



Digital Commons@

Loyola Marymount University
LMU Loyola Law School

Mathematics, Statistics and Data Science
Faculty Works

Mathematics, Statistics and Data Science

2003

Stochastic Game Approach to Air Operations

Ben G. Fitzpatrick

Loyola Marymount University, bfitzpatrick@lmu.edu

William M. McEneaney

University of California - San Diego

Istvan G. Lauko

University of Wisconsin - Milwaukee

Follow this and additional works at: https://digitalcommons.lmu.edu/math_fac



Part of the [Aerospace Engineering Commons](#)

Digital Commons @ LMU & LLS Citation

Fitzpatrick, Ben G.; McEneaney, William M.; and Lauko, Istvan G., "Stochastic Game Approach to Air Operations" (2003). *Mathematics, Statistics and Data Science Faculty Works*. 77.

https://digitalcommons.lmu.edu/math_fac/77

This Article - post-print is brought to you for free and open access by the Mathematics, Statistics and Data Science at Digital Commons @ Loyola Marymount University and Loyola Law School. It has been accepted for inclusion in Mathematics, Statistics and Data Science Faculty Works by an authorized administrator of Digital Commons@Loyola Marymount University and Loyola Law School. For more information, please contact digitalcommons@lmu.edu.

Stochastic Game Approach to Air Operations ^{*}

William M. McEneaney [†] Ben G. Fitzpatrick [‡] Istvan G. Lauko [§]

January 10, 2003

Abstract

A Command and Control (C^2) problem for Military Air Operations is addressed. Specifically, we consider C^2 problems for air vehicles against ground based targets and defensive systems. The problem is viewed as a stochastic game. In this paper, we restrict our attention to the C^2 level where the problem may consist of a few UCAVs or aircraft (or possibly teams of vehicles); less than say, a half-dozen enemy SAMs; a few enemy assets (viewed as targets from our standpoint); and some enemy decoys (assumed to mimic SAM radar signatures). At this low level, some targets are mapped out and possible SAM sites that are unavoidably part of the situation are known. One may then employ a discrete stochastic game problem formulation to determine which of these SAMs should optimally be engaged (if any), and by what series of air vehicle operations. Since this is a game model, the optimal opponent strategy is also determined. We provide analysis, numerical implementation, and simulation for full state feedback and measurement feedback control within this C^2 context.

1 Introduction

In this paper we address a Command and Control (C^2) problem for Military Air Operations, involving air vehicles operating against ground based targets and defenses. The

^{*}Research partially supported by DARPA grant F30602-99-2-0548.

[†]Depts. of Mathematics and Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093-0112, USA, wmceneaney@ucsd.edu

[‡]Department of Mathematics, Loyola Marymount University, Los Angeles, CA 90045, and Tempest Technologies LLC, 8929 Sepulveda Blvd, Suite 208, Los Angeles, CA 90045, bfitzpatrick@lmu.edu

[§]Dept. of Mathematics, University of Wisconsin, Milwaukee

motivation for this study arises from the need to plan operations with unmanned combat air vehicles (UCAVs), and to be able to re-plan in real-time. We pose the basic problem as a stochastic game with partial observation of the states, and we employ dynamic programming and robust control approaches to developing feedback controls. Simulations are employed to examine the optimal controls and observe the system behavior and performance. The problem we consider here is one of relatively short-range planning and execution for a small number of air vehicles against a small number of targets and defensive entities. The goal is the development of planning tools that will allow for analysis of information and replanning of targeting in-flight.

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 challenging factors. In general, the problem size often leads to a number of simplifying assumptions and sub-optimal techniques which must be employed in order to make the problem computationally tractable.

We remark here that the proposed game theoretic techniques are not to be employed in a stand-alone manner. Due to the large size of the problem, various methods are often used to decompose the problem into manageable pieces. Hierarchical problem decompositions require one 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 targeting problem and vehicle guidance problem is considered. At somewhat higher levels in the hierarchy one needs to determine vehicle routes, engagement assignment, scheduling of ordinance preparation, maintenance and repairs. Higher yet in the hierarchy are the resource allocation and strategic planning problems. The level at which we operate in the current effort involves planning in which the problem may consist of a few UCAVs or aircraft (or possibly teams of vehicles), less than say, a half-dozen, enemy surface-to-air missile air defense units (SAMs), a few enemy assets (viewed as targets from our standpoint), and some enemy decoys (assumed to mimic SAM radar signatures). The tools developed here are being integrated with higher-level resource allocation to construct teams and assign areas of operation as well as lower-level routing, deconfliction, and weaponing planners (although the larger integration problem [1] is not addressed here).

At the level under consideration here, some targets are mapped out and possible SAM sites (and their specifications) that are unavoidably part of the situation are known. One may then employ a discrete stochastic game problem formulation to determine which of these SAMs should “optimally” be engaged, and by what series of air vehicle operations. Also, since this is a game model, the optimal opponent strategy is also determined. In formulating the game, one must assign values to the assets involved and decide upon the type of criterion to be optimized. The term “optimal” strategy will be freely used to indicate the strategy which minimizes the worst case (maximum over enemy controls) cost. This approach will be explicated in the definition of game value to follow in Section 2.

In examining the controllers from this approach, we also consider parametric uncertainty, which is a form of imperfect knowledge of the system dynamics. Game simulations

allow the evaluation of various approaches in terms of expected cost and the variance of the cost. These simulations led to the discovery that there are only very few viable strategies – the optimal choice being parameter dependent. One may plot various measures as functions of selected parameters to determine when the situation is near a point where the optimal strategies may jump suddenly.

Another important aspect of the current work is in the treatment of partial observations for the players. 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_∞ 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.

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, researchers working on C^2 have 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 combined with use of the separation principle).

The paper is organized as follows. In Section 2, we consider our underlying “small” stochastic game problem involving only a few air vehicles, a handful of SAMs and a target. We remind the reader that, although only relative small problems will be considered in the examples there, these tools would form one layer in a hierarchical solution to the larger scale air operations problem for UCAVs. That is, the solution of the problem at this level underlies the solution of larger problems at higher levels. The embedding of this control layer within a larger system is now being performed in a joint effort with our industrial partners [1]. 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. The analysis and simulations discussed in these subsections assume perfect knowledge of the system. The consequences of partial, imperfect and corrupted information are studied in Section 3. In that section, both the estimation problem, and the problem of control under imperfect information (in the presence of an adversary) are discussed. The discovery of the limited selection of optimal strategies (Sec. 2) allows one to tractably consider somewhat larger problems at this hierarchical level, and this is employed in Section 3.

2 Solution of Stochastic Games for C^2

This section deals with the C^2 level in which the problem has been reduced to a small stochastic game involving only a few entities. In the simulation results to appear later in this section, there are only two air vehicles, three SAMs and an enemy target; in the computations for the imperfect information case (Section 3), we have increased this to include at least six SAMs and two targets as well as decoy radar emitters. Even that problem size could easily be doubled with today's available computational power. This study was conducted only as a demonstration of approach, and so no effort was made to explore the maximum tractable problem size. Note also that as indicated above, we used hierarchical techniques to deal with the full scale problem [1], [25].

The objects which will be of interest in this section are air vehicles (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 designations will be assumed throughout the paper.

We will reduce the state of the i^{th} (Blue) air vehicle 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" or level of combat capabilities of the air vehicle, 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 air vehicle in detail; we will not include velocity, altitude, mass and so forth as part of the state. The problem is decomposed into separate sub-problems further below in the hierarchy, where more detailed models for each vehicle may be applied [1], [25], [23]. Here, X_i^A will represent a position taking values in a discrete set of position alternatives; the set of possible positions will be defined below. We will suppose that the health status takes 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 air vehicle has been destroyed.

We will assume similar state models for the SAMs. The i^{th} SAM state could be represented by the pair $Y_i^R(t) = (D_i^R(t), X_i^R(t))$ where D_i^R would represent the health status of the SAM, and X_i^R would represent its position. Note that there exist both mobile SAMs, i.e ones with time dependent positions, and fixed-site SAMs. However, since the timescale of the air vehicle movement is much faster than that of the mobile SAM movement (and the typical duration of problems at this hierarchical level is relatively short), we will assume that each $X_i^R(t)$ is constant for the duration of the problem. Consequently, we will simply take $Y_i^R(t) = D_i^R(t)$ for the remainder of the paper. 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, and again, we will make the simplification that $Y_i^T(t) = D_i^T(t)$ without further comment. Let the number of Blue air vehicles 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 air vehicle state process has the value \vec{y}^A at time t . Finally, let \mathcal{Y} be the state-space, i.e. the set of all possible triplets $(\vec{y}^A, \vec{y}^R, \vec{y}^T)$.

The game objectives, assigned at a higher hierarchical level, may vary. For Blue, the objective may sometimes be to destroy a strategic target while minimizing damage to the air vehicle; 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. We remark here that such an approach is conservative from Blue's point of view. That is, if Blue has chosen a criterion to minimize, and Red takes a strategy that maximizes some other criterion, then Red's strategy is not guaranteed to be worst-case in terms of Blue's definition of optimality (as embodied in the cost criterion).

In the problem at hand, the time horizon over which the players' objectives should be met is not fixed. We choose here 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 air vehicles have been destroyed or 2) the Red strategic target(s) has(have) been destroyed and the surviving Blue air vehicles 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

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

where μ_A, μ_R, μ_T are the parameters. Thus, the payoff for a given initial state, $(\vec{Y}^A(0), \vec{Y}^R(0), \vec{Y}^T(0))$, and given control trajectories for both players, $u(\cdot), w(\cdot)$, (where the control spaces will be defined below) is

$$J(\vec{Y}^A(0), \vec{Y}^R(0), \vec{Y}^T(0); u(\cdot), w(\cdot)) \doteq \mathbb{E} \left\{ \Psi(\vec{Y}^A(\tau), \vec{Y}^R(\tau), \vec{Y}^T(\tau)) \right\}. \quad (2)$$

The presence of the expectation in (2) is due to the fact that the dynamics of the health status of the objects will involve random outcomes of engagements and maintenance.

We note here that running costs or finite time horizon approaches can be considered. In general, we would expect these cost functionals to produce different optimal controls than those constructed with exit time cost. Moreover, we would expect planners to develop cost criteria that reflect battle objectives derived from strategic considerations. The purpose of the effort here is not to explore what cost criteria might be most appropriate for a particular strategic plan, but rather to understand and demonstrate feasibility of the general game theoretic approach.

The next two subsections describe the mathematics behind the algorithm to compute optimal engagement plans/re-plans.

