

*Article*

# Interplay of Sensor Quantity, Placement and System Dimension in POD-based Sparse Reconstruction of Fluid Flows

Balaji Jayaraman\*, S M Abdullah Al Mamun and Chen Lu

<sup>1</sup> School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK, USA

\* Author to whom correspondence should be addressed; balaji.jayaraman@okstate.edu.

---

**Abstract:** Sparse recovery of fluid flows using data-driven proper orthogonal decomposition (POD) basis is systematically explored in this work. Fluid flows are manifestations of nonlinear multiscale PDE dynamical systems with inherent scale separation that impact the system dimensionality. Given that sparse reconstruction is inherently an ill-posed problem, the most successful approaches require the knowledge of the underlying basis space spanning the manifold in which the system resides. In this study, we employ an approach that learns basis from singular value decomposition (SVD) of training data to reconstruct sparsely sensed information. This results in a set of four control parameters for sparse recovery, namely, the choice of basis, system dimension required for sufficiently accurate reconstruction, sensor budget and their placement. The choice of control parameters implicitly determines the choice of algorithm as either  $l_2$  minimization reconstruction or sparsity promoting  $l_1$  norm minimization reconstruction. In this work, we systematically explore the implications of these control parameters on reconstruction accuracy so that practical recommendations can be identified. We observe that greedy-smart sensor placement provides the best balance of computational complexity and robust reconstruction for marginally oversampled cases which happens to be the most challenging regime in the explored parameter design space.

**Keywords:** sparse reconstruction, sensors, cylinder flow, SVD, POD, compressive sensing

---

## 1. Introduction

Multiscale fluid flow phenomena in engineering and geophysical settings are invariably data-sparse, i.e. there are more scales to resolve than there are sensors. A major goal is to recover more information

about the dynamical system through reconstruction of the higher dimensional state. To expand on this view, for many practical fluid flow applications, accurate simulations may not be feasible for a multitude of reasons including, lack of accurate models, unknown governing equations or extremely complex boundary conditions. In such situations, measurement data represents the absolute truth and is often acquired from very few probes that limits potential for in-depth analysis. A common recourse is to combine such sparse measurements with underlying knowledge of the flow system, either in the form of idealized simulations or phenomenology or knowledge of a sparse basis to recover detailed information. The former approach is termed as data assimilation while we refer to the latter as Sparse Reconstruction (SR). On the other hand, simulations typically represent a data surplus setting that offer the best avenue for analysis of realistic flows as one can identify and visualize coherent structures, perform well converged statistical analysis including quantification of spatiotemporal coherence and scale content due to the high density of data probes in the form of computational grid points. With growth in computing power, they often generate *big data* contributing to an ever growing demand for quick analytics and machine learning tools [1] to both sparsify, i.e. dimensionality reduction [2–5] and reconstruct the data without loss of information. Thus, tools for *encoding* information into a low-dimensional feature space (convolution) complement sparse recovery tools that help *decode* compressed information (deconvolution). This in essence is a key aspect of leveraging machine learning for fluid flow analysis [6,7]. This work contributes to the broad area of *reconstructing* high resolution fields from sparse data.

A primary target of this work is to address the practical problems of flow sensing and control in the field or a laboratory where a few affordable probes are expected to sense effectively. Recent advances in compressive sensing (CS) [8–11] have opened the possibility of direct sparse sampling [6] of data in real-time without having to collect high resolution information and then downsample. Of course, in the case of direct sampling, there is need to consider a generic basis in which the data has a high probability of being sparse in. Whereas, in the case of collecting high resolution information and then down sampling, one can learn optimal basis from data. Thus, decoding and reconstruction from sparse data have been gaining popularity in their various manifestations such as Gappy Proper Orthogonal Decomposition (GPOD) [12,13], Fourier-based Compressive Sensing (CS) [8–11] and Gaussian kernel-based Kriging [14–16]. A parallel application of such ideas is in the acceleration of nonlinear model order reduction using sparse sampling as hyper-reduction [17–20] tools. Most literature on this topic of sparse recovery focus on theoretical guarantees and estimation of performance bounds or application-driven demonstrations [6,7]. The contribution from this work is three-fold. Firstly, we develop a systematic framework for characterizing the SR performance in terms of accuracy of data recovery that can shed light on practical usage limitations and help with simulation design. Secondly, we explore how these SR methods interact and potentially gain from greedy and smart sensor placement. Thirdly, putting these together, we present a systematic assessment of the interplay between system dimensionality (i.e. a measure of the chosen system complexity), sensor quantity (budget) and their placement and provide recommendations for use in practice for fluid flow applications. As a first step in this direction, we focus our efforts on SR of low-dimensional transient wake flow behind a cylinder at laminar Reynolds numbers ( $Re = 100$ ).

