A Computationally-Feasible Algorithm for Estimation of Opponent Strength in Urban Combat

William M. McEneaney [*] Rajdeep Singh [†]

Abstract

An estimate of the opponent's forces is both a useful object for the commander and an input required by other tools being developed as computational aids for C^2 in urban combat. The standard approach would be to maintain probability distributions over the set of potential opponent force distributions. However, that approach is not computationally feasible. In this paper, we discuss an alternative estimator which requires only a small fraction of the computational power of a laptop computer when operating on realistically-sized problems. The algorithm is developed, error bounds are obtained, and an example of the estimator operating in conjunction with an urban combat C^2 simulator is presented.

1 Introduction

We consider the problem of constructing automated Command and Control (C^2) tools which may be employed as aids in urban combat. In particular, we will focus here on the estimation/reconnaissance component of such tools.

Such a tool could provide the local commander with proposed courses of action, likely potential maneuvers of the opposing forces, as well as estimated enemy positions and strength at those positions (which we refer to as an opponent laydown). The latter item is not only an output object, but is also

[*](corresponding author) Dept. of Mech. and Aero. Eng., Univ. of Cal. San Diego, La Jolla, CA 92093-0411, wmceneaney@ucsd.edu, 858-822-5835. Research supported by DARPA Contract NBCHC040168 and AFOSR Grant FA9550-06-1-0238.

[†]Integrated Systems & Solutions, Information Assurance Group, Lockheed-Martin, rajdeep.singh@lmco.com. Research supported by DARPA Contract NBCHC040168.

an intermediate object needed for computation of the first two items. The proposed tools also need to produce this information in a timely manner with at most the computing resources of a typical modern laptop computer. In particular, the computation of the estimated enemy positions and strength, must be done in real-time with computational resources which are only a rather small fraction of those of a hypothetical laptop.

In the algorithm development below, we consider a problem where the opposing forces consist of indistinguishable fire-teams. The additional complexity of a heterogeneous mixture of opposing forces introduces complexity into the computations and algorithm presentation, but does not change the fundamental techniques developed here. Consequently, we limit the discussion to the case where the opponent forces consist of indistinguishable fire-teams.

In order to help the reader visualize the problem and tool prior to the extensive mathematical development in Sections 2 to 6, one can refer to Figure 3 in Section 7. That figure depicts three snapshots from a tool using this estimator in an urban combat C^2 simulation. (The tool has been in development under the DARPA RAID Program.) The red and blue $*$'s indicate the opposing (Red) forces and our (Blue) forces, respectively. The magenta objects denote buildings, the cyan areas are water, and the black dots indicate graph nodes (to be discussed below). The red and green circles are indicative of the estimated Red strength. A fuller discussion of the application appears in Section 7.

The case where there is ongoing attrition also adds complexity, as one needs to keep track of a set of strength distributions (to be defined below) indexed by total opponent strength over the relevant battlespace. Again, noting that the algorithm discussion below is already quite extensive, we ignore the attrition aspect in order to reduce complexity. The fundamental components of the algorithm will be evident regardless. Alternatively, one can pose the following development as a pure reconnaissance problem.

We may suppose that observations of the opponent are coming in from our ground-forces in the field, as well as from aerial assets. We do not address the issue of heterogeneous observation quality in the discussion below. However, if one is interested in such, it will be clear to such a reader that this issue is handled by adjustment of a certain parameter in the observation update.

This class of problems (urban combat) is well-modeled in the form of a stochastic game under imperfect information. The computational requirements for full solution of such problems far exceed what are now, or in the foreseeable future, available. One reasonable approximation is to apply

methods such as those discussed in [7], [8], [9], [10], [11]. The estimation component of such a system still requires the computation of information states in the form of observation-conditioned probabilities or sets of such. As indicated just above, deployable systems would need to require less than the computational capability of a single laptop. Even with that class of methods, information state computations are far beyond the computational limits imposed for such systems. For example, typical problem sizes have on the order of $N = 30$ opponent units (to be described below) distributed among on the order of $L = 500$ locations. (These locations are typically building corners, intersections, rooms, etc...) Suppose the opponent units (say, fire-teams) are indistinguishable, and that more than one unit may occupy any location. Then the number of laydowns (i.e., the number of distributions of the opposing-force units among the locations) is

$$\binom{L+N-1}{N} \simeq 10^{49}.$$

Thus, one would be working with probability distributions over a set of size roughly 10^{49}. Even with some filtering of the set of good locations based on some initial intelligence and line-of-sight information, one can possibly reduce the location set to a smaller one but even then, these are not objects that one can work with within the computational limits of a fraction of a laptop computer (or any current-generation computer for that matter).

Consequently, we look for objects which approximate the information state, and which have greatly reduced complexity. The *strength distribution,* which we present here, will act as a robust estmator for the system. That is, the following algorithm can be used to process the observational data to produce an estimate of opponent positions and strength at those positions.

The main object in this algorithm will be the strength distribution. One can think of this as being analogous to the mean and covariance in a Kalman filter (although the reader should keep in mind that the strength distribution is a suboptimal estimation object being used due to the need to have a computationally feasible algorithm, and so this is not quite analogous to the Kalman filter situation). As with the Kalman filter, there will be both observation updates and dynamics updates within each time-step of the algorithm (where of course, the observations in any given time-step may be the empty set). As we are not including attrition in this document, the dynamics update corresponds purely to potential movement of the opponent forces.

In the following sections, we present the algorithm. We also indicate the results which guarantee that the algorithm does indeed behave well as an

estimator. In this regard, we prove certain input-output noise bounds, i.e., robustness statements. In particular, we do not presume that the opponent forces movements are a purely stochastic process around some nominal. Instead, we assume that there are unknown opposing commander inputs determining force movement. Similarly, the observational noise in such problems should not be presumed to be purely stochastic, as an intelligent opponent may apply deception where it is fruitful to do so. In such a situation, the robust/input-output approach is more reasonable than a purely probabilistic approach (although we do allow for stochastic inputs as well).

In Sections 2–4, we present the algorithm, i.e., the propagation of the strength distribution. In Section 5, we outline the proven performance bounds. Section 6 contains a discussion of an additional approximation technique which further greatly reduces the computational load. Lastly, in Section 7, we present an example of the algorithm operating in conjunction with an urban combat C^2 simulator.

Throughout the remainder, we refer to the opposing forces are Red, and ours as Blue.

2 Strength Distribution

The system state in urban combat C^2 may be taken to be the locations of the forces and their nature. We simplify the problem, by having the basic objects be Blue and Red units where the Red units are indistinguishable, i.e., all have the same type. A unit might correspond roughly to a fire team. (This estimator can be applied at multiple levels in a command hierarchy. At higher levels in a command hierarchy, the object designated as a unit in the system abstraction might correspond to some larger group of forces.) Each unit will have some associated strength. The strength of a unit will be an abstraction of the health and current effectiveness of the unit, and this will be described by a scalar. The units will move on a graph where the nodes on the graph correspond to positions in the conflict arena. There will be edges between the nodes, and a unit at one node can move only to the nodes which are connected to that node by an edge, i.e., to the adjacent nodes. With this abstraction, the game state is specified by the location/node of each unit and the associated unit's strength. Note that this can include both the Blue and the Red units. As we are considering only the reconnaissance component of the urban combat problem here, we are only estimating the strength and locations of the Red units.

