AFRL-IF-RS-TR-2002-244
Final Technical Report
October 2002

NORTH CAROLINA STATE UNIVERSITY TEAM (NCSU): JOINT FORCES AIR COMPONENT COMMAND (JFACC) EXPERIMENT

North Carolina State University

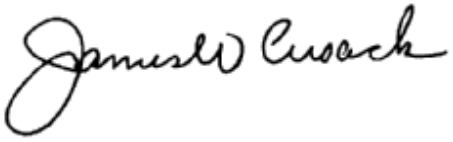
AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-244 has been reviewed and is approved for publication.

APPROVED: 

CARL A. DEFRANCO
Project Engineer

FOR THE DIRECTOR: 

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 2002	Final Aug 99 – Jun 01

4. TITLE AND SUBTITLE
NORTH CAROLINA STATE UNIVERSITY TEAM (NCSU): JOINT FORCES AIR COMPONENT COMMAND (JFACC) EXPERIMENT

5. FUNDING NUMBERS
C - F30602-99-2-0548
PE - 63760E
PR - J108
TA - 00
WU - 01

6. AUTHOR(S)
William M. McEneaney, Kazufumi Ito, Quing Zhang, Ben Fitzpatrick, and Istvan Lauko

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
North Carolina State University
Box 7514, Lower Level Leasar Hall
Raleigh North Carolina 27695

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFSA
26 Electronic Parkway
Rome New York 13441-4514

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2002-244

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Carl A. DeFranco/IFSA/(315) 330-3096/ Carl.DeFranco@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
The Command and Control (C2) problem for Military Air Operations is addressed. The problem is viewed as a stochastic game. Due to the large size of the problem, several techniques are used to decompose the problem into manageable pieces. At the outermost level, hierarchical techniques are used to solve progressively larger problems where the distributions of outcomes at one level become the dynamics of the problem at the next higher level. At the lowest level, the problem may consist of a few aircraft (or possibly packages), less than say a half-dozen, enemy SAMs, a few enemy asstes (viewed as targets from our standpoint), and some enemy decoys (assumed to mimic SAM radar signatures). At this low level, some minimal cost (to our aircraft) routes to the eventual targets are mapped out, and these are used to determine SAM sites (possibly decoys) that are unavoidable. One then employs a discrete stochastic game problem formulation to determine which of these SAMs should optimally be engaged, and by what series of aircraft operations. Since this in a game model, the optimal opponent strategy is also determined. The problem of imperfect information is addressed. The technique also allows the evaluation of various approaches in terms of expected cost and the variance of cost. One may plot these as functions of various parameters to determine when the situation is at a point where the optimal strategies may jump out suddenly.

14. SUBJECT TERMS
JFACC, Command & Control, C2 Experimentation

15. NUMBER OF PAGES
94

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

1 Overview

In the past decade, there has been a great deal of progress in the area of nonlinear games under both full and partial observations (i.e. both state feedback and measurement feedback). These advances have been motivated in part by applications to robust/H-infinity control and estimation, but have obvious application in the area of command and control due to the adversarial aspects of the battlefield. The area of control of stochastic processes is more well-developed, and also has obvious applications in command and control due to the random components of a conflict. The planned experiments will make use of these techniques with the goal of proving (or disproving) the advantages that would be obtained if these techniques were employed. Members of the NCSU team are at the forefront of these areas, and so have unique capabilities to develop such technologies and experiments.

Although it seems obvious that the modeling of the enemy activities as controlled by an intelligent, antagonistic player would lead to better command and control decisions, there are a number of mitigating factors. In particular, there are a number of simplifying assumptions and sub-optimal techniques which must be employed in order to make the problem computationally tractable. The question is then whether the advantages are still significant under these conditions. It is this team's belief that these will be evident in real-world applications as well as in simulations that reasonably model the opponent's command decisions, but this needs to be proven. (Here we are using "proven", not in the rigorous mathematical sense, but in the sense of being reasonably demonstrable via multiple simulations.)

As alluded to above, an obvious difficulty with the application of advanced techniques is the possible computational burden. One approach to the reduction of this burden is the use of hierarchical decompositions of the problem. Techniques for such decomposition in agile manufacturing applications have also undergone significant development in the last decade (and as above, members of the NCSU team have been at the forefront of this). Although these techniques have largely been developed in the context of deterministic and stochastic models, we have expanded these to the (stochastic) game context discussed above.

An important component of higher level military operations is resource (such as aircraft) allocation. Activities such as distribution and re-distribution of resources among a number of geographical regions can be naturally modeled as a discrete-time Markov decision process (MDP). A major advantage of the MDP model lies in its capability to capture events evolving in a discrete fashion. The drawback of such model, however, it its inherent large dimensionality. We attack this problem using a recent development on singular perturbations of finite-state Markov chains. In military control operations, it is common that some variables change more rapidly than other variables. This leads to time scale decomposition. The concept used in attacking the MPD problem is to make use of time scales to classify the states into several groups such that the MDP jumps more frequently within a group and less frequently among groups.

At the lowest level, some minimal cost aircraft routes to the eventual targets are mapped out. Some inverse Lyapunov techniques as well as optimization approaches are used for rapid generation of these routes. These routes are then used to determine SAM sites (possibly decoys) that are unavoidable. One then employs a discrete stochastic game problem formulation to determine which of these SAMs should optimally be engaged, and by what series of aircraft operations. Since this is a game model, the optimal opponent strategy is also determined. Assuming perfect knowledge of the state of the system, one obtains these optimal stratgies (for both sides), via dynamic programming methods. The NCSU team uses an exit cost criterion; that is, the game solution is solved all the way to the various end states. Some technical shortcuts are used to allow us to obtain this solution. Obtaining the complete solution to the problems at the lowest level in the hierarchy is superior to the rolling horizon (euphemistally referred to as the model predictive control) approach for obvious reasons. Further, it allows us to run Monte Carlo simulations with optimal controls and produce plots of the expected outcome and its variance as functions of various parameters. This alows the commander to quickly, visually see when the situation is nearing a point where the optimal strategy makes a sudden jump. (This will be shown below.)

Much of the work done on C^2 for Air Operations has assumed perfect knowledge of the state of the system (battle). However, it is well known that the "fog of war" is a major aspect of most conflicts. Consequently, the NCSU team has recently been addressing the problem of control under imperfect, and even misleading information. This involves formulation of the problem as a stochastic game under partial information. This is a problem which is at the edge of current understanding in the field of control. We make use of an approach which is optimal under limited conditions, and have shown that it leads to significantly better results than the standard techniques (such as extended Kalman filtering).

Summarizing, we apply the following techniques:

- Robust/Game-Theoretic Control (with stochastic components)
- Robust/Game-Theoretic Estimation (and the combined estimation/control problem)
- Hierarchical Decomposition Methods
- Inverse Lyapunov Techniques
- Dynamic Programming.

In Section 2, we consider our underlying "small" stochastic game problem involving only a few aircraft, a handful of SAMs and a target. The solution of the problem at this level underlies the solution of larger problems at higher levels. Consequently, a good deal of work was devoted to a solid understanding of this problem. The analysis and

results are described in detail in the various subsections. In order for the aircraft to reach their destinations, and also as a means for flagging potential threats en route which may need to be dealt with, one needs a tool for generating reasonable aircraft routes. This is discussed in section 3. Once one has solved the above small games, one needs to enlarge the view to much more substantial problems in terms of the number of entities (on both sides) involved. This is done via a hierarchical technique, and a discussion of this technique appears in Section 4. Now, all of the above assumes perfect knowledge of the system. The consequences of partial, imperfect and corrupted information are studied in Section 5. In that section, both the estimation problem, and the problem of control under imperfect information (in the presence of an adversary) are discussed. Section 6 discusses a study of filtering at a higher level in the hierarchy. Lastly, Section 7 moves the discussion to yet a higher level where a commander may be attempting to determine which control technique (of many being offered) to employ. The choice may depend on the situation, and one may even consider a switching meta-controller which chooses between different control algorithms depending on the current state.

2 Solution of Stochastic Games for C^2

This section deals with the lowest level, where the problem has been reduced to a small stochastic game involving only a few entities. (In the results to appear inthis section, there are only two aircraft, three SAMs and an enemy target; in the newer software for the imperfect information case (Section 5), we have increased this to include at least six SAMs and two targets as well as decoys. Even that problem size could easily be doubled with today's available computational power, this study was meant only as a demonstration of approach, and so no such effort was made. Note also that as indicated above, we use the hierarchical technique to deal with the full scale problem - doubling problem size at each increasing level of the hierarchy.)

The objects which will be of interest in this section are aircraft (belonging to what will be termed the "blue" player), SAMs (belonging to the "red" player), and strategic targets (belonging to red). The usage of blue and red desigantions will be assumed throughout the document.

We will reduce the state of the i^{th} (blue) aircraft at time t to a pair, $Y_i^A(t) = (D_i^A(t), X_i^A(t))$ where D_i^A will represent the health status of the aircraft, and X_i^A will represent its position. Note that since the scope of the C^2 problem is large, we will not model the dynamics of each aircraft in detail; we will not include velocity, attitude, mass and so forth as part of the state. When the problem is decomposed into separate subproblems below, we will abuse notation in the sense that in one subproblem, X_i^A will represent a position taking continuous values in $\mathbf{R}^2$, while for the other subproblem this will indicate position among a discrete set of alternatives; the meaning will be completely obvious by context. We will suppose that the health status take values in the discrete set

$\{1,2,3,4\}$ where 1 represents healthy, 2 and 3 represent various levels of damage (or need of maintenance), and 4 indicates that the aircraft has been destroyed.

We will assume similar state models for the SAMs. The i^{th} SAM state will be represented by the pair $Y_i^R(t) = (D_i^R(t), X_i^R(t))$ where D_i^R will represent the health status of the SAM, and X_i^R will represent its position. (Note that there exist both mobile and fixed-site SAMs.) Similar comments as those above can be made with regard to X_i^R. As for the health status of the SAMs, we let D_i^R take values in $\{1,2,3\}$ where 1 represents healthy, 2 represents damage (or need of maintenance), and 3 indicates that the SAM has been destroyed (not repairable). Lastly, we will take a similar model for the strategic targets, where the pair will be denoted as $Y_i^T(t) = (D_i^T(t), X_i^T(t))$ with $D_i^T(t) \in \{1,2,3\}$, where 1, 2 and 3 will have the same meaning as for the SAMs. Let the number of blue aircraft be N_A, the number of red SAMs be N_R, and the number of red strategic targets be N_T. Let $\vec{Y}^A = \{Y_i^A\}_{i=1}^{N_A}$, $\vec{Y}^R = \{Y_i^R\}_{i=1}^{N_R}$ and $\vec{Y}^T = \{Y_i^T\}_{i=1}^{N_T}$. Throughout, we will use the convention of uppercase letters for the state processes and lowercase for values that the state process may take on, that is, $\vec{Y}^A(t) = \vec{y}^A$ indicates that the aircraft state process has the value $\vec{y}^A$ at time t.

The objective is not clearly defined in a mathematical sense. For blue, it may sometimes be to destroy a strategic target while minimizing damage to the aircraft; in other situations it may be more general attrition of both SAMs and targets. In order to simplify matters, we will assume here that both players are using the same objective function. That is, blue is trying to minimize the worst case (maximum) payoff, and red is trying to maximize their worst case (minimum) of the same payoff. The time-horizon over which these objectives should be met is not fixed. We choose to consider an exit cost, without running cost terms. Let τ be the exit time. We define the exit time to be the time when either: 1) all the blue aircraft have been destroyed or 2) the red strategic target(s) has(have) been destroyed and the surviving blue aircraft have returned to base. We let the set of states satisfying one of the exit conditions be denoted by $\mathcal{E}$. In order to capture the objective in a reasonably simple payoff function, one can consider, for instance, a linear payoff with parameters which can be varied depending on the value of the assets such as

$$\Psi(\vec{y}^A, \vec{y}^R, \vec{y}^T) \doteq \mu_A\Big[\sum_{i=1}^{N_A} d_i^A\Big] - \mu_R\Big[\sum_{i=1}^{N_R} d_i^R\Big] - \mu_T\Big[\sum_{i=1}^{N_T} d_i^T\Big] \tag{1}$$

where μ_A, μ_R, μ_T are the parameters. The presence of the expectation in the above equation is due to the fact that the dynamics of the health status of the objects will involve random outcomes of engagements and maintenance.

The next two subsections describe the mathematics behind the algorithm. The reader interested primarily in the application of the algorithm, and its advantages should proceed to Subsection 2.3.

2.1 Discrete Stochastic Game

We consider the problem where a single strategic target is selected, and an approximate path from the blue base to that target has been generated. As discussed above, there may be one or more SAM sites intervening along this path. At this level, the positional dynamics will be specified only in a general way. Let the SAMs be indexed as $\{1, 2, 3, ..., N_R\}$. Let the aircraft position take values in the set $\mathcal{L} \doteq \{B, 1, 2, 3, ..., N_R, N_R + 1\}$ where B indicates the (blue) base and $N_R + 1$ indicates the (red) strategic target. We suppose a discrete time model where each time step occurs only when either an aircraft engages a SAM, an aircraft engages the target, or an aircraft returns to base. More than one such activity can occur at each step. The aircraft control for each aircraft, $U_i^A(t)$, must be specified at each time step. The set of possible values is $\mathcal{U} = \mathcal{L} \cup \{0\}$ where numbers between $U_i^A = 1$ and $U_i^A = N_R + 1$ indicate attack the corresponding red SAM or target, $U_i^A = B$ indicates return to base, and $U_i^A = 0$ indicates "do nothing". Note that the dynamics of the motion is simply $X_i^A(t + 1) = U_i^A(t)$ when $U_i^A(t) \neq 0$ and $X_i^A(t + 1) = X_i^A(t)$ when $U_i^A(t) = 0$. We place some restrictions on the allowable controls. The control actions will be organized into cycles of length, n_c. That is, each cycle will consist of n_c time steps. At the start of each cycle, all aircraft must be at the base. Consequently, we require $U_i^A(t) = B$ for all $i \leq N_A$ and all $t = kn_c - 1$ for all $k \geq 1$. We also require that for any $t = kn_c - 1$

if there exists i such that $X_i^A(t) = B$ and $D_i^A(t) = 1$, then there must be a $k \leq N_A$ with $D_k^A(t) \neq 4$ such that $U_k^A(t) \neq B$.

(CC)

Note that this last requirement forces at least some aircraft to engage red during each cycle for which there is a fully healthy aircraft.

It will be assumed for this subproblem that the SAMs cannot move during the duration of the game. The controls for the i^{th} SAM at (discrete) time t is $G_i^R(t)$, taking values in $\{0, 1\}$ where 0 indicates radar on and 1 indicates radar off. As mentioned in the introduction, when the radar is on, the probability of the SAM inflicting damage on the aircraft rises - as does the probabilty that the aircraft can inflict damage on the SAM.

The health status of each of the objects will transition according to a discrete–time Markov chain model. The transisiton probabilities will be state/control dependent. To simplify matters, assume that multiple aircraft can attack a single SAM, but that the aircraft need only engage one SAM at a time. At each time-step, where a SAM is under attack, we let the transition probabilty be given by the matrices P^{R01}, P^{R02}, P^{R11} and P^{R12} indicating the transition probabilities for the cases where a SAM with radar off is being attacked by a single aircraft, a SAM with radar off is being attacked by multiple aircraft simultaneously, a SAM with radar on is being attacked by a single aircraft, and a SAM with radar on is being attacked by multiple aircraft simultaneously, respectively. Of course, there are many more possibilities, but we consider only these for simplicity. If a SAM radar is on, and the SAM is not under attack during that time step, then we

assume the SAM health status remains constant with probability one. Lastly, if a SAM site is off and not under attack, the health may improve through maintenance, with a tranistion probability given by P^{R00}. The state $d_i^R = 3$ will be an absorbing state for all the transistion matrices. In particular, maintenance cannot repair a SAM once it has entered state 3. The transition probabilities for the red target are the same as those for a SAM with radar off.

Let the corresponding probabilities for the aircraft during engagement be given by P^{A01}, P^{A02}, P^{A11} and P^{A12} where these stand for the same situations as those indicated for the SAMs above. We assume that the probability of transitioning to state 4 (down) is nonzero for all of the above matrices (i.e. that the last columns have no zero entries). We also allow the aircraft to undergo maintenance while at the base ($U_i^A(t) = B$), and let the transition probabilities be P^{A00}. For the aircraft, one must also consider the possibility of damage due to flying over SAMs with radars that are on while enroute from one point to the next. For instance, if $X_i^A(t) = 1$ and $X_i^A(t+1) = U_i^A(t) = 3$, and if SAMs 2 and 4 are between 1 and 3, then aircraft i could suffer damage while flying over each of the SAMs 2 and 4 – if they are on. We let the transition probability for aircraft health due to flying over a SAM that is on ($G_j^R(t) = 1$) and not destroyed ($D_j^R(t) \neq 3$) be P^{A1F} for each SAM that is flown over. In the above example, if SAM 3 is on and aircraft i is the only one attacking, then its transition probability for this time step is given by $P^{A1F}P^{A1F}P^{A11}$. Lastly, the destroyed/down state will be absorbing for all the transition matrices including P^{A00}.

Here we will consider a simplified information pattern that is chosen to mimic the real world situation in a rather loose way. Specifically, we will consider the game where at each time step blue chooses its control given the current state, and then red chooses its control given the current state *plus* the control choice for blue at the current time. In other words, we are interested here in an upper value (recall blue is minimizing and red maximizing). Let the value function for this game be denoted by $V(\vec{y}^A, \vec{y}^R, \vec{y}^T)$. Since it is quite standard, we do not include a proof of the DPE (dynamic programming equation) which is given as follows.

Theorem 2.1 *The value function satisfies*

$$\begin{aligned} V(\vec{y}^A, \vec{y}^R, \vec{y}^T) = \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \Bigg[\\ E\Big\{ V(\vec{Y}^A(1), \vec{Y}^R(1), \vec{Y}^T(1)) \\ \Big| \ \vec{Y}^A(0) = \vec{y}^A, \vec{Y}^R(0) = \vec{y}^R, \vec{Y}^T(0) = \vec{y}^T, \Big\} \Bigg] \\ \doteq \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \mathcal{G}^{\vec{u}^A, \vec{g}^R}[V](\vec{y}^A, \vec{y}^R, \vec{y}^T). \end{aligned}$$

We will now indicate how this value function can be obtained through repeated appplication of the backward dynamic programming operator. First however, we will need the

following lemma which essentially implies that there is a positive probability of reaching the absorbing states in a fixed number of steps.

Lemma 2.2 *There exists $n < \infty$ and $\delta > 0$ such that for any sequence of controls for blue and red*

$$P\Big[(\vec{Y}^A(t+n), \vec{Y}^R(t+n), \vec{Y}^T(t+n)) \in \mathcal{E} \\ \Big| (\vec{Y}^A(t), \vec{Y}^R(t), \vec{Y}^T(t)) = (\vec{y}^A, \vec{y}^R, \vec{y}^T) \Big] \geq \delta$$

for any $(\vec{y}^A, \vec{y}^R, \vec{y}^T)$ where we recall $\mathcal{E}$ was the exit set.

PROOF. Let $t_1 = \min\{s > t : s = kn_c + 1 \text{ for some nonnegative integer} k\}$. Then, by condition (CC), there exists $i_1 \leq N_A$ such that $X^A_{i_1}(t_1) \neq B$, and consequently, there exists $\delta_1 > 0$ (dependent on the choice of transition matrices) such that $P(D^A_{i_1}(t_1) = 4) \geq \delta_1$. Let $\Omega_1 \subseteq \Omega$ (the sampls space) be given by $\Omega_1 = \{\omega \in \Omega : D^A_{i_1}(t_1) = 4\}$. For points in Ω_1 such that $(\vec{Y}^A(t_1), \vec{Y}^R(t_1), \vec{Y}^T(t_1)) \notin \mathcal{E}$ (where not all the aircraft are down), let $t_2 = \min\{s > t_1 : s = kn_c + 1 \text{ for some nonnegative integer } k\}$. Then again by condition (CC), there exists $i_2 \leq N_A$ such that $X^A_{i_2}(t_2) \neq B$, and consequently, $P(\{\omega \in \Omega_1 : D^A_{i_2}(t_2) = 4\}) \geq \delta_1$. Since state 4 is absorbing, This implies that $P(D^A_{i_1}(t) = 4,\, D^A_{i_2}(t) = 4) \geq \delta_1^2$ for all $t \geq t_2$. Proceeding inductively, one finds that by, at most, time $t + n_c N_A$, the state is in E with probability no less than $\delta_1^{N_A}$.

Define the backward dynamic programming (DP) algortihm as follows. Let the terminal value be

$$W(0, \vec{y}^A, \vec{y}^R, \vec{y}^T) = \begin{cases} \Psi(\vec{y}^A, \vec{y}^R, \vec{y}^T) & \text{if } (\vec{y}^A, \vec{y}^R, \vec{y}^T) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

(We remark that the choice of 0 is irrelevant.) Given $W(k, \cdot)$, one computes $W(k-1, \cdot)$ by the backward dynamic programming operator given by

$$W(k-1, \vec{y}^A, \vec{y}^R, \vec{y}^T) = \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \Big[\\ \mathrm{E}\Big\{ W(k, \vec{Y}^A(1), \vec{Y}^R(1), \vec{Y}^T(1)) \\ \Big| \vec{Y}^A(0) = \vec{y}^A, \vec{Y}^R(0) = \vec{y}^R, \vec{Y}^T(0) = \vec{y}^T, \Big\} \Big] \\ \doteq \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \mathcal{G}^{\vec{u}^A, \vec{g}^R}[W(k, \cdot)](\vec{y}^A, \vec{y}^R, \vec{y}^T)$$

if $(\vec{y}^A, \vec{y}^R, \vec{y}^T) \notin \mathcal{E}$ and $W(k-1, \vec{y}^A, \vec{y}^R, \vec{y}^T) = \Psi(\vec{y}^A, \vec{y}^R, \vec{y}^T)$ otherwise.

Lemma 2.3 *This backward dynamic programming propagation operator is a contraction.*

PROOF. Once one has Lemma 2.2, the proof of this lemma is a minor variation of standard results, but in this case for a game with an exit criterion. (See, for instance, [4], [15] for similar results.) We will simply indicate some of the main points. Let W_1 and W_2 be given by the backward DP with possibly different conditions at $k = 0$. For simplicity, use the notation $\vec{y} \doteq (\vec{y}^A, \vec{y}^R, \vec{y}^T)$. Note that (for $k < 0$)

$$\begin{aligned} W_1(k,\vec{y}) - & W_2(k,\vec{y}) \\ &= \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \mathcal{G}^{\vec{u}^A, \vec{g}^R}[W_1(k+1,\cdot)](\vec{y}) \\ &- \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \mathcal{G}^{\vec{u}^A, \vec{g}^R}[W_2(k+1,\cdot)](\vec{y}). \end{aligned}$$

Choose u_1^A to be $\frac{\varepsilon}{n}$-optimal for W_1 and then choose g_1^R to be $\frac{\varepsilon}{n}$-optimal for W_2 given the same control u_1^A as used for W_1. Then

$$\begin{aligned} W_1(k,\vec{y}) - & W_2(k,\vec{y}) \\ &\leq \mathcal{G}^{\vec{u}_1^A, \vec{g}_1^R}[W_1(k+1,\cdot) - W_2(k+1,\cdot)](\vec{y}) + \frac{2\varepsilon}{n}. \end{aligned}$$

Repeating this process, one finds that (for $k < -n$) and proper choice of feedback controls,

$$\begin{aligned} W_1(k,\vec{y}) - & W_2(k,\vec{y}) \\ &\leq \prod_{m=1}^{n} \{\mathcal{G}^{\vec{u}_m^A, \vec{g}_m^R}\}[W_1(k+n,\cdot) - W_2(k+n,\cdot)](\vec{y}) + 2\varepsilon \end{aligned}$$

where we are using the $\prod$ notation to indicate operator composition. Alternatively, one may write this as

$$\begin{aligned} W_1(k,\vec{y}) - W_2(k,\vec{y}) \leq \sum_{\vec{z} \notin \mathcal{E}} \Big\{ & [W_1(k+n,\vec{z}) \\ & - W_2(k+n,\vec{z})] \cdot P^n_{\vec{y},\vec{z}}(\{\vec{u}_m^A\}_{i=1}^n, \{\vec{g}_m^R\}_{i=1}^n) \Big\} + 2\varepsilon \end{aligned}$$

where this last term indicates the probability of transitioning from $\vec{y}$ to $\vec{z}$ in n steps given the feedback control processes specified in the arguments. Using symmetry and the Lemma 2.2, one obtains

$$\begin{aligned} |W_1(k,\vec{y}) - & W_2(k,\vec{y})| \\ &\leq \max_{\vec{z}} \{|W_1(k+n,\vec{z}) - W_2(k+n,\vec{z})|\} \\ &\qquad \sum_{\vec{z} \notin \mathcal{E}} P^n_{\vec{y},\vec{z}}(\{\vec{u}_m^A\}_{i=1}^n, \{\vec{g}_m^R\}_{i=1}^n) + 2\varepsilon \\ &\leq \max\{|W_1(k+n,\vec{z}) - W_2(k+n,\vec{z})|\}(1-\delta) + 2\varepsilon. \end{aligned}$$

This then yields

$$\begin{aligned} \|W_1(k,\cdot) - & W_2(k,\cdot)\|_\infty \\ &\leq (1-\delta)\|W_1(k+n,\cdot) - W_2(k+n,\cdot)\|_\infty. \end{aligned}$$

The proof of convergence is also standard, and so we state the result without proof.

Theorem 2.4 *$W(k, \vec{y}^A, \vec{y}^R, \vec{y}^T)$ converges to the value function $V(\vec{y}^A, \vec{y}^R, \vec{y}^T)$ as $k \downarrow -\infty$ for all points in the state space.*

We remark that since the controls spaces are finite, the controls actually converge in a finite number of steps.

2.2 Reducing the Computations

The above algorithm for the computation of the value function (and corresponding control policies) suffers from the curse of dimensionality typical for DP algorithms. Specifically, notice that computation of $\mathcal{G}^{\vec{u}^A,\vec{g}^R}[W(k,\cdot)](\vec{y})$ may require summing the product of $W(k,\vec{z})$ and $P^{\vec{u}^A,\vec{g}^R}_{\vec{y},\vec{z}}$ over all possible values of $\vec{z}$ for each point $\vec{y}$. More specifically, the computations for $W(k-1,\vec{y})$ (for each $\vec{y}$) require $O(4^{N_A}(N_R+N_T+1)^{N_A}3^{N_R}3^{N_T})$ operations, even without optimization over blue and red control policies. We will discuss one of the methods being used to reduce these computation costs. The method will involve an approximation of W at each step. The result will be that the computational costs *per $\vec{y}$ point* will be reduced from the above exponential growth in the number of dimensions to only linear growth in the number of dimensions. This is a tremendous reduction in computational costs which makes the difference between feasibility and infeasibility of computation for low-dimensional problems. The growth in the number of points at which we must evaluate W remains exponential in the number of dimensions of course.

We introduce the following operator which is essentially an approximation operator for the value function or DP iterates around any given point $\vec{y}$. In order to reduce the notation, we will consider a simplified state space where $\vec{y} = (y_1, y_2, y_3)$ with $y_1 \in \{1,2,3,4\}$ and $y_2, y_3 \in \{1,2,3\}$. This will reduce notation without losing the flavor of the method. Define the matrices A^i for $i = 1,2,3$ given by $A^i_{j,k} = 1$ if $j = k = i$ and $A^i_{j,k} = 0$ otherwise. Then, given $\vec{y}$, define the approximation operator for approximation around $\vec{y}$ by

$$\mathcal{H}_{\vec{y}}[V(\cdot)](\vec{z}) \doteq \begin{cases} \left[\frac{1}{\sum_{i=1}^3 |z_i - y_i|}\right] \cdot \sum_{i=1}^3 [|z_i - y_i| \\ \qquad V(A_i(\vec{z}-\vec{y})+\vec{y})] & \text{if } \vec{z} \neq \vec{y} \\ \\ V(\vec{y}) & \text{if } \vec{z} = \vec{y}. \end{cases}$$

The operator is essentially an approximation operator where convex combinations are used to approximate V for states which are not directly along a basis direction from the point around which V is being approximated. Although we will not discuss the error analysis here, we note that of course the appropriateness of an approximator of this form depends critically on the nature of the value function itself which, in turn, depends on

the choice of terminal payoff, Ψ. Recall that since the problem is rather loosely defined, we have great freedom in the choice of Ψ. Now, note that the approximation operator is a nonexpansive map for any $\vec{y}$. The backward DP operator of the previous section will now be replaced by the approximate backward DP operator given by

$$W(k-1,\vec{y}) \doteq \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \mathcal{G}^{\vec{u}^A,\vec{g}^R}[\mathcal{H}_{\vec{y}}[W(k,\cdot)](\cdot)](\vec{y})$$

if $\vec{y} \notin \mathcal{E}$ and $W(k-1,\vec{y}) = \Psi(\vec{y})$ otherwise. Using the nonexpansivity of this approximation operator and the contraction property of the backward DP, one can obtain the following result in a straightforward manner similar to that of the previous section.

