Simulation-Based Optimal Sensor Scheduling with Application to Observer Trajectory Planning

Sumeetpal Singh[a], Nikolaos Kantas[a], Ba-Ngu Vo[b], Arnaud Doucet[a] and Robin J. Evans[b]
[a]Signal Processing Group, Dept. of Eng., Univ. of Cambridge, UK
[b]Dept. of Elec. and Electronic Eng., Univ. of Melbourne, Australia

***Abstract*— Sensor scheduling has been a topic of interest to the target tracking community for some years now. Recently, research into it has enjoyed fresh impetus with the current importance and popularity of applications in Sensor Networks and Robotics. The sensor scheduling problem can be formulated as a controlled Hidden Markov Model. In this paper, we address precisely this problem and consider the case in which the state, observation and action spaces are continuous valued vectors. This general case is important as it is the natural framework for many applications. In sensor scheduling, our aim is to minimise the variance of the estimation error of the hidden state with respect to the action sequence. We present a novel simulation-based method that uses a stochastic gradient algorithm to find optimal actions. This is in contrast to existing works in the literature that only solve approximations to the original problem.**

I. Introduction

Consider the following continuous state Hidden Markov Model (HMM),

$$X_{k+1} = f(X_k, A_{k+1}, W_k) \in \mathbb{R}^{n_x},$$
$$Y_k = g(X_k, A_k, V_k) \in \mathbb{R}^{n_y},$$

where X_k is the hidden system state, Y_k the observation of the state, and W_k and V_k are i.i.d. noise terms. Unlike the classical HMM model, the evolution of the state and observation processes depends on an input parameter $A_k \in \mathbb{R}^{n_a}$, which is the control or action. In HMM models, one is primarily concerned with the problem of estimating the hidden state, which is achieved by propagating the posterior distribution (or filtering density) $\pi_k(x)dx = \mathbf{P}(X_k \in dx|A_{1:k}, Y_{1:k})$. By a judicious choice of control sequence $\{A_k\}$, the evolution of the state and observation processes can be 'steered' in order to yield filtering densities that give more accurate estimates of the state process. This problem is also known in the literature as the sensor scheduling problem.

Sensor scheduling has been a topic of interest to the target tracking community for the some years now [5], [10], [13], [14], [7]. The classical setting is the problem of tracking a maneuvering target over N epochs. Here X_k denotes the state of the target at epoch k, Y_k the observation provided by the sensor, and A_k some parameter of the sensor that may be adjusted to improve the "quality" of the observation. A measure of tracking performance is the variance of the tracking error over the N epochs:

$$\mathbf{E}\left\{ (\psi(X_k) - \langle \pi_k, \psi \rangle)^2 \right\}, \quad k = 1, \ldots, N, \tag{1}$$

where $\psi : \mathbb{R}^{n_x} \to \mathbb{R}$ is a suitable *test* function that emphasises the components of interest of the state vector we wish to track. The aim is to minimise the tracking error variance with respect to the choice of actions $\{A_1, \ldots, A_N\}$.

When the dynamics of the state and the observation processes are both linear and Gaussian then, the optimal solution to the sensor scheduling problem (1) (when ψ gives a quadratic cost) can be computed off-line; this is not surprising given that the Kalman filter covariance is also independent of the actual realisation of observations. In the general setting studied in this paper, the dynamics can be both non-linear and non-Gaussian, which means that the filtering density π_k, and integration with respect to it, cannot be evaluated in closed-form. Hence, the variance performance criterion itself does not admit a closed-form expression. To further complicate matters, the actions sought are continuous valued, i.e., vectors in $\mathbb{R}^{n_a}$.

To address the complications to do with the non-linear and non-Gaussian dynamics, one could linearise the state and observation model [10]. The majority of works [7], [11], [14] (and references therein), while aim at minimising the tracking error variance, do so approximately by minimising a lower bound to the variance criterion. The bound in question is the Posteriori Cramer-Rao Lower Bound (PCRLB), which is the inverse of the Fisher Information Matrix (FIM). This approach hinges on the ability to propagate recursively the FIM in closed form by a Ricatti-type equation for the non-linear and non-Gaussian filtering problem. Unfortunately, the recursion for the FIM involves evaluating the expectation of certain derivatives of the transition probability density of the state dynamics, as well as the expectation of certain derivatives of the observation likelihood (see (2) and (3) below). As these quantities cannot be evaluated in general except for the linear and Gaussian case, this assumption is either invoked or the authors resort to simulation-based approximations. In addition, the PCRLB bound is not always tight.

As for the complications due to continuous valued actions, the approach in the literature is to discretise $\mathbb{R}^{n_a}$ to a grid. There have also been studies where the continuous state HMM is approximated by a discrete state HMM, and the latter solved using Dynamic Programming [14].

The aim of this paper is to solve the sensor scheduling problem with continuous action space directly, and not a surrogate problem defined through the PCRLB or otherwise.

We make no assumptions of linearity or Gaussianity for analytic convenience, nor do we discretise the state, observation, or action space. We avoid these restrictive modelling assumptions on the continuous state HMM by recourse to methods based on computer simulation (simulation for short).

In order to solve for the optimal sequence of continuous-valued actions, we use an iterative stochastic gradient algorithm. We derive the gradient of the performance criterion with respect to the action trajectory and demonstrate how low variance estimates of it may be obtained using *control variate* [6] techniques. One major advantage of a stochastic gradient based method is that theoretical guarantees are easily obtained. Under suitable regularity assumptions, one can guarantee convergence to a local optimum of the performance criterion, while it is difficult to make similar assertions about the quality of the solutions obtained by other approximate methods proposed in the literature for sensor scheduling.

As an instance of the sensor scheduling problem, we study the observer trajectory planning problem for a bearings-only application. We state theoretical results concerning the convergence of the observer trajectory identified by our stochastic gradient algorithm. Handling multiple observers simultaneously is easy in our proposed framework and numerical results are presented for two observers tracking a maneuvering target.

Notation: The notation that is used in the paper is now outlined. The norm of a scalar, vector or matrix is denoted by $|\cdot|$. For a vector b, $|b|$ denotes the vector 2-norm $\sqrt{\sum_i |b(i)|^2}$. For a matrix A, $|A|$ denotes the matrix 2-norm, $\max_{b:|b|\neq 0} \frac{|Ab|}{|b|}$. For convenience, we also denote a vector $b \in R^n$ by $b = [b(i)]_{i=1,\ldots,n}$, or the i-th component of a vector by $[b]_i$. For scalars $a_{j,i}$, $j = 1,\ldots,m$, $i = 1,\ldots,n$, let $\left[[a_{j,i}]_{j=1,\ldots,m}\right]_{i=1,\ldots,n}$ denote the stacked vector $[a_{1,1},\ldots,a_{m,1},\ldots,a_{1,n},\ldots,a_{m,n}]^T$.) For a function $f : \mathbb{R}^n \to \mathbb{R}$ with arguments $z \in \mathbb{R}^n$, we denote $(\partial f/\partial z(i))(z)$ by $\nabla_{z(i)} f(z)$ and $\nabla f(z) = [\nabla_{z(1)} f(z), \cdots, \nabla_{z(n)} f(z)]^T$. For the vector valued function $F = [F_1, \cdots, F_n]^T : R^n \to R^n$, let ∇F denote the matrix $[\nabla F_1, \cdots, \nabla F_n]$. For a real-valued integrable functions f and g, let $\langle f, g\rangle$ denote $\int f(x)g(x)dx$.