We refer to the above abstract graph as the movement graph. There is

a second graph with the same nodes, but where an edge exists between two nodes if and only if there exists a line-of-sight (LOS) between the two nodes. This graph is referred to as the sight graph.

We consider the reconnaissance problem from the point of view of Blue. The true Red laydown is the actual positions and health status of all the Red units. Blue will have some initial estimate of this ground-truth, and will make observations. We can assume that the observations will be reports of Red forces and their strength at nodes on the graph. These observations will be corrupted, and may contain both false-positives and false-negatives. The observations will be corrupted not only by random noise, but also by adversarially induced errors. For example, one Red unit may try to conceal its existence, while another may "demonstrate" (that is, make its presence obvious, and even attempt to exaggerate the Red strength at that location). Consequently, noise in the observation process is not well-modeled as a purely stochastic process.

Blue will maintain the strength distribution as an ersatz information state. We let $\mathcal{L}$ denote the nodes on the graph, and let S_t denote the strength distribution (maintained by Blue) at time t, where $S_t : \mathcal{L} \to [0, N]$ where N is the total possible Red strength in the battlespace, and we denote the value of S_t at node $l \in \mathcal{L}$ as $[S_t]_l$. We require that $\sum_{l\in\mathcal{L}}[S_t]_l = N$. In other words, S_t is a non-one-sum distribution. We let $\mathcal{S}^N$ denote the space of strength distributions with total mass N, i.e.,

$$\mathcal{S}^N = \left\{ S : \mathcal{L} \to [0, N] \;\middle|\; S_l \in]0, N[\ \forall l \in \mathcal{L}, \ \sum_{l\in\mathcal{L}} S_l = N \right\}.$$

We assume a discrete-time model where $t \in \{0, 1, 2, \dots\}$. The dynamics of the system include movement of Red forces. Each unit can move only to the adjacent nodes in one time-step. Further, more than one unit may occupy a single node at any time. For the purpose of this reconnaissance problem study, we do not include attrition in the model. Blue is observing Red, but not engaging. The observation process is as broadly indicated above. Thus, as with other filters, at each time step, we will have a dynamics update of the strength and an observation update.

We employ the terms observer and estimator, rather than say filter, in this context to emphasize that we are not propagating probabilistic objects. We will demonstrate some properties of this observer such as convergence to the true state. However these statements will have a natural form that is slightly different from the standard statements regarding observers. Further, we will obtain a bound on the estimate errors in terms of the size of

the unknown disturbance inputs. Thus, the strength distribution acts as a robust estimator [1], [2], [12], [13], and an observer [4], [5].

3 Observation Update

At any time t, multiple observations may occur. We index these by $\rho \in]1, n_t^o[$ where n_t^o is the number of observations occurring at time t, and notation $]k, l[$ denotes the set of integers $\{k, k+1, \dots l\}$. Each observation, $Y_{t,\rho}$ takes the form of an ordered pair, (l, y), where $l \in \mathcal{L}$ and $y \in \{0, 1, \dots\}$ where this indicates that a Red strength of y was observed at node l.

As heuristic motivation, consider Bayesian conditional probability propagation. Suppose there is a single Red unit at some unknown node $l \in \mathcal{L}$. Suppose that prior to the observation, the probability that the unit is at node l is denoted by P_l. Suppose one observes the unit at node y. Let $P(y \,|\, \lambda)$ denote the probability that one observes the unit at y given that it is at node λ. Using Bayesian propagation, the a posteriori probability distribution is

$$\hat{P}_l = \frac{P(y|l)}{P(y|l)P_l + \sum_{\lambda \neq l} P(y|\lambda)P_\lambda} P_l,$$

and similarly, for $\lambda \neq l$,

$$\hat{P}_\lambda = \frac{P(y|\lambda)}{P(y|l)P_l + \sum_{\lambda \neq l} P(y|\lambda)P_\lambda} P_\lambda.$$

Suppose one observes $Y_{t,\rho} = (y, l)$. The form of the above Bayesian propagation can be seen as heuristic motivation for the form of the strength observation update. In particular, with a priori (pre-observation) strength S_t, the a posteriori (post-observation) strength, $\widehat{S}_t$, is given by the following, where a specific form for $\delta = \delta(y, [S_t]_l)$ wil be given below.

$$[\widehat{S}_t]_l = \frac{1+\delta}{1 + \frac{\delta [S_t]_l}{N}} [S_t]_l \tag{1}$$

$$[\widehat{S}_t]_\lambda = \frac{1}{1 + \frac{\delta [S_t]_l}{N}} [S_t]_\lambda \quad \forall\, \lambda \neq l. \tag{2}$$

Proposition 3.1 *Suppose* $\sum_{l \in \mathcal{L}} [S_t]_l = N$. *Then,* $\sum_{l \in \mathcal{L}} [\widehat{S}_t]_l = N$.

PROOF. Noting that $\sum_{\lambda \neq l}[S_t]_\lambda = N - [S_t]_l$, one has

$$\sum_{k\in\mathcal{L}}[\widehat{S}_t]_k = \sum_{\lambda\in\mathcal{L}\setminus\{l\}}[\widehat{S}_t]_\lambda + [\widehat{S}_t]_l = \frac{N-[S_t]_l}{1+\frac{\delta[S_t]_l}{N}} + \frac{1+\delta}{1+\frac{\delta[S_t]_l}{N}}[S_t]_l = N. \square$$

In other words, the observation update satisfies a *conservation of strength* property. With this proposition in hand, it is trivial to prove the following.

Corollary 3.2 *The observation update maps $\mathcal{S}^N$ into $\mathcal{S}^N$.*

One needs to choose δ so that the observer has certain desired behaviors. In particular, one would like the *consistency assumption* that repeatedly observing y at node l leads to an estimate sequence such that strength at node l converges to y. We henceforth choose $\delta = \delta(y, [S_t]_l)$ to be

$$\delta(y,s) = k\left[\frac{y-s}{s}\right] \tag{3}$$

where $k \in (0,1)$, and we will see that this choice of δ implies that the consistency assumption will be met.