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 [23], [25]. As discussed above, there may be one or more SAM sites intervening along this path. At our level, the positional dynamics will be specified only in a general way. Let the SAMs be indexed as $\{1, 2, 3, \dots, N_R\}$. Let the air vehicle position take values in the set $\mathcal{L} \doteq \{B, 1, 2, 3, \dots, N_R, N_R + 1\}$ where B indicates the (Blue) base and $N_R + 1$ indicates the (Red) strategic target (we will assume in this section that there is only one target - $N_T = 1$). We suppose a discrete-time event-based model where each time step occurs only when either an air vehicle engages a SAM, an air vehicle engages the target, or an air vehicle returns to base. More than one such activity can occur at each step. The air vehicle control for each air vehicle, $U_i^A(t)$, 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”, i.e the air vehicle is neither to return to base nor to engage the other player in the current discrete time step; in effect, the air vehicle “hides” for that step. 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 for simplicity of the state-space, $X_i^A(t + 1) = X_i^A(t)$ 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 air vehicles 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 $t = kn_c$

if there exists i such that $X_i^A(t) = B$ and $D_i^A(t) = 1$, then there must be an $i_0 \leq N_A$ and $\hat{t} \in [kn_c, (k + 1)n_c - 1)$ with $D_{i_0}^A(t) \neq 4$ such that $U_{i_0}^A(\hat{t}) \notin \{B, 0\}$. (CC)

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

Again, it will be assumed that the SAMs cannot move during the duration of the game. Thus, 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. When the radar is on, the probability of the SAM inflicting damage on the air vehicle rises - as does the probability that the air vehicle can inflict damage on the SAM. This is due to the fact that not only does the SAM radar allow Red to track the Blue air vehicles, but it also allows Blue to detect the SAM site and launch weapons that track the origin of the radar signal. We are modeling this behavior via the health transition probabilities.

The health status of each of the objects will transition according to a discrete-time Markov chain model. The transition probabilities will be state/control dependent. To simplify matters, assume that multiple air vehicles can attack a single SAM, but that the

air vehicle need only engage one SAM at a time (although it may be attacked enroute to and from the target attack moment, as clarified below). At each time-step, where a SAM is under attack, we let the transition probability of the SAM health status 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 air vehicle, a SAM with radar off is being attacked by multiple air vehicles simultaneously, a SAM with radar on is being attacked by a single air vehicle, and a SAM with radar on is being attacked by multiple air vehicles simultaneously, respectively. Of course, there are many more possibilities, but we consider only these for simplicity. For example, suppose $N_R = 1$, $N_A = 1$, $\mathcal{L} = \{B, 1, 2\}$, and that $U_1^A(t) = X_1^A(t+1) = 1$ (so that the single SAM is under attack by the single air vehicle. Suppose the SAM health before attack is $D_1^R(t) = 1$. Then the probabilities of SAM health after attack are

$$P(D_1^R(t+1) = j) = P_{1,j}^{R01} \quad \forall j \in \{1, 2, 3\}.$$

Further, if the SAM health is unknown with probability distribution $P(D_1^R(t) = i) = p_i^R$ for all $i \in \{1, 2, 3\}$, then after this attack, the distribution of SAM health states is given by

$$P(D_1^R(t+1) = j) = \sum_{i=1}^3 P_{i,j}^{R01} p_i^R \quad \forall j \in \{1, 2, 3\}.$$

In vector form,

$$\vec{p}^R(t+1) = (P^{R01})^T \vec{p}^R(t)$$

where $\vec{p}^R(t)$ is the vector of probabilities p_i^R at time t .

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 transition probability given by P^{R00} . The state $d_i^R = 3$ will be an absorbing state for all the transition 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 air vehicle 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). (PD)

We also allow the air vehicle to undergo maintenance while at the base ($U_i^A(t) = B$), and let the transition probabilities be P^{A00} . For the air vehicle, 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 unavoidable on the best route from SAM 1 to SAM 3 (recall a lower level in the hierarchy does path-planning), then air vehicle i could suffer damage while flying near each of the SAMs 2 and 4 – if they are on. We let the transition probability for air vehicle health due to flying within a SAM’s kill zone while that SAM 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 air vehicle 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} (i.e. a downed air vehicle cannot be repaired).

Note that in this last example, the information necessary to determine the overall transition probability for an air vehicle is only the geometric relation of the SAM umbrellas (i.e. kill zones) and the SAM and air vehicle capabilities. Thus, we can model the system state in terms of a directed graph, where each SAM (or target) corresponds to a node on the graph. The directed link from one node to another exists if and only if the second object is under the umbrella of the first. In developing a full set of controls for such a system, we construct a database for the possible configurations the defense could employ, in terms of these directed graphs. For each given directed graph, we can then compute the optimal game value and controls as described herein.

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). This value gives an advantage to Red, and leads to a cautious controller for Blue. One could also consider the lower value, but that is not done here. Let the value function for this game be denoted by $V(\vec{y}^A, \vec{y}^R, \vec{y}^T)$. It is defined as

$$V(\vec{y}^A, \vec{y}^R, \vec{y}^T) = \sup_{\lambda \in \Lambda} \inf_{u \in \mathcal{U}} J(\vec{y}^A, \vec{y}^R, \vec{y}^T; u, \lambda[u]) \quad (3)$$

where J is the cost criterion defined in Equation (2) above, \mathcal{U} denotes the set of Blue controls, and Λ denotes the set of Red’s nonanticipative strategies (nonanticipative mappings from Blue controls to Red controls – see for instance [6]).

Since it is quite standard, we do not include a proof of the DPE (dynamic programming equation) which is given as follows. (Note here that \mathcal{U}^{N_A} represents the set product of \mathcal{U} N_A -times, and similarly for $\{0, 1\}^{N_R}$.)

Theorem 2.1 *The value function satisfies*

$$V(\vec{y}^A, \vec{y}^R, \vec{y}^T) = \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \left[E \left\{ V(\vec{Y}^A(1), \vec{Y}^R(1), \vec{Y}^T(1)) \right. \right. \\ \left. \left. \mid \vec{Y}^A(0) = \vec{y}^A, \vec{Y}^R(0) = \vec{y}^R, \vec{Y}^T(0) = \vec{y}^T, \right\} \right]$$

$$\doteq \min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \mathcal{G}^{\vec{u}^A, \vec{g}^R} [V](\vec{y}^A, \vec{y}^R, \vec{y}^T)$$

where this last line defines the operator $\mathcal{G}^{\vec{u}^A, \vec{g}^R}$, and the entire right hand side of this last line is the application of the dynamic programming operator, $\min_{\vec{u}^A \in \mathcal{U}^{N_A}} \max_{\vec{g}^R \in \{0,1\}^{N_R}} \mathcal{G}^{\vec{u}^A, \vec{g}^R}$.

We will now indicate how the value function can be obtained through repeated application 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 exist $n < \infty$ and $\delta > 0$ such that for any sequence of controls for Blue and Red*

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

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

PROOF. Given t , let $t_1 = \min\{s > t : s = kn_c + 1 \text{ for some non-negative integer } k\}$. Then, by assumption (CC), there exists $i_1 \leq N_A$ such that $X_{i_1}^A(t_1) \neq B$, and consequently by assumption (PD), there exists $\delta_1 > 0$ (dependent on the choice of transition matrices) such that $P(D_{i_1}^A(t_1) = 4) \geq \delta_1$. Let $\Omega_1 \subseteq \Omega$ (the sample space) be given by $\Omega_1 = \{\omega \in \Omega : D_{i_1}^A(k(n_c+1)) = 4\}$. 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 air vehicles are down), let $t_2 = \min\{s > t_1 : s = kn_c + 1 \text{ for some non-negative integer } k\}$. Then again by condition (CC), there exists $i_2 \leq N_A$ such that $X_{i_2}^A(t_2) \neq B$, and consequently, $P(\{\omega \in \Omega_1 : D_{i_2}^A(t_2) = 4\}) \geq \delta_1$. Since state 4 is absorbing, This implies that $P(D_{i_1}^A(t) = 4, D_{i_2}^A(t) = 4) \geq \delta_1^2$ for all $t \geq t_2$. Proceeding inductively, one finds that by, at most, time $t + n_c N_A$, the state is in \mathcal{E} with probability no less than $\delta \doteq \delta_1^{N_A}$. \square

Define the backward dynamic programming (DP) algorithm 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

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

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 repeated n times is a contraction.*

PROOF. Once one has Lemma 2.2, the proof of this lemma is similar to standard proofs, but in this case it is for a game with an exit criterion. (See, for instance, [3], [7] 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 $k < 0$)

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

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

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

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

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

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

$$\begin{aligned} W_1(k, \vec{y}) - W_2(k, \vec{y}) &\leq \sum_{\vec{z} \notin \mathcal{E}} \left\{ [W_1(k+n, \vec{z}) \right. \\ &\quad \left. - W_2(k+n, \vec{z})] \cdot P_{\vec{y}, \vec{z}}^{(n)}(\{\vec{u}_m^A\}_{m=-n}^{-1}, \{\vec{g}_m^R\}_{m=-n}^{-1}) \right\} + 2\varepsilon \end{aligned}$$

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

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

Noting that this holds for all $\varepsilon > 0$ and all $\vec{y} \in \mathcal{Y}$, one has

$$\max_{\vec{y}} \{ |W_1(k, \vec{y}) - W_2(k, \vec{y})| \} \leq \max_{\vec{z}} \{ |W_1(k+n, \vec{z}) - W_2(k+n, \vec{z})| \} (1 - \delta).$$

or,

$$\|W_1(k, \cdot) - W_2(k, \cdot)\|_\infty \leq (1 - \delta)\|W_1(k + n, \cdot) - W_2(k + n, \cdot)\|_\infty$$

where $\|\cdot\|_\infty$ indicates the l_∞ norm. \square

Once one has proven that the backward DP is a contraction, the proof of convergence is 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 $P_{\vec{y}, \vec{z}}^{(1)}(\vec{u}^A, \vec{g}^R)$ 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 an approximation we employed 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 $A_{j,k}^i = 1$ if $j = k = i$ and $A_{j,k}^i = 0$ otherwise. Then, given \vec{y} , define the approximation operator for approximation around \vec{y} by

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

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, 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 the following property of the approximation operator.

Lemma 2.5 *The approximation operator $\mathcal{H}_{\vec{y}}[V(\cdot)]$ is a non-expansive map (with respect to the l_∞ norm) for any $\vec{y} \in \mathcal{Y}$.*

PROOF. Let $\vec{y} \in \mathcal{Y}$. Let $V_1, V_2 : \mathcal{Y} \rightarrow \mathbf{R}$. Then

$$\begin{aligned} \|\mathcal{H}_{\vec{y}}[V_1] - \mathcal{H}_{\vec{y}}[V_2]\|_\infty &= \max_{\vec{z} \in \mathcal{Y}} |\mathcal{H}_{\vec{y}}[V_1](\vec{z}) - \mathcal{H}_{\vec{y}}[V_2](\vec{z})| \\ &= \max_{\vec{z} \neq \vec{y}} \left| \left[\frac{1}{\sum_{i=1}^3 |z_i - y_i|} \right] \cdot \sum_{i=1}^3 [|z_i - y_i| V_1(A^i(\vec{z} - \vec{y}) + \vec{y}) - V_2(A^i(\vec{z} - \vec{y}) + \vec{y})] \right|. \end{aligned}$$

Letting $\vec{z}^1 = (z_1, y_2, y_3)^T$, $\vec{z}^2 = (y_1, z_2, y_3)^T$ and $\vec{z}^3 = (y_1, y_2, z_3)^T$, this can be rewritten as

$$\|\mathcal{H}_{\vec{y}}[V_1] - \mathcal{H}_{\vec{y}}[V_2]\|_\infty = \max_{\vec{z} \neq \vec{y}} \left| \frac{1}{\sum_{i=1}^3 |z_i - y_i|} \sum_{j=1}^3 \{|z_j - y_j| [V_1(\vec{z}^j) - V_2(\vec{z}^j)]\} \right|,$$

and letting $\vec{z}^{\bar{j}} \in \operatorname{argmax} |V_1(\vec{z}^j) - V_2(\vec{z}^j)|$, one has

$$\begin{aligned} &\leq \max_{\vec{z} \neq \vec{y}} \frac{1}{\sum_{i=1}^3 |z_i - y_i|} \left(\sum_{j=1}^3 |z_j - y_j| |V_1(\vec{z}^{\bar{j}}) - V_2(\vec{z}^{\bar{j}})| \right) \\ &= \max_{\vec{z} \neq \vec{y}} |V_1(\vec{z}^{\bar{i}}) - V_2(\vec{z}^{\bar{i}})| \\ &\leq \max_{\vec{z} \in \mathcal{Y}} |V_1(\vec{z}^i) - V_2(\vec{z}^i)| = \|V_1 - V_2\|_\infty. \quad \square \end{aligned}$$

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 the approximation operator and the contraction property of the backward DP, one easily obtains the following result.

