Information Space Receding Horizon Control for Multi-Agent Systems

Z. Sunberg, S. Chakravorty, R. Erwin

***Abstract*—In this paper, we present a receding horizon solution to the problem of optimal scheduling for multiple sensors monitoring a group of dynamical targets. The term 'target' is used here in the classic sense of being the object that is being sensed or observed by the sensors. This problem is motivated by the Space Situational Awareness (SSA) problem. The multi-sensor optimal scheduling problem can be posed as a multi-agent Partially Observed Markov Decision Process (POMDP) whose solution is given by an Information Space (I-space) Dynamic Programming (DP) problem. We present a simulation based stochastic optimization technique that exploits the structure inherent in the problem to obtain variance reduction along with a distributed solution. This stochastic optimization technique is combined with a receding horizon approach which obviates the need to solve the computationally intractable multi-agent I-space DP problem and hence, makes the technique computationally tractable for such problems. The technique is tested on a simple numerical example which is nonetheless computationally intractable for existing solution techniques.**

I. Introduction

In this paper, we consider the problem of optimal scheduling for multiple sensors such that the information gained by the sensors is maximized. The class of problems that is considered is motivated by the so-called Space Situational Awareness (SSA) problem. It is easily shown that the scheduling problem, in general, may be posed as a Partially Observed Markov Decision Problem (POMDP) whose solution is given by an information space (I-space) Dynamic Programming (DP) problem. In the case of multiple agents, the resulting problem is a multiple agent I-space DP problem that is impossible to solve computationally owing to the exponential complexity of the problem in terms of the number of agents and the resulting exponential explosion of the state and control spaces. We propose a generalization of an I-space receding horizon control (I-space RHC: IS-RHC) approach that we had proposed to the single sensor problem in previous work [1], to the case of multiple agents. The solution strategy is termed the I-space RHC multi-agent technique (I-RHC-M). The online stochastic optimization problems that result from the receding horizon approach are solved using a simulation based gradient ascent technique. The underlying structure of the problem allows us to drastically reduce the variance of the gradient estimates while allowing for a distributed implementation of the gradient ascent technique. The technique is tested on a simple example to show the efficacy of the method.

Z. Sunberg is a Graduate Student Researcher in the Department of Aerospace Engineering, Texas A&M University, College Station, TX

Suman Chakravorty is an Associate Professor, Department of Aerospace Engineering, Texas A&M University, College Station, schakrav@aero.tamu.edu

R. Erwin is a Principal Research Scientist, Space Vehicles Directortate, Air Force Research Laboratory, Albuquerque

The optimal sensing problem has its roots in the "exploration-exploitation" trade-off of adaptive control [2]. The trade-off roughly stated is that, in general, we cannot identify a system's unknown parameters in closed loop, the question then being "is there an intelligent way to address the problem" rather than the ad-hoc method of applying non "certainty-equivalence" control inputs at random throughout the identification process. In recent years, the optimal sensing problem has garnered a lot of interest in the Control and Robotics community and is variously known as Information-theoretic Control/ Active Sensing and Dual Control [3]–[13]. Discrete dynamic scenarios such as target tracking [7]–[10], and linear spatially distributed systems [14], [15] have been considered, but relatively very little has been done on the optimal sensing of nonlinear dynamical phenomenon. In the linear dynamical scenario, the optimal scheduling problem results in a deterministic optimal control problem which can be solved using Model Predictive control (see below). In the nonlinear case, the problem is stochastic and thus, is significantly harder to solve since we have to solve the associated stochastic DP problem. In the past decade, there has also been a significant volume of research on the problems of co-operative sensing, estimation and control [16]–[21]. These techniques have considered various classes of multi-agent systems and have proposed distributed estimation and control schemes for such problems including formation keeping, flocking and distributed sensing. The multi sensor scheduling problem we consider in this paper also falls under the category of multi-agent systems, however, the structure of the problems that we consider, motivated by the SSA problem, is unlike any other in the aforementioned literature. In particular, the problem we consider has a time varying graph structure that introduces further complexity into the problem and none of the above techniques are applicable. In this paper, we suggest a receding horizon control approach to the solution of such stochastic multi-agent sequential decision making problems, in particular, I-space sequential decision making problems for multiple agents, that allows us to account for all the complexities introduced by the class of problems representing the SSA problem. Further, the underlying structure of the problem is exploited to obtain variance reduction of the gradient estimation that is required by the technique as well as a distributed implementation of the technique.

It is very well known that stochastic control problems with sensing uncertainty, of which sensor scheduling problems are a special case, can be posed as a Markov Decision Problem (MDP) on the I-state, which is usually the conditional filtered pdf of the state of the system [2], [22], [23]. Unfortunately, it is also equally well known that such problems are notoriously difficult to solve owing to the twin curses of dimensionality and history, so much so that such problems have only been solved for small to moderate sized discrete state space problems (i.e., wherein the underlying state space of the problem is discrete). Initially, exact solution of the POMDPs were sought [23], [24] utilizing the convexity of the cost-to-go function in terms of the I-state. However, these techniques do not scale well. Thus, focus shifted to solving such I-space problems using randomized point based value iteration in which a set of random I-states are sampled in the I-space and an approximate MDP defined on these randomly sampled states is then exactly solved using standard DP techniques such as value/ policy iteration [25]–[27]. These methods have resulted in the solution of much higher dimensional problems when compared to the ones that can be solved using exact techniques, however, these methods still do not scale to continuous state, observation and control space problems. The problem we consider in this paper is a multi-agent POMDP and the state and control space of the problem explodes exponentially in terms of the number of agents involved in the problem. Thus, these problems are exponentially harder to solve computationally when compared to single agent I-space problems. There has been considerable interest of late in solving multi-agent MDP problems that are tailored to exploit the structure that is inherent in such problems and Value/ Policy Iteration as well as reinforcement learning based techniques have been designed to solve such problems [28]–[32]. However, the class of problems that we consider in this paper do not conform to the structure required by these techniques. Further, the above methods only apply to small discrete state space problems and thus, are unable to scale to continuous state and observation spaces that are encountered in the SSA inspired multi-sensor scheduling problem considered in this paper. The I-RHC-M technique sequentially solves open loop optimization problems given the current I-state of the system which precludes having to explore the huge state space of multi-agent MDPs and thereby, keeps the method computationally tractable.