II. Problem Formulation

At time k, let X_k and Y_k be random vectors that model the n_x-dimensional state and its n_y-dimensional observation respectively. Suppose that an action $A_k \in \mathbb{R}^{n_a}$ is applied at time k. The state $\{X_k\}_{k\geq 0}$ is an unobserved Markov process with initial distribution and transition law given by

$$X_0 \sim \pi_0, \quad X_{k+1} \sim p\left(\cdot|X_k, A_{k+1}\right). \tag{2}$$

(The symbol "$\sim$" means distributed according to.) The observation process $\{Y_k\}_{k\geq 1}$ is generated according to the state and action dependent probability density

$$Y_k \sim q\left(\cdot|X_k, A_k\right). \tag{3}$$

Given the sequence of actions $a_{1:k} := \{a_1, ..., a_k\}$ and measurements $y_{1:k} := \{y_1, ..., y_k\}$, the *filtering density* at time k is denoted by π_k, (or $\pi_k^{(y_{1:k}, a_{1:k})}$ to emphasise the dependence on $y_{1:k}$ and $a_{1:k}$,) and satisfies the *Bayes* recursion

$$\pi_k(x) = \frac{q\left(y_k|x, a_k\right) \int p(x|x', a_k)\pi_{k-1}(x')dx'}{\int\int q\left(y_k|x, a_k\right) p(x|x', a_k)\pi_{k-1}(x')dx'dx}. \tag{4}$$

Consider a suitable *test* function $\psi : \mathbb{R}^{n_x} \to \mathbb{R}$ where, for example, ψ could pick out a component of interest of the state vector we wish to estimate. The optimal sensor scheduling problem is to solve

$$\begin{gathered}\min_{A_{1:N}\in\Theta_A} J(A_{1:N}) = \\ \mathbf{E}_{(\pi_0, A_{1:N})}\left\{\sum_{k=1}^N \beta^{N-k}\left(\psi(X_k) - \langle \pi_k, \psi\rangle\right)^2\right\},\end{gathered} \tag{5}$$

where $\Theta_A \subset \left(\mathbb{R}^{n_a}\right)^N$ and for any $1 \leq k \leq N$ and integrable function $h : \left(\mathbb{R}^{n_x}\right)^k \times \left(\mathbb{R}^{n_a}\right)^k \times \left(\mathbb{R}^{n_y}\right)^k \to \mathbb{R}$,

$$\begin{aligned}\mathbf{E}_{(\pi_0, A_{1:k})}\left\{h(X_{1:k}, A_{1:k}, Y_{1:k})\right\} &:= \int h(x_{1:k}, A_{1:k}, y_{1:k}) \\ &\times \Pi_{i=1}^k q\left(y_i|x_i, A_i\right) p\left(x_i|x_{i-1}, A_i\right)\pi_0(x_0)dy_{1:k}dx_{0:k}.\end{aligned} \tag{6}$$

$\beta \in [0, 1]$ is a discount factor and its inclusion favours better tracking performance in the later epochs.

Feedback control: The sensor scheduling problem stated in (5) is an open-loop stochastic control problem. In order to utilise feedback, we will use the *open-loop feedback control* (OLFC) approach, which is described in detail in [1].

Path constraints: For the trajectory planning application, the sequence of actions $A_{1:N}^*$ determined by solving (5) may not be realisable due to motion constraints on the observer. For example, the sequence of possible accelerations in a linear motion model (see Section V) may not be able to realise the desired sequence of positions (actions) $A_{1:N}^*$. Alternatively, we may seek a sequence of positions that belong to some parametric class, e.g., an observer doing a constant velocity turn where the turn rate is to be determined. In some cases, we may summarise observer motion constraints through a bounded mapping

$$F : \left(\mathbb{R}^{n_u}\right)^N \to \left(\mathbb{R}^{n_a}\right)^N, \tag{7}$$

where the action $A_{1:N} = F(U_{1:N})$. For example, $U_{1:N}$ could describe the sequence of accelerations of an observer; see Section V for more details. We would then solve problem (5) subject to the equality constraint

$$A_{1:N} = F(U_{1:N}), \qquad U_{1:N} \in \left(\mathbb{R}^{n_u}\right)^N, \tag{8}$$

i.e. Θ_A now corresponds to the range of the function F.

Simulation and gradient based methods: We do not have a closed-form expression for J because the filtering density π_k and integration with respect to it cannot be evaluated in closed-form in our general setting. To evaluate $J(A_{1:N})$, we could revert to state-space discretisation. One could discretise $\mathbb{R}^{n_x}$, $\mathbb{R}^{n_y}$ and derive the corresponding state evolution and observation laws, i.e. (2) and (3), for the approximating discrete problem. We may then calculate the approximation to $J(A_{1:N})$ for any choice of actions. However this approach is computationally prohibitive and we would be limited to a coarse discretisation and a small horizon N at best. Also, it is not obvious how to choose the grid in $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$, since, for accuracy of the approximation, the grid should be finer in the regions where density in (6) has more mass. In [14], the HMM is discretised and a close-loop formulation of problem (5) is solved. A close-loop formulation of (5) is known as a Partially Observed Markov Decision Process (POMDP). However, solving a POMDP exactly is computationaly very demanding, specifically PSPACE-hard, and various approximation schemes that trade-off accuracy and speed have been devised.

We propose to use simulation with Stochastic Approximation (SA) to minimise $J(A_{1:N})$ when Θ_A is an open (i.e. continuous) set without resorting to discretising $\mathbb{R}^{n_x}$, $\mathbb{R}^{n_y}$ or Θ_A. SA is a gradient descent algorithm that only requires noisy estimates of the cost function gradient, i.e.,

$$A_{1:N,k+1} = A_{1:N,k} - \alpha_k \left(\nabla J(A_{1:N})|_{A_{1:N}=A_{1:N,k}} + \text{noise} \right), \tag{9}$$

where $\nabla J(A_{1:N})$ denotes the gradient of J w.r.t. $A_{1:N}$. The step-size α_k is a non-increasing positive sequence tending to zero. In Section III, we derive the gradient ∇J. Once again, we do not have a closed-form expression for ∇J for the same reasons as in J; the filtering density π_k and integration with respect to it cannot be evaluated in closed-form in our general setting. We will show instead how one may obtain a low variance estimate of ∇J, namely $\widehat{\nabla J}$. The noise in (9) arises precisely because we use $\widehat{\nabla J}$ instead of ∇J. One major advantage of a gradient based method is that theoretical guarantees are easily obtained. Under suitable assumptions on the noise in (9), one can guarantee that $A_{1:N,k}$ eventually converges to a local minimiser of J, while it is difficult to make similar assertions about the quality of the solution obtained by other approximate methods proposed in the literature for sensor scheduling.