With this choice of δ, the observation update takes the form

$$[\widehat{S}_t]_l = G(y,[S_t]_l) \tag{4}$$

$$[\widehat{S}_t]_\lambda = F(y,[S_t]_l)[S_t]_\lambda \tag{5}$$

where

$$G(y,s) = \frac{1+k\left(\frac{y-s}{s}\right)}{1+k\left(\frac{y-s}{N}\right)}s = \frac{s+ky-ks}{1+\frac{ky-ks}{N}} \tag{6}$$

$$F(y,s) = \frac{1}{1+k\left(\frac{y-s}{N}\right)}. \tag{7}$$

$$\tag{8}$$

Also, we define $\mathcal{G}(y,l,S)$ mapping $]0,N[\times\mathcal{L}\times\mathcal{S}^N$ into $\mathcal{S}^N$ by $[\mathcal{G}(y,l,S)]_l = G(y,S_l)$ and $[\mathcal{G}(y,l,S)]_\lambda = F(y,S_l)S_\lambda$ for $\lambda \neq l$.

We now show that the consistency assumption is satisfied with this observation update form if $k \in (0,1)$. This will follow by a variant of the Banach Fixed-Point Theorem. In particular, we show that repeated application of $G(y,\cdot)$ to any $s \in [0,N]$ converges to fixed-point $s = y$.

Theorem 3.3 *Let $S^0 \in \mathcal{S}^N$ with $S^0_l \neq N$, and fix any $y \in]0,N[$. For $i \in \mathcal{N} = \{1,2,\ldots,\infty\}$, let $S^i = \mathcal{G}(y,l,S^{i-1})$. Then $\lim_{i\to\infty}[S^i]_l = y$ and $y = G(y,y)$.*

The proof of Theorem 3.3 appears in Appendix A.

Although, in an urban combat setting, one would not get an observation with $y < 0$, one could imagine a case where $y > N$, that is where a UAV or reconnaissance team reports an observation exceeding the total known Red strength. We briefly address the behavior of the estimator under such a condition.

Theorem 3.4 *Let $S^0 \in \mathcal{S}^N$ with $S^0_l \neq N$, and suppose $y = N + m \in \{N + 1, N + 2, \ldots, \infty\}$. For $i \in \mathcal{N} = \{1, 2, \ldots, \infty\}$, let $S^i = \mathcal{G}(y, l, S^{i-1})$. Then $\lim_{i\to\infty}[S^i]_l = N$ (where $N = G(y, N)$). Further, for $\lambda \neq l$, $\lim_{i\to\infty}[S^i]_\lambda = 0$.*

PROOF. We do not include the proof, but it follows from the observation update definition. □

4 Dynamics Update: Strength Flow

We suppose that the dynamics of the system corresponds to movement of Red units. We must assume that this movement contains unobserved inputs due to each individual unit's own volition and due to Red commander orders. The modeling of such unknown inputs is a difficult task. Stochastic process inputs alone will not typically perform well, c.f., [7], [11] and the references therein. We consider models where the opponent may input any desired movement controls, subject to associated costs. We also consider stochastic motion as well, where this is well-modeled as a Markov chain.

First we consider the purely deterministic motion. In this case, we assume that there is a natural deterministic flow of Red units on the movement graph. Recall that the movement graph is specified by set of nodes, $\mathcal{L}$, and the edges connecting these nodes. The edges may be specified by an $L \times L$ symmetric matrix, $\mathcal{F}$ where $\mathcal{F}_{i,j}$ is one if the nodes i and j are adjacent (i.e., if the two nodes are connected by an edge), and $\mathcal{F}_{i,j}$ is zero otherwise. In the deterministic case, each row of the flow matrix, $\mathcal{F}$, will have a single entry of one with the remaining entries being zero. That is, for each $l \in \mathcal{L}$, there exists $k \in \mathcal{L}$ such that $\mathcal{F}_{l,k} = 1$ and $\mathcal{F}_{l,\lambda} = 0$ for all $\lambda \neq k$. This construction defines a movement model where Red units at node l nominally move to node k. Consequently, given strength distribution S_t at time t, the strength distribution at the next time-step is given by

$$S_{t+1} = \mathcal{F}^T S_t. \tag{9}$$

Continuing with the deterministic model, we suppose that the Red commander might input a control signal changing the movement from the nominal model. The Red movement control at time t will be represented by an $L \times L$ matrix, U_t, modifying the nominal movement model given by $\mathcal{F}$. The movement from node $l \in \mathcal{L}$ at time t will be modified by the elements of the U_t matrix in the l^{th} row. If no movement control occurs at node l at time t, then the l^{th} row of U_t will consist entirely of zeros. Suppose the commander wishes to change the movement from node l at time t. As we still require that $\mathcal{F} + U_t$ be a movement matrix, we set $[U_t]_{l,k} = -1$ where k is the index such that $\mathcal{F}_{l.k} = 1$. Letting the revised single-move destination node be κ, we set $[U_t]_{l,\kappa} = 1$. The remaining entries of the l^{th} row are zeros. With the commander control, the strength distribution dynamics are given by

$$S_{t+1} = [\mathcal{F} + U_t]^T S_t. \tag{10}$$

Proposition 4.1 *The deterministic flow propagation conserves strength, that is, with strength update given by (10) (or (9)),* $\sum_{l\in\mathcal{L}}[S_{t+1}]_l = \sum_{l\in\mathcal{L}}[S_t]_l$.

PROOF. Noting that $\mathcal{F}$ and $\mathcal{F} + U_t$ are stochastic matrices, one sees that the proof is identical to that for Markov chains. Specifically,

$$\begin{aligned}\sum_{l\in\mathcal{L}}[S_{t+1}]_l = \sum_{l\in\mathcal{L}}\sum_{\lambda\in\mathcal{L}}[\mathcal{F}+U_t]_{\lambda,l}\,[S_t]_\lambda &= \sum_{\lambda\in\mathcal{L}}\Big\{\sum_{l\in\mathcal{L}}[\mathcal{F}+U_t]_{\lambda,l}\Big\}[S_t]_\lambda \\ &= \sum_{\lambda\in\mathcal{L}}[S_t]_\lambda. \square\end{aligned}$$

Remark 4.2 Perhaps we should note here that there is a significant computational issue as L can often be on the order of 10^3 to 10^5. However, since we only have nontrivial (reasonably high) strength mass at a small set of locations at time t, say $\mathcal{L}_1^t$, one may split the flow propagation into several small movement transitions using only local nodes. The computational reduction issue will be addressed in more detail in Section 6.

We now turn to the case where we include a stochastic component in the movement propagation model. From a given node, movement to a number of nodes (including the current node) might be quite reasonable, and the movement decisions made by the units among such choices may be well-modeled as stochastic processes. This may be combined with control inputs from the commander (where the movements generated by these controls might also be well-modeled as stochastic processes).