Sparse reconstruction (SR) is inherently ill-posed, under-determined inverse problem where the number of constraints (i.e., sensor quantity) are much less than the number of unknowns (i.e., high resolution field). However, if the underlying system is sparse in a feature (space of basis coefficients) space then the probability of recovering a unique solution increases by solving the reconstruction problem in this lower-dimensional space. The core theoretical developments of such ideas and their first practical applications happened in the realm of image compression and restoration [10,21].

**Sparse Reconstruction with Data-driven Basis:** Data recovery techniques based on Karhunen-Loeve (K-L) procedure with least squares ( $l_2$ ) error minimization, also known as Gappy POD or GPOD [12,13, 17], was originally developed in the nineties to recover marred faces [17] in images. The fundamental idea is to utilize the POD basis computed offline from the data ensemble to *encode* the reconstruction problem into a low-dimensional feature space. This way, the sparse data can be used to recover the sparse unknowns in the space of POD coefficients by minimizing the  $l_2$  errors. If the data-driven POD basis are not known *a priori*, an iterative formulation [12,17] to successively approximate the POD basis and the coefficients was proposed with limited success. While this approach has been shown to work in principle [12,14,22], it is prone to numerical instabilities and inefficiency. Advancements in the form of a progressive iterative reconstruction framework [14] are effective, but impractical for real-time application. In fact, all the aforementioned issues are related to the POD-basis being data-driven and therefore, can represent the data optimally but not easily generalizable. This requires that they be known *a priori* - a stringent requirement for many practical systems as training data is rarely available and even when it is, it may not effectively (i.e. relevant to) the prediction regime. Such limitations make data-driven basis hard to use for practical sparse sensing-based reconstruction. Nevertheless, they find tremendous value in data-driven modeling (machine learning, Koopman operator models [23,24]) applications and nonlinear model order reduction [4] of systems that are statistically stationary where training data is available and tends to remain relevant to the system as it evolves.

**Sparse Reconstruction with Generic Basis:** A way to bypass the above limitations is to use generic basis such as wavelets [25] or Fourier functions. Such choices assume that most practical systems are relatively sparse with respect to these basis choices although not optimal. This is particularly true for image processing applications but may not be optimal for fluid flows whose dynamics obey PDEs (and may include sharp gradients). While avoiding the cost of computing the basis offline, such approaches run into sparsity issues as the basis do not optimally encode the underlying dynamical system. In other words, larger the basis, more sensors are needed for complete and accurate reconstruction. Thus, once again the reconstruction problem is ill-posed because the basis space is not sufficiently low-dimensional resulting in the flow being under sampled. To bypass this, one needs to look for a sparse solution which is usually not realized through  $l_2$  error minimization approaches. The magic of Compressive Sensing (CS) [8–11] is in its ability to overcome this constraint by seeking a solution that can be less sparse than the dimensionality of the chosen feature space using  $l_1$ -norm regularized least-squares reconstruction. Such methods have been successfully applied in image processing using Fourier or wavelet basis and also to fundamental fluid flows [6,7,26–29]. Compressive sensing essentially looks for a regularized sparse solution using  $l_1$  norm minimization of the sparse coefficients by solving a computationally

manageable convex optimization problem. Such sparsity promoting  $l_1$  regularized reconstruction can also be combined with data-driven POD basis to overcome the aforementioned limitations. Successful attempts at this include reconstruction of sparse PIV data [6] and pressure measurements around a cylinder surface [7]. To bypass the limitations from using data-driven POD basis for SR, one often tends to build a library of POD modes from precursor training simulation or measurement data over a range of flow parameters (e.g. Reynolds numbers) and leverage  $l_1$  regularized reconstruction to identify the most relevant  $K$ -sparse solution. Building such a library of POD modes is a one-time investment for a chosen class of flows. Such a framework has been attempted in [7] where POD modes from simulations over a range of Reynolds ( $Re$ ) numbers of a cylinder wake flow were used to populate a library of bases and then used to classify the flow regime based on sparse measurements.

