AIAA-2002-3418

CONTROL FOR UAV OPERATIONS UNDER IMPERFECT INFORMATION

William M. McEneaney * Ben Fitzpatrick †

Abstract

We address Command and Control (C^2) problems for unmanned air vehicles (UAV's) within the framework of stochastic games. The problem we consider involves one unit comprising a few UAV's (typically in the range of two to ten) attacking a small number of targets. The targets are defended by surface-to-air missile (SAM) systems with missiles and associated search and track radars. The opponent may also employ decoy SAM radars. Our goal here is to develop a stochastic game formulation that can provide output feedback controls in the presence of uncertainty and partial information. We approach this goal by combining the estimator and controller via a modified Certainy Equivalence Principle that weighs both the probability of each possible state and the potential cost such system state in a mathematically appropriate way, so as to determine a near optimal control.

1 Introduction

The introduction of unmanned air vehicles (UAV's) involves significant challenges for battle planners. Autonomous and semi-autonomous implementation of certain Command and Control (C^2) functions can reduce workload for human operators and command personnel. Such C^2 automony must necessarily be developed within a risk-averse framework. In this paper, we consider C^2 problems for UAV's within the framework of stochastic games. The particular problem we use to illustrate our techniques involves one team or unit comprising a few UAV's (typically in the range of two to ten) attacking a small number of targets. We denote the attacking team by Blue and the defending team by Red. The targets are defended by surface-to-air missile (SAM) systems with associated search and track radars. Red may also employ decoy SAM radars to confuse (and help defeat) Blue. Our goal is a methodology for devising output feedback control strategies for Blue that operate with noisy, partial information, against an intelligent adversary. The techniques we develop here involve a stochastic game formulation, with a risk-averse Certainty Equivalence approach to state estimation and feedback.

Most of the work in stochastic games and their applications has been done under the assumption of full state information (i.e. full state feedback). However, partial, imperfect and even purposefully corrupted information is a critical part of warfare. After a discussion of full state feedback games, we will begin discussing some simple algorithms for estimation of system state given likely data types. Then we will examine control under imperfect information, and present some initial numerical results.

The first basic idea is that the players handle

*Depts. of Mathematics and Mechanical/Aerospace Engineering, University of California at San Diego, La Jolla, CA 92093-0112, http://math.ucsd.edu/~wmcenean/, wmceneaney@ucsd.edu, Research partially supported by DARPA grant F30602-99-2-0548 and NSF grant DMS-9971546.

†Department of Mathematics, Loyola Marymount University, and Tempest Technologies LLC 8929 Sepulveda Blvd, Suite 208, Los Angeles, CA 90045 310-216-1677, fitzpatrick@tempest-tech.com

uncertainty by maintaining probability distributions on the location and number of the opposing player's forces. As in most traditional approaches to output feedback control, these probability distributions allow each player to estimate the likely states of their 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.

The second basic idea is that, in a stochastic game setting, the estimator should take into account not only the likelihood of the opponent's 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.

Finally, we remark that, while very small in terms of vehicle and target numbers, the problems posed herein are important from a practical point of view. We have considered hierarchical decomposition techniques for building vehicle teams, separating the battlespace, and constructing optimal routes to targets. These decompositions (which we will discuss in future publications) allow us to build large-scale command strategies with building blocks from the study of small problems so that optimal C^2 strategies can be found in computationally tractable ways.

2 The Basic Game Problem and Full State Feedback

We begin with the basic problem of two players operating under full information but with competing goals. The model for this situation is a min-max stochastic game. The game for this C^2 problem involves Red and Blue assets to be preserved/attacked in a battle. The control objective that Blue seeks to minimize and Red seeks to maximize is given by an exit time payoff functional of the general form

$$J(x, u_B(\cdot), u_R(\cdot)) = E\big\{F(X_B(T), X_R(T))\big\},$$

in which F denotes the cost at the (random) exit time, T, $x = (x_b, x_r) = (X_B(0), X_R(0))$ and $u_b(\cdot)$ and $u_r(\cdot)$ are the controls (as functions of time). Here exit is defined in terms of attrition, meaning one player losing all assets. The state vectors X_B and X_R, which denote locations and health states of player assets, have controlled Markow chain dynamics through their controlled missions and the random outcomes of engagments. In many cases of interest, we may choose F to be a linear function of the state vectors. This type of model assigns values to different types of assets through the linear coefficients. In order to solve the problem as a game, coefficients will be positive for the Blue player and negative for the Red player (so that Blue minimizes and Red maximizes). From this point of view, assets are differentiated by the players in terms of value.

This fairly simple structure is sufficiently flexible to model a great variety of asset configurations. The Blue player can have different types of aircraft, and Red may have multiple types of defenses and other assets. The flexibility comes at a price, however: we must define values of assets (through the choice of F) and transition probabilities. For this discrete stochastic game problem, the complexity grows as the number and number of types of assets grows. Computational tractability is a serious issue. As noted above, we are working to develop hierarchical decompositions to mitigate the curse of dimensionality.

