Adaptive Multi-modality Sensor Scheduling for Detection and Tracking of Smart Targets

Chris Kreucher, Doron Blatt, and Alfred Hero
The University of Michigan
Dept. of Electrical Eng. and Computer Science
Ann Arbor, MI 48109-2122
{ckreuche, dblatt, hero}@umich.edu

Keith Kastella
General Dynamics
Advanced Information Systems
Ann Arbor, MI, 48113-4008
Keith.Kastella@gd-ais.com

Abstract— **This paper considers the problem of sensor scheduling for the purposes of detection and tracking of "smart" targets. Smart targets are targets that are able to detect when they are under surveillance and react in a manner that makes future surveillance more difficult. We take a reinforcement learning approach to adaptively schedule a multi-modality sensor so as to most quickly and effectively detect the presence of smart targets and track them as they travel through a surveillance region. An optimal scheduling strategy, which would simultaneously address the issue of target detection and tracking, is very challenging computationally. To avoid this difficulty, we advocate a two stage approach where targets are first detected and then handed off to the tracking algorithm.**

I. Introduction

The problem of sensor scheduling is to determine the best way to task a sensor or group of sensors when each sensor may have many modes and search patterns. Tasking a sensor may include such choices as where to point, what mode to use, and what signal to transmit. In general, sensors must balance complex tradeoffs between competing mission goals, e.g. detection of new targets, tracking of existing targets, and identification of existing targets.

An optimal sensor scheduling algorithm will depend on the posterior distribution of the system state conditioned on sensor measurements. In our application, the system state describes probabilistically both the uncertainty in number of targets and locations of the individual targets. In principle, one could derive an optimal scheduling algorithm that simultaneously treats detection of new targets and tracking of existing targets by defining an appropriate global reward. However, in practice, this is very difficult due to computational considerations. To combat these computational challenges, we take a modular approach and treat the problem in two stages – target detection followed by target tracking. This suboptimal algorithm can be viewed as an approximation to an optimal algorithm which simultaneously considers detection and tracking.

Sensor scheduling is complicated substantially when targets under surveillance are able to detect and respond to sensing activities (so called "smart" targets). In this paper, we consider one such scenario. Specifically, we investigate the situation where a sensor is charged with detecting and tracking a group of moving ground targets and the targets have the ability to detect some of the surveillance actions and respond by concealing their whereabouts.

The paper proceeds as follows. In Section II, we outline the mathematics and strategy of our two stage detection and tracking algorithm. We first give an overview of reinforcement learning methods, and then describe the application of reinforcement learning to the target detection stage and the tracking stage. In Section III, we provide simulation results of the algorithm for two smart targets. The method is compared to random and myopic strategies and shown to provide good performance. Finally, in Section IV we conclude with some summarizing remarks.

II. Smart Target Detection and Tracking

In this section, we describe the details of our two stage detection and tracking algorithm. We first review reinforcement learning and then show its application to each stage.

A. Reinforcement Learning for Optimal Solution of a MDP

The problem of detecting and tracking smart targets can be formulated as an infinite-horizon Markov Decision Process (MDP) [13]. It is well known that the complexity of finding optimal policies for MDP grows exponentially with the state and action spaces [2]. Since the sensor scheduling problem is characterized by extremely large state and action spaces, it is necessary to develop approximate solutions using dimension reduction. We advocate methods from reinforcement learning coupled with function approximation to find approximately optimal policies for the two stages.

1) Infinite-Horizon MDP: A discounted-reward infinite-horizon MDP is defined by a sequence of states $\{S_t\}_{t\geq 0}$ taking values in a state space $\mathcal{S}$, a sequence of actions $\{A_t\}_{t\geq 0}$ taking values in an action space $\mathcal{A}$, and a (possibly random) reward function $r(S_t, A_t)$ that assigns the cost incurred (when negative) or the reward gained (when positive) to the event of being at state S_t and taking action A_t. In our context, the state space characterizes the battlefield. It contains rich information such as the number of targets present, their location, their type, and whether they are stationary or moving. The action space contains all the possible actions. Each action specifies which sensors to use, their mode of operation, and where to point them. The reward system reflects the tradeoffs between costs of deploying a certain sensor and the gain earned from the measurement it collects.