Model Predictive or Receding Horizon Control (MPC/ RHC) is one of the most successful applications of control theoretic techniques in the industry [33], [34]. In the deterministic setting, the MPC technique and the Dynamic Programming technique essentially give the same answer in that they provide the optimal feedback control solution. The MPC techniques solve a sequence of finite horizon open loop control problems in a receding horizon fashion instead of solving the infinite dimensional DP equation offline. In this fashion, constraints on the systems can be taken into account, which is very difficult in DP, provided the open loop optimal control problems can be solved online. This has led to many successful applications [33], [34]. Recently, there has been increasing interest in stochastic receding horizon control (SRHC) approaches [35]–[37] that provide receding horizon approaches to constrained stochastic control problems. However, many of these techniques have been developed for linear systems with analytical models of the dynamics and constraints. However, in our case, an analytical model of the process does not exist, instead we have access to simulations of the process. We propose an SRHC approach to solve the multi-agent I-space sequential decision making problems, wherein a sequence of open loop stochastic optimization problems are solved online in a distributed receding horizon fashion. The online optimization is carried out using a distributed simulation based optimization technique. It should be noted that in the stochastic case, the RHC and DP techniques do not coincide because in the DP formulation, the optimization is over feedback policies and not open loop control sequences as in the I-RHC-M technique. However, such DP problems, in particular, I-space problems, especially multi-agent problems, are computationally intractable in continuous state and observation spaces, and thus, the I-RHC-M technique provides a computationally attractive solution to multi-agent I-space problems. The empirical results show that the I-RHC-M technique does lead to better payoffs in terms of information gains when compared to a shortsighted strategy.

The rest of the paper is organized as follows. In Section II, we formulate the class of multi sensor scheduling problems of interest, primarily motivated by the SSA problem. In Section III, we present the I-RHC-M technique for the solution of this class of problems. In Section IV, we present a simple numerical example involving multiple sensors measuring a group of nonlinear simple pendulums, which nonetheless is intractable for other existing techniques in the literature, as an application, and proof of concept, of the I-RHC-M technique.

II. Model and Problem Formulation

In this section, we model the class of multiple sensor scheduling problems that we are interested in solving in this work. This class of problems is motivated by the Space Situational Awareness problem (SSA) but can be extended in a straightforward fashion to other broader classes of problems.

We are interested in tracking a set of N targets where the state of the i^{th} target is governed by the stochastic ODE:

$$\dot{x}_i = f_i(x_i) + g_i w_i, \tag{1}$$

where w_i is a white process noise term perturbing the motion of target i. The term 'target' is used here in the classic sense of being the object that is being sensed or observed by the sensors.

We assume that there are M sensors $\mathcal{S} = \{S_j\}$, typically $M << N$, and suppose that every sensor j can make a measurement of one among a set of targets at ay given point in time denoted by the set $T^j(t)$, where

$$T^j(t) = \{k \in [1, .., N] | \text{target j is visible to sensor i}\}.$$

We make the following assumption to simplify the presentation of our technique, however, it can be relaxed in a relatively straightforward fashion.

A 1. *Any target "'i", at any time "t", is in the field of view (FOV) of only one sensor.*

Further, let us denote by $S(i,t)$, the unique sensor that can see target i at time t, i.e, $S : T \mathrm{x}\, \mathcal{H} \rightarrow \mathcal{S}$, is an integer valued function that maps the product space of the target set T and the time horizon $\mathcal{H} = [0, .., H]$ into a unique positive integer denoting a particular sensor in the set of sensors $\mathcal{S}$. We make the following assumption.

A 2. *The function $S(i,t)$ is known a priori for a given time horizon $\mathcal{H}$.*

Its obvious that the following relationship holds between $T^j(t)$ and $S(i,t)$:

$$T^j(t) = \{i \in T | \, S(i,t) = j\}. \tag{2}$$

Thus, knowing $S(i,t)$ allows us to find $T^j(t)$ and vice-versa. The above assumption allows us to simplify the problem somewhat by assuring us that the set of control choices available to the different sensors is deterministic, albeit time varying. A more general formulation would allow the set of control choices available to sensor j at time t, say $\mathcal{U}_t$, to be random as well as time varying. Further, the random set $\mathcal{U}_t$ would be dependent on the information states of the targets, $\{\chi_i(t)\}$ at time t. A naive approach would be to allow the choice of every target to every sensor at every time step, however, this would lead to an absolute explosion of the complexity and hence, is not practically useful. We note that the $S(i,t)$ function can be thought of as a "most likely" a priori estimate of the sensors' control choices, and discrepancies due to the stochasticity of the system can be accounted for in the planning phase. For instance, if there is a target in view of a sensor that is not predicted by $S(i,t)$ then the sensor will never look at that target, and if a target that was predicted to be there is not, then the reward for making a measurement of the non-existent target would be negative as the uncertainty would increase, and hence, the control policy would learn to avoid such a choice. We shall have more comments about this aspect of the problem after we have presented our I-RHC-M technique to solve the problem.

Suppose now that a sensor j can make a measurement of precisely one of the targets in its FOV at time t, i.e.,

$$y_i = H_j(x_i) + v_j, \text{where } i \in T^j(t), \tag{3}$$

and v_j is a white measurement noise process corrupting the measurements of sensor j.