In order to carry out a systematic analysis of interplay between the various SR control parameters, namely, desired reconstruction dimension, sensor budget and their placement, we limit ourselves to primarily  $l_2$  minimization approaches (except for comparative analysis where we use  $l_1$ ) and four different sensor placement methods including random sampling (as used in [30]), QR with column pivoting [31,32], Discrete Empirical Interpolation Method (DEIM) [18] and explicit condition number minimization [13]. The suitability of these choices are numerically explored in a discretized parameter space of sensor budget and system dimension. The rest of manuscript is organized as follows. In section 2 we review the basics of sparse reconstruction theory (sec. 2.1), computation of data-driven POD basis (sec. 2.2), role of measurement locations (sec. 2.3), algorithms for sensor placement (sec. 2.4) and sparse recovery (sec. 2.5). In section 3 we discuss how the training data is generated and also summarize the outcomes from sensor placement using the different approaches. In section 4 we discuss the results from our analysis of the SR of the cylinder wake flow using both POD and ELM basis. This is followed by a summary of major conclusions from this study in section 5.

## 2. Formulating the Sparse Reconstruction Problem

Given a high resolution data representing the state of the flow system at any given instant denoted by  $x \in \mathbb{R}^N$ , its corresponding sparse representation given by  $\tilde{x} \in \mathbb{R}^P$  with  $P \ll N$ . Then, the sparse reconstruction problem is to recover  $x$ , when given  $\tilde{x}$  along with information of the sensor locations in the form the measurement matrix  $\mathbf{C} \in \mathbb{R}^{P \times N}$  as shown in eqn. (1). The measurement matrix  $\mathbf{C}$  determines how the sparse data  $\tilde{x}$  is collected from  $x$ . Variables  $P$  and  $N$  are the number of sparse measurements and the dimension of the high resolution field, respectively.

$$\tilde{x} = \mathbf{C}x. \quad (1)$$

In this article, we focus on vectors  $x$  that have a sparse representation in a basis space  $\Phi \in \mathbb{R}^{N \times K}$  such that  $K \ll N$  and yielding  $x = \Phi a$ . Naturally, when one loses the information about the system, the recovery of said information is not absolute as the reconstruction problem is ill-posed, i.e., more unknowns than equations in eqn. (1). Thus the most straightforward approach to recover  $x$  is by computing the inverse of  $\mathbf{C}$  using a least-squares error minimization procedure as shown in eqn. (2). However, this does not result in *stable solutions* as the system is ill-posed (under-determined).

$$\mathbf{C}^+ \tilde{x} = x. \quad (2)$$

## 2.1. Sparse Reconstruction Theory

The theory of sparse reconstruction has strong foundations in the field of inverse problems [33] with applications in diverse fields of study such as a geophysics [34,35] and image processing [36,37]. In this section, we formulate the reconstruction problem which has been presented in CS literature [8,10,38–40]. Many signals tend to be “compressible”, i.e., they are sparse in some  $K$ -sparse basis  $\Phi$  as shown below:

$$x = \sum_{i=1}^{N_b} \phi_i a_i \quad \text{or} \quad x = \Phi a, \quad (3)$$

where  $\Phi \in \mathbb{R}^{N \times N_b}$  and  $a \in \mathbb{R}^{N_b}$  with  $K$  non-zero elements. In the sparse reconstruction formulation above,  $\Phi \in \mathbb{R}^{N \times N_b}$  is used instead of  $\Phi \in \mathbb{R}^{N \times K}$  as the  $K$  most relevant basis vectors for a given data are not identified *a priori*. Consequently, a more exhaustive basis set of dimension  $N_b \approx P > K$  is typically employed. To recover  $N$ -dimensional data, one can atmost use  $N$  basis vectors, i.e.,  $N_b \leq N$ . In practice, the dimension of the candidate basis space need not be  $N$  and can be represented by  $N_b \ll N$  as only  $K$  ( $\leq N_b$ ) of them are needed to represent the acquired signal up to a desired quality. This is typically the case when  $\Phi$  is composed of optimal data-driven basis vectors such as POD modes. The reconstruction problem is then recast as identification of these  $K$  sparse coefficients.

