REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 12-14-2012	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 09/15/2009 - 09/14/2012

4. TITLE AND SUBTITLE A MULTI-RESOLUTION APPROACH TO THE FOKKER-PLANCK-KOLMOGOROV EQUATION WITH APPLICATION TO STOCHASTIC NONLINEAR FILTERING AND OPTIMAL DESIGN	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** FA9550-09-1-0694 **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) D. Yu and S. Chakravorty	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Texas Engineering Experiment Station 1470 William D. Fitch Pky College Station TX	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research Suite 325, Room 3112 875 Randolph Street Arlington, VA 22203-1768	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFOSR **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** AFRL-OSR-VA-TR-2012-1238

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution A - Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this paper, we consider the filtering of systems governed by partial differential equations (PDE). We adopt a reduced order model (ROM) based strategy to solve the problem. We propose an iterative version of the snapshot proper orthogonal decomposition (POD) technique, termed I-POD, to sequentially construct a single ROM for PDEs that is capable of capturing their behavior over the entire state space of the system, and not just around the snapshot trajectory. Further, the technique is entirely data based, and is applicable to forced as well as unforced systems. The I-POD is compared to two other ROM techniques: the Balanced POD (BPOD) and the dynamic mode decomposition (DMD). We apply the ROM generated using the I-POD technique to construct reduced order Kalman filters to solve the filtering problem. The methodology is tested on several 1-dimensional PDEs of interest including the heat equation, the wave equation and 2-dimensional pollutant transport equation.

15. SUBJECT TERMS

Proper Orthogonal Decomposition (POD), Filtering/ Data Assimilation, Partial Differential Equations

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Fariba Fahroo
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	39	**19b. TELEPHONE NUMBER** *(include area code)* 703-696-8429

Project Report (2011-2012)

In the last year of our project " A Multiresolution Approach to the Solution of the Fokker-Planck Equation with Application to Filtering and Stochastic Control", the following papers were submitted/ published:

D. Yu and S. Chakravorty, "An Iterative Proper Orthogonal Decomposition Approach with Application to the Filtering of PDEs", *SIAM J. Sci. Comput.,* under review

Z. Sunberg, S. Chakravorty and R. Erwin, "Information Space Receding Horizon Control", *IEEE Transactions on SMC-B,* accepted, to appear

Z. Sunberg, S. Chakravorty and R. Erwin, "Information Space Receding Horizon Control for Multi-agent Systems", *Proceedings of the 2012 American Control Conference*

The copies of these papers are attached to the document overleaf.

AN ITERATIVE PROPER ORTHOGONAL DECOMPOSITION (I-POD) TECHNIQUE WITH APPLICATION TO THE FILTERING OF PARTIAL DIFFERENTIAL EQUATIONS

D. YU, [†] AND S. CHAKRAVORTY [‡]

Abstract. In this paper, we consider the filtering of systems governed by partial differential equations (PDE). We adopt a reduced order model (ROM) based strategy to solve the problem. We propose an iterative version of the snapshot proper orthogonal decomposition (POD) technique, termed I-POD, to sequentially construct a single ROM for PDEs that is capable of capturing their behavior over the entire state space of the system, and not just around the snapshot trajectory. Further, the technique is entirely data based, and is applicable to forced as well as unforced systems. The I-POD is compared to two other ROM techniques: the Balanced POD (BPOD) and the dynamic mode decomposition (DMD). We apply the ROM generated using the I-POD technique to construct reduced order Kalman filters to solve the filtering problem. The methodology is tested on several 1-dimensional PDEs of interest including the heat equation, the wave equation and 2-dimensional pollutant transport equation.

Keywords: Proper Orthogonal Decomposition (POD), Filtering/ Data Assimilation, Partial Differential Equations.

1. Introduction. In this paper, we are interested in the filtering/ data assimilation in systems that are governed by partial differential equations (PDE). We take a reduced order model (ROM) based approach to the problem. We propose an iterative version of the snapshot proper orthogonal decomposition (POD) technique that allows us to form an ROM of the PDE of interest in terms of the eigenfunctions of the PDE operator. The I-POD is compared to two other ROM techniques: the Balanced POD (BPOD) and the dynamic mode decomposition (DMD) technique. We then apply this ROM, along with the Kalman filtering technique, to form a reduced order filter for the PDE. The filter is constructed in an offline-online fashion where the expensive computations for the ROM construction is accomplished offline, while the online part consists of the reduced order Kalman filter which is much more computational tractable than the full problem. The technique is applied to several one and two dimensional partial differential equations.

We take a ROM based approach to solving the problem of filtering in PDEs. In particular, we rely on the so-called proper orthogonal decomposition (POD), more precisely, the snapshot POD technique, to construct ROMs for the PDE of interest. The POD technique has been used extensively in the Fluids community to produce ROMs of fluid physics phenomenon such as turbulence and fluid structure interaction [1–4]. There has also been work recently to produce so-called balanced POD models to better approximate outputs of interest through an amalgam of the snapshot POD and the balanced model reduction paradigm of control theory [5] to produce computationally efficient balanced POD models of the physical phenomenon of interest [6, 7]. More recently, there has been work on obtaining information regarding the eigenfunctions of a system based on the snapshot POD, called the dynamic mode

[*] This work was supported by AFOSR grant FA9550-09-1-0694

[†] D. Yu is a Graduate Student Researcher, Department of Aerospace Engineering, Texas A&M University, College Station

[‡] S. Chakravorty is an Associate Professor of Aerospace Engineering, Texas A&M University, College Station

decomposition (DMD) [8,9]. However, a couple of issues are central to the construction of the snapshot POD technique: 1) at what times do we take snapshots of the process, and 2) the snapshot POD essentially provides a reduced basis approximation of the localized behavior of a system around the snapshot trajectory, is there a constructive way to infer the behavior of a system from the snapshot POD away from this trajectory? We propose a randomly perturbed iterative version of the snapshot POD (I-POD) which sequentially allows us to sample the process of interest at various different time scales, and with different initial condition. Further, we show that this process allows us to theoretically reconstruct all the eigenfunctions of the original system using either data from experiments, or from numerical simulations (similar to the DMD approach) thereby allowing us to sequentially build a single ROM for the system, utilizing multiple sample trajectories of the system, and by sampling these trajectories at different timescales. To the best of our knowledge, our work is the first that proves that, under certain assumptions, the snapshot POD does extract the eigenfunctions of the underlying operator (though ample empirical evidence is provided, this is not proved in the DMD paper [8]). The I-POD approach is sequential and involves solving a sequence of small eigenvalue problems to get a single ROM of a large scale system. This is in contrast to techniques such as the balanced POD and Eigensystem Realization Algorithm (ERA) [10], that may require the solution of a very large Hankel singular value decomposition problem if the number of inputs/outputs is large. Hence, the I-POD can be a computationally attractive alternative to the BPOD in such a scenario. Also, we show in the paper that the I-POD compares favorably to the BPOD and DMD in accuracy. Further, the sequential extraction of eigenfunctions from different time scales is in general not possible with the DMD approach since it does not perform adjoint simulations as done by the I-POD. Moreover, to the best of the knowledge of the authors, except for some recent work in the Fluid mechanics community [11], the use of POD based ROMs for filtering PDEs is relatively rare.

The problem of estimating dynamic spatially distributed processes is typically solved using the Ensemble Kalman Filter (EnKF) and has been used extensively in the Geophysics literature [12,13] and more recently, in Dynamic Data Driven Application Systems (DDDAS) and traffic flow problems [14–20]. The EnKF is a particle based Kalman filter that maintains an ensemble of possible realizations of the dynamic map. The Kalman prediction and measurement update steps are performed using ensemble operations instead of the traditional matrix operations. A primary issue with the EnKF is the choice of the ensemble realizations and their number. This is almost always done in a heuristic fashion. Also, the prediction stage requires expensive forward simulations of the realizations using a solver which can take a significant amount of time. Thus, real time operation is an issue. In contrast, all the expensive computations for our ROM based technique, namely POD basis and ROM generation, are done offline and hence, real time operation is never an issue given the offline computations. Thus, we may think of our approach as a computationally tractable alternative to the EnKF algorithm. Historically, there has been a lot of theoretical research in the Control Systems community on the estimation and control of systems driven by PDEs [21–30]. In fact, it is well known that for linear PDEs, an infinite dimensional version of the Kalman filter exists which involves the solution of an operator Ricatti equation [31]. This can be very computationally intensive and may be unsuitable for online implementation. In contrast, the major computational complexity of the

ROM based technique that we propose is offline and the online computations are essentially trivial thereby making the technique very suitable for online implementation.

The paper is organized as follows. In section 2, we introduce the filtering problem. In section 3, we present the offline I-POD procedure used to construct the ROM for large scale linear systems. In Section 4, we compare the I-POD technique to the BPOD and DMD techniques. In section 5, we outline the online portion of the reduced order Kalman filter constructed for filtering along with error bounds for the resulting approximations. Further, we apply the offline-online procedure problem to the solution of 1-dimensional PDE filtering problem for the heat equation, the wave equation and a 2-dimensional pollutant transport equation. We also compare the performance of the I-POD based filter with that of the full order Kalman filter.

2. Preliminaries. In this work, we are interested in the filtering of distributed parameter systems, systems whose evolution is governed by a partial differential equation (PDE), given sparse measurements of the spatial-temporal field variable. Mathematically, we are interested in estimating the state of the field variable $X \in H$, for some suitable Hilbert space H. The state X is governed by the operator equation

$$\dot{X} = \mathcal{A}X + \mathcal{W}, \tag{2.1}$$

where $\mathcal{W}$ is a spatially distributed Gaussian white noise process perturbing the motion of the system [32]. We assume that the boundary conditions for the PDE are known. We do not have access to measurement of the entire states, instead we only have access to measurements of the field at some sparse set of spatial locations in the domain of the process given by

$$Y(t_k, x_j) = \mathcal{C}X(t_k) + \mathcal{V}_k^{(j)}, \tag{2.2}$$

where $X(t_k)$ represents the state at the discrete time instant t_k, and $Y(t_k, x_j)$ represents a localized measurement of the state variable at the sparse set of locations given by x_j, $j = 1, \cdots m$, and $\mathcal{V}_k^{(j)}$ is a discrete time white noise process corrupting the measurements at the spatial location x_j. We assume that the differential operator $\mathcal{A}$ is self adjoint with a compact resolvent, and thus, $\mathcal{A}$ has a discrete spectrum with a full set of eigenvectors that forms an orthonormal basis for the Hilbert space H. Further, we assume that the operator $\mathcal{A}$ generates a stable semigroup. In Section IV, we extend the results to non self-adjoint operators.

Given the above assumptions, we can discretize the PDE above using computational techniques such as finite elements (FE)/ finite difference (FD) to obtain a discretized version of the operator equations in Euclidean space $\Re^N$ given by the following:

$$\begin{aligned} x_k &= Ax_{k-1} + w_k, \\ y_k &= Cx_k + v_k, \end{aligned} \tag{2.3}$$

where x_k is a discretized version of the state X that resides in the high dimensional Euclidean space $\Re^N$, while w_k and v_k are discretized versions of the white noise process corrupting the state and measurement equations respectively. Given that the operator $\mathcal{A}$ is self-adjoint, the discretized operator $\mathcal{A}$ is (usually) a symmetric matrix with a full set of eigenvectors and real eigenvalues whose eigenvectors form an orthonormal basis of $\Re^N$ and for suitable large N, are arbitrarily good approximations of the true eigenfunctions of the original operator $\mathcal{A}$. We relax the self-adjoint assumption later in this section. We shall assume throughout this paper that a fine enough discretization is given to us and thus, the behavior of the original system, in terms of the trajectories of the system, is captured sufficiently well by the discretized version of the system. Thus, in the rest of the paper, we only consider the discretized version of the problem. Given the above discretization, a naïve approach to the solution of the filtering problems is to use a standard Kalman filter to solve the problem. The standard Kalman filter problem is given by the following [33]:

Given that x_0, v_k, w_k are jointly gaussian and mutually independent; x_0 is $N(\overline{x}_0, P_0)$, v_k is zero mean, covariance $R_k\delta_{kl}$; w_k is zero mean, covariance $Q_k\delta_{kl}$.

Kalman Filter:

$$\begin{aligned}\hat{x}_{k+1/k} &= (A - K_kC)\hat{x}_{k/k-1} + K_ky_k, \hat{x}_{0/-1} = \overline{x}_0\\ K_k &= \Sigma_{k/k-1}C'(C\Sigma_{k/k-1}C' + R_k)^{-1}\\ \Sigma_{k+1/k} &= A[\Sigma_{k/k-1} - \Sigma_{k/k-1}C'(C\Sigma_{k/k-1}C' + R_k)^{-1}C\Sigma_{k/k-1}]A' + Q_k\\ \hat{x}_{k/k} &= \hat{x}_{k/k-1} + \Sigma_{k/k-1}C'(C\Sigma_{k/k-1}C' + R_k)^{-1}(y_k - C\hat{x}_{k/k-1})\\ \Sigma_{k/k} &= \Sigma_{k/k-1} - \Sigma_{k/k-1}C'(C\Sigma_{k/k-1}C' + R_k)^{-1}C\Sigma k/k - 1 \qquad (2.4)\end{aligned}$$

Due to the very high dimensionality of $\Re^N$, since N can easily run into millions of degrees of freedom (DOF) for a general finely discretized PDE, the Kalman filtering equations are computationally intractable for such high DOF systems. For example, for pollutant transport equation, with $N = 2500$, Σ_k is a $2500 * 2500$ matrix, and we need to compute the inverse of this matrix in the full order Kalman filter at every time step. If we use a ROM, then Σ_k is only a $30 * 30$ matrix, thereby reducing the Kalman filter computations by several orders of magnitude. In general, the computational complexity of the Kalman filter is $\mathcal{O}(N^3)$ where N is the order of the system. Thus, if the order of the ROM $N_r << N$, the order of the full order system, then the computational gains are highly significant. Thus, we need to first suitably reduce the order of the system before we apply Kalman filtering techniques to the above problem.

3. An Iterative Approach to Proper Orthogonal Decomposition (I-POD). Consider the following linear system:

$$x_k = Ax_{k-1}, \text{ given } x(0), \qquad (3.1)$$

where $x_k \in \Re^N$, N is very large and A is a symmetric matrix. Recall that the above high dimensional linear system results from the discretization of a self adjoint linear operator.

A 1. *We assume that there is a unique null vector corresponding to A and that the matrix A is Hurwitz, i.e., the system is stable.*

Suppose that we choose some arbitrary initial condition $x(0)$ and take M snapshots of the system's trajectory at the time instants $t_1 < t_2 < \cdots < t_M$, where these snapshots need not be equi-spaced. Let us denote the data matrix of the stacked snapshots by X, i.e.,

$$X = [x_1, x_2, \cdots, x_M],$$

where $x_i = x(t_i)$. Suppose now that the number of snapshots is much smaller than the dimension of the system, i.e., $M << N$. Then, using the snapshot POD technique, we can get the POD basis T of the trajectory encoded in the snapshot ensemble X as follows:

$$T = XV_p\Sigma_p^{-1/2}, \tag{3.2}$$

where V_p and Σ_p are the eigenvector-eigenvalue pair corresponding to the correlation matrix $X'X$, i.e.,

$$(X'X)V_p = V_p\Sigma_p. \tag{3.3}$$

Note that the $M \times M$ eigenvalue problem to be solved for the POD eigenfunctions is much easier than the high dimensional $N \times N$ eigenvalue problem that needs to be solved if we were interested in solving for the eigenvalues and eigenvectors of A. Given the snapshot POD eigenfunctions, we can obtain a reduced order approximation of the system in Eq. 3.1 as follows:

$$\psi_k = (T'AT)\psi_{k-1} \equiv \tilde{A}\psi_{k-1}, \tag{3.4}$$

where ψ represents the projection of the system state onto the POD eigenfunctions and $\tilde{A}$ represents the reduced order $M \times M$ system matrix.

