Optimal State Estimation under Observation Budget Constraints

Praveen Bommannavar
Management Science and Engineering
Stanford University
bommanna@stanford.edu

Nicholas Bambos
Electrical Engineering and
Management Science and Engineering
Stanford University
bambos@stanford.edu

***Abstract*—In this paper, we consider the problem of monitoring an intruder in a setting where the number of opportunities to conduct surveillance is budgeted. Specifically, we study a problem in which we model the state of an intruder in our system with a Markov chain of finite state space. These problems are considered in a setting in which we have a hard limit on the number of times we may view the state. We consider the Markov chain together with an associated metric that measures the distance between any two states. We develop a policy to optimally (with respect to the specified metric) keep track of the state of the chain at each time step over a finite horizon when we may only observe the chain a limited number of times. The tradeoff captured is the budget for surveillance versus having a more accurate estimate of the state; the decision at each time step is whether or not to use an opportunity to observe the process. We also examine a scenario in which there is a budget constraint as described as well as a cost on observation. Finally, theoretical properties of the solution are presented. Hence, we present the problem of monitoring the state of an intruder using a Markov chain approach and present an optimal policy for estimating the intruder's state.**

Keywords-monitoring; surveillance; budget; resource allocation; dynamic programming; convexity; optimal estimation.

I. Introduction

The importance of monitoring technologies in today's world can hardly be overstated. Indeed, there are volumes dedicated to this field [2] [3]. In recent years, the need for effective security measures has become especially evident. Indeed, at present, Microsoft announces almost one hundred new vulnerabilities *each week* [4]. Perhaps more alarming is the fact that government agencies routinely must manage defenses for network security and are hardly equipped to do so. This is evidenced by the fact that 10 agencies accounting for 98% of the Federal budget have been attacked with as high of a success rate as 64% [5].

This paper is concerned with a mathematical treatment of these important problems, as initially proposed in [1]. Specifically, we consider a scenario in which we model the activities of an intruder as a state in a Markov chain. We develop the problem of monitoring the state in a finite-horizon discrete-time setting where we are only able to make observations a limited number of times. Such a budget arises naturally in wireless settings, for example, where power is at a premium. We present an algorithm for deciding when to use opportunities to view the process in order to minimize the surveillance error. This error is accrued at each time step according to a metric indicating how far from the true state the estimate was. We also consider problems in which additional cost is accrued for each observation that is made. In this way a hard constraint as well as a soft constraint are considered together.

Section II describes some state-of-the-art research in this field as well as our contribution to it. In Section III, we begin by introducing the monitoring problem mathematically. We continue with a derivation of the optimal policy using dynamic programming and then present the implementation of the optimal policy. Section IV gives an adjusted policy in an extended scenario where observations accrue cost in addition to being budgeted. Section V contains a brief note about dealing with large state spaces and in Section VI, we demonstrate performance using numerical results and examine theoretical properties of the solution structure. Finally, in Section VII, we conclude the paper and offer directions for future work in this vein.

II. State-of-the-Art

A growing literature addresses security from a mathematical perspective, with a range of theoretical tools being employed for managing threats. In [6], a network dynamically allocates defenses to make the system secure in the appropriate areas as time progresses. Parallels between the security problem and queuing theory are drawn upon, where vulnerabilities are treated as jobs in a backlog. The model of [7] uses ideas from game theory for intrusion detection where an attacker and the network administrator are playing a non-cooperative game. A related problem is addressed in [8] as well.

More generally, theoretical work in signal estimation has also been greatly developed [9]. Related works have considered aspects of decision making with limitations on the available information. In [10], an estimation problem is considered in which the received signal may or may not contain information. Similar issues are studied in [11] and [12] but in a control theoretic context in which the actuator has a non-zero probability of dropping estimation and control packets.

Approaches in the sensor network literature also attempt to mitigate power usage while tracking an object, as in [13] where ‘smart sleeping policies’ are considered. Algorithms for GPS as studied by the mobile device community also draw on techniques to minimize estimation error in the presence of noise and power limitations [14]. The approach presented here focuses on a hard constraint on energy usage, while [15] approaches a related problem with a constraint on the *expected* energy usage.

The unique aspect of our formulation is the nature of the power limitation. This non-standard constraint was introduced in [16] and developed in other works such as [17]. All of these problems consider finite horizon frameworks in which decisions are usage limited and hence the ability to make actions is a resource to be appropriately allocated.

In this paper, we aim to describe a model for intrusion detection with a notion of a power budget for observations, and continue by seeking an optimal policy for this problem formulation and proving some properties about the solution.

III. Monitoring

Let us now examine the monitoring/surveillance problem in greater detail. In what follows, we shall consider the states of a Markov chain as an abstraction for the position of an intruder in our system. Such a model is able to capture several scenarios. In one, we may wish to spatially monitor the location of an adversary using equipment that has usage constraints. Another situation is that we can consider the state of the intruder to be a location in a data network. Although many interpretations are possible, our goal is to be able to track this state with as little error as possible. We begin by presenting the model in a mathematical state estimation framework, and then present the solution structure.

A. Model

Consider a Markov chain $\mathcal{M}$ with finite state space S, transition matrix P and associated measure $d : S \times S \to \mathbf{R}$ as in Figure 1. The metric gives a sense of how close states are so that we can measure the effectiveness of an estimate of the true state. We assume that the process is known to start at initial state x_0 and we are interested in having an accurate estimate of the process over a finite horizon $k = 1, ..., N-1$. The decision space is simply $u \in \{0, 1\}$ where 0 corresponds to no observation being made and 1 corresponds to an observation being made. When an observation is made, the state x_k of $\mathcal{M}$ is perfectly known. Without an observation, on the other hand, we must form an estimate $\hat{x}_k$ for the state given all observed information thus far. The number of times observations may be made is limited to $M < N$.