The process is initiated with state S_0 followed by action A_0 chosen by the controller and continues with the sequence $S_1, A_1, S_2, A_2, \ldots$. Under the Markovian model, given S_t and A_t, S_{t+1} is independent of all past states and actions. The state transitions are governed by a stationary probabilistic law, denoted by $p(S_{t+1}|S_t, A_t)$, that specifies the distribution of S_{t+1} over $\mathcal{S}$, given S_t and A_t. $p(S_{t+1}|S_t, A_t)$ is either a probability density function when the state space is continuous or a probability mass function when it is discrete.

A stationary policy Π is a map from $\mathcal{S}$ to $\mathcal{A}$ that specifies the action taken at each state. Denote the class of all policies by $\mathcal{P}$. The value function associated with policy Π, denoted by $V^{\Pi}(s)$ is the expected total discounted reward when being in state $S_t = s$ and following policy Π, that is

$$V^{\Pi}(s) = \mathrm{E}\left\{\sum_{\tau=t}^{\infty} \beta^{\tau-t} r(S_\tau, \Pi(S_\tau))|S_t = s\right\} \quad \forall s \in \mathcal{S} \ , \tag{1}$$

where $\beta \in (0, 1)$ is a discount factor, which is included to value future rewards less than immediate rewards. This expectation is taken with respect to the joint distribution of all the targets, which, in the context of smart targets, is highly dependent on the action sequence. Therefore, a direct calculation of this expression is computationally intractable. An optimal policy is a policy that satisfies

$$\Pi^*(s) = \arg\max_{\Pi \in \mathcal{P}} V^{\Pi}(S) \qquad \forall s \in \mathcal{S} \ . \tag{2}$$

It is well known that the optimal policy is the unique solution to Bellman's equation. Unfortunately, when the state and action spaces are large and the state transition density is either computationally complicated or not explicitly available, this methodology is intractable and one must resort to approximate solutions such as Q-learning [2].

2) Q-Learning: The optimal scheduling policy for the two stages is found using Q-learning coupled with function approximation [17], [15], [16]. The learning part relaxes the requirement for an explicit knowledge of the transition density, and function approximation is used to further reduce the dimensionality of the state and action spaces.

Given the optimal value function V^*, the Q-function is

$$Q(s, a) = \mathrm{E}\left\{r(s, a) + \beta V^*(S_{t+1})|S_t = s, A_t = a\right\} \ , \tag{3}$$

i.e., the expected reward when taking action a at state s and then acting optimally. The Q-function satisfies the equation

$$Q(s, a) = \mathrm{E}\left\{r(s, a) + \beta \max_{a \in \mathcal{A}} Q(S_{t+1}, a)|S_t = s, A_t = a\right\} \tag{4}$$

Given the Q-function, optimal actions are computed as

$$\arg\max_{a \in \mathcal{A}} Q(S_t, a) \ . \tag{5}$$

In Q-learning the Q-function is estimated from multiple trajectories of the process. Assume first that both $\mathcal{S}$ and $\mathcal{A}$ are finite. Then, there exists a lookup table representation of $Q(s, a)$. In this case, given an arbitrary initial value of $Q(s, a)$, the one-step Q-learning algorithm ([15], p. 148) is given by the repeated application of the update equation

$$Q(s, a) \leftarrow (1-\gamma)Q(s, a) + \gamma\left(r + \beta \max_{\alpha \in \mathcal{A}} Q(s', \alpha)\right) \ , \tag{6}$$

where each of the 4-tuples $\{S_t = s, A_t = a, S_{t+1} = s', R_t = r\}$ are incurred during the progress of the MDP, and $\gamma \in (0, 1)$ decreases with t. In most realistic problems (the problems discussed herein included) it is infeasible to represent the Q-function in a lookup table, either because the number of states is too large or simply because the state space is continuous. Therefore, we require a function approximation technique to represent the Q-function. The standard and simplest class of Q-function approximators are linear combinations of basis functions (also called features), i.e. $Q(s, a) = \theta^T \phi(s, a)$, where $\phi(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^L$ is a feature vector associated with state s and action a and the coefficients of $\theta \in \mathbb{R}^L$ are to be estimated. Gradient descent is used with the training data to update the estimate of θ, i.e.