Let us motivate our strength flow model in this stochastic case. Suppose the probability that a unit moves from node i to node j in one step is given by $\mathcal{P}_{i,j}$. In this case, $\mathcal{P}$ is the transition matrix for a Markov chain. Let $[p_t^k]_i$ be the probability that Red unit k is at node $i \in \mathcal{L}$ at time t. Then, one has the usual probability update given by

$$p_{t+1}^k = \mathcal{P}^T p_t^k \tag{11}$$

for each unit. Suppose that each unit has strength one. For each node $\lambda \in \mathcal{L}$ and each unit $k \in]1, N[$, define random variable Y_l^i to be one if unit i is at node λ, and zero otherwise. Then, letting $[S_t]_\lambda$ denote the expected strength at node $\lambda \in \mathcal{L}$ at time t, one has

$$[S_t]_\lambda \;\; = \mathbf{E}\left\{\sum_{k=1}^{N} 1 \cdot Y_\lambda^k\right\} = \sum_{k=1}^{N} 1 \cdot \mathbf{E}[Y_\lambda^k] = \sum_{k=1}^{N} [p_t^k]_\lambda. \tag{12}$$

Then,

$$[S_{t+1}]_\lambda \;\; = \sum_{k=1}^{N} [p_{t+1}^k]_\lambda$$

which by (11),

$$= \sum_{k=1}^{N} \sum_{l \in \mathcal{L}} \mathcal{P}_{l,\lambda} [p_t^k]_l = \sum_{l \in \mathcal{L}} \mathcal{P}_{l,\lambda} \left(\sum_{k=1}^{N} [p_t^k]_l \right)$$

which by (12),

$$= \sum_{l \in \mathcal{L}} \mathcal{P}_{l,\lambda} [S_t]_l. \tag{13}$$

Consequently, with strength being the expected number of units at each node, one obtains a strength update model of the form

$$S_{t+1} = \mathcal{P}^T S_t.$$

In keeping with the deterministic model, we take $\mathcal{F} \doteq \mathcal{P}$.

With this model, the nominal strength flow is described by stochastic matrix, $\mathcal{F}$, where more specifically, $\mathcal{F}$ satisfies

$$\mathcal{F}_{i,j} \in [0,1] \quad \forall\, i,j \in \mathcal{L}, \;\text{ and }\; \sum_{j \in \mathcal{L}} \mathcal{F}_{i,j} = 1 \quad \forall\, i \in \mathcal{L}. \tag{14}$$

We again allow commander input, described by matrix U_t. Note that the $\mathcal{F}$ and $\mathcal{F} + U_t$ matrices above were, in fact, stochastic matrices of a specific form. The constraint on control U_t is that

$$
\begin{aligned}
&[\mathcal{F}]_{i,j} + [U_t]_{i,j} \in [0,1] \quad \forall\, i,j \in \mathcal{L} \\
&\sum_{j\in\mathcal{L}} [\mathcal{F}]_{i,j} + [U_t]_{i,j} = 1 \quad \forall\, i \in \mathcal{L}.
\end{aligned} \tag{15}
$$

The same proof as used for Proposition 4.1 yields the following.

Proposition 4.3 *The stochastic flow propagation conserves strength, that is with strength update given by (10) (or (9)), with stochastic flow matrices $\mathcal{F}$ and U_t satisfying (14) and (15) $\sum_{l\in\mathcal{L}}[S_{t+1}]_l = \sum_{l\in\mathcal{L}}[S_t]_l$.*

5 Input to Output Error Bounds and Robustness

There are certain properties that one would like an estimator to have. One of these is a bound on the estimator errors as a function of the size of the unknown inputs in the dynamics and observation processes. Such bounds are often referred to as disturbance attenuation bounds in robust/H_∞ filtering (c.f., [1], [3], [13], [12]), and have the form of input-to-output bounds in the general nomenclature (loosely used) of stability approaches (c.f., [6], [14] among many others). Due to the irresolvable modeling issues with urban combat, specifically the nature of the disturbances as stochastic and/or adversarial, there are multiple forms that such bounds could take. We will bound the $\mathbf{L_1}$ norm of the estimate errors by a function of the $\mathbf{L_1}$ norm of the non-stochastic components of the dynamics and observation noise.

The observation update is more complex than the dynamics update, and the process of obtaining the relevant bounds is correspondingly more complex in the observation-update portion of the computations. We address the dynamics contribution first.

5.1 Dynamics Disturbance Effects

We considered both of the models for the strength dynamics, deterministic and stochastic. We let the true Red unit strength at time t be denoted by H_t. From Section 4, one sees that in the deterministic case, we have

$$
H_{t+1} = [\mathcal{F} + U_t]^T H_t, \tag{16}
$$

while in the stochastic case, we have

$$
\mathbf{E}[H_{t+1}] = [\mathcal{F} + U_t]^T \mathbf{E}[H_t]. \tag{17}
$$

The dynamic update of the strength distribution, in both the deterministic and stochastic cases, is

$$S_{t+1} = \mathcal{F}^T S_t. \tag{18}$$

In the following we use $|s|$ to denote the $\mathbf{L_1}$ norm of a vector, s. For an $L \times L$ matrix, M, we use $|M|$ to denote the induced norm on the matrix as a linear operator. That is, $|M| = \sup_{|s|\leq 1} |Ms|$.

Remark 5.1 It is well-known that the induced norm of the transpose of a stochastic matrix (i.e., a square matrix satisfying (14)) is one. To see this, note that

$$\begin{aligned} |M^T s| &= \sum_{\lambda=1}^{L} \left|[M^T s]_\lambda\right| = \sum_{\lambda=1}^{L} \left[\sum_{l=1}^{L} M_{l,\lambda}|s_l|\right] \\ &= \sum_{l=1}^{L} \left[\sum_{\lambda=1}^{L} M_{l,\lambda}\right] |s_l| = \sum_{l=1}^{L} |s_l| = |s|. \end{aligned}$$

Lemma 5.2 *In the stochastic case, we have*

$$|S_{t+1} - \mathbf{E}[H_{t+1}]| \leq |S_t - \mathbf{E}[H_t]| + |U_t||\mathbf{E}[H_t]| \leq |S_t - \mathbf{E}[H_t]| + N|U_t|.$$

In the deterministic case, we have

$$|S_{t+1} - H_{t+1}| \leq |S_t - H_t| + |U_t||H_t| \leq |S_t - H_t| + N|U_t|.$$

PROOF. The rightmost inequalities are obvious. The deterministic case is an easy reduction from the stochastic case. Consequently, we prove only the first inequality of the lemma statement. From (17) and (18),

$$\begin{aligned} |S_{t+1} - \mathbf{E}[H_{t+1}]| &= \left|\mathcal{F}^T S_t - \left(\mathcal{F}^T + U_t^T\right)\mathbf{E}[H_t]\right| \\ &\leq \left|\mathcal{F}^T (S_t - \mathbf{E}[H_t])\right| + \left|U_t^T \mathbf{E}[H_t]\right| \\ &\leq |S_t - \mathbf{E}[H_t]| + |U_t^T|\,|\mathbf{E}[H_t]|\,. \end{aligned}$$ □

Theorem 5.3 *Let the true strength dynamics update be given by (16) and (17) in the deterministic and stochastic cases, respectively. Let the strength estimator update be given by (18). Then, in the deterministic and stochastic cases, we have*

$$|S_t - H_t| \leq |S_0 - H_0| + N \sum_{r=0}^{t-1} |U_r^T|$$

and

$$|S_t - \mathbf{E}[H_t]| \ \leq |S_0 - \mathbf{E}[H_0]| + N \sum_{r=0}^{t-1} |U_r^T|$$

for all $t \geq 0$, respectively.