The cost of making estimate $\hat{x}_k$ at time k when the true state is actually x_k is $d(x_k, \hat{x}_k)$. If d is a metric, we have the important properties

1. $d(x, y) \geq 0 \quad \forall x, y \in S$
2. $d(x, x) = 0 \quad \forall x \in S$
3. $d(x, y) = d(y, x) \quad \forall x, y \in S$
4. $d(x, z) \leq d(x, y) + d(x, z) \quad \forall x, y, z \in S$

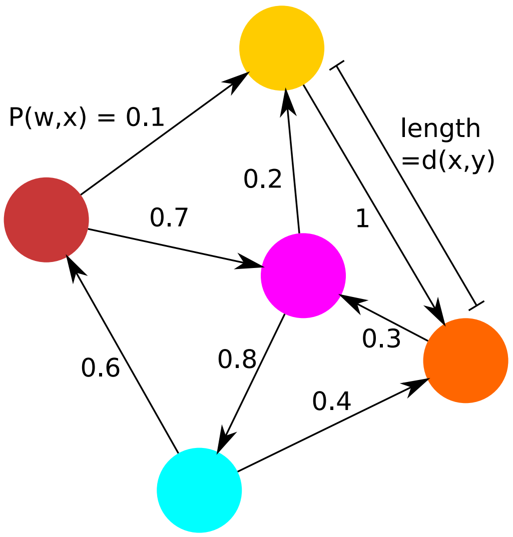

Figure 1. Markov chain with transitions $P(\cdot, \cdot)$ and measure $d(\cdot, \cdot)$. Self loops are captured by outgoing edge probabilities summing to less than one.

At each time k, the state of our system can be represented by $\{(r, s, t); x_{N-t-r}; x_{N-t}\}$ where r is the number of time slots that have passed since the last observation, s is the number of opportunities remaining to make an observation, t is the number of time slots remaining in the problem, x_{N-t-r} is the last observed state of $\mathcal{M}$ and x_{N-t} is the current state. We seek a policy $\pi = \{\mu_k\}_{k=1}^{N-1}$ such that the actions $u_k = \mu_k((r, s, t), x_{N-t-r}) \in \{0, 1\}$ are chosen to minimize the cumulative estimation error. The policy π is admissible if it abides by the additional constraint that the number of times observations are made is no greater than M. Denote the class of admissible policies by Π.

We want to find a policy $\pi^* \in \Pi$ to minimize

$$\mathbf{E}\left\{\sum_{k=1}^{N-1} d(x_k, \hat{x}_k)\right\}$$

It should be noted that the estimate $\hat{x}_k$ depends on the action u_k because if $u_k = 1$ then $\hat{x}_k = x_k$ and there is no estimation error, while if $u_k = 0$ then we must make the best guess of the state that is possible with the known information.

Deciding on the distance metric is an issue of modeling and may be specific to the application at hand. We consider a few alternatives here:

1) Probability of Error: To recover a cost structure that results in the same penalty regardless of which state is chosen in error (probability of error criterion), we simply set the distance metric as

$$d(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases}$$

Such a choice maximizes the likelihood of estimating the correct state.

2) Euclidean distance: We may suppose that states correspond to physical locations - in this case, we may choose to let the distance $d(\cdot,\cdot)$ correspond to the Euclidean distance between states so that best estimates minimize the error as measured spatially.

Several other choices could also be made for a distance metric, such as the well known Metropolis distance or Chebyshev distance. In this paper, we are most interested in keeping track of an intruder, so we shall concern ourselves primarily with the probability of error and Euclidean distances.

B. Dynamic Programming

We use a dynamic programming approach to obtain an optimal policy [18]. Before presenting our algorithm for determining π^*, however, we first develop some important notation. In order to proceed, we must begin by determining several quantities offline. Let $\mathbf{d}(w)$ be the vector of distances of each state from w. Then we proceed by cataloging the quantities

$$\begin{aligned} w_r^*(x) &= arg\min_{w \in S} \left\{ \sum_{y \in S} \mathbf{P}[x_r = y | x_0 = x] d(y,w) \right\} \\ &= arg\min_{w \in S} \{(P^r \mathbf{d}(w))(x)\} \\ e_r^*(x) &= (P^r \mathbf{d}(w_r^*(x)))(x) \end{aligned}$$

for $r = 1, ..., N$. The values $w_r^*(x)$ and $e_r^*(x)$ correspond to the optimal estimate and estimation error, respectively, when we must determine the current state given that r time steps ago we observed that the state was x. There may, in some cases, be an efficient way to determine these quantities, but in general we must do this by simply cataloging these quantities offline through brute force. This may be done with relative ease if the state space is of tractable size or if the specific application displays certain sparsity (if our intruder is moving at a bounded rate then we may narrow down his location to a sparse set of states).

Now we proceed to construct the solution using backwards induction. We begin with $t = 1$, which corresponds to one unit of time remaining in the problem, and then continue for $t = 2, 3, ...$ until we are able to determine a recursion. As we build backwards in time (and forward in t), we let s vary and keep track of the cost $J_{r,s,t}(x)$ where x is a state of the Markov chain. This is depicted graphically in Figure 2, where the index r has been omitted. A given state (s,t) can transition to $(s-1, t-1)$ or $(s, t-1)$. The transition represents whether an observation was made or not - if so, then s is decremented, otherwise it remains the same. In the special case of $s = t$, the only sensible policy is to always use an observation, and in the case of $s = 0$, the only admissible policy is not to make an observation. This is also shown in Figure 2.

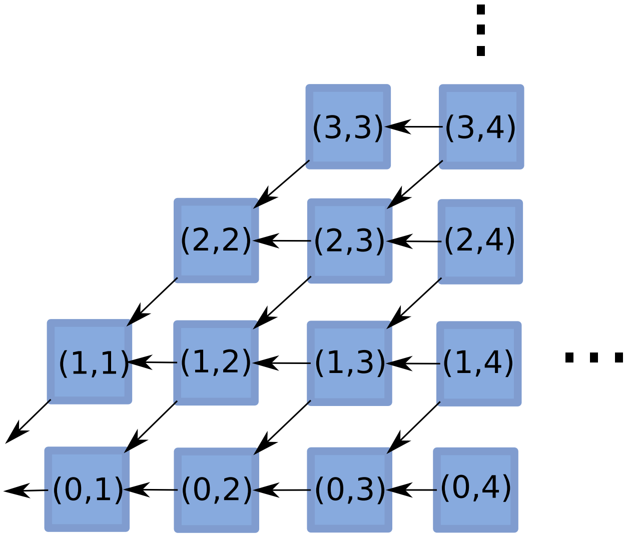

Figure 2. Admissible transitions for backwards induction. The pair (s,t) represents the number of observations and remaining time steps, respectively.

For $t = 1$, we can either have $s = 0$ or $s = 1$. These costs, respectively, are (in vector form)

$$\begin{aligned} J_{(r,0,1)} &= e_r^* \\ J_{(r,1,1)} &= 0 \end{aligned}$$

since not having an observation means we need to make a best estimate, and having an observation leads to zero cost.