Given the measurements of a target i till time t, we assume that some suitable Bayes filter is used to estimate its conditional pdf. Let us denote its probability density function/ Information state (I-state) by $\chi_i(t)$. Let $u_t^{S(i,t)}$ denote the control action of sensor $S(i,t)$ at time t, i.e., the target that sensor $S(i,t)$ chooses to measure from among the targets in its FOV at time t, namely $T^{S(i,t)}(t)$.
Let the incremental reward/ utility/ information gain of taking control $u_t^{S(i,t)}$ for target i, at time t, be denoted by $\Delta\mathcal{I}(\chi_i(t), u_t^{S(i,t)})$. Then, the total reward of using a sequence of time-varying control policies over a time horizon $\mathcal{H}$ for target i, $\{u_t^{S(i,t)}(.)\}_{t=0}^H$, is given by:

$$V(\chi_i, \{u_t^{S(i,t)}(.)\}_{t=0}^H)$$
$$= E[\sum_{t=0}^{H} \Delta\mathcal{I}(\chi_i(t), u_t^{S(i,t)}(\bar{\chi}(t)))/\chi_i(0) = \chi_i]. \tag{4}$$

In the above expression, the expectation is over al information trajectories that result from the feedback policies $u_t^{S(i,t)}(.)$. In general, the feedback control function for any sensor $S(i,t)$ that sees target i at time t, is a function of the composite I-state of all the targets $\bar{\chi} = \{\chi_1, \cdots \chi_N\}$, not just χ_i. We assume that the total reward for the system is the sum of the rewards of the individual targets, i.e.,

$$V(\bar{\chi}, \bar{U}(.)) = \sum_{i=1}^{N} V(\chi_i, \bar{U}(\chi))\}), \tag{5}$$

where $\bar{U}(\chi) = \{u_t^j(\chi)\}$ for all possible sensor-time tuples (j,t). The problem can then be posed as one of maximizing the total reward of the system over all feasible feedback policies of the individual sensors. The feasible control set over which the composite control of the sensors $\bar{U}_t(.) = \{u_t^j(.)\}$ at any time t can take values is time-varying, denoted by $\mathcal{U}_t$ and hence, a Dynamic Programming formulation of the sensor scheduling problem has to be time varying and over a finite horizon. The finite time DP problem can be formulated as follows for all $t \in \mathcal{H}$, along with some suitable terminal cost function $J(0, \bar{\chi}) = \Phi(\bar{\chi})$:

$$V(t, \bar{\chi}) = \min_{\bar{U} \in \mathcal{U}_t} [\Delta\mathcal{I}(\bar{\chi}, \bar{U}) + \int_I p(\bar{\chi}'/\bar{\chi}, \bar{U}) V(t-1, \bar{\chi}') d\bar{\chi}'], \tag{6}$$

where $\bar{\chi}$ and $\bar{U}$ are the composite information-state and control-action taking values in the product space of the individual target information states and the individual sensor control spaces. Exploring the entire state and control spaces is essentially impossible in this case owing to the huge dimensionality of the problem. Further, in this case, the DP solution is necessarily time varying which complicates the solution of the DP problem further.

III. Multi-agent Information Space Receding Horizon Control(I-RHC-M)

In previous work, we have proposed an I-space receding horizon control approach that involves solving an open loop stochastic optimization problem at every time step, for the case of scheduling the measurements of a single sensor. In this section, we shall extend this approach to the problem of multiple sensors in the scenario formulated in the previous section.

A. The Open Loop optimization Problem

First, we shall look at the open loop optimization problem, i.e., an optimization problem where the finite horizon cost function $J(\bar{\chi}, \bar{U})$ is a function of a sequence of a given initial I-state $\bar{\chi}$ and a sequence of open loop control actions $\bar{U}$, as opposed to the feedback control policies considered in the DP formulation in the previous section (note that we distinguish the open and closed loop cost functions using $J(.)$ and $V(.)$ respectively). In particular, we would like to solve the open loop stochastic optimization problem:

$$\min_{\{u_t^j\}} \sum_{i=1}^{N} J(\chi_i, \{u_t^{S(i,t)}\}), \tag{7}$$

where the optimization is over all possible control choices of every sensor-time 2-tuple (j,t). It behooves us to take a closer look at the notation above. In the above notation $u_t^{S(i,t)}$ denotes the control choices of sensor $S(i,t)$ at time t. We use this notation because the total reward of the system can be defined in terms of the individual rewards of the different targets and it further allows us to extract structure from the problem. Since sensor $S(i,t)$ may be seeing other targets $j \in T^{S(i,t)}(t)$, we note that $S(i,t) = S(j,t)$ for all $j \in T^{S(i,t)}(t)$. Thus, the choices $u_t^{S(i,t)} \in T^{(S(i,t)}(t)$, i.e., the sensor $S(i,t)$ can choose to measure any of the targets in $T^{S(i,t)}(t)$ at time t. Hence, the open loop optimization is to maximize the reward of the system given the control choices available to every sensor-time 2-tuple (j,t),and a given initial I-state $\bar{\chi}$ over the finite time horizon $\mathcal{H}$. Note that this is an open loop optimization problem and does not consider the control to be a function of the particular information states that are encountered along an information trajectory.