Our approach to solving the control problem is through the upper value of the game,

$$V(x) = \min_{u_B} \max_{u_R} J(u_B, u_R, x),$$

whose optimizing feedback controls are denoted by $u_B^*(x), u_R^*(x)$. The upper value is determined numerically using dynamic programming and a fairly standard branch and bound techniques. The computation provides a full state feedback control strategy.

A major difficulty, as noted above, is the fact that neither player has access to the exact value of the state vector. Red air defenses must track the Blue aircraft, and if Blue employs multiple types of aircraft with differing functionalities, then Red may need to discriminate Blue types from observation data. Likewise, Blue generally has imperfect information on Red configurations. Red may employ decoys of targets and air defense systems, and assets may be mobile. The approach for output feedback described below is designed to provide a risk-sensitive state estimator that may be fed into the above full state feedback controls [14], [15].

3 The Information State Variables

In order to develop an output feedback approach, we construct information state models that relate observations to the states. Our approach follows the standard Bayesian statistical model of information propagation: probability distributions over the state space are maintained by the players. As each player obtains information through measurements, the probabilities are updated.

The formulation of the 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 $s \in \mathcal{S}$ 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. Currently, there is no software component for Red classification of Blue air vehicles.

The Blue information state models the likelihood of Red entities located at coordinates on a grid of the physical battlespace. This model is a $N_{RT}+1 \times G$ matrix, B, whose entries are the probability of a Red entity of a given type at a particular grid point. That is, $B_{k,g}$ is the probability of a type k entity at grid location g. There are G grid points and N_{RT} types (with an additional type indicating no entity).

For our illustrative example, we assume that Blue knows the location of targets but lacks information on the location of SAMs and emitter decoys. Thus, the number of Red entity types that must be modeled in the information state is $N_{RT} = 2$, so we use $k = 1$ to denote SAM, $k = 2$ to denote emitter, and $k = 3$ to denote no defensive entity.

4 Information Probability Modeling

The information states must be propagated by each player, depending on the type (and quality) of information they obtain. Detection, classification, and location 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.

Generally speaking, in each case, the players maintain probability distributions of the form $p(t, x)$ (where here x indicates a generic variable). The probabilites are propagated via Bayes theorem:

$$p(t+1,x) = \frac{f(y(t)|x)p(t,x)}{\sum_{x'} f(y(t)|x')p(t,x')}.$$

Each player must have a model of the observation process encoded by the measurement probability $f(y(t)|x)$. In the following sections, we describe measurement modeling for each of the players.

5 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 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). Once detection has occurred, Blue may then attempt to classify entities as SAM radars or emitters. 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. Performing the standard receiver operating characteristic analysis will lead to a choice of classification regions and a probability of correct classification. As a simple model function for a two-class discrimination problem, we take

$$pbcc(w) = .5 + .5 * \exp(-w/w_0^b),$$

a function which goes to one as Blue nears the Red entity and goes to 0.5 as the distance increases. While this functional form is quite a simplistic model, it allows for investigation of the concept. More complicated functions can readily be incorporated into this framework.

As noted above, we apply the standard Bayesian approach to modeling the information state updates. For the Blue state, we construct the measurement system using the discrimination model above. The Bayesian formulation allows us to determine the probability that a SAM, emitter, or nothing is at grid location g given what we've observed at the current time.

To pool the information from multiple aircraft sensors, we define

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

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

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

through an application of Bayes' rule.

6 The Estimation Problem for Red

For the Red infomation problem, 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. We denote by $p_d^r(w)$ the probability that a Red SAM detects an aircraft present at a distance w. For simplicity, we take the detect probability to be

$$p_d^r(w) = \frac{1}{1 + (w/w_0^r)^2}$$

where w is the distance from the SAM to the aircraft, and w_0^r is a scaling parameter. More general models can easily be integrated into our methodology. 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, $P_1^s(t)$ 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:

$$\begin{aligned} x(t + \Delta_t) &= x(t) + \Delta_t v(t) + w_x(t) \quad (1) \\ v(t + \Delta_t) &= v(t) + w_v(t), \quad (2) \end{aligned}$$

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.

7 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. We seek here a computationally efficient control algorithm that maintains a risk-averse approach. Determination of an optimal feedback functional is computationally very intensive (if not completely intractable), and some approximations must be considered.

The 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. However, direct feedback of a maximum likelihood state estimator can not be guaranteed to provide even near optimal performance in the types of problems in which we are interested.

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 concept 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 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 ([9], [8], [11], [12].) 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:

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

Both have the same Dynamic Programming Equation:

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

It is by now well-known that risk-averse control converges to a deterministic game as $\varepsilon \downarrow 0$ ([1], [2], [3], [13]). All of this lends credibility to a study of the use of the above Certainty Equivalence approach for our problem (although it will be suboptimal).

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

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

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.

8 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 as noted in Section 2 (cf. [14], [15]). 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.

For this simulation, Blue assets are striker aircraft that can attack any Red asset. Red assets are one of the following three entities: targets (which cannot fight back), SAM air defense systems, and decoy emitters (which appear to be similar to SAMs as discussed above). Blue controls are of a path planning nature: which Red asset do we strike now? For control purposes, one may distill complex geometries down to a few cases. Of primary importance are the coverage "umbrellas" of the Red SAM systems. Relative distances and locations are not as important as whether or not the paths between are "protected" by Red defenses. In general, one may define the Red state space in terms of directed graphs whose nodes denote assets and whose (directional) edges denote assets "protecting" each other.