PROOF. The proof is immediate by Lemma 5.2 and induction. □

5.2 Observation Disturbance Effects

We now obtain bounds on the effects of observation errors. First, we consider the case where there is no observation error, and obtain a bound on the a posteriori strength estimator error given the a priori error.

Let S_t denote the strength distribution prior to the observation, and $\widehat{S}_t$ denote the strength distribution after the observation. The observation is (l, y), i.e., strength y observed at node l. We again let H_t denote the true strength at time t. We consider two cases separately, $[H_t]_l \geq [S_t]_l$ and $[H_t]_l < [S_t]_l$.

In order to reduce the notation, in this discussion, we shall drop the t subscript. More exactly, we will let S_l denote $[S_t]_l$, $\widehat{S}_l$ denote $[\widehat{S}_t]_l$, and H_l denote $[H_t]_l$. First note from (6) that

$$\widehat{S}_l = \frac{S_l + k(y - S_l)}{1 + k\frac{(y - S_l)}{N}}.$$

Solving for y, we obtain

$$\overline{y} \doteq \overline{y}(\widehat{S}_l, S_l) \doteq \frac{\widehat{S}_l - S + kS(1 - \widehat{S}_l/N)}{k(1 - \widehat{S}_l/N)}. \tag{19}$$

With a little work, one finds that

$$\frac{\partial \overline{y}}{\partial \widehat{S}_l} = \frac{N(N - S_l)}{k(N - \widehat{S}_l)^2} > 0,$$

and so we see that $\overline{y}$ is monotonically increasing as a function of $\widehat{S}_l$ Define $y^u = \overline{y}(H_l, S_l)$. The following lemma requires some technical analysis (not included here).

Lemma 5.4 *Suppose $[H_t]_l \geq [S_t]_l$. One has*

$$\left|H - \widehat{S}\right| \leq \begin{cases} |H - S| - \frac{H_l - S_l}{N - S_l}|H_l - S_l| + 2\left[G(y, S_l) - H_l\right] & \text{if } y > y^u, \\ |H - S| - (1 - F(y, S_l))|H_l - S_l| & \text{if } y^u \geq y \geq S_l, \\ |H - S| + 2k|y - H_l| & \text{if } y < S_l. \end{cases}$$

In particular, when $y = H_l$, $|H - \widehat{S}| \leq |H - S| - (1 - F(y, S_l))|H_l - S_l|$.

A similarly technical result holds in the case where $[H_t]_l < [S_t]_l$.

5.3 Combined Disturbance Effects

If one does not observe a reasonably dense set of locations, one cannot guarantee that the estimator will converge to the true Red force laydown. For instance, there could be a single location that is never observed, and where Red is maintaining a hidden unit or group of units. However, combining the results of Sections 5.1 and 5.2, one obtains a disturbance attenuation result. Further, under sufficiently strong assumptions, one can prove that the strength distribution converges to the true Red laydown.

However, it is worth noting that if one blindly accepts all observational data as correct, an estimator would not typically have such desirable behavior, and examples of poor behavior of such estimators are easily constructed.

6 Propagation of Strength Distribution: A Computationally Reasonable Scheme

We now present a scheme to propagate the strength distribution in real-time for the application domain of urban operations. Note that the true strength can only be distributed amongst at most L^R of the total L locations, where L^R is the maximum number of Red units. To model this constraint, we allow the strength distribution, S_t, (at any t) to only have significant strength components ($> \epsilon$) at a few locations (denoted by $\mathcal{L}_1^t$) and almost no strength component ($= \epsilon$) at most of the other locations (denoted by $\mathcal{L}_2^t$). Let $L_1 \doteq \sharp\mathcal{L}_1^t$ and $L_2 \doteq \sharp\mathcal{L}_2^t$, $\forall t$. The definition implies that we will keep the number of locations constant in each set over time. This is a reasonable assumption (since $L^R << L_1$) and obviously allows much better computational speed. It is worth noting that typically $L_1 > L_2$. Define

$$N_1 \doteq \sum_{j \in \mathcal{L}_1^t} [S_t]_j = N - N_2 \tag{20}$$

where

$$N_2 \doteq \sum_{i \in \mathcal{L}_2^t} [S_t]_i = \epsilon L_2 \tag{21}$$

We first establish some constraints, and then prove that under these constraints the convergence results given in section (3) hold for the scheme

outlined in this section. We will need the following assertion (whose proof we do not include due to space constraints).

Theorem 6.1 *For any time t and observation $Y_{t,\rho}$, such that the observation location $l \in \mathcal{L}_2^t$, and the strength observed $y \in]1, 2N[$, if $L_1 \geq \frac{3N}{k}$, then $\bar{s} \geq s^* \geq \hat{s}^*$ where $s^* \doteq \min_{j \in \mathcal{L}_1^t}[S_t]_j$ and $\hat{s}^* \doteq \min_{j \in \mathcal{L}_1^t}[\hat{S}_t]_j$, and $\bar{s} \doteq [\hat{S}_t]_l$.*

We need to continuously update $\mathcal{L}_1^t$ based on dynamic updates (observations and movement) starting from some initial estimates. We only outline the scheme for the update of $\mathcal{L}_1^t$ after the observation process. Note that at any time t, the observation location can belong to either $\mathcal{L}_1^t$ or $\mathcal{L}_2^t$. In the former case, one does not need to update $\mathcal{L}_1^t$ as the strength mass for any location in $\mathcal{L}_2^t$ will not exceed ϵ, if one chooses ϵ to be arbitrarily small. Consider the latter case; suppose one observes $Y_{t,\rho} = (y, l)$ such that $l \in \mathcal{L}_2^t$. The scheme is then summarized as follows. We compute the posteriori strength for the observation $Y_{t,\rho}$, using equations (4) and (5) respectively. In particular for $l \in \mathcal{L}_2^t$ we define

$$\bar{s} \doteq [\hat{S}_t]_l \doteq G(y, \epsilon) = \frac{N(\epsilon + ky - k\epsilon)}{N + ky - k\epsilon} \tag{22}$$

and for each $j \in \mathcal{L} \setminus l$ one has

$$[\hat{S}_t]_j \doteq F(y, \epsilon)[S_t]_j = \frac{N}{N + ky - k\epsilon}[S_t]_j$$