Next, we consider a randomization of the control choices available to any given sensor: instead of the control u_t^j being deterministic, i.e, the sensor chooses to measure exactly one of the targets in its FOV at time t, we assume that the sensor chooses to measure one of the targets in its FOV with a certain probability. Let us denote the probabilities representing the randomized policies for every sensor time tuple (j,t) by $\{\pi_{t,k}^j\}$ where:

$$\pi_{t,k}^j = Prob.(u_t^j = k), \tag{8}$$

i.e., the probability that the j^{th} sensor at time t chooses to measure the k^{th} target in its FOV. Compactly, we shall denote the randomized policy for the sensor time 2-tuple (j,t) by Π_t^j. Also, we shall denote the randomized policies of all the sensor-time tuples by $\bar{\Pi} = \{\Pi_t^j\}$. Given the definitions above, the total reward for target i in following the composite randomized sensor policy $\bar{\Pi} = \{\Pi_t^j\}$ is given by the following:

$$J(\chi_i, \bar{\Pi}) = J(\chi_i, \{\Pi_t^j\}) = \sum_{u_1^{S(i,1)} .. u_H^{S(i,H)}} J(\chi_i, u_1^{S(i,1)}, \cdots, u_H^{S(i,h)}) \pi_{1,u_1^{S(i,1)}}^{S(i,1)} ... \pi_{H,u_H^{S(i,H)}}^{S(i,H)}. \tag{9}$$

The average above is over all possible choices of $u_t^{S(i,t)}$ for all possible $t \in \mathcal{H}$. Further, the total reward in following the randomized policy $\{\Pi_t^j\}$ is then given by:

$$J(\bar{\chi}, \bar{\Pi}) = \sum_{i=1}^{N} J(\chi_i, \bar{\Pi}). \tag{10}$$

B. Simulation based Information Gradient Technique

In the following, we shall use gradient ascent to find a maximum for the total reward of the system. In order to do this, we first need to evaluate the gradient $\frac{\partial J}{\partial \Pi_t^j}$ for every sensor-time 2-tuple (j,t). In particular, we can show that the gradient $\frac{\partial J}{\partial \pi_{t,k}^j}$ is given by the following:

$$\frac{\partial J}{\partial \pi_{t,k}^j} = \sum_{l \in T^j(t)} \sum_{u_1^{S(l,1)} ... u_H^{S(l,H)}} J(\chi_l, u_1^{S(l,1)}, .., u_t^{S(l,t)} = j, .., u_H^{S(l,H)}) \pi_{1,u_1^{S(i,1)}}^{S(l,1)} .... \pi_{H,u_H^{S(l,H)}}^{S(l,H)}. \tag{11}$$

To see why, note that Π_t^j explicitly appears only in the reward expressions of the targets that are in the FOV of sensor j at time t, namely $T^j(t)$. Hence, the gradient only involves contributions from these targets. Further, note that for any $l \in T^j(t)$, by definition $S(l,t) = j$. Hence, the above expression implies that the gradient of the total reward with respect to the probability that the sensor-time pair (j,t) measures the k^{th} target in its field of view is given by the average cost of the information-trajectories of the targets in $T^j(t)$, given that sensor j at time t actually chooses to measure the k^{th} object in its FOV, i.e.,

$$\frac{\partial J}{\partial \pi_{t,k}^j} = \delta J^{(j,t)}(\bar{\Pi}, u_t^j = k), \tag{12}$$

where

$$\delta J^{(j,t)}(\bar{\Pi}, u_t^j = k) = \sum_{l \in T^j(t)} \sum_{u_1^{S(l,1)} ... u_H^{S(l,H)}} J(\chi_l, u_1^{S(l,1)}, .., u_t^{S(l,t)} = j, ..., u_H^{S(l,H)}) \pi_{1,u_1^{S(i,1)}}^{S(l,1)} .... \pi_{H,u_H^{S(l,H)}}^{S(l,H)}, \tag{13}$$

i.e., the average reward of the information trajectories of the targets in in $T^j(t)$, given $u_t^j = k$ and all other sensor-time pairs (j', t') stick to their randomized policies $\Pi_{t'}^{j'}$. The gradient ascent algorithm is the following:

$$\Pi_t^j = \mathcal{P}_P[\Pi_t^j + \gamma \frac{\partial J}{\partial \Pi_t^j}], \tag{14}$$

where $\mathcal{P}_P[.]$ denotes the projection of a vector onto the space of probability vectors P, and γ is a small step size parameter. Note that the policy update for the randomized policy of the sensor-time pair (j,t) need not be a probability vector and hence, the necessity of the projection operator $\mathcal{P}_P[.]$ in the above expression.

Let us also define the following reward function:

$$J^{(j,t)}(\bar{\chi}, \bar{\Pi}) =$$

$$\sum_{l\in T^j(t)} \sum_{u_1^{S(l,1)}\ldots u_H^{S(l,H)}} J(\chi_l, u_1^{S(l,1)}, .., u_H^{S(l,H)}) \pi_{1,u_1^{S(i,1)}}^{S(l,1)} \ldots . \pi_{H,u_H^{S(l,H)}}^{S(l,H)}. \quad (15)$$

The above is the total information gain of the targets in the FOV of sensor j at time t given that we follow the policy $\bar{\Pi} = \{\Pi_t^j\}$. We shall come back to this reward later on in this section when we provide a game theoretic interpretation of the technique presented here. Of course, implementing the deterministic gradient ascent algorithm above entails averaging over multiple realizations of the information trajectories and sensor control sequences. Instead, we use a stochastic gradient ascent technique utilizing only one sample realization of the information trajectory. In particular, we have the following update rule:

$$\Pi_t^j = \mathcal{P}_P[\Pi_t^j + \gamma \widehat{\frac{\partial J}{\partial \Pi_t^j}}], \quad (16)$$

where

$$\widehat{\frac{\partial J}{\partial \pi_{t,k}^j}} = J^{(j,t))}(\omega) \text{ if } u_t^j = k = 0, \text{ o.w.} \quad (17)$$

where ω represents a sample realization of the information process, and $J^{(j,t)}(\omega)$ represents the information gain of the targets in $T^j(t)$ for that particular realization of the information process.