It should be noted that the determination of paths in the physical geometry is a significant challenge. Determining detailed Blue paths within the framework of the stochastic game is computationally very intensive. However, the main impact of the physical path is in the period of time and proximity of danger from air defense. In this example, Blue paths are generated using an obstacle-avoidance algorithm that lays out a path to targets avoiding potential SAMs that are not selected to be attacked [14].

Controls for Red are turning SAM radars and emitters on and off and attacking Blue aircraft with SAMs. The Markov chain involves transition probabilities for engagements between aircraft and SAMs/targets.

We note here that the 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 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 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.

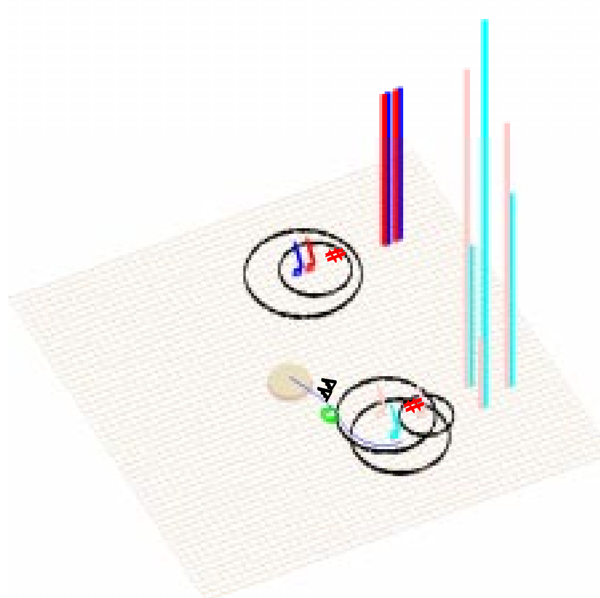

Fig. 1

Applying this simulation for many Monte Carlo engagements, we can assess the expected value for a particular scenario. In Figure 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 $\kappa = 1$ approach (which assumes all observation disturbances are antagonisitc). 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–sensitive approach. Note that the horizontal axis is on a log scale, so that the minimum in κ is rather broad.

Another experiment conducted compares the behavior of maximum likelihood and risk–averse feedback for random and "intelligent" opponents. By intelligent we mean that the Red player employs the control strategy that arises from the min-

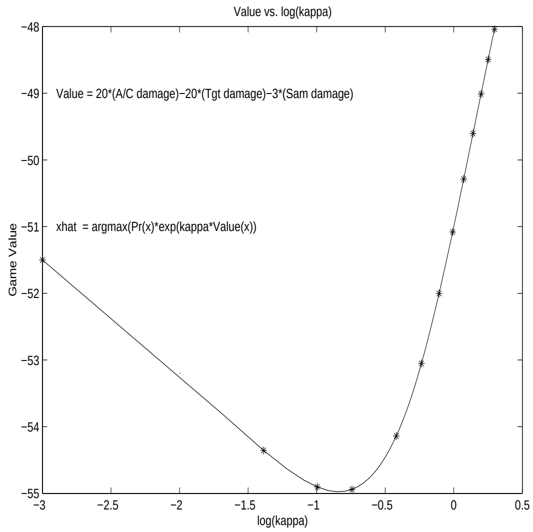

Fig. 2

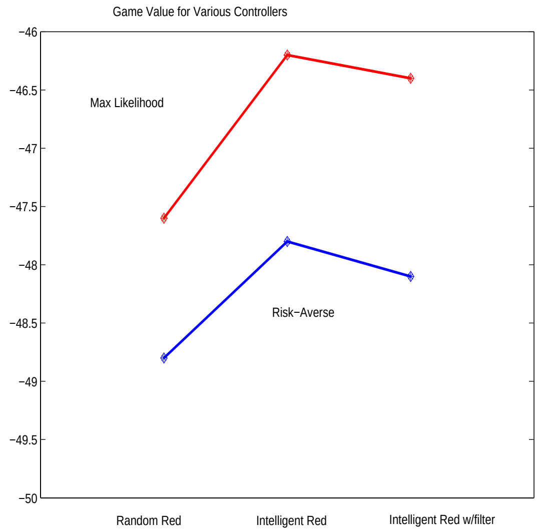

Fig. 3

imax solution. In this scenario, we ran 500 Monte Carlo samples for each of 6 configurations. The differences are in Blue strategy (maximum likelihood and risk-averse) and Red strategy (random, intelligent with full information on Blue, and intelligent with filter/prediction of Blue behavior). Figure 3 illustrates some of the effects. In that figure we see that Blue benefits in either estimation approach when Red employs a random control strategy.

References

[1] 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.

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

[3] 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).

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

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

[13] 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.

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

[15] W.M. McEneaney, B. Fitzpatrick, K. Ito and Q. Zhang, DARPA JFACC NCSU Team Final Report, (2001).