Consider the reduced order system matrix $\tilde{A}$. We know that $\tilde{A}$ is symmetric and thus, has a full eigenvalue decomposition. Let (Λ, P) represent the eigenvalue-eigenvector pair for $\tilde{A}$, i.e.,

$$\tilde{A}P = P\Lambda. \tag{3.5}$$

Noting that $\tilde{A} = P\Lambda P'$, the ROM matrix $\tilde{A}$ transformed to the orthonormal co-ordinates specified by P, can be represented in the modal co-ordinates ϕ as:

$$\phi_k = \Lambda\phi_{k-1}. \tag{3.6}$$

Thus it follows that

$$\Lambda = (P'T')A(TP), \tag{3.7}$$

where T is the POD eigenfunction matrix and P is the ROM eigenfunction matrix. Note that T is $N \times M$ and that P is $M \times M$, and hence, TP is $N \times M$. The above equation looks like an eigendecomposition of the matrix A except that $M << N$ and thus, this is not necessary. Note that the transformation TP denotes the transformation from the original state space to the POD eigenfunction space to the ROM eigenfunction space. In the following, we relate the eigenvalues and eigenvectors of A to the diagonal form Λ and the transformation TP.

A 2. *Assume that "p" eigenvectors of the matrix A are active in the snapshot ensemble X, i.e.,*

$$x_i = \sum_{j=1}^{p} \alpha_j^i v_j, i = 1, 2, ...M,$$

where $p \leq M$ and without loss of generality, it is assumed that the active eigenvectors consist of the first "p" eigenvectors. This assumption essentially implies that the number of modes active within the snapshots is less than the number of snapshots in the ensemble.

Under assumption A1 and A2, the following result is true.

Proposition 3.1. *The columns of the transformation TP are the eigenvectors of A with corresponding eigenvalues encoded in the diagonal matrix Λ, i.e.,*

$$A(TP) = (TP)\Lambda. \tag{3.8}$$

Proof. Recall that $T = XV_p\Sigma_p^{-1/2}$. We have

$$X = V\alpha = [v_1, v_2, \cdots v_p]\begin{bmatrix} \alpha_1^1 & .. & \alpha_1^M \\ . & .. & . \\ \alpha_p^1 & .. & \alpha_p^M \end{bmatrix},$$

where note that V is an $N \times p$ and α is an $p \times M$ matrix. First, consider the case when $p = M$. Then, it follows that

$$\tilde{A} = T'AT = \Sigma_p^{-1/2}V_p'X'AXV_p\Sigma_p^{-1/2} = \underbrace{(\Sigma_p^{-1/2}V_p'\alpha')}_{\text{M*M}}\underbrace{(V'AV)}_{\text{M*M}}\underbrace{(\alpha V_p\Sigma_p^{-1/2})}_{\text{M*M}}.$$

Comparing this to the fact that $\tilde{A} = P\Lambda P'$, it follows that

$$P = \Sigma_p^{-1/2}V_p'\alpha' \equiv \tilde{P}, \tag{3.9}$$

if $\tilde{P}\tilde{P}' = I$. To show this, note that

$$\tilde{P}\tilde{P}' = \Sigma_p^{-1/2}V_p'(\alpha'\alpha)V_p\Sigma_p^{-1/2}. \tag{3.10}$$

Further, $\alpha'\alpha = X'X = V_p\Sigma_pV_p'$, and hence, noting the orthogonality of the columns of V_p, it follows that:

$$\tilde{P}\tilde{P}' = \Sigma_p^{-1/2}V_p'V_p\Sigma_pV_p'V_p\Sigma_P^{-1/2} = I. \tag{3.11}$$

Further, it follows that

$$TP = (XV_p\Sigma_p^{-1/2})(\Sigma_p^{-1/2}V_p'\alpha') = V(\alpha V_p\Sigma_p^{-1/2})(\Sigma_p^{-1/2}V_p'\alpha') = V\tilde{P}'\tilde{P} = V \tag{3.12}$$

i.e., the columns of TP are indeed eigenvectors of A. Moreover, it also follows that owing to the uniqueness of the similarity transformation of $\tilde{A}$ that the eigenvalues corresponding to the eigenvectors in TP are in the diagonal form Λ. Hence, this proves our assertion for the case when $p = M$.

When $p < M$, the rank of the snapshot ensemble X is $p < M$ and hence, the rank of $X'X$ is $p < M$. Thus, it follows that the POD eigenvalues will be non-zero for only p POD eigenfunctions. Therefore, the transformation into the POD basis $T = XV_p\Sigma_p^{-1/2}$ should only include the POD eigenvectors corresponding to the non-zero eigenvalues. This implies that $X'X = \alpha'\alpha = \tilde{V}_p\tilde{\Sigma}_p\tilde{V}_p'$, where $\tilde{\Sigma}_p$ contains the non-zero POD eigenvalues, and $\tilde{V}_p$ contains the corresponding eigenvectors. Further, $T = X\tilde{V}_p\tilde{\Sigma}_p^{-1/2}$, and hence, $\tilde{P}$, as defined in Eq. 3.9, is now $p \times p$. Thus, the analysis above goes through unchanged, and hence, $\tilde{P}\tilde{P}' = I$, and $TP = V$, where V now consists of the p active eigenvectors. □

At this point, we make the following remark.

REMARK 1. *Suppose that $p > M$, i.e., the number of active eigenvectors are more than the number of snapshots. WLOG, let $p = N$. Then*

$$\tilde{A} = T'AT = \Sigma_p^{-1/2}V_p'X'AXV_p\Sigma_p^{-1/2} = (\Sigma_p^{-1/2}V_p'\alpha')V'AV(\alpha V_p\Sigma_p^{-1/2})$$
$$= \underbrace{(\Sigma_p^{-1/2}V_p'\alpha')}_{M*N}\underbrace{\Lambda}_{N*N}\underbrace{(\alpha V_p\Sigma_p^{-1/2})}_{N*M} = \underbrace{\beta'}_{M*M}\underbrace{\Gamma}_{M*M}\underbrace{\beta}_{M*M},$$

where (β, Γ) represents the eigenvalue decomposition of the ROM matrix $\tilde{A}$. Note that now owing to the fact that $N > M$, we can no longer use the uniqueness of the similarity transformation of $\tilde{A}$ to conclude that the transformation $T\beta$ contains the eigenvectors of A.

The above proposition and the remark above suggest a technique through which eigenvectors of the system matrix A can be extracted up to any time-scale. First, we make the following assumption.

A 3. *We assume that there are K characteristic timescales embedded in the matrix A, namely $T_1, \cdots T_K$. Let the eigenvalues corresponding to timescale T_j be $\{\lambda_1^{(j)}, \cdots \lambda_{M_j}^{(j)}\}$ and let the corresponding eigenvectors be $[v_1^{(j)}, \cdots v_{M_j}^{(j)}] \equiv V^{(j)}$, here, M_j is the number of active eigenvectors in the jth timescale. Further, we assume that the timescales are well-separated, i.e., if for some t_k, ${(\lambda_k^{(j)})}^{t_k} \neq 0$, then ${(\lambda_k^{(i)})}^{t_k} \approx 0$ for all $i < j$. The above assumption essentially implies all the eigenvectors corresponding to timescales below a given timescale decay well before the eigenvectors at the given timescale decay.*

At this point, we also need to make sure that all possible eigenfunctions corresponding to any timescale are excited. Under the assumption A1 and A2, the following result assures us of this:

PROPOSITION 3.2. *Let the initial condition $x^{(j)}(0)$ to the linear system in Eq. 3.1 be chosen according to a Gaussian distribution $N(0, \sigma^2 I)$. Then, every eigenvector of A is excited almost surely in at least one of the trajectories $X^{(j)}$.*

Proof. Due to the eigenvalue decomposition of A, we may write:

$$x_k = \sum_{i=1}^{N} (\lambda_i)^k (x(0), v_i) v_i,$$

where $(.,.)$ denotes the inner product in $\Re^N$. The above implies that $(x_k, v_i) = \lambda_i^k (x(0), v_i)$, and hence

$$E|(x_k, v_i)|^2 = (\lambda_i)^{2k} E|(x(0), v_i)|^2 = (\lambda_i)^{2k} v_i' P_0 v_i = \sigma^2 (\lambda_i)^{2k}, \tag{3.13}$$

here, $P_0 = E[x(0)x(0)^T] = \sigma^2 I$. Thus, the i^{th} component of the system trajectory, i.e., the contribution of the i^{th} eigenvector, is a Gaussian random variable with zero-mean and a variance that geometrically decays in time as shown above. Thus, the i^{th} mode is bound to be active for at least one among the ensemble of trajectories. In fact, owing to the Gaussian nature of the component, it is true that its absolute value will be above any given threshold, at any given time, with a finite probability. □

Given the results above and assumption **A3**, we are in a position to outline a procedure that allows us to isolate all eigenfunctions at any given timescale.

Suppose without loss of generality that $T_1 > T_2 \cdots > T_K$. Suppose now that we are interested in isolating all the eigenfunctions corresponding to the timescale T_1. We choose an initial time $t_0^{(1)}$ and subsequent snapshot times $t_n^{(1)}$, $n = 1 : M$, such that $M > M_1$ and such that the initial time $t_0^{(1)} >> T_2$. Thus, the snapshot timing assures us that all the eigenvectors at the timescales below T_1 will have decayed by the snapshot times of interest, and thus, the only participating modes are the eigenfunctions corresponding to timescale T_1. Then, using Propositions 1 and 2, we know that we can isolate all the eigenfunctions at the timescale T_1 given enough snapshot ensembles. In particular, suppose that $X_j^{(1)}$ is the j^{th} snapshot ensemble at timescale T_1. Due to proposition 2, as $j \to \infty$, we know that every eigenfunction in set $V^{(1)}$ is bound to be excited. Further, due to the fact that $M > M_1$, it follows using Proposition 1 that the eigenfunctions of the ROM are the same as the eigenfunctions of A. Thus, every snapshot ensemble gives us some of the eigenvectors $v \in V^{(1)}$ and as $j \to \infty$, we are assured that all possible $v \in V^{(1)}$ are recovered.

Given that we have recovered all the eigenfunctions $V^{(1)}$ corresponding to the longest timescale T_1, we can now iteratively recover all the eigenfunctions at all the subsequent timescales as follows. Given $V^{(1)}$, we randomly choose an initial condition $x(0)$ and form the snapshot ensemble X at snapshot times $t_0^{(2)}, \cdots t_M^{(2)}$, such that number of snapshots $M > M_2$, and the initial time of the snapshot $t_0^{(2)} >> T_3$, i.e., such that all eigenfunctions at timescales shorter than T_2 are absent in the ensemble. Given the snapshot ensemble X we subtract the contributions of the eigenfunctions in $V^{(1)}$, i.e.,

$$\tilde{x}(t_j^{(2)}) = x(t_j^{(2)}) - \sum_{k=1}^{M_1} (\lambda_k^{(1)})^{t_j^{(2)}} (x_0, v_k^{(1)}) v_k^{(1)}. \tag{3.14}$$

Consider the modified snapshot ensemble $\tilde{X}$, it follows that $\tilde{X}$, by construction, only contains eigenfunctions from the set $V^{(2)}$ and thus, following the randomly perturbed POD procedure outline previously, we can recover all the eigenfunctions in $V^{(2)}$. Given $V^{(1)}$ and $V^{(2)}$, we can repeat the removal, and randomly perturbed POD procedure, to recursively obtain all the sets $V^{(n)}$ up to any desired timescale T_n. The above development can be summarized in the following algorithm:

The development above and the I-POD algorithm can be summarized in the following result.

Proposition 3.3. *Under assumptions **A1-A3**, the I-POD algorithm extracts all eigenfunctions $V^{(i)}$ corresponding to any given time scale $T^{(i)}$.*

Remark 2. *In practice, we choose the timescales as following: first, we choose a large enough time so that most of the system eigenfunctions have decayed, then take several snapshots and extract the eigenfunctions corresponding to this timescale T_1. After subtracting these eigenfunctions from the trajectory, we can still use the same timescale T_1, i.e, the she snapshots, or guess a start time of T_2 ($T_2 \leq T_1$), and then use the same, or different snapshots, respectively, to extract the next set of eigenfucntions. In practice, we use the same snapshots to extract eigenvectors until we cannot extract any more in this timescale, before we move to the next timescale.*

Algorithm 1 Algorithm I-POD

```
1. Given timescales $T_1, \cdots T_K$
2. Set $i = 1$, $V^{(0)} = \phi$ (empty set)
3. WHILE $i \leq K$
   DO
    (a) Choose snapshot times $t_0^{(i)}, \cdots t_M^{(i)}$, such that $t_0^{(i)} >> T_{i+1}$ and $M > M_i$
    (b) Set $j = 1$
          i. Choose $x_{0,j}^{(i)}$, the initial condition of the $j^{th}$ snapshot ensemble at
             time scale $T_i$, from $N(0, \sigma^2 I)$ and generate the $j^{th}$ snapshot ensemble $X_j^{(i)}$
         ii. Substract all the slower eigenfunctions from the snapshot ensemble
             using Eq. 3.14, and the previously extracted eigenfunctions from
             the sets $V^{(1)}, V^{(2)}, \cdots V^{(i-1)}$
        iii. Isolate the eigenfunctions at timescale $T_i$, $V^{(i)}$, using the snapshot
             POD. Set $j = j + 1$
         iv. If all eigenfunctions in $V^{(i)}$ have been obtained, go to step (c), else
             go to step (i)
    (c) Set $i = i + 1$
4. Output the eigenfunctions in sets $V^{(1)}, \cdots V^{(K)}$
```

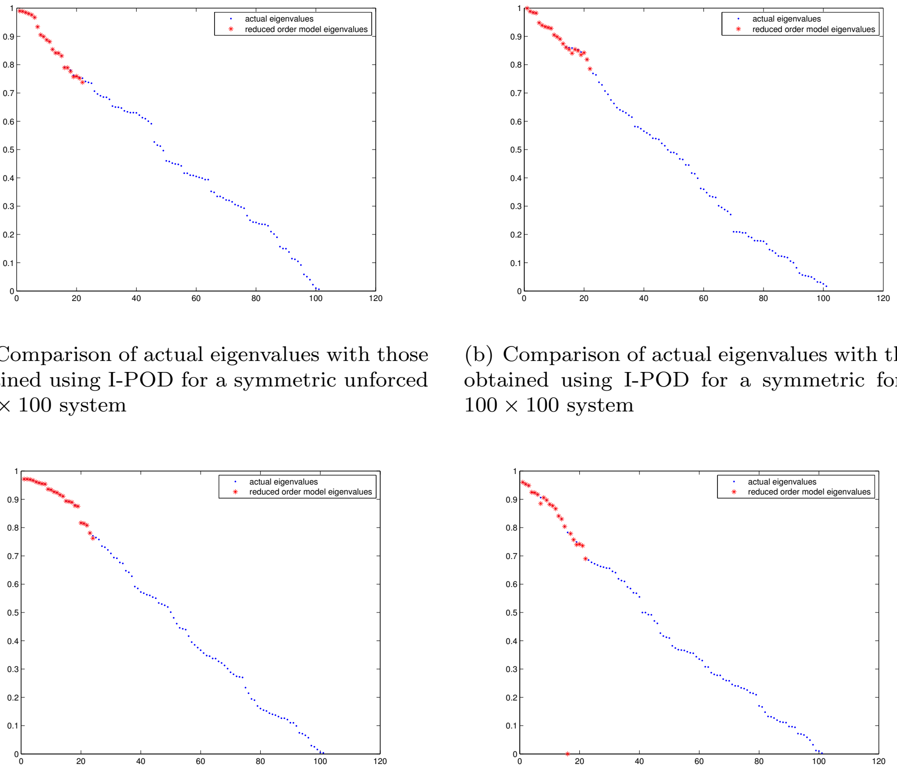
Remark 3. *The timescales $T_1, \cdots T_k$ are dependent on the Physics and can be inferred from physical insight or simulations. The number of snapshots that are required to extract the eigenfunctions have to be "large enough". Of course, it might not be possible to know a priori when M is large enough. However, some amount of trial and error can tell us as to what is a suitable number for M. In fact, a good heuristic measure is to wait long enough before the first snapshot, such that most modes have decayed and we have lesser number of modes participating than the number of snapshots. In fact, this is a heuristic that is often used in the POD literature [8]. This can easily be construed from the eigenvalue decomposition of the snapshot ensemble by checking for zero eigenvalues and eigenvectors, i.e., rank deficiency of the snapshot ensemble.*

We also note that though theoretically we can extract eigenfunctions at all time scales, due to small numerical errors that accumulate, practically, we may only be able to extract the eigenfunctions corresponding to the first few timescales. In most applications, these first few timescales are typically enough to form a good ROM (for instance, please see the examples in Section 4).

