B_L, x^B_H)$ and $x^R = (x^R_L, x^R_H)$. Then at time t, the location of the kth Blue team will be $[x^B_L]_{k,t}$ and the health of the lth Red will be $[x^R_H]_{l,t}$. We will denote $\vec{h}^R_{l,t}$ to be the health distribution of the Red team l. Given the state x, in general the mth component of $\vec{h}^R_{l,t}$ is 1, if $[x^R_H]_{l,t} = m$, where $m \in \mathcal{H}$. The Blue player health also has a similar representation. We will assume that all the teams start in the completely healthy state, i.e. $[\vec{h}^R_{l,0}]_1 = 1, \forall\ l \in \mathcal{L}$ and $[\vec{h}^B_{k,0}]_1 = 1, \forall\ k \in \mathcal{L}$.

Let's define the control in terms of two components, movement and attrition/firing controls

$$[u_t]_k = [[u^M]_{k,t}, \{[u_i^F]_{k,t}\}_{i=1}^3].$$

Then at time t, $[u^M]_{k,t}$ is the movement control and $[u_i^F]_{k,t}$ is the firing control for the kth Blue team (corresponding to an engagement with a Red team i). Similarly $[w_t]_l = [[w^M]_{l,t}, \{[w_i^F]_{l,t}\}_{i=1}^3]$. Given movement controls for the players, one gets the deterministic transition for the location component of the state as

$$[x_L^B]_{k,t+1} = [u^M]_{k,t} \text{ and } [x_L^R]_{l,t+1} = [w^M]_{l,t}$$

where $[x_L^B]_{k,t+1}$ is the node location of the kth Blue team at time $t+1$. Let

$$\mathcal{X}_{\mathcal{L}}^{\mathcal{N}} \doteq \{x \in \mathcal{L}^6 \times \mathcal{H}^6 : ([x_L^B]_{k,t+1} \neq [u_t]_k) \text{ or } ([x_L^R]_{k,t+1} \neq [w_t]_k) \text{ for some } k \in \mathcal{L}\ \}$$

be the set of states which do not correspond to the movement control of at least one Blue or one Red player. Also note that we do not allow maintenance or health recovery, so let

$$\mathcal{X}_{\mathcal{H}}^{\mathcal{N}} \doteq \{x \in \mathcal{L}^6 \times \mathcal{H}^6 : ([x_H^B]_{k,t+1} > [x_H^B]_{k,t}) \text{ or } ([x_H^R]_{k,t+1} > [x_H^R]_{k,t}) \text{ for some } k \in \mathcal{L}\}$$

be the set of states which correspond to improvement in health from time t to time $t+1$. Define

$$\mathcal{X}^{\mathcal{N}} = \mathcal{X}_{\mathcal{H}}^{\mathcal{N}} \bigcup \mathcal{X}_{\mathcal{H}}^{\mathcal{N}} \tag{5.1}$$

Let $\mathbf{U}_t^k$ and $\mathbf{W}_t^l$ be the set of state-feedback controls for Blue team k and Red team l respectively. Given $x \in \mathcal{L}^6 \times \mathcal{H}^6$, $[u_t]_k \in \mathbf{U}_t^k$ and $[w_t]_l \in \mathbf{W}_t^l$ (for each $k \in \mathcal{L}$ and each $l \in \mathcal{L}$), then for any $\bar{x} \in \mathcal{L}^6 \times \mathcal{H}^6$ such that $\bar{x} \in \mathcal{X}^{\mathcal{N}}$, we have

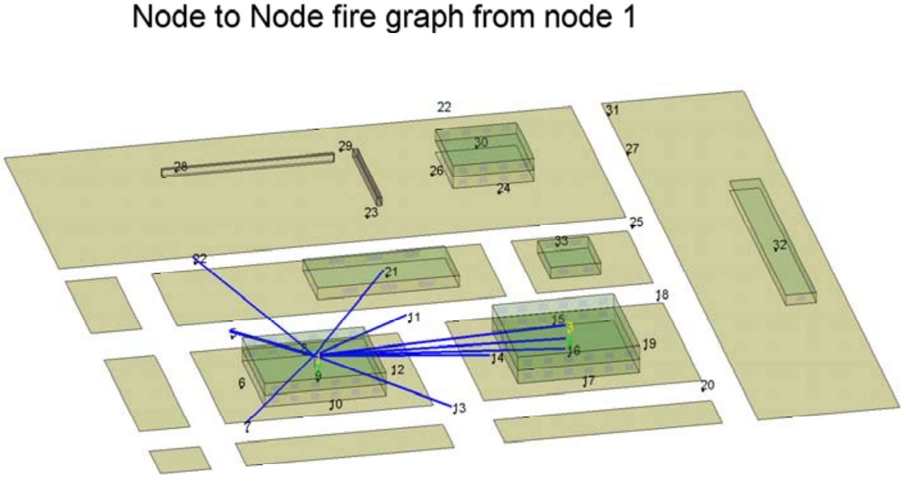
$$\Pr(X_{t+1} = \bar{x} \mid X_t = x, u_t, w_t) = 0.$$

For this example discussion the exit set, X^E, would be a set of states such that all defending teams are in health '3' and all the surviving attacking teams are at the target. Given a state $x \in \mathcal{L}^6 \times \mathcal{H}^6$ and target node, t^*, we will denote the number of attacking teams that survive by $n_{s,x}^A$, the number of attacking teams that survive and are at the target by $n_{s,x}^{A,t^*}$, and the defending teams that survive by $n_{s,x}^D$. Then the definition of the exit set (with t^* as the target) becomes

$$X^E = \{x \in \mathcal{L}^6 \times \mathcal{H}^6 \text{ such that } (n_{s,x}^A = 0) \ \textit{or}\ \Big[(n_{s,x}^D = 0) \text{ and } (n_{s,x}^A = n_{s,x}^{A,t^*})\Big]\} \tag{5.2}$$

Note that by definition that $n^A_{s,x} \geq n^{A,t^*}_{s,x}$ and the second condition in (5.2) holds true even when $n^A_{s,x} = 0$ (no attacking team survives).

We can drop the firing controls ($[u^F_i]_{k,t}$ or $[w^F_i]_{l,t}$), if we allow the teams to fire at any visible opponent team. The visibility or the line of sight mapping can be computed offline as a function of geometry between node to node, node to edge (or vice-versa) and edge to edge. Some examples of type of engagements are given in Figures 5.2-5.3. If no line of sight exists between Blue team k and Red team l at time t, then their is no attrition between that pair of teams at time t. That allows the firing control to be computed deterministically as a function of movement and geometry. So it suffices to look for the optimal path or the movement control for each player and the firing control will be automatically derived from the location and the movement control computation. We now develop the outline for path planning, and attrition (or damage caused by two opposing teams in a firing engagement with each other).



Node to Node fire graph from node 1

Figure 5.2: Node-Node firing graph

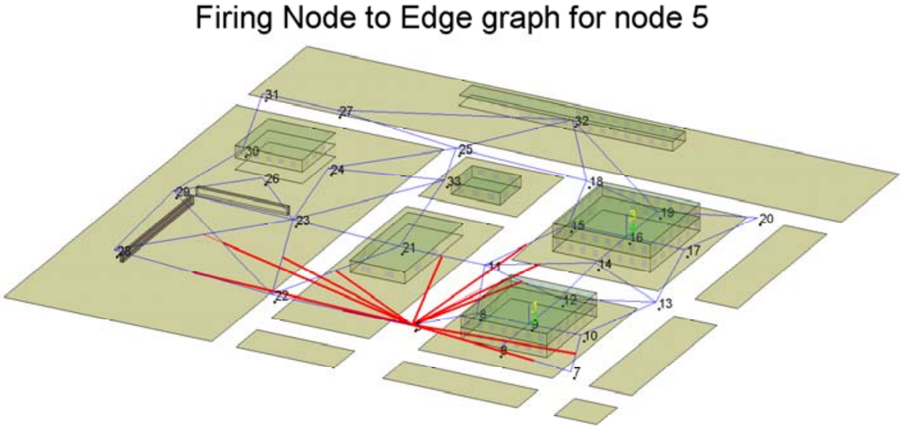


Figure 5.3: Node-Edge firing fraph

Path Planning

In the mathematical sense, the allowable movements are defined by a graph, where the nodes are the locations. Then a movement edge mapping is given by $M : \mathcal{L} \times \mathcal{L} \to B$, where $B = [0, 1]$ and where $M(m, n) = 1$, if and only if there is a direct movement allowed between node m and n. Let

$$E = \{e = (m, n) : (m, n) \in \mathcal{L} \times \mathcal{L}, \text{ and } M(m, n) = 1\}.$$

Then E is the set of all the movement edges. At any time t, a Blue or a Red team can move from a node i to node j if there is an edge on the movement graph. Let

$$E^i = \{j \in \mathcal{L} : e \in E, \text{ where } e = (i, j)\}. \tag{5.3}$$

Then E^i is the set of all nodes having a movement edge connected to node i. Note that $i \in E^i$ and $j \in E^i \iff i \in E^j$. For example in Figure 5.4 there is a movement edge between nodes 1 and 2, so $2 \in E^1$ and $1 \in E^2$, but there is no movement edge between 1 and 10, so $1 \notin E^{10}$ and $10 \notin E^1$.