Theorem 2.6 *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_2, 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 also actually eliminated that position state for the air vehicle.) Then the approximate backward DP takes the form

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

where

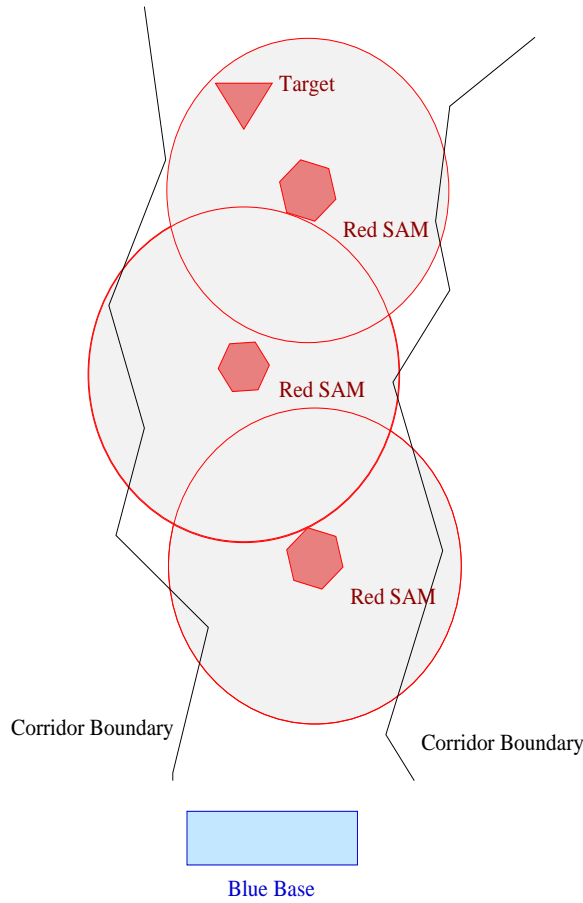
$$Q_{|z_1 - y_1|}^1 \doteq \frac{\sum_{z_2=1}^3 |z_2 - y_2| P_{y_2, z_2}^2 + \sum_{z_3=1}^3 |z_3 - y_3| P_{y_3, z_3}^3}{\sum_{i=1}^3 |z_i - y_i|}$$

with analogous definitions for $Q_{|z_2 - y_2|}^2$ and $Q_{|z_3 - y_3|}^3$. Note that these Q^i may be pre-computed. Thus the approximate DP (4) has only linear growth in the computations which must be performed at each step (*per point in the state space*).

2.3 Test-bed, Monte Carlo Simulation and Landscape Plots

The games controller was tested via Monte Carlo simulation. For the first set of tests, a simple geometry was considered. This is depicted in Figure 1. A flight path 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 one 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.)

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. (A rollback policy is, roughly speaking, one where the SAMs enroute to the target are removed one-by-one starting with the easiest to reach and proceeding towards the desired final target. In figure 1, this obviously entails attacking the lowest SAM in the figure first, and proceeding upwards to the target taking on the second and then third SAMs before

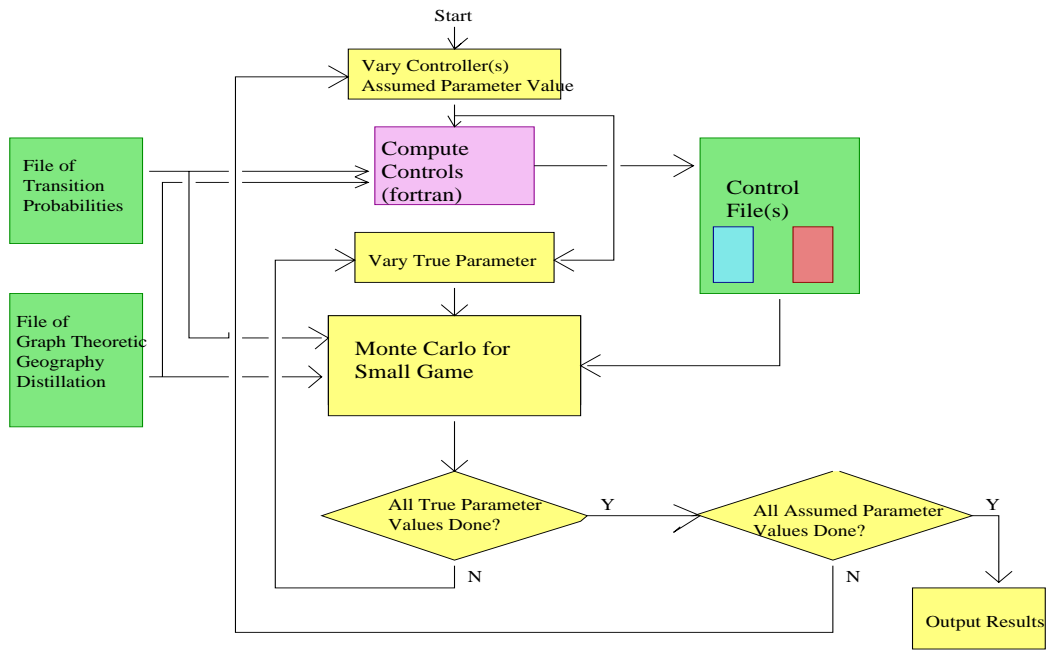


Geography 1 Distillation

Figure 1:

attacking the target.) When considering the entire parameter space (including all the transition probabilities and costs), the optimal Blue control was almost always fly-over or rollback. Only for relatively tiny regions of parameter space was an intermediate policy ever optimal.

A Monte Carlo simulator was used as a tool to look at dependency on certain parameters, and the effect of mis-modeling 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 game simulation series was run. Statistics of game outcomes 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 Figure 2.



Sensitivity Tool for Small Games

MONTE CARLO FOR SMALL GAMES

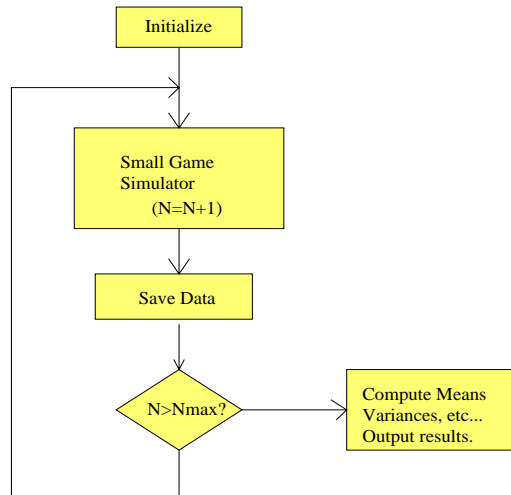


Figure 2:

It was found that the choice of fly-over or rollback was sensitive to the probability of damage to a Blue air vehicle when flying through the SAM umbrella without engaging. Let this probability be denoted by α . (More specifically, the transition probabilities for the air vehicle are 4×4 matrices, and we let the $(1, 4), (2, 4), (3, 4)$ entries be α , the $(1, 1), (2, 2), (3, 3)$ entries be $1 - \alpha$, the $(4, 4)$ be 1, and the rest be zero.) Consequently, Monte Carlo runs were done where the true value of α and the value the controllers believe to be α were varied.

The Monte Carlo data to follow was made with payoff function (1) with parameter values

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

and state transition matrices

$$\begin{aligned}
 P^{A0} &= \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 0.40 & 0.60 & 0 & 0 \\ 0.30 & 0 & 0.70 & 0 \\ 0 & 0 & 0 & 1.00 \end{bmatrix}, & P^{A011} &= \begin{bmatrix} 0.99 & 0 & 0 & 0.01 \\ 0 & 0.95 & 0.025 & 0.025 \\ 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 P^{A021} &= \begin{bmatrix} 1.00 & 0 & 0 & 0.00 \\ 0 & 0.98 & 0.01 & 0.01 \\ 0 & 0 & 0.96 & 0.04 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & P^{A111} &= \begin{bmatrix} 0.82 & 0.01 & 0.02 & 0.15 \\ 0 & 0.82 & 0.03 & 0.15 \\ 0 & 0 & 0.75 & 0.25 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 P^{A121} &= \begin{bmatrix} 0.92 & 0.01 & 0.02 & 0.05; \\ 0 & 0.92 & 0.03 & 0.05; \\ 0 & 0 & 0.87 & 0.13; \\ 0 & 0 & 0 & 1 \end{bmatrix}, & P^{foon} &= \begin{bmatrix} 1 - \alpha & 0 & 0 & \alpha \\ 0 & 1 - \alpha & 0 & \alpha \\ 0 & 0 & 1 - \alpha & \alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 P^{R0} &= \begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0.7 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & P^{R011} &= \begin{bmatrix} 0.7 & 0 & 0.3 \\ 0 & 0.6 & 0.4 \\ 0 & 0 & 1 \end{bmatrix}, & P^{R021} &= \begin{bmatrix} 0.4 & 0 & 0.6 \\ 0 & 0.35 & 0.65 \\ 0 & 0 & 1 \end{bmatrix} \\
 P^{R111} &= \begin{bmatrix} 0.7 & 0 & 0.3 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}, & P^{R121} &= \begin{bmatrix} 0.3 & 0 & 0.7 \\ 0 & 0.25 & 0.75 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

The functionality of the state probability transition matrices for Blue and Red are as follows:

P^{A0} - Blue; Air vehicle recovery at base;

P^{A011} - Blue; Single air vehicle attacks, attacked SAM's radar is off;

P^{A021} - Blue; Multiple air vehicles attack, attacked SAM's radar is off;

P^{A111} - Blue; Single air vehicle attacks, attacked SAM's radar is on;

P^{A121} - Blue; Multiple air vehicles attack, attacked SAM's radar is on;

P^{foon} - Blue; Air vehicle state transition for fly-overs, flown-over SAM's radar is on;

P^{R0} - Red; SAM's recovery while not engaged;

P^{R011} - Red; Single air vehicle attacks, attacked SAM's radar is off;

P^{R021} - Red; Multiple air vehicles attack, attacked SAM's radar is off;

P^{R111} - Red; Single air vehicle attacks, attacked SAM's radar is on;

P^{R121} - Red; Multiple air vehicles attack, attacked SAM's radar is on;

The transition probabilities for a target under attack are the same as those for a SAM with radar off.