We make the following remarks about the structure of the problem that allows us to extract significant variance reduction in the gradient estimates as well as a distributed implementation of the gradient algorithm.

Remark 1. ***Variance Reduction:*** *The variance of the gradient estimate is due to two reasons: 1) the randomness of the target information trajectories and 2) the randomness of the sensor policies. The stochastic gradient technique has structure that allows us to alleviate both to a large extent. The variance of the estimate $\widehat{\frac{\partial J}{\partial \Pi_t^j}}$ is reduced by orders of magnitude since we need only simulate the I-trajectories of the objects in $T^j(t)$ in order to obtain an estimate as opposed to having to simulate the I-trajectories of all the objects. Also, note that the gradient only depends on the sensor-time tuples $\Pi_\tau^{S(l,\tau)}$, for all $l \in T^j(t)$ and $\tau \in \mathcal{H}$. Hence, we need not simulate the policy of every sensor-time pair, only $\Pi_\tau^{S(l,\tau)}$ as defined above, thereby further reducing the variance of the gradient estimate.*

Remark 2. ***Distributed Implementation:*** *The remark above also tells us as to how to obtain a distributed implementation of the gradient ascent algorithm. Suppose that we have a CPU for every sensor-time tuple (j,t). This processor evaluates the gradient $\frac{\partial J}{\partial \Pi_t^j}$, and from the above remark, it follows that the CPU need only know the policies of the sensor time tuples $\Pi_\tau^{S(l,\tau)}$, $l \in T^j(t)$ and $\tau \in \mathcal{H}$, in order to evaluate the gradient. Thus, the CPU for the sensor-time pair (j,t) need only be connected with the processors for the sensor-time pairs $(S(l,\tau),\tau)$, where $l \in T^j(t)$ and $\tau \in \mathcal{H}$, i.e, the processor need only know the policies of the sensor-time pairs that affect the information reward for the targets within $T^j(t)$, the FOV of sensor j at time t. This allows for a sparse connection graph among the processors thereby facilitating a distributed implementation of the gradient ascent algorithm.*

Finally, we examine the effect of the known $S(i,t)$ function on the simulation based optimization problem presented above. Due to the stochasticity of the system, there are bound to be cases when: 1) a target that is predicted to be in the FOV of a sensor is not there, and 2) a target that is not predicted in the FOV is actually there. The first case implies that there is a negative reward to the sensor if it makes a measurement of an object that is not in its FOV since this leads to a loss of information regarding the system and therefore, the gradient based technique wold not choose such an action if the target does not happen to be in the FOV of the sensor frequently. The second case implies that the sensor will never make a measurement of the unforeseen target. This, of course, implies that the policy might be non-optimal since the sensor does not consider the unforeseen target, however, in our opinion, this is a small price to pay for keeping the problem computationally tractable and amenable to a solution procedure.

C. Convergence

The stochastic information gradient algorithm is guaranteed to converge to one of the set of Kuhn-Tucker points of the function $J(\bar{\chi}, \{\Pi_t^j\})$ with the constraints being that the randomized policy for every sensor-time pair (j,t), Π_t^j needs to be a probability vector. The proof is essentially the same as that of the stochastic optimization problem in the I-RHC technique, and thus, we only state the result in the following without a proof.

In the following we drop all reference to the initial I-state $\bar{\chi}$ in the optimization problem for $J(\bar{\chi}, \bar{\Pi})$ and refer to the function as only $J(\bar{\Pi})$. The gradient of the function $J(.)$ with respect to $\bar{\Pi}$ is denoted by $\mathcal{G}(\bar{\Pi})$. Let $\{q_i(\bar{\Pi}) \leq 0\}$ denote the inequality constraints on the problem for some $i = 1, ...K$ and $h_i(\bar{\Pi}) = 0$ denote the equality constraints for some $i = 1, \cdots L$. The inequality constraints are all linear and are of the form $0 \leq \pi_{t,k}^j \leq 1$ for all $t \in \mathcal{H}$, sensors j and relevant choices of sensor j at time t, $k \in T^j(t)$. The equality constraints are linear and of the form $\sum_k \pi_{t,k}^j = 1$ for all $t \in \mathcal{H}$ and all sensors j. However, note that the total reward function $J(\bar{\chi}, \bar{\Pi})$ is multilinear and in general, can have multiple local minima. Let the compact set defined by the constraints above, the space of stochastic policies, be denoted by P. Let us denote the set of stationary points of $J(.)$ by S where

$$S = \{\bar{\Pi} : \mathcal{G}(\bar{\Pi}) - \sum_i \lambda_i \nabla q_i(\bar{\Pi}) - \sum_j \mu_j \nabla h_i(\bar{\Pi}) = 0, \ \lambda_i \geq 0\}, \quad (18)$$

where $\lambda_i = 0$ whenever $q_i(\bar{\Pi}) < 0$ and $\lambda_i \geq 0$ otherwise. Note that the set S is the collection of all the Kuhn-Tucker (K-T) points of the function $J(\bar{\Pi})$. The set is non empty since $J(\bar{\Pi})$ is continuous and the set P of stochastic policies is compact

and therefore, the function will attain its extrema in P. Moreover, the set S_i can be decomposed into disjoint, connected and compact subsets S such that $J(\bar{\Pi}) = constant = C_i$ over each S_i ([38], p. 126), since $J(.)$ and $q_i(.)$ are twice continuously differentiable. Let the step size parameters satisfy the following conditions:

$$\sum_n \epsilon_n = \infty, \ \sum_n \epsilon_n^2 < \infty.$$

Then the following result holds:

Proposition 1. *The sequence of policy updates $\bar{\Pi}_n \rightarrow S_i$, for some unique i, almost surely, i.e., the stochastic gradient algorithm (Eq. 16 converges to a set of stationary (K-T) points such that the value of $J(.)$ on each such set S_i is constant.*