In many practical situations,  $\Phi$  and  $K$  along with  $N_b, N$  are input by the user. Standard *transform coding* [25] practice in image compression involves collecting a high resolution sample, transforming it to a Fourier or wavelet basis space where the data is sparse, retain the  $K$  most relevant coefficients while discarding the rest of the information. This is the basis of JPEG and JPEG-2000 compression standards [25]. The *sample and then compress* mechanism still requires acquisition of high resolution samples and processing them before reducing the dimensionality. This is highly challenging as handling large amounts of data is difficult in practice due to demands on processing power, storage, and time. Compressive sensing [8,10,38–40] focuses on direct sparse sensing based inference of the  $K$ -sparse coefficients by essentially combining the steps in equations 1 and 3 as below:

$$\tilde{x} = \mathbf{C} \Phi a = \Theta a, \quad (4)$$

where  $\Theta \in \mathbb{R}^{P \times N_b}$  is the map between the basis coefficients  $a$  that represent the data in a feature space and the sparse measurements,  $\tilde{x}$  in physical space. The challenge in solving for  $x$  using the under determined eqn. (1) is that  $\mathbf{C}$  is ill-conditioned and  $x$  in itself is not sparse. However, when  $x$  is sparse in  $\Phi$ , the reconstruction using  $\Theta$  in eqn. (4) becomes practically feasible (for  $P \gtrsim K$ ) by solving for  $a$ . Thus, one effectively seeks a  $K$ -sparse  $a$  with  $P$  constraints (given by  $\tilde{x}$ ) using established methods from linear algebra and constrained optimization.

### 2.1.1. Case 1: For $K = N_b$

For the over determined system with  $P > K = N_b$ ,  $a$  is estimated using a regularized least squares solution based on the normal equation as  $a = (\Theta)^{L+} \tilde{x} = (\Theta^T \Theta + \lambda I)^{-1} \Theta^T \tilde{x}$  for a chosen  $\lambda$ . This is obtained by minimizing the appropriate cost function given by  $J_{cost} = \|\tilde{x} - \Theta a\|_2^2 + \lambda \|a\|_2^2$ . This regularized least-squares solution procedure for the overdetermined case is nearly identical to the original GPOD algorithm developed by Everson and Sirovich [17] if  $\Phi$  is chosen as the POD basis. However,

$\tilde{x}$  in GPOD contains zeros as placeholders for all the missing elements whereas the above formulation retains only the measured data points.

When  $P \leq K = N_b$ , and the system is under-determined with non-unique solutions, one looks for a minimum norm reconstruction of  $a$ . This is achieved by minimizing the corresponding  $s$ -norm of  $a$  ( $s$  chosen appropriately) and  $x$  is then recovered from eqn. (3).  $s$  chosen as 2 yields the minimum  $l_2$  norm reconstruction of  $x$  by penalizing the larger elements of  $a$ . The  $l_2$ -regularized method finds an  $a$  that minimizes the expression shown in eqn. (5).

$$\begin{aligned} l_2 \text{ norm minimization reconstruction : } a &= \operatorname{argmin} \|a'\|_2 \text{ such that } \Theta a' = \tilde{x} \\ l_2 \text{ cost function to be minimized : } &\min\{\mathcal{T}(\tilde{x} - \Theta a) + \|a\|_2^2\} \end{aligned} \quad (5)$$

One can solve for  $\mathcal{T}$  and  $a$  in eqn. (5) using Lagrange multipliers approach to yield a solution that is a right pseudo-inverse of  $\Theta$  as in eqn. (6) below as long as  $\Theta$  has a full rank of  $P$ :

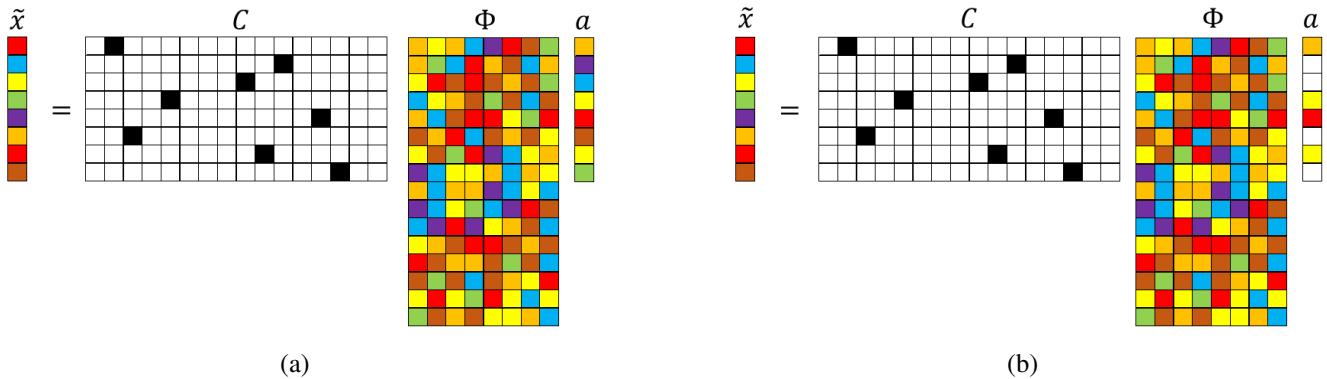
$$a = (\Theta)^{R+} \tilde{x} = \Theta^T (\Theta \Theta^T)^{-1} \tilde{x}. \quad (6)$$