For the purposes of reader understanding, the output value has been multiplied by -1 and had a constant added so that the plotted values are averages of terminal payoffs of the form

$$\hat{V} = \mu^A \sum_i^{N_A} (3 - D_i^A) - \mu^R \sum_i^{N_R} (D_i^R - 2) - \mu^T (D^T - 2).$$

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 air vehicles and SAMs. Note that in this case, larger numbers are better for Blue and vice-versa.

Most runs are made with 2000 sample points (a few were made with only 1000 sample points). Each set of figures takes from 1-8 hours to produce on a typical workstation. The controllers are computed 11 times (for each of 11 α values), and typically 2000 simulation runs are made for each of the 121 points on the graph. Note that on the plots, the controller-assumed value of α is denoted as "Control P_k ". The true value of α is denoted as "Actual P_k ". The vertical axis varies from figure to figure as listed below.

The first set of data is for the case of two air vehicles and one SAM site.

- Figure 3 is the sample mean value.
- Figure 4 is the sample standard deviation.
- Figure 5 is the sample mean number of surviving air vehicles
- Figure 6 is the sample mean number of surviving SAMs
- Figure 7 is the sample mean number of surviving targets

(Recall that the game ends when all the Blue air vehicles are down, or the Red target is destroyed and the surviving Blue air vehicles have returned to base.)

Pic.#1 MC experiment: flyover-risk sensitivity -- game value

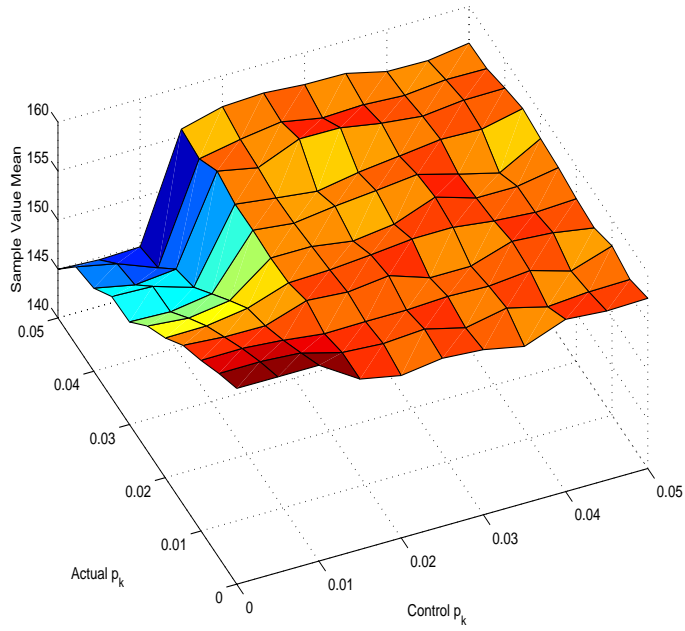


Figure 3:

Pic.#2 MC experiment: flyover-risk sensitivity -- game value (s.d.)

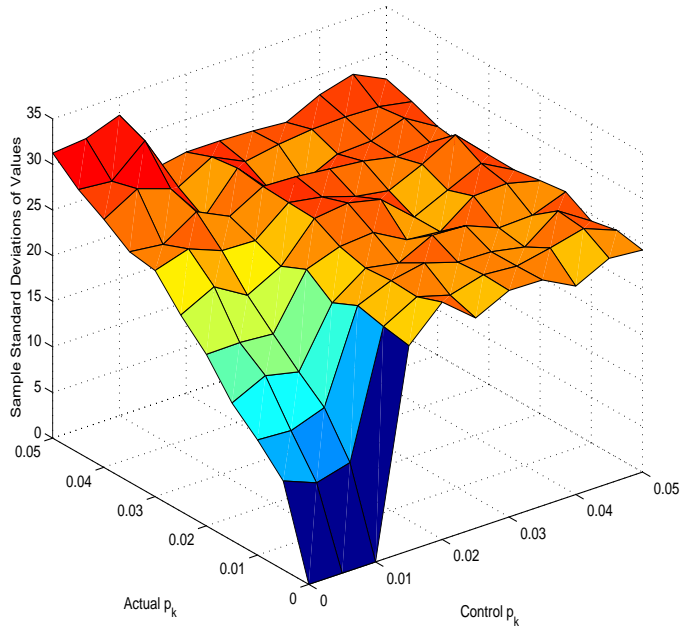


Figure 4:

Pic.#3 MC experiment: flyover-risk sensitivity -- avg. surviving ACs

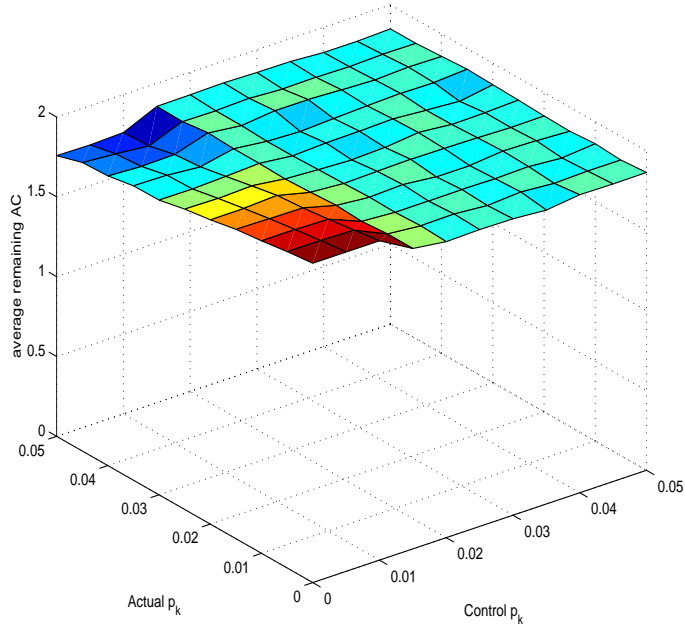


Figure 5:

Pic.#4 MC experiment: flyover-risk sensitivity -- avg. surviving SAMs

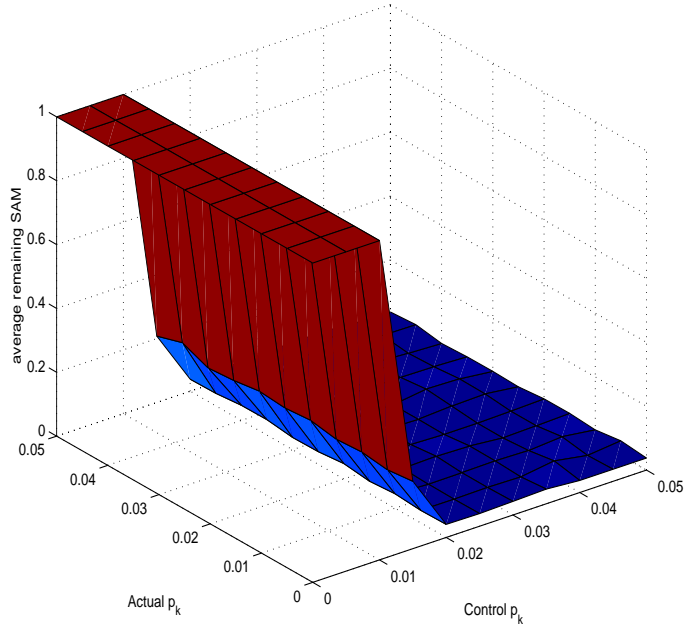


Figure 6:

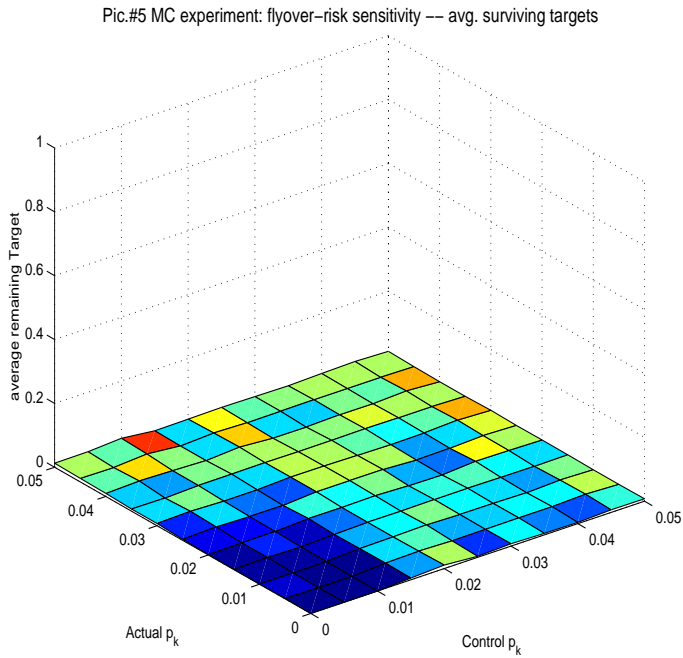


Figure 7:

Note the following:

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.

Pic.#1 MC experiment: flyover-risk sensitivity — game value

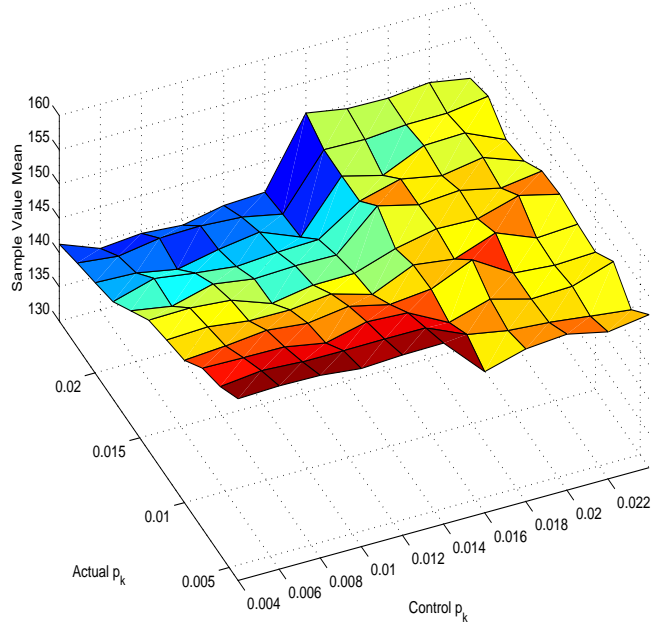


Figure 8:

The second set of data is for the case of two air vehicles and three SAM sites. We use the parameters listed earlier.

- Figure 8 is the sample mean value.
- Figure 9 is the sample standard deviation.
- Figure 10 is the sample mean number of surviving air vehicles
- Figure 11 is the sample mean number of surviving SAMs
- Figure 12 is the sample mean number of surviving targets

Pic.#2 MC experiment: flyover-risk sensitivity -- game value s.d.

