A Randomized Iterative POD based approach to Filtering of Distributed Parameter Systems

S. Chakravorty
Associate Professor, Aerospace Engg.
Texas A&M University
College Station, TX

Acknowledgement: Dan Yu

SIAM Conference on UQ 2012
April 2- June 5, 2012
Raleigh, NC
Outline

- Motivation
- Problem Formulation
- Related Work
- Random Iterative POD (RI-POD)
- Application to filtering of DPS
- Conclusions
Motivation

We are motivated by the problem of *optimal sequential sensor placement design for a spatially distributed process* such that the information content regarding the process is maximized.

For instance: flow control, pollutant transport, wildfires. The above goal is *central to the design of Dynamic Data Driven Application Systems (DDDAS)*

An *integral part of the problem* is to be able to *reconstruct the state of the spatially distributed process in real-time using only sparse measurements* from a distributed, possibly mobile set of sensors: this problem is basically the problem of *filtering* the so-called *distributed parameter systems (DPS)*.

![Filter Results](image)
Problem Formulation

Let the *distributed parameter system* be specified by the following *randomly perturbed linear operator equation*:

\[ \dot{X} = AX + \mathcal{W}, \]

along with the measurement equation:

\[ Y(t_k, x_j) = CX(t_k) + \mathcal{V}_k^{(j)} \]

We assume that the *operator is self-adjoint with a compact resolvent*, and thus, has a complete set of orthonormal eigenfunctions. In particular, we *assume that there exists a fine enough discretization of the operator equation* above that may be considered to be the truth and we shall *consider the discretized system in the rest of the talk*.

\[ \dot{x} = Ax + w, \]

\[ y_k = Cx_k + v_k. \]
Problem Formulation

The matrix $A$ is symmetric and has a complete set of eigenvectors due to the compact self-adjointness of the operator equation.

A naïve approach to filtering the above problem would be to use standard filtering techniques such as the Kalman filter on the discretized DPS. However, the Kalman Filter is $O(N^2)$ in the dimension of the state $N$. Thus, it is prohibitively expensive for DPS systems where $N$ is typically very large.

Hence, we shall first form an ROM of the DPS of interest and then, use Kalman filtering techniques on the ROM based system.
Related Work

There has been a lot of theoretical research in the estimation and control of distributed Parameter systems [[9]-19], Banks and Kunish 1989, Krstic and Smyshlaev 2008]: The filtering problem is solved using an operator Ricatti equation which can be very hard to solve computationally in real time

A popular method to solve the DPS filtering problem is a particle based approach called the ensemble Kalman filter (EnKF) [Evensen 1994]. The EnKF involves propagating snapshots of a DPS in real-time and then, use the measurements to perform the Kalman updates in an ensemble fashion. However, the number needed for an adequate ensemble and the snapshot propagation can be an issue for real-time operation.

We adopt an ROM based approach to the filtering problem, specifically based on the snapshot proper orthogonal decomposition (POD) approach

The POD has long been used in the fluids community to obtain reduced order approximations to fluid flow problems such as turbulence and fluid structure interaction [Lumley 1970, Sirovich 1987, Hall et. al. 1991]
Related Work

The *balanced POD* was proposed as a computationally tractable approach to constructing a *reduced order input-output description of a large scale system using the balanced model truncation technique* [Moore 1981] in alliance with the *snapshot POD* [Willcox 2002, Crowley 2005]

The *balanced POD*, and the related *ERA (Eigensystem Realization Algorithm)* from the System Identification literature, involve the *solution of a large Hankel Singular Value Decomposition (SVD) problem*. The *balanced POD* is also typically *applied to single input systems* as its based on a single trajectory of the system

There has been relatively very little research in *applying POD based ROMs to the problem of filtering DPS* with the exception of some recent work [Rowley 2010]

The *POD provides a local reduced basis representation* of the system of interest. Can it be used to *infer global behaviour of the system*? The *dynamic mode decomposition (DMD) uses the POD to extract some of the global eigenfunctions* in certain fluid flow problems [Schmid 2010]
Related Work

We propose a randomized iterative variant of the snapshot POD (RI-POD) that allows us to sequentially extract all the eigenfunctions of an operator. The technique provides guidance as to how the snapshots of the process need to be taken in order to infer the eigenfunctions. Moreover, the technique is entirely data based.

The RI-POD involves solving a sequence of small eigenvalue (or SVD) problems as opposed to solving a large Hankel Singular Value Decomposition Problem as in balanced POD or ERA, and is easily applied to multiple input multiple output (MIMO) systems.