D. A Game Theoretic Interpretation

The problem of optimizing the total reward function $J(\bar{\chi}, \bar{\Pi})$ can also be interpreted in a game theoretic fashion. Consider every sensor-time pair (j,t) to be an agent and the randomized policy of the pair, Π_t^j, to be the corresponding mixed strategy of the player (j,t) (a mixed strategy is a randomized policy over the choices available to the agent as opposed to a pure strategy which chooses a unique alternative). Also, recall the reward of the sensor-time pair (j,t) given by $J^{(j,t)}(\bar{\chi}, \bar{\Pi})$ (cf. Eq. 15). This can be considered as the pay-off of agent (j,t). Thus, the game can be thought of as one in which every sensor-time pair (j,t) tries to maximize its payoff $J^{(j,t)}(\bar{\chi}, \bar{\Pi})$, namely the total information gain of all the objects in the FOV of sensor j at time t, i.e., the set $T^j(t)$. Since this is a finite player game with finite number of choices for every agent (j,t), it has atleast one Nash equilibrium, i.e., a strategy for each pair (j,t), denoted by Π_t^{j*}, such that it is the best response to the randomized policies of all other sensor time pairs (j',t').

However, the stochastic information gradient technique presented previously need not converge to a Nash equilibrium. The sensor-time policies Π_t^i do converge to local maxima of the individual pay-off functions $J^{(j,t)}(.,.)$ w.r.t. Π_t^j, while fixing all other sensor-time policies $\Pi_{t'}^{j'}$, but not necessarily a global equilibrium, which would be a Nash equilibrium. However, note that the goal of our scheduling technique is to find a maximum of the total reward function $J(\bar{\chi}, \bar{\Pi})$ and not necessarily the maximization of the individual payoff functions $J^{(j,t)}(\bar{\chi}, \bar{\Pi})$ (or find a Nash equilibrium for the game). Thus, there is no guarantee that the algorithm would actually converge to a Nash equilibrium of the game-theoretic problem.

E. Receding Horizon Control

We have presented a simulation based stochastic gradient technique to get a minimum of the total reward $J(\bar{\chi}, \bar{\Pi})$ with respect to the sensor-time randomized policies $\bar{\Pi} = \{\Pi_t^j\}$ given some initial information state $\bar{\chi}$. In the following, we may recursively solve such open loop optimization problems at every time step given the current information state to obtain a receding horizon solution to the sensor scheduling problem for multiple sensors.

Suppose at the initial time the information state of the system is $\bar{\chi}_0$. Then, given this initial information state, we use the stochastic information gradient technique presented previously to obtain a maximum for the total information reward of the system over the randomized policies of every sensor-time pair (j,t) over some given horizon $\mathcal{H}$. Then, we implement the first time step of the policies for every sensor j, and take measurements of the targets as specified by the control policies at the first time step. Then, we use suitable filtering techniques to update the information state of the system to obtain a new information state $\bar{\chi}'$. Then, we set $\bar{\chi}_0 = \bar{\chi}'$, and repeat the information gradient technique to obtain a minimum of the total reward over the next horizon $\mathcal{H}$ given the new information state $\bar{\chi}'$. The technique can be summarized in the I-RHC-M algorithm below.

Algorithm 1 Algorithm I-RHC-M

- Given initial information state $\bar{\chi}_0$ and lookahead horizon $\mathcal{H}$
  1) Use the stochastic information gradient technique (Eq. 16) to obtain a minimum of the total reward $J(\bar{\chi}_0, \{\Pi_t^j\})$ over all sensor-time pairs (j,t) over the horizon $\mathcal{H}$.
  2) Output converged T-step policy Π_t^{j*} for every sensor-time pair (j,t)
  3) Observe noisy measurement z based on the first step of policy $\{\Pi_t^{j*}\}$ and update information state using a suitable filter to obtain the new I-state χ_1.
  4) Set $\chi_0 = \chi_1$ and go to Step 1.
- End

Remark 3. *It should be noted that the receding horizon solution of the multi-sensor scheduling problem is not the same as the solution of the suitable Dynamic Programming problem, if the DP problem could be solved. The RHC solution will furnish a trajectory based feedback solution in that it is dependent on the current information state of the system. However, because it is a stochastic scenario, the open loop optimization performed by the information gradient technique is not the same as the DP solution, which happens to be over feedback policies as opposed to open loop policies. However, as has already been noted, solving the DP problem is essentially impossible and thus, the RHC solution provides a long sighted policy that is adjusted based on the current information state, and which may be expected to outperform myopic policies.*

IV. Illustrative Example

In this section, we shall apply the I-RHC-M technique developed in the previous section to a simple problem involving multiple simple pendulums that mimics the SSA problem. Although the problem is relatively simple, nevertheless it is so high dimensional that no existing technqiue in the literature

can be used to solve this problem. We shall comment more about the computational complexity of the current problem a little later in the section. We consider a set of N simple (nonlinear) pendulums governed by the stochastic differential equations:

$$\ddot{\theta}_i = \frac{g}{l}\sin\theta_i + w_i, \tag{19}$$

where θ_i is the angular displacement of the i^{th} pendulum and w_i is a white noise process affecting the motion of the pendulum. We consider a set of sensors with disjoint FOVs. The FOV of the j^{th} sensor is defined to be the angular displacement set $\mathcal{F}^j = [\theta_l^j, \theta_u^j]$. We assume that a sensor j can measure the state of a pendulum i only when it is its field of view $\mathcal{F}^j$ according to the following observation equation:

$$y_i^j = \bar{\theta}_i + v^j, \tag{20}$$

where $\bar{\theta}_i$ denotes the state of the i^{th} pendulum, and v^j is a white noise process corrupting the measurements of the j^{th} sensor. Note that the above simple problem has the flavor of the SSA problem, in that each sensor has a bounded FOV, and can measure a target if and only if its within the FOV. Further, the pendulum problem is periodic like the SSA problem and thus, targets periodically leave and enter the FOVs of the different sensors. For the numerical examples below, we apply our I-RHC-M technique to a situation where $N = 4$ and $M = 3$, i.e, there are 4 targets and 3 sensors to measure them. The initial states of the pendulums are chosen in a random fashion and we assume that the statistics of the process noise corrupting the dynamics of each pendulum, and the sensor noise corrupting the measurements of each sensor, are the same. We assume that the Gaussian assumption holds in this problem and use extended Kalman filters (EKF) to approximate the filtered densities, or I-states of the pendulums. The information gain metric used in this work is the difference in the determinants of the information (inverse of the covariance) matrix of the targets and the total information gain is the sum of the information gains of the different targets. The state space of each pendulum is 2 dimensional. Given that the Gaussian approximation holds, the I-state of every pendulum can be specified by its mean and covariance and hence, the I-state of each pendulum is 6 dimensional. Thus, given 4 pendulums, the joint state space of the problem is 24 dimensional. Also, the control set is finite and is equal to $4^3 = 64$. Of course, the structure of the problem implies that the actual number of choices that any sensor has is far fewer than 4. Also, note that the observation space is continuous. Thus, the DP problem that needs to be solved to tackle the above problem resides in a 24 dimensional state space and consequently, none of the existing techniques can solve such a high dimensional problem, given even the extensive computing resources available today (the highest dimensional DP problem that can be solved is usually 6 to 8 dimensional). Thus, even this simple example, shows the degree of computational complexity that is inherent in the problem and to the best of our knowledge, the I-RHC-M procedure is the only one that is capable of tackling such problems.

The results of our numerical simulations are showed in

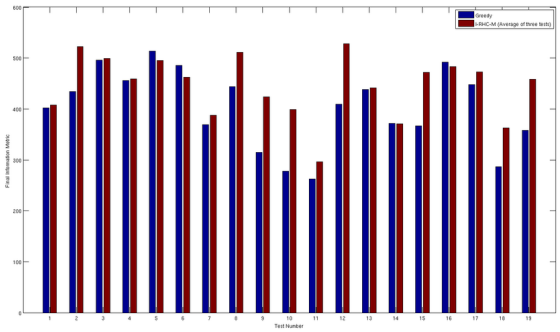

Fig. 1. Performance of the I-RHC-M algorithm in the average case scenario

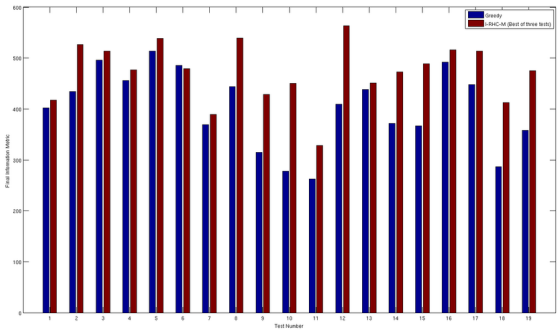

Fig. 2. Performance of the I-RHC-M algorithm in the best case scenario

Figs. 1 and 2. For comparison, we chose a greedy policy as it is the only other technique, other than the I-RHC-M technique, that scales to high dimensional problems such as the one considered in this paper. The I-RHC-M technique had a lookahead horizon of 10 timesteps and the information gains were evaluated over a total time horizon of 20 timesteps. In Fig. 1, we show the average gain/ loss of the I-RHC-M method, averaged over three runs of the I-RHC-M technique, over that of the greedy policy, for twenty different initial conditions, i.e., we run the I-RHC-M technique three different times for each initial condition and compare the average information gain over these runs to the information gain of the greedy policy. Note that the I-RHC-M policies will, in general, be different for different runs due to the stochasticity of the algorithm. In Fig.2, we compare the information gain of the best of the three I-RHC-M runs to the information gain of the greedy policy. Note that there is no guarantee that the I-RHC-M policy can beat the greedy policy, atleast theoretically. However, as can be seen from the plots, the I-RHC-M technique does beat the greedy policy most of the time. From the above plots, it was found that the I-RHC-M technique provided an improvement of approximately 13% over the greedy policy in the averaged case, and an improvement of 20% in the best of three case. If the cost function used is a product of the information gains for the individual oscillators, then the information gain obtained by the I-RHC-M technique over the greedy policy is much higher (it is almost 10 times better in that case). However, using

the product gain function implies that the parallelizability and variance reduction properties of the technique no longer hold. To solve this issue, we can look at the logarithm of the product of information gain as our objective function, in which case the situation is similar to the case when the objective function is the sum of information gains, in terms of the parallelizability and variance reduction properties of the method. Using the log-product gain function involves some minor changes to the stochastic gradient update formulas. However, we have still not obtained results for this case but our conjecture is that we will be able to realize very high information gains just as in the case of the product gain function using the log-product formulation.

V. Conclusion

In this paper, we have introduced an information space receding horizon control technique for multi-agent systems, termed the I-RHC-M technique, with application to the SSA problem. The method is based on a simulation based stochastic gradient technique that is used to solve a finite horizon stochastic optimization problem recursively at every time step, thereby providing a feedback solution to the problem. We have shown that the method is highly parallelizable and naturally inherits a variance reduction property owing to its structure. We have also shown that the method is capable of handling very high dimensional continuous state and observation space problems for multi-agent systems that no other existing technique can claim to solve. We have tested our technique on a simple example, which is nonetheless computationally intractable for other existing solution techniques, and have shown that the method achieves significant improvement over a greedy policy (the only other computationally viable strategy). In the future, we shall concentrate on developing the "log-product" version of the technique which we believe will give us information gains commensurate with the "product" information gain case while inheriting the parallelizability and variance reduction of the "sum" information gain case.