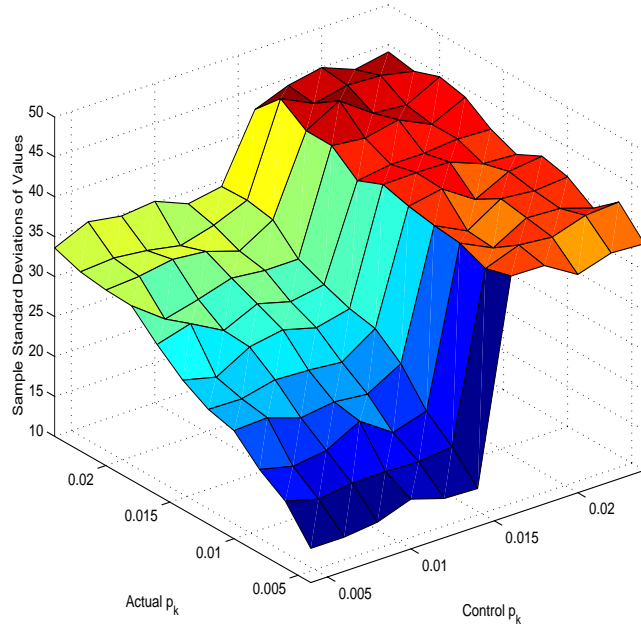


Figure 9:

Pic.#3 MC experiment: flyover-risk sensitivity -- avg. surviving A/C

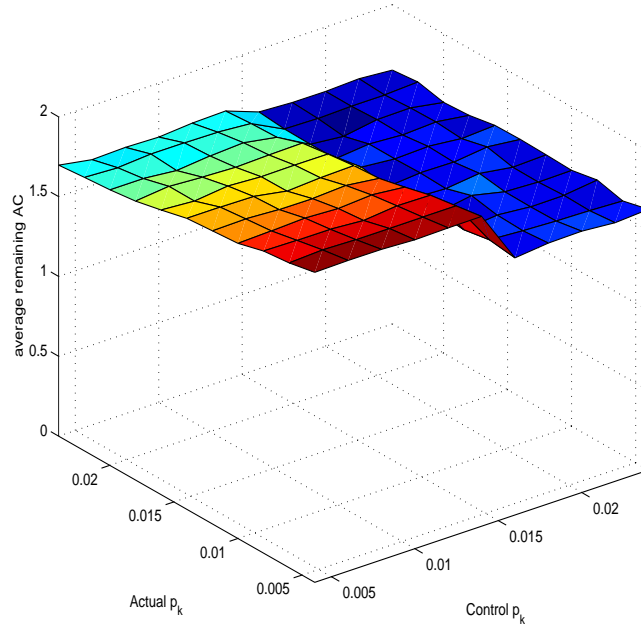


Figure 10:

Pic.#4 MC experiment: flyover-risk sensitivity -- avg. surviving SAMs

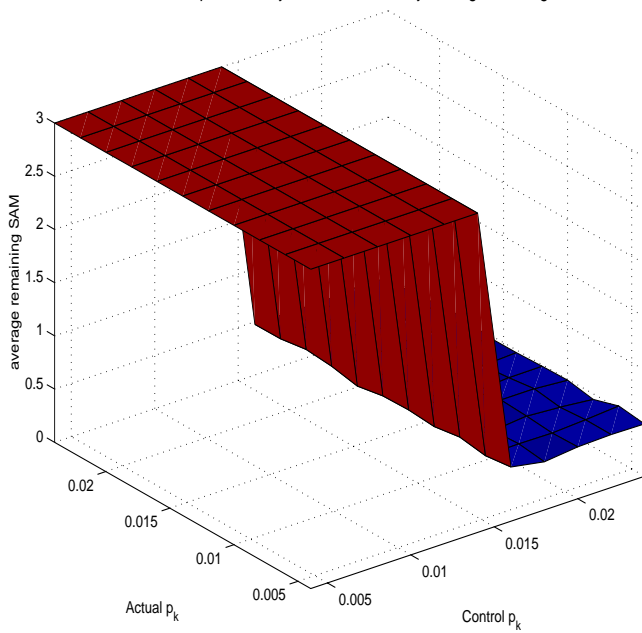


Figure 11:

Pic.#5 MC experiment: flyover-risk sensitivity -- avg. surviving targets

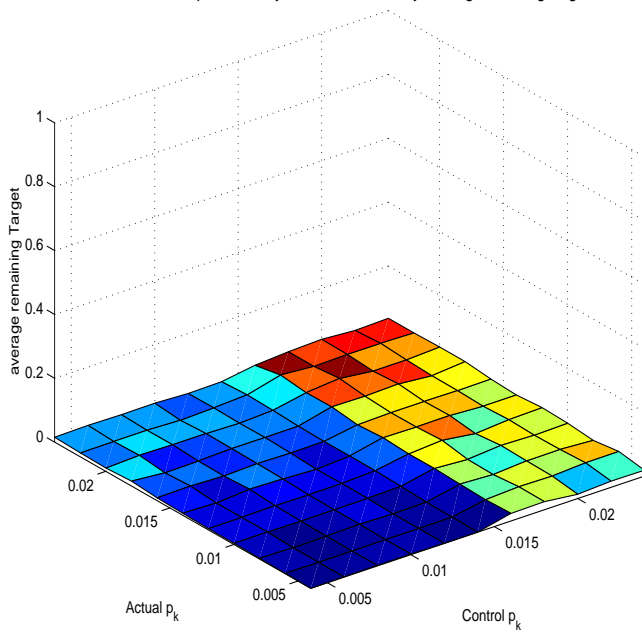


Figure 12:

Note the following:

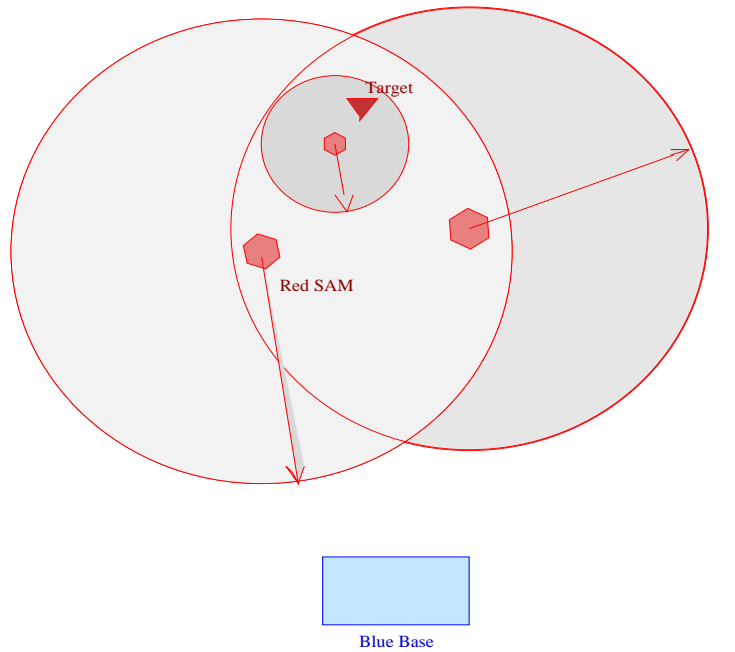
1. The value still has two “regions”. In the left region it is roughly linear (over the region), and in the right region it is constant. Again the structure is essentially the same.
2. The standard deviation increased.
3. The switch-over point from fly-over to rollback is different.
4. A close examination of Figure 2.3.16 (surviving SAMs) indicates that the switch-over is not quite complete from one control α to the next. 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.

2.4 Differing Geographies

The controller can deal with more complicated geometries than that of the previous data sets. We have run it for the geographical configuration depicted in Figure 13. The geographical configuration is employed in the computations simply by determining the Red SAM sites whose kill zone must be flown through in going from SAM/target A to SAM/target B for each possible pair (A, B) . This information is stored in a file and utilized in the computations as described above. Thus, all possible sets of interdependencies can be stored in files and used as appropriate. One should note that the number of possible sets of interdependencies would grow rapidly for large configurations. Enumeration and approximation of such sets would be a subject of further study if this technique was to be used on larger problems than those being considered at this level in the hierarchy. Monte Carlo results have been generated for this example geometry (i.e. Figure 13) as well. It is worth noting that this geometry is nearly at the opposite end of the spectrum from that considered in the plots depicted above; In this geometry, the SAM kill zones are greatly overlapping. The plots for this geometry, depicted on Figures 14–18 are nonetheless rather similar to those above.

2.5 Differing SAM lethalties

Previously, there was only one type of SAM in our Small Game Controller and Test-bed. In this subsection, we indicate the changes that were made to allow the SAMs to



Geography 2 Di

Figure 13:

have both different effective radii and different lethality (lethalities). The possibility of different lethality SAMs had a significant effect on the shape of the value landscape and the optimal controls.

Note that the different radii are an off-line matter which is eliminated in the distillation of the geography to a directed graph as discussed above. Thus different radii (or even more general coverage shapes) do not affect the computations, nor do they increase the number of cases to be considered.

In the previous subsection, the optimal Blue strategy was essentially always either fly-over or rollback. With multiple SAM lethalties, the number of possibilities increases. With two SAM lethalties, there are three control regions for Blue (fly-over, partial rollback, rollback). See Figures 19–22 for an example with two air vehicles, two SAMs and a target. The SAMs had two different lethalties.

Note that there are now two discontinuities in the Blue control policy. An obvious question is whether the two control policy jumps in those figures were due to two SAM lethalties or to two SAMs? Figures 23–24 are for two air vehicles and three SAMs where the SAMs have two lethality types. Two SAMs are weaker and one is stronger while under direct attack. the fly-over damage probabilities are the 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.