### 2.1.2. Case 2: For $K < N_b$

When  $K \ll N_b$ , one typically looks for a sparse solution of  $a$ . The  $l_2$  norm minimization and  $l_2$  regularization approaches discussed above provide numerical stability, but do not produce sparse solutions. A natural way to enhance sparsity of  $a$  is to minimize  $\|a'\|_0$ , i.e., minimize the number of non-zero elements such that  $\Theta a' = \tilde{x}$  is satisfied. It has been shown [41] that with  $P = K+1$  ( $P > K$  in general) *independent* measurements, one can recover the sparse coefficients with high probability using minimum  $l_0$  norm reconstruction. This condition can be heuristically interpreted as each measurement needing to excite a different basis vector  $\phi_i$  so that its coefficient  $a_i$  can be optimally identified. If two or more measurements excite the same basis  $\phi_j$  then additional measurements may be needed to produce acceptable reconstruction. On the other hand, for  $P \leq K$  independent measurements, the probability of recovering the sparse solution is highly diminished. Nevertheless,  $l_0$ -reconstruction is a computationally complex,  $np$ -complete and poorly conditioned problem with no stability guarantees.

$$\begin{aligned} l_1 \text{ norm minimization reconstruction : } a &= \operatorname{argmin} \|a'\|_1 \text{ such that } \Theta a' = \tilde{x} \\ l_1 \text{ cost function to be minimized : } &\min\{\|\tilde{x} - \Theta a\|_2^2 + \lambda \|a\|_1\} \end{aligned} \quad (7)$$

The popularity of compressed sensing arises on account of the theoretical advances [42–45] guaranteeing near-exact reconstruction of the uncompressed information by solving for the  $K$  sparsest coefficients using  $l_1$  norm minimization methods. The  $l_1$  reconstruction is a relatively simpler convex optimization problem (as compared to  $l_0$ ) and solvable using linear programming techniques for basis pursuit [8,38,46] and shrinkage methods [47]. Theoretically, one can perform the simplistic brute force search to locate the largest  $K$  coefficients of  $a$  that satisfy eqn. (7), but the computational effort increases exponentially with dimension. To overcome this burden, a host of greedy algorithms [9,11,48] have been developed to solve the  $l_1$  norm minimization problem in eqn. (7) with complexity  $\mathcal{O}(N^3)$  for  $N_b \approx N$ . However, this approach requires  $P > \mathcal{O}(K \log(N_b/K))$  measurements [8,38,42] to reconstruct the  $K$ -sparse vectors with high probability. The schematic illustration of both  $l_2$  and  $l_1$ -based formulations



**Figure 1.** Schematic illustration of  $l_2$  (left) and  $l_1$  (right) minimization reconstruction for sparse recovery using a single-pixel measurement matrix. The numerical values in  $C$  are represented by colors: black (1), white (0). The other colors represent numbers that are neither 0 nor 1. In the above schematics  $\tilde{x} \in \mathbb{R}^P$ ,  $C \in \mathbb{R}^{P \times N}$ ,  $\Phi \in \mathbb{R}^{N \times N_b}$  and  $a \in \mathbb{R}^{N_b}$ , where  $N_b \leq N$ . The number of colored cells in  $a$  represents the system sparsity  $K$ .  $K = N_b$  for  $l_2$  and  $K < N_b$  for  $l_1$ .

are presented in Figure 1.

Solving the  $l_1$  minimization problem shown in eqn. (7) is complicated relative to the  $l_2$  minimization solution described in eqn. (5). This is because, unlike the cost function to be minimized in eqn. (5), the cost function in eqn. (7) is not differentiable when  $a_i = 0$  which necessitates an iterative solution instead of a closed form solution. Further, the minimization of the  $l_1$  cost function is also an unconstrained optimization problem that is commonly converted into a constrained optimization as shown in eqn. (8). Here  $t$  is a user defined sparsity knob to ‘shrink’ the coefficients.

$$l_1 \text{ norm constrained minimization : } \min \|\tilde{x} - \Theta a\|_2^2 \text{ such that } \|a\|_1 < t \quad (8)$$

This constrained optimization in eqn. (8) is quadratic in  $a$  and therefore, a quadratic programming problem with the feasible region bounded by polyhedron (in the space of  $a$ ). There exists two classes of  $l_1$  solution methodologies: (i) least absolute selection and shrinkage operator or LASSO [47] and (ii) basis pursuit denoising [46]. LASSO and its variant essentially convert the constrained formulation into a set of linear constraints. Recently popular approaches include greedy methods such as optimal matching pursuit(OMP) [11,27] and interior point methods [49]. An intuitive iterative sequential least-squares thresholding framework is used by Brunton et al. [50]. The idea here is to achieve ‘shrinkage’ by repeatedly zeroing out the coefficients smaller than a given choice of hyperparameter.