Theorem 2.5 *The approximate backward DP operator is a contraction, and the corresponding iterates converge to a fixed point of the operator.*

Lastly, we indicate the promised reduction in computation via the approximation. Recall that each of the transitions is independent. Suppose for this simplified problem that the transition matrices for y_1, y_3, y_3 are given by P^1, P^2, P^3 respectively, where we are suppressing the dependence of each P on the states and controls. (Note that in this simplified problem, we have actually eliminated that position state for the aircraft.) Then the approximate backward DP takes the form

$$\begin{aligned} & W(k-1,\vec{y}) \qquad (2) \\ & \doteq \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} W(k,\vec{y}) P^1_{y_1,y_1} P^2_{y_2,y_2} P^3_{y_3,y_3} \\ & \quad + \sum_{z_1=1}^{4} W(k,z_1,y_2,y_3) Q^1_{|z_1-y_1|} P^1_{y_1,z_1} \\ & \quad + \sum_{z_2=1}^{3} W(k,y_1,z_2,y_3) Q^2_{|z_2-y_2|} P^2_{y_2,z_2} \\ & \quad + \sum_{z_3=1}^{3} W(k,y_1,y_2,z_3) Q^3_{|z_3-y_3|} P^3_{y_3,z_3} \qquad (3) \end{aligned}$$

where

$$Q^1_{|z_1-y_1|} \doteq \frac{\sum_{z_2=1}^{3} |z_2-y_2| P^2_{y_2,z_2} + \sum_{z_3=1}^{3} |z_3-y_3| P^3_{y_3,z_3}}{\sum_{i=1}^{3} |z_i-y_i|}$$

with analogous definitions for $Q^2_{|z_2-y_2|}$ and $Q^3_{|z_3-y_3|}$. Note that these Q^i may be precomputed. Thus the approximate DP (3) has only linear growth in the compuations which must be performed at each step (per point in the state space).

2.3 Testbed, Monte Carlo Simulation and Landscape Plots

The games controller was tested via Monte Carlo simulation. The first purpose was to ensure that the game controller was bug-free (it wasn't). The second was to explore the structure of the results, and their dependencies on system parameters.

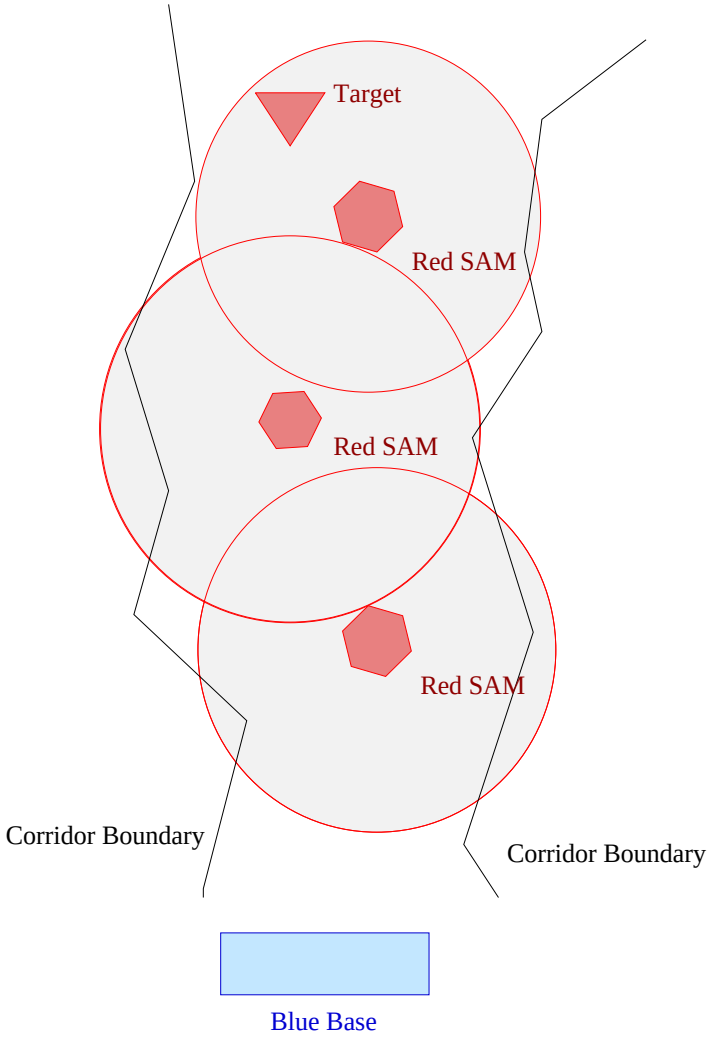
For the first set of tests, a simple geometry was considered. This is depicted in figure 1. A corridor has been determined. The corridor is such that one must pass within the umbrella of each SAM to get to the next, and finally to the target. The SAMs are numbered 1 to 3 from bottom to top. When there are less than three SAMs in some of the tests, that may correspond to any of the SAMs in Figure 1a being missing. (The problem is mathematically independent of which specific ones are missing in this geometry.)

A number of bugs were found and removed. Most notably, a problem where the controller was not bookkeeping the number of aircraft acting in tandem properly was corrected.

It was soon noticed that the most significant feature of the controllers was the choice for blue of whether to fly-over the SAMs without attacking or to perform a rollback policy (removing the first, then the second, then the third before the target). Intermediate policies occurred only for a small range of cases.

Based on feedback from the program office, it was decided to use the Monte Carlo simulator to look at dependency on certain parameters, and the effect of mismodeling of those parameters. This study was undertaken with a software package which was referred to as the Sensitivity Tool. It varied both actual parameters in the simulation and the corresponding values assumed by the controller. For each such possibility, a Monte Carlo series was run (generally with approximately 1000-2000 sample games). The results were plotted in three-dimensional figures where the horizontal axes corresponded to true and assumed values of various parameters. The Sensitivity Tool and the embedded Monte Carlo simulator are depicted in Figures 1b and 1c.

It was found that the choice of fly-over or rollback was sensitive to the probability of damage to a blue aircraft when flying through the SAM umbrella without engaging. Let this probability be denoted by α. (More specifically, the transition probabilities for the aircraft are 4×4 matrices, and the $(1,4),(2,4),(3,4)$ entries are α, the $(1,1),(2,2),(3,3)$ entries are $1-\alpha$, the $(4,4)$ is 1, and the rest are zero.) Consequently, Monte Carlo runs were done where the true value of α and the value the controllers believe to be α were varied. For most of the runs, both blue and red had the same value of α. This was done for computational reasons. A small number of runs were done where red had the true value, and so only blue had an incorrect value. These verified that the main structure we will see is due to the blue controller. (The runs where red had the true value will be discussed later.)



Geography 1 Distillation

Fig. 1a.

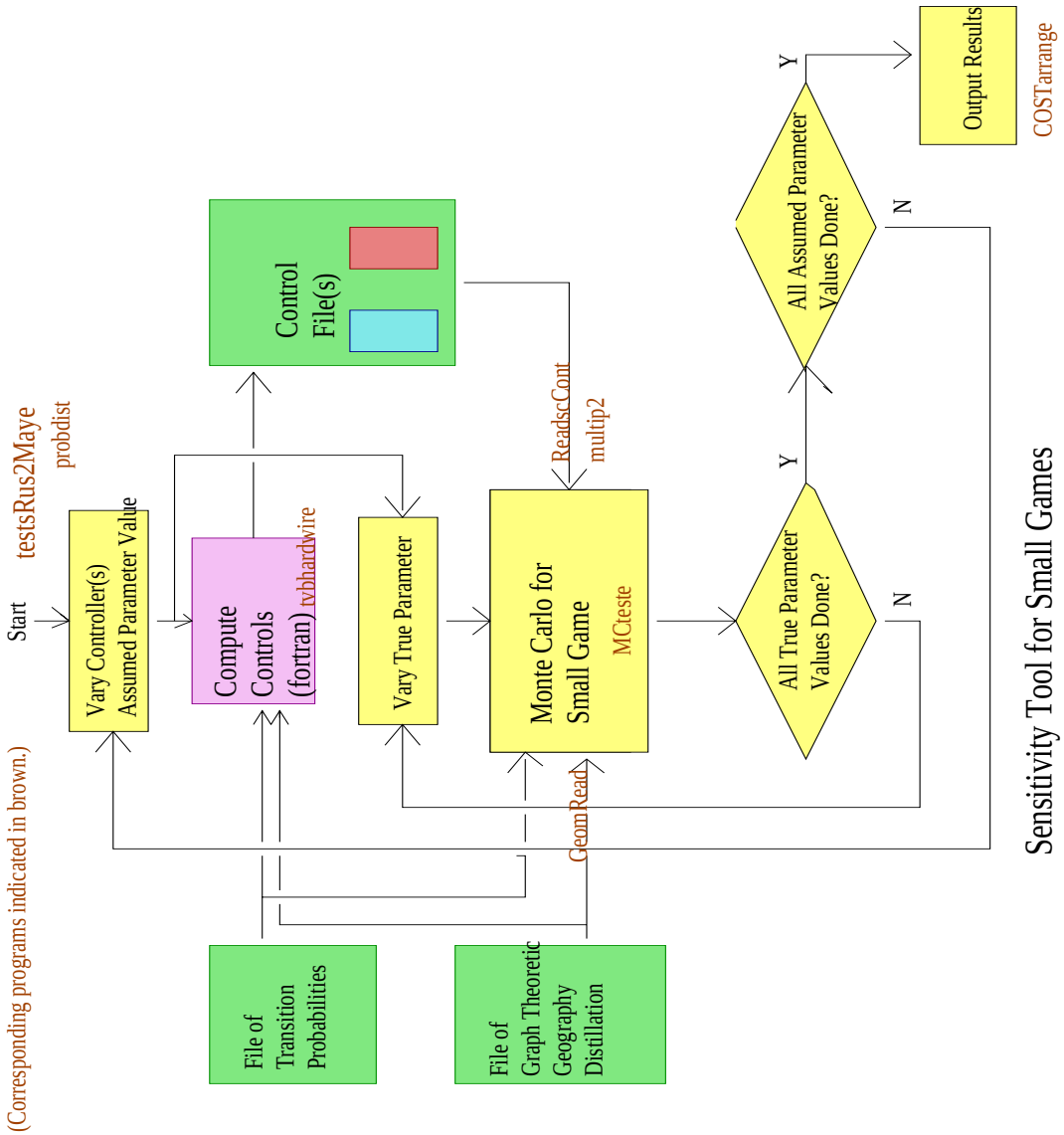

Sensitivity Tool for Small Games
(Corresponding programs indicated in brown.)
Start
testsRus2Maye
probdist
Vary Controller(s) Assumed Parameter Value
Compute Controls (fortran)
rvbhardwire
Control File(s)
Vary True Parameter
ReadscCont
multip2
Monte Carlo for Small Game
MCteste
All True Parameter Values Done?
All Assumed Parameter Values Done?
Y
N
Y
N
Output Results
COSTarrange
GeomRead
File of Transition Probabilities
File of Graph Theoretic Geography Distillation

Fig. 1b.

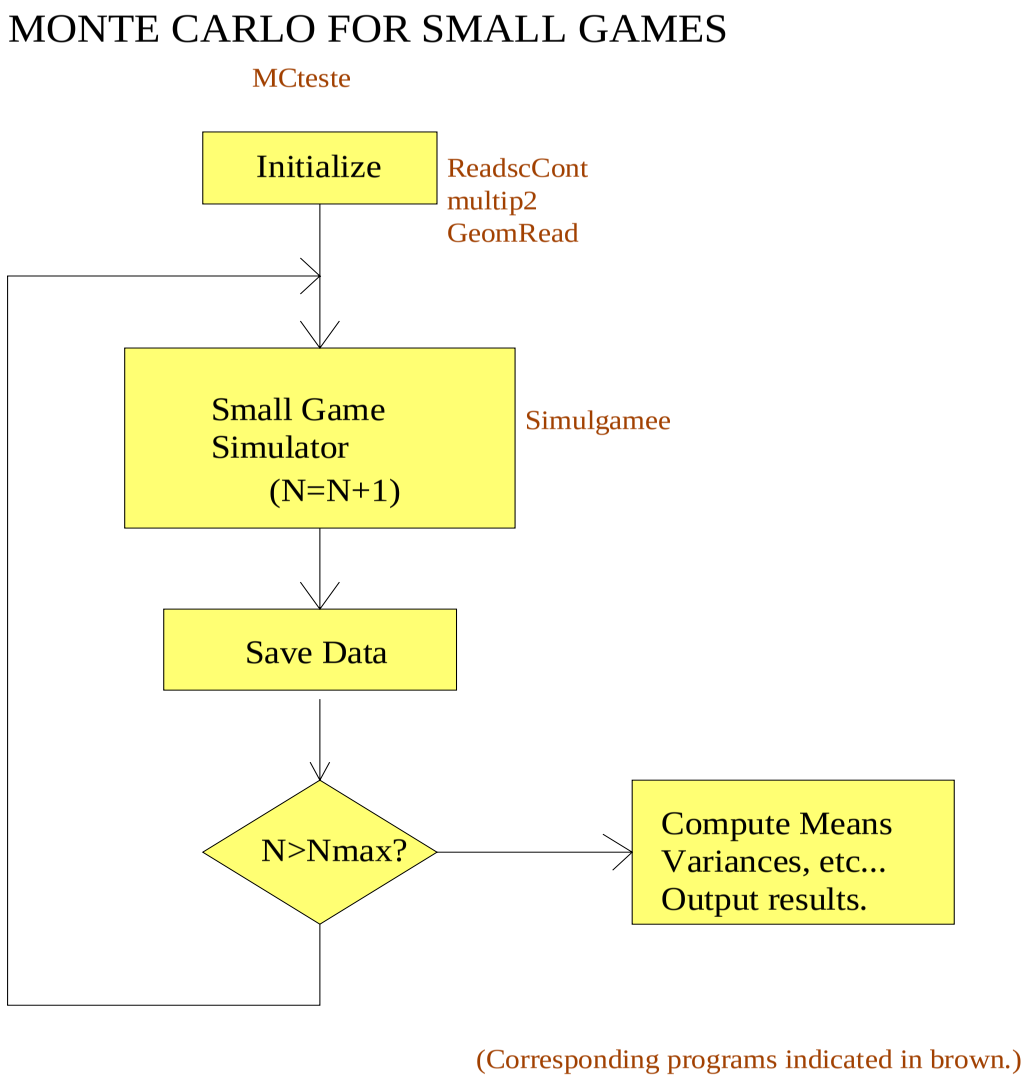

MONTE CARLO FOR SMALL GAMES
MCteste
Initialize
ReadscCont
multip2
GeomRead
Small Game
Simulator
(N=N+1)
Simulgamee
Save Data
N>Nmax?
Compute Means
Variances, etc...
Output results.
(Corresponding programs indicated in brown.)

Fig. 1c.

The Monte Carlo data to follow was made with the following parameter values: For notation, see the "pmrmodel" documentation (in ps or pdf format).

$\mu^A = 20,\ \mu^R = 3,\ \mu^T = 20$

$pda0 = [1, 0, 0, 0;\ 0.4, 0.6, 0, 0;\ 0.3, 0, 0.7, 0;\ 0, 0, 0, 1]$

$pda011 = [0.99, 0.00, 0.00, 0.01;\ 0.0, 0.95, 0.025, 0.025;\ 0.0, 0.0, 0.9, 0.1;\ 0.0, 0.0, 0.0, 1.0]$

$pda021 = [1.0, 0.0, 0.0, 0.00;\ 0.00, 0.98, 0.01, 0.01;\ 0.00, 0.00, 0.96, 0.04;\ 0.00, 0.00, 0.00, 1.00]$

$pda111 = [0.82, 0.01, 0.02, 0.15;\ 0.00, 0.82, 0.03, 0.15;\ 0.00, 0.00, 0.75, 0.25;\ 0.00, 0.00, 0.00, 1.00]$

$pda121 = [0.92, 0.01, 0.02, 0.05;\ 0.00, 0.92, 0.03, 0.05;\ 0.00, 0.00, 0.87, 0.13;\ 0.0, 0.0, 0.0, 1.0]$

$pdadr3 = [0.995, 0.000, 0.00, 0.005;\ 0.00, 0.98, 0.01, 0.01;\ 0.00, 0.00, 0.97, 0.03;\ 0.0, 0.0, 0.0, 1.0]$

$pdfoon =$ (see above)

$pdr0 = [1, 0, 0;\ 0.3, 0.7, 0;\ 0, 0, 1]$

$pdr011 = [0.7, 0.0, 0.3;\ 0.0, 0.6, 0.4;\ 0.0, 0.0, 1.0]$

$pdr021 = [0.4, 0.0, 0.6;\ 0.0, 0.35, 0.65;\ 0.0, 0.0, 1.0]$

$pdr111 = [0.7, 0.0, 0.3;\ 0.0, 0.5, 0.5;\ 0.0, 0.0, 1.0]$

$pdr121 = [0.3, 0.0, 0.7;\ 0.0, 0.25, 0.75;\ 0.0, 0.0, 1.0]$

For the purposes of reader understanding, the output value has been multiplied by -1 and had a constant added so that it takes the form

$$V = \mu^A \sum_i (3 - D_i^A) - \mu^R \sum_i (D_i^R - 2) - \mu^T (D^T - 2).$$

In this case, the minimum value is 0 (rout for blue), and the maximum value is $3\mu^A n_A + 2\mu^R n_R + 2\mu^T$ where n_A and n_R are the numbers of aircraft and SAMs. Note that in this case, larger numbers are better for blue and vice–versa.

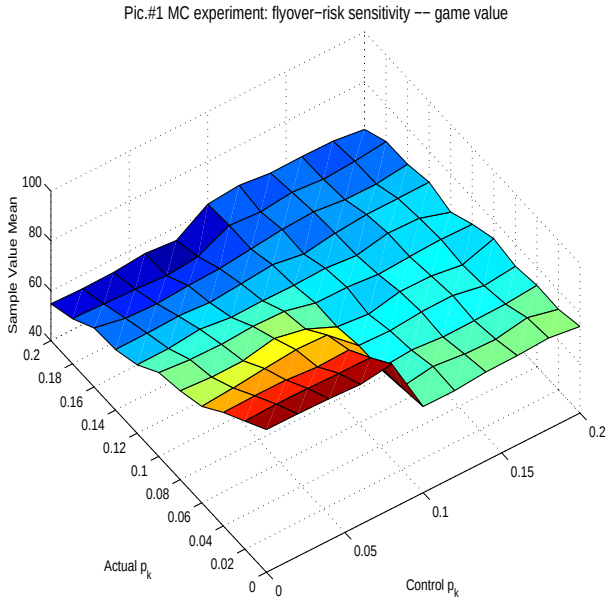


Fig. 2.3.1

All runs are made with 2000 sample points unless otherwise specified. Each set of figures takes from 2-16 hours to produce on a typical workstation. The controllers are computed 11 times (for each α), and 2000 simulation runs are made for each of the 121 points on the graph.

The first set of data is for the case of one aircraft and one SAM site.

- Figure 2.3.1 is the sample mean value.
- Figure 2.3.2 is the sample standard deviation.
- Figure 2.3.3 is the sample mean number of surviving a/c
- Figure 2.3.4 is the sample mean number of surviving SAMs
- Figure 2.3.5 is the sample mean number of surviving targets
- Figure 2.3.6 is the sample mean number of cycles until the end of the game.

(Recall that the game ends when all the blue a/c are down, or the red target is destroyed.)

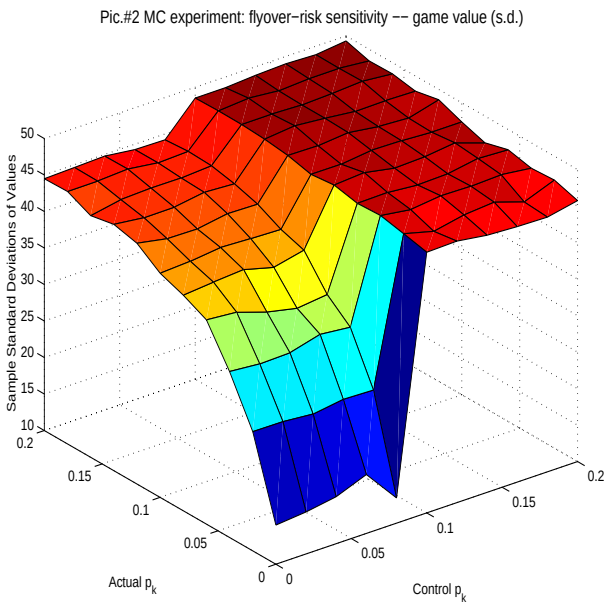


Fig. 2.3.2

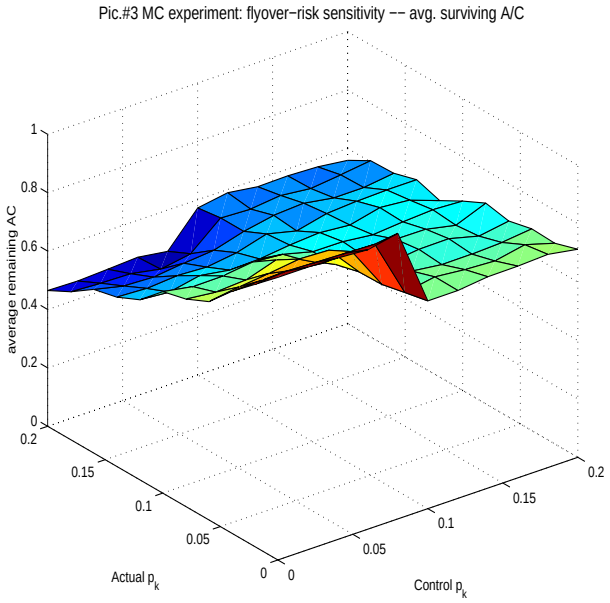


Fig. 2.3.3

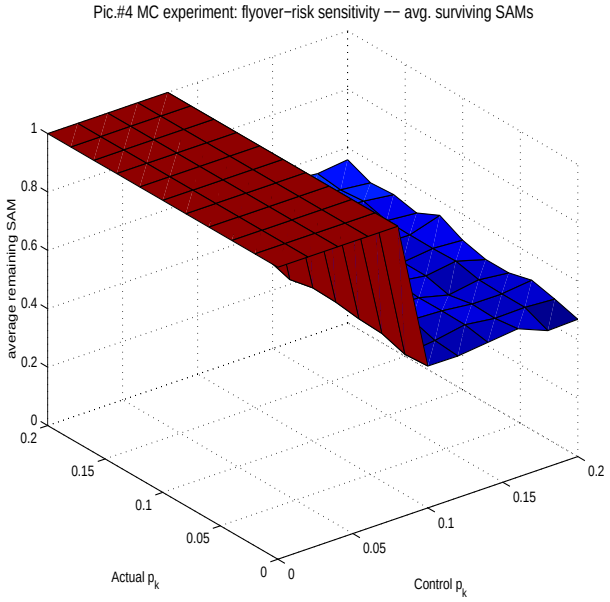


Fig. 2.3.4

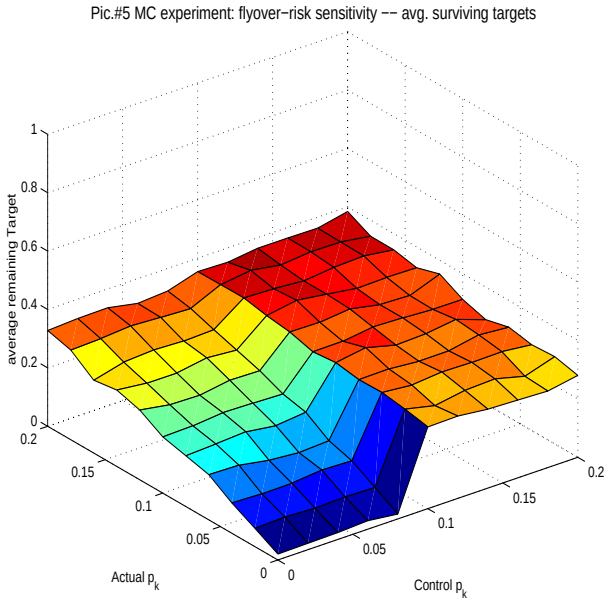


Fig. 2.3.5

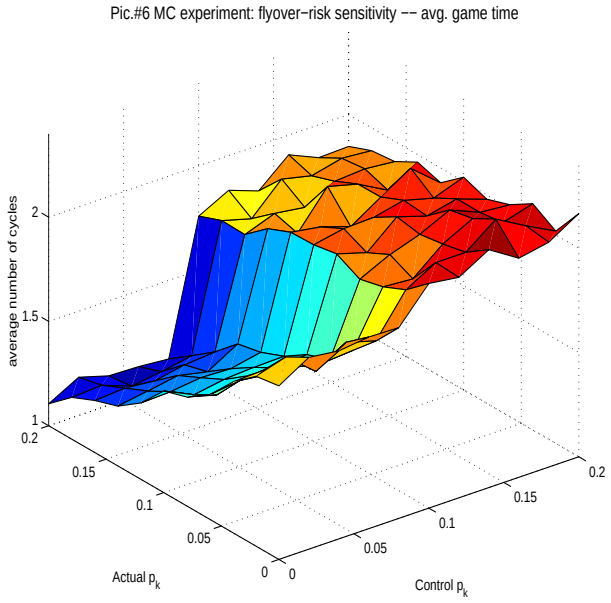


Fig. 2.3.6

Note:

1. The value has two "regions". In the left region it is roughly linear (over the region), and in the right region it is constant.
2. The value is monotonically decreasing as a function of the true α.
3. For each line of constant true α, the value takes on its maximum at the same value of control α. (The blue controller effect dominates.)
4. The right side corresponds to rollback, the left to fly-over.
5. The standard deviation for the rollback is higher than that of the fly-over even when the mean value is lower.
6. The average number of surviving units is constant on the right region.

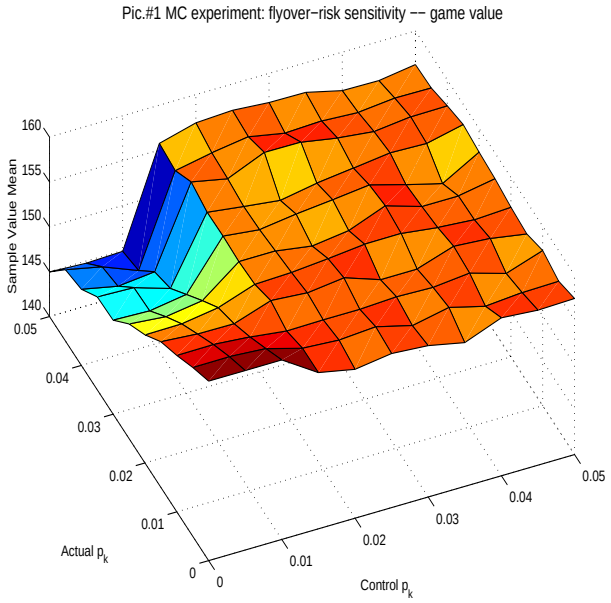


Fig. 2.3.7

The second set of data is for the case of two aircraft and one SAM site.

- Figure 2.3.7 is the sample mean value.
- Figure 2.3.8 is the sample standard deviation.
- Figure 2.3.9 is the sample mean number of surviving a/c
- Figure 2.3.10 is the sample mean number of surviving SAMs
- Figure 2.3.11 is the sample mean number of surviving targets
- Figure 2.3.12 is the sample mean number of cycles until the end of the game.