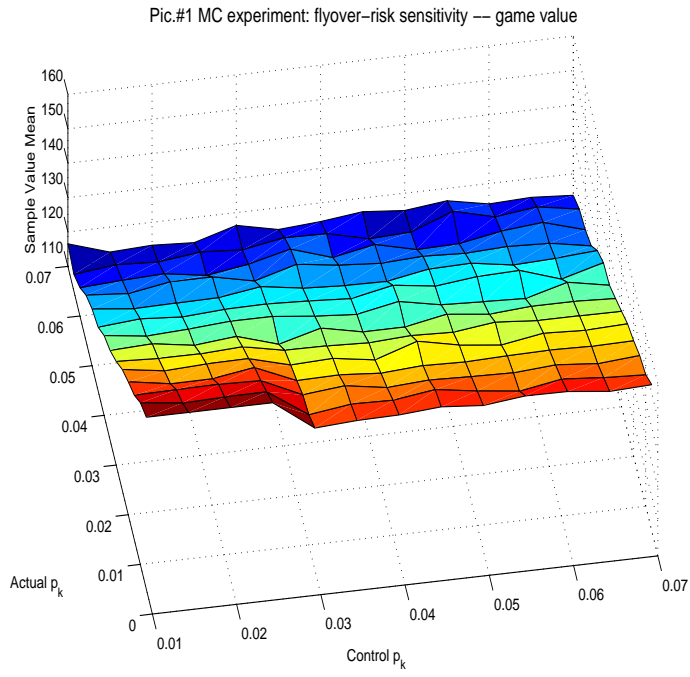


Figure 14:

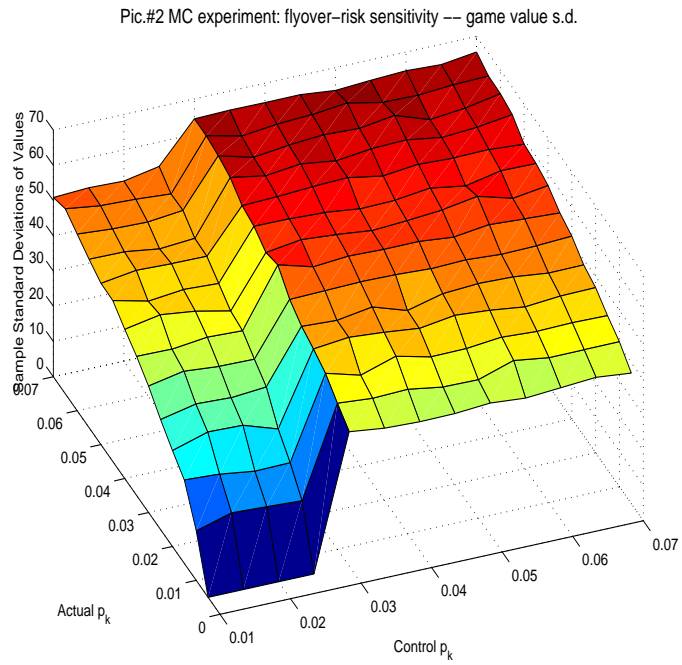


Figure 15:

Pic.#3 MC experiment: flyover-risk sensitivity -- avg. surviving A/C

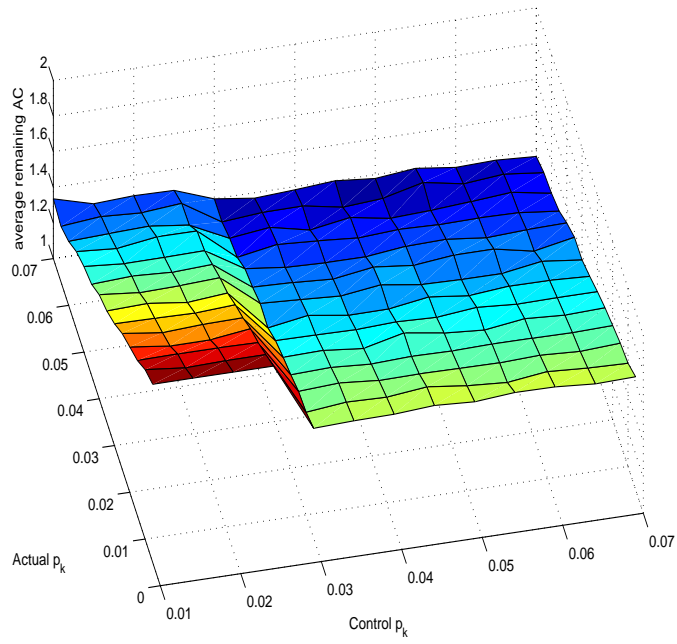


Figure 16:

Pic.#4 MC experiment: flyover-risk sensitivity -- avg. surviving SAMs

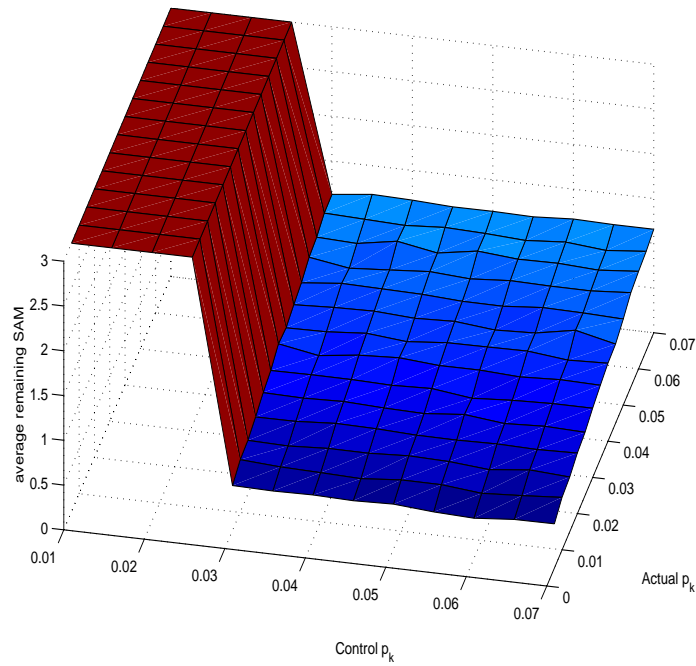


Figure 17:

Pic.#5 MC experiment: flyover-risk sensitivity -- avg. surviving targets

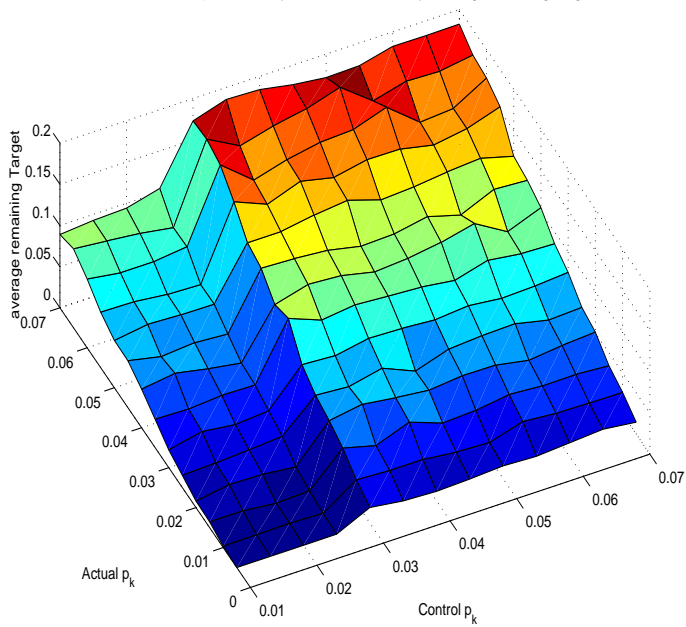


Figure 18:

Pic.#1 MC experiment: flyover-risk sensitivity -- game value

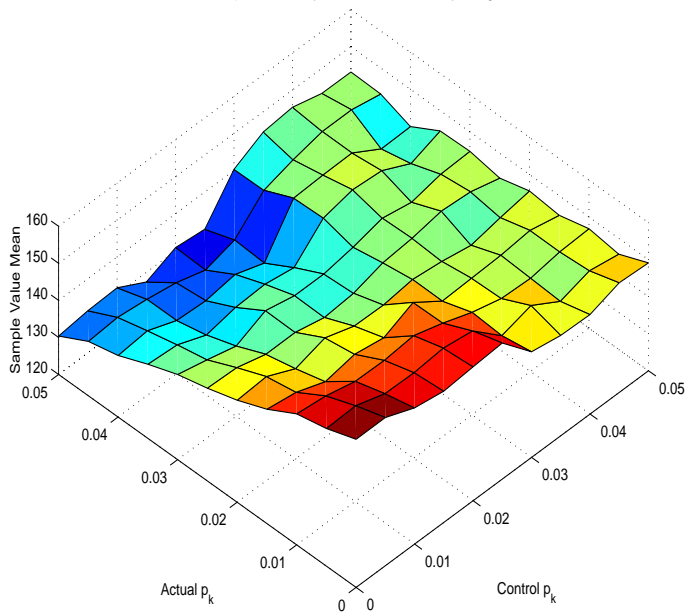


Figure 19:

Pic.#2 MC experiment: flyover-risk sensitivity -- game value s.d.

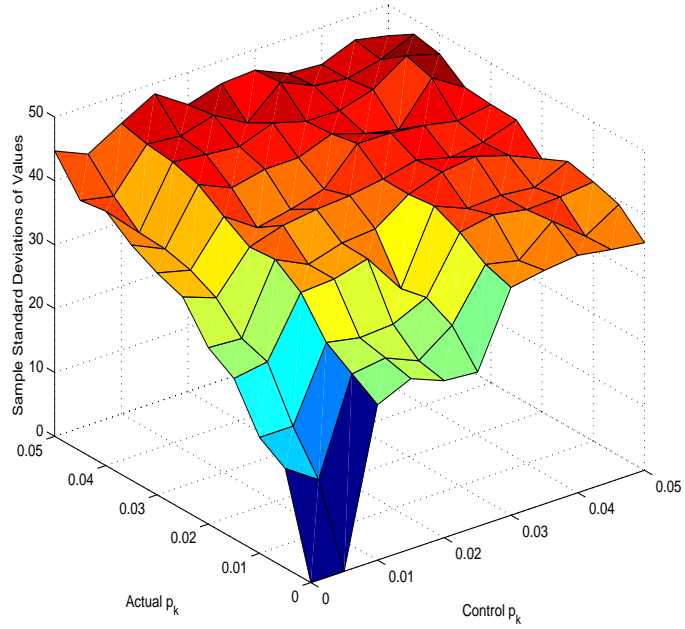


Figure 20:

Pic.#3 MC experiment: flyover-risk sensitivity -- avg. surviving A/C

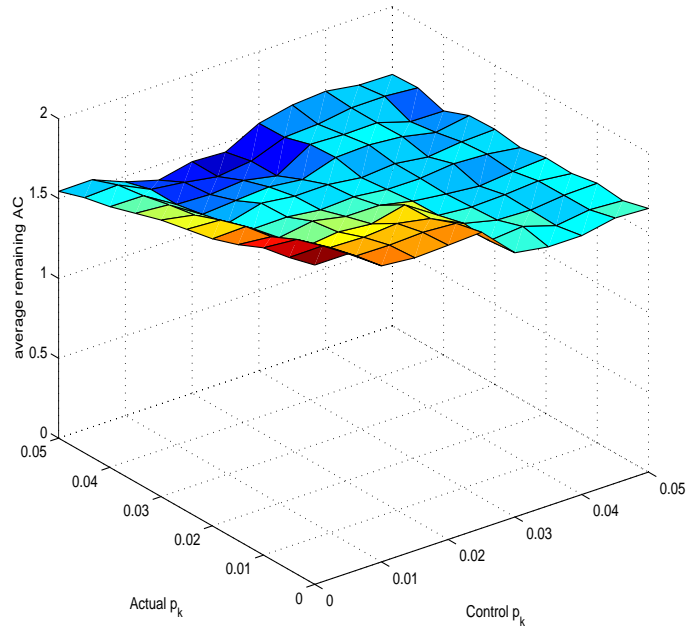


Figure 21:

Pic.#4 MC experiment: flyover-risk sensitivity -- avg. surviving SAMs

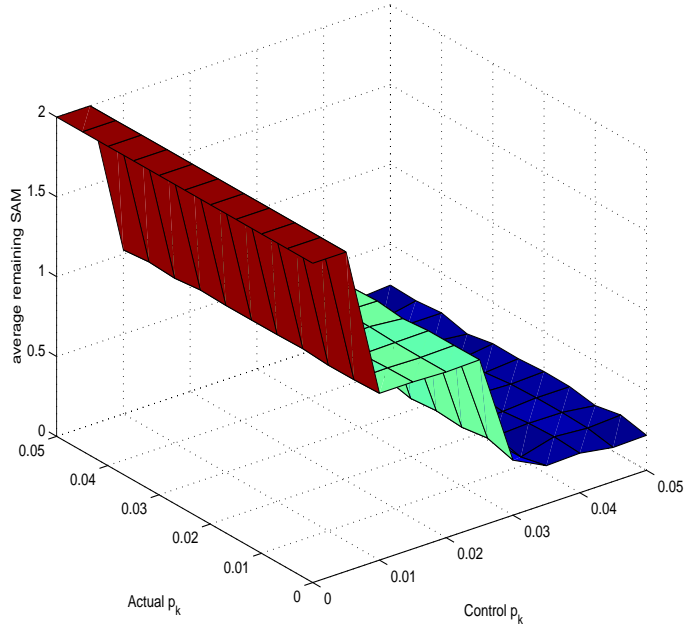


Figure 22:

Pic.#3 MC experiment: flyover-risk sensitivity -- avg. surviving A/C

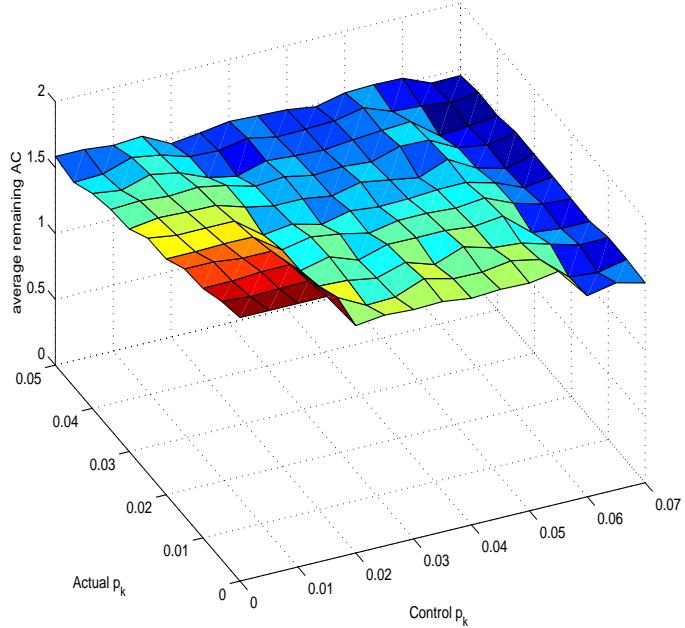


Figure 23:

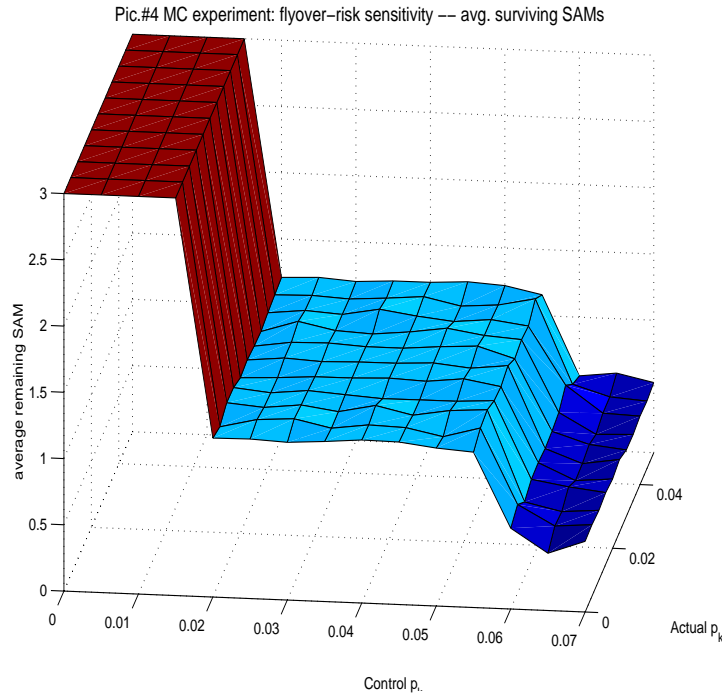


Figure 24:

The main result here is that one does not need to search over all of Blue air vehicle control space. The search appears to be reduced to only $n + 1$ (or possibly 2^n if there is not a simple single ordering of differing SAM capabilities) 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.

Note also that in the decision process, one would be interested in the change in standard deviation across the discontinuities as well as the change in average value. In some cases, a reduction (improvement) in game value across a Blue strategy discontinuity is accompanied by an increase in standard deviation of the outcome – which may (or may not) be undesirable. Further, one would be interested not only in optimal planning, but also closeness to the switching boundaries.

3 The C^2 Estimation and Control under Imperfect Information

The work of the previous section, as with much of the research into stochastic games, involves control 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 air vehicles against Red player SAMs and targets. However, the Red player will have the additional capability of using decoy RF emitters to confound the Blue player's attempt to suppress defenses.

Recall that $Y_i^A(t) = (D_i^A(t), X_i^A(t))$ denotes the health and position state of air vehicles i , $Y_i^R(t) = (D_i^R(t), X_i^R(t))$ denotes the health and position state of the i^{th} SAM, and $Y_i^T(t) = (D_i^T(t), X_i^T(t))$ denotes the health and position state of the i^{th} target. To these, we add a state variable for the decoy emitters, denoted by $Y_i^E(t) = (D_i^E(t), X_i^E(t))$. These state variables comprise the true system state.

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 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 so as to balance the objective function and its measurement of risk with the probability distributions of 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.

3.1 The Estimation Problem for Blue

We assume that the Blue player begins with an intelligence preparation of the battlefield that includes a fixed (perhaps large) number of sites that may contain SAM systems or decoys (or nothing). Furthermore, the intelligence indicates an upper bound on the possible number of SAM systems fielded by Red. Thus, the goal for Blue's information analysis is to decide which of all the potential sites are SAM sites.

We denote by g_1, \dots, g_K the list of sites which could potentially hold a SAM or a decoy, and let N_m denote the maximum number of SAMs (which being assumed known here through some "intelligence preparation of the battlefield"). We define the identification

of Red entities at a site g by

$$I(g) = \begin{cases} 1 & \text{if a SAM is at } g \\ 2 & \text{if a Decoy is at } g \\ 3 & \text{if nothing is at } g, \end{cases}$$

and we extend I by componentwise evaluation to operation on the vector g_1, \dots, g_K . The set of all identifications has 3^K elements, but this set is reduced significantly if we take into account the constraint that the number of 1's occurring in the vector must be $\leq N_m$ where N_m is the maximum number of SAMs (obtained from an exogenous information source). We denote by \mathcal{S} the set of all K tuples with elements from $\{1, 2, 3\}$ having no more than N_m 1's. Blue's information state, P_t^b , at time t is a probability mass function on the set \mathcal{S} .

Let $N_{\mathcal{S}}$ denote the number of elements in \mathcal{S} . We initialize the Blue information state by taking $P_0^b(s) = 1/N_{\mathcal{S}}$, for each $s \in \mathcal{S}$.

We assume that the Blue electronic system has signal processing capability to detect and classify emitters 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 denote by $p_d^b(w, e)$ the probability that a Blue air vehicle detects a defensive entity (SAM radar or emitter) that has emission property e , from a distance w , and we define $p_c^b(w, e)$ to be the probability that a Blue air vehicle correctly classifies a Red entity that has emission property e at a distance w . The emission property e is either 0 (off) or 1 (on). Of course, if there is no entity at the location under consideration, then we must have $e = 0$. 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 that can be intergrated into the approach.

To build up an information state vector, we must integrate the observations from each grid cell and from each air vehicle. We treat the air vehicle measurements as independent, and we also assume that each grid cell's observations are independent, for each fixed air vehicle. We denote the i^{th} air vehicle's observation as $Y^i = (Y_1^i, \dots, Y_K^i)$, where we assume each $Y_k^i = 1, 2, 3$. The full observation then is $Y = (Y^1, \dots, Y^{N_A})$, and the conditional probability of observing $y = (y^1, \dots, y^{N_A})$, given a Red state configuration s and an emission vector e is given by:

$$p_o^b(y) = \prod_{i=1}^{N_A} \prod_{k=1}^K \left[p_d^b(w_{i,k}, e_k) p_c^b(w_{i,k}, e_k) I_{y_k^i = s_k} + p_d^b(w_{i,k}, e_k) \frac{1 - p_c^b(w_{i,k}, e_k)}{2} I_{y_k^i \neq s_k} + \frac{1}{3} (1 - p_d^b(w_{i,k}, e_k)) \right]$$

where $w_{i,k}$ is the distance from air vehicle i to site k .

We apply the standard Bayesian approach to modeling the information state updates. We denote by $P_t^b(s)$ the conditional probability of the Red state being s given the observations up through time t . Then,

$$P_{t+1}^b(s) = \frac{p_o^b(y(t+1))P_t^b(s)}{\sum_{s' \in \mathcal{S}} p_o^b(y(t+1))P_t^b(s')}.$$

We remark here that without the constraint on the number of SAMs, the entity types at the locations could be treated as independent. The constraint implicitly introduces a correlation between sites, since the total number of SAMs must be at most N_m .

This propagation provides the Blue player with a means of inferring the locations of SAMs and decoys from the sensor data. This simple model captures the essential features of planning (and replanning) in the presence of uncertainty and information collection over time.

3.2 The Estimation Problem for Red

The estimation problem for Red involves tracking instead of classification (we have only one type of Blue air vehicle in this analysis). The Red SAM radar system must detect Blue air vehicles and locate them in the battlespace. We denote by $p_d^r(w)$ the probability that a Red SAM is able to detect an air vehicle present at a distance w . Of course, to detect and track, the SAM radar must be on. If the SAM is on, then detection of any given air vehicle is a random event where the detect probability depends on the distance from the SAM to the air vehicle. In particular, we take the detect probability to be $p_d^r(w) = p_{d0}^r \exp(-w/w_0)$ where w is the distance from the SAM to the air vehicle, and w_0 is a scaling parameter. This functional form is chosen for illustrative purposes: any other form can be used as well. 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 (i.e., no altitude component). For the purposes of understanding the impact of uncertainty in the game outcome, the additional complexity of altitude tracking is not important.

The Red player then uses a sub-optimal filter to track the Blue air vehicle. 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, $P_1^s(t)$

being the probability that the air vehicle is such that $S_R(t) = 1$ (i.e. the probability that the air vehicle 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 since altitude is not modeled).