Moving on to $t = 2$, the values of s can range from $s = 0$, $s = 1$ or $s = 2$. For $s = 0$ we have

$$J_{(r,0,2)} = e_r^* + e_{r+1}^*$$

since we would need to make an optimal estimate with no further information for the next two time slots. If $s = 1$, there are two choices: use an opportunity to make an observation so that $u = 1$, or do not observe, in which case $u = 0$. These choices can be denoted with superscripts above the cost function for each stage:

$$\begin{aligned} J_{(r,1,2)}^{(0)}(x) &= e_r^*(x) + J_{(r+1,1,1)}(x) = e_r^*(x) \\ J_{(r,1,2)}^{(1)}(x) &= 0 + \sum_{y \in S} P[x_{N-2} = y | x_{N-2-r} = x] e_1^*(y) \end{aligned}$$

For $u = 0$, we accrue error for the current time slot and no error afterwards. When an observation is made, no error is accrued for the current time slot $N-2$, but there is error in the next time slot, which depends on the current observation. In vector form, we may write

$$J^{(0)}_{(r,1,2)} = e^*_r + J_{(r+1,1,1)} = e^*_r$$
$$J^{(1)}_{(r,1,2)} = P^r e^*_1$$

We now introduce some new notation:

$$\begin{aligned}\Delta_{(r,1,2)} &= J^{(0)}_{(r,1,2)} - J^{(1)}_{(r,1,2)} \\ &= e^*_r - P^r e^*_1\end{aligned}$$

so that if $\Delta_{(r,1,2)}(x) \leq 0$, then we should not make an observation, whereas we should make an observation if $\Delta_{(r,1,2)}(x) > 0$. We proceed now by defining sets $\tau_{(r,1,2)}$ and $\tau^c_{(r,1,2)}$ such that

$$x \in \tau^c_{(r,1,2)} \Leftrightarrow \Delta_{(r,1,2)}(x) \leq 0$$
$$x \in \tau_{(r,1,2)} \Leftrightarrow \Delta_{(r,1,2)}(x) > 0$$

and we also define an associated vector $\mathbf{1}_{(r,1,2)} \in \{0,1\}^S$

$$\mathbf{1}_{(r,1,2)}(x) = \begin{cases} 1 & \text{if } x \in \tau^c_{(r,1,2)} \\ 0 & \text{otherwise} \end{cases}$$

Moving on to $s=2$, we have $J_{(r,2,2)} = 0$, since there are as many opportunities to observe the process as there are remaining time slots. We continue with $t=3$:

$$J_{(r,0,3)} = e^*_r + e^*_{r+1} + e^*_{r+2}$$

since there are three time slots to make estimates for with no new information arriving. For $s=1$, we again have a choice of $u=0$ and $u=1$. For $u=0$, we accrue a cost for the current stage, and then count the future cost depending on the current state:

$$\begin{aligned}J^{(0)}_{(r,1,3)}(x) = e^*_r(x) &+ \mathbf{1}_{(r+1,1,2)}(x) J^{(0)}_{(r+1,1,2)}(x) \\ &+ (1 - \mathbf{1}_{(r+1,1,2)}(x)) J^{(1)}_{(r+1,1,2)}(x)\end{aligned}$$

and combining terms gives us

$$\begin{aligned}J^{(0)}_{(r,1,3)}(x) = e^*_r(x) &+ J^{(1)}_{(r+1,1,2)}(x) \\ &+ \mathbf{1}_{(r+1,1,2)}(x)\Delta_{(r+1,1,2)}(x)\end{aligned}$$

which after substituting the value of $J^{(1)}_{(r+1,1,2)}(x)$ and putting things in vector form gives us:

$$J^{(0)}_{(r,1,3)} = e^*_r + P^{r+1}e^*_1 + diag(\mathbf{1}_{(r+1,1,2)})\Delta_{(r+1,1,2)}$$

Now we consider the $u=1$ case:

$$\begin{aligned}J^{(1)}_{(r,1,3)}(x) &= 0 + \sum_{y \in S} P[x_{N-3} = y | x_{N-3-r} = x] J_{(1,0,2)}(y) \\ &= \sum_{y \in S} P[x_{N-3} = y | x_{N-3-r} = x](e^*_1(y) + e^*_2(y))\end{aligned}$$

which can be put in vector form:

$$J^{(1)}_{(r,1,3)} = P^r(e^*_1 + e^*_2)$$

We now write the expression for $\Delta_{(r,1,3)} = J^{(0)}_{(r,1,3)} - J^{(1)}_{(r,1,3)}$:

$$\begin{aligned}\Delta_{(r,1,3)} = e^*_r + P^{r+1}e^*_1 &+ diag(\mathbf{1}_{(r+1,1,2)})\Delta_{(r+1,1,2)} \\ &- P^r(e^*_1 + e^*_2)\end{aligned}$$

Continuing with $s=2$,

$$J^{(0)}_{(r,2,3)}(x) = e^*_r(x) + 0$$

whereas for $u=1$,

$$\begin{aligned}J^{(1)}{}_{(r,2,3)}(x) = 0 + &\sum_{y \in S} P[x_{N-3} = y | x_{N-3-r} = x] \\ &\Big(\mathbf{1}_{(1,1,2)}(y) J^{(0)}_{(1,1,2)}(y) + (1 - \mathbf{1}_{(1,1,2)}(y)) J^{(1)}_{(1,1,2)}(y)\Big)\end{aligned}$$

where we have accounted for the cost stage by stage: in the current stage, no error is accrued since an observation is made but future costs depend on the observation that is made. That is, future costs depend on whether the current state x_{N-3} is observed to be in the set $\tau_{(1,1,2)}$. Averaging over these, we obtain the expression above. Combining like terms as above, we arrive at:

$$\begin{aligned}J^{(1)}_{(r,2,3)}(x) = 0 + &\sum_{y \in S} P[x_{N-3} = y | x_{N-3-r} = x] \\ &\Big(J^{(1)}_{(1,1,2)}(y) + \mathbf{1}_{(1,1,2)}(y)\Delta_{(1,1,2)}(y)\Big)\end{aligned}$$