The I-POD technique is a **completely data based technique** and does not need knowledge of the system matrix A. Note that ultimately, the ROM $\tilde{A} = T'AT$, contains all the information regarding the eigenfunctions of the operator A under the assumptions above. Again note that $T = XV_p\Sigma_p^{-1/2}$, and thus, it follows that $\tilde{A} = T'AXV_p\Sigma_p^{-1/2} = T'\tilde{X}V_p\Sigma_p^{-1/2}$, where $\tilde{X}$ is the one time step advanced version of the snapshot ensemble X (in the discrete time case), and can be obtained directly from simulation or experimental data.

It should also be noted that Proposition 1 **does not distinguish between forced systems and unforced systems** since Assumption 2 under which the result is valid only asks for certain sufficient conditions on the active eigenfunctions of the system in the snapshot ensemble. Since the forced response of a linear system is also expressed in terms of the eigenfunctions, the I-POD procedure is valid for forced systems as well as long as Assumption 2 is valid. Hence, the procedure can be used on experimental data, where the system response may be forced. Of course, the issue is that Assumption 2 underlying Proposition 1 may not be satisfied for forced systems. However, in our experiments we do see that this assumption is indeed satisfied and that we can actually extract the eigenfunctions of the forced system using the I-POD procedure (this can also be seen from the results in [8]).

Representative results from our experiments are shown in Figure 3.1. In Figure 3.1(a) and Figure 3.1(b), we compare the actual eigenvalues of a randomly generated 100×100 system with those obtained by the I-POD procedure, for an unforced as well as a forced (constant forcing) symmetric system. The results show that the I-POD eigenvalues agree very well with the actual eigenvalues.



(a) Comparison of actual eigenvalues with those obtained using I-POD for a symmetric unforced 100×100 system

(b) Comparison of actual eigenvalues with those obtained using I-POD for a symmetric forced 100×100 system

(c) Comparison of actual eigenvalues with those obtained using I-POD for a non-symmetric unforced 100×100 system

(d) Comparison of actual eigenvalues with those obtained using I-POD for a non-symmetric forced 100×100 system

FIG. 3.1. *Eigenvalues extraction results using I-POD*

3.1. Extension to Non Self-adjoint Operators. In the following, we show how the I-POD technique can be extended to non self-adjoint operators. Again, we restrict our attention to a suitably fine discretization of the underlying infinite dimensional systems given by a large scale matrix A. Suppose that X is the snapshot data from the simulation of the response (forced/ unforced) of the matrix A. Let V_p and Σ_p be the eigenvectors and eigenvalues resulting form the eigendecomposition of the snapshot data $X'X$, and let $T = XV_p\Sigma_p^{-1/2}$ be the corresponding POD transformation. The reduced order model is then given by $\tilde{A} = T'AT$. Assuming that the ROM can be diagonalized, we write the similarity transformation for the ROM, $\tilde{A} = P\Lambda P^{-1}$. Given that assumption 2 holds (the number of snapshots are greater than the number of active eigenvectors), we have the following result.

Proposition 3.4. *The eigenvalues of the ROM $\tilde{A}$, given by the diagonal matrix Λ, are also eigenvalues of the full order model A, and the corresponding eigenvectors are given by the transformation TP.*

Proof. Let $X = V\alpha$, as before, where V denotes the active eigenvectors of A in the snapshots, and α is the coefficient matrix of the eigenvectors for all the snapshots. Let the number of active eigenvectors be equal to the number of snapshots. Then

$$\tilde{A} = \Sigma_p^{-1/2} V_p' \alpha' V' A V \alpha V_p \Sigma_p^{-1/2} = \overbrace{\Sigma_p^{-1/2} V_p' \alpha' V' V}^{\tilde{P}'(V'V)} \tilde{\Lambda} \overbrace{\alpha V_p \Sigma_p^{-1/2}}^{\tilde{P}} = P\Lambda P^{-1}, \tag{3.15}$$

where $\tilde{\Lambda}$ are the eigenvalues of A corresponding to the eigenvectors V. Thus, if we show that $\tilde{P}$ is the inverse of $\tilde{P}'(V'V)$, then due to the uniqueness of the similarity transformation of $\tilde{A}$, it follows that $\tilde{P} = P^{-1}$ and $\Lambda = \tilde{\Lambda}$. To show this, note that:

$$\tilde{P}'(V'V)\tilde{P} = \Sigma_p^{-1/2} V_p' \alpha' (V'V) \alpha V_p \Sigma_p^{-1/2}. \tag{3.16}$$

Here $\alpha'(V'V)\alpha = X'X = V_p\Sigma_p V_p'$, and therefore, using the orthogonality of the columns of V_p, it follows that

$$\tilde{P}'(V'V)\tilde{P} = \Sigma_p^{-1/2} V_p' V_p \Sigma_p V_p' V_p \Sigma_p^{-1/2} = I. \tag{3.17}$$

Hence, $\tilde{P}$ and $\tilde{P}'(V'V)$ are inverses of each other. It can also be easily shown that $TP = V$. Further, the case when the number of snapshots is greater than the number of active eigenvectors can be treated identically to the symmetric case considered in Proposition 1. □

In the non-symmetric case, $P^{-1}T'$ does not contain the left eigenvectors of A as was the case for symmetric systems. In fact, $P^{-1}T'$ is the pseudo-inverse of $TP = V$, i.e., $P^{-1}T' = (V'V)^{-1}V'$. Thus, even though we know the right eigenvectors through the POD transformation TP, we do not have knowledge of the left eigenvectors from POD.

In order to gain knowledge of the left eigenvectors, we need to generate data Y from the adjoint simulations, i.e., using matrix A'. Using this adjoint simulation data Y, the left eigenvectors of A, which are the same as the right eigenvectors of A' up to a multiplicative constant, can be found using the I-POD procedure. In other words, the POD is used to get the right eigenvectors of A' using adjoint simulation data Y thereby providing us knowledge of the left eigenvectors of A. Further, random initial conditions can be used to generate the eigenvalues, as well as the left and right eigenvectors, using the simulation data from A and its adjoint A', in a sequential fashion. Finally, the left/ right eigenvectors corresponding to the different time scales may also be extracted sequentially, identical to the symmetric case. Note that in the non-symmetric case, the left eigenvectors are necessary to extract the eigenfunctions at the different time scales using the sequential I-POD procedure, since the initial amplitudes of the different eigenmodes are given by the projection of the initial conditions on the corresponding left eigenmodes (note that the DMD cannot do the same since it does not utilize the adjoint information).

In Figure 3.1(c) and Figure 3.1(d), we show the comparison of the eigenvalues obtained using I-POD with the true eigenvalues for an unforced non-symmetric system as well as a forced non-symmetric system. In fact, we can find as many eigenvalues as we want, here we only show the first several dominant eigenvalues.

4. I-POD using input/ output description. In the following, we show how to extract the left and right eigenvectors of the system using POD, and the input/ output description of the system, as opposed to the random initial conditions that have been used so far. We shall also compare and contrast the I-POD with two closely related approaches, the Balanced POD (BPOD), and the Dynamic Mode Decomposition (DMD) approach.

For simplicity, we shall consider a discrete time system (due to the technical problems of dealing with impulse responses in continuous time systems), and also, for ease of comparison to the related BPOD and DMD approaches:

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_{k-1}, \\ y_k &= Cx_k. \end{aligned} \tag{4.1}$$

Let the input influence matrix be denoted by $B = [b_1, \cdots b_p]$ and the output matrix by $C = [c_1, \cdots c_m]'$.

In this case, the columns of the input influence matrix, b_j, serve as the initial conditions for the simulation of the system A, leading to the snapshot ensemble X_j while the transposed rows of the output matrix, c_i', serve as the initial conditions for the simulations of the adjoint system A', leading to the adjoint snapshot ensemble Y_i. The first set of simulations in terms of the snapshot ensembles X_j leads to a set of right eigenvalue-eigenvectors (Λ_r, V_r), while the adjoint simulations (Y_i) lead to a set of left eigenvalues and eigenvectors denoted by (Λ_l, V_l). In particular, in order to obtain the pair (Λ_r, V_r), we do the snapshot POD procedure, followed by diagonalization of the ROM, on every trajectory X_j sequentially. Thus, corresponding to every snapshot ensemble X_j, we will obtain the corresponding set of eigenvalues/ eigenvectors (Λ_r^j, V_r^j). For every subsequent snapshot ensemble X_j, we only add the eigenvalues/ eigenvectors that are not already extracted from X_k, $k = 1, ..., j-1$. A similar procedure can be used to extract the left eigenvalue/ eigenvector pair (Λ_l, V_l). Of these eigenvalue-eigenvectors pairs, we only keep those left/ right eigenvectors that correspond to the eigenvalues in the intersection of the set of eigenvalues in Λ_l, and in Λ_r. Thus, this allows us to keep the modes of the system that are both observable and controllable.

Suppose we discard the POD modes corresponding to the small singular values of the POD, then, quantitatively, we retain only the most observable and most controllable modes of the system (A, B, C) in V_l and V_r respectively. If we only consider those eigenvectors that are in the intersection of Λ_r and Λ_l as mentioned above, then we include the modes of the system that are most observable as well as controllable, in the ROM. However, it is not necessary that this method result in a balanced decomposition. Thus, we can see that the I-POD can sequentially extract the eigenmodes present in a single trajectory as well as iteratively extract the eigenmodes from different simulation trajectories.

Another way of extracting the controllable/ observable eigenmodes corresponding to the input/ output description is to perform an SVD on the cross correlation between the primal and the adjoint simulations, i.e., an SVD of $Y_i' X_j$ for all i, j. In particular, we write:

$$\begin{aligned} Y_i' X_j &= U_p \Sigma_p V_p', \\ \tilde{A} &= (\Sigma_p^{-1/2} U_p' Y_i') A (X_j V_p \Sigma_p^{-1/2}) = P \Lambda_{ij} P^{-1}, \\ \Lambda_{ij} &= \underbrace{(P^{-1} \Sigma_p^{-1/2} U_p' Y_i')}_{\Phi_{ij}'} A \underbrace{(X_j V_p \Sigma_p^{-1/2} P)}_{\Psi_{ij}}. \end{aligned} \tag{4.2}$$

In the above, it can be shown that Λ_{ij} contains the most observable and controllable eigenmodes present in the responses Y_i and X_j. However, it is not necessarily true that Φ_{ij} and Ψ_{ij} contain the most observable and controllable left and right eigenvectors respectively. However, the left and right eigenvectors corresponding to the eigenvalues in Λ_{ij} can be found by doing the eigenvalue decomposition of $Y_i' Y_i$ and $X_j' X_j$ respectively. In the experiments that follow, however, we use Φ_{ij} and Ψ_{ij} as the left/ right eigenvectors.

4.1. Comparison with Balanced POD. In the following, we assume that the system of interest has m outputs and p inputs, and we have N snapshots for each input/ output trajectory. In this case, the BPOD has to solve the SVD problem of a matrix of dimension (mN) x (pN). The I-POD on the other hand solves $(m + p)$ eigenvalue problems of dimension NxN. In case we use cross-correlations, we would have to solve (mp) SVD problems of dimension NxN. Hence, it can be seen that the I-POD would be computationally much more efficient than the BPOD if the number of inputs/ outputs of a system is large. This is because the BPOD requires to solve an SVD problem for a matrix formed by stacking all input/ output correlation matrices $Y_i' X_j$ in a large Hankel matrix, whereas the I-POD uses the eigendecomposition of $Y_i' Y_i$, and $X_j' X_j$ (or $Y_i' X_j$) sequentially, and can iteratively build up the ROM due to the invariance of the underlying eigenmodes to the different trajectories of the system.

In the following, two examples comparing the Balanced POD and I-POD are shown. In the examples, we define the output error as:

$$E_{output} = \|Y_{true} - Y_{red}\|, \tag{4.3}$$

where Y_{true} are the outputs of the true system and Y_{red} are the outputs of the reduced order system.
The state error is defined as:

$$E_{state} = \|X_{true} - X_{red}\|, \tag{4.4}$$

where X_{true} are the states of the true system and X_{red} are the states of the reduced order system.

4.1.1. Heat Equation. The heat transfer by conduction along a slab is given by the partial differential equation:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + f \tag{4.5}$$

The length of the slab is $5m$ and the continuous spatial domain X is divided into 100 grid cells of equal length. The model is simulated for a period of 100 minutes and the time horizon is discretized into 50000 time steps, α is thermal diffusivity, and takes the value 0.01. f is the constant source in the middle of the slab, and f takes the value 50 $^\circ C/minute$.

The boundary condition is: $T(0,t) = T(L,t) = 0, \forall t > 0$. The initial condition is 200 °C along the slab. The temperature of the slab is measured at five equi-spaced points along its length.

For Balanced POD, 100 snapshots from the first 3000 time steps are collected, the snapshots are equi-spaced. We use 20 modes to construct the Balanced POD based ROM. For I-POD, we use the same initial conditions (input/ output description of the system) as Balanced POD. We also collect the snapshots from the first 3000 time steps, but in 6 timescales. Using the I-POD technique, we can collect more modes sequentially, and we use 30 modes to construct the I-POD based ROM. The comparison between the Balanced POD and I-POD is shown in Figure 4.1. In Figure 4.1(a)-(b), we compare the eigenvalues extracted by I-POD and Balanced POD (i.e., eigenvalues of the ROM constructed by BPOD) .