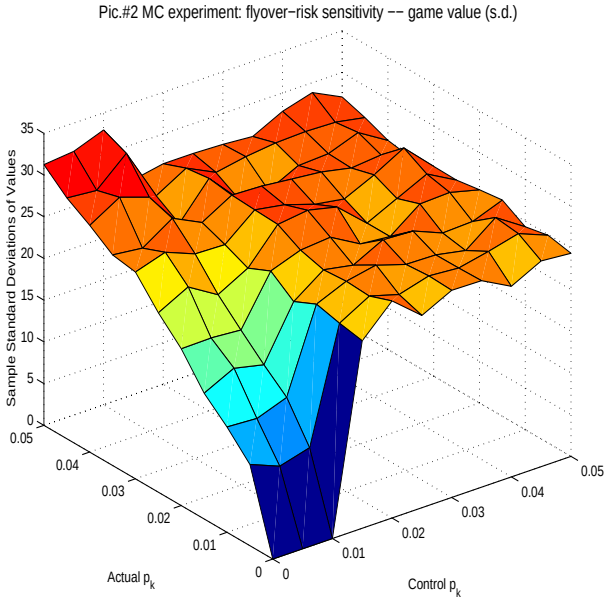


Fig. 2.3.8

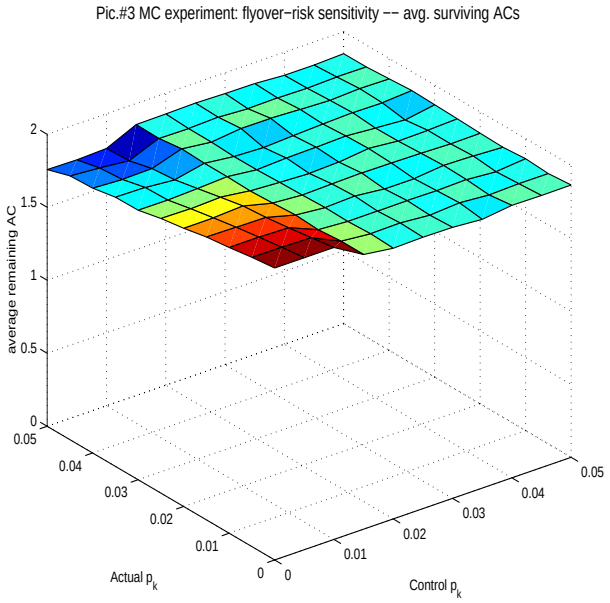


Fig. 2.3.9

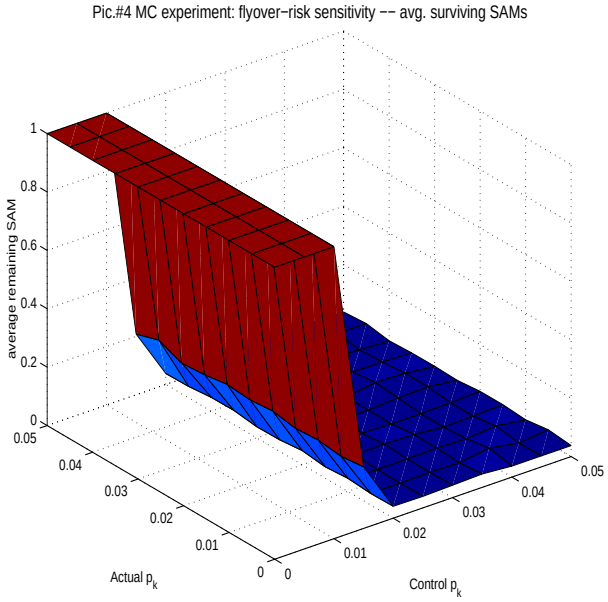


Fig. 2.3.10

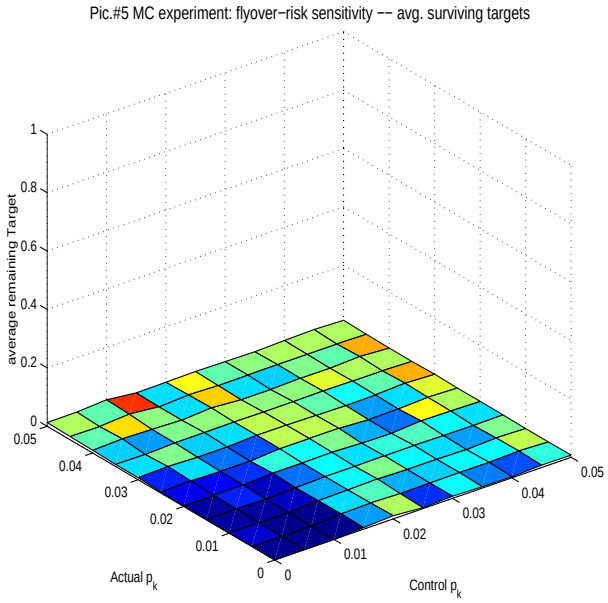


Fig. 2.3.11

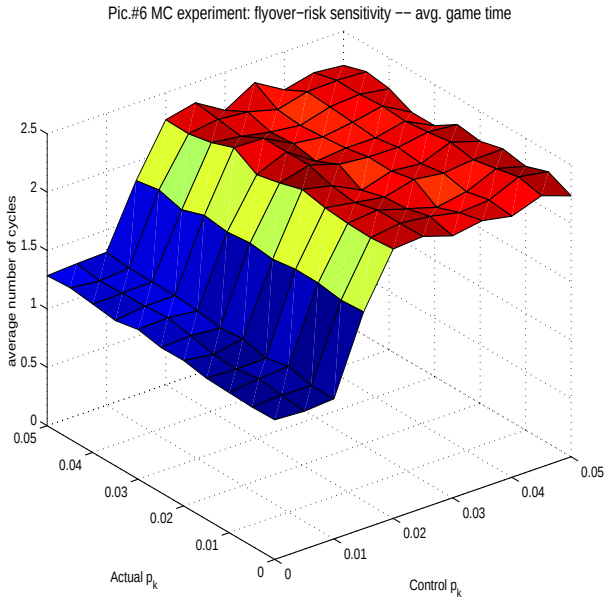


Fig. 2.3.12

Note:

1. The value still has two "regions". In the left region it is roughly linear (over the region), and in the right region it is constant.
2. The switch-over point from fly-over to rollback is different.
3. The structures of all figures are the same as for the previous data set.

The third set of data is for the case of two aircraft and three SAM sites.

- Figure 2.3.13 is the sample mean value.
- Figure 2.3.14 is the sample standard deviation.
- Figure 2.3.15 is the sample mean number of surviving a/c
- Figure 2.3.16 is the sample mean number of surviving SAMs
- Figure 2.3.17 is the sample mean number of surviving targets
- Figure 2.3.18 is the sample mean number of cycles until the end of the game.

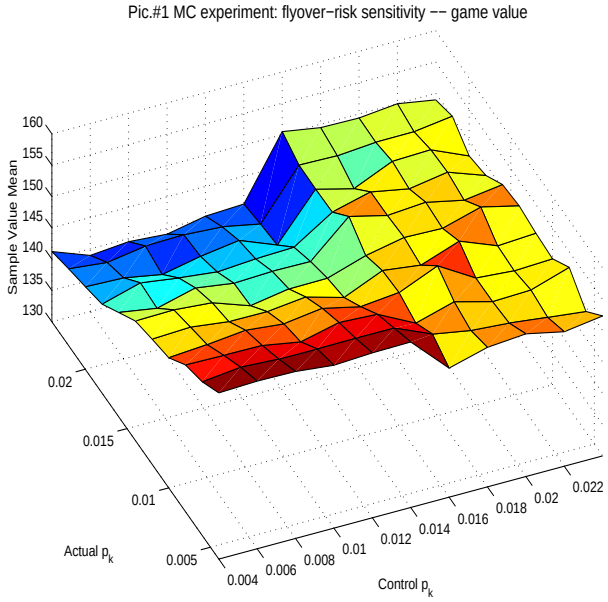


Fig. 2.3.13

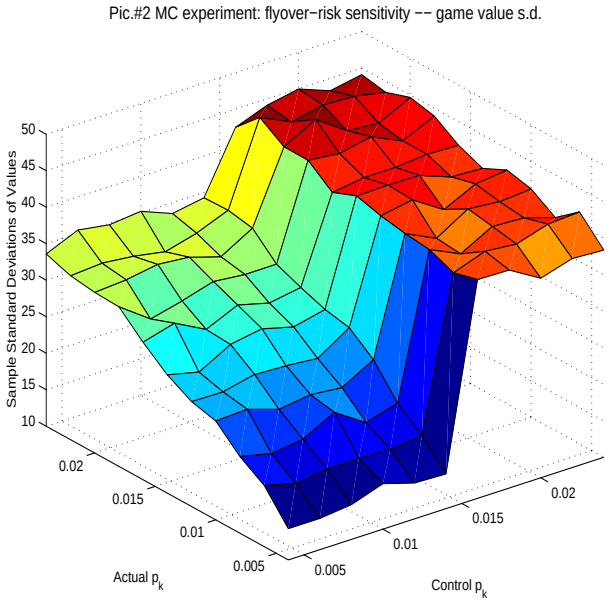


Fig. 2.3.14

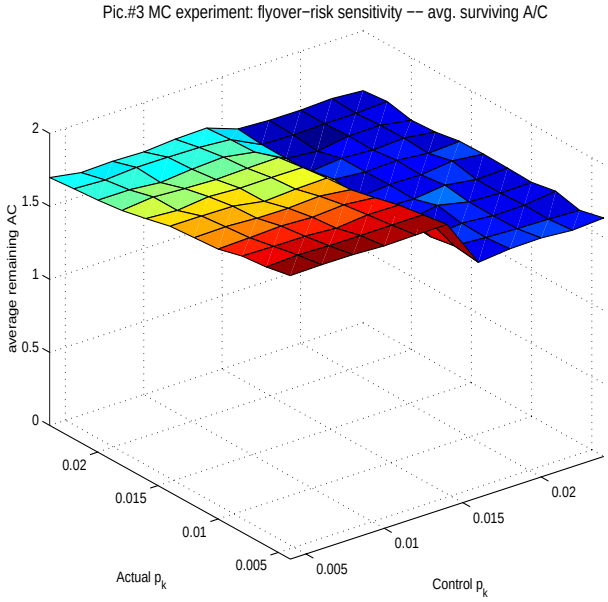


Fig. 2.3.15

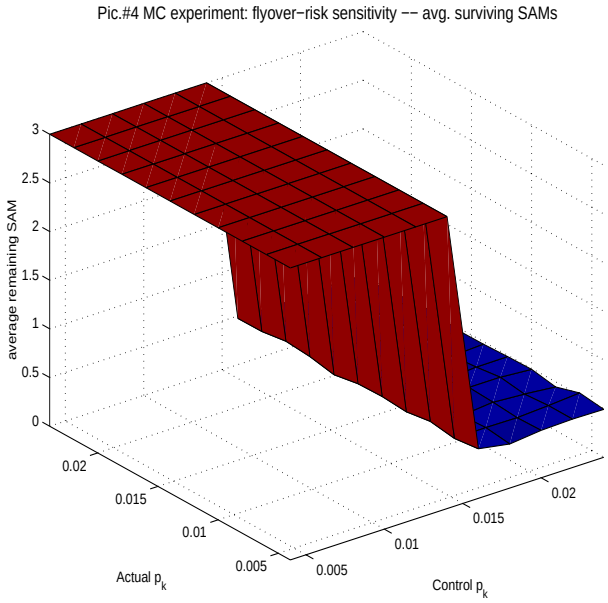


Fig. 2.3.16

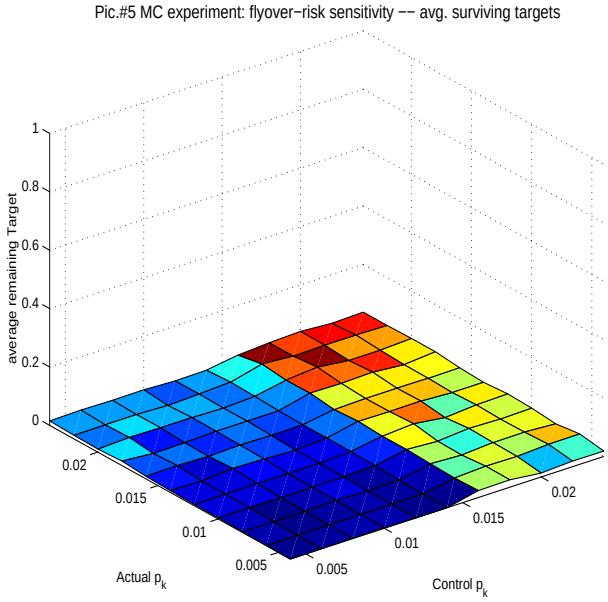


Fig. 2.3.17

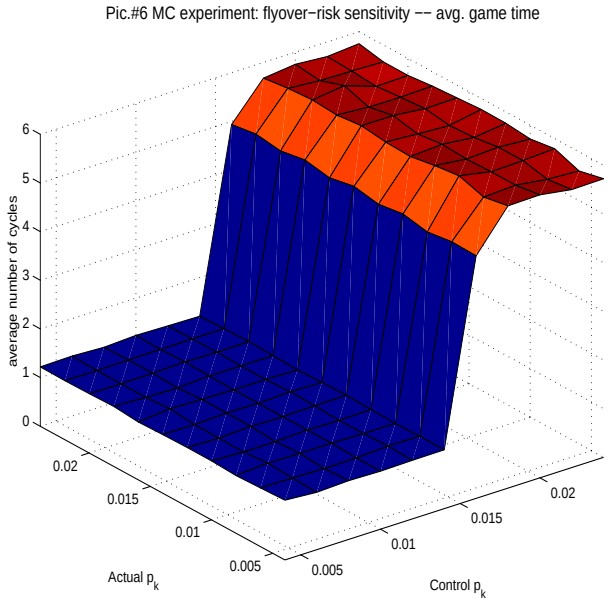


Fig. 2.3.18

Note:

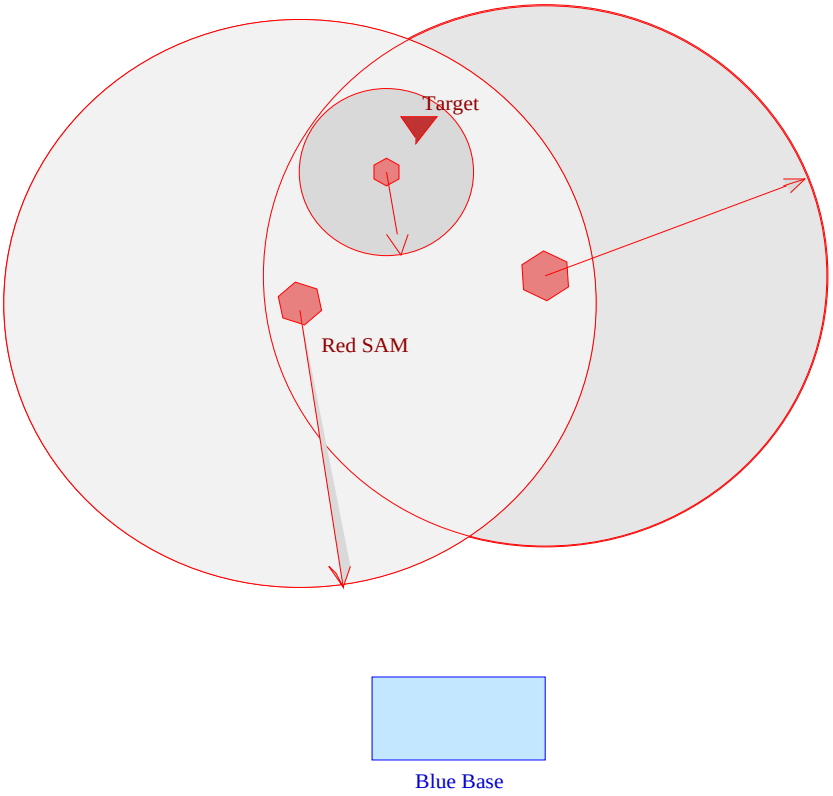
1. The value still has two "regions". In the left region it is roughly linear (over the region), and in the right region it is constant. Again the structure is essentially the same.

2. The data is noisier.

3. The switch-over point from fly-over to rollback is different.

4. A close examination of Figure 2.3.16 (surviving SAMs) indicates that the switch-over is not quite complete from one control α to the next. A little bit of the switch-over is not complete until after two steps. Actually, from other runs, we have seen that there is a reduction from complete fly-over to rollback in several stages, but the change is so rapid (as this parameter changes) that to a good approximation, it is a single switching point.

5. Note that in Figure 2.3.13, the value may not quite have its maximum relative to the line true $\alpha = 0.014$ at control $\alpha = 0.014$. This may represent some small error due to approximations made in the control computations.



Geography 2 Di

Fig. 2.3.19

Different Geographies

The controller can deal with more complicated geometries than that of the previous data sets. We have run it with the geographical distillation depicted in Figure 2.3.19. The geography is distilled into a file where the red sites whose umbrella must be flown through in going from site A to site B are recorded. The controller has been tested for this example geometrie as well. The landscape plots for these geometries are similar to those above and are not included..

2.4 Differing SAM lethalities

Previously, there was only one type of SAM in our Small Game Controller and Testbed. In this subsection, we indicate the changes that were made to allow the SAMs to have both different effective radii and different lethality (strengths). The possiblity of different strength SAMs had a significant effect on the shape of the value landscape and the optimal controls.

The different radii are an off-line matter which is eliminated in the geography distillation file.

The different lethality has been added to both the control software and the Small Game testbed. In the previous subsection, the optimal Blue strategy was essentially always either fly-over or rollback. With mutiple SAM strengths, the number of possibilities increases. With two SAM strengths, there are three control regions for Blue (fly-over, partial rollback, rollback). See Figures on following pages for an example with two a/c, two SAMs and a target. The SAMs had different lethalities.

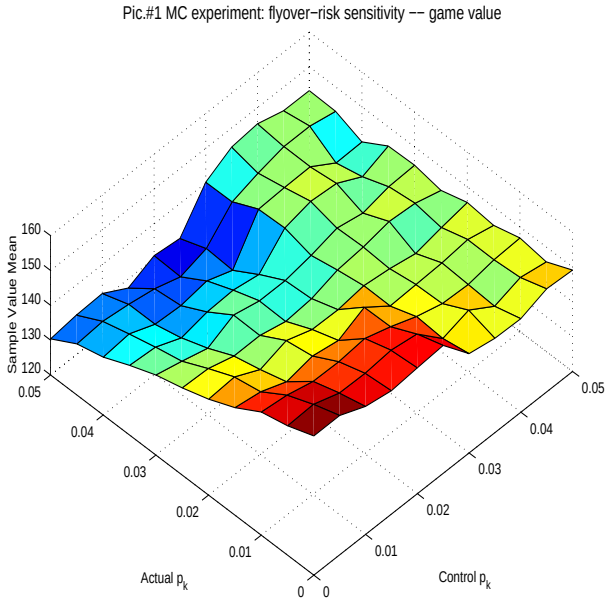


Fig. 2.4.1: Two SAM lethalities, Game Values

An obvious question is whether the two control policy jumps in the previous slide were due to two SAM strengths or to two SAMs? Figures 2.4.7–2.4.9 are for 2 a/c and 3 SAMs where SAMs have two strength types. Two SAMs are weaker and one is stronger while under direct attack; kept fly-over damages same for both types in this example. Note that there are only TWO policy jumps, thus indicating that the effect is related to the number of types of SAMs not the number of SAMs.

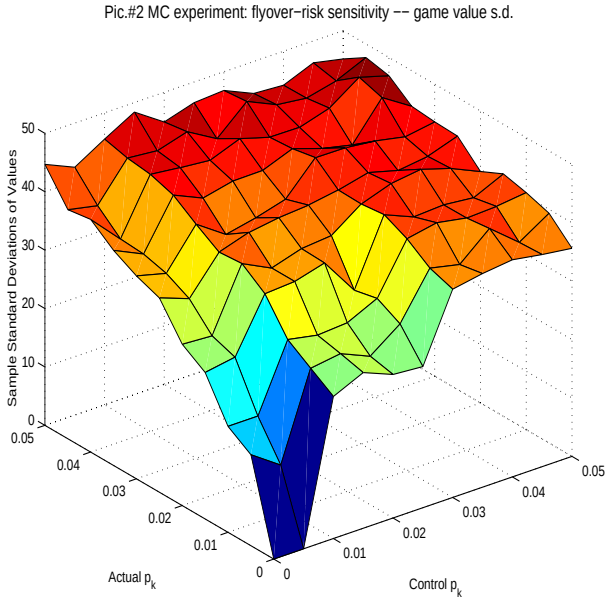


Fig. 2.4.2: Two SAM lethalities, Sample S.D. of Game Values

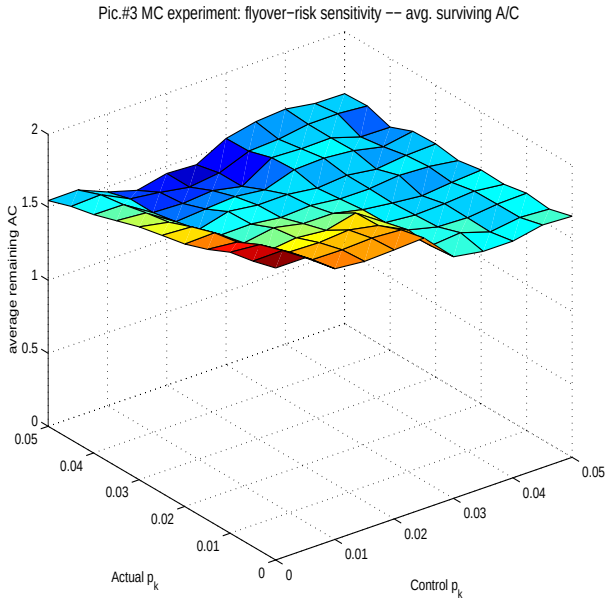


Fig. 2.4.3: Two SAM lethalities, Average remaining aircraft

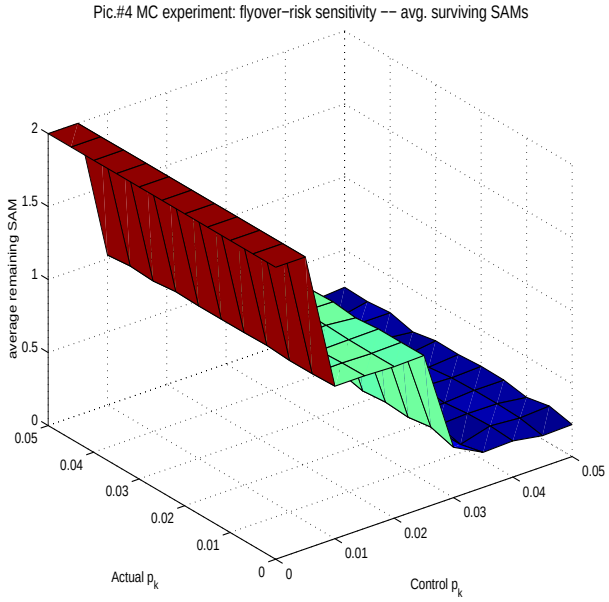


Fig. 2.4.4: Two SAM lethalities, Average remaining SAMs

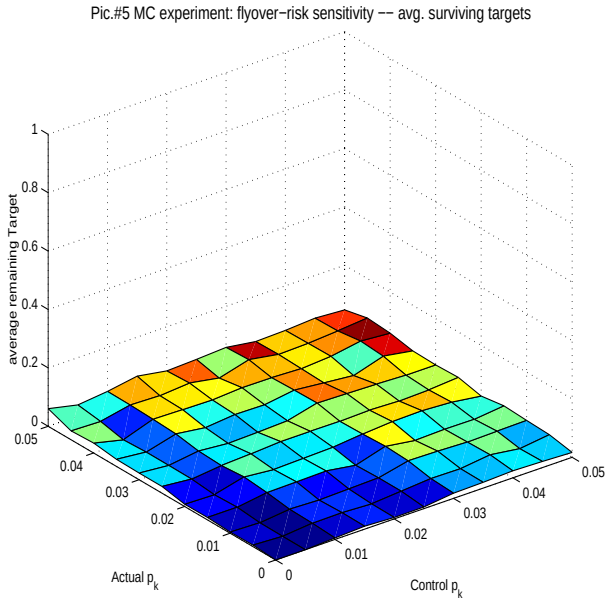


Fig. 2.4.5: Two SAM lethalities, Average remaining Targets

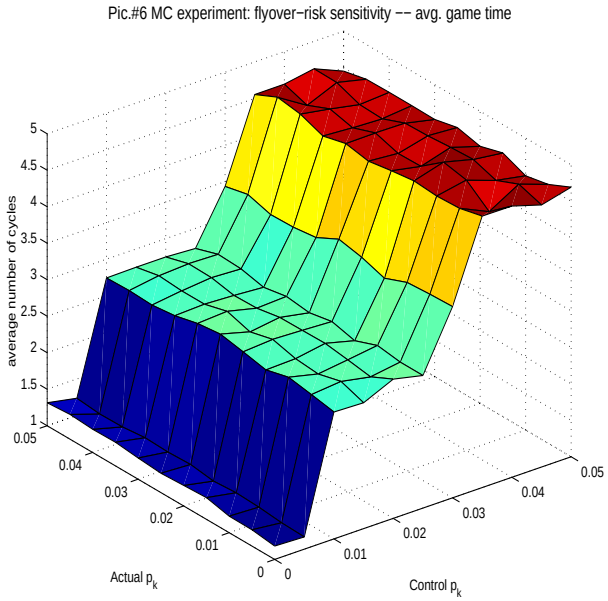


Fig. 2.4.6: Two SAM lethalities, Average game time (cycles)

The major result here is that one does not need to search over all of a/c control space. The search appears to be reduced to only $n+1$ (or maybe 2^n) policies for blue where n is the number of SAM lethality types. (Differing radii of coverage do not affect this.) This implies a HUGE computational savings; much larger problems solvable at this low level in the hierarchy.

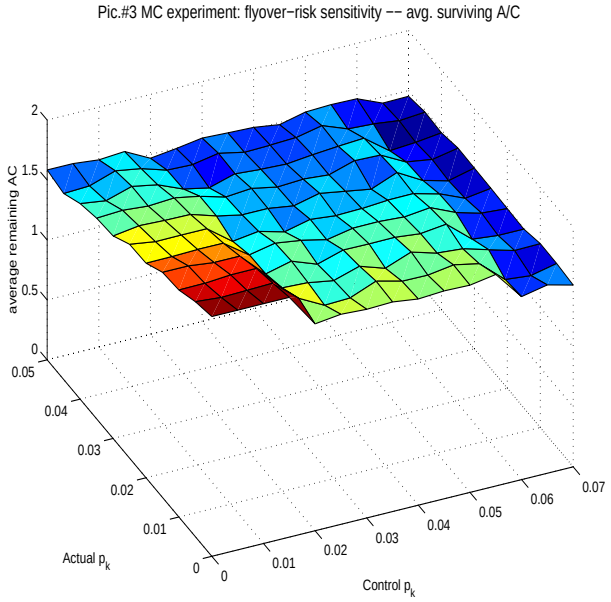


Fig. 2.4.7: Two SAM lethalities 3 SAMs, Average remaining aircraft

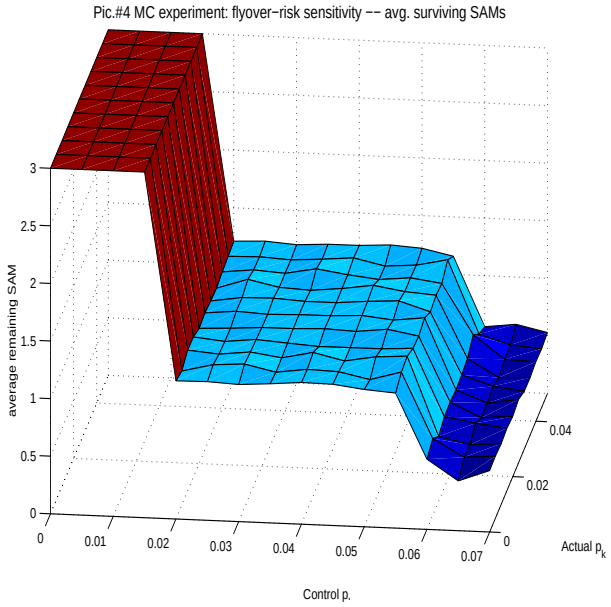


Fig. 2.4.8: Two SAM lethalities 3 SAMs, Average remaining SAMs

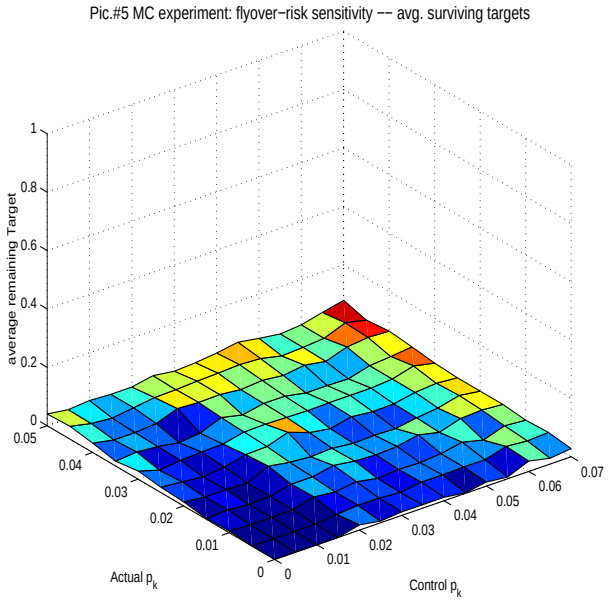


Fig. 2.4.9: Two SAM lethalities 3 SAMs, Average remaining Targets

2.5 Incorrect Assumptions of System Parameters

In the above subsections, the controllers for both the Blue and the Red players were computed from the same minimax code under the same assumptions. This yields the worst-case opponent from the Blue perspective. A modified version of the Small Game testbed was produced where the controllers do not operate under the same assumptions. Both the model parameters (transition probabilities) and the payoff functions that each player uses may differ from that used by the other player.

A small number of results were generated. In the examples from Subsection 2.3, Blue and Red both assume a value of $\mu_R = 3$ for the SAMs. (Note: $\mu_A = 20$, $\mu_T = 20$.) Here however, we modify this so that Blue assumes $\mu_R = 3$, and Red assumes $\mu_R = 20$. Results were plotted according to the Blue payoff function.