Substituting the expression for $J^{(1)}_{(1,1,2)}(y)$, we get

$$\begin{aligned}J^{(1)}_{(r,2,3)}(x) = &\sum_{y \in S} P[x_{N-3} = y | x_{N-3-r} = x] \\ &\left(\sum_{z \in S} P[x_{N-2} = z | x_{N-3} = y] e^*_1(z) \right. \\ &\left. + \mathbf{1}_{(1,1,2)}(y)\Delta_{(1,1,2)}(y)\right)\end{aligned}$$

We simplify the expression by bringing the first summation in the parentheses. Then we apply the Kolmogorov-Chapman equation to get

$$\begin{aligned}J^{(1)}{}_{(r,2,3)}(x) = &\sum_{z \in S} P[x_{N-2} = z | x_{N-3-r} = x] e^*_1(z) \\ + &\sum_{y \in \tau^c_{(1,1,2)}} P[x_{N-3} = y | x_{N-3-r} = x]\Delta_{(1,1,2)}(y)\end{aligned}$$

Putting this into vector form, we have the expression:

$$J^{(1)}{}_{(r,2,3)} = P^{r+1}e^*_1 + P^r diag(\mathbf{1}_{(1,1,2)})\Delta_{(1,1,2)}$$

We use these expressions to get $\Delta_{(r,2,3)}$.

$$\Delta_{(r,2,3)} = e^*_r - P^{r+1}e^*_1 - P^r diag(\mathbf{1}_{(1,1,2)})\Delta_{(1,1,2)}$$

Finally, letting $s=3$, we get

$$J_{(r,3,3)}(x) = 0$$

This process can be continued for $t = 4, 5, \ldots$. For each stage (r,s,t), we may determine $J^{(0)}_{(r,s,t)}$ and $J^{(1)}_{(r,s,t)}$. These costs then allow us to determine when we should make an observation in the process and when we should not. The implementation of this policy is detailed in the following subsection.

C. Solution

We now present a method for constructing an optimal policy. We do this by storing for each (r,s,t) a subset of S, denoted by $\tau^c_{(r,s,t)}$, which is the set of last observed states for which we do not use an opportunity to view the process when we are at stage (r,s,t). That is, if the last observed state x was seen r time slots ago, it is in the set $\tau^c_{(r,s,t)}$, there are s opportunities remaining to make observations and there are t time slots remaining in the horizon then we should not make an observation at this time and simply make an estimate $w^*_r(x)$. On the other hand, if $x \in \tau_{(r,s,t)}$ then we should make an observation at stage (r,s,t) and accrue zero cost for that stage.

More precisely, an optimal policy π^* is given by

$$u_{(r,s,t)}(x) = \begin{cases} 0 & \text{if } x \in \tau^c_{(r,s,t)} \\ 1 & \text{otherwise} \end{cases}$$

Let us introduce three vector valued functions: $F_{(r,s,t)}, \Delta_{(r,s,t)} \in \mathbf{R}^S$ and $\mathbf{1}_{(r,s,t)} \in \{0,1\}^S$. We fill in values for these functions by using the following recursions:

$$\begin{aligned} F_{(r,s,t)} &= F_{(r+1,s-1,t-1)} \\ &\quad + P^r diag(\mathbf{1}_{(1,s-1,t-1)})\Delta_{(1,s-1,t-1)} \\ \Delta_{(r,s,t)} &= e^*_r + F_{(r+1,s,t-1)} - F_{(r,s,t)} \\ &\quad + diag(\mathbf{1}_{(r+1,s,t-1)})\Delta_{(r+1,s,t-1)} \\ \mathbf{1}_{(r,s,t)}(x) &= \begin{cases} 0 & \text{if } \Delta_{(r,s,t)}(x) > 0 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

for $1 < s < t < N$ and $1 \leq r \leq N - t + 1$. We also have the boundary conditions

$$F_{(r,t,t)} = 0, \quad F_{(r,1,t)} = P^r \sum_{j=1}^{t-1} e^*_j, \quad \Delta_{(r,t,t)} = e^*_r$$

These recursions allow us to determine the sets $\tau^c_{(r,s,t)}$ for s,t,r in the bounds specified, which in turn defines our optimal policy. Specifically, we assign

$$x \in \tau^c_{(r,s,t)} \Leftrightarrow \Delta_{(r,s,t)}(x) \leq 0$$

We conclude by giving expressions for the cost-to-go from any particular state when a particular action $u \in \{0,1\}$ is taken. The superscripts denote whether or not an observation will be made in the current stage.

$$\begin{aligned} J^{(0)}_{(r,s,t)} &= e^*_r + F_{(r+1,s,t-1)} + diag(\mathbf{1}_{(r+1,s,t-1)})\Delta_{(r+1,s,t-1)} \\ J^{(1)}_{(r,s,t)} &= F_{(r,s,t)} \end{aligned}$$

Observe that $\Delta_{(r,s,t)}$ is the difference between these two quantities. Hence, $\Delta_{(r,s,t)}$ functions as a method of determining whether or not to make an observation in the current time step.

We note that although the curse of dimensionality can make the operations required for the solution to be intractable for large scale problems, the structure of specific problems may allow us to generate good approximations to the solution. For medium sized problems, we see that with the given algorithms we do not need to conduct any sort of value iteration to converge at the optimum, but rather the dynamic programming has been reduced to matrix multiplications. Hence, the algorithm provided here outperforms conventional Dynamic Programming tools such as Dynamic Programming via Linear Programming or value iteration because this algorithm has been tailored to our specific problem. In the following section we apply our results to small example problems.

IV. Extension: Observation Cost

The results obtained thus far have imposed a hard constraint on the number of opportunities available for observations, however no explicit cost was accrued from making an observation. This can indeed be the case in scenarios where a sensor network is tracking an adversary with a predetermined budget that can be exhausted. In other situations, however, one may also imagine that there would be an explicit cost on the observation in addition to the hard constraint. This explicit cost could come from resources necessary to scan a network, for example. One may still like to keep the number of disruptions below a certain number, but also consider the cost of taking an action as well. In this section, we extend the methods used in the previous section to accommodate this modified model.