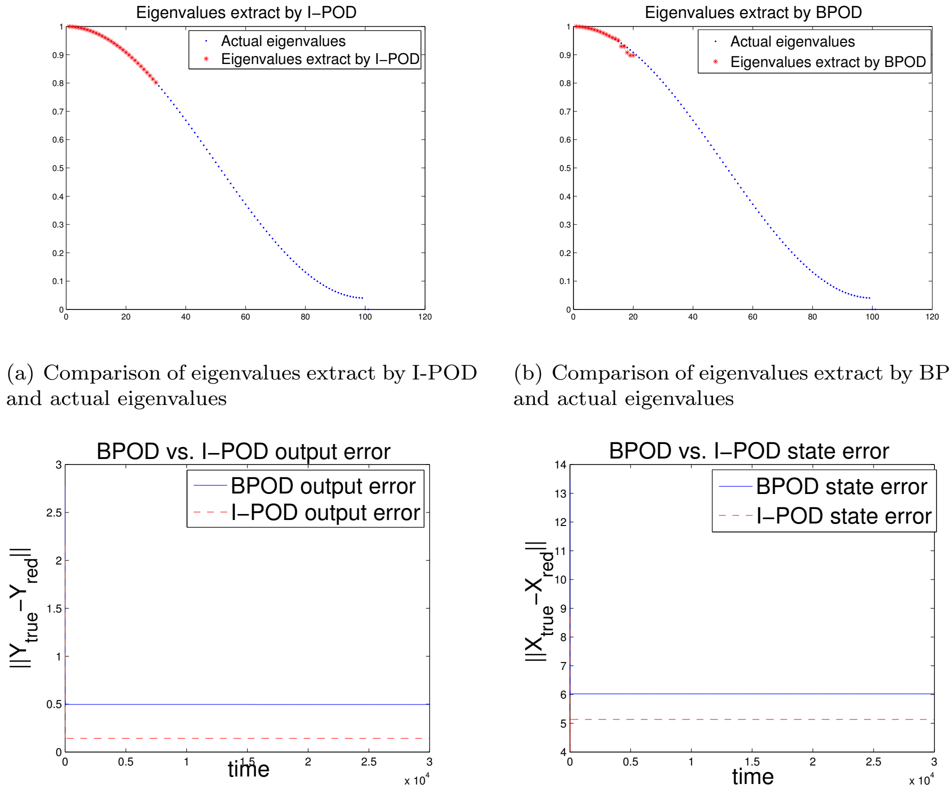
To test the ROM, we use 20 different random initial conditions and take the average output/state error over these 20 simulation. In Figure 4.1(c), we compare the output errors of the two algorithm. In Figure 4.1(d), we compare the state errors of the two algorithm.

4.1.2. Pollutant Transport Equation. The two-dimensional pollutant transport equation describe the contaminant transport is:

$$\frac{\partial c(x,y,t)}{\partial t} = D_x \frac{\partial^2 c(x,t)}{\partial x^2} + D_y \frac{\partial^2 c(x,t)}{\partial y^2} - v_x \frac{\partial c(x,t)}{\partial x} - v_y \frac{\partial c(x,t)}{\partial y} + S_s, \tag{4.6}$$

where c is concentration of the contaminant, D is dispersion and takes value 0.6 here, v is velocity in the x and y directions and takes value 1, and S_s is source of pollutant.

In simulation, there are three obstacles and three sources in the field. The initial condition for simulation is zero. We use Neumann boundary conditions. Ten measurements are taken equi-spaced along the diagonal of the field. The actual field at the end of simulation is shown in Figure 4.2. The field is discretized into a 50$*$50 grid.



(a) Comparison of eigenvalues extract by I-POD and actual eigenvalues

(b) Comparison of eigenvalues extract by BPOD and actual eigenvalues

(c) Comparison of output errors

(d) Comparision of state errors

Fig. 4.1. *Comparison between BPOD and I-POD for Heat equation*

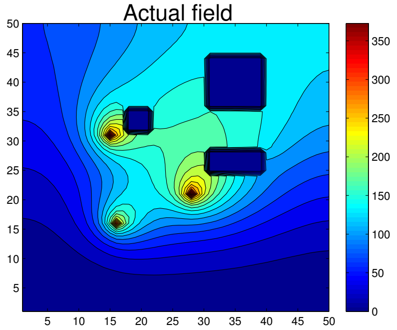
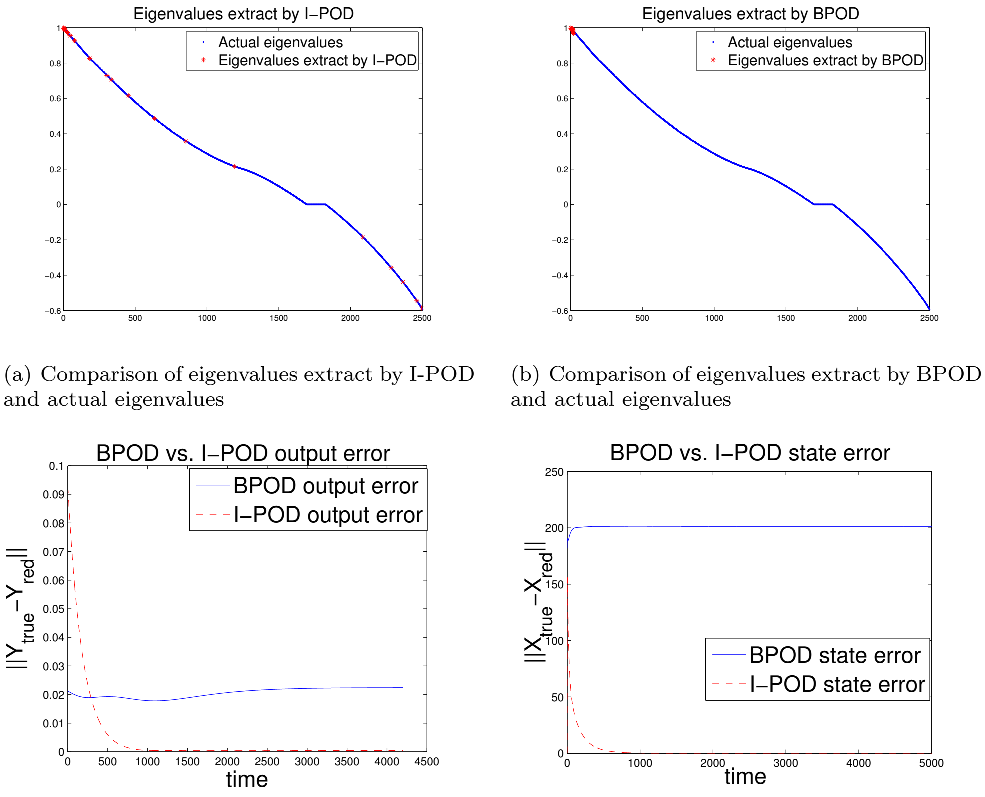


Fig. 4.2. *Actual field at the end of simulation for pollutant transport equation*

For Balanced POD, we collect the 100 snapshots from the first 3000 time steps. The snapshots are equi-spaced, and we can construct the BPOD based ROM using 22 modes. Note that we can solve the singular value decomposition problem using ten measurements and three inputs for this problem, but we are not able to solve the large singular value decomposition problems as the measurements increase, for instance, if we measure the entire field. For I-POD, we use random initial condition, and collect the 100 snapshots from the first 3000 time steps. The snapshots are equi-spaced, and we construct the I-POD based ROM (using $Y'X$) using 30 modes.



(a) Comparison of eigenvalues extract by I-POD and actual eigenvalues

(b) Comparison of eigenvalues extract by BPOD and actual eigenvalues

(c) Comparison of output errors

(d) Comparison of state errors

Fig. 4.3. *Comparison between BPOD and I-POD for Pollutant Transport Equation*

The comparisons between the Balanced POD and I-POD is shown in Figure 4.3. In Figure 4.3(a)-(b), we compare the eigenvalues extract by I-POD and Balanced POD. To test the ROM, we use different random forcing and take the average output/state error over 20 different simulations. In Figure 4.3(c), we compare the averaged output errors of the two algorithm. In Figure 4.3(d), we compare the averaged state errors of the two algorithm.

From the two examples, we can see that at the beginning of the simulation, the output error using I-POD is larger than Balanced POD, but the errors decay fast and I-POD is more accurate then Balanced POD, as time increases. The state error is lower for the I-POD when compared to the BPOD.

4.2. Comparison with DMD. The I-POD can be construed as a sequence of DMDs to extract the eigenmodes of a system from a given trajectory targeting the different timescales. Further, the I-POD can be thought of as a sequence of DMDs that iteratively extracts the eigenmodes present in different trajectories of a system. The I-POD also utilizes adjoint simulations to obtain information regarding the left eigenvectors of a system, and hence, can construct a more accurate ROM of the system in modal space than the DMD. To see this, note that the ROM in modal space is given by:

$$\begin{aligned} \psi_k = V_l' A V_r \psi_{k-1} + V_l' B w_{k-1}, \\ y_k = C V_r \psi_k. \end{aligned} \tag{4.7}$$

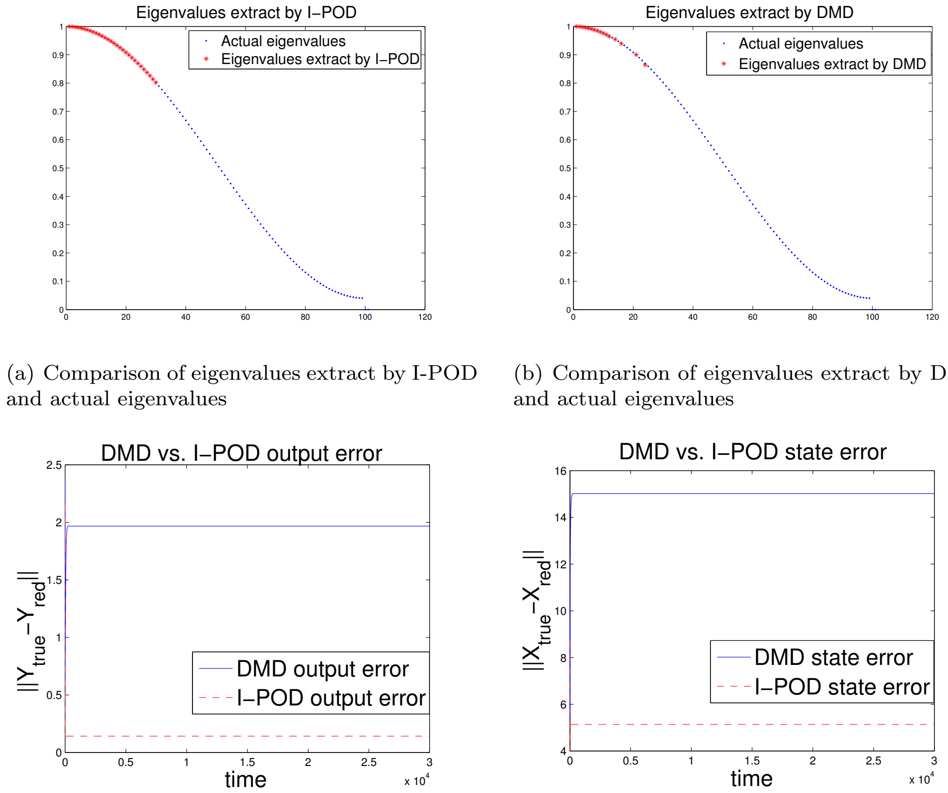
The I-POD can accurately extract the left eigenvectors V_l from the adjoint simulations while the DMD has to rely on a pseudo-inverse of V_r as an approximation of the left eigenvectors V_l. Thus, theoretically, the I-POD approximation is guaranteed to be better than the DMD because it exploits the adjoint information. The DMD also does not have the time scale targeting property of the I-POD given a single trajectory of a system. Finally, the I-POD also theoretically shows why a DMD type procedure is capable of extracting the underlying eigenmodes of the system (under Assumption A2).
Here, we compare the two algorithms for the heat equation and pollutant transport equation.

4.2.1. Heat equation. For DMD, we use a random initial condition (if use the actual initial condition, it will give a worse results), and collect the 100 snapshots from the first 3000 time steps. The snapshots are equi-spaced, and we construct DMD based ROM using 16 modes. We construct the I-POD based ROM as mentioned before. The comparisons between the DMD and I-POD are shown in Figure 4.4. In Figure 4.4(a)-(b), we compare the eigenvalues extract by I-POD and DMD. To test the ROM, we use 20 different random initial conditions and take the average output/state error of these 20 simulation. In Figure 4.4(c), we compare the averaged output errors of the two algorithm. In Figure 4.4(d), we compare the averaged state errors of the two algorithm.

4.2.2. Pollutant Transport Equation. For DMD, we use the random initial condition, and collect the 100 snapshots from the first 3000 time steps. The snapshots are equi-spaced, and we can construct the DMD based ROM using 30 modes. We construct the I-POD based ROM as mentioned before. The comparison between the DMD and I-POD is shown in Figure 4.5. In Figure 4.5(a)-(b), we compare the eigenvalues extract by I-POD and DMD. To test the ROM, we use 20 different random time forcing and take the average output/state error over 20 simulations. In Figure 4.5(c), we compare the average output errors of the two algorithm. In Figure 4.5(d), we compare the average state errors of the two algorithms.

From the two examples, we can see that I-POD is significantly better than DMD, especially for the 2D pollutant transport equation. However, this should not be a surprise since the I-POD uses different time-scales, as well as the adjoint information to get a higher fidelity ROM.

5. Application of I-POD to Filtering of Partial Differential Equations. Consider now the continuous-discrete filtering of the distributed parameter system in



(a) Comparison of eigenvalues extract by I-POD and actual eigenvalues

(b) Comparison of eigenvalues extract by DMD and actual eigenvalues

(c) Comparison of output errors

(d) Comparision of state errors

Fig. 4.4. *Comparison between DMD and I-POD for Heat equation*

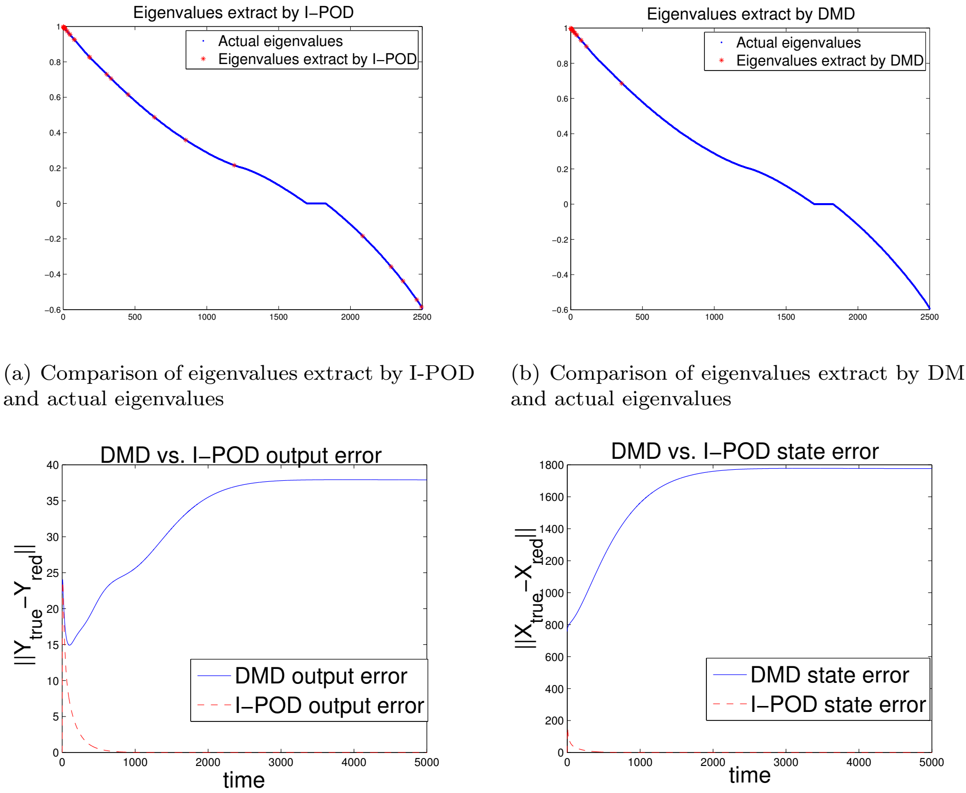
the high dimensional discretized setting:

$$x_k = Ax_{k-1} + w_k, \tag{5.1}$$

$$y_k = Cx_k + v_k, \tag{5.2}$$

where recall that $y_k \in \Re^p$ is the measurement at time t_k, w_k is white noise process perturbing the systems while v_k is a measurement white noise process corrupting the measurement at time t_k. Here, w_k and v_k are independent and the initial condition $x(0)$ is independent of w_k and v_k too. Typically, the measurements are very sparse, i.e., $p << N$, and N is very large. Hence, using standard estimation theoretic techniques such as the Kalman filter for filtering the above system is out of question owing to the high dimensionality of the resulting problem (the Kalman filter requires $O(N^3)$ operations at every update step. Thus, it is vitally important that suitable ROMs be devised to alleviate the computational intractability of the problem above. Since we are considering the discrete setting for filtering, let us assume that the measurements are taken time T apart.