III. The Cost Gradient and its Simulation-Based Approximation

In this section, we derive the gradient of the cost function (5) with respect to $A_{1:N}$. We then propose a suitable simulation-based approximation for optimising with SA. Because the problem (5) is solved for a fixed initial state distribution π_0, henceforth, we omit reference to π_0 in the notation for $\mathbf{E}_{(\pi_0, A_{1:N})}$ and denote the probability with respect to which this expectation is taken by $\mathbf{P}_{A_{1:N}}$.

Keeping in mind that $\left(\psi(X_k) - \langle \pi_k, \psi \rangle\right)^2$ is a function of the form $h(X_{1:k}, A_{1:k}, Y_{1:k})$, then (6) implies

$$\mathbf{E}_{A_{1:N}} \left\{ \left(\psi(X_k) - \langle \pi_k, \psi \rangle\right)^2 \right\} = \\ \mathbf{E}_{A_{1:k}} \left\{ \left(\psi(X_k) - \langle \pi_k, \psi \rangle\right)^2 \right\}.$$

Thus, $\nabla_{A_l} \mathbf{E}_{A_{1:N}} \left\{ \left(\psi(X_k) - \langle \pi_k, \psi \rangle\right)^2 \right\} = 0$ for $l > k$. For $l \leq k$, using (6),

$$\begin{aligned}
&\nabla_{A_l} \int \left(\psi(x_k) - \left\langle \pi_k^{(y_{1:k}, A_{1:k})}, \psi \right\rangle \right)^2 \\
&\times \Pi_{i=1}^k q\left(y_i|x_i, A_i\right) p\left(x_i|x_{i-1}, A_i\right) \pi_0(x_0) dx_{0:k} dy_{1:k} \\
&= \int \left(\psi(x_k) - \left\langle \pi_k^{(y_{1:k}, A_{1:k})}, \psi \right\rangle \right)^2 \\
&\times \nabla_{A_l} \left[\Pi_{i=1}^k q\left(y_i|x_i, A_i\right) p\left(x_i|x_{i-1}, A_i\right) \right] \pi_0(x_0) dx_{0:k} dy_{1:k} \\
&+ \int \nabla_{A_l} \left[\left(\psi(x_k) - \left\langle \pi_k^{(y_{1:k}, A_{1:k})}, \psi \right\rangle \right)^2 \right] \\
&\times \Pi_{i=1}^k q\left(y_i|x_i, A_i\right) p\left(x_i|x_{i-1}, A_i\right) \pi_0(x_0) dx_{0:k} dy_{1:k}.
\end{aligned}$$

The first term of the gradient is

$$\begin{aligned}
\mathbf{E}_{A_{1:N}} \{ \quad & \left(\psi(X_k) - \langle \pi_k, \psi \rangle\right)^2 \\
& \times \left[\tfrac{\nabla_{A_l} q(Y_l|X_l, A_l)}{q(Y_l|X_l, A_l)} + \tfrac{\nabla_{A_l} p(X_l|X_{l-1}, A_l)}{p(X_l|X_{l-1}, A_l)} \right] \}
\end{aligned}$$

The second term of the gradient is

$$\begin{aligned}
&\mathbf{E}_{A_{1:N}} \left\{ \nabla_{A_l} \left[\left(\psi(X_k) - \left\langle \pi_k^{(Y_{1:k}, A_{1:k})}, \psi \right\rangle \right)^2 \right] \right\} \\
&= -2 \mathbf{E}_{A_{1:N}} \{ \left(\psi(X_k) - \left\langle \pi_k^{(Y_{1:k}, A_{1:k})}, \psi \right\rangle \right) \\
&\times \nabla_{A_l} \left\langle \pi_k^{(Y_{1:k}, A_{1:k})}, \psi \right\rangle \} = 0,
\end{aligned}$$

where the last line follows from conditioning on $Y_{1:k}$. It follows from the above derivation that to obtain an unbiased estimator of $\nabla_{A_l} J(A_{1:N})$ for a given $A_{1:N}$, one samples a realisation of states and observations $(Y_{1:N}, X_{0:N}) \sim \mathbf{P}_{A_{1:N}}$ and forms the following estimate,

$$\begin{aligned}
\widehat{\nabla_{A_l} J}(A_{1:N}) &= \sum_{k=l}^{N} \beta^{N-k} \mathbf{E}_{A_{1:N}} \{ \left(\psi(X_k) - \langle \pi_k, \psi \rangle\right)^2 \\
&\times \left[\frac{\nabla_{A_l} q\left(Y_l|X_l, A_l\right)}{q\left(Y_l|X_l, A_l\right)} + \frac{\nabla_{A_l} p\left(X_l|X_{l-1}, A_l\right)}{p\left(X_l|X_{l-1}, A_l\right)} \right] |Y_{1:k} \},
\end{aligned} \tag{10}$$

where we have added the conditioning on $Y_{1:k}$ as it leads to a lower variance gradient estimate.[1]

We now describe how to implement the gradient estimate (10). In sensor scheduling applications concerning target tracking, the state process X_k is the state of the target to be tracked and often evolves independently of the action. Henceforth, we assume this independence for simplicity in presentation, i.e. $p\left(X_k|X_{k-1}\right)$, and remark that the work may also be extended to the more general case of state evolution and control dependence.[2] Now, define the real-valued function called the *score* [12],

$$S\left(y, x, a\right) := q\left(y|x, a\right)^{-1} \nabla_a q\left(y|x, a\right) \in \mathbb{R}^{n_a}. \tag{11}$$

To implement (10), we see that we require both the marginal π_k and the full posterior $\pi_{0:k}$ for all N epochs, i.e., for

[1] The variance is reduced since, for two jointly distributed random variables X and Y, $var(E(X|Y)) = var(X) - E(var(X|Y))$, and $E(var(X|Y)) > 0$.

[2] In methods that use the PCRLB [7], [11], [14], even after assuming $\{X_k\}$ evolves independently of $\{A_k\}$, one still needs to evaluate the expectation of derivatives of $\ln p\left(X_k|X_{k-1}\right)$ w.r.t. X_k and X_{k-1}, while this is not needed in (10).