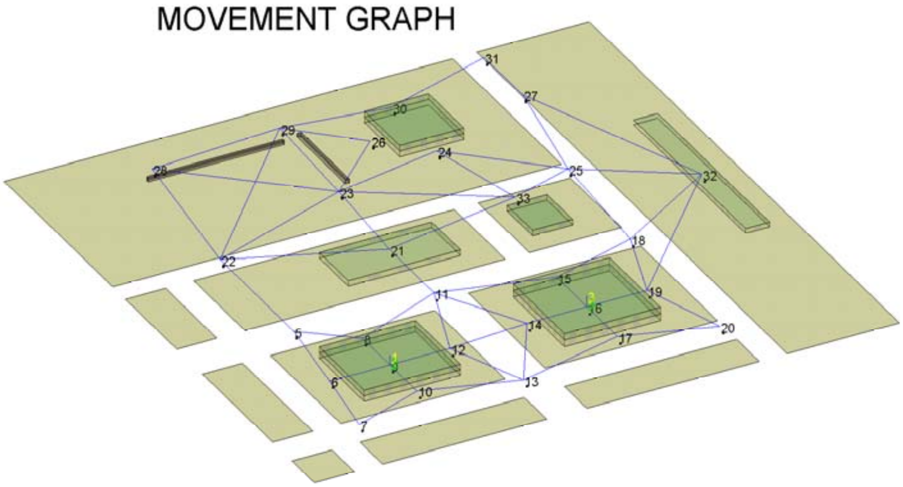


Figure 5.4: Movement graph

The state dependent movement control sets for Blue and Red players at time t is $\mathbf{U}_t^k$ (for the kth Blue team) and $\mathbf{W}_t^l$ (for the lth Red team). Note that given the state x at any time t, $\mathbf{U}^{\mathbf{k}}_{\mathbf{t}} \subseteq E^{[x_L^B]_{k,t}}$ and $\mathbf{W}^{\mathbf{l}}_{\mathbf{t}} \subseteq E^{[x_L^R]_{l,t}}$. In the examples to follow, the building on the right side of the graph (with node number 32 inside that building) will be the target (t^*) to be captured by the attacking team. Let $D : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$. Define

$$D(m,n) = \begin{cases} \sqrt{((m_x - n_x)^2 + (m_y - n_y)^2 + (m_z - n_z)^2)} & \text{if } n \in E^m; \\ \infty & \text{otherwise.} \end{cases}$$

Then, D gives the actual distance (or edge length) between two nodes if an edge exists between them or assigns ∞ to allow for ignoring that path as a potential choice for movement. Let $P^F : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$, be the weighting factor for computing the shortest path based on the geometry. For example, $P^F(m,n) = 1.1$, if m and n are in open area; the distance along an open area edge is penalized by a factor of 1.1. Similarly $P^F(m,n) = 0.95$, if m is in an open area and n is inside a building. Note that if m and n are not of the same type (for example, both are not in the open area), then

$P^F(m,n) \neq P^F(n,m)$. Then the weighted edge length is given by

$$\bar{D}(m,n) \doteq D(m,n)P^F(m,n)$$

where, $\bar{D} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$. Given $\bar{D}$ and E^i, and using standard path planning algorithms (Dijkstra 1959) one can compute the following:

- D^*: shortest distance matrix with the (m,n) entry, $D^*(m,n)$, giving the shortest distance between node m and node n, $D^* : \mathcal{L} \times \mathcal{L} \to \mathbf{R}$.
- P^S: shortest path index matrix, with the (m,n) entry, $P^S(m,n)$, giving the penultimate node along the shortest path between m and n, $P^S : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$.
- l^*: shortest path steps matrix with the (m,n), $l^*(m,n)$, giving the number of nodes along the shortest path from m to n, $l^* : \mathcal{L} \times \mathcal{L} \to \mathbf{N}$.

Note that the first node along the shortest path between m and n is given by $P^S(n,m)$. The shortest distance matrix D^*, gives a measure of proximity to the goal or target and may be used in computing some expected payoff incurred at the terminal time. Similarly, l^* is useful for a graph with equal edge lengths to be used as a measure of proximity to the target. Note that we will be using an equal edge length graph or in other words $D(m,n) = D(p,l)$, for any $(m,n) \in E$ and any $(p,l) \in E$. So, for this particular example we will use l^* as a measure of proximity to the target.

We allow for minor variations along the shortest path from node any node $m \in \mathcal{L}$ to another node $n \in \mathcal{L}$. One of the variations of these paths allows a team to stay at the current location for the current time step. Another variation allows for paths that lead from node m to the target t^* in $l^*(m,t^*)+1$ steps (or one more time step than the the number of steps required by following the shortest path). Note that the best movement will be decided from the cost computations based on not only the shortest distance path but also on the outcome of the engagements as the teams move along their chosen paths. Thus allowing for the path variations introduces some dynamic noise into the system and allow for some possible paths choices for teams heading towards the target. In particular we allow for maximum 5 paths from any node m to the (pre-assigned) target t^*, which are all computed offline. We will refer to any of these path variations as the near-shortest path.

While moving towards the target along the shortest path is an important movement option, one can also compute shortest path to the nearest safe or strategically advantageous node location. We will refer to such course of movement control action as the 'Protect' option. In the example to be discussed, the team choosing the protect option will move to the rooftop of the nearest building. Given the node location m, one can define

$$S^{P,m} = \{n \in \mathcal{L} \text{ such that } n \text{ is on the rooftop of a building}\}$$

and

$$\bar{S}^{P,m} = \{\bar{n} \in S^{P,m} : D^*(m, \bar{n}) \leq D^*(m, n) \text{ , for } \forall\ n \neq \bar{n}\}. \tag{5.4}$$

Given a team at the node position m, the protect option will correspond to moving along the shortest path from m to some $\bar{n} \in \bar{S}^{P,m}$ using the shortest path index mapping P^S.

We will store the kth node visited along the shortest path from m to n as $\mathbb{P}(k, m, n)$, where $\mathbb{P}(1, m, n) = m$, and where n may be the actual target of the game or the node corresponding to the protect option (i.e. $n \in S^{P,m}$). We now outline a small algorithm to obtain the first K nodes along the shortest path from node m to n, i.e. $\mathbb{P}(1 : K, m, n)$. Set $c = 0$, and $k = n$,

```
While ($c < K$ and $P^S(m, k) \neq m$)
    {$k = P^S(m, k)$, $c = c + 1$, $\mathbb{P}(c, m, n) = k$}
If (c< $K$)
    {$\mathbb{P}(c + 1 : K, m, n) = k$.}                    (SP-A)
```

Some team may also choose to move towards the nearest opponent location and one can compute this online in real time given the information on the opponent location (using the above algorithm). Then, the shortest path between Blue team k and the nearest opponent (Red) team $n^o_{B,k}$ can be computed using P^S, where,

$$n^o_{B,k} = \{\bar{l} \in \mathcal{L} : D([x^B_L]_{k,t}, [x^R_L]_{\bar{l},t}) < D([x^B_L]_{k,t}, [x^R_L]_{l,t}),\ \text{for each } l \neq \bar{l},\ l \in \mathcal{L}\} \tag{5.5}$$

The location component of the state transition update is simple: the movement control of each team at time t becomes its new location at time $t + 1$, gives the the location component of the state at time $t + 1$. However some attacking teams may be going towards the target and some may be going towards the nearest defending team to strike an engagement. We define some terminology for various movement options or strategies for the teams. Let the pre-defined target node location be t^*.

- M^1 or (T-T-T): All three teams moving along a near- shortest path towards t^*.
- M^2 or (T-T-P): Two teams moving along a near-shortest path towards t^* and one team towards strategic points or protect nodes (possibly higher elevations) for protecting its own team members.
- M^3 or (T-P-D): One team each going to t^* along a near-shortest path, to the protect node, and along the shortest path to the nearest opponent (attacker) respectively.
- Define movement options for the attacking player as:

$$O^A \doteq [M^1, M^2]$$

- Define movement options for the defending player as:

$$O^D \doteq [M^1, M^2, M^3]$$

Note that the team movement is dependent on the option chosen as defined above. For example at any time t, Blue movement option when it is attacking will be, $M_t^B \in O^A$, whereas while defending it will choose a movement option, $M_t^B \in O^D$. Further, given M_t^B one can define $d_k^{M_t^B}$, to be the destination of the Blue team k when choosing option M_t^B. Then,

$$d_k^{M_t^B} = \begin{cases} t^* & \text{if } (M_t^B \in O^A) \text{ , team } k \text{ assigned to go to the target ;} \\ p^* & \text{if } (M_t^B \in O^A) \text{ , team } k \text{ assigned to to do protect, } p^* \in S^{P,[x_L^B]_{k,t}} \text{ ;} \\ n_{B,k}^o & \text{if } (M_t^B = M^3) \text{ , team } k \text{ assigned to go to the nearest Red, } n_{B,k}^o \end{cases} \tag{5.6}$$

Similarly, destination of the Red team l when choosing option M_t^R is

$$d_l^{M_t^R} = \begin{cases} t^* & \text{if } (M_t^R \in O^A) \text{ , team } l \text{ assigned to go to the target ;} \\ p^* & \text{if } (M_t^R \in O^A) \text{ , team } l \text{ assigned to to do protect, } p^* \in S^{P,[x_L^R]_{l,t}} \text{ ;} \\ n_{R,l}^o & \text{if } (M_t^R = M^3) \text{ , team } l \text{ assigned to go to the nearest Blue, } n_{R,l}^o \end{cases} \tag{5.7}$$

Given $d_k^{M_t^B}$ and $[x_L^B]_{k,t}$, the path (or movement controls) for the Blue team k from time t onwards can be computed using $\mathbb{P}(:, [x_L^B]_{k,t}, d_k^{M_t^B})$

$$[u^M]_{k,t+\tilde{t}} = \mathbb{P}(\tilde{t}+1, [x_L^B]_{k,t}, d_k^{M_t^B}) \tag{5.8}$$

and

$$[w^M]_{l,t+\tilde{t}} = \mathbb{P}(\tilde{t}+1, [x_L^R]_{l,t}, d_l^{M_t^R}). \tag{5.9}$$

The path of each team is based on the choice of options for each player as defined above and also on the assignment of a movement type to each team by the player. We now discuss the other critical component of this example, the attrition modelling.

Attrition Data Generation

Since the attrition level is based on line of sight between nodes, or node to edge (or edge to node), and edge to edge, one can again compute some attrition data based on geometry offline. Let various possible engagement modes, $C \in P^E$, where

$$P^E = \{NN, NE, EN, EE\} \tag{5.10}$$

. The individual elements are simply the acronyms for the type of engagement, i.e. 'NN' implying a node to node engagement. Note that for $C \in P^E$, the teams involved in the engagement may be stationery or moving. However, allowing for being stationery at a node as a possible movement control, we will refer to an engagement $C \in P^E$ as an engagement between teams with movement controls c_a and c_d respectively. Note that the subscripts implies that teams assume the roles of an 'attacker' and a 'defender'. For a given $C \in P^E$, c_a and c_d are defined as below:

$$\begin{cases} c_a \in \mathcal{L} \text{ and } c_d \in \mathcal{L} & \text{if } C = (NN); \\ c_a \in \mathcal{L} \text{ and } c_d \in E & \text{if } C = (NE); \\ c_a \in E \text{ and } c_d \in \mathcal{L} & \text{if } C = (EN); \\ c_a \in E \text{ and } c_d \in E & \text{if } C = (EE). \end{cases} \tag{5.11}$$

At any time t, given a Blue team k and a Red team l and their respective destinations (obtained using movement options $[M^B]_t$ and M_t^R and (5.7)), the movement controls for the Blue team k and the Red team l are obtained using (5.8) and (5.9) respectively. Without loss of generality, let (5.8) give a Blue control $[u^M]_{k,t} = [x^B_L]_{k,t}$, which implies that the Blue team stays at the current location. Similarly, (5.9) give a Red control $[w^M]_{l,t} \neq [x^R_L]_{l,t}$, which implies that the Red team will be moving to location $[w^M]_{l,t} \in E^{[x^R_L]_{l,t}}$. The resulting controls will then define the potential engagement mode, C, between the two teams . Considering attrition from Blue team k on Red team l, we have $C = (NE)$, with $c_a = [u^M]_{k,t}$ and $c_d = ([x^R_L]_{l,t}, [w^M]_{l,t})$. Considering attrition from Red team l on Blue team k, we have $C = (EN)$, with $c_a = ([x^R_L]_{l,t}, [w^M]_{l,t})$ and $c_d = [u^M]_{k,t}$. We will hereon use an attrition level matrix, $A^C_{B,h}(.,.)$ or $A^C_{B,h}$, for the attrition caused by a Blue team in health state h on a Red team (with the appropriate arguments being chosen depending on C given by $[M^B]_t$ and M^R_t, (5.7), (5.8) and (5.9)). In fact one only needs to compute the various attrition level matrix (geometry based) for the Blue attrition on Red for health level 3, $A^C_{B,3}$, and use some constant mappings to obtain

- Attrition level matrices from Blue on Red when Blue is in health state 2. Note that the attrition level on Red will be less for a Blue team in 'damaged' state compared to the when the attacking Blue team is in 'OK' state.
- Attrition level matrices from Red on Blue when Red is in health state 3 and in health state 2. These are expected to be lower than the corresponding attrition levels caused by Blue on Red.
- Finally for Blue or Red in a health state 2 or 3, one can compute the reduced attrition due to the attacker being under fire/attack itself. Again the attrition level, caused by the attacking team under fire itself, will be reduced due to this affect.