$$\begin{aligned}\theta \quad &\leftarrow \quad \theta + \gamma\left(r + \beta \max_{a'} Q(s', a') - Q(s, a)\right) \nabla_\theta Q(s, a) \\ &= \quad \theta + \gamma\left(r + \beta \max_{a'} \theta^T \phi(s', a') - \theta^T \phi(s, a)\right) \phi(s, a) \ ,\end{aligned}$$

Once the learning of the vector θ is completed, optimal actions can be computed according to $\arg\max_{a \in \mathcal{A}} \theta^T \phi(S_t, a)$.

B. Detection of Smart Targets using Reinforcement Learning

The target detection stage is formulated as a Bayesian hypothesis testing problem in which one is trying to decide between $M \geq 2$ hypotheses: $\mathrm{H}_1, \ldots, \mathrm{H}_M$. The observed system is modelled as a MDP with a finite state space $\mathcal{S}$ with cardinality N. Each hypothesis corresponds to a different subset of the states and it is assumed that there are no transitions between states that are associated with different hypotheses.

At each time t, one of K modes denoted by $\Sigma_1, \ldots, \Sigma_K$ is used to collect a measurement $\mathbf{z}_t$, or alternatively a final decision is made. The possible actions available at each time are $\mathcal{A} = \{\Sigma_1, \ldots, \Sigma_K, D\}$, where D stands for the action of making the final decision. After action D the detection process ends and a reward is granted for a correct decision.

Denote by $f_k(\mathbf{z}|s)$ the conditional density of a measurement collected by mode k given the system is at state s. The state transition probabilities of the Markov process $p(S_{t+1}|S_t, A_t)$ depend on the deployed sensor mode. The possible states in $\mathcal{S}$ are enumerated from 1 to N and the transition probabilities are summarized in the matrices $\mathbf{A}_k$, $k = 1, \ldots, K$, where $[\mathbf{A}_k]_{nl} = p(S_{t+1} = n|S_t = l, \Sigma_k)$, $\quad n, l = 1, \ldots N$ is the probability that the system moves from state l to state n when sensor mode k is used.

The dependency on the deployed sensor mode is applicable when a target can sense that it is being observed and may react accordingly, e.g. hide or unfold its radar antenna. Since the number of states is finite and known, we can use the vector notation $\mathbf{p}_t$, to denote the posterior probability vector of the target states given $\mathbf{Z}_t$. Using this notation, when sensor k was deployed and collected measurement $\mathbf{z}_{t+1}$, the time update is

$$\mathbf{p}_{t+1} = \frac{\mathbf{A}_k \mathrm{diag}([f_k(\mathbf{z}_{t+1}|1), \ldots, f_k(\mathbf{z}_{t+1}|N)])\mathbf{p}_t}{\mathrm{sum}\left(\mathbf{A}_k \mathrm{diag}([f_k(\mathbf{z}_{t+1}|1), \ldots, f_k(\mathbf{z}_{t+1}|N)])\mathbf{p}_t\right)} \ , \tag{7}$$

where $f_k(\mathbf{z}_{t+1}|n)$ denotes the conditional density of a measurement that was collected by sensor k given that the system is in state n, and for any vector $\mathbf{v}$, $\mathrm{diag}(\mathbf{v})$ is a diagonal matrix with the elements of $\mathbf{v}$ on its diagonal, and $\mathrm{sum}(\mathbf{v})$ is the sum of its elements. Therefore, a policy can be defined as a map from $\mathcal{S}^N$, the simplex of N-dimensional probability vectors, to $\mathcal{A}$. The expected total reward at information state $\mathbf{p}_t$ becomes

$$V^{\Pi}(\mathbf{p}_t) = \mathrm{E}\left\{\sum_{\tau=t}^{\infty} \beta^{\tau-t} r(\mathbf{p}_\tau, \Pi(\mathbf{p}_\tau))\right\} \ , \tag{8}$$