In summary, the reconstruction framework is characterized by three parameters,  $N_b, K, P$ .  $N_b$  is the candidate basis dimension employed for this reconstruction which is bounded by  $N$ , i.e.  $N_b \lesssim N$ .  $K$  represents the desired reconstruction sparsity and chosen such that if these features are predicted accurately, then the achieved reconstruction meets a desired quality. The more sparse a system, the smaller  $K$  is for a desired reconstruction quality and is specified by the user based on knowledge of the system.  $P$  represents the available sensor quantity as input to the problem.  $N$  is the preferred dimension of the reconstructed state. The interplay of  $N_b, K$ , and  $P$  determine the choice of algorithm employed,

i.e., whether the reconstruction is based on least squares minimization,  $l_2$  norm minimization or sparsity enabling  $l_1$  approaches as summarized in Table 1.

**Table 1.** The choice of sparse reconstruction algorithm based on problem design using parameters  $P$  (sensor sparsity),  $K$  (targeted reconstruction sparsity) and  $N_b$  (candidate basis dimension).

Case	$K - N_b$ Relationship	$P - K$ Relationship	Algorithm	Reconstructed Dimension
1	$K = N_b$	$P \geq K$	least squares ( $l_2$ )	$K$
2	$K = N_b$	$P < K$	min. norm recons. ( $l_1$ ) or ( $l_2$ )	$P$
3	$K < N_b$	$P \geq K$	min. norm recons. ( $l_1$ )	$K$

All of the above sparse recovery estimations are conditional upon the measurements (rows of  $C$ ) being incoherent with respect to the sparse basis  $\Phi$ . This is usually accomplished by using a random choice of sensor placement, especially when  $\Phi$  is made up of Fourier functions or wavelets. If the basis functions  $\Phi$  are orthonormal, such as wavelet or POD basis (with inherent hierarchy), one can discard the majority of the small coefficients in  $a$  (setting them as zeros) and still retain reasonably accurate reconstruction. The mathematical explanation for this has been previously shown in [10]. However, it should be noted that incoherency is a necessary, but not sufficient condition for exact reconstruction. Exact reconstruction requires optimal sensor placement to capture the most information for a given flow field. In this study, we assess the interplay between sensor quantity, quality and desired system sparsity for POD-based sparse reconstruction.

## 2.2. Data-driven Basis Computation using POD

In the SR framework, basis such as POD modes, Fourier functions, and wavelets [8,10] can be used to generate low-dimensional representations for both  $l_2$  and  $l_1$ -based methods. While an exhaustive study on the effect of different choices on reconstruction performance is potentially useful, in this study we focus on POD-based SR. A similar effort has been reported in [6] where a comparison between discrete cosine transform and POD bases was performed.

Proper orthogonal decomposition (POD), also known as Principal Components Analysis (PCA) or Singular Value Decomposition (SVD), is a dimensionality reduction technique that computes a linear combination of low-dimensional basis functions (POD modes) and weights (POD coefficients) from snapshots of experimental or numerical data [2,4] through eigen-decomposition of the spatial correlation tensor of the data. It was introduced in the turbulence community by Lumley [51] to extract coherent structures in turbulent flows. The resulting singular vectors or POD modes represent an orthogonal basis that maximizes the energy capture from the flow field. For this reason, such eigenfunctions are considered *optimal* in terms of energy capture and other optimality constraints are theoretically possible. Taking advantage of the orthogonality, one can project these POD basis onto each snapshot of data in a Galerkin sense to deduce coefficients that represent evolution over time in the POD feature space. The optimality of the POD basis also allows one to effectively reconstruct the full field information with knowledge of very few coefficients, a feature that is attractive for solving sparse reconstruction problems such as in eqn. (4). However, this is contingent on the spectrum of the spatial correlation tensor of the data having sufficiently rapid decay of the eigenvalues, i.e. it supports a low-dimensional representation. This

is typically not true in the case of turbulent flows with very gradual decay of energy across the singular values. Further, in such dynamical systems, the small scales with low-energy can still be dynamically important and will need to be reconstructed, thus requiring significant number of sensors.