We model the geometrical aspect of attrition using the approach outlined below. For example, $A^{NN}_{B,h}(m,n)$ will denote the attrition level caused by a Blue team at node m (in health state h) on a Red team at node n. Then

$$A^{NN}_{B,h} : \mathcal{L} \times \mathcal{L} \rightarrow \text{ (some subset of } \mathbb{Z}^+).$$

Note that a higher value of $A^{NN}_{B,h}(m,n)$ will correspond to higher attrition or damage.

For a higher ground point, m, and a lower ground point, n, one has $A^{NN}_{B,h}(m,n) > A^{NN}_{B,h}(n,m)$. For a point inside the building, m, and a node point ,n, in open area, one has $A^{NN}_{B,h}(m,n) > A^{NN}_{B,h}(n,m)$. Also one naturally has $A^{NN}_{B,\bar{h}}(m,n) > A^{NN}_{B,\underline{h}}(m,n)$, where $\underline{h} \in \mathcal{H}$, $\bar{h} \in \mathcal{H}$, and $\bar{h} > \underline{h}$.

The dynamic interaction (or engagement or fire-exchange) can happen from a team at node $m \in \mathcal{L}$ on a team moving along an edge $e \in E$ or vice-versa, and also between two teams, one moving along the edge $e_1 \in E$ and the other team moving along the edge $e_2 \in E$. We will assume that we have also pre-computed the attrition level matrices for these engagement types where Blue team is attacking with full health. Namely we have pre-computed $A^{C}_{B,3}$, for all $C \in P^E$. Note that for a given health level h one may also choose $A^{NE}_{B,h}(m,e) \geq A^{EN}_{B,h}(e,m)$. We only have three health levels with the 0 level corresponding to no engagement, so we will mainly talk about attrition levels 3 and 2.

Let $\mathcal{A}^{NN}_{L} = \{0,1,2,3,4,5,6,7,8,9,10\}$ define the set of attrition levels (the severity of damage) that a team may cause on an opponent team in a node to node engagement. Let $\mathcal{A}^{NN}_{\overline{L}} = \{0,1,2,3,4,5,6,7,8\}$, $\mathcal{A}^{NN}_{\underline{L}} = \{0,1,2,3,4,5,6,7\}$ and $\mathcal{A}^{NN}_{\widehat{L}} = \{0,1,2,3,4,5,6\}$. Obviously $\mathcal{A}^{NN}_{\overline{L}} \subset \mathcal{A}^{NN}_{L}$, $\mathcal{A}^{NN}_{\underline{L}} \subset \mathcal{A}^{NN}_{\overline{L}}$ and $\mathcal{A}^{NN}_{\widehat{L}} \subset \mathcal{A}^{NN}_{\underline{L}}$. In general, one can define $\mathcal{A}^{C}_{L}$, $\mathcal{A}^{C}_{\overline{L}}$, $\mathcal{A}^{C}_{\underline{L}}$, and $\mathcal{A}^{C}_{\widehat{L}}$ for $C \in P^E$. Let $M^{C}_{B(3\to 2)} : \mathcal{A}^{C}_{L} \to \mathcal{A}^{C}_{\overline{L}}$, then

$$A^{C}_{B,2}(c_a, c_d) = M^{C}_{B(3\to 2)}(A^{C}_{B,3}(c_a, c_d))$$

will give us the attrition level matrix by a 'damaged' Blue on Red, for $C \in P^E$ and c_a and c_d given by (5.11). One example of such mapping is as given below:

$$M^{C}_{B(3\to 2)}(n) = \begin{cases} n-2 & \text{if } n \geq 6; \\ n-1 & \text{if } 3 \leq n < 6; \\ n & \text{if } n < 3. \end{cases} \tag{5.12}$$

In particular, $M^{NN}_{B(3\to 2)} : \mathcal{A}^{NN}_{L} \to \mathcal{A}^{NN}_{\overline{L}}$, for obtaining the reduced attrition due to the Blue being in health state 2 instead of state 3, in a node to node interaction. We will assume similar mapping definitions as (5.12) for the subsequent discussion (with appropriate domain and range of a specific mapping). For asymmetric attrition levels, let $M^C_{B\rightarrow R} : \mathcal{A}^C_L \rightarrow \mathcal{A}^C_{\overline{L}}$, then

$$A^C_{R,3}(c_a, c_d) \doteq M^C_{B\rightarrow R}(A^C_{B,3}(c_a, c_d))$$

be the attrition level caused by a Red team on a Blue team with $C \in P^E$ and c_a and c_d given by (5.11). One can define $M^C_{B\rightarrow R}(A^C_{B,3})$ in similar manner as $M^C_{B(3\rightarrow 2}$ given by (5.12). In particular, $M^{NN}_{B\rightarrow R} : \mathcal{A}^{NN}_L \rightarrow \mathcal{A}^{NN}_{\overline{L}}$, will be used for obtaining the asymmetric attrition between the two players, in a node to node interaction. We obtain he attrition level caused by a Red team in health state 2 on a Blue team using the mapping $M^C_{R(3\rightarrow 2)} : \mathcal{A}^C_{\overline{L}} \rightarrow \mathcal{A}^C_{\underline{L}}$,

$$A^C_{R,2}(c_a, c_d) = M^C_{R(3\rightarrow 2)}(A^C_{R,3})(c_a, c_d).$$

Finally, we compute the reduced attrition due an attacking team being under 'attack itself'; being fired upon from more than one defending team (or being under fire from at least another defending team when attacking a certain defending team). Let $\bar{A}^C_{B,3}$ be the reduced attrition by a completely health Blue team (under attack itself) in a C type engagement with a Red team, then

$$\bar{A}^C_{B,3}(c_a, c_d) \doteq \bar{M}^C_{B,3}(A^C_{B,3}(c_a, c_d))$$

where, $\bar{M}^C_{B,3} : \mathcal{A}^C_L \rightarrow \mathcal{A}^C_{\overline{L}}$. Let $\bar{A}^C_{B,2}$ be the reduced attrition by a damaged Blue team (under attack itself) in a C type engagement with a Red team, then

$$\bar{A}^C_{B,2}(c_a, c_d) \doteq \bar{M}^C_{B,2}(A^C_{B,2}(c_a, c_d))$$

where, $\bar{M}^C_{B,2} : \mathcal{A}^C_{\overline{L}} \rightarrow \mathcal{A}^C_{\underline{L}}$. Let $\bar{A}^C_{R,3}$ be the reduced attrition by a completely healthy Red team (under attack itself) in a C type engagement with a Blue team, then

$$\bar{A}^C_{R,3}(c_a, c_d) \doteq \bar{M}^C_{R,3}(A^C_{R,3}(c_a, c_d))$$

where, $\bar{M}^C_{R,3} : \mathcal{A}^C_{\overline{L}} \rightarrow \mathcal{A}^C_{\underline{L}}$. Finally, let $\bar{A}^C_{R,2}$ be the reduced attrition by a damaged Red team (under attack itself) in a C type engagement with a Blue team, then

$$\bar{A}^C_{R,2}(c_a, c_d) \doteq \bar{M}^C_{R,2}(A^C_{R,2}(c_a, c_d))$$

where, $\bar{M}^C_{R,2} : \mathcal{A}^C_{\underline{L}} \rightarrow \mathcal{A}^C_{\hat{L}}$. Note that the last mapping will be used in real-time, since the movement controls will be decided by the players in real-time. We will use the appropriate attrition level depending on whether the attacking team is 'under attack itself'. For example, as discussed before, given a Blue team k in health h_k and a Red team l in health h_l, one can obtain the appropriate engagement type, C, and the arguments, c_a and c_d, to be used for obtaining $A^C_{B,h_k}(c_a, c_d)$ and $A^C_{R,h_l}(c_a, c_d)$. Let us define

$$1^{B,k}_l = \begin{cases} 1 & \text{if } A^C_{B,h_k}(c_a, c_d) > 0; \\ 0 & \text{otherwise ;} \end{cases} \tag{5.13}$$

At time t, given the state x_t and the controls given by (5.8) and (5.9), one can compute (5.13) for every Red team paired with the Blue team k (and for all k, and vice-versa). Then define

$$1_{B,k} = \begin{cases} 1 & \text{if } \sum_{l=1}^{3}(1^{B,k}_l) > 2; \\ 0 & \text{otherwise ;} \end{cases} \tag{5.14}$$

Then $1_{B,k}$ will be the 'under attack' indicator function for the Blue team k. Similarly, one can define $1_{R,l}$ to be the 'under attack' indicator function for the Red team l. Then, given the locations and healths of all teams, for a given Blue team k in health h_k we define ,

$$A^{C,*}_{B,h_k}(c_a, c_d) = \begin{cases} A^C_{B,h_k}(c_a, c_d) & \text{if } 1_{B,k} = 0; \\ \bar{A}^C_{B,k}(c_a, c_d) & \text{if } 1_{B,k} = 1; \end{cases} \tag{5.15}$$

where the choice of C (and c_a and c_d) is again dependent on the Red team which is paired with the Blue team k in using (5.15). This implies that we will use the matrix $\bar{A}^C$ if the attacking team is under attack itself or the matrix A^C otherwise. Let $P(a)$ be a pre-defined health transition matrix (of size 3×3) corresponding to an attrition level $a \in A^C_{.,.}$. Clearly $P(0) = I_{3\times3}$. Let the attrition matrices for attrition level 3 and n ($n > 3$) take the form:

$$P(3) = \begin{bmatrix} a & b & c \\ 0 & d & f \\ 0 & 0 & 1 \end{bmatrix}$$

$$P(n) = \begin{bmatrix} a_1 & b_1 & c_1 \\ 0 & d_1 & f_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Then one would expect some monotonicity of the last column entries c and f. In particular for the above choice of attrition levels one should assure $c_1 > c$ and $f_1 > f$.