$1 \leq k \leq N$. We propose to approximate these quantities using a mixture Dirac delta-masses,

$$\hat{\pi}_{0:k}(x_{0:k}) := \sum_{j=1}^{L} w_k^{(j)} \delta_{X_{0:k}^{(j)}}(x_{0:k}), \tag{12}$$

where $\delta_{X_{0:k}^{(j)}}$ denotes the Dirac delta-mass located at $X_{0:k}^{(j)}$ and the *importance weights* $\{w_k^{(j)}\}_{j=1}^{L}$ are non-negative scalars that sum to one. The approximation to π_k, namely $\hat{\pi}_k$, follows by marginalising $\hat{\pi}_{0:k}$. There are a number of ways to define such a point-mass approximation. For example, the simplest scheme would be to sample L independent state trajectory realisations $\left\{X_{0:N}^{(j)}\right\}_{j=1}^{L}$ from $\left(\Pi_{i=1}^{N} p\left(x_i|x_{i-1}\right)\right)\pi_0(x_0)$. The importance weights would then be

$$w_k^{(j)} := \{\sum_{j=1}^{L} \Pi_{i=1}^{k} q\left(Y_i|X_i^{(j)}, A_i\right)\}^{-1} \Pi_{i=1}^{k} q\left(Y_i|X_i^{(j)}, A_i\right). \tag{13}$$

For any integrable function h, $\int h(x_{0:k})\hat{\pi}_{0:k}(x_{0:k})dx_{0:k}$ converges to $\int h(x_{0:k})\pi_{0:k}(x_{0:k})dx_{0:k}$ as $L \to \infty$ (see [4, Ch. 2] for a precise statement of the mode of convergence). Practically though, we would prefer a small sample size L and this simple scheme of sampling from the state transition model can result in the majority of the importance weights $w_k^{(j)}$ being very small. There are number of remedies proposed for this in the Sequential Monte Carlo literature [4, Ch. 1.3.2]. For example, the importance sampling step can be designed to mininise the conditional variance of the importance weights by sampling $\left\{X_{0:N}^{(j)}\right\}_{j=1}^{L}$ from a Markov transition density that takes the observations into account, i.e., $X_{k+1}^{(j)} \sim k(x_{k+1}|X_k^{(j)}, Y_{k+1})$.

We emphasise that standard techniques from the Sequential Monte Carlo literature can be adopted in constructing an approximation of the form (12) to the full posterior, but we do not study this issue in detail here. To simplify the presentation, we will focus on the simple scheme of sampling from the state transition model.

Now, for a given $A_{1:N}$, one samples a realisation of states and observations $(X_{0:N}, Y_{1:N}) \sim \mathbf{P}_{A_{1:N}}$ and forms the following biased estimate of $\nabla_{A_l} J(A_{1:N})$ (10),

$$\begin{gathered} \textstyle\sum_{k=l}^{N} \beta^{N-k}\{\left\langle \hat{\pi}_{0:k}, \psi_k^2(\cdot) S\left(Y_l, \cdot, A_l\right)\right\rangle \\ + \left\langle \hat{\pi}_k, \psi_k \right\rangle^2 \left\langle \hat{\pi}_{0:k}, S\left(Y_l, \cdot, A_l\right)\right\rangle \\ -2\left\langle \hat{\pi}_k, \psi\right\rangle \left\langle \hat{\pi}_{0:k}, \psi_k(\cdot) S\left(Y_l, \cdot, A_l\right)\right\rangle\}. \end{gathered} \tag{14}$$

Implementing the SA recursion (9) with the gradient estimate (14) is straightfoward. However, to prove its convergence, we would not be able to use standard SA results. Even though (14) is a noisy estimate of $\nabla_{A_l} J(A_{1:N})$, the noise is not zero-mean due to the bias of the simulation-based approximation to π_k and $\pi_{0:k}$. To assert convergence of (9) to a minima of J, we would have to gradually increase the number of samples L to remove the bias. This approach is infeasible in practice as the computational complexity of the SA recursion increases with each iteration. Similar conditions are required for convergence of SA driven by sample averages [12].

Henceforth, we fix the set of L state trajectory samples $\left\{X_{0:N}^{(j)}\right\}_{j=1}^{L}$ that is used to approximate $\pi_{0:k}$ for all k and form the following approximation to J[3],

$$\hat{J}(A_{1:N}) = \sum_{k=1}^{N} \beta^{N-k} \mathbf{E}_{(\pi_0, A_{1:N})} \left\{ \left\langle \hat{\pi}_k, \psi^2 \right\rangle - \left\langle \hat{\pi}_k, \psi \right\rangle^2 \right\}. \tag{15}$$

We will then derive an unbiased estimate of the gradient of $\hat{J}$ in a similar manner to J above and minimise $\hat{J}$ via SA. This approach is easier to analyse and we show that, under suitable assumptions, SA converges to a local minimum of $\hat{J}$ almost surely. Note that since the error in the approximation $\hat{\pi}_{0:k}$ diminishes as the sample size L increases, $\hat{J}$ will be a good approximation to J for sufficiently large L.

In the same way as gradient of J was derived in (10), we have

$$\nabla_{A_l} \hat{J}(A_{1:N}) =$$

$$\sum_{k=1}^{N} \beta^{N-k} \nabla_{A_l} \mathbf{E}_{A_{1:N}} \left\{ \left\langle \hat{\pi}_k, \psi^2 \right\rangle - \left\langle \hat{\pi}_k, \psi \right\rangle^2 \right\} =$$

$$\mathbf{E}_{A_{1:N}}\{\sum_{k=l}^{N} \beta^{N-k}(\left\langle \hat{\pi}_k, \psi^2 \right\rangle - \left\langle \hat{\pi}_k, \psi \right\rangle^2) S\left(Y_l, X_l, A_l\right)\} + \tag{16}$$

$$\mathbf{E}_{A_{1:N}}\{\sum_{k=l}^{N} \beta^{N-k}(\nabla_{A_l}\left\langle \hat{\pi}_k, \psi^2 \right\rangle - 2\left\langle \hat{\pi}_k, \psi \right\rangle \nabla_{A_l}\left\langle \hat{\pi}_k, \psi \right\rangle)\}, \tag{17}$$

where[4]

$$\begin{aligned} \nabla_{A_l}\left\langle \hat{\pi}_k, \psi \right\rangle &= \left\langle \hat{\pi}_{0:k}, \psi S\left(Y_l, \cdot, A_l\right)\right\rangle \\ &\quad - \left\langle \hat{\pi}_k, \psi \right\rangle \left\langle \hat{\pi}_{0:k}, S\left(Y_l, \cdot, A_l\right)\right\rangle. \end{aligned} \tag{18}$$

It is now straightforward to obtain a simulation-based approximation of $\nabla \hat{J}(A_{1:N})$. For a given $A_{1:N}$, one samples a realisation of states and observations $(Y_{1:N}, X_{0:N}) \sim \mathbf{P}_{A_{1:N}}$ and forms the following unbiased estimate of $\nabla_{A_l} \hat{J}(A_{1:N})$: For $l = 1, \ldots, N$

$$S\left(Y_l, X_l, A_l\right) \sum_{k=l}^{N} \beta^{N-k}\left(\left\langle \hat{\pi}_k, \psi^2 \right\rangle - \left\langle \hat{\pi}_k, \psi \right\rangle^2\right)$$

$$+ \sum_{k=l}^{N} \beta^{N-k}\left(\nabla_{A_l}\left\langle \hat{\pi}_k, \psi^2 \right\rangle - 2\left\langle \hat{\pi}_k, \psi \right\rangle \nabla_{A_l}\left\langle \hat{\pi}_k, \psi \right\rangle\right). \tag{19}$$

A. Variance Reduction by Control Variates

In implementation, we found that the variance of the gradient approximation (19) (or (14)) was quite large. This is because we are approximating high dimensional integrals

[3] Note that by a conditioning argument, $J(A_{1:N})$ can be written as $\sum_{k=1}^{N} \beta^{N-k} \mathbf{E}_{A_{1:N}} \left\{ \left\langle \pi_k, \psi^2 \right\rangle - \left\langle \pi_k, \psi \right\rangle^2 \right\}$.