In order to form a suitable ROM of the above system, suppose we keep only N_r of the eigenfunctions of A as modes of the ROM. Under assumption A1-A2, the expected value of the error between the true system and the ROM at any time is given by the following result.



(a) Comparison of eigenvalues extract by I-POD and actual eigenvalues

(b) Comparison of eigenvalues extract by DMD and actual eigenvalues

(c) Comparison of output errors

(d) Comparison of state errors

Fig. 4.5. *Comparison between DMD and I-POD for Pollutant Transport Equation*

Proposition 5.1. *The expected value of the squared error in keeping only N_r modes in the ROM is given by*

$$E||e_k||^2 = \sum_{i=N_r+1}^{N} (\lambda_i)^{2k} E|(x(0), u_i)|^2 + \sum_{i=N_r+1}^{N} \sigma_i^2 (\frac{(\lambda_i)^{2k} - 1}{(\lambda_i)^2 - 1}),$$

$$\sigma_i^2 = u_i' R_W u_i, \qquad (5.3)$$

where $e_k = x(t_k) - x_{red}(t_k)$, $x(t_k)$ is the state of the true system, $x_{red}(t_k)$ is the state of the reduced order system, $(.,.)$ denotes the inner product in R^n, u_i are the left eigenvectors, and R_w represents the covariance of the white noise process w_k.

Note that the first term in the above expression is due to the initial conditions while the second term is due to the random perturbation w_k.

Proof. The error incurred in keeping only N_r modes in the ROM is given by:

$$e_k = \sum_{i=N_r+1}^{N} (\lambda_i)^k (x(0), u_i) v_i + \sum_{i=N_r+1}^{N} \Delta_i^w (t_k) v_i,$$

where

$$\Delta_i^w(t_k) = \sum_{\tau=0}^{t_k} (\lambda_i)^{t_k-\tau} c_i^w(\tau),$$

$$c_i^w(\tau) = (w_\tau, u_i).$$

Here, u_i are the left eigenvectors while v_i are the right eigenvectors. Then, it follows that:

$$E||e_k||^2 = \sum_i (\lambda_i)^{2k} E|(x(0), u_i)|^2 + \sum_i E|\Delta_i^w(t_k)|^2, \tag{5.4}$$

where for notational ease the subscript i is used to denote the summation from $N_r + 1$ to N. Then,

$$|\Delta_i^w(t_k)|^2 = \sum_{\tau=0}^{t_k} \sum_{s=0}^{t_k} \lambda_i^{(t_k-\tau)} \lambda_i^{(t_k-s)} c_i^w(\tau) c_i^w(s). \tag{5.5}$$

Noting that

$$c_i^w(\tau) c_i^w(s) = (w_\tau, u_i)(w_s, u_i) = \sum_{j,k} w_j(\tau) w_k(s) u_{ij} u_{ik},$$

where u_{ij} denotes the j^{th} component of u_i, it follows that

$$E[c_i^w(\tau) c_i^w(s)] = \sum_{j,k} u_{ij} u_{ik} E[w_j(\tau) w_k(s)] = u_i' R_w \delta[\tau - s] u_i, \tag{5.6}$$

where $\delta[.]$ denotes the Kronecker delta function. Then, substituting the above equation back into Eq. 5.5, and using the result in Eq. 5.4, while using the sampling property of the Kronecker delta function under an integral, the result follows. □

Thus, we have:

$$E||e_k||^2 \leq \underbrace{\sum_{i=N_r+1}^{N} (\lambda_i)^{2k} E|(x(0), u_i)|^2}_{\text{initial condition error}} + \underbrace{\sum_{i=N_r+1}^{N} \sigma_i^2 (\frac{(\lambda_i)^{2k} - 1}{(\lambda_i)^2 - 1})}_{\text{random perturbation error}},$$

$$\sigma_i^2 = u_i' R_w u_i \tag{5.7}$$

The above expression gives an estimate of the error made in keeping only N_r modes in the solution. Note that the measurement equations are immaterial in these error estimates since they do not alter the system equations in any fashion. In fact, the above is an a priori estimate that is averaged over all possible future observations.

Given the measurement time interval T, and the probability density function of the initial state x_0, we can neglect those modes such that $\lambda_i^{2T} \approx 0$ and thus, the perturbation due to the initial condition is negligible. Of course, the error due to the stochastic perturbations remains, however, theoretically, we can get all λ_i since that is assured us by Proposition 3 and I-POD, and hence, we can make an a priori error estimate regarding the error made in keeping only N_r modes of the system.

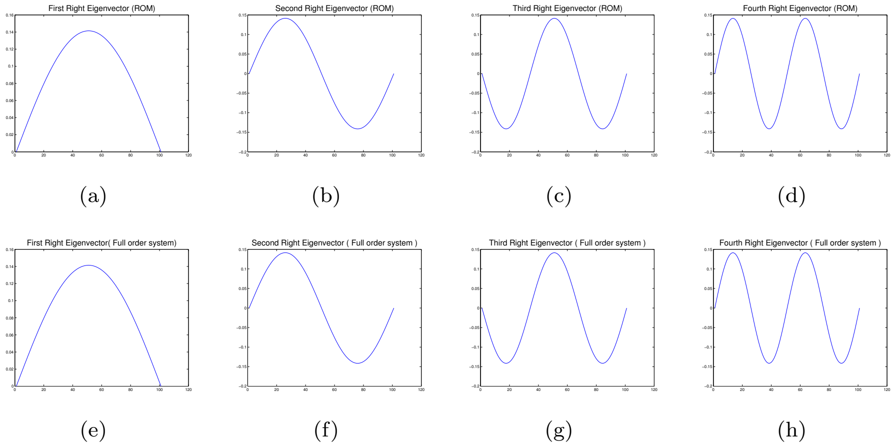
Given that we have sufficient number of modes in our ROM such that the error in the reduced solution is within some pre-specified bounds, the filtering of the PDE proceeds as follows. We choose those eigenfunctions that need to be kept, given that they have already been extracted using the I-POD procedure, and form the ROM for the filtering problem as follows. Define the transform $V_r = [v_1, \cdots v_{N_r}]$ consisting of the retained right eigenmodes and $V_l = [u_1, \cdots u_{N_r}]$, the retained left eigenmodes. The filtering ROM is the following:

$$\psi_k = (V_l' A V_r)\psi_{k-1} + V_l' B w_k,\ \psi(0) = (x(0), V_l),$$
$$y_k = (CV_r)\psi_k + v_k, \tag{5.8}$$

The above system now results in an $N_r \times N_r$ filtering problem with $N_r << N$ and thus, standard estimation theoretic methods such as the Kalman filter can be used to solve the problem. In the following sections, we show the application of the I-POD based filtering technique to the 1-D Heat equation, the 1-D wave equation and the 2-D pollutant transport equation.

5.1. Heat Equation. The heat equation is given is Section 4, here, we use the same initial condition and boundary condition as in Section 4.

First, we compare the right eigenvectors extracted by the I-POD technique with the full order system in Figure 5.1.

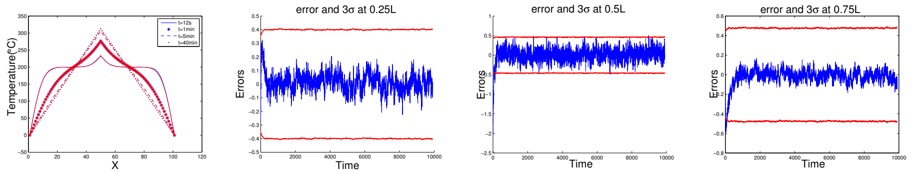


(a) (b) (c) (d)

(e) (f) (g) (h)

FIG. 5.1. *Comparison of right eigenvectors extracted by I-POD (on the top) and full order system (on the bottom)*

The comparisons between the reduced order filter and real system at four different time steps are shown in Figure 5.2(a). The red curves are the real heat profiles and the blue curves are filter results from the reduced model. In Figure 5.2(b)-(d), the error and the 3σ boundary for the reduced model, at three different chosen location are shown too.



(a) Comparison of ROM filter (in blue) with actual heat profile (in red)

(b) Error and 3σ error bounds at 0.25L

(c) Error and 3σ error bounds at 0.5L

(d) Error and 3σ error bounds at 0.75L

Fig. 5.2. *Filter results for heat equation*

It can be seen that the I-POD ROM based Kalman filter provides a good estimate of the temperature profile for the problem in that the errors stay within the 3 σ error bounds (a common engineering measure for verifying the consistency of a filter).

5.2. Wave Equation. The one dimensional wave equation describing a vibrating string on a finite interval is:

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} \tag{5.9}$$

The string vibration is simulated for a period of 1.5 seconds and the time domain T is discretized into 200 time steps. The length of the string is 1m and is divided by $M = 100$ grid points. The variable c is the wave speed, and takes value 1 here.

The boundary conditions for a fixed end string are:

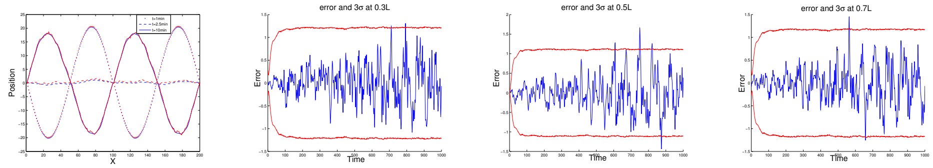
$$u_{x=0} = 0; u_{x=M} = 0. \tag{5.10}$$

Displacements are measured at five equi-spaced points along the string. A random initial condition is used for generating the reduced order model, while the initial condition of the string for simulation is:

$$\begin{gathered} u_0 = 20sin(2\pi \frac{x}{M}), \\ \frac{\partial u_0}{\partial t} = \begin{cases} 4\frac{x}{M} & \text{if } x \leq \frac{M}{2} \\ 4 - 4\frac{x}{M} & \text{if } x \geq \frac{M}{2}, \end{cases} \end{gathered} \tag{5.11}$$

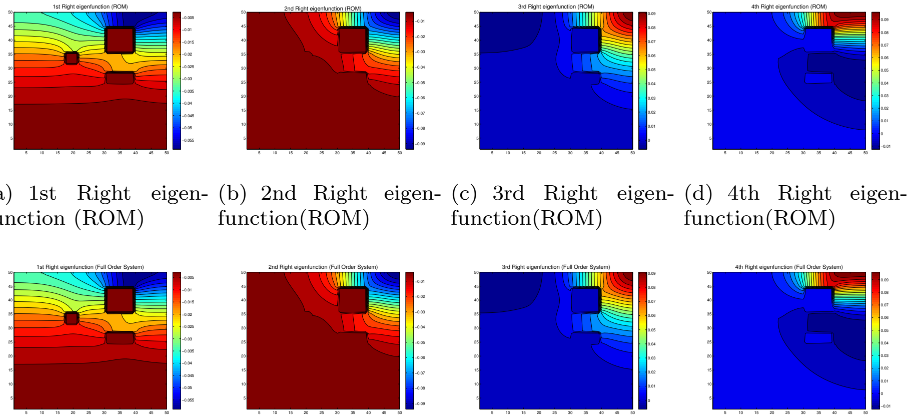
In Figure 5.3(a), we show the comparison between the real wave profiles and the reduced model filter at three different time steps. The real wave profiles are in red while the filter results are shown in blue. Also, in Figure 5.3 (b)-(d), the errors and the 3σ boundary for the reduced model at three different chosen location are shown.

Again, these results show that the ROM based filter is capable of getting a good estimate of the wave profile based on the noisy measurements.



(a) Comparison of ROM filter (in blue) with actual wave profile (in red)
(b) Error and 3σ error bounds at 0.3L
(c) Error and 3σ error bounds at 0.5L
(d) Error and 3σ error bounds at 0.7L

FIG. 5.3. *Filter results for wave equation*



(a) 1st Right eigenfunction (ROM)
(b) 2nd Right eigenfunction(ROM)
(c) 3rd Right eigenfunction(ROM)
(d) 4th Right eigenfunction(ROM)

(e) 1st Right eigenfunction
(f) 2nd Right eigenfunction
(g) 3rd Right eigenfunction
(h) 4th Right eigenfunction

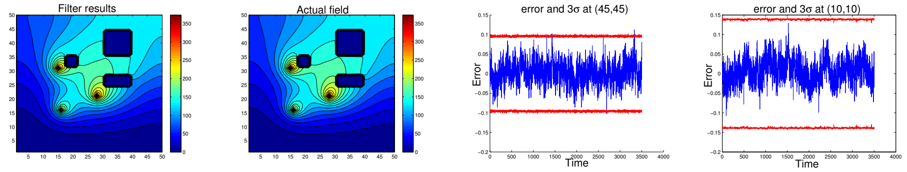
FIG. 5.4. *Comparison of Right eigenfunctions*

5.3. Pollutant transport equation. The 2D pollutant transport equation is given in Section 4, the initial condition and boundary condition are the same as in Section 4. The first four right eigenfunctions extracted by I-POD technique (on the top) and the full order system (on the bottom) are compared in Figure 5.4.

Figure 5.5 (a) and (b) compare the filter results with the actual field at the end of the simulation. Also, in Figure 5.5 (c)-(d), the errors and the 3σ boundary for the reduced model at two different chosen location are shown.

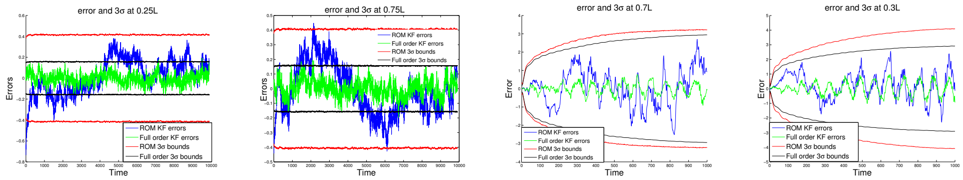
These results show that for the 2-dimensional pollutant transport case, the ROM based filter is capable of getting a good estimate of the actual fluid field based on the noisy measurements.

5.4. Comparison with Full Order Kalman filter. Now, we compare the accuracy and the computational cost of the I-POD ROM based Kalman filter with the full order Kalman Filter. For the heat equation, we need 15.62s using ROM filtering, and 18.45s using the full order Kalman filter. The error and 3σ boundary



(a) ROM Filter result at the end of simulation (b) Actual field at the end of simulation (c) Error and 3σ error bounds at (45,45) (d) Error and 3σ error bounds at (10,10)

Fig. 5.5. *Filter results for Pollutant transport equation*



(a) Comparison of KF for heat equation (b) Comparison of KF for heat equation (c) Comparison of KF for wave equation (d) Comparison of KF for wave equation

Fig. 5.6. *Comparison for ROM Kalman filter and Full order Kalman filter*

comparing ROM and full order Kalman fIlter at two different locations are shown in Figure 5.6 (a) and (b). For wave equation , we need 2s using ROM filtering, and 5.4s using the full order Kalman filter. The error and 3σ boundary comparing ROM and full order filter at two different locations are shown in Figure 5.6 (c) and (d). For pollutant transport equation, we need 765.5s using ROM filtering, and we cannot implement the full order Kalman filter in real time due to the high dimensionality of the problem. As can be seen from these examples, the error and the confidence bounds of the estimates are better in the case of the full order Kalman filter, however, at a much higher computational overhead. It should also be noted that though the errors and confidence bounds of the ROM filter are higher, the ROM filter is nonetheless consistent, i.e., the ROM filter errors remain within the ROM confidence bounds.