with optimal policy is $\Pi^* = \arg\max_{\Pi\in\mathcal{P}} V^{\Pi}(\mathbf{p})$. The Q-function is defined over the N-dimensional simplex $\mathcal{S}^N$ and for any action $a \in \mathcal{A}$ by

$$Q(\mathbf{p}_t, a) = \mathrm{E}\left\{r(\mathbf{p}_t, a) + \beta V^*(\mathbf{p}_{t+1})\right\} \ , \tag{9}$$

which is the expected reward when taking action a at information state $\mathbf{p}_t$ and then acting optimally. As described earlier, the dimensionality of the information state space is reduced by a linear parametrization, and Q-learning is used to approximate the Q-function. Given Q, one finds the optimal policy by taking the action that maximizes it at any given information state.

C. Tracking of Smart Targets using Reinforcement Learning

Tip-offs from the detection algorithm are used to initialize a tracking algorithm which finely geolocates and tracks moving targets. Targets are tracked by recursively estimating a conditional probability density known as the Joint Multitarget Probability Density (JMPD) [7], [8]. In this paper, we restrict ourselves to the case where the number of targets is known and fixed and the state vectors of individual targets are a scalar. More general implementations are given in [8].

1) The JMPD and Particle Filter Approximation: In the tracking stage, the state s of the system (see Section II-A) is given by the joint multitarget probability density. In this subsection, we show how the state is derived and how states are combined with measurements to determine the next state.

We define the joint multitarget conditional probability density $p(\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots \mathbf{x}_t^{T-1}, \mathbf{x}_t^T|\mathbf{Z}_t, T_t)$ as the probability for T targets with states $\mathbf{x}^1, \mathbf{x}^2, \ldots \mathbf{x}^{T-1}, \mathbf{x}^T$ at time t based on a set of observations $\mathbf{Z}_t$. As before, $\mathbf{Z}_t$ refers to the collection of measurements up to and including time t, i.e. $\mathbf{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_t\}$, where each of the $\mathbf{z}_i$ may be a single measurement or a vector of measurements made at time i. Each of the state vectors $\mathbf{x}^i$ in the JMPD is a vector quantity and may (for example) be of the form $[x, \dot{x}, y, \dot{y}]'$. For convenience, the density will be written more compactly as $p(\mathbf{X}_t, T_t|\mathbf{Z}_t)$.

The sample space of $\mathbf{X}$ is very large. It contains all possible configurations of state vectors $\mathbf{x}^i$. We find that a particle filter based representation of the JMPD allows tractable implementation [8]. The particle filter approximation represents the JMPD by a collection of weighted samples, i.e.

$$p(\mathbf{X}, T|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \ . \tag{10}$$

2) Information Based Myopic Sensor Management: We use the JMPD to make tasking decisions. A good measure of the quality of an action is the reduction in entropy expected to be induced. Therefore, the reward (see Section II-A) will be given by the information gained. To schedule a sensor, we enumerate all possible sensing actions and calculate the *expected* gain in information associated with each possible action.

The calculation of information gain between two densities f_1 and f_0 is done using the Rényi information divergence [14], [5], also known as the α-divergence:

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha - 1} ln \int f_1^\alpha(x) f_0^{1-\alpha}(x) dx \ . \tag{11}$$

In our application, we are interested in computing the divergence between the predicted density $p(\mathbf{X}_{t+1}|\mathbf{Z}_t)$ and the updated density, $p(\mathbf{X}_{t+1}|\mathbf{Z}_{t+1})$.

We choose the sensing action that makes the divergence between the current density and the density after a new measurement largest. Since we do not know the outcome of a sensing action until after the action is taken, we calculate the expected divergence and use this to schedule the sensor. The expected value may be written formally as an integral over all possible outcomes $\mathbf{z}$ when performing sensing action m, i.e.

$$||D_\alpha||_m = \int d\mathbf{z} p(\mathbf{z}|\mathbf{Z}_t, m) D_\alpha\left(p(\cdot|\mathbf{Z}_t, \mathbf{z})||p(\cdot|\mathbf{Z}_t)\right) \ . \tag{12}$$

3) Information Based Non-myopic Sensor Management: As discussed in Section II-A, in many situations a non-myopic sensor management strategy provides sensor tasking decisions having better performance than the myopic strategy. In particular, in the setting considered here where targets are “smart” and react to sensing actions, the regret of choosing a poor action, e.g. active sensing, is long lasting as the effect of an action persists over time. Therefore, a non-myopic strategy will be far superior to a myopic strategy.