[4] It is possible to compute $\nabla_{A_l}\left\langle \hat{\pi}_k, \psi \right\rangle$ when re-sampling is employed in $\hat{\pi}_k$; resampling is commonly used in the Sequential Monte Carlo literature to yield approximations to $\hat{\pi}_k$

using simulation and more so, with moderate sample sizes. Naturally, it would be possible to reduce the variance by simply increasing the number of samples. We do not wish to do so, as our aim is to extract the most accurate estimates of the quantities of interest for a given set of samples. Control Variates are widely used to reduce the variance in simulation-based approximations by only collecting additional statistics from the samples. We now describe how control variates may be implemented for our problem.

For a random variable W, consider the problem of estimating $\mathbf{E}(W)$ where we have access to a zero-mean random variable Z correlated with W. Rather than using a realisation of W as an unbiased estimate, we use $W - bZ$ where b is a constant. The estimator $W - bZ$ is also unbiased. Furthermore, the function of b

$$\mathbf{var}(W - bZ) = \mathbf{var}(W) - 2b\mathbf{cov}(W, Z) + b^2\mathbf{var}(Z) \quad (20)$$

is convex and is minimised at $b^* = \mathbf{cov}(W, Z)/\mathbf{var}(Z)$, which implies the variance of the estimate $W - b^*Z$ of $\mathbf{E}(W)$ is less than the variance of the estimate W. The random variable Z is referred to as the control variate (CV) and we call b the control variate constant [6]. In the context of the gradient estimate in (19), we found in implementation that reducing the variance of the estimate of (16) was sufficient.

The score in (16) is zero-mean, i.e.

$$\mathbf{E}_{(\pi_0, A_{1:N})}\left\{S\left(Y_l, X_l, A_l\right)\right\} = 0,$$

and we use it as the CV. Doing so yields the following unbiased estimator of $\nabla_{A_l}\hat{J}$ instead of (19),

$$\begin{aligned}&\text{diag}\left(S\left(Y_l, X_l, A_l\right)\right)\Big(\sum_{k=l}^{N}\beta^{N-k}(\left\langle\hat{\pi}_k, \psi^2\right\rangle - \left\langle\hat{\pi}_k, \psi\right\rangle^2)\mathbf{1} - b_l\Big)\\ &+ \sum_{k=l}^{N}\beta^{N-k}\left(\nabla_{A_l}\left\langle\hat{\pi}_k, \psi^2\right\rangle - 2\left\langle\hat{\pi}_k, \psi\right\rangle\nabla_{A_l}\left\langle\hat{\pi}_k, \psi\right\rangle\right),\end{aligned} \quad (21)$$

where $\mathbf{1} \in \mathbb{R}^{n_a}$ and the CV constant (vector) $b_l \in \mathbb{R}^{n_a}$ is to be determined in order to minimise the variance of the estimate. Noting that the optimal CV constant is a solution of the minimisation problem (20), we may employ the following SA algorithm to converge to it,

$$\begin{aligned}b_l \longleftarrow \quad b_l \quad &-\beta\text{diag}\left(S\left(Y_l, X_l, A_l\right)\right)\left(\text{diag}\left(S\left(Y_l, X_l, A_l\right)\right)b_l\right.\\ &\left.-\sum_{k=l}^{N}\beta^{N-k}(\left\langle\hat{\pi}_k, \psi^2\right\rangle - \left\langle\hat{\pi}_k, \psi\right\rangle^2)\mathbf{1}\right),\end{aligned} \quad (22)$$

where β is the step-size. Under suitable assumptions stated in Section IV, we will have b_l converging to

$$\begin{aligned}&\mathbf{E}_{(\pi_0, A_{1:N})}\left\{\text{diag}\left(S\left(Y_l, X_l, A_l\right)\right)^2\right\}^{-1}\\ &\times\mathbf{E}_{(\pi_0, A_{1:N})}\{\text{diag}\left(S\left(Y_l, X_l, A_l\right)\right)^2\\ &\times\textstyle\sum_{k=l}^{N}\beta^{N-k}\left(\left\langle\hat{\pi}_k, \psi^2\right\rangle - \left\langle\hat{\pi}_k, \psi\right\rangle^2\right)\mathbf{1}\}.\end{aligned} \quad (23)$$

The same approach applies when minimising the variance of the gradient estimate (14) with control variates.

IV. Solving the Sensor Scheduling Problem with Stochastic Approximation

We now state the simulation-based algorithm that will be used to solve the sensor scheduling problem. It is a two time-scale SA algorithm to minimise $\hat{J}$ using the reduced variance estimate of $\nabla\hat{J}$ given by (21) and (22). We do so for the (general) case with action path constraints as specified in (8). Note that although we could also apply algorithm (27)-(30) below with the gradient estimate (14), we minimise $\hat{J}$ instead as its convergence is easier to study; see comments immediately following (14).

We introduce the following functions to make the presentation of the main algorithm concise. For each $A_{1:N}$, define the functions $h_{i,A_{1:N}} : \left(\mathbb{R}^{n_x}\right)^{N+1} \times \left(\mathbb{R}^{n_y}\right)^{N} \to \left(\mathbb{R}^{n_a}\right)^{N}$, $i = 1, 2$, as follows:

$$\begin{aligned}&h_{1,A_{1:N}}(X_{0:N}, Y_{1:N}) :=\\ &[S\left(Y_l, X_l, A_l\right)\sum_{k=l}^{N}\beta^{N-k}\left(\left\langle\hat{\pi}_k, \psi^2\right\rangle - \left\langle\hat{\pi}_k, \psi\right\rangle^2\right)]_{l=1,\ldots,N},\end{aligned} \quad (24)$$

$$\begin{aligned}&h_{2,A_{1:N}}(X_{0:N}, Y_{1:N}) :=\\ &[\sum_{k=l}^{N}\beta^{N-k}\left(\nabla_{A_l}\left\langle\hat{\pi}_k, \psi^2\right\rangle - 2\left\langle\hat{\pi}_k, \psi\right\rangle\nabla_{A_l}\left\langle\hat{\pi}_k, \psi\right\rangle\right)]_{l=1,\ldots,N}.\end{aligned} \quad (25)$$

Note that

$$\begin{aligned}\nabla\hat{J}(A_{1:N}) = \mathbf{E}_{A_{1:N}}\{&h_{1,A_{1:N}}(X_{0:N}, Y_{1:N})\\ &+h_{2,A_{1:N}}(X_{0:N}, Y_{1:N})\}, \qquad \in \left(\mathbb{R}^{n_a}\right)^{N}.\end{aligned}$$

For technical reasons concerning the convergence of the two time-scale SA algorithm below, we introduce the positive scalar valued function $\Gamma : \left(\mathbb{R}^{n_a}\right)^{N} \to (0, \infty)$,

$$\Gamma(b) := (1 + |b|)^{-1}C, \quad (26)$$

where C is a positive constant. The function Γ is needed to ensure that the CV constants remain bounded almost surely (details in [13]).