Define the minimum strength in the set $\mathcal{L}_1^t$ as $s^* = \min_{j \in \mathcal{L}_1^t} [\hat{S}_t]_j$ and the location, l^*, of this minimum strength be $l^* = \text{argmin}_{j \in \mathcal{L}_1^t} [\hat{S}_t]_j$ Recall that the Theorem 6.1 ensures that we always get, the observation y at any location $l \in \mathcal{L}_2^t$ to yield, $\bar{s} > s^*$. We update $\mathcal{L}_1^t$ and $\mathcal{L}_2^t$ as follows ,

$$\mathcal{L}_1^{t+1} = \begin{cases} \mathcal{L}_1^t & \text{if } l \in \mathcal{L}_1^t \\ \mathcal{L}_1^t \bigcup l \setminus \{l^*\} & \text{otherwise} \end{cases} \tag{23}$$

$$\mathcal{L}_2^{t+1} = \begin{cases} \mathcal{L}_2^t & \text{if } l \in \mathcal{L}_1^t \text{ ;} \\ \mathcal{L}_2^t \bigcup l^* \setminus \{l\} & \text{otherwise} \end{cases} \tag{24}$$

Finally we redistribute any strength lost after the update in (23) and (24). Let $[\bar{S}_t]_k$ denote the strength at location k after obtaining the sets $\mathcal{L}_1^{t+1}$ and $\mathcal{L}_2^{t+1}$. In particular, for $k \in \mathcal{L}_2^{t+1}$

$$[\bar{S}_{t+1}]_k = \epsilon \tag{25}$$

The readjustment given above ensures that the total strength stays conserved in the set $\mathcal{L}_2^{t+1}$ at N_2. Let's define $\bar{N}_1 \doteq \sum_{j\in\mathcal{L}_1^{t+1}}[\hat{S}_t]_j$ and $\delta N_1 \doteq N_1 - \bar{N}_1$, then for all $i \in \mathcal{L}_1^{t+1}$

$$[\bar{S}_{t+1}]_i = [\hat{S}_t]_i \left(1 + \frac{\delta N_1}{\bar{N}_1}\right) \tag{26}$$

Note that

$$\sum_{i\in\mathcal{L}_1^{t+1}} [\hat{S}_t]_i \frac{\bar{N}_1 + \delta N_1}{\bar{N}_1} = \bar{N}_1 \frac{N_1}{\bar{N}_1} = N_1. \tag{27}$$

Then using (25) and (27), we get the following corollary.

Corollary 6.2 *The observation update scheme given in (22-26) maps $\mathcal{S}^N$ into $\mathcal{S}^N$.*

Now we present the argument that given Theorem 6.1, our scheme outlined above retains the convergence properties given by Theorems 3.3 and 3.4. Note that the convergence is proved with the properties of $G(y, s)$ and its first and second derivatives. If we implement the scheme outlined above, the only potential argument is that with observations at multiple locations, certain node locations may move from $\mathcal{L}_1^t$ to $\mathcal{L}_2^{t+1}$, and certain others may move from $\mathcal{L}_2^t$ to $\mathcal{L}_2^{t+1}$. However with repeated observations, any location at which the observation is repeated will always end in the set $\mathcal{L}_1^{t+\delta t}$ for some δt, and stay there. Note that this is ensured by our assumption, $L_1 > L^R$, the fact that $y \subset \mathbf{Z}^+$ and Theorem 6.1 (even with all the L^R locations having repeated observations). We skip the proof details because of space constraints.

7 Application Example

We present an example application which motivates the advantages of the proposed estimator. Consider a graph with almost 10000 nodes as shown in Figure 1. The following color scheme is used for the terrain in the 2-D plot. The black dots are the open area nodes (including the building corners), the cyan colored dots are the boundary nodes of the rivers, the blue colored circles are the bridges, and the magenta colored dots are the interior points of the buildings (enclosed areas). The Red teams' true locations are shown in the figure by red colored $*$'s whereas the Blue teams true locations (completely known) are indicated by blue colored $*$'s. The strength

distribution estimator only knows partial information about the Red true locations; indicated by the red circle enclosing the red $*$ in the top plot of Figure 1. The red colored circles are used to depict the strength distribution estimation (centered at the location with the radius of the circle indicating the strength estimate at that location).

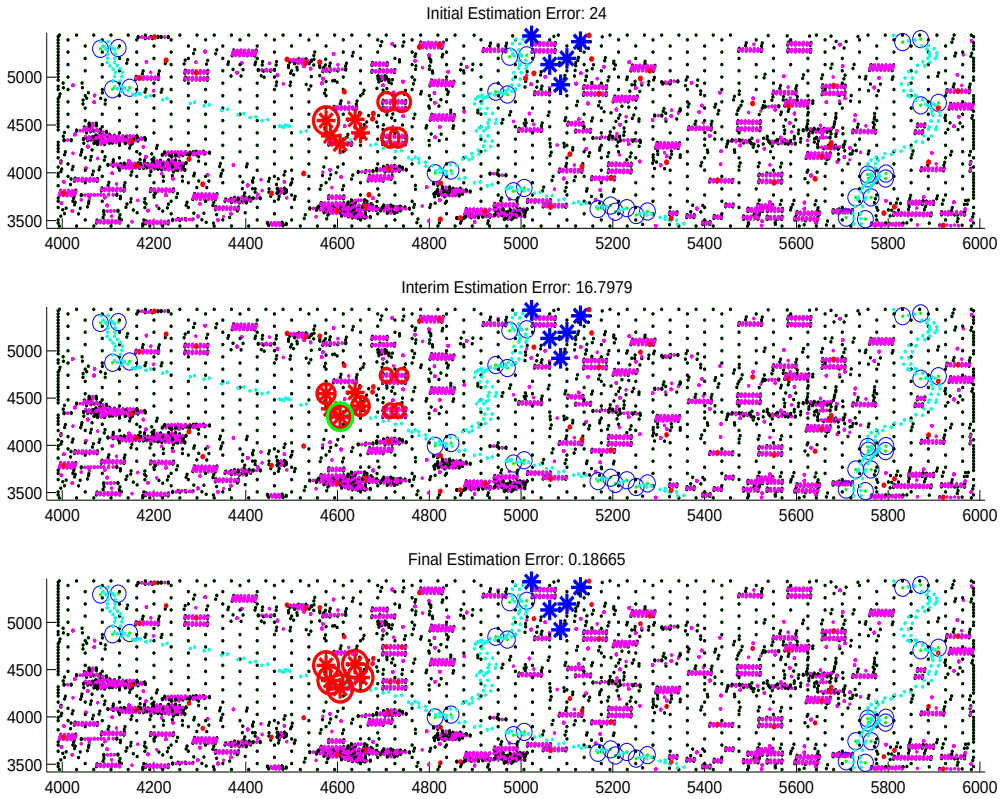

Figure 1: Observation Update Example

We assume that we have a priori knowledge of the maximum number of Red teams. Let $L^R = 5$, with each Red team having a scalar strength value of 3 in the beginning of the game scenario. Other parameter values used in this example are $k = 0.9$, $L_1 = 25$, and $L_2 = 9975$. Recall,

that $\mathcal{F}_{i,j}$ denotes the proportion of strength mass flowing from node i to j. The individual entries in this matrix must be based on terrain and some Red behavior modelling. To simplify the situational example, we allow for uniform diffusion movement model with $\mathcal{F}_{ii} = 0.7$, $\forall i$ and the rest of the strength mass being distributed through the adjacent (connected) edges on the movement graph.