References

[1] S. Chakravorty and R. Erwin, "Information space receding horizon control," in *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, 2011.

[2] P. R. Kumar and P. P. Varaiya, *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentic Hall, NJ: Prentice Hall, 1986.

[3] B. Grocholsky, *Information-theoretic Control of Multiple Sensor Platforms, Phd thesis*. Sydney, Australia: Australian Center for Field Robotics, University of Sydney, 2002.

[4] C. Stachnis, *Exploration and Mapping with Mobile Robots, PhD Thesis*. Freiburg, Germany: University di Freiburg, 2006.

[5] F. B. et. al., "Information based adaptive robot exploration," in *Proc. of 2002 IEEE/ RSJ Int. conf. on Intell. rob. sys.*, 2002.

[6] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, "Information-theoretic coordinated control of multiple sensor platforms," in *Proc. of 2003 IEEE Int. Conf. Rob. Aut.*, 2003.

[7] T. H. Chung *et al.*, "On a decentralized active sensing strategy using mobile sensor platforms in a network," in *Proc. IEEE Int. Conf on Dec. Cont.*, 2004.

[8] S. Aranda *et al.*, "On optimal sensor placement and motion coordination for target tracking," in *Proc. IEEE Int. conf. Dec. Cont.*, 2005.

[9] R. Olfati-Saber, "Distributed tracking for mobile sensor networks with information-driven mobility," in *Proc. Amer. Cont. Conf. (ACC)*, 2007.

[10] G. M. Hoffman *et al.*, "Mutual information methods with particle filters for mobile sensor network control," in *Proc. IEEE CDC*, 2006.

[11] H.-L. choi, *Adaptive Sampling and Forecasting with Mobile Sensor Networks, PhD thesis*. Cambridge, MA: Department of Aeronautics and Astronautics, MIT, 2008.

[12] H.-L. Choi, J. P. How, and J. Hansen, "Ensemble based adaptive targeting of mobile sensor networks," in *Proceddings of the American Control Conference*, 2007.

[13] P. Frazier, W. B. Powell, and S. Dayakin, "A knowledge gradient policy for sequential information collection," *SIAM Journal of Control and Optimization*, vol. 47, pp. 2410–2439, 2008.

[14] A. Y. Khapalov, "Optimal measurement trajectories for distributed parameter systems," *Systems and Control Letters*, vol. 18, pp. 467–477, 1992.

[15] J. A. burns *et al.*, "A distributed parameter control approach to optimal filtering and smoothing with mobile sensor networks," in *Proc. Mediterrenean Control Conf.*, 2009.

[16] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control: theory and applications*. London, UK: Springer-Verlag, 2008.

[17] W. Ren *et al.*, "Information consensus in multiple vehicle co-operative control," *IEEE Control Systems Magazine*, vol. 27, pp. 71–82, 2007.

[18] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1465–1476, 2004.

[19] P. Ogren *et al.*, "Cooperative control of mobile sensor networks:adaptive gradient climbing in a distributed environment," *IEEE transactions on Automatic Control*, vol. 49, pp. 1292–1302, 2004.

[20] R. M. Murray, "Recent research in cooperative control," *ASME Journal of Dynamical Systems, Measurement and Control*, vol. 129, 2007.

[21] J. Shamma, *Cooperative control of distributed multi-agent systems*. Wiley-Interscience, 2008.

[22] D. P. Bertsekas, *Dynamic Programming and Optimal Control, vols I and II*. Cambridge: Athena Scientific, 2000.

[23] L. P. Kaelbling and M. L. Littman, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[24] M. L. Littman, *Algorithms for Sequential Decision Making, Ph.D. Thesis*. Brown University, PRovidence, RI: Computer Science Dept., 1996.

[25] N. Roy, G. Gordon, and S. Thrun, "Finding approximate pomdp solutions through belief compression," *Journal of artificial intelligence research*, vol. 23, pp. 1–40, 2005.

[26] M. Spaan and N. Vlassis, "Perseus: Randomized point-based vallue iteration for pomdps," *Journal of artificial intelligence research*, vol. 24, pp. 195–220, 2005.

[27] J. Pineau, N. Roy, and S. Thrun, "A hierarchical approach to pomdp planning and execution," in *Workshop on Hierarchy and Memory in Reinforcement Learning*, 2001.

[28] C. Guestrin *et al.*, "Multi-agent planning with factored mdps," in *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[29] ——, "Efficient algorithms for factored mdps," *Journal of Artificial Intelligence Research*, vol. 19, pp. 399–468, 2003.

[30] K. Kim and T. Dean, "Solving factored mdps with non-homogeneous partitioning," in *International Joint Conference on ARtificial Intelligence*, 2001.

[31] J. Kok and N. Vlassis, "Sparse co-ordinated q learning," in *International Conference on Machine Learning*, 2004.

[32] L. Busoniu *et al.*, "A comprehensive survey of multi-agent reinforcement learning," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 38, pp. 156–172, 2008.

[33] T. Keviczky and G. J. Balas, "Software-enabled receding horizon control for autonomous uav guidance," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 29, 2006.

[34] D. Mayne *et al.*, "Constrained model predictive control: Stability and otpimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[35] J. A. Primbs and C. H. Sung, "Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise," *IEEE transactions on Automatic Control*, vol. 54, pp. 221–230, 2009.

[36] F. Herzog *et al.*, "Model predictive control for portfolio selection," in *Proc. of the American Control Conference*, 2006.

[37] D. A. Castanon and J. M. Wohletz, "Model predictive control for dynamic unreliable resource allocation," in *Proc. of the IEEE Int. Conf. Dec. Control*, 2002.

[38] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New york, NY: Springer, 2003.