The two time-scale SA algorithm for solving the sensor scheduling problem: For conciseness, let

$$\theta = U_{1:N}, \tilde{\theta} = A_{1:N} \quad \left(= F(\theta)\right), \omega = (X_{0:N}, Y_{1:N}).$$

$$\begin{aligned}\theta_{k+1} =&\theta_k - \alpha_{k+1}\Gamma(b_k)\nabla F(\theta_k)\\ &\times\left(h_{1,\tilde{\theta}_k}(\omega_{k+1}) + h_{2,\tilde{\theta}_k}(\omega_{k+1}) - S_{\tilde{\theta}_k}(\omega_{k+1})b_k\right),\end{aligned} \quad (27)$$

$$\begin{aligned}b_{k+1} = b_k &- \beta_{k+1}S^2_{\tilde{\theta}_k}(\omega_{k+1})b_k\\ &+ \beta_{k+1}S_{\tilde{\theta}_k}(\omega_{k+1})h_{1,\tilde{\theta}_k}(\omega_{k+1}),\end{aligned} \quad (28)$$

$$\omega_{k+1} \sim \mathbf{P}_{\tilde{\theta}_k}, \quad (29)$$

$$\tilde{\theta}_k = F(\theta_k), \qquad k \geq 0, \quad (30)$$

where

$$\begin{aligned}&S_{A_{1:N,k}}(X_{0:N,k+1}, Y_{1:N,k+1}) =\\ &\text{diag}\left([S\left(Y_{l,k+1}, X_{l,k+1}, A_{l,k}\right)]_{l=1,\ldots,N}\right).\end{aligned} \quad (31)$$

(Note that $U_{1:N,k} = \theta_k$, $A_{1:N,k} = \tilde{\theta}_k$, $(X_{0:N,k+1}, Y_{1:N,k+1}) = \omega_{k+1}$.)

Assumption 1: The step-size sequences $\{\alpha_k\}$ and $\{\beta_k\}$ are non-negative, sum to infinity, are squared summable, and for some $p > 0$ satisfy $\sum_k \left(\frac{\alpha_k}{\beta_k}\right)^p < \infty$

Typically, the step-sizes are

$$\alpha_k = k^{-\alpha}, \qquad \beta_k = k^{-\beta}, \tag{32}$$

where $\alpha > \beta > 0.5$. Thus, $\sum_k \left(\frac{\alpha_k}{\beta_k}\right)^p < \infty$ may only be satisfied for a large positive p. Since α_k tends to zero more quickly than β_k, the recursion for the action (27) is said to evolve on a slower time-scale than that for the CV constants (28). By having $U_{1:N,k}$ evolve more slowly than b_k, we allow b_k to 'track' the optimal CV constants, which depend on the point at which the gradient $\nabla \hat{J}$ is evaluated (see (23)). In [13] we establish the convergence of algorithm (27)-(30) for the choice of step-sizes in Assumption 1. However, in the numerical example presented in Section VI, we set function $\Gamma(b) = 1$, use constant step-sizes $\alpha_k = \alpha$ and $\beta_k = \beta$ and still demonstrate convergence. For SA in general, decreasing step-sizes are essential for almost sure convergence. If fixed step-sizes are used, then we may still have convergence but now the iterates "oscillate" about their limiting values with variance proportional to the step-size.

The convergence of a two time-scale SA algorithm related to (27)-(30) was studied in [8]. We may write the slow time-scale process in a more general form than (27) as,

$$\theta_{k+1} = \theta_k + \alpha_{k+1} H_{k+1}.$$

If the parameter θ_k does not change, say $\theta_k = \theta$ for all k, the process $\{b_k\}$ would converge to some $\bar{b}(\theta)$. When θ_k varies slowly, we would like the process $\{b_k\}$ to track $\bar{b}(\theta_k)$. Under certain regularity assumptions on the process $\{H_k\}$ [8], it can be shown that this would also be the case when θ_k did change. As for the convergence of $\{\theta_k\}$, we may use the line of proof in [2] to show $\liminf_k \left|\nabla(\hat{J} \circ F)(\theta_k)\right| = 0$. Details are in the [13].

V. Application to Observer Trajectory Planning

In observer trajectory planning, we wish to track a maneuvering target for N epochs. At epoch k, let X_k denote the state of the target, A_k the position of the observer and Y_k the partial observation of the target state, i.e., $Y_k = g(X_k, A_k, V_k)$, where V_k denotes noise. Typically, the observer has its own motion model and we let X_k^o denote state of the observer. The observer state descriptor usually includes its position and therefore A_k corresponds to certain components of X_k^o. The aim of OTP is to adaptively maneuver the observer to optimise the tracking performance the target.

We do not need to specify the target model explicitly. Our only concern is that we can sample from the model. Maneuvering targets models such as the *jump Markov linear system* (JMLS) [3] may be used. In Section VI, we consider a maneuvering target in the examples. However we need to be more specific about the observer. As indicated in (8), we require an observer model of the form $A_{1:N} = F(U_{1:N})$ where we exert control on the observer positions $A_{1:N}$ through the variables $U_{1:N}$. For instance, the accelerations of the observer could be determined from the input $U_{1:N}$, which will in turn determine the observer trajectory. This is precisely the model we adopt for the examples in Section VI. Let the state of the observer be $X_k^o = \left[r_{x,k}^o, v_{x,k}^o, r_{y,k}^o, v_{y,k}^o\right]^T$, with $A_k = \left[r_{x,k}^o, r_{y,k}^o\right]^T$. For example, we could assume a kinematic model for the evolution of the state,

$$X_{k+1}^o = G X_k^o + H \times c_1 \times \arctan(c_2 U_{k+1}) \tag{33}$$

where matrices G and H follow the standard definition [10]. As in the standard kinematic model, the initial state X_0^o is fixed and $U_{k+1} \in \mathbb{R}^2$ determines the acceleration in the x and y directions. The difference here is the inclusion of function arctan and constants c_1 and c_2. Function $\arctan$ and its first two derivatives are bounded. Also, $\arctan$ is linear about zero, which makes a nice choice of saturating function for the acceleration; naturally the acceleration cannot be unbounded. The constants c_1, c_2 alter the saturation behaviour of the acceleration. The observer trajectory is completely determined once X_0^o and $U_{1:N}$ are given. The function F in (7) is now implicitly defined by (33).

In the bearings-only model, the observation process $\{Y_k\}_{k \geq 0}$ $(\subset \mathbb{R})$ is generated according to

$$Y_k = \arctan\left(\frac{r_{x,k} - A_k(1)}{r_{y,k} - A_k(2)}\right) + V_k, \tag{34}$$

where $V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_Y^2)$. In our simulation-based framework, we require that the observation process density is known and is differentiable w.r.t. A_k. The bearings-only case is one such example. At this point we will assume that the x and y position of the target correspond to the first and third component of the state descriptor X_k, i.e. $X_k = [r_{x,k}, \cdot, r_{y,k}, \cdots]^T$, which is usual convention in the literature. See convergence results in Proposition 2 and 3 below.