6. Conclusion. In this paper, we have introduced a data-based iterative snapshot POD (I-POD) approach to form ROMs for large scale linear systems. We have used the I-POD based ROM to form a reduced order Kalman filter for application to the filtering of linear partial differential equations. We have compared the performance of the ROMs formed by the I-POD to those formed using BPOD and DMD, and shown that the I-POD compares favorably with these techniques. We have shown the application of the I-POD ROM based filtering technique to the heat, wave and pollutant transport equations. As can be seen from the results, the linear ROM based filtering performed well when compared to the full order Kalman filter, while taking significantly less time. The most pressing need at this point is to be able to extend the I-POD technique to nonlinear systems. Further, we will also apply the technique to more realistic 2 and 3-dimensional partial differential equations that may be encountered in practice such as fluid flow problems.

REFERENCES

[1] M. Loeve, *Probability Theory.* New York: Van Nostrand, 1955.

[2] G. Berkooz *et al.*, "The proper orthogonal decomposition in the analysis of turbulent flows," *Ann. Rev.. Fl. Mech.*, vol. 25, pp. 539–575, 1993.

[3] K. C. Hall *et al.*, "Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows," *AIAA Journal*, vol. 38, pp. 1853–1862, 2000.

[4] L. Sirovich, "Turbulence and dynamics of coherent structures. part 1: Coherent structures," *Quarterly of Applied Mathematics*, vol. 45, pp. 561–571, 1987.

[5] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability and model reduction," *IEEE Transactions on Automatic Control*, vol. 26, pp. 17–32, 1981.

[6] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA Journal*, vol. 40, pp. 2323–2330, 2002.

[7] C. W. Rowley, "Model reduction of fluids using balanced proper orthogonal decomposition," *International Journal fo Bifurcation and Chaos*, vol. 15, pp. 997–1013, 2005.

[8] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[9] C. W. Rowley *et al.*, "Spectral analysis of nonlinear analysis," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.

[10] J.-N. Juang, *Applied System Identification.* Englewood Cliffs, NJ: Prentice Hall, 1994.

[11] S. Ahuja and C. W. Rowley, "Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators," *Journal of Fluid Mechanics*, vol. 645, pp. 447–478, 2010.

[12] L. Bertino, G. Evensen, and H. Wackernagel, "Combining geostatistics and kalman filtering for data assimilation in an estuarine system," *Inverse Problems*, vol. 18, pp. 1–23, 2002.

[13] P. L. Houtekamer and H. L. Mitchell, "A sequential ensemble kalman filter filter for atmospheric data assimilation," *Mon. weath. Rev.*, vol. 129, pp. 123–137, 1999.

[14] G. Evensen, "Inverse methods and data assimilation in nonlinear ocean models," *Physica D*, vol. 77, pp. 108–129, 1994.

[15] ——, "Advanced data assimilation for strongly nonlinear dynamics," *Mon. weath. Rev.*, vol. 125, pp. 1342–1354, 1997.

[16] G. B. et. al., "Analysis scheme in the ensemble kalman filter," *Mon. Weath. Rev.*, vol. 126, pp. 1719–1724, 1998.

[17] G. Evensen and P. J. V. Leeuwen, "An ensemble kalman smoother for nonlinear dynamics," *Mon. Weather Rev.*, vol. 128, pp. 1852–1867, 2000.

[18] G. Evensen, "The ensemble kalman filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[19] J.Mandel *et al.*, "A dynamic data driven wildland fire model," in *Springer Lecture Notes on Computer science (LNCS), 4487*, 2007.

[20] D. B. Work *et al.*, "An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices," in *Proc. IEEE Int. Conf. Dec. Cont.*, 2008.

[21] A.Bensoussan *et al.*, *Representation and Control of Infinite Dimensional Systems, Vols. I and II.* Birkhauser, 1992.

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

[23] ——, "Observability of hyperbolic systems with interior moving sensors," *Lecture notes in Control and Info. Sci.*, vol. 185, pp. 489–499, 1993.

[24] ——, "l^∞ exact observability of the heat equation with scanning pointwise sensor," *SIAM J. Control and Optim.*, pp. 1037–1051, 1994.

[25] 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.

[26] J. A. Burns and B. King, "Optimal sensor location for robust control of distributed parameter systems," in *Proc. IEEE Conf. Dec. Control*, 1993, pp. 3697–3972.

[27] A. E. Jai and A. J. Pritchard, *Sensors and Actuators in Analysis of Distributed Parameter Systems.* New York, NY: John Wiley, 1988.

[28] Y. Sakawa, "Observability and related problems for partial differential equations of parabolic type," *SIAM J. Control*, vol. 13, pp. 14–27, 1975.

[29] M. Krstic and A. Smyshlaev, *Boundary Controls of PDEs.* Philadelphia, PA: Advances in Design and Control: SIAM, 2008.

[30] R. Vasquez and M. Krstic, *Control of Turbulent and Magnetohydrodynamic Flows.* Boston, MA: Systems and Control: Foundations and Applications, Birkhauser, 2007.

[31] H. T. Banks and K. Kunisch, *Estimation Techniques for Distributed Parameter Systems.* Boston, MA: Systems and Control: Foundations and applications, Birkhauser, 1989.

[32] R. F. Curtain and H. J. Zwart, *An Introduction to Infinite - Dimensional Linear Systems*

Theory. NewYork: Springer, 1995.

[33] B. Anderson and J. B. Moore, *Optimal Filtering.* Mineola,NY: Dover Publications; 1 edition, 2005.

Information Space Receding Horizon Control

Z.Sunberg, S. Chakravorty, R. Scott Erwin

Abstract—**In this paper, we present a receding horizon solution to the optimal sensor scheduling problem. The optimal sensor scheduling problem can be posed as a 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, combined with a receding horizon approach, obviates the need to solve the computationally intractable I-space DP problem. The technique is tested on a sensor scheduling problems in which a sensor must choose among the measurements of N dynamical systems in a manner that maximizes information regarding the aggregate system over an infinite horizon. While simple, such problems nonetheless lead to very high dimensional dynamic programming problems to which the receding horizon approach is well suited.**

I. Introduction

In this paper, we consider the problem of optimal sensor scheduling such that the information gained by the sensor is maximized. In particular, this problem is motivated by the so-called Space Situational Awareness (SSA) problem where a large number of space-based targets, such as space debris, satellites and asteroids, have to be monitored using far fewer sensors, while maximizing the information we have about these objects [1]. 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. We propose a receding horizon control (I-space RHC: IS-RHC) approach to solve such I-space DP problems. The online stochastic optimization problems that result from the receding horizon approach are solved using a simulation based gradient ascent technique. The IS-RHC technique is tested on a simple scheduling problem where the sensor has a choice between measurements of N dynamical systems.

In recent years, the optimal sensor scheduling problem has garnered much interest in the Control and Robotics community and is variously known as Information-theoretic Control/ Active Sensing and Dual Control [2], [3]. Discrete dynamic scenarios such as target tracking [3] 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 online using Model Predictive Control [4]. In the nonlinear case, the problem is stochastic and thus, is significantly harder to solve because of the associated computationally intractable stochastic DP problem. In this paper, we suggest a receding horizon control approach to the solution of such stochastic sequential decision making problems, in particular, I-space sequential decision making problems, that bypasses the need to solve the stochastic DP problem.

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 Information state (I-state), which is usually the conditional filtered pdf of the state of the system [5]–[7]. 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 [7] 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 [8]–[10]. 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 and control space problems.

Model Predictive or Receding Horizon Control (MPC/ RHC) has been one of the most successful applications of control theoretic techniques in the industry [4]. 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. We propose a similar approach to solve I-space sequential decision making problems, wherein a sequence of open loop stochastic optimization problems are solved online in a receding horizon fashion. However, in the stochastic case, the answers of the RHC and the DP techniques do not coincide because in the DP formulation, the optimization is over feedback policies and not open loop control sequences. However, such DP problems, in particular, I-space problems, are virtually computationally intractable in continuous state spaces and thus, the IS-RHC technique provides a computationally attractive solution to the I-space problems. We note here that Monte Carlo based methods for solving MDPs [11], [12] are

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

Suman Chakravorty is an Assistant 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

another computationally attractive alternative to solving such high dimensional problems, however, a detailed comparison of this technique with the MC techniques is beyond the scope of this paper, and is left to future work. At the same time, the empirical results reported in this paper show that the IS-RHC technique does lead to better payoffs in terms of information gains when compared to a shortsighted policy.

The original contributions of this paper are as follows: 1) we propose an online receding horizon approach to the solution of the POMDP problem that results from the sensor scheduling problem, and 2) we propose a simulation based gradient ascent approach to the solution of the stochastic optimization problems resulting from the receding horizon approach at every time step. Our technique is valid for any nonlinear autonomous system with a discrete control space.

The rest of the paper is organized as follows. In Section II, we formulate the sensor scheduling problem. In Section III, we present the IS-RHC technique. In Section IV, we present the convergence analysis of the technique and in Section V, we present a simple numerical example as application of the IS-RHC technique.

II. Motivation and Problem Formulation

In this section, we introduce the sensor scheduling problem that we wish to solve in this paper. Consider a dynamical system with state denoted by X where $X = [X^{(1)}, X^{(2)}, ...X^{(N)}]^T$ and $X^{(i)}$ is a vector that represents the state of a dynamical subsystem whose dynamics may (or may not) be coupled with the dynamics of the other dynamical subsystems. In the case of the SSA problem, the state of satellite i is denoted by $X^{(i)}$. Let the dynamics of the entire system (the entire collection of subsystems) be represented by the following nonlinear difference equation:

$$X_k = F(X_{k-1}) + G(X_{k-1})W_{k-1}, \tag{1}$$

where $F(.)$ and $G(.)$ are nonlinear functions, and $\{W_k\}$ is an uncorrelated white noise sequence. If the sub dynamical systems were decoupled the above equation would decompose into N independent difference equations, one for each of the sub-states $X^{(i)}$. The measurement equation for the state of the system is denoted by the following (possibly) nonlinear equation:

$$z_k = H_{u_k}(X_k) + V_k, \tag{2}$$

where $\{V_k\}$ is a zero mean uncorrelated white noise sequence, and $H_{u_k}(.)$ is a nonlinear measurement function which is determined by u_k. u_k is a control variable that can take values from 1 to N; $u_k = i$ denotes that we make a measurement of sub-state $X^{(i)}$ at time k. Although this description implies that we can only measure one sub-component of X at any time step, we might have the choice of making $P > 1$ measurements as well. For notational simplicity we shall concentrate only on $P = 1$ in the following discussion. The generalization to $P > 1$ is quite straightforward.

Let χ_k denote the pdf of the state X at time k. We shall call χ_k the information state of the system since it encodes our knowledge (or lack thereof) about the system state X. Given the information state (I-state) at time $k-1$, χ_{k-1}, the I-state at time k, χ_k, will depend on the particular component that is chosen for measurement at time k and hence on the control variable u_k. It is also clear that χ_k is dependent on the noisy observation at time k, z_k. However, z_k is a random variable, and thus χ_k is also random given χ_{k-1} and the control u_k. Given χ_{k-1}, the control u_{k-1} and the current observation z_k, the current I-state χ_k is obtained via the Bayes filtering equation (Ch. 5, pp. 252, [5]). In fact, the evolution of the I-state is governed by a Markov chain (MC) whose transition density function is denoted by $p(\chi'/\chi, u)$ [5], [6].

Optimal Sensor Scheduling Problem: Our objective in this work is to maximize the total information about the dynamical system over the infinite horizon. To this end, let us define the information gain metric $\Delta I(\chi, u)$ denoting the expected information gain in choosing control u at I-state χ. An excellent discussion of metrics for sensor tasking problems can be found in [13]. Let $0 < \beta < 1$ denote a discount factor that quantifies the fact that the information gains in the immediate future are more important to us than the information gains further out in the future. We wish to solve the following discounted sequential decision making problem over all possible feedback policies $u(.)$:

$$V^*(\chi) = \max_{u(.)} V(\chi, u(.)), \text{where} \tag{3}$$

$$V(\chi, u(.)) \equiv E[\sum_{t=1}^{\infty} \Delta I(\chi_t, u(\chi_t))\beta^t | \chi_0 = \chi], \tag{4}$$

for all possible information states χ. Since the I-state χ is governed by a controlled Markov chain, the answer to the above problem is provided by solving the following Dynamic Programming problem:

$$V^*(\chi) = \max_u [\Delta I(\chi, u) + \beta \int p(\chi'|\chi, u) V^*(\chi') d\chi']. \tag{5}$$

Clearly the above DP problem resides in a continuous and very high dimensional state space thereby making the problem computationally intractable.

III. Information Space Receding Horizon Control (IS-RHC)

In this section, we shall propose a simulation based Receding Horizon Control approach (IS-RHC) to solve the I-space MDP problem that was posed in the previous section.

A. Stochastic Relaxation of Optimization Problem

Consider again the statement of the I-space MDP given in Eq. 3. Given that the expected information gain is uniformly bounded above, i.e., $|\Delta I(\chi, u)| < M < \infty$ for all (χ, u) and given the discount factor $\beta < 1$, and given any arbitrarily small error tolerance $\epsilon > 0$, there always exists a finite time T such that the finite horizon T-step discounted information gain $J_T(\chi, u_0, u_1..., u_T)$ for any infinite horizon control sequence $\{u_t\}_{t=1}^{\infty}$ is arbitrarily close to the infinite horizon discounted cost-to-go for the same control sequence. Thus, in the following we shall concentrate on solving the discounted finite horizon I-space MDP assuming that a finite horizon T and discount factor β is given such that the above approximation holds thereby leaving us with a finite dimensional optimization problem as opposed to the infinite dimensional problem resulting from the original infinite horizon case. Define the T-step information gain in following the T-step control sequence $U = \{u_1, \cdots, u_T\}$ from I-state χ as follows:

$$J(\chi, u_1, \cdots, u_t) = E_\chi[\sum_{t=1}^{T} \Delta I(\chi_t, u_t)\beta^t | \chi_0 = \chi], \tag{6}$$

where the notation $E_\chi[.]$ denotes that the expectation is over the sample paths $\{\chi_0 = \chi, \chi_1, \cdots, \chi_T\}$ that are particular $T-$step realizations of the I-space process. We have dropped the subscript T for notational convenience. The above equation is different from Eq. 4 because the expectation above is with respect to an open loop policy while the one in Eq. 4 is with respect to a feedback policy. Further, we define the optimal T-step information gain as follows:

$$J^*(\chi) = \max_{u_1 \cdots u_T} J(\chi, u_1, \cdots u_T).$$

Define a randomized policy $\Pi = \{\pi_1, \cdots \pi_T\}$ where π_t is a probability vector such that $\pi_{t,j} = \Pr(u_t = j)$ where $\pi_{t,j}$ denotes the j^{th} component of π_t. Thus, in the randomized policy, we do not take a particular control action at time t, instead we take the control action $u_t = j$, $j = 1 \cdots N$, with a probability $\pi_{t,j}$ and $\sum_j \pi_{t,j} = 1$ for all t. Further, define the T-step information gain in following stochastic policy Π from I-state χ as follows:

$$J_s(\chi, \Pi) = E_{\chi,u}[\sum_{t=1}^{T} \Delta I(\chi_t, u_t)\beta^t | \chi_0 = \chi],$$

where the notation $E_{\chi,u}[.]$ denotes the fact that the expectation in the above equation is now with respect to both the sample paths $\{\chi_1, \cdots \chi_T\}$ and control sequences $\{u_1, \cdots u_T\}$. Then, it can be seen that the following relationship holds:

$$J_s(\chi, \Pi) = \sum_{u_1, \cdots u_T} J(\chi, u_1, \cdots u_T)\pi_{1,u_1}..\pi_{T,u_T}, \tag{7}$$

the summation above is a T-dimensional sum where each u_t can take one of N values. In the following, for notational convenience, we shall abuse notation and denote the expected information gain due to a stochastic policy $J_s(.)$ by $J(.)$ (the symbol for information gain due to a deterministic policy). We wish to solve the optimization problem:

$$J^*(\chi) = \max_{\Pi} J(\chi, \Pi), \tag{8}$$

given some I-state χ.