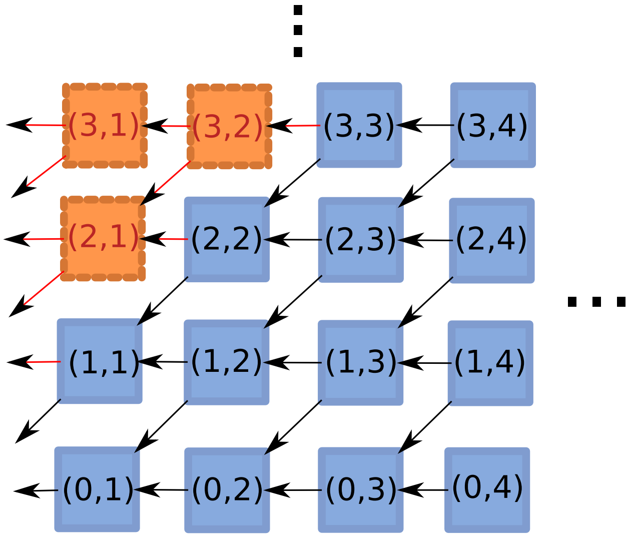

Figure 3. New (previously inadmissible) states and transitions. The pair (s,t) represents the number of observations and remaining time steps, respectively.

A. New Model

We proceed with the same formulation proposed in Section III with an added feature to the model: the cost of making an observation (taking an action $u = 1$) is c. We may now ask what the interpretation of this is as related to our original distance measure $d(\cdot, \cdot)$. We suppose that a linear cost is attached with the estimation error at each time step. This cost is measured in the same units as the cost c of making an observation. Note that our problem statement in this form now allows for a new degree of freedom that was not seen in the previous section: in the backwards induction process, it is now necessary to consider states for which $s > t$ (Figure 3), since it is possible to come to such a state by not using an observation even when $s = t$. This would happen if the cost of using an observation is prohibitively high, a scenario left unconsidered earlier.

B. Dynamic Programming

Beginning with stage $t = 1$, we can reuse our previous calculation of

$$J_{(r,0,1)} = e_r^*$$

since an added observation cost does not change this quantity. In the $s = 1$ case, however, we now must decide whether it is worthwhile to use this observation opportunity. We can write:

$$J^{(0)}_{(r,1,1)} = e_r^* \quad J^{(1)}_{(r,1,1)} = c$$

where we have abbreviated c to be the vector where all elements are c. This results in a function

$$\Delta_{(r,1,1)} = e_r^* - c$$

with associated set $\tau_{(r,1,1)}$. Larger values of s behave the same way as $s = 1$. For $t = 2$, we again recycle the result

$$J_{(r,0,2)} = e_r^* + e_{r+1}^*$$

and must consider $s = 1$ as follows:

$$\begin{aligned} J^{(0)}_{(r,1,2)}(x) &= e_r^* + c + \mathbf{1}_{(r+1,1,1)}(x)\Delta_{(r+1,1,1)}(x) \\ J^{(1)}_{(r,1,2)}(x) &= c + P^r e_1^* \end{aligned}$$

Once again, $\Delta_{(r,1,2)}$ and $\tau_{(r,1,2)}$ can be found by construction.

For the $s = 2$ case we must again look at both $u = 0$ and $u = 1$ cases.

$$\begin{aligned} J^{(0)}_{(r,2,2)}(x) &= e_r^* + c + \mathbf{1}_{(r+1,2,1)}(x)\Delta_{(r+1,2,1)}(x) \\ J^{(1)}_{(r,2,2)}(x) &= 2c + \mathbf{1}_{(r+1,1,1)}(x)\Delta_{(r+1,1,1)}(x) \end{aligned}$$

Larger values of s behave exactly the same as $s = 2$, and Δ as well as τ can be constructed in the usual way.

We can continue in the same manner as the previous section, incrementing t and s accordingly. We omit these details and present the solution, whose structure closely mirrors the no-cost case.

C. Solution

An optimal policy is given by

$$u_{(r,s,t)}(x) = \begin{cases} 0 & \text{if } x \in \tau^c_{(r,s,t)} \\ 1 & \text{otherwise} \end{cases}$$

We again have three vector valued functions: $F_{(r,s,t)}, \Delta_{(r,s,t)} \in \mathbf{R}^S$ and $\mathbf{1}_{(r,s,t)} \in \{0,1\}^S$. We fill in values for these functions by using the following recursions:

$$\begin{aligned} F_{(r,s,t)} &= F_{(r+1,s-1,t-1)} + c \\ &\quad + P^r diag(\mathbf{1}_{(1,s-1,t-1)})\Delta_{(1,s-1,t-1)} \\ \Delta_{(r,s,t)} &= e_r^* + F_{(r+1,s,t-1)} - F_{(r,s,t)} \\ &\quad + diag(\mathbf{1}_{(r+1,s,t-1)})\Delta_{(r+1,s,t-1)} \\ \mathbf{1}_{(r,s,t)}(x) &= \begin{cases} 0 & \text{if } \Delta_{(r,s,t)}(x) > 0 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

for $1 < s, t < N$ and $1 \leq r \leq N - t + 1$. We also have the boundary conditions

$$F_{(r,1,t)} = c + P^r \sum_{j=1}^{t-1} e_j^*, \quad \Delta_{(r,s,1)} = e_r^* - c$$

These recursions allow us to determine the sets $\tau^c_{(r,s,t)}$ for s, t, r in the bounds specified, which in turn defines our optimal policy. Specifically, we assign

$$x \in \tau^c_{(r,s,t)} \Leftrightarrow \Delta_{(r,s,t)}(x) \leq 0$$

We conclude by giving expressions for the cost-to-go from any particular state when a particular action $u \in \{0, 1\}$ is taken. The superscripts denote whether or not an observation will be made in the current stage.

$$\begin{aligned} J^{(0)}_{(r,s,t)} &= e_r^* + F_{(r+1,s,t-1)} + diag(\mathbf{1}_{(r+1,s,t-1)})\Delta_{(r+1,s,t-1)} \\ J^{(1)}_{(r,s,t)} &= F_{(r,s,t)} \end{aligned}$$

The modification to our algorithm is surprisingly minimal - we only need to add cost c in the appropriate places to consider this larger class of problems. Indeed, in this case we are able to profit from the work that was required in Section III.

D. Implementation Optimization

Note that in this modified solution structure, the number of possible dynamic programming states has approximately doubled. This is due to the fact that dynamic programming states for which $s > t$ are now possible. However, once can also see that for quantities indexed as (r, s, t) where $s > t$, the values are exactly the same as for $s = t$. In fact, the only thing changing is the indexing, since there is no utility to observations that cannot be used. For reducing complexity during deployment then, one could simply collapse $s > t$ states into the $s = t$ state, but for the purposes of clarity and accounting for observation usage, we have chosen to represent them as different states.