The case was for 2 a/c and 2 SAMs. This leads to control lookup tables of approximately 13,000 lines. The tables used different Red controls in approximately 2,000 lines (of the 13,000). However, the outcomes were not significantly affected for this pair of different of assumptions. The new payoff is depicted in Figure 2.5.1, and the difference between this value and that for the case where both assume $\mu_R = 3$ is depicted in Figure 2.5.2. The point is that in this particular example, although many control strategy lines were changed, the critical lines were not changed.

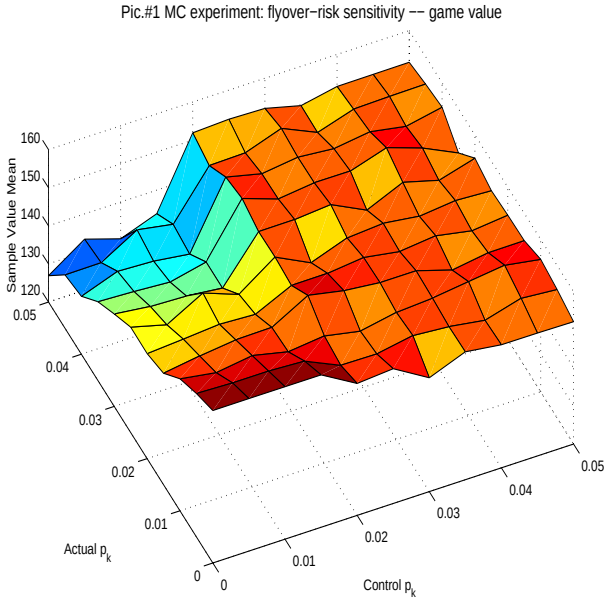


Fig. 2.5.1: Red's $\mu^R = 20$ (not same as Blue), No. of iterations = 500

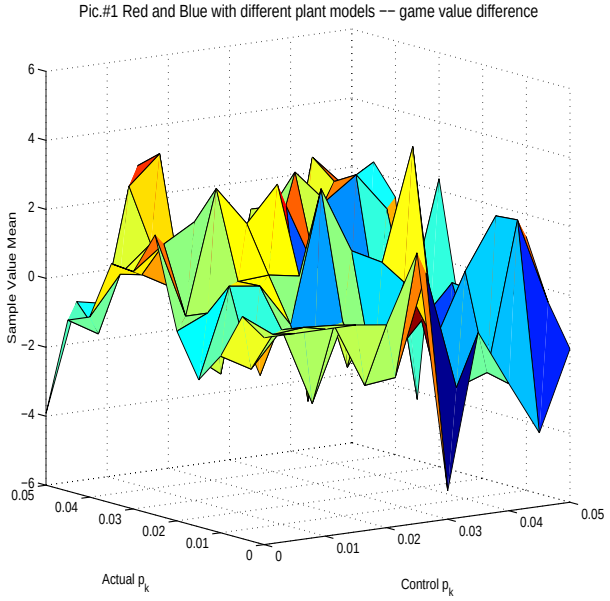


Fig. 2.5.2: Difference in value functions

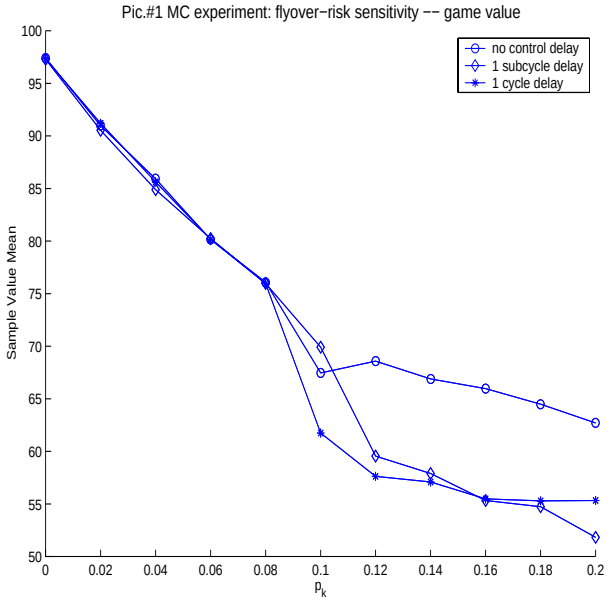


Fig. 2.6.1: Value function

2.6 Observation Delays

We did a small study of the effects of observation delays. For this study, we assumed that the observations of the state are perfect, but are delayed by some fixed amount of time. In the experiment, both players experienced the same informational delays. The results in Figures 2.6 are for the case of one aircraft and one SAM.

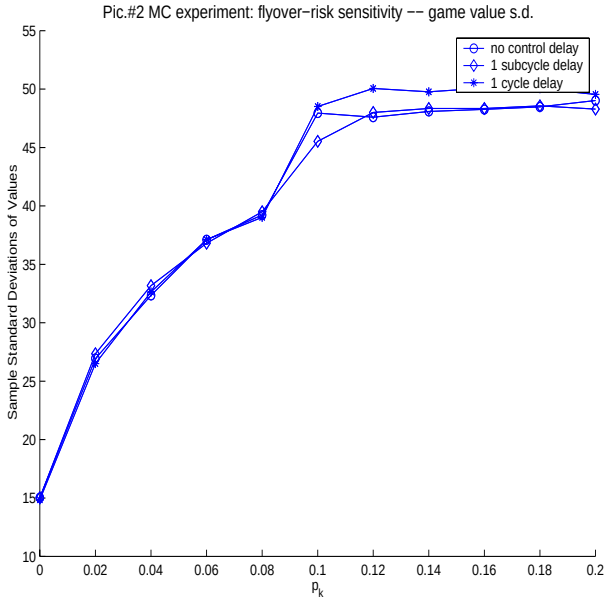


Fig. 2.6.2: Standard Deviation

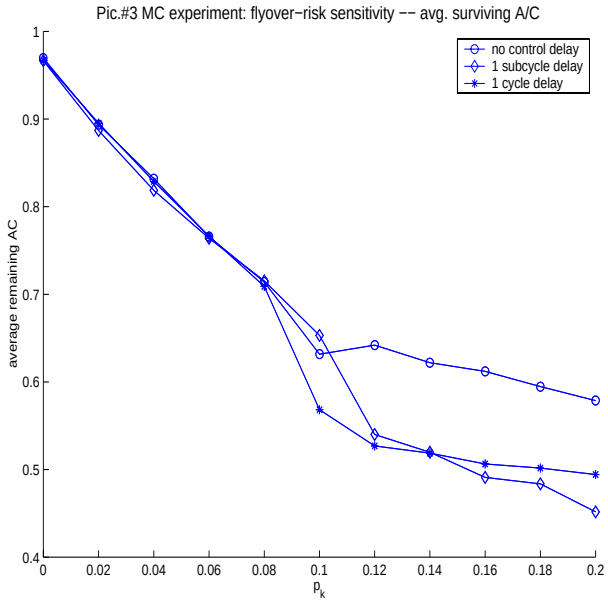


Fig. 2.6.3: Average remaining aircraft

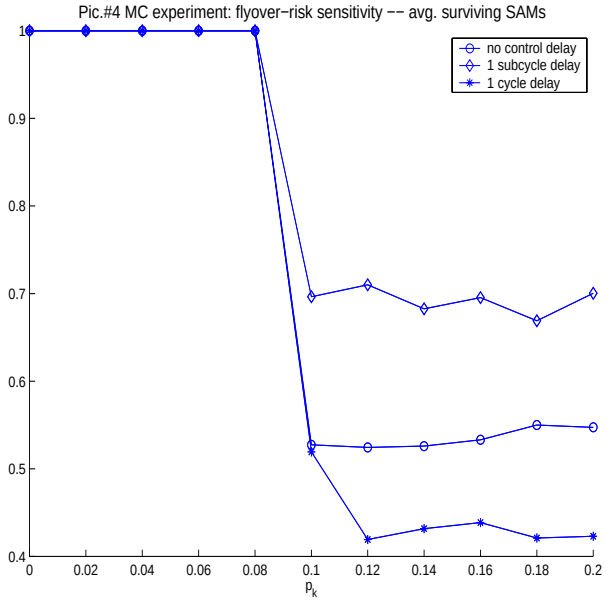


Fig. 2.6.4: Average remaining SAMs

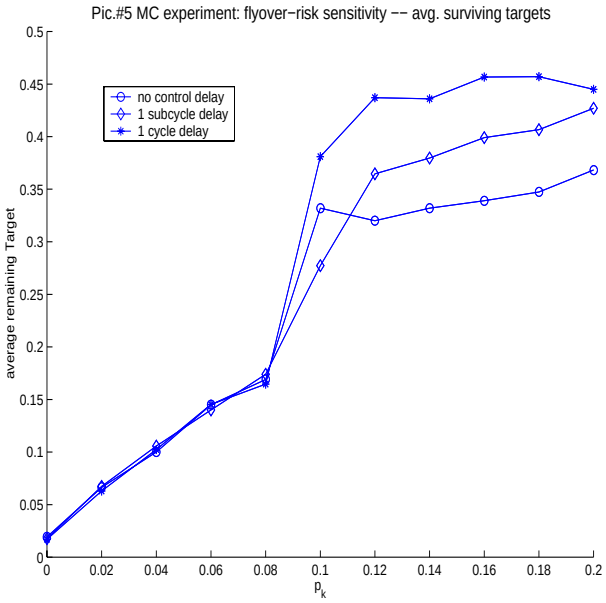


Fig. 2.6.5: Average remaining targets

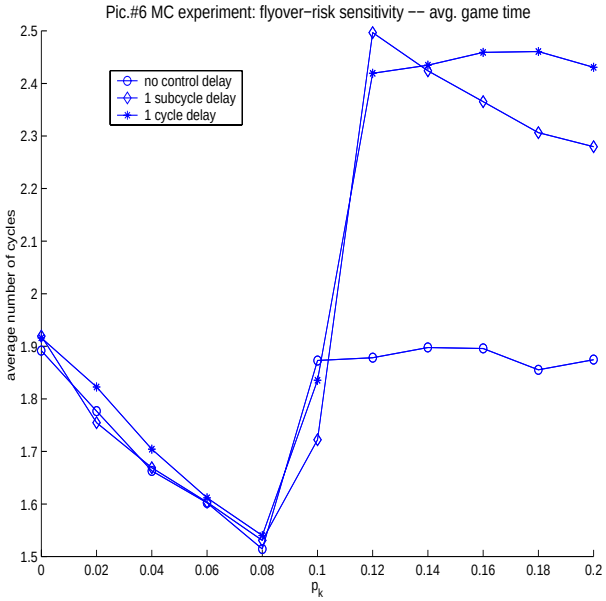


Fig. 2.6.6: Average game time (cycles)

Since we have delay in both controllers, one does not see a monotone decay in the value as a function of delay. However, the delay does tend to have a more deleterious effect on Blue. In the fly-over region, the delay has negligible effect. Note finally, that the length of the game tends to increase.

3 Optimal and Near-Optimal Route Generation

The first step taken is to find the optimal routes to the target(s) for the aircraft while avoiding the SAMs. First we discuss the control problem formulation.

3.1 Control Problem Formulation

We formulate the optimal routing problem from the base $\bar{x}^A = (\bar{x}_1^A, \bar{x}_2^A)$ to the target $x^T \in \mathbf{R}^2$ as an optimal (exit) control problem;

$$\min_u \quad \int_0^\tau (1 + \sum_{i=1}^{N_R} \sigma_i \, \ell_i(X^A - x_i^R)) \, dt \tag{3.1}$$

subject to $X^A(0) = \bar{x}^A$, $X^A(\tau) = x^T$ and

$$\frac{d}{dt} X^A(t) = u(t), \quad |u(t)| \le 1, \tag{3.2}$$

where $X^A(t) \in \mathbf{R}^2$ is the location of an aircraft, $u(t) = (u_1(t), u_2(t))$ is the velocity control, and x_i^R is the i^{th} opponent SAM site with strength σ_i, $1 \le i \le N_R$. The loss function ℓ_i represents a loss due to flying close to the site i, and for example $\ell_i(x^A - x_i^R) = \frac{1}{|x^A - x_i^R|}$. That is, the optimal route is determined so that the sum of time and total loss is minimized. The optimal control of (3.1) is given by the feedback law [14]

$$u(t) = -\frac{V_{x^A}(X^A(t))}{|V_{x^A}(X^A(t))|} \tag{3.3}$$

where the value function V satisfies the Hamilton-Jacobi-Bellman equation

$$\begin{aligned} -|V_{x^A}(x^A)| + \ell(x^A) = 0, \quad V(x^T) = 0 \\ \text{with } \ell(x^A) = 1 + \sum_{i=1}^{N_R} \sigma_i \, \ell_i(x^A - x_i^R), \end{aligned} \tag{3.4}$$

Next, we describe the numerical method to HJ equation (3.4). Let $V_{i,j}$ denote an approximation of V at each grid point (x_i, y_j) which is uniformly distributed over a square domain Ω in $\mathbf{R}^2$. Let $h > 0$ be the stepsize and we define the backward and forward difference

$$\begin{aligned} (D_x^-)_{i,j} V = \tfrac{V_{i,j} - V_{i-1,j}}{h}, \quad (D_x^+)_{i,j} V = \tfrac{V_{i+1,j} - V_{i,j}}{h} \\ (D_y^-)_{i,j} V = \tfrac{V_{i,j} - V_{i,j-1}}{h}, \quad (D_y^+)_{i,j} V = \tfrac{V_{i,j+1} - V_{i,j}}{h} \end{aligned} \tag{3.5}$$

We use the upwinding method of Godnov to discretize (3.4):

$$\sqrt{[\max((D_x^-)_{i,j} V, -(D_x^+)_{i,j} V)]^2 + [\max((D_y^-)_{i,j} V, -(D_y^+)_{i,j} V)]^2} = \ell_{i,j}$$

where

$$\ell_{i,j} = \ell(x_i, y_j).$$

We employ the fixed point iterate [36]: let $V_{i,j}^n$ denote the n-th iterate and we update $V_{i,j}^{n+1}$ by solving (3.5) for $V_{i,j}^{n+1}$ at each grid point, given $V_{i+1,j}^n$, $V_{i-1,j}^n$, $V_{i,j+1}^n$ $V_{i,j-1}^n$. The exact step is given as

$$a_{i,j} = \min(V_{i-1,j}^n, V_{i+1,j}^n), \quad b_{i,j} = \min(V_{i,j-1}^n, V_{i,j+1}^n), \quad c_{i,j} = \ell_{i,j}^2$$

$$s_{i,j} = c_{i,j} - (a_{i,j} - b_{i,j})^2$$

$$\begin{cases} V_{i,j}^{n+1} = \frac{1}{2}(a_{i,j} + b_{i,j}) + \sqrt{c_{i,j} + s_{i,j}} \text{ if } s_{i,j} \geq 0 \\ V_{i,j}^{n+1} = \min(a_{i,j} + b_{i,j}) + \sqrt{c_{i,j}} \qquad \text{if } s_{i,j} < 0 \end{cases}$$

We have the boundary (exit) condition $V_{\bar{i},\bar{j}} = 0$ at the target grid and also we set $V_{i,j} = \infty$ at the boundary of Ω. The initial iterate can be set is as $V_{i,j}^0 = |(x_i, y_j) - x^T|$.

We solve the closed-loop equation

$$\frac{d}{dt} X^A(t) = -\frac{V_{x^A}(X^A(t))}{|V_{x^A}(X^A(t))|}$$

by the finite difference method, given stepsize $\Delta t > 0$. We approximate V_x at the grid point (x_i, y_j) by the central difference approximation

$$\Psi_{i,j} = \left(\frac{V_{i+1,j} - V_{i-1,j}}{2h}, \frac{V_{i,j+1} - V_{i,j-1}}{2h} \right)$$

and use the bi-linear interpolation at (x, y) in the i, j sub-square, i.e.,

$$V_{x^A}(X^A) \sim \Psi(x, y) = (1 - d_1)((1 - d_2)\Psi_{i,j} + d_2 \Psi_{i,j+1}) + d_1((1 - d_2)\Psi_{i+1,j} + d_2 \Psi_{i+1,j+1})$$

where $X^A = (x, y)$ and $d_1 = x - x_i$ and $d_2 = y - y_j$. Thus,

$$X_{k+1}^A - X_k^A = -\Delta t \, \frac{\Psi(X_k^A)}{|\Psi(X_k^A)|}.$$

In summary

- Optimal routing problem is formulated as an optimal control problem.
- Optimal route is determined by Dynamic Programming (DP) principle.
- We develop an efficient and robust algorithm based on DP.
- The algorithm is implemented on Matlab and runs under 30 sec for 100 by 100 grid.

In Figure 3.1 we show

(1) The contour of the Value function (the potential curves for cost to the target (0,0) from the base) Red (high) to Blue (low).

(2) Routes from various starting points (black lines).

(3) Each route is normal to the potential curves.

(4) 7 SAMs (spots on Fig.) are covering the target at (0,0) with equal strength.

The algorithm assigns the risk factor to each selected route and determines a sequence of SAMs which we may be engaged with on the way to the target. The algorithm is also used to determine the accessibility to the selected targets from a specified air base by formulating the exit problem to the specified base.

We have also tested the algorithm by varying the strength σ_i of SAMs. The uncertainty of SAM location can be incorporated in our formulation by replacing the point location to the normal distribution with zero mean and selected variance. It is observed that the algorithm generates the routes which reflect to changes in the SAM strength and uncertainty of their locations.

The geographical constraints can be incorporated by modifying the loss function ℓ accordingly.

3.2 Multi-Body Dynamic Formulation

Our proposed optimal routing algorithm is quite efficient. It is most useful in the operational and planning level. In order to incorporate the SAM movement and agile and dynamical changes in their uncertainty and conditions, we propose the following feedback law based on multi-body interaction dynamics. The algorithm can be implemented in real time and on-board. Let x_j^T be the target location with value w_j for $1 \le j \le N_T$. A route $X^A(t),\ t \ge 0$ is determined as a solution to

$$\frac{d}{dt}X^A(t) = -\frac{W_{x^A}(X^A(t))}{|W_{x^A}(X^A(t))|}, \quad X^A(0) = \bar{x}^A \tag{3.6}$$

where the potential function W is given by

$$W(x^A) = \sum_{j=1}^{N_T} w_j\,|x^A - x_j^T| + \sum_{i=1}^{N_R} \sigma_i\, U(|x^A - x_i^R|),$$

e.g., $U(|x^A - x_i^R|) = \frac{1}{|x^A - x_i^R|}$. Thus, the force field W_{x^A} is given by

$$-W_{x^A}(x^A) = -\sum_{j=1}^{N_T} w_j\,\frac{x^A - x_j^T}{|x^A - x_j^T|} + \sum_{i=1}^{N_R} \sigma_i\,\frac{x^A - x_i^R}{|x^A - x_i^R|^3}.$$

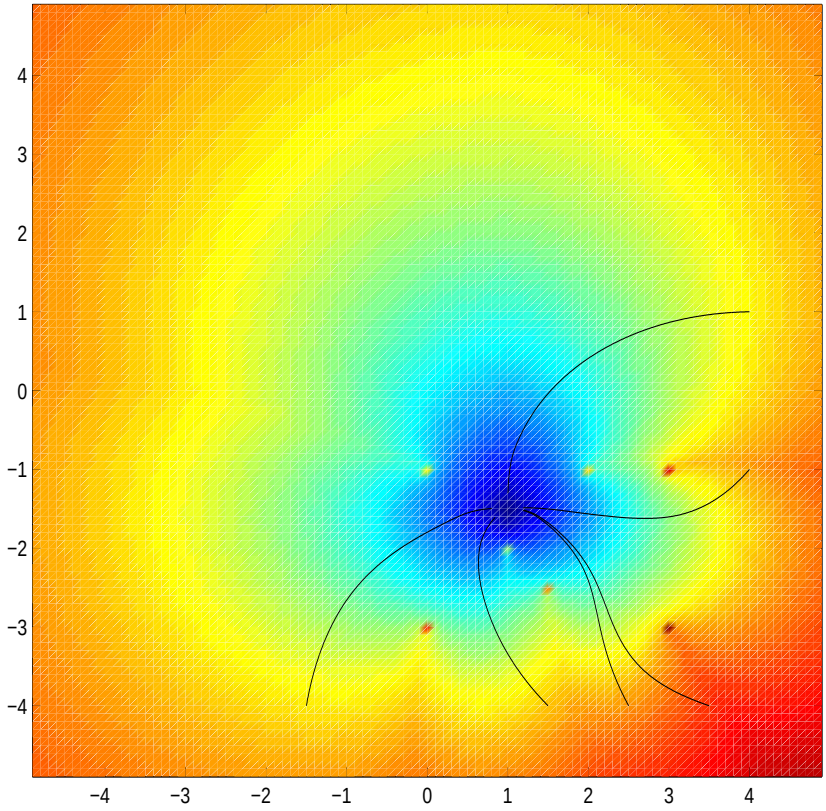


Fig. 3.1

Here the term $-\frac{x^A - x_j^R}{|x^A - x_j^R|}$ represents an attracting force to the target j and the term $\frac{x^A - x_i^R}{|x^A - x_i^R|^3}$ is for a repelling force from the SAM site i.

We can relate a closed loop system (3.6) to the optimal control problem (3.1)–(3.2) as follows. We define the performance index $\ell(x^A)$ by $\ell = |W_{x^A}|$. Note that if $\sigma_i = 0$ and $N^A = 1$, then $V = W = |x^A - x^T|$ and $u(t) = -\frac{X^A(t) - x^T}{|X^A(t) - x^T|}$ is optimal. $|W_x(x^A)|$ attains local minima and maxima at the same points as $\ell(x^A)$ defined by (3.4) does.

We compared the proposed algorithm with the optimal routes we generated. We observed that the algorithm generates a similar route to the optimal one with appropriately chosen SAM strength σ_i In the following Figure we show our comparison results.

Similarly, we also construct a movement of SAMs as follows. We assume that they protect the targets while avoiding voids.

$$\frac{d}{dt} X_i^R(t) = \frac{\tilde{W}_{x_i^R}(X^A(t), X^R(t))}{|\tilde{W}_{x_i^R}(X^A(t), X^R(t))|} \tag{3.7}$$

where the potential function $\tilde{W}$ is given by

$$\begin{aligned} \tilde{W}(x^A, x^R) = &- \sum_{j=1}^{N_T} w_j \, |x_i^R - x_j^T| \\ &+ \sum_{i=1}^{N_R} \sum_{j=1}^{N_R} \tilde{\sigma}_{i,j} \, \tilde{U}(|x_i^R - x_j^R|) + \sum_{i=1}^{N_R} \sigma_i \, U(|x^A - x_i^R|) \end{aligned}$$

e.g., $\tilde{U}(|x_i^R - x_j^R|) = \frac{1}{|x_i^R - x_j^R|}$. Thus the force field $\tilde{W}_{x^R}$ is given by

$$\begin{aligned} \tilde{W}_{x_i^R}(x^A, x^R) = &- \sum_{j=1}^{N_T} w_j \, \frac{x_i^R - x_j^T}{|x^R - x_j^T|} \\ &+ \sum_{j \neq i}^{N_R} \tilde{\sigma}_{i,j} \, \frac{x_i^R - x_j^R}{|x_i^R - x_j^R|^3} - \sigma_i \, \frac{x_i^R - x^A}{|x_i^R - x^A|} \end{aligned}$$

with attracting force $-\frac{x_i^R - x_j^T}{|x_i^R - x_j^T|}$ to the target j and repelling forces $\frac{x_i^R - x_i^R}{|x_i^R - x_j^R|^3}$.

A combined closed loop dynamics (3.6)–(3.7) has the game theoretic interpretation that is similar to the one for the optimal control problem. We consider the differential game

$$\begin{aligned} \min_u \max_v \int_0^\tau \Big(&\sum_{\ell=1}^{n} \Big[\sum_{j=1}^{N} w_j \, |x_\ell^A - x_j^T| + \sum_{i=1}^{m} \sigma_i \, \ell_i(x_\ell - y_i) \Big] \\ &- \sum_{i=1}^{m} \sum_{j=1}^{m} \tilde{\sigma}_{i,j} \, \tilde{\ell}(x_i^R - x_j^R) + \sum_{j=1}^{N} w_j \, |(x_i^R - x_j^T|) \, dt \end{aligned} \tag{3.8}$$

subject to

$$\frac{d}{dt}X_\ell^A(t) = u_\ell(t),\ |u_\ell| \le 1,\ \ \frac{d}{dt}X_i^R(t) = v(t),\ |v_i| \le \beta.$$

The optimal feedback solution to (3.8) is given by

$$u(t) = -\frac{V_{x^A}(x^A(t), x^R(t))}{|V_{x^A}(x^A(t), x^R(t))|},\ \ v(t) = \beta\,\frac{V_{x^R}(x^A(t), x^R(t))}{|V_{x^R}(x^A(t), x^R(t))|}$$

where the game value V satisfies

$$-|V_{x^A}(x^A, x^R)| + \beta\,|V_{x^R}(x^A, x^R)| + \ell(x^A, x^R) = 0. \tag{3.9}$$

Now we set $V = \hat{W}$ by

$$\begin{aligned}\hat{W} = &\sum_{\ell=1}^{n}[\sum_{j=1}^{N} w_j\,|x_\ell^A - x_j^T| + \sum_{i=1}^{m}\sigma_i\,U(x_\ell^A - x_i^R)] \\ &-\sum_{i=1}^{m}\sum_{j=1}^{m}\tilde{\sigma}_{i,j}\tilde{U}(x_i^R - x_j^R) + \sum_{j=1}^{N} w_j\,|(x_i^R - x_j^T|)\,dt\end{aligned}$$

and

$$\ell(x^A, x^R) = |\hat{W}_{x^A}(x^A, x^R)| - \beta\,|\hat{W}_{x^R}(x^A, x^R)|.$$

Then (3.8) holds.

4 Expanding Problem Size with Hierarchical Techniques

This section is divided into two parts. The first is concerned with dynamic allocation games with complete observation. The main approach involves game theoretical studies and hierarchical decomposition methods. The last subsection contains some mathematical proofs.

4.1 Hierarchical Games with Complete Observation

Two troops (red and blue) are engaged in a battlefield. Blue troop's assets include aircrafts (e.g. bombers) and its goal is to damage or destroy red targets (e.g. oil refinery) and surrounding SAMs (surface-to-air missiles). Red troop on the other hand operates SAMs to protect their resources (targets) and aims at damaging or destroying blue aircrafts.

In this section, we consider the case with completely observable states, i.e., all red states are available to blue troop's decision making.

Assume that the battle field is geographically divided into two regions R_1 and R_2; see Fig. 1.

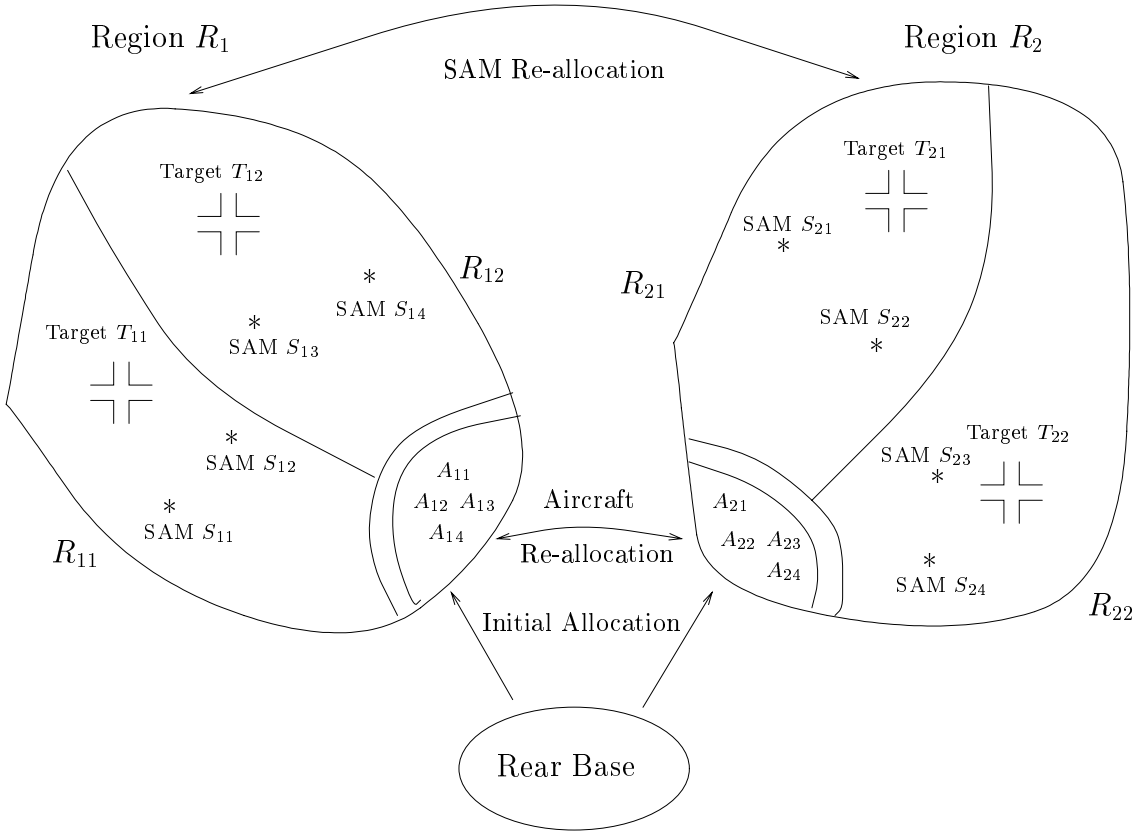


Fig. 1. The Model

To illustrate the idea of hierarchical control, let us divide further each of these regions into two smaller sub-regions R_{ij}, $i, j = 1, 2$ so that $R_1 = R_{11} \cup R_{12}$ and $R_2 = R_{21} \cup R_{22}$.