B. Simulation based Stochastic Gradient Method

Let us write $\pi_{t,N} = 1 - \pi_{t,1} - ... - \pi_{t,N-1}$. Then, Eq. 7 reduces to the following equation:

$$J(\chi, \Pi) = \sum_{u_1 ... u_T} [\sum_{j=1}^{N} J(\chi, u_1, ..., u_t = j, .., u_T)\pi_{t,j}] \pi_{1,u_1}...\pi_{T,u_T},$$

$$= \sum_{u_1 ... u_T} \pi_{1,u_1}...\pi_{T,u_T}[\sum_{j=1}^{N-1} J(\chi, u_1, ..., u_t = j, .., u_T)\pi_{t,j} + J(\chi, u_1, ..., u_t = N, ..., u_T)(1 - \pi_{t,1} - ... - \pi_{t,N-1})]. \tag{9}$$

Note that $J(\chi, \Pi)$ is a multi-linear function of the probabilities $\pi_{t,j}$. Then, from Eq. 9, it follows that:

$$\frac{\partial J(\chi, \Pi)}{\partial \pi_{t,j}} = \sum_{u_1..u_T} \pi_{1,u_1}..\pi_{T,u_T} \{J(\chi, u_1, .., u_t = j, .., u_T) - J(\chi, u_1, .., u_t = N, .., u_T)\}. \tag{10}$$

Consider the term $\sum_{u_1..u_T} \pi_{1,u_1}..\pi_{T,u_T} J(\chi, u_1, .., u_t = j, .., u_T)$. This is nothing but the expected T-step information gain in following stochastic policy Π whenever $u_t = j$. Define

$$J_{(t,j)}(\chi, \Pi) = \sum_{u_1..u_T} \pi_{1,u_1}..\pi_{T,u_T} J(\chi, u_1, .., u_t = j, .., u_T), \tag{11}$$

where subscript (t, j) denotes the gradient of $J(\chi, \Pi)$ with respect to $\pi_{t,j}$. Then, using the above definition and Eq. 10, it follows that

$$\frac{\partial J(\chi, \Pi)}{\partial \pi_{t,j}} = J_{(t,j)}(\chi, \Pi) - J_{(t,N)}(\chi, \Pi), \tag{12}$$

for all t and all j. Thus, by simulating sample I-space trajectories under the stochastic policy Π, we can estimate the gradient of the T-step information gain function $J(\chi, \Pi)$ with respect to each of the control probabilities $\pi_{t,j}$. Then, the policy Π can be improved by ascending along the gradient $\frac{\partial J(\chi,\Pi)}{\partial \Pi}$. Note that the gradient $\frac{\partial J(\chi,\Pi)}{\partial \Pi}$ is a $T \times N$ matrix whose (t, j) element is $\frac{\partial J(\chi,\Pi)}{\partial \pi_{t,j}}$. Mathematically, this means we adjust the stochastic policy at iteration n (not to be confused with time t) as follows:

$$\Pi_{n+1} = \mathcal{P}_P\{\Pi_n + \epsilon_n \frac{\partial J(\chi, \Pi)}{\partial \Pi}|_{\Pi = \Pi_n}\}, \tag{13}$$

where ϵ_n is a small step size and $\mathcal{P}_P(.)$ denotes a projection onto the space of stochastic policies P. The projection is necessary since the new policy update need not satisfy the constraints required to be satisfied by a stochastic policy. This projection results in a quadratic programming problem whenever the constraints are violated. However, estimating $\frac{\partial J(\chi,\Pi)}{\partial \Pi}$ exactly is intractable owing to the large number of simulations required to do the estimation. Instead, we can form a noisy estimate of $\frac{\partial J(\chi,\Pi)}{\partial \Pi}$ from a single sample path (simulation) of the I-space process as follows. Recall Eq. 12 and suppose that ω is a sample realization of the I-space process, where $\{\chi_1(\omega), u_1(\omega), \cdots \chi_T(\omega), u_T(\omega)\}$ denotes the sample I-space/ control space path, with associated information gain $J(\omega)$. Then, the information gradient equation 12 can be approximated by using the noisy information gradient estimate:

$$\begin{aligned}\frac{\partial \widehat{J(\chi,\Pi)}}{\partial \pi_{t,j}} &= \frac{J(\omega)}{\pi_{t,j}} \text{ if } u_t(\omega)=j,\\ &= \frac{-J(\omega)}{\pi_{t,N}} \text{ if } u_t(\omega)=N,\\ &= 0, \text{ o.w..}\end{aligned} \tag{14}$$

Thus, using the noisy information gradient from the Eq. 14, the policy update equation may be written as follows:

$$\Pi_{n+1} = \mathcal{P}_P\{\Pi_n + \epsilon_n \frac{\partial \widehat{J(\chi,\Pi)}}{\partial \Pi}|_{\Pi=\Pi_n}\}, \tag{15}$$

where $\frac{\partial \widehat{J(\chi,\Pi)}}{\partial \Pi}$ is a $T \times N$ matrix whose (t,j) element is $\frac{\partial \widehat{J(\chi,\Pi)}}{\partial \pi_{t,j}}$. Using the noisy policy update Eq. 15 above, we improve the policy by ascending the gradient and in the limit, we would hope to reach an optimum point for the information gain function $J(\chi,\Pi)$. A convergence analysis for the above procedure is provided in Section IV. A remark regarding the noisy information gradient equation 14 is in order here.

Remark 1. *Note the division by $\pi_{t,j}$ or $\pi_{t,N}$ in Eq. 14. If $\pi_{t,j}=0$, then the aforementioned division results in the indeterminate form $0/0$. Physically, if $\pi_{t,j}=0$ for some t,j, then the control $u_t=j$ will never be taken under policy* Π. *Thus, we can never form an estimate of $J_{t,j}(\chi,\Pi)$, the information gain whenever $u_t=j$, and therefore can never form an estimate of the gradient of the information gain with respect to $\pi_{t,j}$. Hence, Eq. 14 is only used when $\pi_{t,j}\neq 0$. In the case when $\pi_{t,j}=0$, we can circumvent the problem as follows. Before any I-space sample path simulation, we choose one of the time steps t for which $\pi_{t,j}=0$ for some j with some positive probability δ, i.e., we follow π_t in choosing u_t with probability $(1-\delta)$ else we do the following: we randomly choose one of the j such that $\pi_{t,j}=0$ and apply this control action at time t. For all other τ we follow π_τ in choosing control actions u_τ. In essence, using the above procedure, we sporadically explore actions that would otherwise never be taken under* Π *with a small positive probability while keeping the rest of the policy the same. This allows us to form an unbiased estimate of the information gradient with respect to element (t,j) for which $\pi_{t,j}=0$. This exploratory procedure is analogous to the exploration steps that are used during the policy evaluation step in Actor-Critic algorithms for Reinforcement learning, as well as the exploratory off-policy steps that are used in Q-learning [14], [15] .*

C. IS-RHC Algorithm

Suppose at time $t=0$ the I-state of the system is χ_0. Also suppose that we are given some initial guess for the optimal T-step stochastic policy, say Π_0. Then using the simulation based noisy gradient estimate from Eq. 14 and the policy improvement step from Eq. 15, we can ascend the gradient of the function $J(\chi,\Pi)$ and find an optimum w.r.t Π. This gives us a T-step policy $\Pi_0^* = \{\pi_1^*...\pi_T^*\}$. As in the standard Receding Horizon control approach, we apply the control u_1 according to π_1^*. Next we observe the noisy measurement at time 1, z_1, and update our I-state according to the Bayes filteringing equation to get the I-state at time 1, χ_1. Assuming that the underlying system is autonomous (note that Eq. 1 is time independent and hence, autonomous), then we can reset time to 0, make χ_1 our new initial I-state χ_0 and repeat the procedure outlined above. In this fashion, at every time step, given the current I-state, the T-step stochastic optimization can be done online and applied in a receding horizon fashion. Mathematically, the RHC-based feedback control for I-state χ can be written as:

$$u_{RHC}(\chi) = e_1^{\mathrm{t}} \arg\max_{\Pi} J(\chi,\Pi), \tag{16}$$

where e_1 is the first unit vector in R^T (e_1 isolates the control at the first time instant of the T-step open loop control policy). The above recursive procedure is summarized in the pseudo-code IS-RHC.

Algorithm 1 Algorithm IS-RHC

```
• Given initial information state $\chi_0$, lookahead horizon $T$,
  initial policy $\Pi_1$ and error tolerance $\delta$
    1) $n=1$, define $||\Pi_1-\Pi_0|| = \delta+1$
    2) WHILE $||\Pi_n-\Pi_{n-1}|| > \delta$
       DO
        a) Generate sample I-space path $\{\chi_t(\omega)\}_{t=1}^T$ starting with initial I-state $\chi_0$.
        b) Use Eq. 14 to form the noisy estimate of the information gradient using the sample path.
        c) Use Eq. 15 to update the policy.
    3) Output converged T-step policy $\Pi^* = [\pi_1^*..\pi_T^*]$ and choose control $u_1$ according to $\pi_1^*$.
    4) Observe noisy measurement $z_1$ and update using the Bayes filter equation to obtain the new I-state $\chi_1$.
    5) Set $\chi_0=\chi_1$ and go to Step 1.
• End
```

Remark 2. *The feedback policy that results from the IS-RHC (ref. Eq. 16) is different from that which would result from solving the DP equation (ref. Eq. 4). In the DP case the expectation is with respect to the sample paths that are generated as a result of a feedback policy $u(.)$ while in the case of IS-RHC the expectation is with respect to the sample paths generated by an open loop (not feedback) sequence of control actions $\{u_1,\cdots u_T\}$.*

D. Convergence Analysis

In the following we drop all reference to the initial I-state χ in the optimization problem $J(\chi,\Pi)$ and refer to the function as only $J(\Pi)$. The gradient of the function $J(.)$ with respect to Π is denoted by $\mathcal{G}(\Pi)$. Let $\{q_i(\Pi)\leq 0\}$ denote the constraints on the problem for some $i=1,...K$. The constraints are all linear and are of the form $0\leq \pi_{t,j}\leq 1$ for all $t\in\{1...T\}$, $j\in\{1...N\}$, and $\sum_j \pi_{t,j}\leq 1$ for all $t\in\{1..T\}$. However, note that the information gain function 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

$$S = \{\Pi : \mathcal{G}(\Pi) - \sum_i \lambda_i q_i(\Pi) = 0,\ \lambda_i \geq 0\}, \tag{17}$$

where $\lambda_i = 0$ whenever $q_i(\Pi) < 0$ 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 $J(\Pi)$. The set is non-empty since $J(\Pi)$ 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 $J(\Pi) = \text{constant} = C_i$ over each S_i (see [16], p. 126), since $J(.)$ and $q_i(.)$ are twice continuously differentiable. Let the step size parameters satisfy the following conditions which is a standard assumption for most stochastic approximation algorithms [16]:

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

Then the following result holds:

Proposition 1. *The sequence of policy updates $\Pi_n \to S_i$ for some unique i.*

Proof: Write the constrained noisy policy update equation 15 as

$$\Pi_{n+1} = \Pi_n + \epsilon_n(\hat{\mathcal{G}}(\Pi_n) + z_n), \tag{18}$$

$z_n \in -C(\Pi_n)$, where $C(\Pi)$ denotes the infinite convex cone generated by the outward normals of the active constraints at point Π. A convex cone generated by vectors v_1 and v_2 is the set $\{v : v = \alpha_1 v_1 + \alpha_2 v\}$ where $\alpha_{1,2} \geq 0$. Note that $z_n = 0$ if Π_n is in the interior of the constraint set P. Further, the above expression may be rewritten as follows:

$$\begin{aligned} Y_n = \hat{\mathcal{G}}(\Pi_n) = \mathcal{G}(\Pi_n) + \underbrace{\hat{\mathcal{G}}(\Pi_n) - E[\hat{\mathcal{G}}(\Pi_n)|\Pi_n]}_{\psi_n} \\ + \underbrace{E[\hat{\mathcal{G}}(\Pi_n)|\Pi_n] - \mathcal{G}(\Pi_n)}_{\gamma_n}. \end{aligned} \tag{19}$$

It can easily be seen that $\hat{\mathcal{G}}(.)$ is an unbiased estimator of $\mathcal{G}(.)$ and hence, it follows that $\gamma_n = 0$ above. Further, due to the same reason, $E[\psi_n|\Pi_n] = 0$ and thus, $\{\psi_n\}$ is a Martingale difference sequence. These, along with the step size condition above, imply that all the assumptions of Theorem 2.1 (p. 127) of [16] are satisfied and hence, $\Pi_n \to S_i$ for some unique i. Q.E.D.

IV. Illustrative Examples

In this section, we shall present an application of the IS-RHC technique to an illustrative example containing N decoupled 2-dimensional oscillators. This abstract class of problems involving an array of simple oscillators was chosen because it is relatively simple, yet resembles the motivating satellite tracking problem in that the dynamical subsystems exhibit periodic behavior. The decoupled nature of the oscillators only affects the filtering algorithm used to generate the sample I-space paths, otherwise, the method is independent of such coupling.

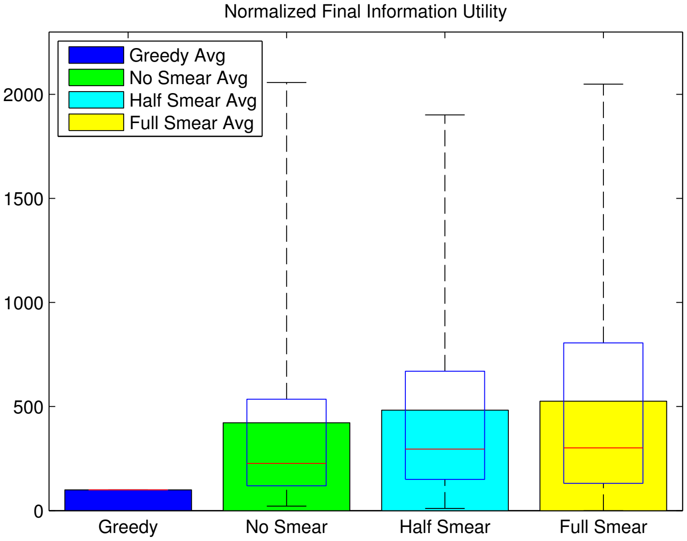


Fig. 1. Comparison of the RHC and shortsighted policies for a sample initial I-state for the case of $N = 4$ Van Der Pol oscillators. The bars represent the mean final information utility; the box plots show the median (red line), first (upper blue line) and third quartiles (lower blue line), and range of the results. There are 3 versions of the RHC policy. In the "no smear" approach, we initialize the solution of any N-step horizon with the solution of the previous N step horizon, over the first N-1 steps of the current horizon while smearing the last step, i.e, setting all probabilities at that step equal. In the half smear approach, the solution is initialized by randomizing a part of solution over the previous time horizon, while in the full smear approach, the solution of any time horizon is fully randomized, i.e., all actions at every time are set to be equally likely thereby erasing all knowledge from the optimization over the previous time horizon.