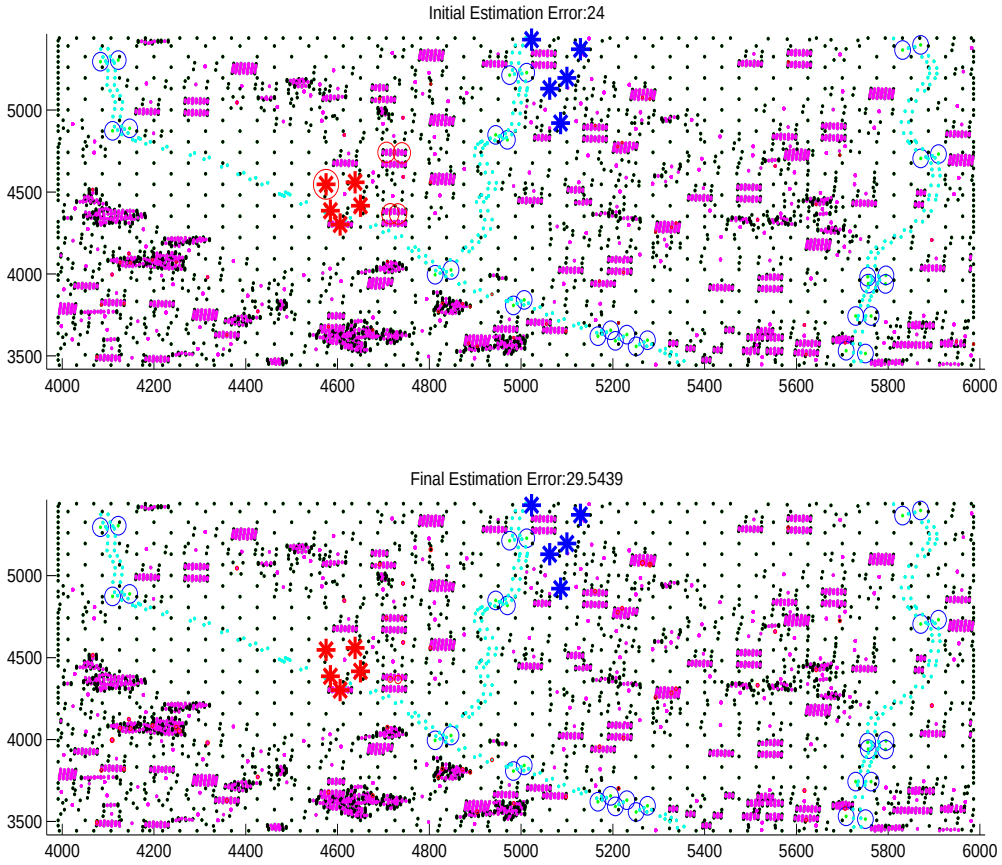

Figure 2: Flow Update /Dynamics Example

The initial strength distribution has only one Red team information and hence the rest of the strength estimation red circles are appearing currently at other locations (with minimum or no LOS from the current location of the Blue teams). We will assume that the observations can be based on LOS between the Blue and the Red team's current locations as well as through UAVs or other aerial sensors. At any time t, any sequence of observations is used to update the strength distribution observation propagation (shown as green colored circles at the current observed location in the second plot of Figure 1). As this process is repeated without the dynamics and one gets enough observations for each Red team, the estimation converges to the ground truth (as shown in the final plot of the Figure 1 by strength estimations, as red circles, enclosing the true strength, as red $*$'s). The three plots also show that the estimation error reduces with repeated observations. Note that the convergence for each observation can be handled with a different rate based on the sensor error or the confidence factor by choosing an appropriate value of k in equations (6) and (7).

Figure 2 shows the pure dynamics update from the same initial condition without any observation. We consider the Blue teams to stay at their initial locations throughout for simplicity. We also consider, as the worst case, the commander input for Red to "stay put". Another intuitive and easily implementable input could be movement of all the Red teams such that they all move to the (connected) adjacent node with least number of LOS connections. As expected, the dynamic update without any new sensor information leads to diffusion of the strength to the surrounding nodes (based on the movement graph). This is depicted in Figure 2 by tiny red dots (negligible strength) spread out across the plot instead of the clearly visible red circle in Figure 1. The strength distribution based estimation error increases but is bounded by the sum of the initial estimation error and the norm (or the size) of the commander's control input. Since the commander input is to "stay put", the entries in U matrix correspond to negative of the entries in the $\mathcal{F}$ matrix everywhere and an entry 1 on the diagonal. The upper bound in Theorem 5.3 is then $(24 + 5(1)(n-1))$, where $n = 100$ is the number of dynamic updates (leading to the upper bound of 519).

Finally we show an example of the combined case where we track the Red teams using both the observation and the dynamics from the above two examples in Figure 3. In the lower plot of Figure 3, one can clearly see that the diffusion of the strength is partially negated by the incoming observation data. The estimation error for this case falls between the first two cases; higher than the pure observation process and lower than the pure dynamical movement case (as expected).