For simplicity, we assume that there are four targets, one in each of these four sub-regions. A target has two states {functional, destroyed} denoted by $\{1, 0\}$, respectively. Here "functional" means it is either operational or partially operational. We use T_{ij} to denote the state of the target in sub-region R_{ij}, $i, j = 1, 2$.

Decisions that concern both blue and red troops involve allocations of assets between different regions or sub-regions over time.

Owing to the inherent complexity of command control systems, it is difficult to obtain exact optimal solutions. To reduce the overall complexity of the underlying system, it is necessary to resort to hierarchical control approach via aggregation and disaggregation methods. Analysis of various aggregation methods in connection with manufacturing

systems and continuous-time dynamic systems and their near optimality can be found in [38], [39] and references therein. In this report, we consider the following hierarchical structure:

High level. Re-allocation between R_1 vs. R_2. These allocations are usually costly and are *infrequent* events. These decisions are made at higher level of the command control hierarchy.

Low level. Re-allocation between R_{i1} vs. R_{i2}, $i = 1, 2$, These are less costly and occur *occasionally*. Such decisions are made at lower level of the hierarchy.

Such decomposition approach is natural in command control systems because the interaction among each regions is weak compared with the interaction within the regions.

Let N^a denote the total number of blue aircraft units and N^s the total number of red SAM units, respectively, that are initially available.

In this report, let us first focus on the upper level decision making. Lower level problems will be treated subsequently. For $i = 1, 2$, we consider the aggregated target variable T_i defined as $T_i = T_{i1} + T_{i2}$. Recall that $T_{ij} \in \{0, 1\}$. Thus $T_i \in \{0, 1, 2\}$.

Re-allocation Decisions: Both blue and red troops have the option of moving their assets from one region to the other at certain costs. For simplicity, here the allocation is assumed to be instantaneous. Delays in these allocations can be handled in a similar fashion.

Given $i = 1, 2$, let X_i^a denote the total number of blue aircrafts in region R_i and let X_i^s denote the total number of red SAM units in R_i.

The blue troop decides if there is a need to move a number of aircraft units from one region to the other. We use U_i^a to denote the new aircraft allocation in region R_i.

Similarly, the red troop makes a decision over time on the allocation of its SAM units and the new allocation denoted by U_i^s in R_i, for $i = 1, 2$.

A fixed cost is incurred each time a re-allocation is made. For example, if l^a aircraft units are moved from R_1 to R_2, then corresponding cost to blue troop is

$$K^a \cdot l^a, \text{ for given } K^a > 0.$$

In this case, $U_1^a = X_1^a - l^a \geq 0$ and $U_2^a = X_2^a + l^a$.

Similarly, if l^s SAM units are moved from R_1 to R_2, then

$$K^s \cdot l^s, \text{ for some } K^s > 0,$$

is incurred and $U_1^s = X_1^s - l^s \geq 0$ and $U_2^s = X_2^s + l^s$.

We use the notation $X^a = (X_1^a, X_2^a)$, $X^s = (X_1^s, X_2^s)$, $T = (T_1, T_2)$, $U^a = (U_1^a, U_2^a)$, and $U^s = (U_1^s, U_2^s)$. We also use $I_X = (X^a, X^s, T)$ and $I_U = (U^a, U^s, T)$ to denote the states before and after a re-allocation, respectively.

Transition Probabilities. Given the current state vector I_X, if there is no re-allocation decision is made, then the states of aircrafts, SAMs, and targets can jump to any states according to a conditional probability given I_X. If there is a need for a re-allocation, then such allocation will be made immediately and I_X is changed to $I_U = (U^a, U^s, T)$. The distribution of the new state vector will be determined by the conditional probability given I_U.

In each region R_i, the jump rates of X_i^a depends on X_i^s. Similarly, the jump rates in X_i^s depends on X_i^a and the jump rates in T_i depends on both X_i^a and X_i^s.

Objective Function. The game is considered to be over if either all the aircrafts are destroyed, i.e., $\{X^a = 0\}$ or all the targets are destroyed, i.e., $\{T = 0\}$.

The objective of blue troop is to make re-allocation decisions over time so as to minimize $P(X^a = 0)$ and maximize $P(T = 0)$. On the other hand, red troop wants to maximize $P(X^a = 0)$ and minimize $P(T = 0)$ by allocating its SAM units.

Let $\mathcal{M}$ denote the state space of $I_X = (X^a, X^s, T)$. Also, let $\mathcal{M}_*$ and $\partial\mathcal{M}$ denote classes of transient states and absorbing states, respectively. Then $\mathcal{M} = \mathcal{M}_* \cup \partial\mathcal{M}$.

For each $I_X, I_U \in \mathcal{M}$, define the re-allocation cost function

$$G(I_X, I_U) = K^a|X_1^a - U_1^a| - K^s|X_1^s - U_1^s|,$$

Recall that $X_1^a + X_2^a = U_1^a + U_2^a$ and $X_1^s + X_2^s = U_1^s + U_2^s$. It follows that $G(I_X, I_U) = K^a|X_2^a - U_2^a| - K^s|X_2^s - U_2^s|$.

Let $I_X(n)$ denote the state vector at time n. Define the stopping time

$$\tau = \min\{n : \ I_X(n) \in \partial\mathcal{M}\}.$$

We also define the terminal cost

$$\Psi(I_X) = \chi_{\{X^a=0\}} - \chi_{\{T=0\}}.$$

where χ_A is the indicator function of a set A.

Note that

$$E\Psi(I_X(\tau)) = P(X^a(\tau) = 0) - P(T(\tau) = 0).$$

The objective function can be given as follows:

$$J(I_X, U^a(\cdot), U^s(\cdot)) = E\left\{\sum_{n=0}^{\tau} G(I_X(n), I_U(n)) + \Psi(I_X(\tau))\right\}.$$

The red troop wants to maximize J and the blue troop wants to minimize $\max_{U^s} J$.

Let $\mathcal{G} = \{f : \ \mathcal{M} \to R^1$ and f satisfies the boundary condition $v = \Psi$ on $\partial\mathcal{M}\}$.

Given $f \in \mathcal{G}$, let

$$H(f)(I_X) = \begin{cases} \min_{U^a} \max_{U^s} \Big\{ \sum_{J \in \mathcal{M}} P(I_U, J) f(J) + G(I_X, I_U) \Big\} & \text{for } I_X \in \mathcal{M}_* \\ \Psi(I_X) & \text{for } I_X \in \partial\mathcal{M}. \end{cases}$$

where $P(I, J)$ is the transition probability from state I to state J.

The associated Isaacs equation is given by

$$v = H(v).$$

We arrange the order of $\mathcal{M}$ so that the corresponding transition matrix has the form

$$P = \begin{pmatrix} P_1^* & P_2^* \\ 0 & I \end{pmatrix},$$

where P_1^* corresponds to the transient states in $\mathcal{M}_*$ and I is an identity matrix that corresponds the absorbing states in $\partial\mathcal{M}$.

Assumption (A). We assume that all eigenvalues of P_1^* are inside the unit circle.

Assumption A implies that

$$||P_1^* b|| \leq \alpha ||b||$$

for any vector b and $0 \leq \alpha < 1$, where the norm $||(b_1, \ldots, b_k)|| = \sqrt{b_1^2 + \cdots + b_k^2}$. Clearly, α determines the how fast the state vector reaches $\partial\mathcal{M}$, which in turn determines the convergence rate when solving the corresponding Isaacs equation.

Under Assumption A, there should be no other recurrent states other than the absorbing states in $\partial\mathcal{M}$. Namely, the game has to come to an end with either aircrafts or targets destroyed.

Theorem. Under Assumption A, the following assertions hold.

(1) (Uniqueness) The Isaacs equation $v = H(v)$ has a unique solution.

(2) (Convergence) Given $v_0 \in \mathcal{G}$, define $v_1 = H(v_0)$. In general, given $v_n \in \mathcal{G}$ and define $v_{n+1} = H(v_n)$. Then v_n converges to v, which is the solution to the Isaacs equation $v = H(v)$. Moreover, the convergence rate is of order α, i.e.,

$$||v_n - v|| \leq \frac{\alpha^n}{1 - \alpha} ||v_1 - v_0||.$$

(3) (Verification Theorem) Let v be a solution to $v = H(v)$. Then

$$v(I_X) \leq J_{\max}(I_X, U^a(\cdot)) := \max_{U^s(\cdot)} J(I_X, U^a(\cdot), U^s(\cdot))$$

Moreover, let U_*^a denote a minimizer of

$$\max_{U^s} \Big\{ \sum_{J \in \mathcal{M}} P(I_U, J) f(J) + G(I_X, I_U) \Big\}.$$

Then U^a_* is optimal, i.e.,

$$v(I_X) = J_{\max}(I_X, U^a_*(\cdot)).$$

Initial Allocation.

In practice, it is important for both sides to allocate their resources appropriately at the beginning of the game. This is especially the case when the re-allocation cost is high. One approach within the framework of dynamic games can be given as follows.

Let $v(I_X) = v(X^a_1, X^a_2, X^s_1, X^s_2, T_1, T_2)$. For fixed (T_1, T_2), the blue troop chooses (X^a_1, X^a_2) to minimize $\max_{(X^s_1, X^s_2)} v(I_X)$ subject to $X^a_1 + X^a_2 = N^a$.

Given these (X^a_1, X^a_2), the red troop chooses (X^s_1, X^s_2) to maximize $v(I_X)$ subject to $X^s_1 + X^s_2 = N^s$.

Example. Take $N^a = 8$, $N^s = 8$, and $K^a = 0.1$, $K^s = 0.2$. Let I^a denote either X^a_1 (or X^a_2) and let I^s denote either X^s_1 (or X^s_2). We take

$$P(I^a \to (I^a - 1)|I^s) = \frac{0.3I^s}{0.3I^s + 1};$$

$$P(I^s \to (I^s - 1)|I^a) = \frac{0.3I^a}{0.3I^a + 1};$$

Let $p_0 = \dfrac{0.05I^a + 1}{0.2I^s + 1}$, $p_1 = \dfrac{p_0}{p_0 + 1}$, and $p_2 = \dfrac{1}{p_0 + 1}$. Consider

$$P(T \to T'|I^a, I^s) = \begin{pmatrix} 1 & 0 & 0 \\ p_1 & p_2 & 0 \\ 0 & p_1 & p_2 \end{pmatrix}$$

Given $T_1 = T_2 = 2$, the optimal initial allocation is

$$(X^a_1, X^a_2, X^s_1, X^s_2) = (4, 4, 4, 4).$$

Moreover, given the state vector, the optimal policies can be found in a lookup table. For instance,

if $(X^a_1, X^a_2, X^s_1, X^s_2, T_1, T_2) = (4, 4, 3, 5, 2, 2)$, then $(U^a_1, U^a_2, U^s_1, U^s_2) = (4, 4, 3, 5)$; no changes needed.

If $(X^a_1, X^a_2, X^s_1, X^s_2, T_1, T_2) = (4, 4, 3, 5, 0, 2)$, then $(U^a_1, U^a_2, U^s_1, U^s_2) = (0, 8, 3, 5)$, i.e., one only needs to move 4 aircraft units from R_1 to R_2.

If $(X^a_1, X^a_2, X^s_1, X^s_2, T_1, T_2) = (1, 1, 5, 3, 0, 2)$, then $(U^a_1, U^a_2, U^s_1, U^s_2) = (0, 2, 4, 4)$, i.e., move 1 aircraft unit and 1 SAM unit from R_1 to R_2.

Lower Level re-allocation: Let us discuss briefly lower level allocation. We only consider the re-allocations between regions R_{11} and R_{12} because the situation is similar in regions R_{21} and R_{22}.

In this case, the state variables are the number of targets T_{11}, T_{12} such that $T_{1j} \in \{0, 1\}$; blue aircraft units X^a_{11}, X^a_{12}, where X^a_{1i} denotes the total number of aircraft units in region R_{1i}, $i = 1, 2$ and red SAM units X^s_{11}, X^s_{12}, where X^s_{1i} denotes the total number of SAM units in region R_{1i}, $i = 1, 2$.

The re-allocation decisions are involved to change $(X^a_{11}, X^a_{12}) \to (U^a_{11}, U^a_{12})$ such that $X^a_{11} + X^a_{12} = U^a_{11} + U^a_{12}$ for blue troop and change $(X^s_{11}, X^s_{12}) \to (U^s_{11}, U^s_{12})$ such that $X^s_{11} + X^s_{12} = U^s_{11} + U^s_{12}$ for red troop.

A setup cost is incurred if a re-allocation decision is made. For example, if l^a aircraft units are moved from R_{11} to R_{12}, then $K^a_1 \cdot l^a$ is incurred and $U^a_{11} = X^a_{11} - l^a$ and $U^a_{12} = X^a_{12} + l^a$. Similarly, if l^s units are allocated from R_{11} to R_{12}, then $K^s_1 \cdot l^s$ is incurred and $U^s_{11} = X^s_{11} - l^s$ and $U^s_{12} = X^s_{12} + l^s$. Typically, $K^a_1 \leq K^a$ and $K^s_1 \leq K^s$. This is because the cost for moving within a region is greater than when moving between regions.

Transition probabilities and objective functions can be determined similar to that for the upper level model.

Sensitivity Tests (Upper Level Only)

We use the Monte Carlo method to test the sensitivity of the the control policies with respect to changes in various parameters. Consider the model with a set of "true" parameters of transition probabilities and setup costs. We generate sample paths using these true parameters.

Then we add a small perturbation ε to each of these parameters. The perturbed model produces a set of control policies $U^{a,\varepsilon}, U^{s,\varepsilon}$ (this is a time consuming part). Such control policies lead to the corresponding theoretical upper value V^ε and Monte Carlo averages J^ε.

The following parameters are used in the tests.

Setup Cost: $K^a = 0.1 + 0.1\varepsilon$ and $K^s = 0.2 + 0.1\varepsilon$.

Transition Probabilities: Let I^a denote either X^a_1 or X^a_2 and let I^s denote either X^s_1 or X^s_2. We take

$$P(I^a \to (I^a - 1)|I^s) = \frac{(0.3 + 0.1\varepsilon)I^s}{(0.3 + 0.1\varepsilon)I^s + 1};$$

$$P(I^s \to (I^s - 1)|I^a) = \frac{(0.3 + 0.1\varepsilon)I^a}{(0.3 + 0.1\varepsilon)I^a + 1};$$

Let $p_0 = \dfrac{(0.05 + 0.1\varepsilon)I^a + 1}{(0.2 + 0.1\varepsilon)I^s + 1}$ and

$$p_1 = \frac{p_0}{p_0 + 1}, p_2 = \frac{1}{p_0 + 1}.$$

$$P(T \to T'|I^a, I^s) = \begin{pmatrix} 1 & 0 & 0 \\ p_1 & p_2 & 0 \\ 0 & p_1 & p_2 \end{pmatrix}$$

The numerical results are summarized in the following tables. Two cases are considered.

Case 1: $I_{X^0} = (X_1^a, X_2^a, X_1^s, X_2^s, T_1, T_2) = (4, 4, 2, 6, 2, 2)$.

In this case, $V^0(I_{X^0}) = -0.6844$ and $J^0(I_{X^0}) = -0.4910$.

Table 1. Perturbations in Transition Probabilities

ε	V^ε	$\lvert V^\varepsilon - V^0\rvert$	J^ε	$\lvert J^\varepsilon - J^0\rvert$
0.5	-0.6732	0.0112	-0.4780	0.0130
0.1	-0.6826	0.0018	-0.4930	0.0020
0.01	-0.6842	0.0002	-0.4910	0.0000

Table 2. Perturbations in Setup Costs

ε	V^ε	$\lvert V^\varepsilon - V^0\rvert$	J^ε	$\lvert J^\varepsilon - J^0\rvert$
0.5	-0.5626	0.1218	-0.3675	0.1235
0.1	-0.6599	0.0245	-0.4701	0.0209
0.01	-0.6819	0.0025	-0.4885	0.0025

Case 2: $I_{X^0} = (X_1^a, X_2^a, X_1^s, X_2^s, T_1, T_2) = (4, 4, 4, 4, 2, 2)$.

Now, $V^0(I_{X^0}) = -0.7196$ and $J^0(I_{X^0}) = -0.6490$.

Table 3. Perturbations in Transition Probabilities

ε	V^ε	$\lvert V^\varepsilon - V^0\rvert$	J^ε	$\lvert J^\varepsilon - J^0\rvert$
0.5	-0.7120	0.0076	-0.5900	0.0590
0.1	-0.7186	0.0010	-0.6330	0.0160
0.01	-0.7195	0.0009	-0.6490	0.0000

Table 4. Perturbations in Setup Costs

ε	V^ε	$\lvert V^\varepsilon - V^0\rvert$	J^ε	$\lvert J^\varepsilon - J^0\rvert$
0.5	-0.6121	0.1075	-0.5245	0.1245
0.1	-0.6979	0.0217	-0.6191	0.0299
0.01	-0.7174	0.0022	-0.6480	0.0011

Observations: As can be seen from Tables 1-4,

(1) $|V^\varepsilon - V^0| \to 0$ as $\varepsilon \to 0$; this suggests that the value function V^ε is continuous in ε.

(2) J^0 is close to V^0; this means that the averaged value function using the Monte Carlo method is close to the corresponding theoretical value.

(3) $|J^\varepsilon - J^0| \to 0$ as $\varepsilon \to 0$; this suggests that one can do nearly as good as the optimal policies when using the perturbed policies.

These tests suggest that the control policies obtained are robust in the sense that the corresponding outcomes are not much affected when using close-to-real parameters.

How to determine the value of parameters?

(1) Transition probabilities can be determined using historical data and standard statistical tests.

(2) Setup costs can be derived using actual dollar amount of each re-allocation plus the consideration of time delay needed for the allocation.

"Value."

(1) Without the hierarchical approach: only deal with 2-4 aircraft units and 4-6 sam units; Now we can handle much larger aircraft and SAM units.

(2) Dramatic reduction of computational effort and time.

(3) The hierarchical approach also provides initial allocations, which makes the system free from near future re-allocation.

(4) Historical data and human expertise can be used to model a real life scenario, which makes it possible to use a computer to help human to make decisions that are too hard to obtain otherwise.

Models with Other Objectives Functions: We may also consider other objective functions. For example, one may consider the model in which the blue troop wants to minimize the expected exit time $E\sum_{k=0}^{\tau}\rho^k$, where $\tau = \min\{n:\ T(n)=0\}$ and $0<\rho<1$ is a discount factor and the red troop wants to maximize $E\sum_{k=0}^{\tau}\rho^k$. In this case, the corresponding Isaacs equation is

$$v(I_X) = \min_{U^a}\max_{U^s}\left\{1+\rho\sum_{J\in\mathcal{M}}P(I_U,J)v(J)+G(I_X,I_U)\right\},$$

with boundary condition $v|_{T=0}=0$.

The other type of objective function typically used in the literature is

$$J = E\left\{\sum_{k=0}^{\infty}\rho^k F(I_X(n))+\rho^k G(I_X(n),I_U(n))\right\}.$$

where $F(I_X) = -a(X_1^a + X_2^a) + b(X_1^s + X_2^s) + c_1T_1 + c_2T_2$ for positive constants a, b, c_1, c_2. The associated Isaacs equation is given by

$$v(I_X) = \min_{U^a} \max_{U^s} \left\{ F(I_X) + \rho \sum_{J \in \mathcal{M}} P(I_U, J) v(J) + G(I_X, I_U) \right\}.$$

All results stated in the above theorem can be extended to incorporate these two cases except that there is no need for Assumption A because of the discount factor involved.

4.2 Proofs Related to the Hierarchical Technique

In this appendix, we provide proofs of results.

For any $f \in \mathcal{G}$ and $I_X = (X^a, X^s, T) \in \mathcal{M}_*$, $I_U = (U^a, U^s, T) \in \mathcal{M}_*$, define the mapping

$$H(f)(I_X) = \min_{U^a} \max_{U^s} \Big\{ \sum_{J \in \mathcal{M}} P(I_U, J) f(J) + G(I_X, I_U) \Big\}$$

and $H(f)$ satisfies the boundary conditions.

Proof of Theorem. (Part 1). Let $f, f' \in \mathcal{G}$. Given $I_X \in \mathcal{M}_*$, there exists U_*^a such that, for all U^s,

$$H(f')(I_X) = \max_{U^s} \Big\{ \sum_{J \in \mathcal{M}} P(I_U^0, J) f'(J) + G(I_X, I_U^0) \Big\},$$

where $I_U^0 = (X^a, X^s, U_*^a, U^s, T)$. Moreover, there exists U_*^s such that

$$\max_{U^s} \Big\{ \sum_{J \in \mathcal{M}} P(I_U^0, J) f(J) + G(I_X, I_U^0) \Big\} = \sum_{J \in \mathcal{M}} P(I_U^*, J) f(J) + G(I_X, I_U^*),$$

where $I_U^* = (X^a, X^s, U_*^a, U_*^s, T)$. Then we have

$$\begin{aligned} H(f)(I_X) - H(f')(I_X) &\le \sum_{J \in \mathcal{M}} P(I_U^*, J) f(J) + G(I_X, I_U^*) \\ &\quad - \sum_{J \in \mathcal{M}} P(I_U^*, J) f'(J) + G(I_X, I_U^*) \\ &\le \sum_{J \in \mathcal{M}_*} P_1^*(I_U^*, J)(f(J) - f'(J)) \\ &\le \alpha ||f - f'||. \end{aligned}$$

Similarly, replace the role of f and f', we obtain the opposite inequality. In view of the assumption on P_1^*, it follows that

$$||H(f) - H(f')|| \le \alpha ||f - f'||. \ \square$$

(Part 2.) It is easy to show that

$$||v_{n+1} - v_n|| = ||H(v_n) - H(v_{n-1})|| \le \alpha ||v_n - v_{n-1}||.$$

It follows that

$$||v_{n+1} - v_n|| \leq \alpha^n ||v_1 - v_0||.$$

This implies that $v = \lim_{n\to\infty} v_n$ exists and is a solution to the Isaacs equation $v = H(v)$ with boundary condition $v = \Psi$ on $\partial\mathcal{M}$.

(Part 3.) This part is similar to the proof in [4, p. 14]. We only sketch the proof for the sake of completeness.

Using $v = H(v)$, we have

$$v(I_X(n)) \leq \max_{U^s(n)} \left\{ E[v(I_X(n+1))|U^a(n), U^s(n), I_X(n)] + G(I_X(n), I_U(n)) \right\}.$$

The equality holds if $U^a(n) = U^a_*(n)$. This leads to

$$v(I_X(0)) \leq \max_{U^s(\cdot)} J(I_X(0), U^a(\cdot), U^s(\cdot)).$$

The equality holds if $U^a(n) = U^a_*(n)$ for each n. This completes the proof.

5 The C^2 Estimation and Control under Imperfect Information

Most of the work done under JFACC was control (including games) under full state information (i.e. full state feedback). However, partial, imperfect and even purposefully corrupted information is a critical part of warfare. In this section, we address estimation and control under partial/imperfect information. In the latter case, we take particular care to address the presence of an intelligent adversary in the system. We will discuss some simple algorithms for estimation of system state given likely data types first. Then we will discuss control under imperfect information in the final two subsections.

We will use the same problem as given in the previous sections where the Blue player is sending aircraft against Red player SAMs and targets.

The first basic idea is that the players handle uncertainty by maintaining probability distributions on the location and number of their opposing player forces. As in most traditional approaches to output feedback control, these probability distributions allow each player to estimate the likely states of his or her opponent. With a state estimate, one may then apply the control derived from the full state feedback analysis. This approach is by far the most common treatment of control under partial or incomplete information. It remains, however, to derive the state estimator from the probability distributions.

The second basic idea is that the estimator should take into account not only the likelihood of the opponents' states but also the risk associated with those states. Encoded by the the value function, the risk or loss associated with certain states is computed in the full state feedback game situation. Our approach integrates the estimation and

control to balance the objective function and its measurement of risk with the probability distributions modeling the likely states of the opponent.

In this section, we describe the estimation and output feedback problems in the context of command and control applications. We also provide some results derived from a detailed Monte Carlo simulation of the processes involved.

5.1 The Information State Variables

The formulation of this stochastic game relies on two separate state variables for each player: the "true" state and the information state. Each player maintains knowledge of his own state, as well as an information state quantifying his uncertainty in his opponent's state. Thus, the state variable $s \in SS$ is composed of four components: the true Blue state, the Blue information state (estimation of Red), the true Red state, and the Red information state (estimation of Blue). The true states have been discussed above.

The Red information state consists of track filter parameters needed to estimate the location of the Blue aircraft. For each blue aircraft detected, the Red information state maintains an estimate of the position, the aircraft velocity, and the covariance of these quantities.

The Blue information state is a $3 \times G$ matrix, whose entries are the probability of a Red entity of a given type at a particular grid point. That is, $bi_{k,g}$ is the probability of a type k entity at grid location g, where $k = 1$ denotes SAM, $k = 2$ denotes emitter, and $k = 3$ denotes nothing.

5.2 Information Probability Modeling

To determine the appropriate transition probabilities, we will need probabilities to quantify how engagements are resolved and how information propagates.

The fundamental probabilities in an engagement are the probability that a Blue attack destroys a Red entity and the probability that a Red SAM attack destroys a Blue aircraft. These probabilities have been described above. Here we focus on the probability distributions for the information states.

Detection and classification probabilities are the primary entities of this modeling effort. Both players have detection problems. The Red player has the additional problem of establishing a track on the Blue aircraft, while the Blue player has the problem of discriminating between SAM radars with track and defensive missile guidance capability and emitters, which are decoys emitting a signal that "sounds like a SAM radar."

We denote by $p_d^r(w)$ the probability that a Red SAM detects an aircraft present at a distance w. Likewise, we denote by $p_d^b(w)$ the probability that a Blue aircraft detects a defensive entity (SAM radar or emitter) that is turned on (i.e., emitting a signal).

5.3 The Estimation Problem for Blue

For the Blue player, we assume that the electronic system has signal processing capability to perform a classification based on received signals. We model this capability with a simple and flexible two-class statistical discriminator, which is encapsulated in statistical error probabilities. We define $pbcc(w, l)$ to be the probability that a Blue aircraft correctly classifies a Red entity of type l at a distance w. This simple model applies to many, if not most, radar-based discrimination schemes. The underlying data processing of the radar returns could range from the standard linear discriminator to a neural net or nearest-neighbor classifier. Any of these schemes will have probability of correct classification.

We apply the standard Bayesian approach to modeling the information state updates. For the Blue state, we begin with the "observation likelihood." That is, we seek to determine the probability that a SAM, emitter, or nothing is at grid location g given what we've observed at the current time. We set

$$pbo_{(\cdot),g} = \sum_{k=1}^{N^A} \Big(1 - \frac{w_k}{W}\Big) pbcc(w_k, bs_{1,k}, rs_{(\cdot),g}),$$

in which w_k denotes the distance between the $k-th$ aircraft and the grid location g, $W = \sum_{k=1}^{N^A} w_k$, and the function $pbcc$ is the probability that Blue correctly classifies the entity at a site. Conceptually, the function should decrease with w_k : small distance should translate into more accurate classification. The dependence on the Blue state should be simple: as long as the Blue aircraft is alive ($bs_{1,g} \neq -1$), then $pbcc$ depends only on the actual Red state and the distance. Then the Blue information update formula becomes

$$bi'_{l,g} = \frac{pbo(l, g) bi(l, g)}{\sum_{g'} pbo(l, g) bi(l, g)},$$

through an application of Bayes' rule.

5.4 The Estimation Problem for Red

We assume that Red observes aircraft existence and location. We assume that a Red detection of an aircraft by a SAM requires that that particular SAM radar be on. If the SAM is on, then detection of any given aircraft is a random event where the detect probability depends on the distance from the SAM to the aircraft. In particular, we take the detect probability to be

$$p_d^r = \frac{1}{1 + (r/d)^2}$$

where r is the distance from the SAM to the aircraft, and d is a scaling parameter. We assume that if there is a detection, then Red also obtains a position observation. In particular, we simplify the problem for the purposes of this study by having a position observation with spherical error covariance rather than taking into account the details of

range, azimuth and elevation components of the observation, as well as the possibility of doppler measurements. We also place the entire problem in a two-dimensional space (no altitude component).