We use Q-learning with linear function approximation to learn a policy which behaves non-myopically. The training process involves generation of $\{$*state, action, next state, immediate reward*$\}$ 4-tuples over a large number of training episodes. In the training process, the immediate reward of an action is computed using the actual gain in information as measured by the Rényi Divergence.

III. Simulation Results

We consider a model problem in which an airborne platform is trying to detect and track a set of ground targets. The airborne platform has available a multimode sensor that is able to use an active mode (e.g. radar) or a passive mode (e.g. EO/IR). The sensor is able to quickly steer an antenna so as to focus attention on specific regions of the surveillance area. This is a simple model of a real platform like the USAF JSTARS, which has a 24ft antenna installed on the underside of the aircraft, is able to scan electronically in azimuth and is able to choose between several modes of operation including moving target indicator and synthetic aperture radar.

In this simulation, targets are characterized by their position in one dimension. Targets are "smart" in that they sense when they are under surveillance by an active sensor and react to make future surveillance activities more difficult. The number and location of the targets is unknown initially and our task is to detect and track the targets. The model problem considered here is summarized in Figure 1.

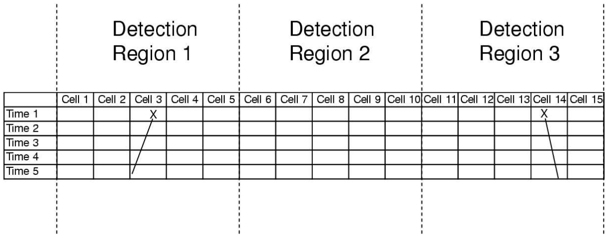

Fig. 1. An illustration of the model problem. The surveillance region is broken into detection regions. The detection algorithm schedules the sensor to most quickly determine the presence or absence of targets in each detection region. Upon detecting targets, the tracking algorithm is tipped off with the regions in which targets exist. The tracking algorithm then determines sensor resource allocations that allows refinement of the initial location and tracking as the targets move through the surveillance area.

A. Target Detection

Each detection region is modelled as taking one of three states: s_1 no target present, s_2 an exposed target is present, and s_3 a camouflaged target is present. There are two hypotheses: H_1 (no target present) and H_2 (a target is present, exposed or camouflaged). The target can move from state 2 to state 3 if it senses that it is being observed. However, it has a tendency to return from state 3 to state 2 if it no longer senses that it is being observed, e.g. it may be less effective in state 3.

Intelligence sources provide a prior on the initial state of the target, which constitutes the initial information state of the process $\mathbf{p}_0$. The platform has one of three sensor modes to deploy. Sensor mode i, deployed at time t provides an independent measurement $z_i(t)$. For the simulation considered here, measurements are assumed conditionally Gaussian.

Modes 1 and 3 represent active modes, which can be sensed by the target, and sensor mode 2 represents a passive mode which cannot be detected by the target. When the target is in hide mode, it has an incentive to return to the exposed state. Sensor mode 3 is less favorable then sensor mode 1 regardless of the system state. It provides less information on the target, and when it is used there is a higher probability that the target will detect it. It was included in this study in order to show that the optimal learned policy will indeed never use it.