V. Extension: Large State Spaces

We now include a short note about dealing with very large state spaces. For the most part, when state spaces become prohibitively large in this type of problem setting, one must consider the specific structure of the problem at hand to find a technique to either approximate or simply the true problem. There are, however, a couple general methods to cut down on the problem size.

A. Breadth First Search

In some cases, the size of state space is much larger than the subset of states that are reachable for the process in the time horizon under consideration. This is unlikely to happen since the size of a set covered by breadth first search exponentially increases, but for problems with a small time horizon, it is a good first step and suffers no performance loss. The downside is that this approach is applicable only for shorter time horizon problems.

B. Agglomeration of States

Another way to reduce the complexity of the problem at hand is to examine the distance measure and combine states that are close to each other compared to the average distance between states. A threshold can be set for how close two states must be in order to warrant agglomeration. This threshold can be used to bound the additional error accrued due to this simplification. This method can be effective if there is a high probability that the process will take many of the longer transitions over the course of the problem, but can be a poor approximation technique if the intruder spends many time steps traversing the smaller arcs of the graph.

C. Truncation of States

Still another way to reduce the number of states under consideration in the solution for this problem is to find those states that are probabilistically unlikely to be reached. These states on the Markov chain can be omitted. Indeed, in the case of hypercubes and euclidean distance as the state space and measure respectively, there are results bounding the probability with which the process will drift outside a given radius. Depending on the resources at hand, one can perform the prescribed state space reduction in several ways. One method can be to simulate many paths and eliminate those that have not been reached often. Another technique can be to simply find the transitions in the Markov chain with lowest probability and remove them until many states are no longer reachable. The downside of eliminating seldom reached paths could also introduce the danger of missing new intrusion patterns.

D. Combination

In reality, a combination of these approaches should be attempted when attempting to simplify a problem. One can combine the agglomeration and truncation approaches by combining only those states that satisfy a proximity metric in addition to being unlikely to be reached.

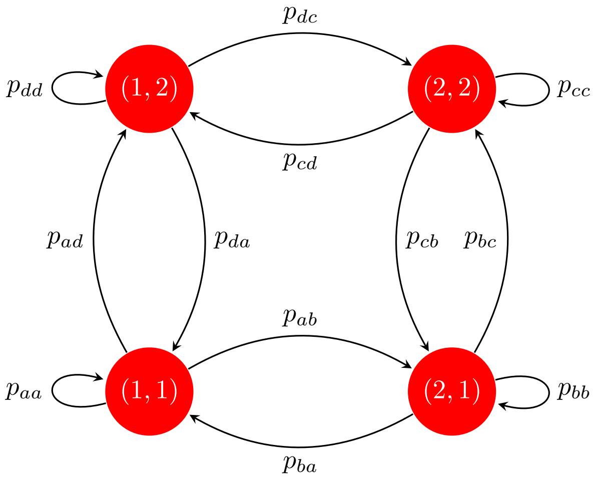

Figure 4. Markov chain $\mathcal{M}_{2\times 2}$

VI. Numerical Results

Let us now examine the performance of our algorithm. Everything that follows pertains to the no-cost observation case, unless explicitly stated. We fix a horizon length and plot the cost that the prescribed algorithm accrues versus the number of opportunities to make observations. Let us consider Markov chains of the type $\mathcal{M}_{n\times n}$ in Figure 4, which is an n-by-n grid of states where the transition probabilities are given in the figure. Such a construction is simple enough for quick simulation but can capture the inherent variations that our algorithm is able to leverage.

A. Surveillance

Suppose we would like to track the position of an intruder in an environment modeled by the Markov chain $\mathcal{M}_{3\times 3}$ over a discrete-time horizon of 30 time slots. However, updating the location of the intruder requires battery power of a mobile device due to communications with a satellite and hence we are not able to request the position of the intruder at every time. Fixing the initial position of the device to be $(2,1)$, let us vary the number of opportunities to retrieve the true location from 0 to 30. The distance metric we take is the standard Euclidian norm, which may be represented in matrix form as:

$$D = \begin{bmatrix} 0 & 1 & 2 & 1 & \sqrt{2} & \sqrt{5} & 2 & \sqrt{5} & \sqrt{8} \\ 1 & 0 & 1 & \sqrt{2} & 1 & \sqrt{2} & \sqrt{5} & 2 & \sqrt{5} \\ 2 & 1 & 0 & \sqrt{5} & \sqrt{2} & 1 & \sqrt{8} & \sqrt{5} & 2 \\ 1 & \sqrt{2} & \sqrt{5} & 0 & 1 & 2 & 1 & \sqrt{2} & \sqrt{5} \\ \sqrt{2} & 1 & \sqrt{2} & 1 & 0 & 1 & \sqrt{2} & 1 & \sqrt{2} \\ \sqrt{5} & \sqrt{2} & 1 & 2 & 1 & 0 & \sqrt{5} & \sqrt{2} & 1 \\ 2 & \sqrt{5} & \sqrt{8} & 1 & \sqrt{2} & \sqrt{5} & 0 & 1 & 2 \\ \sqrt{5} & 2 & \sqrt{5} & \sqrt{2} & 1 & \sqrt{2} & 1 & 0 & 1 \\ \sqrt{8} & \sqrt{5} & 2 & \sqrt{5} & \sqrt{2} & 1 & 2 & 1 & 0 \end{bmatrix}$$

and we choose the transition matrix to be

$$P = \begin{bmatrix} 0 & 0.1 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0.8 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0.8 & 0 & 0 \\ 0 & 0.7 & 0 & 0.15 & 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.4 & 0.1 \end{bmatrix}$$

where we have ordered the states by the first index and then the second (that is in order $(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)$). We note that the topology of the state space with the chosen distance metric is rather uniform, but the transition probabilities widely vary from state to state.

We expect the estimation error to monotonically decrease with the number of opportunities to learn the true state. In Figure 5, we see that this indeed the case, and also compare it to a benchmark strategy of randomly distributing observations.

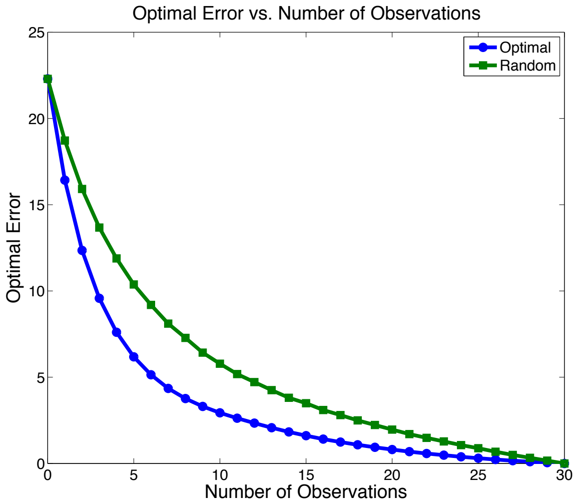

Figure 5.