Health Transition

Recall that each team has complete state information including the node positions and the health level on all other teams (allies and opponents). Health transition of team k at any time is dependent on the current health level and the location of all those opposing teams that are capable of exchanging fire/attrition and all those allied teams that can provide cover-fire (to attenuate the attrition level of opposing teams) to team k.

We now give the computation for the transition of the kth Blue and the lth Red team. For given state x, movement options (M_t^B, M_t^R) and movement type assignment (to target etc.), the health transition of the kth Blue team from time t to $t+1$ is given by

$$\vec{h}^B_{k,t+1} = \prod_{l=1}^{3} [\bar{P}^{B,R}_{k,l}] \vec{h}^B_{k,t} \tag{5.16}$$

where

$$\bar{P}^{B,R}_{k,l} = \sum_{h=1}^{3} P(A^{*,C}_{R,h}(.,.)[\vec{h}^R_{l,t}]_h$$

is the average health transition matrix from all the Red teams on the Blue team k. Similarly the health transition of the lth Red team from time t to $t+1$ is given by

$$\vec{h}^R_{l,t+1} = \prod_{k=1}^{3} [\bar{P}^{R,B}_{l,k}] \vec{h}^R_{l,t} \tag{5.17}$$

where

$$\bar{P}^{R,B}_{l,k} = \sum_{h=1}^{3} P(A^{*,C}_{B,h}(.,.))[\vec{h}^B_{k,t]}]_h$$

Note that the arguments of $A^{C,*}$ (defined in (5.15)) in the above equation are obtained using corresponding path computation as given by (5.8) and (5.9) and definitions of connection set E^i and P^E as in (5.3) and (5.10). Also, $\vec{h}^B_{k,t} \in \mathcal{H}^D$ and $\vec{h}^R_{l,t} \in \mathcal{H}^D$ where, $\mathcal{H}^D$ will denote the set of distributions over $\mathcal{H}$. The above health transition computation can be repeated for all Blue and Red teams and for all time t along the paths given by (5.8) and (5.9).

Max-Min Cost Optimization and Control Computation

We note that even for this example the state space is quite large to allow for a recursive Dynamic Programming recursion, once the state-feedback value V_t in defined in terms on state $x \in \mathcal{X}$. We also include running cost to allow for pragmatic controls (or in some scenario's the teams may choose to stay at their current locations). Let the game start at time t and in state $X_t = x$, which is known to both players. We will assume here that Red is attacking and Blue is defending (though one can switch the dummy names to get the reverse scenario). The players will choose the optimal movement option at time t by optimizing the total cost, $\bar{C}(X_{[t,t+\bar{t})})$ (to be defined), by projecting the state $X_t = x$ using movement options and the corresponding paths as discussed in the previous subsections. Note that we are including some running cost here by including the state process $X_{[t,t+\bar{t})}$ instead of only including $X_{t+\bar{t}}$ in the arguments of $\bar{C}$.

For this particular example we project the path of the teams for $\tilde{t}$ time steps according to the chosen options and then forcing all teams (for the remaining $\bar{t}-\tilde{t}$ steps) to move along the shortest path to target node, t^*, from their current location at time $\tilde{t}$. Given M_t^B and M_t^R, the location component of $X_.$ is determined using (5.8) and (5.9). The health distribution of the Blue team b at time s is given by $[X_H^B]_{b,s} \sim \vec{h}_{b,s}^B$, and the health distribution of the Red team r at time s is given by $[X_H^R]_{r,s} \sim \vec{h}_{r,s}^R$, where $t \leq s < t + \tilde{t}$. The health distribution propagation (for $\vec{h}_{b,s}^B$ and $\vec{h}_{r,s}^R$) happens using (5.17) and (5.16). Let $\bar{\mathcal{H}}^D \subset \mathcal{H}^D$ such that

$$\bar{\mathcal{H}}^D = \{\vec{h} \in \mathcal{H}^D : [\vec{h}]_i = 1 \text{ for some } i \in \{1, 2, 3\}\}. \tag{5.18}$$

where $[\vec{h}]_k$ corresponds to the probability of the health state being k. Then, $\bar{\mathcal{H}}^D$ is the set of distributions corresponding to the corner of the simplex. Also for any $\vec{h} \in \bar{\mathcal{H}}^D$ such that $[\vec{h}]_k = 1$, let us retrieve the index corresponding to unity probability mass as $\nu\left[\vec{h}\right] \doteq k$.

From time $t + \tilde{t}$ onwards, let $X_.$ propagate with the location component being determined using (5.8) and (5.9) with $\mathbb{P}$ now being determined for starting location $X_{t+\tilde{t}}$ and destination t^*. We now project all possible trajectories corresponding to starting each team with a health distribution in the set $\bar{\mathcal{H}}^D$ at time $t + \tilde{t}$. Let, $T^I = \{1, 2, 3\}$, denote the common set of indices for the teams of each player. Let $\bar{\vec{h}}_b^B \in \bar{\mathcal{H}}^D$ and

$\bar{\vec{h}}^R_r \in \bar{\mathcal{H}}^D$, for each $b \in T^I$ and each $r \in T^I$. Let $\vec{\mathbf{h}}^B_{b,t+\tilde{t}} = \bar{\vec{h}}^B_b$ and $\vec{\mathbf{h}}^R_{r,t+\tilde{t}} = \bar{\vec{h}}^R_r$. Then at time s, health of the Blue team b is distributed according to $\vec{\mathbf{h}}^B_{b,s}$, $[X^B_H]_{b,s} \sim \vec{\mathbf{h}}^B_{b,s}$, for $t + \tilde{t} \leq s$. Similarly, health of the Red team r at time s is distributed according to $\vec{\mathbf{h}}^R_{r,s}$, $[X^R_H]_{r,s} \sim \vec{\mathbf{h}}^R_{r,s}$, for $t + \tilde{t} \leq s$. As before, the health distribution propagation (for $\vec{\mathbf{h}}^B_{b,s}$ and $\vec{\mathbf{h}}^B_{b,s}$) happens using (5.17) and (5.16). The total cost, $\bar{C}(X_\cdot)$, to be optimized will have contributions from the following factors:

- Goal achievement. Some cost incurred by Red, if Blue has some surviving units at the target, and this grows iteratively for every time step a Blue team stays at the target.
- The survival of the attacking Blue teams, each surviving team has a value V_B.
- The survival of the defending Red teams, each surviving team has a value V_R.

For computing each of the above components we need some more terminology. Let 1_{t^*} be the indicator function for a team being at the target t^*, i.e. $1_{t^*}(l) = 1$, if $l = t^*$. The running cost will provide the attacking team some incentive to move towards the target (which is one of the goals), by assigning an iteratively additive bonus (points) V^{t^*} for each surviving attacking team k at the target for each time r, $t \leq r \leq t + \bar{t}$. We also assign a bonus V^{c,t^*} for each surviving team k to be in close proximity of the target at time $t + \tilde{t}$. The proximity to target can be decided either by the shortest distance matrix, $\bar{D}^*$ or the steps from target l^*. Let $1_{c,t^*}$ be the indicator function for a team being in close proximity to the target t^*, i.e. $1_{c,t^*}(l) = 1$, if $\bar{D}^*(l, t^*) < \mu^D$, where μ^D is some threshold distance dependent on the particular game. Similarly, for a graph with equal edge lengths, one can define $1_{c,t^*}(l) = 1$, if $l^*(l, t^*) < l^{**}$, where l^{**} is some threshold number of steps dependent on the particular game. Now we construct the running cost $\mathbf{C}^{R,X_\cdot}_{t,t+\tilde{t}}$, $\widetilde{\mathbf{C}}^{R,X_\cdot}_{t+\tilde{t},t+\bar{t}}$ and the terminal cost $\mathbf{C}^{T,X_\cdot}_{t+\bar{t}}$ components. Let

$$\mathbf{C}(X_s) = \sum_{b=1}^{3} \{1_{t^*}([X^B_L]_{b,s}) V^{t^*} (1 - [\vec{h}^B_{b,s}]_3)\}$$

for $t \leq s \leq t + \tilde{t}$ and

$$\mathbf{C}^{R,X_\cdot}_{t,t+\tilde{t}} = \sum_{s=t+1}^{t+\tilde{t}} [\mathbf{C}(X_s)] + (1_{c,t^*}([X^B_L]_{b,t+\tilde{t}}) V^{c,t^*} (1 - [\vec{h}^B_{b,t+\tilde{t}}]_3)). \tag{5.19}$$

Similarly one can define

$$\tilde{\mathbf{C}}(X_s) = \sum_{b=1}^{3} \left[1_{t^*}([X_L^B]_{b,s}) V^{t^*} \right] (1 - [\vec{\mathbf{h}}_{b,s}^B]_3)$$

for $t + \tilde{t} < s \leq t + \bar{t}$ and

$$\widetilde{\mathbf{C}}_{t+\tilde{t},t+\bar{t}}^{R,X_.} = \sum_{s=t+\tilde{t}+1}^{t+\bar{t}-1} \left[\tilde{\mathbf{C}}(X_s) \right]. \tag{5.20}$$

The cost component at time $t + \bar{t}$ is

$$\overline{\mathbf{C}}_{t+\bar{t}}^{X_{t+\bar{t}}} = \left[[\sum_{b=1}^{3} (1 - \vec{\mathbf{h}}_{b,t+\bar{t}}^B)] V_B - [\sum_{r=1}^{3} (1 - \vec{\mathbf{h}}_{r,t+\bar{t}}^R) V_R] + \tilde{\mathbf{C}}(X_{t+\bar{t}}) \right]. \tag{5.21}$$