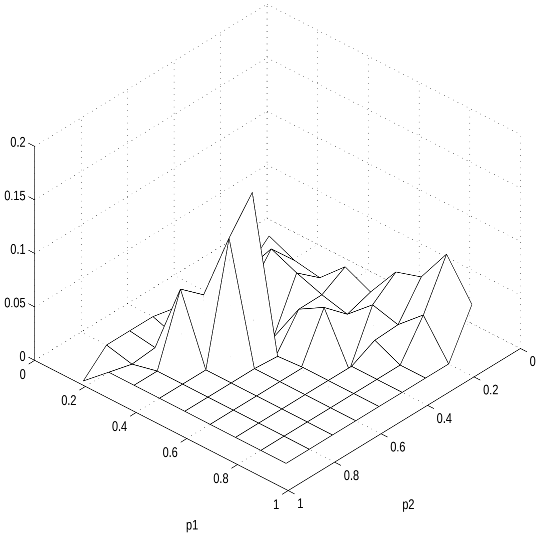

Fig. 2. An improvement over the random allocation policy.

Q-learning (Section II-A.2) was used to approximate the optimal policy. The basis functions (or features) were chosen to be indicator functions of disjoint regions of $\mathcal{S}^3 \times \mathcal{A}$ that correspond to quantization of the simplex $\mathcal{S}^3$ into 55 disjoint regions for each action in $\mathcal{A}$.

The myopic strategy for this problem is to make an immediate decision based on the prior with taking any measurements. Therefore, the estimated optimal policy was compared to a randomized policy in which actions are chosen uniformly. The improvement in terms of the difference in averaged value, estimated from 2000 Monte Carlo simulations at each information state, is presented in Fig. 2.

B. Target Tracking

We assume the target detection algorithm has detected targets in Regions 1 and 3 and passed this information to the target tracking algorithm. At each time step, the sensor is able to measure a single cell to determine the presence or absence of targets. The sensor can use the active (mode 1) or passive (mode 2) modes described above. Sensor modes are characterized by a detection probability P_d and a false alarm probability P_f. These probabilities are linked together via SNR by $P_d = P_f^{1/(1+SNR)}$. This model of sensor returns corresponds to thresholding of Rayleigh distributed energy from targets in Rayleigh distributed background noise as is seen on GMTI radar systems. Note that the sensor characteristics are defined differently than in the detection portion of the algorithm. Unlike the detection regions considered earlier, a sensor cell is now a small area and targets can easily move between cells necessitating the fine grained model.

When the target is in visible mode, the active mode works with high detection probability and low false alarm probability,

$P_d = .9$ and $P_f = 1e{-}4$ (corresponding to SNR = 20dB). The passive sensor mode works with detection probability $P_d = .5$ and false alarm probability $P_f = 1e - 4$ (SNR = 10dB). When in hide mode, both modes are severely degraded and correspond to a target with SNR = 0dB.

Targets can sense when the active mode is used and move into hide mode to prevent further interrogation. Additionally, targets that have moved into hide mode tend to move back into visible mode when the passive sensor mode is used. The parameters of interest can be summarized by the following transition probabilities when for each of the two sensor modes:

$$\begin{bmatrix} Pr(\textit{visible to visible}) & Pr(\textit{visible to hide}) \\ Pr(\textit{hide to visible}) & Pr(\textit{hide to hide}) \end{bmatrix}$$

A myopic strategy makes tasking decisions based only on the expected immediate reward. Here the myopic strategy will advocate using the active mode at all times. Depending on the transition probabilities, this may immediately force the targets into hide mode, making them difficult to observe in future time steps. A non-myopic strategy, on the other hand, will take into account the effect of current actions on future information gaining ability and be more prudent in using the active mode.

In the simulation, we use

$$\text{Transition Matrix Active Sensor Mode} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\text{Transition Matrix Passive Sensor Mode} = \begin{bmatrix} 1 & 0 \\ .2 & .8 \end{bmatrix}$$

which indicates that the target always moves into hide when the active mode is used, and moves from hide to visible with probability .2 when the active mode is used.

We trained a Q-function as discussed in Section II. In Figure 3, we present results of target localization using the Q-learning strategy detailed in Section II. We compare this performance to (a) a random strategy, (b) a myopic strategy, (c) a random strategy that only uses the passive mode, and (d) a myopic strategy that only uses the passive mode. The Q-learning strategy performs as well or better than the best of the four competing strategies in both cases.

IV. Conclusion

In this paper, we have investigated the problem of sensor scheduling for detection and tracking of smart moving ground targets from an airborne sensor. Since the targets of interest are able to detect and respond to certain sensing actions, it is mandatory that the long term ramifications be taken into account when choosing current sensing actions. This necessity for non-myopic sensor scheduling leads to a very computationally challenging problem.