For the case of multiple observers, say p of them, let the position of observer l at epoch k be denoted by $A_k^{(l)}$. Also, assume that each observer measures a bearings angle according to (34) independently of the other observers, i.e., observer l receives the measurement $Y_k^{(l)}$ generated according to the model (34) based on its own position $A_k^{(l)}$. We stack these observations together as a vector-valued observation $Y_k = [Y_k^{(1)}, \ldots, Y_k^{(p)}]^T$. Likewise, we stack the positions to form the 'effective' position $A_k = \left[\left(A_k^{(1)}\right)^T, \cdots, \left(A_k^{(p)}\right)^T\right]^T$. The observation density for this multiple observer case is

$$q\left(Y_k | X_k, A_k\right) = q(Y_k^{(1)} | X_k, A_k^{(1)}) \times \cdots \times q(Y_k^{(p)} | X_k, A_k^{(p)}).$$

It is apparent that we are now effectively in the original single observer setting and may proceed to solve the multiple OTP problem as above.

A. Convergence For Bearings-Only Tracking

For the bearings-only observation model, we have the following sufficient conditions for the convergence of the slow and fast time-scale. The sufficient conditions (35)-(38) are only restrictions on the target state transition model and the range of the function F that is used to map the sequence $U_{1:N}$ (which could be accelerations) to the observer positions for all N epochs.

Proposition 2: Consider algorithm (27)-(30) for the bearings-only observation model (34). Suppose the following assumptions hold:

$$\sup_{A_{1:N}\in\text{range}(F)} \mathbf{E}_{A_{1:N}}\left\{\left|\frac{1}{X_i(3)-A_i(2)}\right|^p\right\} < \infty, \qquad 1\le i\le N, \quad p>0, \tag{35}$$

$$\sup_{A_{1:N}\in\text{range}(F)} \max_{l}\left|\frac{1}{X_i^{(l)}(3)-A_i(2)}\right| < \infty, \qquad 1\le i\le N, \tag{36}$$

$$\inf_{1\le i\le N}\inf_{A_{1:N}\in\text{range}(F)} \mathbf{E}_{A_{1:N}}\left\{\sigma_Y^{-2}\frac{\frac{1}{(X_i(3)-A_i(2))^2}}{\left[1+\left(\frac{X_i(1)-A_i(1)}{X_i(3)-A_i(2)}\right)^2\right]^2}\right\} > 0, \tag{37}$$

$$\inf_{1\le i\le N}\inf_{A_{1:N}\in\text{range}(F)} \mathbf{E}_{A_{1:N}}\left\{\sigma_Y^{-2}\frac{\left[\frac{X_i(1)-A_i(1)}{(X_i(3)-A_i(2))^2}\right]^2}{\left[1+\left(\frac{X_i(1)-A_i(1)}{X_i(3)-A_i(2)}\right)^2\right]^2}\right\} > 0. \tag{38}$$

Then, almost surely, $\sup_k |b_k| < \infty$ and $\lim_k \left|b_k - \overline{S^2}(A_{1:N,k})^{-1}\overline{S\times h_1}(A_{1:N,k})\right| = 0$. Furthermore, if F has bounded second order derivatives then, $\liminf_k \left|\nabla(\hat{J}\circ F)(U_{1:N,k})\right| = 0$.

Proof: See [13]. ■

Note that the expectation operator $\mathbf{E}_{A_{1:N}}\{\cdot\}$ above is an abbreviation for $\mathbf{E}_{(X_{0:N},Y_{1:N})\sim\mathbf{P}_{A_{1:N}}}\{\cdot\}$. For the function F defined in (33), the last assumption concerning bounded second order derivatives hold. Condition (36) relates to the samples used to approximate the posterior density in (12)-(13). Also, the first and third component of the target state is its x and y coordinate respectively.

The next result gives the conditions under which assumptions (35)-(38) hold. This result basically says that if the support of $X_{0:N}$ and the range of function F do not intersect, then the assumptions hold and we have the desired convergence of two time-scale SA for OTP. It is interesting to note that the scenario in which the support of $X_{0:N}$ and the range of F do not intersect is a standard setting studied by previous works on OTP for bearings-only observations (see references in the Introduction), and hence the conditions of Proposition 2 are not restrictive for the application.

Proposition 3: Write the mapping $F : R^{2N} \rightarrow R^{2N}$ as $F = [F_{1,1}, F_{1,2}, \cdots, F_{N,1}, F_{N,2}]^T$. (Note that $A_i(j) = F_{i,j}(U_{1:N})$.) Suppose that the density of $X_{0:N}$, $f(x_{0:N})$, has a compact support $\mathcal{K}_f \subset R^{4(N+1)}$. Furthermore, suppose that for each $1 \le i \le N$, the compact set $\mathcal{K}_{f,i} := \{x_i(3) | x_{0:N} \in \mathcal{K}_f\}$ does not intersect with the closure of the set range$(F_{i,2})$, i.e., there exists a compact set $\mathcal{K}_{A,i}$ such that range$(F_{i,2}) \subset \mathcal{K}_{A,i}$, and $\mathcal{K}_{f,i} \cap \mathcal{K}_{A,i} = \varnothing$. Then, conditions (35)-(38) are satisfied.

Proof: See [13]. ■

VI. Numerical Example

In all examples below, $\psi(X_k) = w_1 X_k(1) + w_2 X_k(3)$. Weights $w_1, w_2 \in [0,1]$ are selected to trade-off accuracy in tracking the x- and y- coordinates. We solve for the optimal open-loop observer trajectory under a variety of tracking scenarios, namely, with a fast observer, a slow observer and two cooperating observers. Open loop feedback control is implemented for the two observers case. All examples consider a maneuvering target.

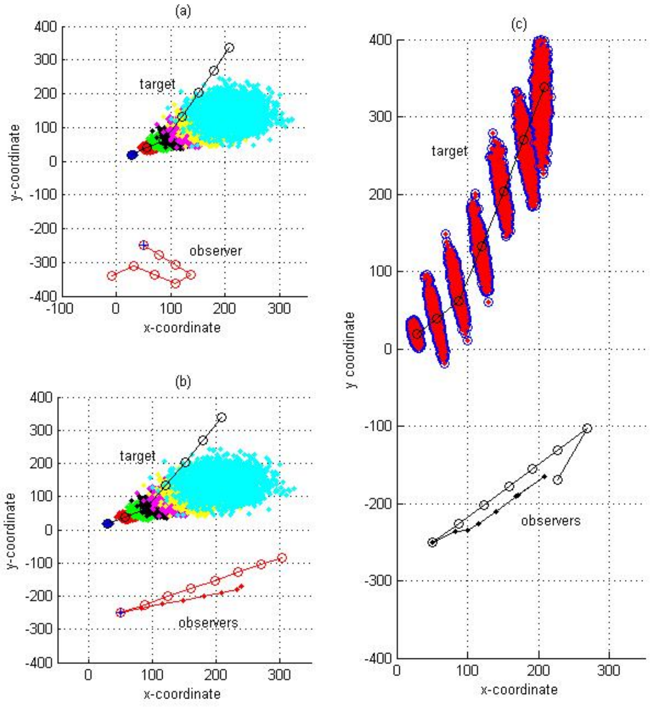

Fig. 1. (a) One observer open loop trajectory with target, (b) Two observers open loop trajectory with target, (c) Two observers OLFC trajectory with target and particle filter. Fast observers commence from $(50, -250)$. Target starts at $(0, 0)$ and moves in the north-west direction.