We will use the notation

$$\sum_{\vec{\bar{h}}^B, T^I}^{B} = \sum_{\vec{\bar{h}}_1^B \in \bar{\mathcal{H}}^D} \sum_{\vec{\bar{h}}_2^B \in \bar{\mathcal{H}}^D} \sum_{\vec{\bar{h}}_3^B \in \bar{\mathcal{H}}^D}$$

and

$$\sum_{\vec{\bar{h}}^R, T^I}^{R} = \sum_{\vec{\bar{h}}_1^R \in \bar{\mathcal{H}}^D} \sum_{\vec{\bar{h}}_2^R \in \bar{\mathcal{H}}^D} \sum_{\vec{\bar{h}}_3^R \in \bar{\mathcal{H}}^D}$$

where we imply to sum over all the combinations for all the Blue (Red) team healths in the set $\bar{\mathcal{H}}^D$. Then the total cost can be written as

$$\bar{C}(X_.) = \mathbf{C}_{t,t+\tilde{t}}^{R,X_.} + \sum_{\vec{\bar{h}}^B, T^I}^{B} \sum_{\vec{\bar{h}}^R, T^I}^{R} \left[\prod_{b=1}^{3} \prod_{r=1}^{3} \left[\vec{h}_{r,t+\tilde{t}}^R \right]_{\nu\left[\vec{\bar{h}}_r^R\right]} \left[\vec{h}_{b,t+\tilde{t}}^B \right]_{\nu\left[\vec{\bar{h}}_b^B\right]} \right] \left[\widetilde{\mathbf{C}}_{t+\tilde{t},t+\bar{t}}^{R,X_.} + \overline{\mathbf{C}}_{t+\bar{t}}^{X_{t+\bar{t}}} \right]. \tag{5.22}$$

Then the following max-min computation gives the optimal movement options for both teams:

$$V(X_t) = \max_{M^i \in O^A} \min_{M^j \in O^D} \left[\bar{C}(X_.) \right]. \tag{5.23}$$

Let

$$M^{*B} \in \operatorname*{argmax}_{M^i \in O^A} \min_{M^j \in O^D} \bar{C}(X_.) \tag{5.24}$$

and

$$M^{*R,M^i} \in \operatorname*{argmin}_{M^j \in O^D} \bar{C}(X_.).$$

In particular

$$M^{*R,M^{*B}} \in \operatorname*{argmin}_{M^j \in O^D} \bar{C}(X_.). \tag{5.25}$$

Since we are formulating an exit set game, we keep repeating the above control decision process given by (5.23)-(5.25), until time t^E such that $X_{t^E} = x$, with $x \in X^E$.

5.0.2 Simulating Real World Behavior

We present some simulation results at this point using some snapshots of the simulation at chosen time instants to illustrate the observed behavior or strategy patterns. More specifically we will discuss the 'Protect' behavior of Blue and the 'Feint' behavior of Red. Recall that the attacking team is the maximizer and defending team is the minimizer.

Protect Behavior

To illustrate the protect behavior we assume that the attacking Blue team are in a specific locations, with one of the teams at the rooftop of a building near the target. In the Figure 5.5 the Blue team 1 is on rooftop of a building at node 3. The other two Blue teams are both located at node 23. The defending Red teams start at node locations 32, 20, and 31. All teams start in the health state 1, which gives $x = \{3, 23, 24, 32, 31, 20, 1, 1, 1, 1, 1, 1\}$ as the initial state.

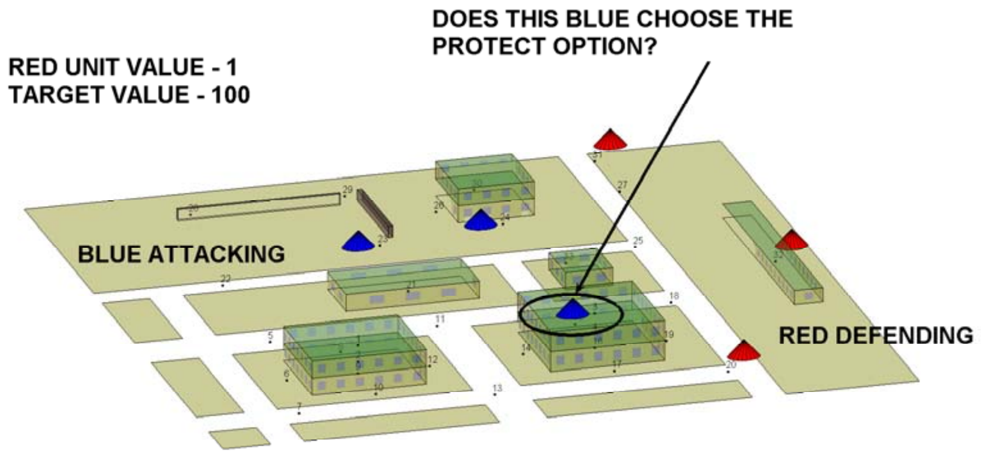


Figure 5.5: Protect scenario: Introduction

The players compute the optimal movement options and corresponding path or movement controls using (5.24)-(5.25), (5.8)-(5.9), and (5.16)-(5.17). If the Blue team 1 at node 3 control computation yields the movement control $[u^M]_{1,1} = 3$, we say that the Blue team chooses to provide cover fire to its allies moving towards the target. It chooses to stay on the rooftop and fire at the opponent teams in line of sight. The movement control to 'stay put' at the current location can come from either the protect option M^2 or M^1 since we allow for some path variations to the shortest path going towards the target. One needs to ascertain if the Blue player chose the protect option or M^2 as the optimal option from (5.24) and if so, what team was assigned to go to the nearest rooftop. If the Blue team 1 at node 3 is the one assigned to go to the nearest rooftop, in this it case stay at its current location, node 3 ($3 \in S^{P,3}$ by (5.4), in fact $S^{P,3} = \{3\}$). The example scenario is run with three different sets varying the value of Blue team survival V_B and the initial state x. In the snapshot captured in Figure 5.6, $V_B = 1$, $V_R = 5$, $V^{t^*} = 100$, and $V^{c,t^*} = 30$. The initial state is $x = \{3, 23, 24, 32, 31, 20, 1, 1, 1, 1, 1, 1\}$.

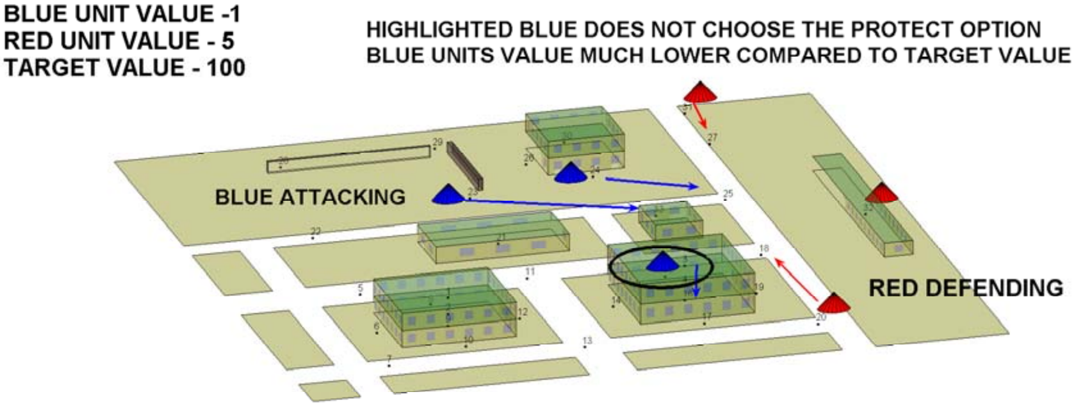


Figure 5.6: Protect scenario: Low Blue survival cost and engagement

The automated Red control using (5.25) gives $M^{*R,M^{*B}} = M^1$ (or all Red going towards the target). The automated optimal control computation using (5.24) gives $M^{*B} = M^1$, so all the Blue teams are going to the target. In particular the Blue team starting at node 23 is moving towards the target and heading towards node 25 and team 1 is moving down from rooftop going towards the target to node 4. Clearly the Blue player does not choose the protect option in this case. However in the snapshot captured in Figure 5.7 with $V_B = 3$, the automated optimal control computation using (5.24) gives $M^{*B} = M^2$, the Blue team 1 at node 3 is assigned the 'To protect' movement.

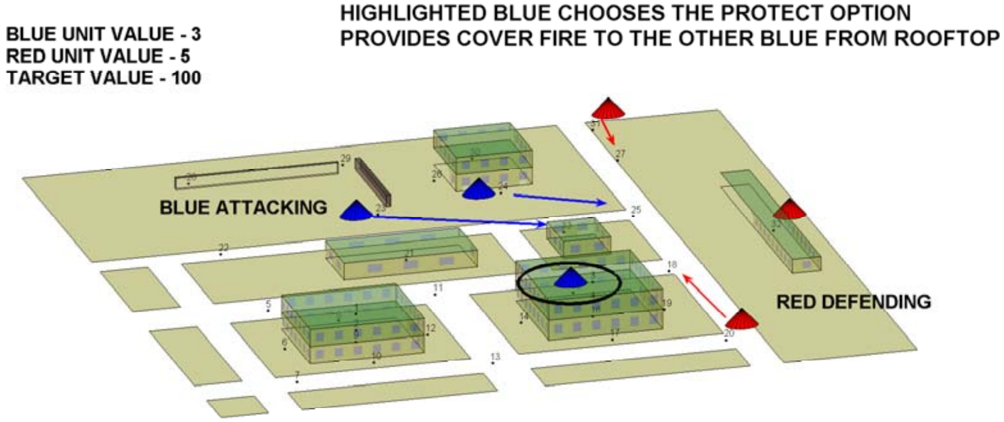


Figure 5.7: Protect scenario: Low Blue survival cost and engagement-II

Though the Blue teams, at node 23 and 24, have same movement controls as before, team 1 now stays at node 3. In fact, any further increase in V_B will yield similar results. One can conclude that the Blue team chooses to protect (or provide cover-fire) to other Blue teams, if V_B is large enough for the Blue player to value its own teams survival (reflected by the contribution to $\bar{C}(X_.)$ in (5.22).