We apply the RI-POD to develop a reduced order Kalman filter for the problem of filtering DPS.
Random Iterative POD (RI-POD)

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.

Given the snapshot ensemble $X = [x_1, x_2, \cdots, x_M]$, snapshot POD transforms $A$ by the transformation $T = XV_p\Sigma_p^{-1/2}$, where

$$(X'X)V_p = V_p\Sigma_p.$$

The transformation $T$ is an $M\times N$ matrix and transforms the original system into the reduced order system given that the number of snapshots $M << N$, the dimension of the original system:

$$\dot{\psi} = (T'AT)\psi \equiv \tilde{A}\psi$$
Random Iterative POD (RI-POD)

Consider an eigenvalue decomposition of the ROM: \( \tilde{A}P = PA \), that diagonalizes the ROM into the modal co-ordinates: \( \dot{\phi} = \Lambda \phi \).

It can easily be seen that: \( \Lambda = (P'T')A(TP) \)

The transformation \( TP \) is an \( M \times N \) matrix. The above looks suspiciously like an eigendecomposition of \( A \), i.e., \( TP \) contains \( M \) of the eigenvectors of \( A \)

A 2. Assume that “a” eigenvectors of the matrix \( A \) are active in the snapshot ensemble \( X \), i.e.,

\[ x_i = \sum_{j=1}^{a} \alpha_j^i v_j, \]

where \( a \leq M \) and without loss of generality, it is assumed that the active eigenvectors consist of the first “a” eigenvectors. This assumption essentially implies that the number of modes active within the snapshots is less than the number of snapshots in the ensemble.
Random Iterative POD (RI-POD)

**Proposition 1.** The columns of the transformation \( TP \) are the eigenvectors of \( A \) with corresponding eigenvalues encoded in the diagonal matrix \( \Lambda \), i.e.,

\[
A(TP) = \Lambda(TP).
\]  

(12)

**Remark 1.** Suppose that \( a > M \), i.e., the number of active eigenvectors are more than the number of snapshots. WLOG, let \( a = N \). Then

\[
\tilde{A} = T'AT = \Sigma_p^{-1/2}V_p'X'AXV_p\Sigma_p^{-1/2}
= (\Sigma_p^{-1/2}V_p'\alpha')V'AV(\alpha V_p\Sigma_p^{-1/2})
= (\Sigma_p^{-1/2}V_p'\alpha')\Lambda(\alpha V_p\Sigma_p^{-1/2})
= \beta' \Gamma \beta,
\]

where \((\beta, \Gamma)\) 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 \( A \) to conclude that the transformation \( T\beta \) contains the eigenvectors of \( A \). In fact, some of them might be the same as the eigenvectors of \( A \), however, it is not necessary. In particular, theoretically, we cannot conclude anything regarding the relationship of \( T\beta \) to the eigenfunctions of \( A \).
Random Iterative POD (RI-POD)

The results in the previous slide imply that we are *guaranteed to pick out the eigenvectors in snapshot POD if the number of participating e-vectors is less than the number of snapshots.*

How do we assure that *all possible eigenmodes are actually excited* in the POD?

**RANDOMIZATION.**

**Proposition 2.** Let the initial condition to the linear system in Eq. 5 be chosen according to a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. Let the $j^{th}$ such trajectory be denoted by $X^{(j)}$. Then, every eigenfunction of $A$ is excited almost surely, i.e., given any eigenfunction, there is at least one trajectory $X^{(j)}$ such that the eigenfunction is active within the ensemble as $j \to \infty$.

In particular, it may be shown that the *component of the $i^{th}$ eigenvector is zero mean Gaussian with variance* given by:

$$E|(x(t), v_i)|^2 = e^{2\lambda_i t} E|(x(0), v_i)|^2$$

$$= e^{2\lambda_i t} v_i' R_0 v_i = \sigma^2 e^{2\lambda_i t}.$$
Random Iterative POD (RI-POD)

A 3. We assume that there are $K$ characteristic timescales embedded in the matrix $A$, namely $T_1, \ldots, T_K$. Let the eigenvalues corresponding to timescale $T_j$ be $\{\lambda_1^{(j)}, \ldots, \lambda_{M_j}^{(j)}\}$ and let the corresponding eigenvectors be $[v_{1}^{(j)}, \ldots, v_{M_j}^{(j)}] \equiv V^{(j)}$. Further, we assume that the timescales are well-separated, i.e., if for some $t$, $e^{\lambda_k^{(j)}t} \neq 0$, then $e^{\lambda_k^{(i)}t} \approx 0$ 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.