Fast Observers: The setting for this example is a maneuvering target that is to be tracked by a single fast observer and two cooperating fast observers. The term fast is to be understood in the sense that the observer in the subsequent example is significantly more constrained in its motion. The observer motion model is given in (33), with a fast or slow observer defined by setting constant c_1. In Figure 1(a) the optimal open-loop trajectory of the observer is plotted for a horizon 7 problem. The maneuvering target trajectory is also shown. The cloud of particles surrounding the maneuvering target are target trajectory samples drawn from the target dynamic model (2). Note that the target maneuver was intentionally chosen to be far more drastic than is predicted by its model. This was done to contrast the constructed open-loop and open loop feedback control trajectories. In Figure 1(b), we show the difference in the optimal open-loop trajectory obtained when there are two fast observers. Figure 1(c) shows the open-loop feedback trajectory obtained for the same two fast observers. The cloud of particles shown is now the particle filter tracking the maneuvering target. We note that the OLFC trajectory performs more maneuvers than the equivalent open loop one.

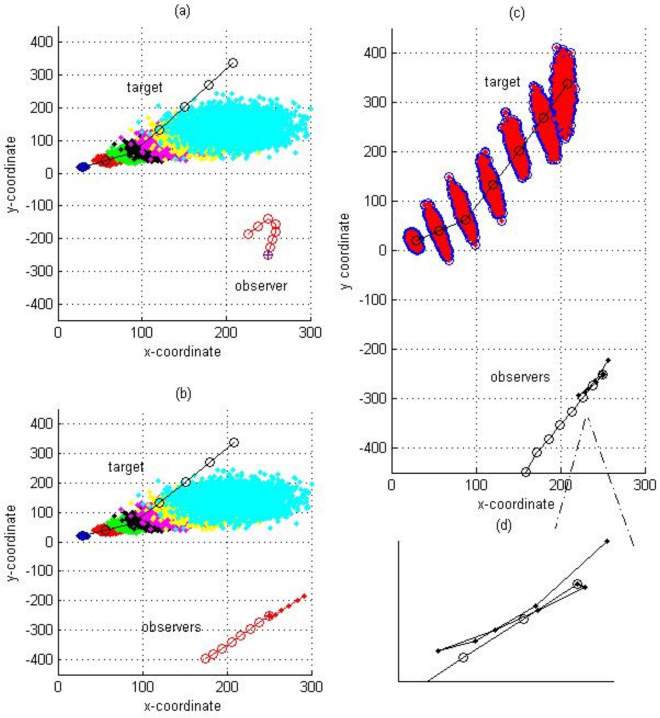

Fig. 2. (a) One observer open loop trajectory with target, (b) Two observers open loop trajectory with target, (c) Two observers OLFC trajectory with target and bootstrap filter, (d) Magnification of the OLFC trajectory of one of the observers. Slow observers commence from coordinate $(250, -250)$.

Slow Observers: We now contrast the optimal trajectory of the fast observers above with that of slow observers. Figure 2(a) shows the optimal open-loop trajectory of one slow observer, and Figure 2(b) that of two cooperating slow observers. Note that a single slow observer is obliged to do more maneuvers to improve the tracking performance since it is significantly more constrained in motion. Figure 2(c) shows the open-loop feedback trajectory obtained. Note that the two observers maneuver in Figure 2(c) much more than in Figure 2(b) as they are responding to the target maneuver.

VII. Conclusion

In this paper, we proposed a novel simulation-based method to solve the sensor scheduling problem for the case in which the state, observation and action spaces are continuous valued vectors. This is in contrast to existing works in the literature that only solve approximations to the original problem, e.g. via the PCRLB. This general continuous state-space case is important as it is the natural framework for many applications, like observer trajectory planning. We presented a novel simulation-based method that used two timescale stochastic approximation to find optimal actions and stated convergence results for the bearings-only tracking problem. It is possible to extend the method to sensor scheduling based on an information criterion like the Kullback-Leibler criterion [9]

References

[1] D.P. Bertsekas, *Dynamic programming and optimal control*. Belmont: Athena Scientific, 1995.

[2] D. P. Bertsekas and J. N. Tsitsiklis, “Gradient Convergence in Gradient Methods with Errors,” *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 627-642, 2000.

[3] A. Doucet, N.J. Gordon and V. Krishnamurthy, “Particle filters for state estimation of jump Markov linear systems,” *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, Mar. 2001.

[4] A. Doucet, J.F.G. de Freitas and N.J. Gordon *Sequential Monte Carlo methods in practice*. New York: Springer, 2001.

[5] R.J. Evans, V. Krishnamurthy and G. Nair, “Sensor adaptive target tracking over variable bandwidth networks,” in *Model Identification and Adaptive Control*, G.C. Goodwin (Ed.), Springer-Verlag, London, pp. 115-124, 2001.

[6] P.W. Glynn and R. Szechtman, “Some new perspectives on the method of control variates,” in *Monte Carlo and Quasi-Monte Carlo Methods 2000*, K.T. Fang, F.J. Hickernell and H. Niederreiter, Eds., Springer-Verlag, Berlin, pp. 27–49, 2002.

[7] M.L. Hernandez, T. Kirubarajan, Y. Bar-Shalom, “Multisensor resource deployment using posterior Cramer-Rao bounds,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, April, 2004

[8] V. R. Konda and J. N. Tsitsiklis, “Linear Stochastic Approximation Driven by Slowly Varying Markov Chains”, Systems and Control Letters, Vol. 50, No. 2, 2003, pp. 95-102.

[9] A. Logothetis and A. Isaksson “On sensor scheduling via information theoretic criteria,” in *IEEE Conf. Decision Control*, 1999, pp. 2402–2406.

[10] A. Logothetis, A. Isaksson and R.J. Evans “An information theoretic approach to observer path design for bearings-only tracking,” in *IEEE Conf. Decision Control*, 1997, pp. 3132–3137.

[11] S. Paris, J.-P. Le Cadre, “Planification for terrain-aided navigation,” in *Proceedings of the 5th International Conference on Information Fusion*, pp. 1007-1014, Annapolis, Maryland, 2002.

[12] G.Ch. Pflug, *Optimization of stochastic models: the interface between simulation and optimization*. Boston: Kluwer, 1996.

[13] S. Singh, N. Kantas, B. Vo, A. Doucet and R. Evans, “On the convergence of a stochastic optimisation algorithm for optimal observer trajectory planning,” Cambridge University Technical Report, 2005.

[14] O. Tremois and J.-P. LeCadre, “Optimal observer trajectory in bearings-only tracking for manoeuvring sources,” *IEE Proc. Radar, Sonar Navig.*, vol. 146, no. 1, pp. 31–39, Feb. 1999.

[15] H. W.J. Lee, K.L. Teo and E.B. Lim, “Sensor schedulling in continuous time,” *Automatica*, vol. 37, pp. 2017-2023, 2001.