However, with $V_B = 3$, but now moving the other two Blue teams farther from the target, say at node 21, one gets an interesting result. Refer to the snapshot in Figure 5.8, in which the initial state is $x = \{3, 21, 21, 32, 31, 20, 1, 1, 1, 1, 1, 1\}$. Starting

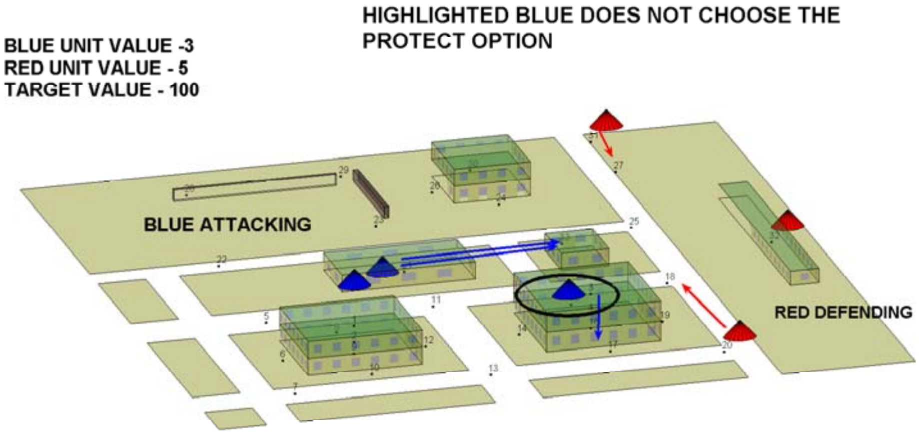


Figure 5.8: Protect scenario: High Blue survival cost and no-engagement

at this location the two Blue teams at node 21, the automated Blue movement option is $M^{B*} = M^1$. In particular the movement of the two Blue teams at node 21 to node 31 does not allow for any engagement with any Red teams (for the movement controls given by their respective optimal movement option in (5.25)). No engagement happens due to no line of sight between the Red teams starting positions, edge movements and the starting positions of the two Blue teams (node 21) or their respective movement edge $(21 - 33)$. Even with Blue cost $V_B = 3$, the Blue team 1 at node 3 chooses to go towards the target since its other team members are not in need of immediate cover-fire and hence the iterative bonus to be at the target V^{t^*} seems to play a bigger role in this scenario.

Feint Behavior

Now we turn to the study a behavior commonly known as Feint. For the Feint scenario, certain Red player chooses some Blue teams and wants to divert their attention

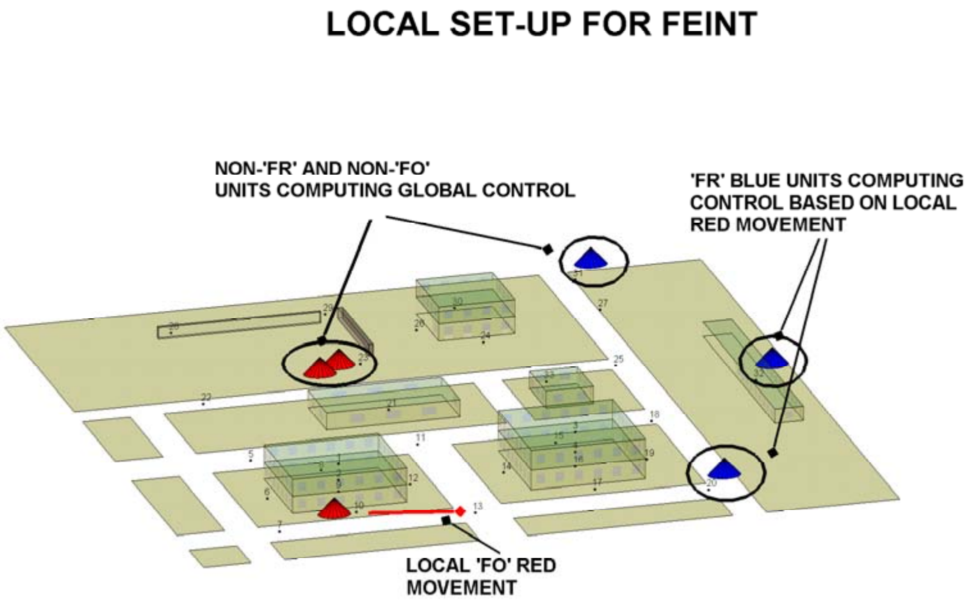

LOCAL SET-UP FOR FEINT
NON-'FR' AND NON-'FO' UNITS COMPUTING GLOBAL CONTROL
'FR' BLUE UNITS COMPUTING CONTROL BASED ON LOCAL RED MOVEMENT
LOCAL 'FO' RED MOVEMENT

Figure 5.9: Feint scenario: Local set up

so that some other Red teams face less resistance, from the remaining Blue teams, on their way to the target or some strategic location. We call the Blue teams (group) that Red wants to lure as the ’Feint-Receptive’ or ‘FR’ Blue team(s) and the Red teams that are being employed to create this diversion will be called the ’Feint-Offensive’ or ‘FO’ Red teams. Generally the non-‘FO’ Red teams (more in number than the ‘FO’ Red teams) ’stays put’ or in a low or no activity state for the time the ‘FO’ Red teams are advancing to induce engagement with the ‘FR’ Blue teams. Also the ‘FR’ Blue team is generally chosen to be larger in size than the non-‘FR’ set of Blue teams, so that if Red succeeds in drawing the larger ‘FR’ Blue group, it can take the advantage of asymmetric engagement on the ignored route (non-‘FR’ Blue vs non-‘FO’ Red to cause maximum damage).

In full state-feedback, the individual Blue and Red teams can compute the movement options and corresponding controls based on knowledge of all the Red and Blue team states. So even in the scenario being set up to demonstrate Feint, controls for the ‘FR’ Blue can be computed using (5.19)-(5.24) taking into account all the Blue and

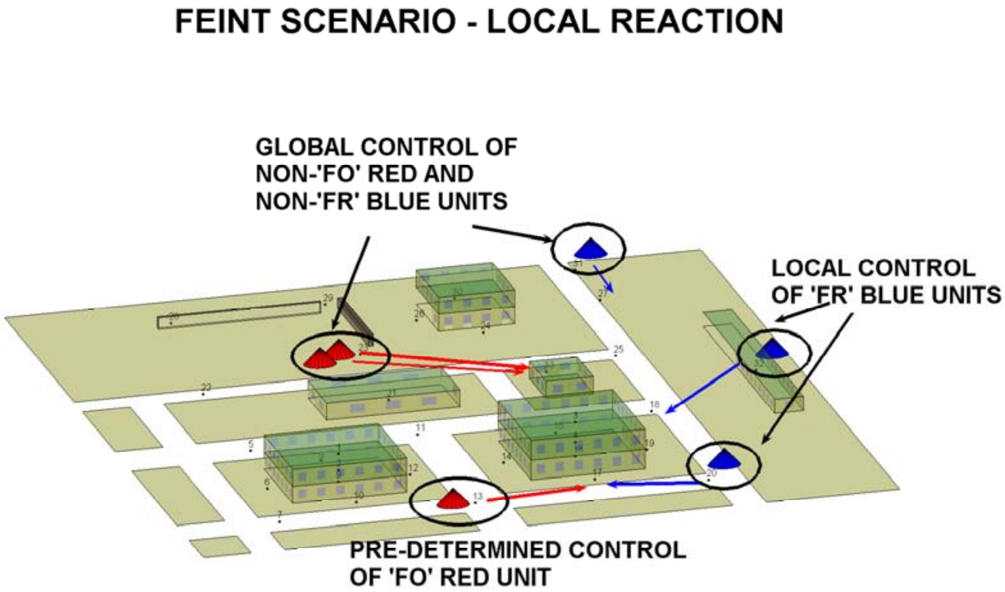

FEINT SCENARIO - LOCAL REACTION
GLOBAL CONTROL OF NON-'FO' RED AND NON-'FR' BLUE UNITS
LOCAL CONTROL OF 'FR' BLUE UNITS
PRE-DETERMINED CONTROL OF 'FO' RED UNIT

Figure 5.10: Feint scenario: Local reaction

the Red teams. We will call this computation as global control computation and the set of control choices for the ‘FR’ Blue teams as $\tilde{U}_G^{FR}$. We also compute the global controls for all the other Blue and Red teams; the controls for ‘FO’ are designed to induce the ‘FR’ Blue and hence they are pre-determined by the Red player.

Note that to achieve this effect the ‘FO’ Red teams are trying to engage the ‘FR’ Blue teams into a local engagement (where the ‘FR’ Blue team ignore the presence of the non-‘FR’ teams which are farther away. In this set up the ‘FR’ Blue compute the controls by reacting locally to counter to the ‘FO’ Red teams purposeful movement/control. In a little more details, the ‘FR’ Blue teams compute their best movement option using (5.19)-(5.24), but only considering the contribution from the ‘FR’ Blue teams and ‘FO’ Red teams to the payoff and propagating the health distributions for these teams as well (involved in a local engagement, ignoring the other Blue and Red teams). In the simulation runs, this computation can be done by simply turning the initial health state of the non-‘FR’ Blue teams and non-‘FO’ Red teams to 3 (destroyed) and proceed with the regular algorithm. We call this the local reaction of the ‘FR’ Blue teams and the set

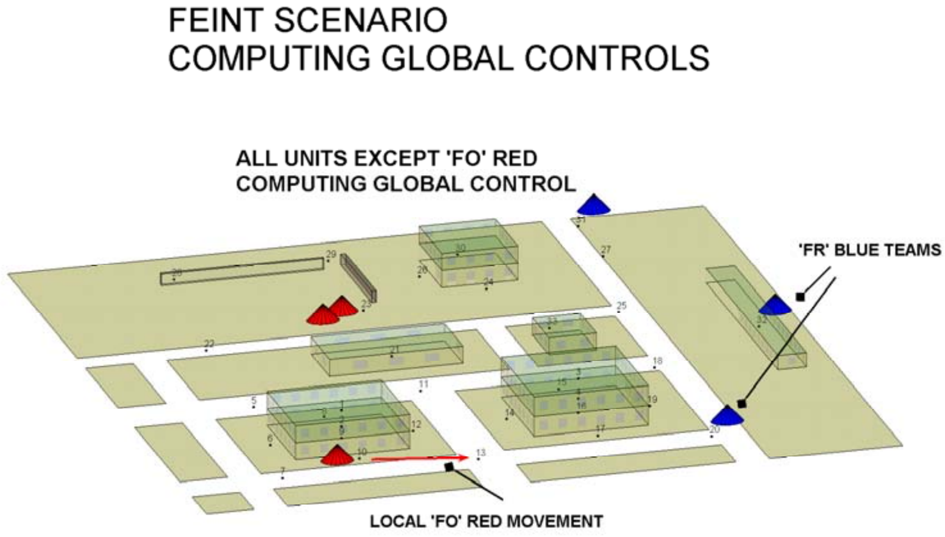


Figure 5.11: Feint scenario: Global set up

of controls for the ‘FR’ teams is called $\tilde{U}_L^{FR}$.

One can continue this for few time steps. Then we ’Feint-alert’ the Blue player (cautioning that the Red player is attempting Feint) if

$$[\tilde{U}_L^{FR}]_i \neq [\tilde{U}_G^{FR}]_i$$

In other words if the local Blue control for each ‘FR’ team ’i’ does not match with the global Blue control for the ’i’th ‘FR’ team, we conclude that the Red player is attempting to Feint and draw the Blue teams into some sub-optimal local reaction to gain advantage by the use the non-‘FO’ Red teams (that were ignored in the local ‘FR’ Blue control computation). Note that employing full state-feedback Feint is hard to achieve since full information is available to both the players. The study is motivated by some observed scenarios in a military application where team units may react locally.