The information gain metric that is used is the following:

$$\Delta I(\chi, u) = E[det(P_{\chi,u}^{-1}) - det(P_\chi^{-1})],$$

where $det(A)$ represents the determinant of the matrix A, P_χ is the covariance of the I-state χ and $P_{\chi,u}$ is the covariance of the I-state resulting from taking control u at I-state χ. The experimental results shown below are for a set of 4 2-dimensional Van der Pol oscillators. We assume that an EKF can be used to keep track of the I-states and hence, the I-state can be approximated as a Gaussian distribution. The state space of the I-space MDP problem in this case is then 4 x (2 + 3) = 20 dimensional (2 dimensional mean and 3 dimensional covariance times the number of oscillators). We test our technique against a short sighted policy that only looks one step ahead while our RHC technique looks five steps ahead. The results of our experiments are summarized in Figures 1 and 2. In Figure 1, we see that the average gain of the I-RHC technique over that of a shortsighted policy is approximately 500-600%. The information gain is calculated over a horizon of 20 times steps for 300 different initial conditions. In Figure 2, we show a particular tasking example wherein the RHC outperforms the shortsighted policy by a very large margin (approximately 25 times better). This simple example provides empirical evidence that the IS-RHC technique does indeed result in control policies that maximize the information gained about the system over the long run

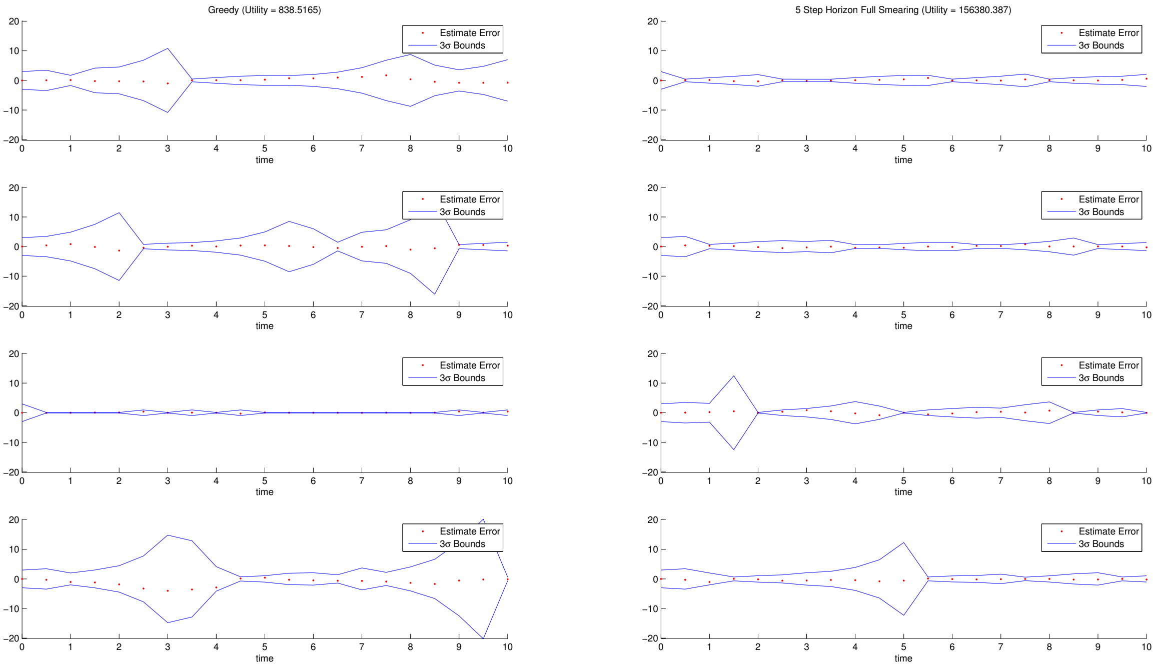


Fig. 2. Comparison of the RHC and shortsighted policies for a sample initial I-state for the case of $N = 4$ oscillators. The shortsighted oscillator repeatedly measures the third oscillator which has high process noise and a low sensor noise, thereby giving the highest short term information gain. The RHC techniques, on the other hand, let the uncertainty of the third oscillator grow, while measuring other oscillators, knowing that the third oscillator can be measured later and its uncertainty reduced. This long sighted strategy results in a much higher information gain for the RHC techniques when compared to the shortsighted strategy.

when compared to shortsighted policies. Further, to the best of our knowledge, even for the simple examples considered in this paper, the dynamic programming problems are very high dimensional and the I-space RHC technique seems to be the only one, in addition to Monte Carlo based methods [11], [12], that is capable of handling such high dimensional problems. However, the comparison with these MC based methods is beyond the scope of this paper and will be done in a follow up paper.

V. Conclusion

In this paper, we have proposed a receding horizon control based approach to solve I-space MDP, termed IS-RHC, instead of solving the associated computationally intractable DP equation. We proposed a simulation based stochastic gradient technique for solving the open loop stochastic optimization problem that results at every time step due to the IS-RHC technique. We have tested the IS-RHC on a simple example, which nevertheless result in large DP problems that are well beyond the capability of existing techniques, and the results show that the IS-RHC technique does result in significant improvement in the information gained regarding the system when compared to a shortsighted policy. Further research will focus on testing the IS-RHC on more realistic examples as well as extending the formulation such that constraints on the information process and the problem of decentralized control for multiple sensors can be taken into account.

References

[1] T. S. Kelso *et al.*, "Improved conjunction analysis via collaborative space situational awareness," in *Proc. AAS/ AIAA Spaceflight Mechanics Meeting*, 2008.

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

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

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

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

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

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

[8] 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.

[9] M. Spaan and N. Vlassis, "Perseus: Randomized point-based vallue iteration for pomdps," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[10] J. Pineau *et al.*, "Anytime point based approximations for large pomdps," *Journal of Artificial Intelligence Research*, vol. 27, pp. 335–380, 2006.

[11] M. Kearns *et al.*, "A sparse sampling algorithm for near-optimal planning in large markov decision processes," in *Proceedings of the IJCAI*, 1999.

[12] L. Kocsis and C. Szepesvari, "Bandit based monte carlo planning," in *In: ECML-06. Number 4212 in LNCS*, 2006.

[13] A. O. Hero III, D. A. Castanon, D. Cochran, and K. Kastella, Eds., *Foundations and Applications of Sensor Management*. Springer US, 2009.

[14] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Boston, MA: Athena Scientific, 1996.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[16] H. J. Kushner and G. G. Yin, *Stochastic Aproximations and Recursive Algorithms and Applications, 2nd Edition*. New York, NY: Springer-Verlag, 2003.

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. 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 [2]–[12]. Discrete dynamic scenarios such as target tracking [6]–[9], and linear spatially distributed systems [13], [14] 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 [15], [16]. 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 probability density function (pdf) of the state of the system [17]–[19]. 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 [19], [20] utilizing the

Z. Sunberg is a Graduate Student Researcher in Aerospace Engineering, Texas A&M University, College Station, TX, sunbergzach@gmail.com

Suman Chakravorty is an Associate Professor 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

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 [21]–[23]. 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. 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 with Value/ Policy Iteration as well as reinforcement learning based techniques [24]–[28]. However, the class of problems that we consider in this paper not only do not conform to the structure required by these techniques, but also have continuous state and observation spaces to which these methods do not scale. 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) has been applied very successfully in industry [29], [30]. In the deterministic setting, the MPC technique and the Dynamic Programming technique essentially give the same answer. 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. This has led to many successful applications [29], [30]. Recently, there has been increasing interest in stochastic receding horizon control (SRHC) approaches [31]–[33] 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. In our case, an analytical model of the process does not exist, instead we have access to simulations. We propose an SRHC approach to solve multi-agent I-space sequential decision making problems. A sequence of open loop stochastic optimization problems are solved online using a distributed simulation-based optimization technique in a receding-horizon fashion. 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. 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 is nonetheless intractable for other existing techniques in the literature, as a 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 a broader class 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 any given point in time denoted by the set $T^j(t)$, where

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

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, $S : T\text{x}\mathcal{H} \to \mathcal{S}$, 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. 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.

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 pdf / 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.

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 control policies over a time horizon $\mathcal{H}$ for target i, $\{u_t^{S(i,t)}(.)\}_{t=0}^{H}$, is given by:

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

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.

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. 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 reward 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 reward functions using $J(.)$ and $V(.)$ respectively). In particular, we would like to solve the open loop stochastic optimization problem:

$$\max_{\{u_t^j\}} \sum_{i=1}^{N} J(\chi_i, \{u_t^{S(i,t)}\}), \tag{5}$$

where the optimization is over all possible control choices of every sensor-time 2-tuple (j,t). 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:

$$\pi_{t,k}^j = Prob.(u_t^j = k), \tag{6}$$

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:

$$\begin{aligned} J(\chi_i, \bar{\Pi}) = J(\chi_i, \{\Pi_t^j\}) = &\sum_{u_1^{S(i,1)}..u_H^{S(i,H)}} \\ J(\chi_i, u_1^{S(i,1)}, \cdots, u_H^{S(i,h)})&\pi_{1,u_1^{S(i,1)}}^{S(i,1)} ... \pi_{H,u_H^{S(i,H)}}^{S(i,H)}. \end{aligned} \tag{7}$$

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:

$$J(\bar{\chi}, \bar{\Pi}) = \sum_{i=1}^{N} J(\chi_i, \bar{\Pi}). \tag{8}$$

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:

$$\frac{\partial J}{\partial \pi_{t,k}^j} = \sum_{l \in T^j(t)} \sum_{u_1^{S(l,1)} ... u_H^{S(l,H)}} J(\chi_l, u_1^{S(l,1)}, .., u_t^{S(l,t)} = j, .., u_H^{S(l,H)}) \pi^{S(l,1)}_{1,u_1^{S(i,1)}} .... \pi^{S(l,H)}_{H,u_H^{S(l,H)}}. \tag{9}$$

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.

The gradient ascent algorithm is the following:

$$\Pi_t^j = \mathcal{P}_P \left[\Pi_t^j + \gamma \frac{\partial J}{\partial \Pi_t^j} \right], \tag{10}$$

where γ is a small step size parameter, and $\mathcal{P}_P[.]$ denotes the projection of a vector onto the space of probability vectors P (needed because the policy updated policy of sensor-time pair (j, t) is not gauranteed to fall in P).

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. We replace $\frac{\partial J}{\partial \Pi_t^j}$ in Eq. 10 with $\widehat{\frac{\partial J}{\partial \Pi_t^j}}$ where

$$\begin{aligned} \widehat{\frac{\partial J}{\partial \pi_{t,k}^j}} &= J^{(j,t))}(\omega) \text{ if } u_t^j = k \\ &= 0, \text{ o.w.} \end{aligned} \tag{11}$$

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.

Remark 1. ***Distributed Implementation:*** *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, because $\frac{\partial J}{\partial \Pi_t^j}$ depends only on the I-trajectories of the targets in $T^j(t)$, the CPU need only simulate trajectories of those targets. Thus the CPU need only know a subset of the policies for the network of sensors - those for sensors that can see the targets in $T^j(t)$, that is $\Pi_\tau^{S(l,\tau)}$, $l \in T^j(t)$ and $\tau \in \mathcal{H}$. This allows for a sparse connection graph among the processors thereby facilitating a distributed implementation of the gradient ascent algorithm.*

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.

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 $i = 1, \cdots L$. 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

$$\begin{aligned} S = \{\bar{\Pi} : \mathcal{G}(\bar{\Pi}) \\ - \sum_i \lambda_i \nabla q_i(\bar{\Pi}) - \sum_j \mu_j \nabla h_i(\bar{\Pi}) = 0, \, \lambda_i \geq 0\}, \end{aligned} \tag{12}$$

where $\lambda_i = 0$ whenever $q_i(\bar{\Pi}) < 0$ and $\lambda_i \geq 0$ otherwise. 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 $\bar{\Pi}_n \to S_i$, for some unique i almost surely, i.e., the stochastic gradient algorithm (Eq. 10, 11) converges to a set of stationary (K-T) points such that the value of $J(.)$ on each such set S_i is constant.*

D. Receding Horizon Control

We have presented a simulation-based stochastic gradient technique to get a maximum 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 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. 10, 11) 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 $\Pi_t^{j*}$ 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 $\chi_1$.
    4) Set $\chi_0 = \chi_1$ and go to Step 1.
• End
```

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 it. We consider a set of N simple (nonlinear) pendulums subject to white process noise. We have access to M noisy angle sensors with disjoint FOVs.

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.For the numerical examples below, we apply our I-RHC-M technique to a situation where there are 4 targets ($N = 4$) and 3 sensors to measure them ($M = 3$). The initial states of the pendulums are chosen randomly and we assume that the noise statistics are identical for each of the pendulums an sensors. We assume that the Gaussian assumption holds in this problem and use extended Kalman filters 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. 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.

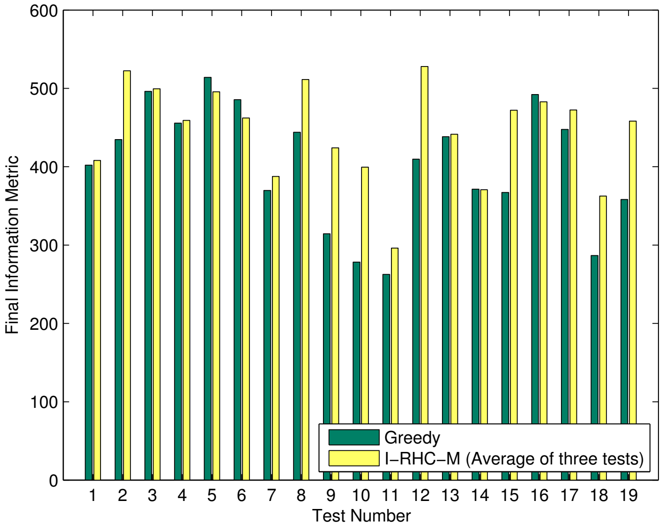


Fig. 1. Performance of the I-RHC-M algorithm in the average case scenario

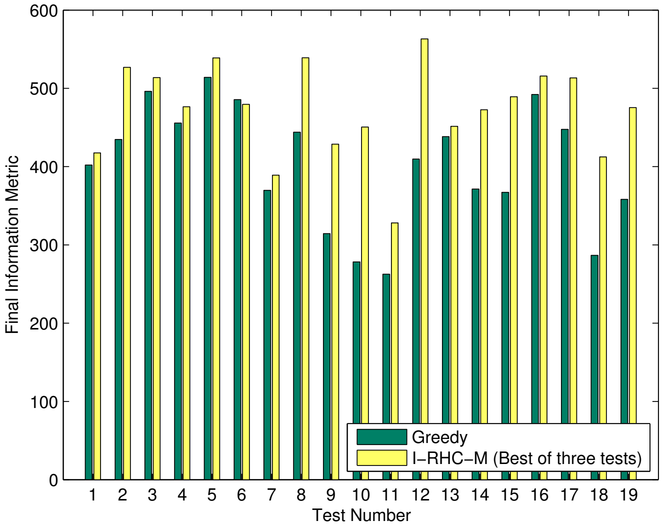


Fig. 2. Performance of the I-RHC-M algorithm in the best case scenario

The results of our numerical simulations are shown in Figs. 1 and 2. For comparison, we chose a "greedy" myopic policy as it is the only technique, other than the I-RHC-M technique, that scales to high dimensional problems such as this. The I-RHC-M technique had a lookahead horizon of 10 timesteps and the information gains were evaluated over a total time 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, compared to 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, at least theoretically. However, as can be seen from the plots, the I-RHC-M technique does beat the greedy policy most of the time. 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.

It should be noted that, if we use a reward function that is the product of the individual reward functions, the reward gains over the greedy policy are much larger (on the order of 10 times better), and this structure is more consistent with the description of the reward function as the amount of information gained about the system of targets. In order to maintain the parallelization-friendly characteristics of the reward function from Remark 1, the logarithm of this function must be used as the reward function. We have not yet obtained results using this reward function, but our conjecture is that we will be able realize very large information gains with it.

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).

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

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

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

[5] 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.

[6] 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.

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

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

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

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

[11] 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.

[12] 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.

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

[14] 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.

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

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

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

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

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

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

[21] 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.

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

[23] 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.

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

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

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

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

[28] 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.

[29] 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.

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

[31] 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.

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

[33] 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.