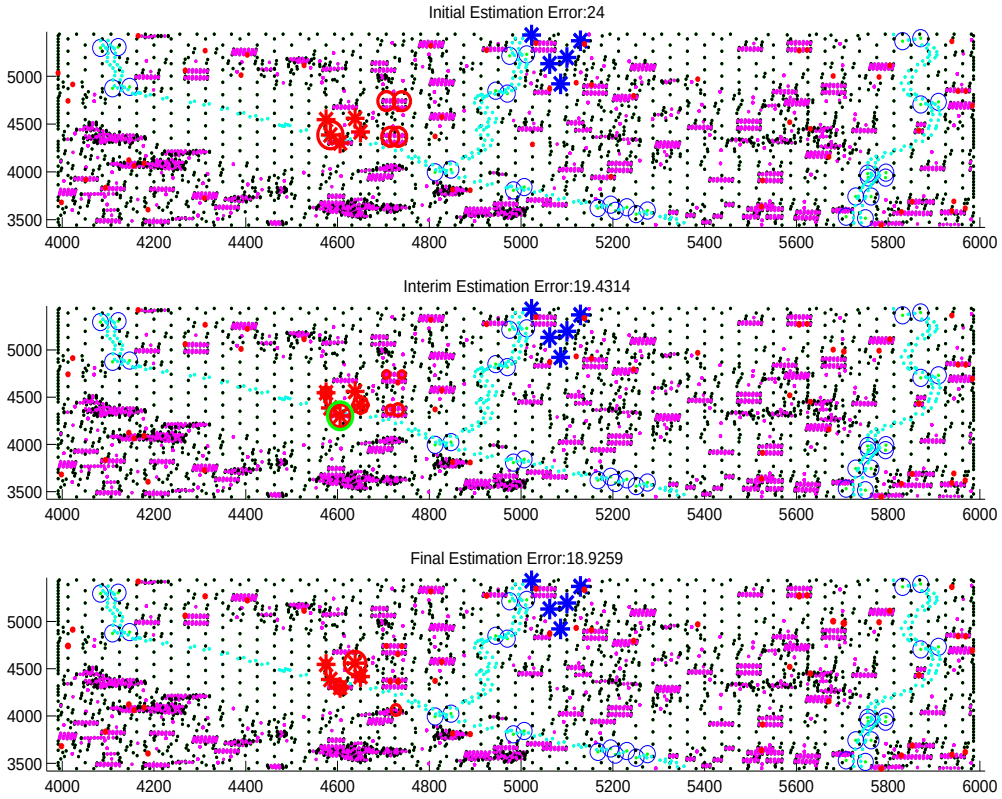

Figure 3: Observation and Dynamics, Combined Example

The time involved for the combined update using the proposed scheme in Section 6 is of the order of milli-seconds which is at least three to four orders better in magnitude than the existing schemes. The strength distribution is a reasonable output for providing the commander with an assessment of the hot-spots (or zones for Blue to watch for) in form of Red team strength distribution across the map. It is also a reasonable object to feed into a game engine that can assess the risk associated with a specific Red strength distribution along with other key critical components of the value function (c.f., [7, 9, 10]).

References

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

[2] W.H. Fleming and W.M. McEneaney, "Robust limits of risk sensitive nonlinear filters", Math. Control, Signals and Systems, 14 (2001), 109–142.

[3] 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.

[4] A.J. Krener and M.Q. Xiao, "Observers for linearly unobservable nonlinear systems", Systems and Control Letters, 46 (2002) 281–288.

[5] A.J. Krener, "A Lyapunov theory of nonlinear observers", *Stochastic Analysis, Control, Optimization and Applications* (Eds. W.M. McEneaney, G.G. Yin and Q. Zhang), Birkhauser (1999) 409–420.

[6] M. Krstic, I. Kanellakopoulos and P.V. Kokotovic, *Nonlinear and Adaptive Control Design,* Wiley, New York, 1995.

[7] W.M. McEneaney and R. Singh, "Robustness against deception", *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind,* Chapman and Hall/CRC, New York (2007), 167–208.

[8] W.M. McEneaney and R. Singh, "Deception in Autonomous Vehicle Decision Making in an Adversarial Environment", Proc. AIAA Conf. on Guidance Navigation and Control, (2005).

[9] W.M. McEneaney and R. Singh, "Unmanned Vehicle Operations under Imperfect Information in an Adversarial Environment ", Proc. AIAA Conf. on Guidance Navigation and Control, (2004).

[10] W.M. McEneaney, "Some Classes of Imperfect Information Finite State-Space Stochastic Games with Finite-Dimensional Solutions", Applied Math. and Optim., 50 (2004), 87–118.

[11] W.M. McEneaney, B.G. Fitzpatrick and I.G. Lauko, "Stochastic Game Approach to Air Operations", IEEE Trans. on Aerospace and Electronic Systems, 40 (2004), 1191–1216.

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

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

[14] E.D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," SIAM J. Control and Optim., 21 (1983), 462–471.

Appendix A: Proof of Theorem 3.3

The proof of the convergence of the S^i_λ for $\lambda \neq l$ follows easily from (7) and the convergence of S^i_l, and so we only prove the claims regarding S^i_l.

Let $s^0 \in [0, N]$, and consider iteration $s^{i+1} = G(y, s^i)$. There will be two cases to consider, $s^0 < y$ and $s^0 > y$. (The case $y = s^0$ is immediate as $G(y, y) = y$.)

In the first case, one can directly apply the Banach Fixed-Point Theorem, and we now do so. In order to apply the Banach Fixed-Point Theorem, we need only prove that $G(y, \cdot)$ is a contraction on $[0, y]$, and that $G(y, y) = y$. The latter merely proves that the unique fixed-point must be $s = y$, and is easily seen by inspection of (6). To prove the first item, we need to show that $G(y, [0, y]) \subseteq [0, y]$ and that $|\frac{\partial G}{\partial s}(y, s)| < 1$ for all $0 \leq s \leq y$. First note that, by the right-hand side of (6),

$$G(y, 0) = ky/[1 + ky/N] > 0. \tag{28}$$

Then, again by the right-hand side of (6),

$$G(y, s) = \frac{(1-k)s + ky}{1 + k\left(\frac{y-s}{N}\right)} \leq \frac{1}{1 + k\left(\frac{y-s}{N}\right)} \leq y. \tag{29}$$

Now, multiplying the numerator and denominator of the right-hand side of (6) by N, and differentiating yields

$$\frac{\partial G}{\partial s}(y, s) = \frac{(1-k)N[N + k(y-s)] - [(s(1-k) + ky)N](-k)}{(N + ky - ks)^2},$$

which upon rearrangement yields

$$\frac{\partial G}{\partial s}(y, s) = \frac{N[(1-k)N + ky]}{[N + k(y-s)]^2}. \tag{30}$$

$$< \frac{N + k(y-N)}{N + k(y-s)} < 1. \tag{31}$$

Combining (28), (29) and (31) implies $G(y,s) \in [0,y]$. Note from (30) and the fact that $y \leq N$,

$$\frac{\partial G}{\partial s}(y,s) > 0. \tag{32}$$

Combining this with (31) yields $\frac{\partial G}{\partial s}(y,s) \in (0,1)$. Consequently, the above noted conditions for application of the Banach Fixed-Point Theorem apply, and one finds that $s^i \to \bar{s} \in [0,y]$ where $\bar{s}$ is the unique fixed point of $G(y,\cdot)$. However, as noted above $G(y,y) = y$, and therefore $\bar{s} = y$.

We now turn to the other case, $s^0 > y$. First note, that $G(y,y) = y$ and $G(y,N) = N$ where the latter follows by inspection of (6). Combining these with the monotonicity implied by (32), one finds that for $s \in [y,N]$, $G(y,s) \in [y,N]$. Consequently, $s^i \in [y,N]$ for all i. Now, differentiating (30) a second time yields

$$\frac{\partial^2 G}{\partial s^2}(y,s) = \frac{2kN[(1-k)N+y]}{[N+k(y-s)]^3} > \frac{2kN[(1-k)N+y]}{[(1-k)N]^3} > 0$$

on $[y,N]$. Consequently, $G(y,s) < s$, on (y,N). This implies that s^i is strictly monotonically decreasing given any $s^0 \in (y,N)$, and as shown just above, bounded below by y. Consequently, there is some $\bar{s} \in [y,N)$ such that $s^i \downarrow \bar{s}$, and $\bar{s} = G(y,\bar{s})$. However, this last equality is impossible unless $\bar{s} = y$, and so $s^i \downarrow y$. □