For this example the choice was driven by geometry, distance between the Blue and the Red teams, line of sight and some experimentation. Another hard aspect of studying Feint is to find out the ‘FO’ Red teams and the ‘FR’ Blue teams. In this

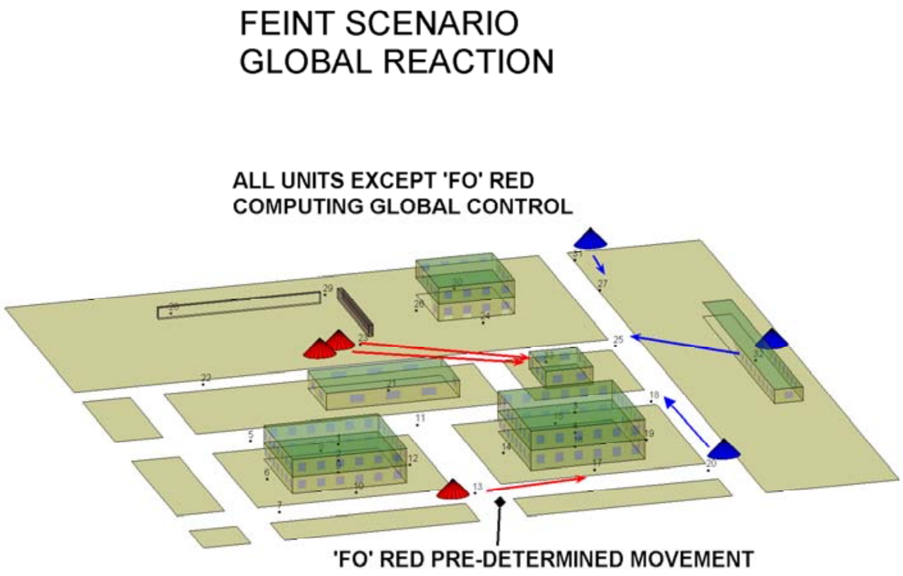


Figure 5.12: Feint scenario: Global reaction

example global and local controls were computed for 2 successive time steps. The ‘FO’ Red teams control were pre-determined, the non-‘FO’ Red teams stayed put for the first time step and computed global control for the second time step, the ‘FR’ teams computed both global and local controls for two successive time steps and the non-‘FR’ Blue teams compute global controls for both time steps. Figures 5.9 and 5.11 identify the ‘FO’ and non-‘FO’ Red teams and ‘FR’ and non-‘FR’ Blue teams and mark them for local or global control computation. In this scenario the Red player wants to move all its teams to the strategic location node 32 (t^*) on the west side of the layout. The Red team at node 10 is the single ‘FO’ Red team and Blue teams at nodes 32 and 20 form the ‘FR’ Blue teams. The other non-‘FO’ Red teams are both at node 23 and the non-‘FR’ Blue team is at node 31. The ‘FO’ Red team causes initial movement along the edge $10 - 13$ on its way to the target node 32. The other Red teams just wait at their initial locations. Blue computes $\tilde{U}_L^{FR}$ and for ‘FR’ teams at 32 and 20 and the non-‘FR’ Blue team at node 31 computes global control in Figure 5.9, whereas in Figure 5.11, Blue computes $\tilde{U}_G^{FR}$ and for ‘FR’ teams at 32 and 20 ant the non-‘FR’ Blue team at node 31 also computes global control.

Refer to Figure 5.10 as the first step transition from Figure 5.9, the case where 'FR' Blue teams compute the local control. In the local reaction Blue team at node 32 in the 'FR' teams moves to node 18, whereas the team at node 20 moves to node 17 to provide resistance to the advancing Red 'FO' team. However in Figure 5.12, the first step transition, using global controls for the corresponding teams, is to move to nodes 25 and 18 respectively. Clearly $[\tilde{U}_L^{FR}]_{32} \neq [\tilde{U}_G^{FR}]_{32}$ and $[\tilde{U}_L^{FR}]_{20} \neq [\tilde{U}_G^{FR}]_{20}$. Thus, we conclude that the 'FO' Red team at node 10 is trying to divert the Blue 'FR' teams teams attention from the Red teams at 23 and 25. In the evolution of the simulation Red is able to achieve its objective but since the simulation results are random, employing Feint may not always end in Red's favor.

This chapter is in part a reprint of the materials as is appears in,

Rajdeep Singh, William M. McEneaney - *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, CRC press, To appear.

The dissertation author was the primary author and the co-author listed in these publications directed and supervised the research.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The standard state-feedback solution to a two-player zero-sum stochastic game in the discrete-time/discrete-space domain is obtained using a lower value. An application to the MAG example problem in the state-feedback section and analysis of state-feedback solution highlighted the differences in the nature of state-feedback and partially-observed games, which make the latter substantially harder to solve. The state-feedback solution is then used to obtain the certainty-equivalent or the 'MLS' control (and some other single-distribution based control) for the Blue player when Blue uses a stochastic modelling of the Red control.

With the recently proposed Deception-Robust theory, see (McEneaney 2004), the Blue player can generate a deception-robust controller (with partial information based on observation conditioned distributions). In this deception-robust approach, the information state for Blue takes the form of a maximal cost over the space of feasible conditional probability distributions on the state. This information state combined with a certain "generalized state-feedback" value function generates the Blue controller. Whereas the theoretical results of the deception-robust approach, referred in this dissertation, were firmly established, some results related to the information state are rederived with a refined mathematical definition of the information state. In particular the dependence of the earlier definition of the information state (at any time t) on $[\vec{w}]_{[0,t)} \in [W^n]^t$ is now removed. The information state propagation and certain robustness properties were re-derived with this new definition. The above redefinition does not change any of basic results from (McEneaney 2004) but served the purpose of providing a somewhat simplified mathematical definition of the information state.

The application of various Blue approaches (including the deception-robust theory) to the MAG gave us the following main results:

- The deception-robust control is the optimal Blue control and robust to adversarial noise in the observation process.

- The deception-robust control never under-performs the 'MLS' controller. The advantage of the deception-robust approach varies on the information level and the parameter regime, but is the highest in the most imperfect information scenario. This confirms the robustness properties of the deception-robust approach (and its usefulness for such games) and also highlights the inadequacy of the 'MLS' controller due to its high sensitivity on the 'MLS' estimate.

- With the the max-plus sum of max-plus delta functions, the information state propagation complexity is equivalent to the complexity of propagation of Q_t, the set of feasible distributions. Initial distributions of type q_{ij}^G (at different possible Red states) gives the best Blue payoff compared to using relatively flatter (q^U or q^{NU}, uniform or non-uniform) distributions in the deception-robust approach. The same lack of knowledge can be modelled using different set of initial distributions, of which the one which models the initial distribution assuming an intelligent adversary is better for Blue (using distributions of type q_{ij}^G).

- A lower number of initial distributions is needed if some intel is available on the initial Red state or some reduced set of Red controls W^*, from which the actual Red control is being exercised (which also leads to slower growth of Q_t).

- When using distributions of type q_{ij}^G, the pruning speed is slower (or size of Q_t is not reduced substantially after pruning). However, the performance is not very sensitive with the error tolerance, μ, so a relaxed error tolerance is admissible allowing for higher pruning speeds. The growth of Q_t is slower using initial distributions of type q^U or q^{NU}, but one needs tighter/stricter error tolerance, μ, for getting the desired performance.

- In the partially observed game with adversarial control, the 'MLS' may not reflect the true representation of the actual state. The 'MLS' controller performance is comparatively reasonable with less imperfect information (say, correct apriori knowledge of W^n) and some good stochastic model of the Red control (a $p^B_{\vec{w}}$ which closely represents the actual Red control). However, this may not be guaranteed in a partially-observed game. Results using good stochastic model only show reasonable improvements in performance for the MAG example when Blue also has apriori knowledge of W^n (and not the incorrect set W^n_I).

- The Risk-Sensitive (RS) Blue control outperforms the 'MLS' control with appropriately chosen parameter κ. The deception-robust control out-performs the RS control with up to 50% reduction in the payoff.

- The key to tractability is that the costs are only initial and final, and in particular, the costs to the players to affect the observation process is only indirectly felt through the effects those control have on the state process (and hence the terminal state).

- Deception is useful for the Red player but its utility depends on the parameter regime and the control decision process of Blue. With good sensor models (that detect even stealthy Red entities quite well) and intel (essentially leading to more accurate information), it becomes difficult to achieve deception, sometimes, even when Blue uses the standard 'MLS' control. The reduction in payoff (better for Blue) as the observation probability of the stealthy Red entities increases, confirms this conclusion.

The basic building blocks of an approach to deception-enabled control have been constructed. An automated deception-enabled Red control was proposed under the assumption (A-RI). It is demonstrated that if Red has a (perfect) model of the Blue controller where that Blue controller is restricted to operating of a observation-conditioned probability process using a stochastic Red model, then one obtains a Red controller which employs deception where appropriate. This result is not surprising since Red is omniscient or it has all the possible information, including the information set of Blue. More importantly and not so obviously, it is seen that this deception-enabled Red control performs rather well when Red is significantly mismodelling the simulation dynamics and/or the Blue control algorithm. The main results for the deception-enabled study are summarized below

- For Red, under assumption (A-RI), the optimal control is not identical to the optimal state-feedback Red control, even though Red has complete state information. In particular, it demonstrates that a control for Red, which is not state-feedback optimal, may nonetheless be optimal for Red due to the deceptive effect it has on the Blue observation process.

- The usefulness of deception might be restricted variably on various parameters sets. The utility of deceptive control based on the stealthiness of Red entities is dependent on the parameter regime, naturally related to the quality of information dissemination associated within that parameter regime. Recall, that the state-feedback optimal control uses stealth for affecting the observations. In particular with increasing observation probability of the stealthy Red entities, the advantage of obtaining better payoffs for Red is undermined.

- The mismodelling results for the case where the assumption (A-RI) does not hold (due to the Red players's internal Blue algorithm not being the same as the actual Blue algorithm) also indicate that the automated control proposed under assumption (A-RI) still never gives a worse payoff than using the state-feedback optimal control for Red. In particular when Red internal mismatched model is poorer than the actual Blue algorithm we found that Red does worse. In the flip case, or the case where Red uses a better internal Blue model than actual Blue algorithm, the results show that the automated deception-enabled control outperforms state-feedback optimal Red control. The results in that case are however not conclusive about a definite advantage to use a better internal Blue model. A larger number of simulation runs could possibly give a more concrete and useful result.