The above assumption implies that the eigenvalues of the operator $A$ are well-separated. This suggests a method to extract all possible eigenfunctions of the operator in an iterative fashion as follows:

First consider the longest time scale $T_1$. Due to the assumption above, if the start time of the snapshots is large enough and the number of snapshots is high enough, then Propositions 1 and 2 assure us that all the eigenfunctions at the longest time scale can be isolated using the randomized POD.

Then consider the second longest timescale $T_2$. Having knowledge of the slowest eigenvectors and eigenvalues, we may clean the ensemble by subtracting the contributions of these eigenvectors:

$$\tilde{x}(t_j^{(2)}) = x(t_j^{(2)}) - \sum_{k=1}^{M_1} e^{\lambda_k^{(1)}t_j^{(2)}} (x_0, v_k^{(1)}) v_k^{(1)}.$$
Random Iterative POD (RI-POD)

Thus, given that we have stripped the slowest eigenvectors out, we are now back to square one, and thus, by choosing a long enough start time and large enough number of snapshots, we are assured (theoretically) that we can iteratively obtain all the eigenfunctions of operator $A$ summarized in the RI-POD procedure below:

**Algorithm 1 Algorithm RI-POD**

1) Given timescales $T_1, \ldots, T_K$
2) Set $i = 1$, $V^{(0)} = \phi$
3) WHILE $i \leq K$
   DO
   a) Choose snapshot times $t_0^{(i)}, \ldots, t_M^{(i)}$, such that $t_0^{(i)} >> T_{i+1}$ and $M > M_i$
   b) Set $j = 1$
      i) Choose $x_0^{(j)}$, the initial condition of the $j^{th}$ snapshot ensemble at time scale $T_j$, from $N(0, \sigma^2 I)$ and generate the $j^{th}$ snapshot ensemble $X_j^{(i)}$
      ii) Clean all the slower eigenfunctions from the snapshot ensemble using Eq. 16, and the previously extracted eigenfunctions from the sets $V^{(1)}, V^{(2)}, \ldots V^{(i-1)}$
      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 $V^{(1)}, \ldots, V^{(K)}$
Random Iterative POD (RI-POD)

The **RI-POD procedure** can be summarized in the following result:

**Proposition 3.** Under assumptions A1-A3, the RI-POD algorithm can extract all eigenfunctions $V^{(i)}$ corresponding to any given time scale $T^{(i)}$.

Practically though, we can extract only the first few timescales which is typically good enough for (our) applications.

The **RI-POD procedure is an entirely data based procedure**, i.e., it requires no knowledge of the system matrix $A$. To see this, note that:

\[ \tilde{A} = T' AT = T' AXV_p \Sigma_p^{-1/2} = T' \tilde{X} V_p \Sigma_p^{-1/2} \]

Discrete time case: $\tilde{X}$ is one time step advanced version of the snapshot ensemble $X$.

Continuous time case: advance the snapshots by a very short time $T$, $X = X_0 X$, where $X_0$ is the short time advanced ensemble, and then obtain $\tilde{X} = AX^T$. 

where $\tilde{X}$ denotes the one time-step shifted snapshot ensemble $X$ (similar to the DMD procedure [28])
Non Self-Adjoint Operators

We can prove that RI-POD also works well for non self-adjoint operators. With random initial condition, we can get right eigenvectors from simulation data $X$ and left eigenvectors from adjoint simulation data $Y$, using $A'$. 

**RI-POD using input/ output description**

Consider a discrete time system:

$$
x_k = Ax_{k-1} + Bu_{k-1},
\quad y_k = Cx_k. \quad (14)
$$

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

$b_i$: initial conditions for the simulation of the system $A$ to extract a set of right eigenvalue-eigenvectors $(\Lambda_r, V_r)$.

$c_j'$: initial conditions for the simulations of the adjoint system $A'$ to extract a set of left eigenvalues and eigenvectors denoted by $(\Lambda_l, V_l)$.

Of these eigenvalue-eigenvectors pairs, we only keep those left/ right eigenvectors that correspond to the eigenvalues in the intersection of $\Lambda_l$ and $\Lambda_r$. Thus, this allows us to keep the modes of the system that are both observable and controllable.
Application to DPS Filtering

We adopt an ROM based approach to solving the DPS filtering problem. In particular, we shall apply the RI-POD procedure to form the ROM of a DPS by keeping only the first $N_r$ modes of the system.