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 air vehicle's dynamics:

$$\begin{aligned}x(t + \Delta_t) &= x(t) + \Delta_t v(t) + w_x(t) \\v(t + \Delta_t) &= v(t) + w_v(t),\end{aligned}$$

in which x and v denote the air vehicle 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 air vehicle which they detect, and that the Red defense pools the information into an observation vector $Y(t)$. For an air vehicle at position $x(t)$, the components of $Y(t)$ are

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

where the i subscript indicates the i^{th} SAM radar's observation of position and ζ_i is the noise (assumed Gaussian and spherical) in the i^{th} observation. We used only a very simple observation model and applied only a basic Kalman filter here as the goal is to begin to understand the effect of partial information on game-theoretic controls.

Finally, we 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 air vehicle that was observed. Generally speaking, track association is, in and of itself, a challenging problem with many interesting phenomena to be studied. However, the track association problem is not relevant to the study we are making here, and the additional complication would obscure some of the basic properties of our investigation of the C^2 problem at hand.

3.3 Blue Control under Imperfect Information

Having defined our information states in terms of probabilistic models and Bayes' rule, we now proceed with the tasks of developing state estimators and integrating observers into the control system. As noted above, the goals of this project involve developing and understanding control strategies for air operations in the presence of uncertainty. Our particular interest has been in strategies that counter intelligent opponents in a robust way. Toward that end, we seek here to include the 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.

One approach is to embed the problem into an infinite dimensional full state feedback control system (see, for instance, [15], [20]). Certain theoretical results are known for such

an approach, but not within the stochastic game context. Moreover, the computational burden is prohibitive, even with the small problems under consideration here.

Thus, current practice relies on traditional approaches to output feedback control, involving 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/controller, discussed below, that allows us to tune the relative importance between the likelihood of possible states and the risk of mis-estimation of the state. Let us motivate this proposed approach in a little more detail.

In deterministic games under partial information, the certainty equivalence principle indicates that one should use the optimal control corresponding to the state given by

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

where P is the information state and V is the value function (assuming uniqueness 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 ([20], [15] [19], [26], [27].) The next three paragraphs discuss the mathematics which lead to the heuristic for the algorithm described in 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 a class of linear/quadratic problems.)

A risk-averse stochastic control problem is given by

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

This risk-averse stochastic control problem is equivalent to the stochastic game [9], [10], [11], [29]:

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

Both have the same Dynamic Programming Equation:

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

It is by now well-known that risk-averse control converges to a deterministic game as $\varepsilon \downarrow 0$ ([10], [11], [12], [29], [18], [31], [33],[17]). All of this yields motivation for 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

$$p(t, x) = k(t) \exp\left\{-\frac{1}{2}(x - \bar{x}(t))^T C^{-1}(t)(x - \bar{x}(t))\right\}.$$

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

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

Interestingly, Q and C^{-1} satisfy the same Riccati equation (or, equivalently, Q^{-1} and C satisfy the same Riccati equation). \hat{x} and \bar{x} satisfy identical equations as well. Therefore, $P(t, x) = \log p(t, x) +$ “time-dependent constant” [27], [8].

The above three paragraphs motivate the algorithm proposed and studied herein. This algorithm is the following: apply state feedback control at

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

where p is the probability distribution based on the observation process and filter for Blue of Section 3.1, and V is state feedback stochastic game value of Section 2. 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.

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

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 [15], [20], i.e.

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

As $\kappa \rightarrow \infty$, 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.

3.4 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 integrate observations through the Bayes’ propagation defined above, and the states estimates are applied via the above generalized certainty equivalence principle. The control inputs are then determined accordingly.

The output feedback control developed here relies on computation of the value and strategies from the full state feedback computations. However, these computations can be performed in advance, off-line, and stored. The generalized certainty equivalence principle is then used to access these pre-computed full state feedback controls.

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

Figure 25, which is a snapshot of the simulation in progress, illustrates the process. Included in this image are air vehicles (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 circles indicate kill radii for the SAMs. The bar graphs to the right of the battle cartoon indicate the classification probabilities 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 air vehicle position estimates for the Red information state.

Applying this simulation for many engagements in a Monte Carlo study, we can assess the expected value for a particular scenario. In Figure 26, we have selected a scenario which has 3 SAM sites, 2 decoy emitters, and 2 air vehicles attacking two targets. Running the simulation for 2000 Monte Carlo samples, we can assess the impact of the risk-averse

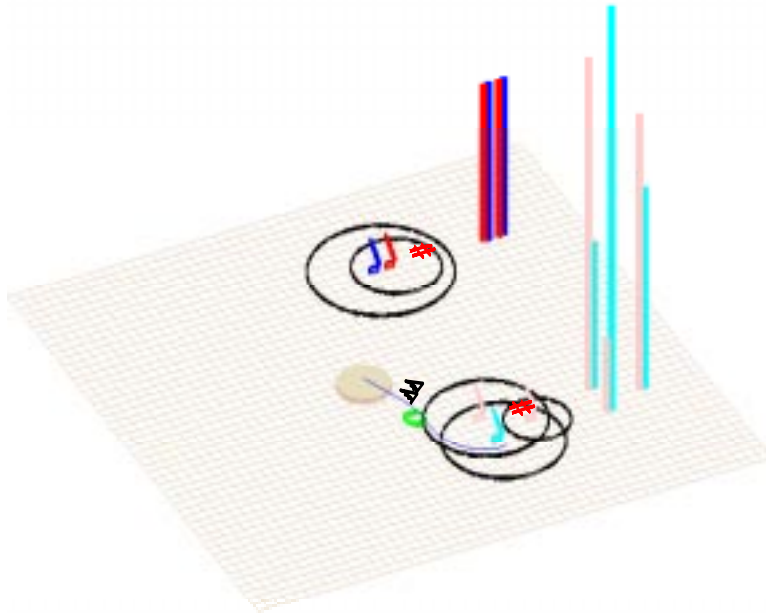


Figure 25:

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 ($\kappa = 0$) and the deterministic game based approach ($\kappa = 1$) which assumes all noise is antagonistically generated. Applying the traditional separated controller/estimator approach ($\kappa = 0$) produces reduced performance, which means that the Blue player is more likely to lose air vehicles 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.

References

- [1] M. Adams, W.M. McEneaney et al., “Mixed Initiative Planning and Control under Uncertainty”, Proceedings First AIAA UAV Symposium, Portsmouth, VA, May 22-25, (2002).
- [2] T. Basar and P. Bernhard, *H_∞-Optimal Control and Related Minimax Design Problems*, Birkhäuser (1991).
- [3] D.P. Bertsekas, *Dynamic Programming, Deterministic and Stochastic Models*, Prentice-Hall, Englewood, 1987.
- [4] 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.

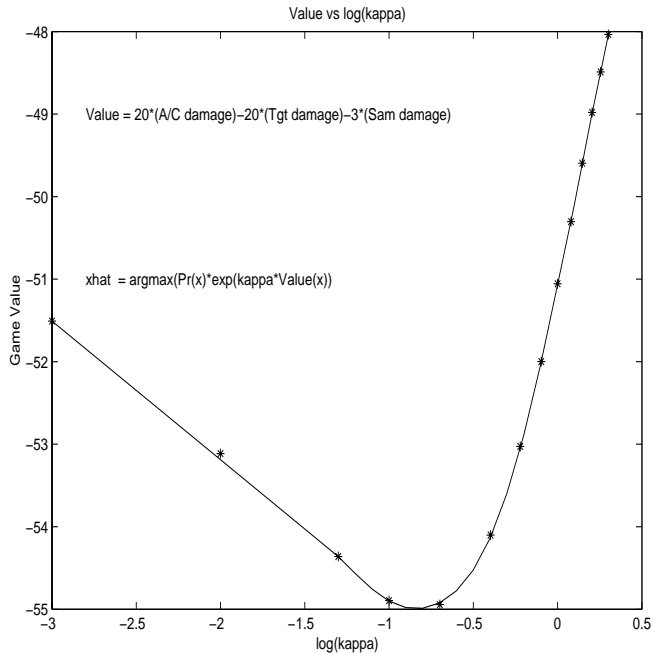


Figure 26:

- [5] 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.
- [6] R. J. Elliott and N. J. Kalton, “The existence of value in differential games”, *Memoirs of the Amer. Math. Society*, **126** (1972).
- [7] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*, Springer (1997).
- [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, “Robust limits of risk sensitive nonlinear filters”, *Math. Control, Signals and Systems* **14** (2001), 109–142.
- [10] 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.
- [11] W. H. Fleming and W. M. McEneaney, “Risk-sensitive control with ergodic cost criteria”, *Proceedings 31st IEEE Conf. on Dec. and Control*, (1992).
- [12] 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).
- [13] 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.

- [14] 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.
- [15] J.W. Helton and M.R. James, **Extending H_∞ Control to Nonlinear Systems**, SIAM (1999).
- [16] 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.
- [17] D.H. Jacobson, “Optimal stochastic linear systems with exponential criteria and their relation to deterministic differential games”, IEEE Trans. Automat. Control, 18 (1973), pp. 124–131.
- [18] M.R. James, “Asymptotic analysis of non-linear stochastic risk-sensitive control and differential games”, Math. Control Signals Systems, 5 (1992), pp. 401–417.
- [19] 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.
- [20] 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.
- [21] J. Jelinek and D. Godbole, “Model Predictive Control of Military Operations”, Proc. 39th IEEE CDC, Sydney (2000), 2562–2567.
- [22] W.M. McEneaney and B.G. Fitzpatrick, “Control for UAV Operations under Imperfect Information”, Proceedings First AIAA UAV Symposium, Portsmouth, VA, May 22-25, (2002).
- [23] W.M. McEneaney and K. Ito, “Stochastic Games and Inverse Lyapunov Methods in Air Operations”, Proc. 39th IEEE CDC, Sydney (2000), 2568–2573.
- [24] W.M. McEneaney, “Convergence and Error Analysis for a Max-plus Algorithm”, 39th IEEE Conf. on Decision and Control, Sydney (2000), 1194–1199.
- [25] W.M. McEneaney, K. Ito, Q. Zhang, B. Fitzpatrick and I. Lauko, DARPA JFACC Project, NCSU Team Final Report, (2000).
- [26] W.M. McEneaney, “Robust/game-theoretic methods in filtering and estimation”, First Symposium on Advances in Enterprise Control, San Diego (1999), 1–9.
- [27] W.M. McEneaney, “Robust/ H_∞ filtering for nonlinear systems”, Systems and Control Letters, **33** (1998) 315–325.

- [28] W.M. McEneaney, “Robust control and differential games on a finite time horizon”, *Math. of Controls Signals and Systems*, **8** (1995), 138–166.
- [29] 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.
- [30] H. Mukai, et al., “Game-Theoretic Linear-Quadratic Method for Air Mission Control”, *Proc. 39th IEEE CDC, Sydney* (2000), 2581–2586.
- [31] T. Runolfsson, “Risk-sensitive control of Markov chains and differential games”, *Proceedings of the 32nd IEEE Conference on Decision and Control*, 1993.
- [32] 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.
- [33] P. Whittle, *Risk-sensitive linear/quadratic/Gaussian control*, *Adv. Appl. Prob.*, **13** (1981), pp. 764–777.