Another expected property is that for fixed r, t, as s increases, the number of states for which $\Delta_{(r,s,t)} \geq 0$ decreases. That is, we expect that having more opportunities to make observations results in a more liberal optimal policy, and vice versa. We see this in Figure 6.

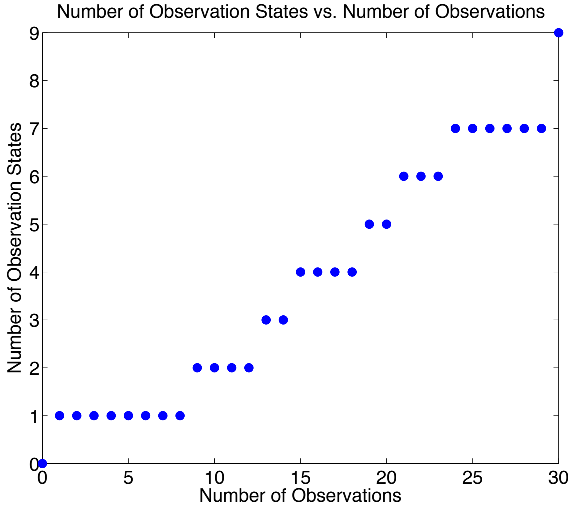

Figure 6.

Finally, let us consider one more plot with the same Markov chain states and distance metric, but a different transition matrix. Specifically, let us choose transitions that have uniform probabilities to each neighbor. The resulting matrix is given by

$$P = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

This removes most of the variability from the problem - in fact, the only non-uniformity is due to the fact that the state space is not large compared to the horizon of the problem, and hence, the edges introduce some small amount of variation. In Figure 7, it is apparent that the optimal policy is practically a straight line. There is not much variability to exploit, so we aren't able to exploit situations with sparse observations as we could in Figure 5.

B. Analysis of Performance

We now note several properties of our curve in Figures 5,6 and 7. In some cases, the justification for the property is clear and we briefly explain it, where as in others we delve into a more complete proof.

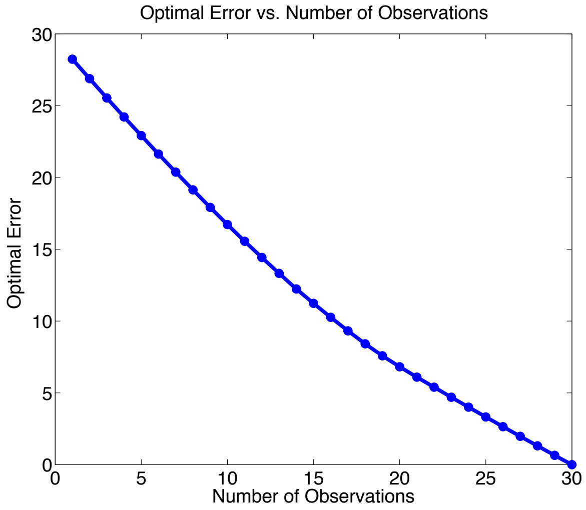

Figure 7.

1) Endpoints: First, the endpoints in an estimation error vs. number of observations plot are fixed no matter what policy is used. This is because when there are zero opportunities to make observations or there are 30 chances to view the process, there is no way to come up with policies that result in different decisions. There is only one way to allocate opportunities to observe the process. Moving now to Figure 6, we note that the end points in this type of curve are also fixed. Specifically, in any plot of number of states with $\Delta_{(r,s,t)} \geq 0$ vs. number of observations, we have the points $(0,0)$ and $(t, |S|)$. The point $(0,0)$ is guaranteed because without any observations, there is no chance that one can be made at any state. The point $(t, |S|)$ is certain because when the number of observations is equal to the number of time steps remaining in the problem, the cost $J^{(0)}$ of not observing can never be less than the cost $J^{(1)}$ of making an observation.

2) Diminishing Returns: Next, in Figure 5 we note that our algorithm outperforms a benchmark strategy of randomly placing observations over the 30 time slots. We see that the greatest "savings" occurs when we have a sparsity of opportunities to make observations. This can be quantified by how strong the convexity of this curve is. We will shortly prove that these curves are always convex, but the salient point here is that as opportunities to observe the process are more readily available, there is a law of diminishing returns and these opportunities become less valuable. The degree of convexity depends greatly on the transition matrix P of the Markov chain. For example, if the grid $\mathcal{M}_{n\times n}$ has transitions that are all equal, the optimal policy comes out to be almost a straight line, as we verified in Figure 7. This is because there is little variation in the Markov chain to exploit. A highly variable Markov chain would allow a single observation to reduce much variability in future predictions, hence reducing error drastically with a small budget.

3) Monotonicity: In the two types of plots we have given, monotonicity is another property that is present in general. First, let us consider plots of the type in Figure 5 and 7. In error vs number of observations plots, we can prove monotonicity by contradiction. Suppose that these curve are not guaranteed to be monotonically decreasing and that there exists some model and s such that $J_{(r,s,t)} < J_{(r,s+1,t)}$. The the policy for $s+1$ could not possibly be optimal, because we can simply apply the policy for s to the same problem and achieve better performance. The plot in Figure 6 is monotonically increasing, a property that also matches our expectation: as the number of opportunities to make observations increases, the probability of making one (under a uniform prior) increases, and therefore the number of states that result in an observation being made should increase.