In the Urban Warfare Modelling section, we demonstrated how some commonly observed real-world war behaviors can be modelled appropriately and automatedly generated in a simulated gaming environment. The discussion also looked into achieving computational efficiency with offline data generation and some heuristics that reasonably capture the dynamical behavior of the problem.

We now look at some natural extension of the research in this dissertation as future work.

6.2 Future Work

The deception-robust controller explicitly reasons about deception and handles deception better that the risk-averse approach, but this improvement comes at a substantial computational cost. For a given, fixed computational limit, depending on the specific problem, the additional approximations which must be made in order for the deception-robust controller to be computed may reduce its effectiveness, and it is not obvious which approach will be more successful. The results from this dissertation shed some light on the utility or the value of specific intel, which makes the deception-robust approach computationally less burdensome. However, the benefits of using the computational intensive deception-robust approach vis-a-vis a more standard speedier approach like the 'MLS' or the Risk-Sensitive approach seems to be problem specific. Any technology or heuristic which assists in determining which approach the Blue player should be using (based on a given problem) will be very useful. Such decision making will certainly be driven by the level of imperfection or, alternatively speaking, the level of intel for a given problem. There are several challenges with implementing the deception-robust and the Risk-sensitive approach that need more attention.

- For the RS approach, currently, κ is based on a simulation results repeated over some specific parameter regimes. Naturally the optimal κ would be problem specific. The computation of an optimal κ is an area of future study. Also, for large problems, it is not even feasible to compute $V(x)$ offline, $\forall\ x \in \mathcal{X}$ (which is generally achievable for reasonably sized problems). Instead, one may use hierarchical techniques to decompose the problem. Further, even this may not be sufficient to make the problem computationally tractable. In that case, one may allow Blue and Red to search only over move strings of several steps where these move strings may be partially random. One then applies some heuristic value approximation at the end of the short time-horizon look-ahead. This is the approach we used for computing a heuristic value, in implementing the state-feedback example, discussed in the Urban Warfare modelling section. Lastly, instead of computing $V_t(i)$ for all $i \in \mathcal{X}$, one can restrict the computation to only those i for which $[\widehat{q}_t]_i$ is not too far below the argmax over all $[\widehat{q}_t]_i$. Of course, this shortcut is only possible if the computation is done in real-time.

- The computation or some reasonable approximation of $V(x, q)$ (much more computationally intensive than $V(x)$) is also an area of future study. Simplistic approximations like the one used in this dissertation were good enough for the example problem discussed, but one needs to find more general approximation techniques. These approximations should be representative of the advantage to Red (or the disadvantage to Blue), resulting from the Blue player's belief that the true state X is well-represented by the distribution q. For example, the following approximation lacks the required representation.

$$V_t(j, q) \approx \sum_{i \in \mathcal{X}} \bar{V}_t(j, i)[q]_i \tag{6.1}$$

where we assume that we can somehow compute $V_t(j, i)$. Let's assume that given some $j \in \mathcal{X}$, $V(j, i)$ takes positive values when the state i is very different from the actual j and negative values when the state i is very close to the actual state j (lower payoff is favorable to Blue). We also assume that at terminal time T, $\bar{V}_T(j, i)$ only depends on j (similar to $V_T(j, q)$). If at terminal time T, $\bar{V}(j, i) = \mathcal{E}(j)$, then using (6.1) one gets,

$$V_T(j, q) \approx \sum_{i \in \mathcal{X}} \mathcal{E}(j)[q]_i = \mathcal{E}(j)$$

which confirms that the suggested approximation has the correct form. Let us assume that at some time t, $[q_t^1]_i = 0.5$ and $[q_t^1]_j = 0.5$ and $[q_t^1]_k = 0 \quad \forall\ k \neq i$

and $\forall\ k \neq j$ and $k \in \mathcal{X}$). Let's assume that states i and j are such that i is favorable for Blue whereas j is not favorable to Blue (say in some equal measure). Now assume that $[\tilde{q}_t^1]_i = \frac{1}{N}$ ($\forall\ i \in\ \mathcal{X}$, where $N \doteq \#\mathcal{X}$ or the dimension of $\mathcal{X}$, total number of states). The two distributions (q_t^1 and $\tilde{q}_t^1$) give very different beliefs that Blue might have of the actual state, however one can expect $V_t(j, q^1)$ not be very different from $V_t(j, \tilde{q}^1)$ with the form given by (6.1).

Deception-enabled controllers under assumption (A-RI) rely heavily on perfect modeling of opponent controllers. This formulation is naturally susceptible to mismodelling by Red of the Blue control algorithm, as the space of opponent controllers may be huge. The question of existence of potentially deceptive automated controllers which do not presuppose a model of the opponent control algorithm is nontrivial. Hence, two-player zero-sum stochastic games where the information patterns for both players are not nested will be a challenging future research.

For the Urban warfare modelling, the next step is to extend the modelling to the partial information set-up. This problem is much harder using even with the risk-sensitive Blue approach. Hierarchical methods, mapping the results from the small scale problems to approximate ensemble behaviors of higher scale problems (using clustered states) would be a challenging future study.

Bibliography

Akian, M. (1999), 'Densities of idempotent measures and large deviations', *Trans. Amer. Math. Soc.* **351**, 4515–4543.

Basar, T. & Bernhard, P. (1991), *H_∞ –Optimal Control and Related Minimax Design Problems*, Birkhäuser.

Basar, T. & Olsder, G. (1982), *Dynamic Noncooperative Game Theory, Classics in Applied Mathematics Series*, Originally pub. Academic Press.

Cuninghame-Green, R. (1979), *Minimax Algebra, Lecture Notes in Economics and Mathematical Systems 166*, Springer.

D. Ghose, M. Krichman, J. S. & Shamma, J. (2000), Game theoretic campaign modeling and analysis, *in* 'Proceedings of the 39th IEEE CDC Conference', Sydney, Australia, pp. 2556–2561.

Dijkstra, E. W. (1959), 'A note on two problems in connexion with graphs', *Numerische Mathematik. I* pp. 269–271.

Dinah Rosenberg, E. S. & Vieille, N. (2004), 'Stochastic games with a single controller and incomplete information', *SIAM Journal on Control and Optimization* **43**, 86–110.

D.P. Bertsekas, D.A. Castañon, M. C. & Logan, D. (1999), Adaptive multi-platform scheduling in a risky environment, *in* 'Advances in Enterprise Control Symp. Proc.', pp. 121–128.

Elliott, R. J. & Kalton, N. J. (1972), 'The existence of value in differential games', *Memoirs of the Amer. Math. Society* **126**.

Fleming, W. (1964), 'The convergence problem for differential games II', *Contributions to the Theory of Games* **5**.

Fleming, W. (2004), 'Max-plus stochastic processes', *Applied Math. and Optim.* **48**.

Fleming, W. H. (1997), 'Deterministic nonlinear filtering', *Annali Scuola Normale Superiore Pisa, Cl. Scienze Fisiche e Matematiche, Ser. IV* **25**, 435–454.

Fleming, W. H. & Soner, H. M. (1992), *Controlled Markov Processes and Viscosity Solutions*, Springer-Verlag, New York.

Fleming, W. & McEneaney, W. (1992*a*), Risk–sensitive control with ergodic cost criteria, *in* ‘Proceedings 31st IEEE Conf. on Dec. and Control’.

Fleming, W. & McEneaney, W. (1992*b*), Risk–sensitive optimal control and differential games, *in* ‘Springer Lecture Notes in Control and Information Sciences’.

Fleming, W. & McEneaney, W. (1995), ‘Risk sensitive control on an infinite time horizon’, *SIAM J. Control and Optim.* **33**, 1881–1915.

Friedman, A. (1971), *Differential Games*, Wiley, New York.

Heise, S. & Morse, H. (2000), The DARPA JFACC program: Modeling and control of military operations, *in* ‘Proceedings of the 39th IEEE CDC Conference’, Sydney, Australia, pp. 2551–2555.

Helton, J. & James, M. (1999), Extending H_∞ control to nonlinear systems: Control of nonlinear systems to achieve performance objectives, *in* ‘SIAM’, pp. 2551–2555.

James, M. (1992), ‘Asymptotic analysis of non-linear stochastic risk-sensitive control and differential games’, *Math. Control Signals Systems* **5**, 401–417.

James, M. R. & Baras, J. S. (1996), ‘Partially observed differential games, infinite dimensional HJI equations, and nonlinear H_∞ control’, *SIAM J. Control and Optim.* **34**, 1342–1364.

James, M. & Yuliar, S. (1995), ‘A nonlinear partially observed differential game with a finite-dimensional information state’, *SIAM J. Control and Optim.* **26**, 137–145.

J.B. Cruz, M.A. Simaan, e. a. (2000), Modeling and control of military operations against adversarial control, *in* ‘Proceedings of the 39th IEEE CDC Conference’, Sydney, Australia, pp. 2581–2586.

Jelinek, J. & Godbole, D. (2000), Model predictive control of military operations, *in* ‘Proceedings of the 39th IEEE CDC Conference’, Sydney, Australia, pp. 2562–2567.

João Hespanha, Yusuf Ateskan, H. K. (2000), Deception in non-cooperative games with partial information, *in* ‘Proceedings of the 2nd DARPA-JFACC Symposium on Advances in Enterprise Control’.

McEneaney, W. (1999*a*), Robust/game–theoretic methods in filtering and estimation, *in* ‘First Symposium on Advances in Enterprise Control’, San Diego, pp. 1–9.

McEneaney, W. (1999*b*), ‘Robust/H_∞ filtering for nonlinear systems’, *Systems and Control Letters* **33**, 315–325.

McEneaney, W. (2004), ‘Some classes of imperfect information finite state-space stochastic games with finite-dimensional solutions’, *Applied Math. and Optim.* **50**, 87–118.

McEneaney, W., Lauko, I. & Fitzpatrick, B. (2004), ‘Stochastic game approach to air operations’, *IEEE Trans. on Aerospace and Electronic Systems* **40**, 1191–1216.

Olsder, G. & Papavassilopoulos, G. (1988), ‘About when to use a searchlight’, *J. of Math. Analysis and Applics.* **136**, 466–478.

P. Bernhard, A.-L. Colomb, G. P. (1987), ‘Rabbit and hunter game: Two discrete stochastic formulations’, *Comput. Math. Applic.* **13**, 205–225.

Resnick, S. I. (1998), *A Probability Path*, Birkhäuser.

Runolfsson, T. (1993), Risk–sensitive control of markov chains and differential games, *in* ‘Proceedings of the 32nd IEEE Conference on Decision and Control’.

Swarup, A. & Speyer, J. (2004), Characterization of LQG differential games with different information patterns, *in* ‘Proceedings of the 43RD IEEE CDC Conference’, Bahamas.