Since the eigen-decomposition of the spatial correlation tensor of the flow field requires handling a system of dimension  $N$ , it requires significant computational expense. An alternative method is to compute the POD modes using the method of snapshots [52] where the eigen-decomposition problem is reformulated in a reduced dimension (assuming the number of snapshots in time is smaller than the spatial dimension) framework as summarized below. Consider that  $X \in \mathbb{R}^{N \times M}$  (different from  $x \in \mathbb{R}^N$ ) is the full field representation with only the fluctuating part, i.e., the temporal mean is taken out of the data.  $N$  is the dimension of the full field representation and  $M$  is the number of snapshots. The procedure involves computation of the temporal correlation matrix  $\bar{C}_M$  as:

$$\bar{C}_M = X^T X. \quad (9)$$

The resulting correlation matrix  $\bar{C}_M \in \mathbb{R}^{M \times M}$  is symmetric and an eigendecomposition problem can be formulated as:

$$\bar{C}_M V = V \Lambda, \quad (10)$$

where the eigenvectors are given in  $V = [v_1, v_2, \dots, v_M]$  and the diagonal elements of  $\Lambda$  are the eigenvalues  $[\lambda_1, \lambda_2, \dots, \lambda_M]$ . Typically, both the eigenvalues and corresponding eigenvectors are sorted in descending order such as  $\lambda_1 > \lambda_2 > \dots > \lambda_M$ . The POD modes  $\Phi$  and coefficients  $a$  can then be computed as

$$\Phi = X V \sqrt{\Lambda^{-1}}. \quad (11)$$

One can represent the field  $X$  as a linear combination of the POD modes  $\Phi$  as shown in eqn. (3) and leverage orthogonality to compute the Moore-Penrose pseudo-inverse, i.e.,  $\Phi^\dagger = \Phi^T$ , and compute the POD coefficients,  $a \in \mathbb{R}^{M \times M}$  as shown in eqn. (12),

$$a = \Phi^T X. \quad (12)$$

It is worth mentioning that subtracting the temporal mean from the input data is not critical to the success of this procedure. In fact, retaining the mean of the data during SVD computation generates an extra mean mode which modifies the energy spectrum. For the low-dimensional cylinder wake used in this study, the dominant mode when performing SVD with mean captures 98% energy whereas for the SVD without mean, the dominant mode captures 49% energy. Based on our experience over the course of this research study, these choices have minimal impact on the reconstruction performance. In fact, during practice, one does not use POD basis with the mean removed as the sparse data includes the mean that is not known *a priori*.

Using the snapshot procedure for the POD/SVD computation fixes the maximum number of POD basis vectors to  $M$  which is typically much smaller than the dimension of full state vector,  $N$ . If one wants to reduce the dimension further, then a criterion based on energy capture is devised so that the modes carrying the least amount of energy *truncated* to dimension  $K < M$ . For many common fluid flows, using the first few POD modes and coefficients are sufficient to capture almost all the relevant dynamics. However, for turbulent flows with large-scale separation, a significant number of POD modes will need to be retained.

### 2.3. Measurement Locations, Data Basis and Incoherence

Recalling from subsection 2.1, the reconstruction performance is strongly tied to the measurement matrix,  $C$  being *necessarily* incoherent with respect to the low-dimensional basis  $\Phi$  [10] and this is usually accomplished by using random elements to populate the matrix. In practice, one can adopt two types of random sampling of the data, namely, single-pixel measurement [7,53,54] or random projections used in compressive sensing [8,10,39]. Typically, single-pixel measurement refers to measuring information at chosen spatial locations such as measurements by unmanned aerial systems (UAS) in the atmospheric fields or point probes. The resulting matrix  $C$  is structured as  $C \leftarrow [e_{\varrho_1}, e_{\varrho_2}, \dots, e_{\varrho_p}]^T$ , where  $e_{\varrho_p}$  is column vector with zeros except at the index  $p$  where it assumes a value of 1.