The Red player then uses a sub-optimal filter to track the Blue aircraft. The Blue State in the Red filter model consists of a situation state, S_R taking values in $\{1, 0, B\}$ for "in air", destroyed and at base, as well as a position vector and a velocity vector. Ideally, the filter would have a position/velocity estimate and covariance corresponding to each possible path of S_R up to the current time. Of course, this explodes exponentially as time moves forward, and so we take the standard approach of only carrying a finite number of these along. In particular, at each time step, the filter is reduced to three probabilities for S_R, $P^s(t)$; specifically, $P^s(t)$ is a three-vector with for instance, $P^s_1(t)$ being the probabilty that the aircraft is such that $S_R(t) = 1$ (i.e. the probability that the aircraft is in the air). Corresponding to each element of this three vector is a mean position/velocity vector and a corresponding covariance (a 4×4 matrix).

The position/velocity means and covariances are updated by the standard Kalman filter equations. More specifically, we assume for simplicity that the SAM uses a straightforward state space model for an aircraft's dynamics:

$$x(t + \Delta_t) = x(t) + \Delta_t v(t) + w_x(t) \tag{4}$$
$$v(t + \Delta_t) = v(t) + w_v(t), \tag{5}$$
$$\tag{6}$$

in which x and v denote the aircraft position and velocity vectors, and w_x and w_v denote plant noise in the position and velocity models. For the observation updates, we assume each SAM radar observes the aircraftwhich they detect, and that the Red defense pools the information into an observation vector $Y(t)$. For an aircraft at position $x(t)$, the components of $Y(t)$ are

$$Y_i(t) = x(t) + \varepsilon_i(t)$$

where the i subscript indicates the i^{th} SAM radar's observation of position. We include the simplest case here to begin to understand the effect of partial information on game-theoretic controls.

Finally, note that we do not consider the track association problem here. In other words, we assume that when the SAMs receive an observation, they correctly associate that observation with the corresponding aircraft that was observed. The track association problem is not relevant to the study we are making here, and the additional complication would be detrimental to our investigation of the C^2 problem at hand.

5.5 Blue Control under Imperfect Information

Having defined our information states in terms of probabilistic models, we now proceed with the tasks of developing estimators and integrating observers into the control system.

The goals of this research project involve developing and understanding control strategies for air operations. Our particular interest has been in strategies that counter intelligent opponents in a robust way. Toward that end, we seek here to include these robustness consideration into the estimation problem: that is, we seek estimators that balance accuracy of estimation with the risk associated with conducting the air operation.

Traditional approaches to output feedback control involve the separation principle, or the certainty equivalence principle. The basic idea is to develop feedback controls for the full state feedback problem and apply them *replacing the state with a state estimator.* The most common estimator used is the maximum likelihood estimator. It is well known that, for linear control systems with quadratic cost criteria, the separation principle control coincides with the optimal control.

Another Certainty Equivalence Principle exists in robust control. We have applied a generalization of this estimator, discussed below, that allows us to tune the relative importance between the likelihood of possible states and the risk of being in those states. Let us motivate this in a little more detail.

The problem of Stochastic Games under Partial Observations (without resorting to replacement of state by the information state, which is hugely higher dimensional – infinite-dimensional in continuous state problems) is NOT solved. The Certainty Equivalence Principle (sometimes true – usually not) allows one to separate the the filtering and control components to some extent. In deterministic games under partial information, the Certainty Equivalence implies that one should use the optimal control corresponding to the state given by

$$\overline{x} \in \operatorname{argmax} [P(t,x) + V(t,x)]$$

where P is the information state and V is the value function (assuming uniquueness of the argmax of course). Here, 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., and McEneaney ([21], [20], [28], [29].) The next three paragraphs discuss the mathematics which lead to the heuristic for the algorithm described in the fourth paragraph below. Readers uninterested in these details should skip directly to the fourth paragraph below.

The deterministic information state is very similar to the *log* of probability density in stochastic formulations for terminal/exit cost problems. (In fact, this is exactly true for certain linear/quadratic problems.)

A risk-averse stochastic control problem is given by

$$\begin{aligned}
&d\xi_t = f(\xi(t), u(t))\, dt + \sqrt{\varepsilon}\sigma(\xi(t))\, dW_t \\
&\xi_0 = x \\
&J_\varepsilon(x,u) = \varepsilon \log \mathrm{E}\left\{ e^{\frac{1}{\varepsilon} L(\xi(\cdot), u(\cdot))} \right\} \\
&V_\varepsilon(x) = \inf_u J_\varepsilon(x,u).
\end{aligned}$$

This risk-averse stochastic control problem is equivalent to the stochastic game:

$$\begin{aligned}
&d\xi_t = [f(\xi(t), u(t)) + \sigma(\xi(t))w(t)]\,dt + \sqrt{\varepsilon}\sigma(\xi(t))\,dW_t \\
&\xi_0 = x \\
&J_\varepsilon(x, u, w) = \mathrm{E}\left\{L(\xi(\cdot), u(\cdot)) - \frac{1}{2}\|w\|^2\right\} \\
&V_\varepsilon(x) = \inf_{\text{"}u\text{"}} \sup_w J_\varepsilon(x, u, w).
\end{aligned}$$

Both have the same Dynamic Programming Equation:

$$\begin{aligned}
0 = V_t &+ \varepsilon \sum_{i,j} (\sigma\sigma^T)_{i,j} V_{x_i,x_j} \\
&+ \inf_u \Big\{ [f(x,u)]^T \nabla V + L(x,u) \Big\} \\
&+ \sup_w \Big\{ [\sigma(x)w]^T \nabla V - \frac{1}{2}|w|^2 \Big\} \\
= V_t &+ \varepsilon \sum_{i,j} (\sigma\sigma^T)_{i,j} V_{x_i,x_j} + \inf_u \Big\{ [f(x,u)]^T \nabla V + L(x,u) \Big\} \\
&+ \frac{1}{2}[\nabla V]^T \sigma\sigma^T \nabla V.
\end{aligned}$$

It is by now well-known that risk-averse control converges to a deterministic game as $\varepsilon \downarrow 0$ ([11], [12], [13], [32]). All of this lends credibility to a study of the use of the above Certainty Equivalence approach for our problem (although it will be sub-optimal).

In the stochastic linear/quadratic problem formulation, the information state at any time, t, is characterized as a Gaussian distribution, say

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

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

$$P(t,x) = -\tfrac{1}{2}(x - \widehat{x}(t))^T Q(t)(x - \widehat{x}(t)) + r(t).$$

Interestingly, Q and C^{-1} satisfy the same Riccati equation (or, equivalently, Q^{-1} and C satisfy the same Riccati equation). $\widehat{x}$ and $\overline{x}$ satisfy identical equations as well. Therefore, $P(t,x) = \log[p(t,x)] +$ "time-dependent constant".

The above three paragraphs form the (partially) heuristic argument behind our algorithm. This algorithm is: apply state feedback control at

$$\text{argmax}\{\log[p(t,x)] + \kappa V(t,x)\}$$

where p is the probability distribution based on the above observation process and filter for Blue (or Red), and V is state feedback stochastic game value. Here, $\kappa \in [0, \infty)$ is

a measure of risk-aversion. Note that $\kappa = 0$ implies that one is employing a maximum likelihood estimate in the state feedback control (for the game), i.e.

$$\text{argmax}\{\log[p(t,x)]\} = \text{argmax}\{p(t,x)\}.$$

Note also (at least in linear-quadratic case where $\log p(t,x) = P(t,x)$ (modulo a constant), $\kappa = 1$ corresponds to the deterministic game Certainty Equivalence Principle, i.e.

$$\text{argmax}\{P(t,x) + V(t,x)\}.$$

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.

Assuming Certainty Equivalence allows us to use our earlier experimental result (see above sections): The optimal Blue strategy is always either rollback or fly-over. This reduces our search over Blue controls by an order of magnitude for our problem.

5.6 Numerical Experiments with Robust Blue Control under Imperfect Information

We have developed a simulation for the partially observed problem, which uses as an input the full state feedback controls computed using the software described in previous sections. However, now for the Blue controller, we combine the estimator and controller via the risk-averse technique described in the previous section. The simulation generates observations and battle outcomes according to the appropriate probability models and evolves the information states as the engagement progresses. The controllers observe the state, and input the controls accordingly.

We note here that the new control software for the partially observed stochastic game allows any number of SAMs up to 6, that is without needing the hierarchical control. (The simulator and estimator do not have any hard bounds on the number.) It also allows any number (up to number of grid points) of possible SAM/Emitter locations. However, a practical detail is that one needs to store "tables" for each possible geometry distillation of 6 SAMs. The maximum number of Blue aircraft and Red targets is two each (without the hierarchical controller). Recall that the geometry distillation describes which Red entities lie under which other SAM umbrellas. (Many different geometries may have the same distillation.)

The example below has only a few SAMs and decoys, but this is not necessary. One has the standard exponential growth in computation with number of aircraft (or packages), number of SAMs and number of targets. One has slower growth in real-time computation with number of decoys.

Figure 5.1, which is a snapshot of the simulation in progress, illustrates the process. Included in this image are aircraft (in black), SAM sites with radar (in pink if on, red if off), emitter sites (in cyan if on, blue if off), and targets (in magenta). The black

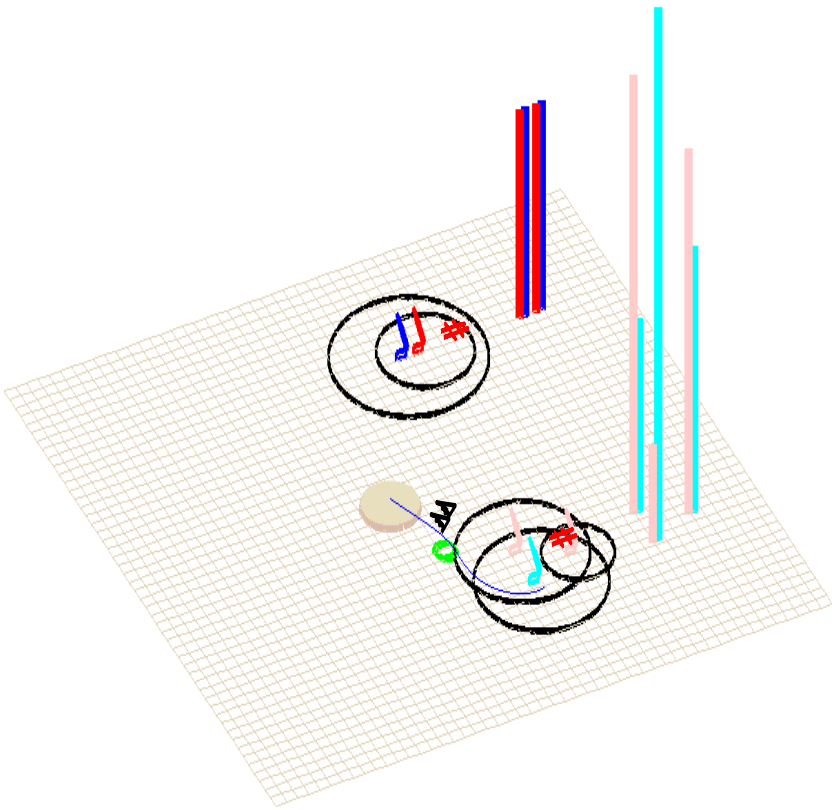


Fig. 5.1

circles indicate kill radii for the SAMs. The bar graphs to the right of the battle cartoon indicate the likelihoods for Blue for each site: Blue must estimate based on observations the probability that a site is a SAM radar or an emitter decoy. Specifically, the red/pink bars indicate the probability that the site is a SAM, and the blue/cyan bars indicate the probability that it is an emitter. (We also allow a probability that there is nothing at that location.) Also pictured are green circles which give the 2σ radii of the aircraft position estimates for the Red information state.

Applying this simulation for many Monte Carlo engagements, we can assess the expected value for a particular scenario. In Figure 5.2, we have selected a scenario which has 3 SAM sites, 2 emitters, and 2 aircraft attacking two targets. Running the simulation for 2000 Monte Carlo samples, we can assess the impact of the risk-averse estimator weight parameter κ on the outcome. The plot below shows that there is an optimal value in between applying the straight maximum likelihood estimator and the $\kappa \to \infty$ approach (which ignores all observations – assuming the worst-case state). Applying the traditional separated controller/estimator approach ($\kappa = 0$) produces reduced performance, which means that the Blue player is more likely to lose aircraft under this approach than under the risk-averse combined controller/estimator of the previous section, which takes a more game-theoretic, risk–averse approach. Note that the horizontal axis is on a log scale, so

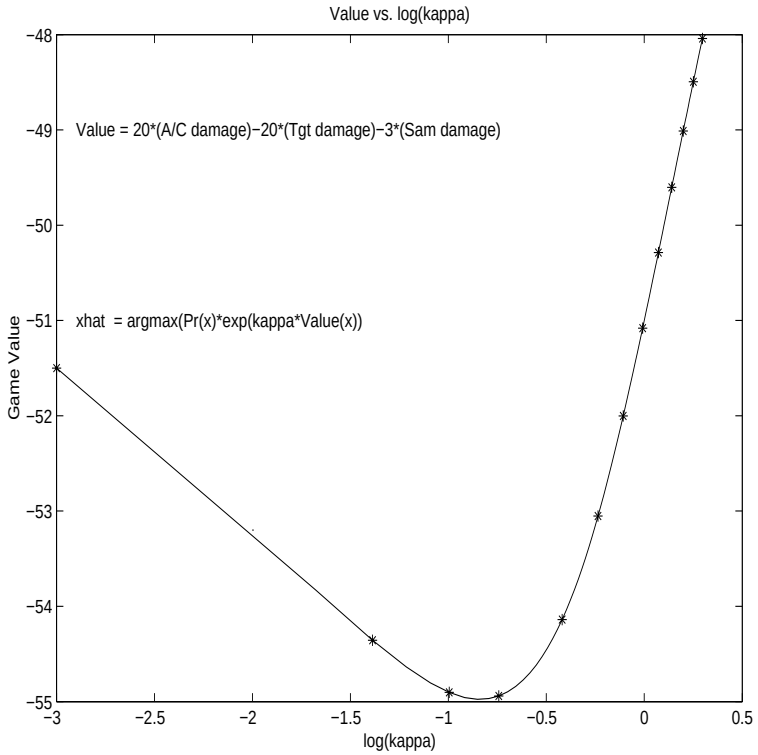


Fig. 5.2

that the minimum in κ is rather broad.

6 A State Estimator for the upper Hierarchical Levels

In this section, we consider a filter for a higher level in the problem hierarchy in which the Red state is not completely available. Instead, the state with additive noise is observable.

Let X_n denote the number of SAMs (or state of targets) in a given region at time n, which is not directly observable. One only observes $Y_n = X_n + \text{noise}$. The objective is to estimate X_n using information $\{Y_0, Y_1, \ldots, Y_n\}$.

Let $X_n \in \{0, 1, \ldots, m\}$ be a Markov chain with transition probability matrix $P = (p_{ij})_{(m+1)\times(m+1)}$. The observation process is

$$Y_n = X_n + \sigma(n, X_n)W_n, \ n = 0, 1, 2, \ldots,$$

where σ is a function of n and x, and $\{W_n\}$ is a sequence independent Gaussian $N(0, 1)$ random variables.

Algorithm.

Let $\pi_i(n) = P(X_n = i|Y_0, Y_1, \ldots, Y_n)$. Then for $i = 0, 1, \ldots, m$,

$$\begin{cases} \pi_i(n+1) = \pi_i(n) + (P-I)^*\pi_i(n) \\ \qquad\qquad +\pi_i(n)\left(\dfrac{i - \sum_{j=0}^m j\pi_j(n)}{\hat{\sigma}(n)}\right)\left(\dfrac{Y_{n+1} - \sum_{j=0}^m j\pi_j(n)}{\hat{\sigma}(n)}\right) \\ \qquad \pi_i(0) = p_i^0 \text{ given,} \end{cases}$$

where

$$p_i^0 = P(X_0 = i|Y_0),$$
$$\hat{\sigma}(n) = \sum_{j=0}^m \sigma(n,j)\pi_j(n)$$

and

$$(P-I)^*\pi_i(n) = \sum_{j=0}^m (p_{ji} - \delta_{ji})\pi_j(n)$$

and $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Remark. This algorithm is derived from an exact optimal filter in a continuous-time model. It provides fast and in the meantime fairly reliable estimates for finite-state Markov chain.

Using the conditional probabilities, we define the maximum likelihood estimate of X_n

$$X_n^{\max} = i \text{ if } \pi_i(n) = \max\{\pi_j(n) : \ j = 0, 1, \ldots, m\}.$$

We also define the conditional mean

$$\widetilde{X}_n = \sum_{j=0}^m j\pi_j(n)$$

and its integer version

$$\widehat{X}_n = i \text{ if } i - \frac{1}{2} \le \widetilde{X}_n < i + \frac{1}{2} \text{ for } i = 0, 1, \ldots, m.$$

Remark. Note that it is difficult to obtain a recursive form for $\widetilde{X}_n$ because the underlying filtering problem is nonlinear.

Numerical Experiments.

In our numerical simulations, we take $m = 8$ and

$$P = \begin{pmatrix} 0.3 & 0.3 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.2 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.5 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.6 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.5 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0.6 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.5 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.7 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 \end{pmatrix}$$

We consider $\sigma(n,x) = 0.2x + \sigma_0$. We fix $X_0 = 8$ and take the initial distribution (with given i_0) to be of the form $p^0_{i_0} = 1$ and $p^0_i = 0$ for $i \neq i_0$.

First, choose $i_0 = 8$ and vary σ_0, i.e., we have precise initial estimate on $\{p^0_i\}$. The dependence on observation noise is given in Table 5.

Table 5. Dependence on observation noise.

σ_0	0.1	0.5	1	1.5	2
$E(X_n - \widehat{X}_n)^2$	0.929	1.018	1.163	1.306	1.446
$E(X_n - X_n^{\max})^2$	0.711	0.840	1.024	1.244	1.447

It is clear that the larger the observation noise, the greater the estimation error.

In Figures 1.1 and 1.2, sample paths and the corresponding conditional probabilities are plotted (with $\sigma_0 = 0.5$). Then in Figures 2.1 and 2.2, we plotted the same functions with $\sigma_0 = 2$. In this case, the observation noise is much greater (as seen on Fig 2.2). However, the filtering algorithm performs quite well.

It is interesting to see (Figures 1.2 and 2.2) how the conditional probabilities evolve following the state jumps.

Then, we fix $\sigma_0 = 0.5$ and vary i_0. We examine how the algorithm works with poorly chosen initial distributions.

Table 6. Dependence on initial probabilities.

i_0	0	1	2	3	4	5	6	7	8
$E(X_n - \widehat{X}_n)^2$	3.003	2.678	2.561	1.950	1.550	1.310	1.137	1.036	1.018
$E(X_n - X_n^{\max})^2$	2.380	2.181	2.056	1.558	1.233	1.057	0.915	0.851	0.840

As can be seen from this table that error increases continuously as the choice i_0 moves away from the true value $i_0 = 8$. However, as shown in Figures 3.1 and 4.1, these errors disappear quickly in both cases with $\sigma = 0.5$ and $\sigma = 2$.

Sensitivity Tests.

First we consider sensitivity with respect to observation noise

$$\sigma(n,x) = (0.2 + \varepsilon)x + (0.5 + \varepsilon).$$

We fix $i_0 = 8$.

Table 7. Perturbations in Observation Noise

ε	0.5	0.4	0.3	0.2	0.1	0.05	0.01	0
$E(X_n - \widehat{X}_n)^2$	1.221	1.162	1.112	1.065	1.027	1.021	1.013	1.018
$E(X_n - X_n^{\max})^2$	1.496	1.310	1.153	1.022	0.895	0.855	0.838	0.840

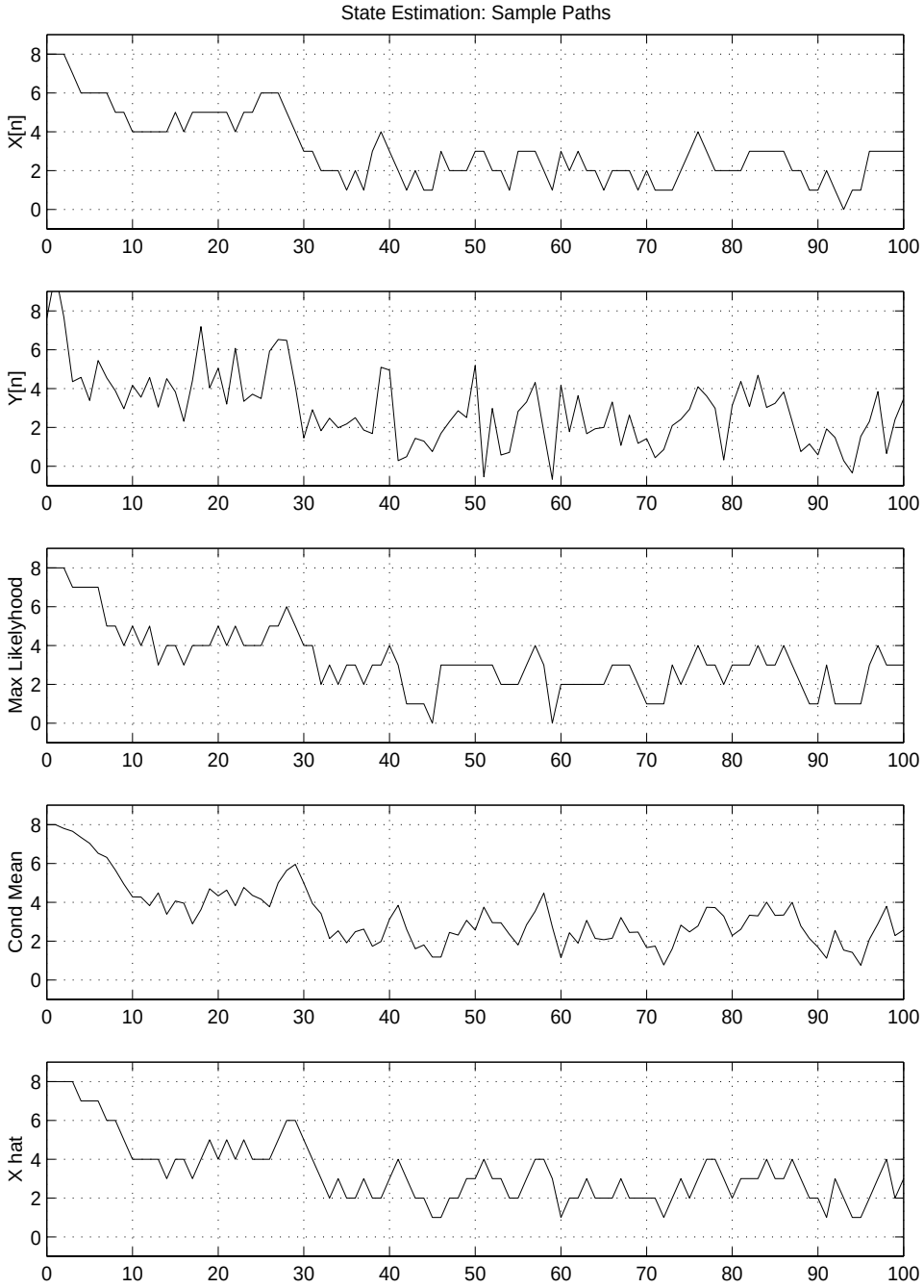


Fig. 1.1. $i_0 = 8$ and $\sigma = 0.5$.

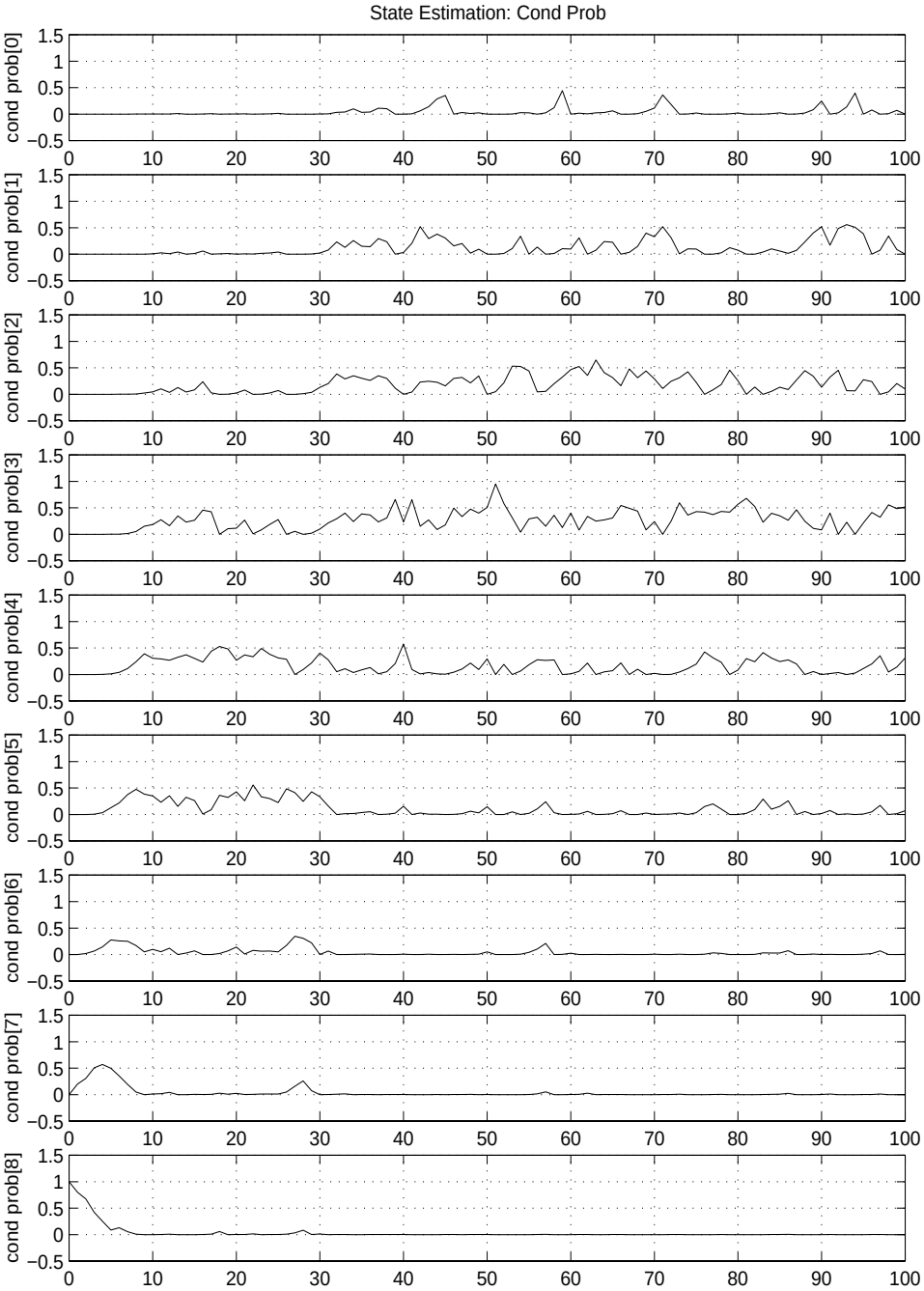


Fig. 1.2. $i_0 = 8$ and $\sigma = 0.5$.