4) Convexity: Finally, we observe the convexity of the optimal cost vs. number of observations curve. Indeed, it is consistent with our intuition that having an extra opportunity to make observations should be of greater utility when observations are sparse and less utility when they are abundant. We can sketch a proof for this. First note that the inequality we would like to prove is

$$J_{(r,s,t)} \leq \frac{1}{2}\left(J_{(r,s-1,t)} + J_{(r,s+,t)}\right)$$

in the range $0 < s < t$. We can rewrite this as

$$2J_{(r,s,t)} \leq J_{(r,s-1,t)} + J_{(r,s+,t)}.$$

Let us change perspective at this point and consider this a problem not in allocating observations, but rather in allocating 'holes', or instances without observations. We want to dynamically schedule holes in a way that the estimation error accrued due to the presence of these holes is minimal. Estimation error is only accrued for holes, not for observations. Let us now consider two processes happening in parallel of horizon t and the same value r. One process has $\hat{s}$ holes to be allocated, and the other has $\hat{s}-1$ holes to be allocated. Suppose we must now choose a process to which another hole is to be added. That is, an additional estimate must be formed on one of the two processes in such a way to minimize the total estimation error of the two processes. It is clear that we should choose the process with fewer holes since this process has more information about the state of the process and hence is likely to induce a lower estimation error increase due to the additional hole. We can see this more graphically in Figure 8.

Returning to our original problem, we can translate this to conclude that it is preferable, in the event of two processes with $J_{(r,s-1,t)}$ and $J_{(r,s,t)}$, to add an observation to the one with fewer observations. That is, $2J_{(r,s,t)}$ is preferable to $J_{(r,s-1,t)} + J_{(r,s+1,t)}$, which is what we wanted to show.

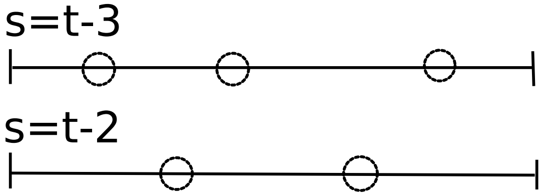

Figure 8. Allocation of holes to two processes.

VII. Conclusion and Future Work

In this paper, we have described a problem in monitoring over a finite horizon when there are a limited number of opportunities to conduct surveillance. We mathematically model this as a problem of state estimation. In the estimation problem we hope to minimize the distortion from estimating the state of a Markov chain when the number of time the process may be viewed is limited to a few times over the total horizon. The distortion is measured using a specified metric $d(x, y)$, which tells us how "far apart" states x and y are.

In our optimal policy, a set of recursive equations with boundary conditions give a practical method for determining an optimal policy. Although the policy could have been determined using standard methods in dynamic programming, such as value iteration, the algorithm given here relies only on the ability to store data and conduct matrix multiplications. Hence, larger problems can be handled before intractability results due to state space complexity.

Extensions to this basic formulation are then covered, such as a treatment of the same problem with a cost on making observations. Techniques for handling problems with very large state spaces are also discussed. Finally, several structural properties of the solution are presented.

There are many further problems to consider in future work. Rather than fixing the problem of interest to a particular horizon length, we may consider problems with a variable horizon. That is, we might consider problems in which the Markov chain dictates a random stopping time for the process during which we may only make observations a limited number of times. Additionally, there are practical scenarios in which one does not have complete information about the transition matrix. In this case, we may be interested in coupling parameter estimation with efficient budget allocation. Finally, distributed problems in which many sensors are available for measurement but each has a battery limitation are of great interest, and certainly can be explored in the context of the budgeted estimation scheme suggested here.

Overall, the area of budgeted estimation holds much promise, and there are many avenues left to investigate in this power limited framework.

Acknowledgment

The first author is supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

[1] P. Bommannavar and N. Bambos, "Optimal State Surveillance under Budget Constraints," in *Proceedings of the Second International Conference on Emerging Network Intelligence*, Florence, Italy: IARIA, October 2010, pp. 68-73.

[2] E. Wilson, *Network Monitoring and Analysis: A Protocol Approach to Troubleshooting*. Prentice Hall, 2000.

[3] D. Josephsen, *Building a Monitoring Infrastructure with Nagios*. 1st ed., Prentice Hall, 2007.

[4] Microsoft Security Center, Retrieved from http://technet.microsoft.com/en-us/security, May, 2010, accessed: Jan 2012.

[5] General Accounting Office, Information Security: Computer Attacks at Department of Defense Pose Increasing Risks. GAO/AIMD-96-84, May, 1996.

[6] R. A. Miura-Ko and N. Bambos, "Dynamic risk mitigation in computing infrastructures," in *Third International Symposium on Information Assurance and Security*. IEEE, 2007, pp. 325 - 328.

[7] K. C. Nguyen, T. Alpcan, and T. Basar, "Fictitious play with imperfect observations for network intrusion detection," *13th Intl. Symp. Dynamic Games and Applications (ISDGA)*, Wroclaw, Poland, June 2008.

[8] T. Alpcan and X. Liu, "A game theoretic recommendation system for security alert dissemination," in *Proc. of IEEE/IFIP Intl. Conf. on Network and Service Security (N2S 2009)*, Paris, France, June 2009.

[9] H.V. Poor, *An Introduction to Signal Detection and Estimation*. 2nd ed., Springer-Verlag, 1994.

[10] N. E. Nahi, "Optimal recursive estimation with uncertain observation," in *IEEE Transactions on Information Theory*, vol. 15, no. 4, pp. 457 - 462, July 1969.

[11] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9 , pp. 1453 - 1464, September 2004.

[12] S. Yuksel, O. C. Imer and T. Basar, "Constrained state estimation and control over communication networks," in *Proc. of 38th Annual Conference on Information Science and Systems (CISS)*, Princeton, NJ, March 2004.

[13] J. Fuemmeler and V.V. Veeravalli, "Smart Sleeping Policies for Energy Efficient Tracking in Sensor Networks," IEEE Transactions on Signal Processing, vol. 56 no. 5: pp. 2091-2102, May 2008.

[14] N. Agarwal, J. Basch, P. Beckmann, P. Bharti, S. Bloebaum, S. Casadei, A. Chou, P. Enge, W. Fong, N. Hathi, W. Mann, A. Sahai, J. Stone, J. Tsitsiklis, and B. Van Roy, “Algorithms for GPS Operation Indoors and Downtown,” GPS Solutions, Vol. 6, No. 3, pp. 149-160, December 2002.

[15] S. Appadwedula, V.V. Veeravalli, and D.L. Jones, “Energy Efficient Detection in Sensor Networks,” in *IEEE Journal on Selected Areas in Communications*, vol. 23 no. 4, pp. 693-702, April 2005.

[16] O.C. Imer. Optimal Estimation and Control under Communication Network Constraints. Ph.D. Dissertation, UIUC, 2005.

[17] P. Bommannavar and N. Bambos, Patch Scheduling for Risk Exposure Mitigation Under Service Disruption Constraints. Technical Report, Stanford University, 2010.

[18] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientic, 1995.