Recall the (discretized) DPS filtering problem:

$$\dot{x} = Ax + w$$
$$y(t_k) = Cx(t_k) + v(t_k)$$

Proposition 4. The expected value of the squared error in keeping only $N_r$ modes in the ROM is given by

$$E||e(t)||^2 = \sum_{i=N_r+1}^{N} e^{2\lambda_i t} E|\langle x(0), v_i \rangle|^2 + \sum_{i=N_r+1}^{N} \sigma_i^2 \left(\frac{e^{2\lambda_i t} - 1}{2\lambda_i}\right),$$

$$\sigma_i^2 = v'_i R_w v_i,$$  \hspace{1cm} (19)

where $R_w$ represents the covariance of the white noise process $w$. The first term in the above expression is due to the initial conditions while the second term is due to the random perturbation $w$. 
Application to DPS Filtering

The previous result gives an *a priori bound on the error incurred in the estimate of the state if we use the ROM consisting of the $N_r$ modes* where:

$$E||e(t)||^2 \leq \sum_{i=N_r+1}^N e^{2\lambda_i t} E|(x(0), v_i)|^2 + \sum_{i=N_r+1}^N \frac{\sigma_i^2}{2\lambda_i} \cdot$$

The filtering ROM is the following:

$$\dot{\psi} = (V'_i A V_r)\psi + V'_i B w, \quad \psi_i(0) = (x(0), u_i),$$

$$y(t_k) = (C V_r)\psi + v_k,$$

$\psi_i$: $i^{th}$ component of the ROM state $\psi$.

We filter the system by using a standard *Kalman filter on the ROM* above.
DPS Filtering: Heat Equation

The heat transfer by conduction along a slab is given by the partial differential equation:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2},$$  \hspace{1cm} (16)

thermal diffusivity: \( \alpha = 0.01 \)

The slab is assumed to be heated to 300 °C at its left end and the temperature maintained there, the right end is insulated. The temperature of the slab is measured at five equispaced points along its length.
Heat Equation

The comparisons between the reduced order filter and real system at five different time steps are shown in Figure 1(a). The red curves are the real heat profiles and the blue curves are filter results from the reduced model. In Figure 1(b)-(d), the error and the $3\sigma$ boundary for the reduced model, at three different chosen location are shown too.

Figure 1: Filter results for heat equation
2-D Pollutant Transfer

The two-dimensional pollutant transport equation describing the contamination transport is:

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

(21)

\(c\) : concentration of the contaminant  
\(D\) : dispersion  
\(\nu\) : velocity in the \(x\) and \(y\) directions  
\(S_s\) : source of pollutant.

In simulation, there are three obstacles and three sources in the field. A random initial condition is used for generating the reduced order model, and initial condition for simulation is zero. We use Neumann boundary conditions, and take ten measurements equispaced along the diagonal of the field.
2-D Pollutant Transport

Figure 5 (a) and (b) compare the filter results with the actual field at the end of the simulation.

(a) ROM Filter result at the end of simulation
(b) Actual field at the end of simulation
(c) Error and 3σ error bounds at point (45, 45)
(d) Error and 3σ error bounds at state component (15, 10)

Figure 5: Filter results for Pollutant transport equation

Also, in Figure 5 (c)-(d), the errors and the 3σ boundary for the reduced model at two different chosen location are shown.
2-D Pollutant Transport

The eigenvalues extracted from the model using the actual system, and its adjoint, utilizing the input/output description, of the system are shown below corresponding to the respective right and left eigenvectors.
Conclusions

We have proposed a randomized iterative snapshot POD based approach (RI-POD) that is capable of extracting all the eigenfunctions of a linear operator. The method is completely data-based.

We have applied the RI-POD to obtain an ROM based Kalman filtering technique for the filtering of DPS.

The method works well for linear problem but has issues for nonlinear problems due to the sensitivity to the initial conditions used to generate the ROM. Thus, there is a need to extend the RI-POD in a systematic fashion to the nonlinear case. We would also like to account for unknown forcing terms and parameters.

The ROM based filtering technique also has to be applied to more realistic 2 and 3-dim. DPS. In particular, we would like to address the problems of pollutant transport, flow estimation and the estimation of wildfires.

We would also like to solve the problem of optimal sensor placement design, both static and sequential, such that the information content regarding the process being monitored is maximized: one of the goals of DDDAS.
References