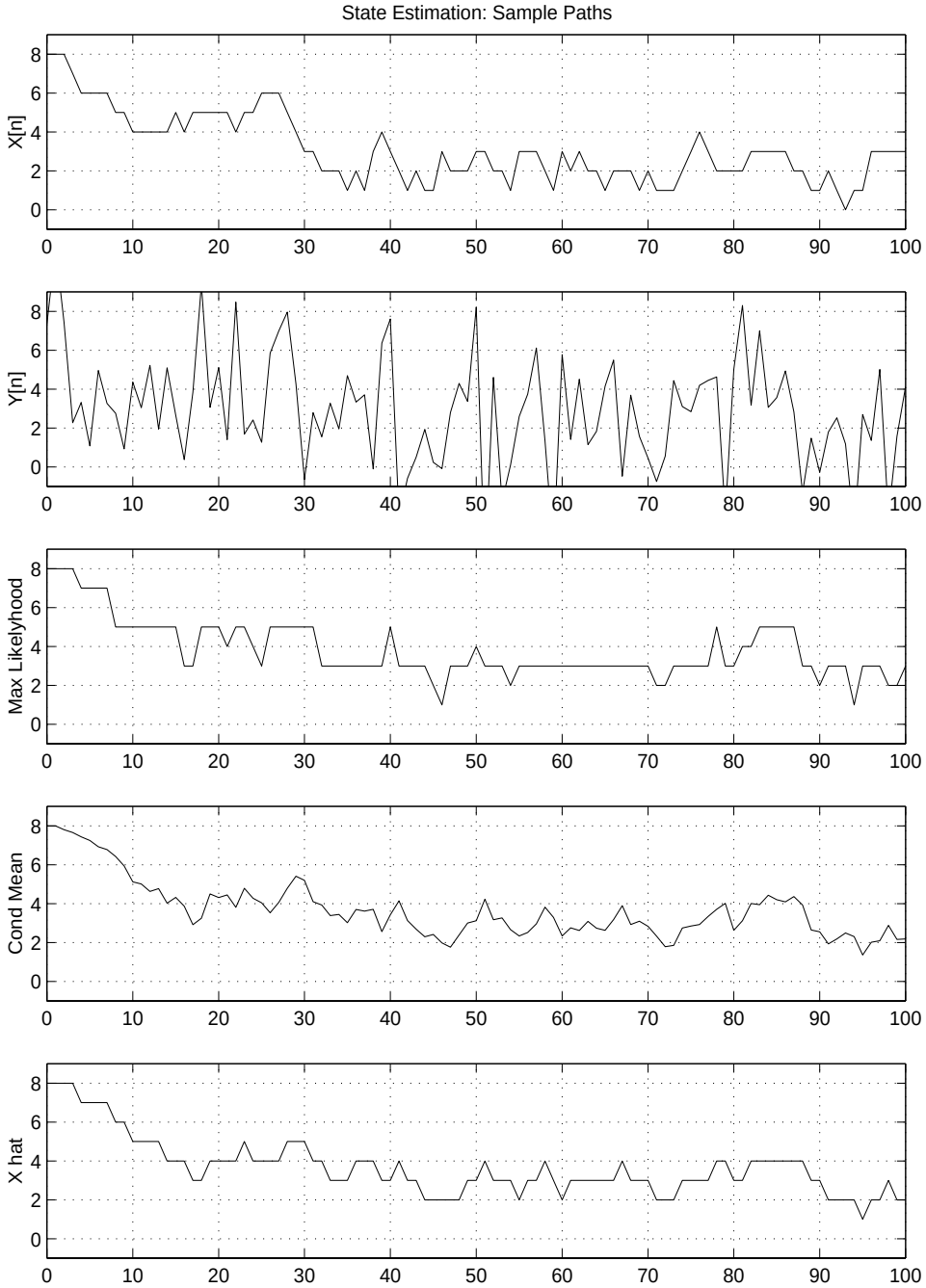


Fig. 2.1. $i_0 = 8$ and $\sigma = 2$.

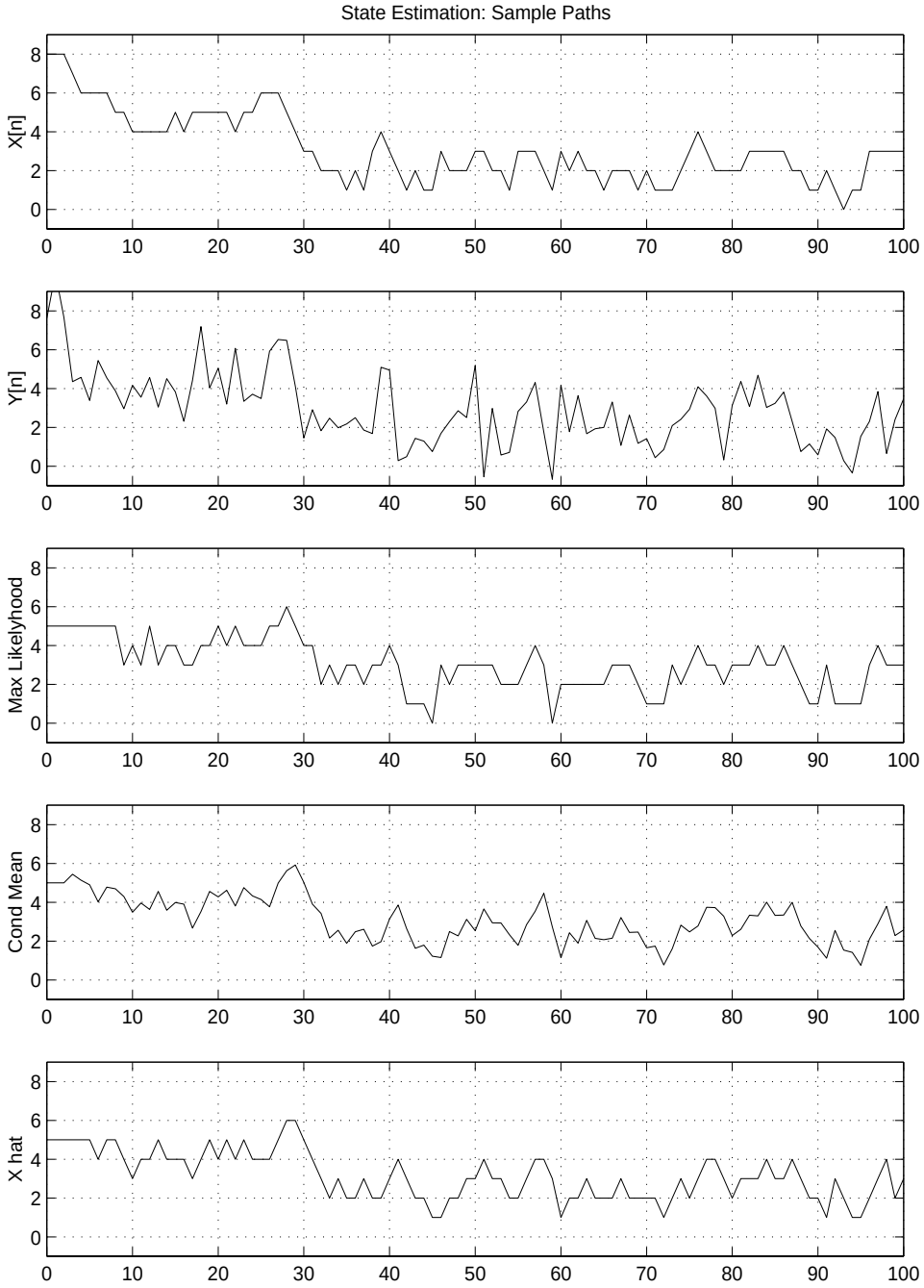


Fig. 3.1. $i_0 = 5$ and $\sigma = 0.5$.

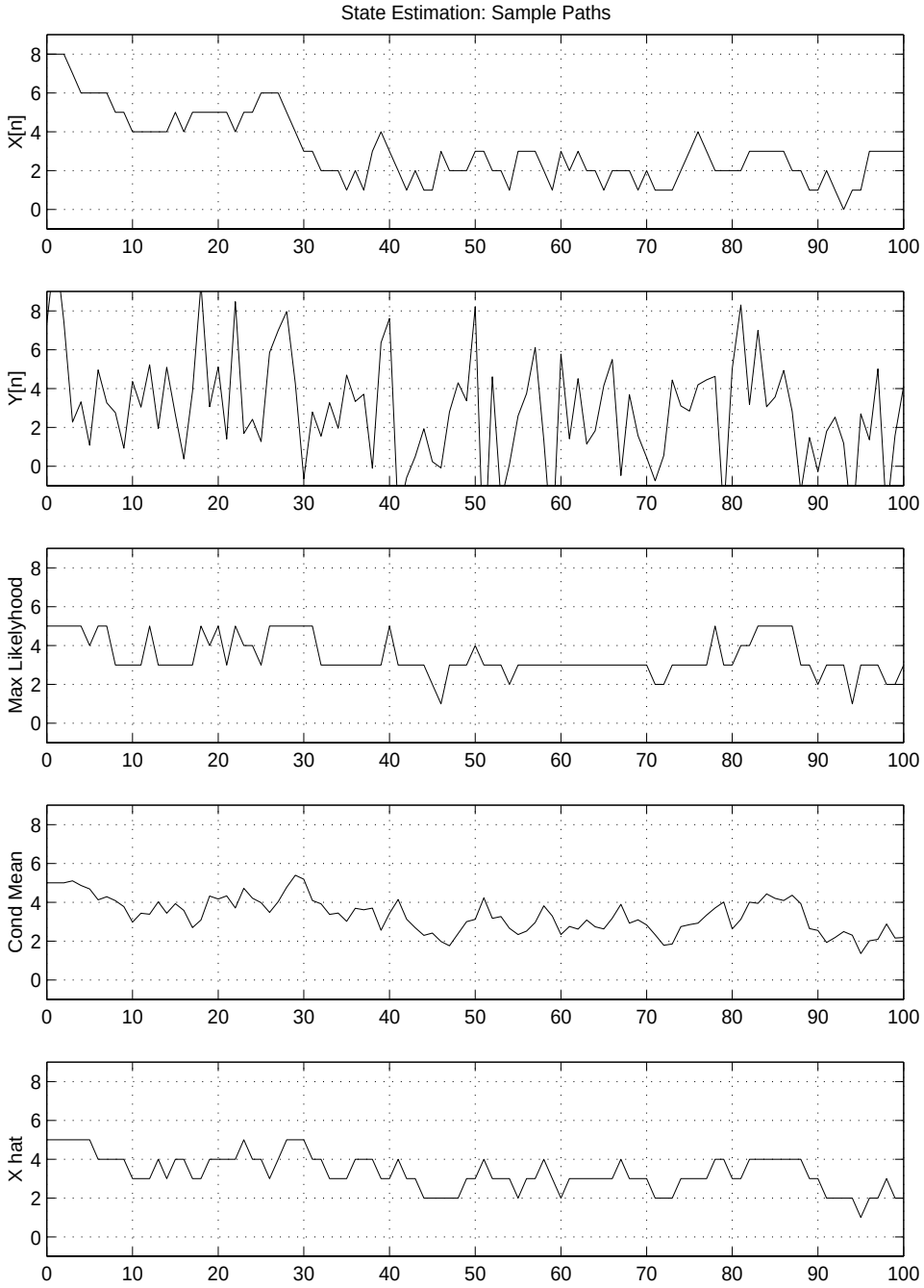


Fig. 4.1. $i_0 = 5$ and $\sigma = 2$.

Next we fix $\sigma(n,x) = 0.2x + 0.5$ and $i_0 = 8$. We add a perturbation to transition matrix P:

$$P^{\varepsilon} = \begin{pmatrix} 0.3+\varepsilon & 0.3-\varepsilon & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0.3-\varepsilon & 0.2+\varepsilon & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.5-\varepsilon & 0.2+\varepsilon & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.6-\varepsilon & 0.2+\varepsilon & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.5-\varepsilon & 0.3+\varepsilon & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0.6-\varepsilon & 0.2+\varepsilon & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.5-\varepsilon & 0.2+\varepsilon & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.7-\varepsilon & 0.2+\varepsilon \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2+\varepsilon & 0.8-\varepsilon \end{pmatrix}$$

The numerical results are as follows:

Table 8. Perturbations in Transition Probabilities.

ε	0.2	0.1	0.05	0.01	0
$E(X_n - \widehat{X}_n)^2$	1.278	1.002	0.999	1.016	1.018
$E(X_n - X_n^{\max})^2$	1.141	0.830	0.821	0.835	0.840

These results demonstrate that the algorithm is robust in the sense that the output depends on parameters in a continuous fashion. The implication is that one does not have to have exact value of various parameters σ and P. Approximations will do nearly as good.

In the completely observable case, the optimal allocation for blue troop is given by

$$(U_1^a(n), U_2^a(n)) = U_*^a(I_X(n)) = U_*^a(X_1^a(n), X_2^a(n), X_1^s(n), X_2^s(n), T_1(n), T_2(n)).$$

When the states for targets and SAMs are not completely observable, we consider the model

$$\begin{aligned} Y_1^T(n) &= T_1(n) + \sigma_1^T w_1^T(n), \\ Y_2^T(n) &= T_2(n) + \sigma_2^T w_2^T(n), \\ Y_1^s(n) &= X_1^s(n) + \sigma_1^s w_1^s(n), \\ Y_2^s(n) &= X_2^s(n) + \sigma_2^s w_2^s(n). \end{aligned}$$

Then our nonlinear filtering scheme gives

$$(\widehat{X}_1^s, \widehat{X}_2^s, \widehat{T}_1, \widehat{T}_2)$$

One may replace the state variables be these estimates and use

$$(\widehat{U}_1^a(n), \widehat{U}_2^a(n)) = U_*^a(X_1^a(n), X_2^a(n), \widehat{X}_1^s(n), \widehat{X}_2^s(n), \widehat{T}_1(n), \widehat{T}_2(n))$$

to control the actual system. Such approach typically leads to close to optimal policies; see [35].

7 A Means of Evaluating Various Approaches to the C^2 Problem

Command and Control (C^2) problems in the military domain are now being addressed via modern control techniques. More specifically, one may view the battle as a plant to be controlled toward some goal. Obviously, the plant is an extremely large system involving both man and machine components interacting over some changing physical space. The level of detail and/or abstraction that commanders at various levels face is, of course, highly variable. The choice of plant model for any such system is quite unclear. Further, a commander faced with some plant state may have an array of controllers available, each suggesting an action different from the others in some way (see, for instance, [6], [18], [16], [22], [33], [34], [5], [17], [37]). The question is how to choose among such an array of controllers. One may consider the very large space of all plants consistent with the battle under consideration. By plants one of course means not just the variables, but also the dynamics by which the system propagates. With this view, the question becomes: in which region of (true) plant space should a given controller be used in preference to another? Although this is obviously a very difficult problem, one may be able to make some progress in a rigorous fashion. The resulting algorithms (selecting controllers dependent on location in plant space) might best be termed a meta–controller.

Even in the most general sense, the best control formulation may not be at all clear. For instance, one controller might choose to assign value to the elimination of enemy surface–to–air missiles (SAMs), while another may eliminate SAMs as a means of reaching a goal target (whose elimination may have value) at minimum risk; that is, there may not be any term in the cost criterion corresponding to the value of a SAM. As another example, one controller may focus solely on attrition while another may directly assign value to geographical position of assets. The general form of the cost may be different between controllers. For instance, one may attempt to minimize a discounted cost criterion while another may attempt to control to some desired exit criterion. Further still, and even more interesting, one controller may model certain stochastic elements of the dynamics while another may focus on a deterministic game formulation.

This section will only begin to address the problem. Clearly, when military planners consider the various approaches and the arguments of each approache's ardent proponents, decisions will need to be made regarding the best approach. Consequently, we consider here an initial outline of the problem, and some potential tools. It is hoped that more progress can be made so that the military planners can have some aids in determining the "best" approaches.

We remark that the approach to be described here is one where the true plant is likely quite different from the controller model since nonlinear control problems can only be solved (even numerically) for relatively low-dimensional systems. However, one can generate very complex computer simulations of a true system, and this has certainly been done for military C^2 problems. Thus, the approach described below would be used with

a "true" plant being generated by a very detailed computer simulation with the dynamic models being used by the controllers being much simpler. Throughout, we will refer to this high-fidelity simulation as the real world state; this enables one to use Monte Carlo techniques to study the real world outcomes when appropriate.

The optimization analysis at a general level is rather simple, and appears in Subection 7.1. In Subsection 7.2, this is extended to a switching meta-controller. In Subsection 7.3, extensions to game models are discussed briefly. We consider a small example in Subsection 7.4 to indicate that the concepts are not vacuous.

7.1 Optimization Analysis

We now display the relatively simple analysis which could be used to compare the effectiveness of two controllers which may have completely different plant models with different criteria which are being optimized.

One must have a true plant whose state is represented at time t as $X^w(t)$ taking values in some set $\mathcal{X}^w$. The controls which the commander could use in this true plant (again actually a high fidelity simulation of the real world) are denoted as $u^w(t)$. There must exist some dynamics for this true plant. Rather than specify particular dynamics, we will attempt to keep the discussion rather general, allowing a variety of dynamic models such as deterministic models

$$\frac{dX^w}{dt} = f(t, X^w(t), u^w(t)) \tag{7}$$

and

$$X^w(t_{k+1}) = f(t_k, X^w(t_k), u^w(t_k)) \tag{8}$$

where in the latter case, we will assume the state is defined over continuous time as $X^w(t) = X^w(t_k)$ for all $t \in (t_{k-1}, t_k]$. We will assume that there exist unique solutions for all feedback controls $u^w(\cdot)$ to be supplied by the control algorithms. We will also allow stochastic analogues of these dynamics such as

$$dX_t^w = f(t, X^w(t), u^w(t))\, dt + \sigma(t, X^w(t))\, dB_t \tag{9}$$

and

$$X^w(t_{k+1}) = X^w(t_k) + f(t_k, X^w(t_k), u^w(t_k)) + \sigma(t, X^w(t))W(t_k). \tag{10}$$

Here, $B.$ would be a vector-valued Brownian motion, and $W(\cdot)$ would be a vector-valued random sequence. Again, we assume that all conditions for existence and uniqueness of solutions hold for any controls, u^w, generated by the control algorithms. (For instance, in the diffusion case, we presume the control designers restrict themselves to progressively measurable controls, but this will not be the main focus of this paper, so we ignore the details for now.) Again, this paper will only attempt to lay out a problem that has

appeared in the C^2 area in a general way, and so we minimize discussion of the associated mathematical machinery.

Now, if one is going to apply a control algorithm, say algorithm i, one must have a model of the true plant with which the controller will be computed. Let the state (at time t) in this model for control algorithm i be denoted by $X^i(t)$, taking values in $\mathcal{X}^i$. Then, in order to compute the corresponding control, one must specify a mapping from the real world state to the model state. That is, the control designer must specify a well-defined mapping

$$M^i : \mathcal{X}^w \to \mathcal{X}^i. \tag{11}$$

Thus, given plant trajectory $X^w([t_0, t])$ (where t_0 will denote the initial time), the control at time t will be designed on the basis of $X^i([t_0, t])$. Then, via some algorithm (algorithm i in this case), the controller will compute the control to be applied at time t, $u^i(t)$ with $u^i(t) \in \mathcal{U}^i$, or possibly $\mathcal{U}^i_{X^i(t)}$ in the state-dependent control set case. Note that one may have $u^i(t) = F^i(X^i(t))$ or possibly, $u^i(t) = F^i(X^i([t_0, t]))$ if the control depends on the entire trajectory rather than just the current state. (For now, we do not specifically denote an observation process.)

Next, to actually apply the controls back in the real world, the designer must also specify some mapping from $\mathcal{U}^i$ (or $\mathcal{U}^i_{X^i(t)}$) into $\mathcal{U}^w$ (or $\mathcal{U}^w_{X^w(t)}$ in the state-dependent control set case). That is, the designer must specify

$$N^i : \mathcal{U}^i \to \mathcal{U}^w \tag{12}$$

or in the state-dependent control set case,

$$N^i_{X^w(t)} : \mathcal{U}^i_{X^i(t)} \to \mathcal{U}^w_{X^w(t)} \tag{13}$$

where the subscript on N is necessary since the specification of the mapping must be such that the range is restricted to $\mathcal{U}^w_{X^w(t)}$. Thus, (in the case of state-dependent control set)

$$u^w_i(t) = N^i_{X^w(t)} \circ F^i \circ M^i(X^w(t)) \tag{14}$$

if the controller only depends on the current state, and

$$u^w_i(t) = N^i_{X^w(t)} \circ F^i \circ \{M^i[X^w(r)]([t_0, t])\} \tag{15}$$

otherwise (where r is a dummy variable).

We will assume throughout that the commander may supply an exit set, $\mathcal{E}^w$, and certainly will supply an exit time

$$\tau^w \doteq \min\{T, \inf\{t : X^w(t) \in \mathcal{E}^w\}\},$$

where $T < \infty$ (and obviously $\tau^w = T$ if no exit set is specified). In the simplest situation, the commander or planner may have a very specific cost criterion in mind, say

$L^w(X^w([t_0, \tau^w], u^w([t_0, \tau^w])$ which they may wish to minimize. For example, let us suppose that the dynamics of the real world are stochastic, and further, let us suppose that one wishes to minimize a criterion such as

$$\mathcal{L}^w(t_0, x_0^w, \mathcal{A}^i) \doteq E\left\{L^w(X^w([t_0, \tau^w], u^w([t_0, \tau^w])\right\}$$

with $X^w(t_0) = x_0^w$ where $\mathcal{A}^i$ represents the triple $\mathcal{A}^i = (M^i, F^i, N^i)$. Then the problem is simple. For instance, suppose one only needs to compare two control algorithms, $\mathcal{A}^1$ and $\mathcal{A}^2$. The problem is reduced to merely comparing $\mathcal{L}^w(t_0, x_0^w, \mathcal{A}^1)$ and $\mathcal{L}^w(t_0, x_0^w, \mathcal{A}^2)$ at the initial time to see which yields the minimum expected payoff. (Note that there is actually a time-dependent generalization to this where one may use such a criterion as a switching criterion in a meta-controller. Some initial development in this direction appears in the next section.) One might remark that the minimizing control algorithm might vary depending on the initial state.

Unfortunately, the criterion which should be minimized may not be obvious. The problem under consideration may be only a small part of an overall conflict, and the criterion may be quite loosely defined, as mentioned in the introduction. At the early stages of an examination of this overall problem of applying control in a C^2 environment, the military will not give a specific criterion to be minimized. Further, the plants to be studied may be quite variable – encompassing a wide variety of military scenarios. The control designers themselves must propose criteria which the controllers will minimize (based, of course, on communication with operational experts). Note that τ^w may not correspond to the exit time for the control models. It is required that each controller have some cost criterion which it is attempting to minimize. Let the cost be $L^i(X^i(\cdot), u^i(\cdot))$. This cost must be well-defined for all possible paths $X^i(\cdot)$. In particular, if there is an exit set $\mathcal{E}^i$ and exit/terminal time τ^i, then L^i must be defined for paths such that $\tau^w < \tau^i$. Alternatively, if X^i enters $\mathcal{E}^i$ at some time $t^i < \tau^w$, then the controller must define some "zero control", u_0^i to be applied while $X^i(t) \in \mathcal{E}^i$. Lastly, we will assume that the range of L^i, $\mathcal{R}(L^i)$, to satisfy $\mathcal{R}(L^i) \subseteq [0, 1]$. (Note that there are a variety of simple mappings to achieve this range condition if it is not natural.) We note that one would typically expect $L^i(X^i(\cdot), u^i(\cdot)) = 0$ to indicate an outcome with no Blue losses and total rout of Red forces, and vice-versa for $L^i(X^i(\cdot), u^i(\cdot)) = 1$, so a fixed range condition is quite natural in the context of a C^2 problem.

Now we return to the commander's predicament in the absence of an L^w. Let $X_i^w(\cdot) : [0, \tau^w] \to \mathcal{X}^w$ be the path generated by control algorithm $\mathcal{A}^i$, i.e. by applying control u_i^w given by (14),(15) for all $t \in [0, \tau^w]$. Also suppose that $[N^i]^{-1}$ exists (and assume that N^i is not state-dependent). Then define the cost of considering control algorithm $\mathcal{A}^j$ according to the cost metric supplied by controller i denoted (in the deterministic case) as follows (where we drop the t_0, x_0^w arguments for simplicity)

$$\mathcal{L}^i(\mathcal{A}^j) \doteq L^i[\mathcal{M}^i(X_j^w(\cdot)), [\mathcal{N}^i]^{-1}(u_j^w(\cdot))] \tag{16}$$

where

$$\mathcal{M}^i(X_j^w(\cdot))(t) \doteq M^i(X_j^w(t)) \qquad \forall\, t \in [0, \tau^w] \tag{17}$$
$$[\mathcal{N}^i]^{-1}(u_j^w(\cdot))(t) \doteq [N^i]^{-1}(u_j^w(t)) \qquad \forall\, t \in [0, \tau^w].$$

Note that if the real world dynamics are random, with some underlying probability space $(\Omega, \mathcal{F}_t, P)$, then one actually has $X_i^w(\cdot) = [X_i^w(\cdot)](\omega)$ where $\omega \in \Omega$, and one should modify (16) to

$$\mathcal{L}^i(\mathcal{A}^j) \doteq E\left\{L^i[\mathcal{M}^i(X_j^w(\cdot)), [\mathcal{N}^i]^{-1}(u_j^w(\cdot))]\right\}, \tag{18}$$

in which case, $\mathcal{L}^i(\mathcal{A}^j)$ is the *expected* cost of considering control algorithm $\mathcal{A}^j$ according to the cost metric supplied by controller i. We remind the reader that this may depend on the initial state $X^w(t_0) = x_0^w$.

Now, one would typically expect

$$\mathcal{L}^i(\mathcal{A}^i) \leq \mathcal{L}^i(\mathcal{A}^j) \tag{19}$$
$$\mathcal{L}^j(\mathcal{A}^j) \leq \mathcal{L}^j(\mathcal{A}^i) \tag{20}$$

(although this can certainly be violated since the true dynamics do not necessarily correspond to those assumed by the controllers).

Let us suppose (19),(20) hold. Even in this situation, a commander (with no a priori preferences) could still construct an objective criterion for choosing between two control algorithms, $\mathcal{A}^1$ and $\mathcal{A}^2$. One obvious approach is to consider

$$\frac{\mathcal{L}^1(\mathcal{A}^1)}{\mathcal{L}^1(\mathcal{A}^2)} \quad \text{and} \quad \frac{\mathcal{L}^2(\mathcal{A}^2)}{\mathcal{L}^2(\mathcal{A}^1)}. \tag{21}$$

To eliminate technicalities, let us assume that all four of these costs in the ratios are nonzero. Then, if (19), (20) hold,

$$\frac{\mathcal{L}^1(\mathcal{A}^1)}{\mathcal{L}^1(\mathcal{A}^2)}, \ \frac{\mathcal{L}^2(\mathcal{A}^2)}{\mathcal{L}^2(\mathcal{A}^1)} \in (0, 1].$$

Then one can simply compute

$$d_{1,2} \doteq \frac{\mathcal{L}^1(\mathcal{A}^1)}{\mathcal{L}^1(\mathcal{A}^2)} - \frac{\mathcal{L}^2(\mathcal{A}^2)}{\mathcal{L}^2(\mathcal{A}^1)},$$

and one would choose $\mathcal{A}^1$ if $d_{1,2} < 0$ and vice-versa (with no preference if $d_{1,2} = 0$ of course).

However, if the number of controllers exceeds two, one can show that there may not be an optimal choice by this approach. That is, following the above procedure in the case of three controllers, one may construct a situation where $d_{1,2} < 0$, $d_{2,3} < 0$, and $d_{3,1} < 0$. Thus, this procedure is not useful.

On the other hand, in general, one can form a single criterion by which to judge the controllers from the set of criteria provided by the control designers, and can then evaluate

each control algorithm according to this combination criterion. Two obvious combination criteria are (in the case of N controllers) as follows.

$$\mathcal{H}_{max}(\mathcal{A}^i) \doteq \max_{1\le j\le N} \mathcal{L}^j(\mathcal{A}^i) = \max_{1\le j\le N} E\{L^j[\mathcal{M}^j(X_i^w(\cdot)), [\mathcal{N}^j]^{-1}(u_i^w(\cdot))]\},$$

(where expectation is included in case the system is stochastic) and

$$\mathcal{H}_{sum}(\mathcal{A}^i) \doteq \sum_{j=1}^{N} \mathcal{L}^j(\mathcal{A}^i) = \sum_{j=1}^{N} E\{L^j[\mathcal{M}^j(X_i^w(\cdot)), [\mathcal{N}^j]^{-1}(u_i^w(\cdot))]\}.$$

The optimal control algorithms for these two combination criteria are obviously given by

$$\overline{\mathcal{A}}_{max} \doteq \operatorname*{argmin}_i \mathcal{H}_{max}(\mathcal{A}^i) \quad \text{and} \quad \overline{\mathcal{A}}_{sum} \doteq \operatorname*{argmin}_i \mathcal{H}_{sum}(\mathcal{A}^i).$$

Note that if the ranges of the $\mathcal{L}^i$ are very different, one may choose to first re-scale the range by $\tilde{\mathcal{L}}^j(t_0, x_0^w, \mathcal{A}^i) \doteq \mathcal{L}^j(t_0, x_0^w, \mathcal{A}^i)/[\max_{x_0^w}\{\mathcal{L}^j(t_0, x_0^w, \mathcal{A}^j)\}]$ so as to normalize the payoffs.

7.2 Switching Meta–Controller

One may adapt the above optimization over control algorithms approach in order to develop a switching meta-controller. A few more definitions will be required.

In the previous section, a fixed control algorithm was chosen at the initial time, t_0, and this remained fixed until τ^w. Now we will allow the choice of control algorithm to switch as time moves forward. Let $M^i, \mathcal{M}^i, F^i, u^i, N^i, \mathcal{N}^i, u_i^w, X_i^w$ be defined as before. Since this paper is introductory in nature, let us suppose that the (possible) switching times are prespecified as $\{t_k\}_{k=0}^{\overline{K}}$ with $t_{k+1} = t_k + \Delta_t$ for $k \ge 0$ and t_0 still being the initial time. Also, let $\overline{K} > T/\Delta_t$ where we recall $\tau^w \le T$ regardless of control choice (and sample path in the stochastic case). In the case of the discrete-time dynamics of (8), (10), we suppose the times t_k coincide with the time-steps given in the dynamics.