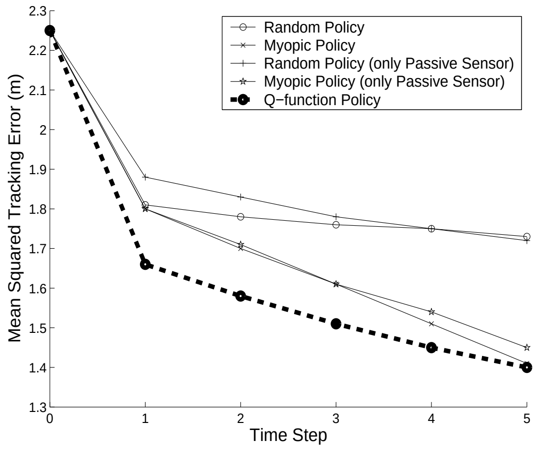

Fig. 3. Target tracking performance. Included are a random strategy, a myopic strategy, a random strategy that uses only the passive mode, a myopic strategy that uses only the passive mode, and the Q-learning strategy.

We have addressed this numerical challenge with a two stage approach, where both stages are solved using reinforcement learning. The surveillance area is first partitioned into a set of detection regions and a detection algorithm determines the presence or absence of a target in each region. Upon detection, a tracking algorithm is used to finely geolocate and track targets as they move through the region.

V. Acknowledgements

This work was supported under the United States Air Force contract F33615-02-C-1199, AFRL contract SPO900-96-D-0080 and by ARO-DARPA MURI Grant DAAD19-02-1-0262. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

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

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.

[3] D. P. Bertsekas and D. Castanon, “Rollout Algorithms for Stochastic Scheduling Problems”, *Journal of Heuristics*, Vol. 5, pp. 89-108, 1999.

[4] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer Publishing, New York, 2001.

[5] A. O. Hero, B. Ma, O. Michel and J. Gorman, “Applications of entropic spanning graphs,” *IEEE Signal Processing Magazine Special Issue on Mathematics in Imaging)*, Vol 19, No. 5, pp 85-95, Sept. 2002.

[6] W. Schmaedeke and K. Kastella, “Event-averaged maximum likelihood estimation and information-based sensor management”, *Proceedings of SPIE*, vol. 2232, Orlando, FL, June 1994, pp. 91-96.

[7] C. Kreucher, K. Kastella, A. O. Hero III, “Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density”, *Under review at IEEE Transactions on AES.*

[8] C. Kreucher, K. Kastella, and A. O. Hero, “Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density”, *Proc. SPIE*, San Diego, California, August 2003.

[9] V. Krishnamurthy, “Algorithms for Optimal Scheduling and Management of Hidden Markov Model Sensors”, *IEEE Transactions on Signal Processing*, Vol. 50, no. 6, pp. 1382-1397, June 2002.

[10] V. Krishnamurthy and D. Evans, “Hidden Markov Model Multiarm Bandits: A Methodology for Beam Scheduling in Multitarget Tracking”, *IEEE Trans. on Signal Processing*, Vol. 49, pp. 2893-2908, Dec. 2001.

[11] W. Lovejoy, “A survey of algorithmic methods for partially observed Markov decision processes”, *Annals of Operations Research*, 1991.

[12] R. Mahler, “Global Optimal Sensor Allocation”, *Proceedings of the Ninth National Symposium on Sensor Fusion*, vol. I, March 12-14, 1996, Naval Postgraduate School, Monterey CA, pp. 347-366.

[13] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, 1994.

[14] A. Rényi, “On measures of entropy and information”, *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics, and Probability*, volume 1, pp. 547-561, 1961.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, MIT Press, 1998.

[16] B. Van Roy, “Neuro Dynamic Programming: Overview and Recent Trends”, *Handbook of Markov Decision Procewsses: Methods and Applications*, 2001.

[17] C. Watkins, “Learning from Delayed Rewards”, Thesis, University of Cambridge, England 1989.