Let us again first consider the case where the commander or planner has a specified cost criterion in mind. This would take the form $L^w(X^w([t_0, \tau^w]), u^w([t_0, \tau^w])$. We now modify this by fixing any $t_k < \tau^w$, and letting the cost accrued up to time t_k (i.e. over $[t_0, t_k)$) be denoted by

$$\underline{L}_{t_k}^w(X^w([t_0, \tau^w]), u^w([t_0, \tau^w]) = \underline{L}_{t_k}^w(X^w([t_0, t_k \wedge \tau^w)), u^w([t_0, t_k \wedge \tau^w))),$$

and the remaining cost to go over $[t_k, \tau^w]$ be denoted by

$$\overline{L}_{t_k}^w(X^w([t_0, \tau^w]), u^w([t_0, \tau^w]) = \overline{L}_{t_k}^w(X^w([t_k, \tau^w]), u^w([t_k, \tau^w]))$$

where $\overline{L}_{t_k}^w(X^w([t_k, \tau^w]), u^w([t_k, \tau^w]) = 0$ if $t_k > \tau^w$. Let $K = \sup\{k \le N : t_k < \tau^w\}$. Let $I_k = [t_k, t_{k+1})$ if $k < K$ and $I_K = [t_K, \tau^w]$. Also, (assuming $t_k < \tau^w$) let the cost accrued over time interval I_k be denoted by

$$\widehat{L}_{t_k}^w(X^w([t_0, \tau^w]), u^w([t_0, \tau^w]) = \widehat{L}_{t_k}^w(X^w(I_k), u^w(I_k))$$

where to be more specific, this will be the cost over $[t_k, t_{k+1})$ if $t_{k+1} < \tau^w$ and over $[t_k, \tau^w]$ otherwise. We assume

$$\begin{aligned}&\underline{L}^w_{t_k}(X^w([t_0, t_k \wedge \tau^w)), u^w([t_0, t_k \wedge \tau^w)) + \overline{L}^w_{t_k}(X^w([t_k, \tau^w]), u^w([t_k, \tau^w]) \\ &= L^w(X^w([t_0, \tau^w]), u^w([t_0, \tau^w]) \qquad (A1)\end{aligned}$$

for all k. Then, by $(A1)$,

$$L^w(X^w([t_0, \tau^w]), u^w([t_0, \tau^w]) = \left\{ \sum_{k=0}^{K} \widehat{L}^w_{t_k}(X^w(I_k), u^w(I_k)) \right\}.$$

Let

$$i_0 \in \operatorname*{argmin}_i \overline{\mathcal{L}}^w(t_0, x^w_0, \mathcal{A}^i) \doteq \operatorname*{argmin}_i E\{\overline{L}^w_{t_0}(X^w([t_0, \tau^w]), u^w([t_0, \tau^w]))\}.$$

Suppose the control algorithm so chosen at time t_0 is denoted by $\mathcal{A}^{i_0}$ with resulting control $u^w_{i_0}$. We define $\overline{u}^w$ over the time interval $[t_0, t_1 \wedge \tau^w)$ as $\overline{u}^w([t_0, t_1 \wedge \tau^w)) = u^w_{i_0}([t_0, t_1 \wedge \tau^w))$. Similarly, the resulting trajectory is $\overline{X}^w([t_0, t_1 \wedge \tau^w))$. For each j and $t \in [t_0, t_1 \wedge \tau^w)$, one may then obtain the image of this trajectory under controller j's map, M^j, as $\overline{X}^j(t) \doteq M^j(\overline{X}^w(t))$.

Although, the goal here is to lay out some general beginnings for the theory, in order to be more clear let us presume here that we have a stochastic system (as, for instance, given by (9)). We assume

$$\overline{X}^w(t_k) = \lim_{s \uparrow t_k} \overline{X}^w(s) \qquad \forall\, k > 0, \quad a.s. \qquad (A2)$$

Denote the control algorithm chosen for this first time interval as $\overline{\mathcal{A}}_{I_0} \doteq \mathcal{A}^{i_0}$. Then, using $(A2)$, $\overline{\mathcal{A}}_{i_0}$ defines $\overline{X}^w$ over $[t_0, t_1 \wedge \tau^w]$, and let us suppose $\tau^w > t_1$. Then, for time interval I_1, we may define, for each controller i, $u^i(t) = F^i \circ M^i[X^w_i(t)]$ and so feedback $u^w_i(t) = N^i_{\overline{X}^w(I_0) \cup X^w_i([t_1, t))} \circ F^i \circ M^i[X^w_i(t)]$ for all $t \in I_1$, where $X^w_i(t_1) = \overline{X}^w(t_1)$. Let

$$i_1 \in \operatorname*{argmin}_i E_{t_1, \overline{X}^w(t_1)}\{\overline{L}^w_{t_1}(X^w_i([t_0, \tau^w]), u^w_i([t_0, \tau^w]))\}$$

where the subscripts on the expectation indicate that it is conditioned on the state at time t_1. Then let

$$\overline{X}^w(t) = \begin{cases} X^w_{i_0}(t) & \text{if } t \in I_0, \\ X^w_{i_1}(t) & \text{if } t \in I_1 \end{cases}$$

and

$$\overline{u}^w(t) = \begin{cases} u^w_{i_0}(t) & \text{if } t \in I_0, \\ u^w_{i_1}(t) & \text{if } t \in I_1. \end{cases}$$

Lastly, let

$$\overline{\mathcal{A}}_{I_0 \cup I_1} \doteq \begin{cases} \mathcal{A}^{i_0}(t) & \text{if } t \in I_0, \\ \mathcal{A}^{i_1}(t) & \text{if } t \in I_1. \end{cases}$$

Proceeding inductively, one obtains the switching meta-controller, $\overline{\mathcal{A}}$, over the entire period. We denote the cost as $\mathcal{L}^w(t_0, x_0^w, \overline{\mathcal{A}})$ or simply $\mathcal{L}^w(\overline{\mathcal{A}})$.

The following theorem is straightforward to prove.

Theorem 7.1 *$\mathcal{L}^w(\overline{\mathcal{A}}) \leq \mathcal{L}^w(\mathcal{A}^i)$ for all $i \in \{1, 2, ..., N\}$.*

PROOF.

$$\begin{aligned}
\mathcal{L}^w(\overline{\mathcal{A}}) &= E_{t_0,x_0^w}\Big\{\underline{L}^w_{t_K}(\overline{X}^w(I_0 \cup I_1 \cup \ldots \cup I_{K-1}), \overline{u}^w(I_0 \cup I_1 \cup \ldots \cup I_{K-1})) \\
&\qquad + \min_{1\leq i_K \leq N} E_{t_K,\overline{X}^w(t_K)}\{\overline{L}^w_{t_K}(X^w_{i_K}(I_K), u^w_{i_K}(I_K))\}\Big\} \\
&= E_{t_0,x_0^w}\Big\{\sum_{k=0}^{K-1} \widehat{L}^w_{t_K}(\overline{X}^w(I_k), \overline{u}^w(I_k)) \\
&\qquad + \min_{1\leq i_K \leq N} E_{t_K,\overline{X}^w(t_K)}\{\widehat{L}^w_{t_K}(X^w_{i_K}(I_K), u^w_{i_K}(I_K))\}\Big\} \\
&\leq E_{t_0,x_0^w}\Big\{\sum_{k=0}^{K-2} \widehat{L}^w_{t_K}(\overline{X}^w(I_k), \overline{u}^w(I_k)) \\
&\quad + \min_{1\leq i_{K-1} \leq N} \min_{1\leq i_K \leq N} E_{t_{K-1},\overline{X}^w(t_{K-1})}\{\widehat{L}^w_{t_{K-1}}(X^w_{i_{K-1}}(I_{K-1}), u^w_{i_{K-1}}(I_{K-1})) \\
&\qquad\qquad + \widehat{L}^w_{t_K}(X^w_{i_K}(I_K), u^w_{i_K}(I_K))\}\Big\} \\
&\leq E_{t_0,x_0^w}\Big\{\sum_{k=0}^{K-2} \widehat{L}^w_{t_K}(\overline{X}^w(I_k), \overline{u}^w(I_k)) \\
&\qquad + \min_{1\leq i_{K-1} \leq N} E_{t_{K-1},\overline{X}^w(t_{K-1})}\{\overline{L}^w_{t_{K-1}}(X^w_{i_{K-1}}(I_{K-1} \cup I_K), u^w_{i_{K-1}}(I_{K-1} \cup I_K))\}\Big\},
\end{aligned}$$

and continuing this process

$$\begin{aligned}
&\leq \min_{1\leq i_0 \leq N} E_{t_0,x_0^w}\Big\{\overline{L}^w_{t_0}(X^w_{i_0}([t_0, \tau^w]), u^w_{i_0}([t_0, \tau^w]))\Big\} \\
&\leq \mathcal{L}^w(\mathcal{A}^i). \square
\end{aligned}$$

In the (more likely) case where the commander or planner does not have an a priori choice for L^w, one can proceed similarly, but with the modification induced by using the combination criteria (such as H_{max}) in place of L^w as described at the end of the previous section. We do not include the details of the trivial modifications. Also, one obvious extension of this would be to allow control algorithm switching at any time; this will not be pursued here, but left for later papers/researchers.

7.3 Game Models

In the above analysis, the models were restricted to deterministic and stochastic dynamics. In the C^2 arena, game models are also certainly appropriate, and so we make a few

comments in this section. The techniques described above may be modified in obvious ways to deal with such models. Although, one could consider both deterministic and stochastic games, as well as continuous time/space and discrete time/space models, let us restrict the discussion here say, a simple continuous time/space deterministic real-world dynamics model. We suppose there is an antagonistic player in the real-world attempting to maximize the criterion the controller or planner is trying to minimize. (Similar models might be used as the control algorithms.) For instance, suppose the dynamics are

$$\begin{aligned} \frac{dX^w}{dt} &= f(t, X^w, u^w, v^w) \\ X^w(t_0) &= x_0^w \end{aligned}$$

where u^w remains our control, but now the dynamics are affected by the controls of the antagonistic player denoted by v^w. Suppose for instance, that one has restrictions in the real-world of $u^w \in \mathcal{U}^{M_u} \doteq \{u^w \in C[t_0, \infty) : |u^w(t)| \leq M_u \quad \forall t \in [t_0, \infty)\}$ and $v^w \in \mathcal{V}^{M_v} \doteq \{v^w \in C[t_0, \infty) : |v^w(t)| \leq M_v \quad \forall t \in [t_0, \infty)\}$. Assume that f is sufficiently well-behaved to guarantee existence and uniqueness for all controls in $\mathcal{U}^{M_u}$ and $\mathcal{V}^{M_v}$ for all $t \in [t_0, \infty)$ (or at least for all $t \in [t_0, \tau^w]$).

A standard cost criterion might take the form

$$\mathcal{L}^w(t_0, x_0^w, \mathcal{A}^i) = \sup_{v^w \in \mathcal{V}^{M_v}} \left[\int_0^{\tau^w} F(s, X_i^w(s), u_i^w(s), v^w(s))\, ds + \psi(X_i^w(\tau^w)) \right].$$

Analyses similar to those of the previous sections yield optimal meta-controllers and switching meta-controllers. We will not pursue this further.

7.4 Example

The purpose of this section is to indicate that the concept of a meta–controller based on choosing among a possible set of controllers is not entirely vacuous. We will use an example to indicate this. Again, since this paper is only intended to introduce the area as a possible field of study, the example will be rather simple. However, we will allow the models to vary quite significantly from the original system, so that nontrivial behaviors will be manifest. In a real world system, it would be expected that the system models used by the controllers would be tied down somewhat by the realities of the true system.

Consider a simple discrete-time stochastic real-world model given as follows. Let $\mathcal{X}^w = \{0, 1, 2, 3\}$. Let the dynamics be

$$X^w(t_{k+1}) = \begin{cases} f(X^w(t_k), u^w(t_k), W^w(t_k)) & \\ \quad \doteq X^w(t_k) + u^w(t_k) + W^w(t_k) & \text{if } f(X^w(t_k), u^w(t_k), W^w(t_k)) \in \mathcal{X}^w, \\ 0 & \text{if } f(X^w(t_k), u^w(t_k), W^w(t_k)) < 0, \\ 3 & \text{if } f(X^w(t_k), u^w(t_k), W^w(t_k)) > 3, \end{cases}$$

where $u^w \in \{0, -1, -2\}$ and W^w is a time-uncorrelated random process with $P(W^w(t_k) = 1) = 1/2$ and $P(W^w(t_k) = 0) = 1/2$. (We set $t_{k+1} - t_k = 1$.) We will not suppose an exit set for the commander, but a fixed exit time of $T = 20$.

Let there be three control algorithms where $\mathcal{X}^1 = \mathcal{X}^2 = \mathcal{X}^3 = \mathcal{X}^w$, and let M^i be the identity for all i. Also let $\mathcal{U}^1 = \mathcal{U}^2 = \mathcal{U}^3 = \mathcal{U}^w$ with N^i the identity as well. Suppose however that the dynamics for the three models are

$$X^1(t_{k+1}) = \begin{cases} f^1(X^1(t_k), u^1(t_k), W^1(t_k)) \\ \quad = X^1(t_k) + 2u^1(t_k) + W^1(t_k) & \text{if } f^1(X^1(t_k), u^1(t_k), W^1(t_k)) \in \mathcal{X}^w, \\ 0 & \text{if } f^1(X^1(t_k), u^1(t_k), W^1(t_k)) < 0, \\ 3 & \text{if } f^1(X^1(t_k), u^1(t_k), W^1(t_k)) > 3, \end{cases}$$

$$X^2(t_{k+1}) = \begin{cases} f^2(X^2(t_k), u^2(t_k), W^2(t_k)) \\ \quad = X^2(t_k) + u^2(t_k) + 2W^2(t_k) & \text{if } f^2(X^2(t_k), u^2(t_k), W^2(t_k)) \in \mathcal{X}^w, \\ 0 & \text{if } f^2(X^2(t_k), u^2(t_k), W^2(t_k)) < 0, \\ 3 & \text{if } f^2(X^2(t_k), u^2(t_k), W^2(t_k)) > 3, \end{cases}$$

$$X^3(t_{k+1}) = \begin{cases} f^3(X^3(t_k), u^3(t_k), W^3(t_k)) \\ \quad = X^3(t_k) + u^3(t_k) + W^3(t_k) & \text{if } f^3(X^3(t_k), u^3(t_k), W^3(t_k)) \in \mathcal{X}^w, \\ 0 & \text{if } f^3(X^3(t_k), u^3(t_k), W^3(t_k)) < 0, \\ 3 & \text{if } f^3(X^2(t_k), u^3(t_k), W^3(t_k)) > 3, \end{cases}$$

with the probabilities for W^1, W^2 and W^3 being identical to those for W^w.

Suppose the commander or planner does not have an a priori cost criterion. Let the criteria proposed by the controllers be

$$L^1 = E\left\{\sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k [(X^1(t_k))^2 + 3|u^1(t_k)|]\right\}$$

$$L^2 = E\left\{\sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k [(X^2(t_k))^2 + |u^2(t_k)|]\right\}$$

$$L^3 = E\left\{\Psi(X^3(\tau^3)) + \sum_{k=0}^{K_3} 8|u^3(t_k)|\right\}$$

where in the last cost criterion, the exit set is $\{0, 3\}$ with $\Psi(0) = 0$ and $\Psi(3) = 16$, τ^3 is the corresponding exit time, and K_3 is the last index prior to exit.

The corresponding dynamic programming equations are as follows. For the first controller, it is

$$V^1(x) = x^2 + \min_u\Big\{3|u| + \frac{1}{2}E[V^1(f^1(x, u, w))]\Big\} \qquad \forall\, x \in \{0, 1, 2, 3\}.$$

Similarly, for the second controller it is

$$V^2(x) = x^2 + \min_u\Big\{|u| + \frac{1}{2}E[V^2(f^2(x, u, w))]\Big\} \qquad \forall\, x \in \{0, 1, 2, 3\}.$$

For the third controller, one automatically has $V^3(0) = 0$ and $V^3(3) = 16$. For the remaining states, the dynamic programming equation is

$$V^3(x) = x^2 + \min_u\{8|u| + E[V^3(f^3(x,u,w))]\} \qquad \forall\, x \in \{1,2\}.$$

Solving the dynamic programming equations for the first control model, one finds that the value function and optimal feedback control are

$$V^1(0) = 5/3, \quad V^1(1) = 5, \quad V^1(2) = 11, \quad V^1(3) = 18$$

and

$$F^1(0) = 0, \quad F^1(1) = 0, \quad F^1(2) = -1, \quad F^1(3) = 0.$$

Proceeding similarly for the second control model, one obtains the value

$$V^2(0) = 3, \quad V^2(1) = 5, \quad V^2(2) = 9, \quad V^2(3) = 49/3.$$

In this case, there are multiple optimal feedback controls; we will choose

$$F^2(0) = 0, \quad F^2(1) = -1, \quad F^2(2) = -2, \quad F^2(3) = -2.$$

Lastly, for the third control model, one finds

$$V^3(0) = 0, \quad V^3(1) = 16, \quad V^3(2) = 16, \quad V^3(3) = 16.$$

Note that the controls are not well-defined for the exit state (where we recall the true exit time may differ from the controllers), and so we will arbitrarily choose controls for the points in the exit set. We obtain the feedback control

$$F^3(0) = -1, \quad F^3(1) = -1, \quad F^3(2) = 0, \quad F^3(3) = 0.$$

Note that the above value functions are not necessarily identical to $\mathcal{L}^i(t_0, x_0^w, \mathcal{A}^i)$ since the dynamics used in propagating the true state may differ from that in the models. Also, note that the dynamics and criteria were time-independent in this problem.

Once one has determined the feedback controllers (and given that M^i and N^i are identities), it is easy to determine the associated costs via dynamic programming equations. We present the results in tabular form.

x_0^w	$\mathcal{L}^1(\mathcal{A}^1)$	$\mathcal{L}^1(\mathcal{A}^2)$	$\mathcal{L}^1(\mathcal{A}^3)$	$\mathcal{L}^2(\mathcal{A}^1)$	$\mathcal{L}^2(\mathcal{A}^2)$	$\mathcal{L}^2(\mathcal{A}^3)$	$\mathcal{L}^3(\mathcal{A}^1)$	$\mathcal{L}^3(\mathcal{A}^2)$	$\mathcal{L}^3(\mathcal{A}^3)$
0	1.667	2	6	0.5	1	2	0	0	0
1	5	6	7.333	1.5	3	3.333	76	16	16
2	11	12	11.333	7.167	7	11.333	84	24	16
3	18	19.5	18	18	13.5	18	16	16	16

Using these results, one is able to compute $d_{1,2}$, $d_{1,3}$ and $d_{2,3}$ as discussed in Section 7.1. Using $d_{1,2}$ yields a preference of $\mathcal{A}^1$ over $\mathcal{A}^2$ if $x_0^w = 0, 1, 2$, and vice-versa for the other state. Using $d_{1,3}$ yields a preference of $\mathcal{A}^1$ over $\mathcal{A}^3$ if $x_0^w = 0$, vice-versa for $x_0^w = 2, 3$, and no preference for $x_0^w = 3$. Using $d_{2,3}$ yields a preference of $\mathcal{A}^2$ over $\mathcal{A}^3$ everywhere. Comparing these three sets of results, one sees that $\mathcal{A}^1$ is the preference if $x_0^w = 0$, and that $\mathcal{A}^2$ is the preference if $x_0^w = 3$. However, when $x_0^w = 1, 2$, one sees that these pairwise preferences do not lead to any single overall preference among the three.

We now proceed to compute the combination criteria $\mathcal{H}_{sum}$ and $\mathcal{H}_{max}$.

x_0^w	$\mathcal{H}_{sum}(\mathcal{A}^1)$	$\mathcal{H}_{sum}(\mathcal{A}^2)$	$\mathcal{H}_{sum}(\mathcal{A}^3)$	$\mathcal{H}_{max}(\mathcal{A}^1)$	$\mathcal{H}_{max}(\mathcal{A}^2)$	$\mathcal{H}_{max}(\mathcal{A}^3)$
0	2.167	3	8	1.667	2	6
1	84	25	26.667	76	16	16
2	102.167	43	38.667	84	24	16
3	52	49	52	18	19.5	18

Using combination criterion $\mathcal{H}_{sum}$ in the controller optimization method of Section 7.1 leads to a choice of $\mathcal{A}^1$ if $x_0^w = 0$, $\mathcal{A}^2$ if $x_0^w = 1$, $\mathcal{A}^3$ if $x_0^w = 2$, and $\mathcal{A}^2$ if $x_0^w = 3$.

We also note that the above choices could be used in a feedback form to yield a switching meta-controller as described in Section 7.2. Noting that the choice at each switching time depends only on the remaining cost to come, one sees that one needs to compute also $\mathcal{H}_{sum}$ over controllers 1 and 2 for $t > \tau^3$; to avoid confusion, let us denote this as $\mathcal{H}_{sum}^{1,2}$ This is given by

x_0^w	$\mathcal{H}_{sum}^{1,2}(\mathcal{A}^1)$	$\mathcal{H}_{sum}^{1,2}(\mathcal{A}^2)$	$\mathcal{H}_{sum}^{1,2}(\mathcal{A}^3)$
0	2.167	3	8
1	6.5	9	10.667
2	18.5	19	22.667
3	36	33	36

The resulting switching meta-control is identical to the optimized control choice above for $t \leq \tau^3$, that is one chooses $\mathcal{A}^1$ if $x^w = 0$, $\mathcal{A}^2$ if $x^w = 1$, $\mathcal{A}^3$ if $x^w = 2$, and $\mathcal{A}^2$ if $x^w = 3$. In other words, one obtains the real world feedback given by $\overline{u}^w(0) = 0$, $\overline{u}^w(1) = -1$, $\overline{u}^w(2) = 0$, and $\overline{u}^w(3) = -2$. For $t > \tau^3$, one obtains $\mathcal{A}^1$ if $x^w = 0$, $\mathcal{A}^1$ if $x^w = 1$, $\mathcal{A}^1$ if $x^w = 2$, and $\mathcal{A}^2$ if $x^w = 3$. In other words, one obtains the real world feedback given by $\overline{u}^w(0) = 0$, $\overline{u}^w(1) = 0$, $\overline{u}^w(2) = -1$, and $\overline{u}^w(3) = -2$ after τ^3. It is not difficult to check that the switching controller satisfies the statement of Theorem 7.1.

One can also obtain the optimized and switching controllers for $\mathcal{H}_{max}$. These differ from those for $\mathcal{H}_{sum}$ only at $x^w = 3$. Specifically, the optimized controller choice and the switching controller (for all t) have $\overline{u}^w(3) = 0$.

References

[1] T. Basar and P. Bernhard, **H_∞ –Optimal Control and Related Minimax Design Problems**, Birkhäuser (1991).

[2] F.L. Baccelli, G. Cohen, G.J. Olsder and J.-P. Quadrat, **Synchronization and Linearity**, John Wiley (1992).

[3] M. Bardi and I. Capuzzo-Dolcetta, "Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations", Birkhauser, Boston, 1997.

[4] D.P. Bertsekas, *Dynamic Programming, Deterministic and Stochastic Models*, Prentice-Hall, Englewood, 1987.

[5] D.P. Bertsekas, D.A. Castañon, M.L. Curry and D. Logan, Adaptive Multi-platform Scheduling in a Risky Environment, Advances in Enterprise Control Symposium Proc., (1999), DARPA–ISO, 121–128.

[6] J.B. Cruz, M.A. Simaan, et al., Modeling and Control of Military Operations Against Adversarial Control, Proc. 39th IEEE CDC, Sydney (2000), 2581–2586.

[7] R. J. Elliott and N. J. Kalton, "The existence of value in differential games", Memoirs of the Amer. Math. Society, **126** (1972).

[8] W.H. Fleming, "Deterministic nonlinear filtering", Annali Scuola Normale Superiore Pisa, Cl. Scienze Fisiche e Matematiche, Ser. IV, **25** (1997), 435–454.

[9] W.H. Fleming and W.M. McEneaney, "A max–plus based algorithm for an HJB equation of nonlinear filtering", SIAM J. Control and Optim., 38 (2000), 683–710.

[10] W.H. Fleming and W.M. McEneaney, "Robust limits of risk sensitive nonlinear filters", Math. Control, Signals and Systems (to appear).

[11] W. H. Fleming and W. M. McEneaney, "Risk sensitive control on an infinite time horizon", SIAM J. Control and Optim., Vol. 33, No. 6 (1995) 1881–1915.

[12] W. H. Fleming and W. M. McEneaney, "Risk–sensitive control with ergodic cost criteria", Proceedings 31st IEEE Conf. on Dec. and Control, (1992).

[13] W. H. Fleming and W. M. McEneaney, "Risk–sensitive optimal control and differential games", (Proceedings of the Stochastic Theory and Adaptive Controls Workshop) Springer Lecture Notes in Control and Information Sciences 184, Springer–Verlag (1992).

[14] W.H. Fleming and H.M. Soner, "Controlled Markov Process and Viscosity Solutions", Springer-Verlag, New York, 1993.

[15] J. Filar and K. Vrieze, **Competitive Markov Decision Proceses**, Springer (1997).

[16] D. Ghose, M. Krichman, J.L. Speyer and J.S. Shamma, Game Theoretic Campaign Modeling and Analysis, Proc. 39th IEEE CDC, Sydney (2000), 2556–2561.

[17] W.D. Hall and M.B. Adams, Closed-loop, Hierarchical Control of Military Air Operations, Advances in Enterprise Control Symposium Proc., (1999), DARPA–ISO, 245–250.

[18] S.A. Heise and H.S. Morse, The DARPA JFACC Program: Modeling and Control of Military Operations, Proc. 39th IEEE CDC, Sydney (2000), 2551–2555.

[19] M. Horton and W.M. McEneaney, "Computation of max–plus eigenvector representations for nonlinear H_∞ value functions", 1999 American Control Conference, 1400–1404.

[20] M.R. James and S. Yuliar, "A nonlinear partially observed differential game with a finite-dimensional information state", Systems and Control Letters, **26**, (1995), 137–145.

[21] M. R. James and J. S. Baras, "Partially observed differential games, infinite dimensional HJI equations, and nonlinear H_∞ control", SIAM J. Control and Optim., **34** (1996), 1342–1364.

[22] J. Jelinek and D. Godbole, Model Predictive Control of Military Operations, Proc. 39th IEEE CDC, Sydney (2000), 2562–2567.

[23] V.P. Maslov, "On a new principle of superposition for optimization problems", Russian Math. Surveys, **42** (1987) 43–54.

[24] W.M. McEneaney, "Error Analysis of a Max–plus Algorithm for a First-Order HJB Equation", Workshop on Max-Plus Algebras, First Symposium on System Structure and Control, Prague, 2001 (to appear).

[25] W.M. McEneaney, "Convergence and Error Analysis for a Max–plus Algorithm", 39th IEEE Conf. on Decision and Control, Sydney (2000), 1194–1199.

[26] W.M. McEneaney, "The max–plus eigenvector algorithm for nonlinear H_∞ control", Proc. American Control Conf. 2000.

[27] W.M. McEneaney and M. Horton, "Max–plus eigenvector representations for nonlinear H_∞ value functions", 37th IEEE Conf. on Decision and Control (1998), 3506–3511.

[28] W.M. McEneaney, "Robust/game–theoretic methods in filtering and estimation", First Symposium on Advances in Enterprise Control, San Diego (1999), 1–9.

[29] W.M. McEneaney, "Robust/H_∞ filtering for nonlinear systems", Systems and Control Letters, **33** (1998) 315–325.

[30] W.M. McEneaney, "A Uniqueness result for the Isaacs equation corresponding to nonlinear H_∞ control", Math. Controls, Signals and Systems, **11** (1998), 303–334.

[31] W.M. McEneaney, "Robust control and differential games on a finite time horizon", Math. of Controls Signals and Systems, **8** (1995), 138–166.

[32] W. M. McEneaney and P. Dupuis, "A risk–sensitive escape criterion and robust limit", Proceedings 33rd IEEE Conf. on Dec. and Control, (1994) 4195–4197.

[33] W.M. McEneaney and K. Ito, Stochastic Games and Inverse Lyapunov Methods in Air Operations, Proc. 39th IEEE CDC, Sydney (2000), 2568–2573.

[34] H. Mukai, et al., Game-Theoretic Linear-Quadratic Method for Air Mission Control, Proc. 39th IEEE CDC, Sydney (2000), 2581–2586.

[35] Q. Zhang, Nonlinear filtering and control of a switching diffusion with small observation noise, *SIAM Journal on Control and Optimization*, Vol. 36, pp. 1738-1768, (1998).

[36] E.Rouy and A.Tourin, A viscosity solutions approach to shape-from-shading, SIAM Numer. Anal. 29 (1992), 867-884.

[37] S. Stubberud et al., Automation of Command and Control Planning Using a Markov Process-Based Technique, Advances in Enterprise Control Symposium Proc., (1999), DARPA–ISO, 137–148.

[38] S. P. Sethi and Q. Zhang, *Hierarchical Decision Making in Stochastic Manufacturing Systems*, Birkhäuser, Boston, 1994.

[39] G. Yin and Q. Zhang, *Continuous-Time Markov Chains and Applications: A Singular Perturbation Approach*, Springer-Verlag, New York, 1998.