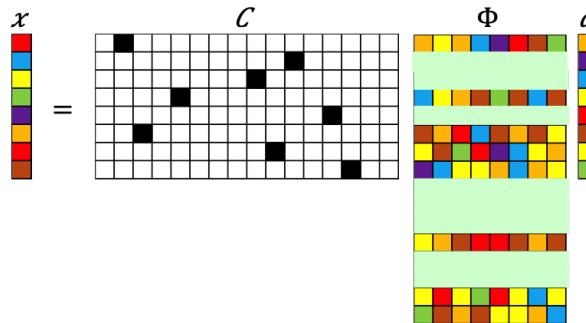
Another popular choice of measurements employed in the compressive sensing or image processing community is random projections where the measurement matrix is populated using random numbers based on a chosen distribution (Gaussian, Bernoulli) on to which the full state data is projected. In theory, the random matrix is highly likely to be incoherent with any generic basis [10], and hence suitable for sparse recovery purposes. However, for most fluid flow and other practical sensing applications, the sparse data is usually sourced from point measurements. Irrespective of the approach adopted, the measurement matrix  $C$  (eqns. (1)-(2)) and basis  $\Phi$  should be incoherent to ensure optimal sparse reconstruction. This essentially implies that one should have sufficient measurements distributed in space to excite the different modes relevant to the data being reconstructed. Mathematically, this is related to  $C\Phi$  being full rank and invertible. There exist metrics to estimate the extent of coherency between  $C$  and  $\Phi$  in the form of an *coherency* number,  $\mu$  as shown in eqn. (13) [55],

$$\mu(C, \Phi) = \sqrt{N} \cdot \max_{i \leq P, j \leq K} |\langle c_i, \Phi_j \rangle|, \quad (13)$$

where  $c_i$  is a row vector in  $C$  (i.e.  $c_i = e_{\varrho_j}$ ) and  $\phi_j$  is a column vector of  $\Phi$ .  $\mu$  typically ranges from 1 (incoherent) to  $\sqrt{N}$  (coherent). The smaller the  $\mu$ , the less measurements one needs to reconstruct the data using minimum norm approaches. This is because the coherency parameter enters as the pre-factor in the lower-bound for the sensor quantity in  $l_1$ -based CS for accurate recovery. There exist optimal sensor placement algorithms such as K-means clustering, the data-driven Online sparse Gaussian Processes [56], physics-based approaches based on inflection in the modal shapes [57] and mathematical approaches [13] that minimize condition number of  $\Theta$  and maximize determinant of the Fisher information matrix [58]. A thorough study on the role of sensor placement on reconstruction quality is much needed and is an active topic of research. In this study, we assess three different greedy approaches for nearly optimal sensor placement in sparse recovery applications, namely, the Discrete Empirical Interpolation Method (DEIM) [18,59], iterative condition number minimization [60] and the reduced matrix  $QR$ -factorization with column pivoting as reviewed in [61]. We compare these approaches with outcomes from a random sensor placement that can be easily implemented using random number generators in *Matlab*. To ensure the resulting measurement matrices are incoherent with respect to the POD basis, we also estimate the coherency numbers in the following discussion.

### 2.4. Algorithms for Sensor Placement

In this section, we summarize the different sensor placement algorithms employed in this research study. Optimal sensor placement for a given data, especially for a flow field that evolves over time is highly challenging and is an ongoing topic of active research. The goal of optimal point sensor placement for reconstruction is to identify and activate only a few rows of the basis matrix  $\Phi$  that effectively conditions the matrix  $\Theta$  (for  $P = K = N_b$ ) or its variants  $\mathbf{M} = \Theta^T \Theta$  or  $\mathbf{M} = \Theta \Theta^T$  depending on if  $P > K = N_b$  or  $P < K = N_b$  respectively). This is schematically illustrated in fig. 2.



**Figure 2.** Schematic illustration of sparse sensor placement. The pastel colored rectangles represent rows activated by the sensors denoted in the measurement matrix through dark squares.

To design smart sensor placement, one needs an optimization criteria which in this case is to minimize reconstruction error when using a small number of sensors which, of course depends on the choice of basis  $\Phi$ . Since reconstruction from sparse data in general requires inversion of  $\Theta$  or  $\mathbf{M}$ , most smart sensing strategies are designed to improve the condition of  $\Theta$ ,  $\mathbf{M}$  for inversion purposes by optimizing their spectral content in the form of its determinant, trace, spectral radius or condition number. A direct method of optimizing such metrics requires searching over the different possible sensor selections resulting in combinatorial complexity. Thankfully, there exist a variety of greedy algorithms [13,18,62] that have been shown to be successful in the context of fluid flow data.

#### 2.4.1. Random Sensor Placement

The most simple and efficient sensor placement strategy is to sample at random locations. This is commonly accomplished using a random number generator for the sensor locations, accomplished easily using packages such as *Matlab* or *Python* with functions such as `randperm()`. In this study, we sample from simulation data by choosing the first  $P$  values from a random permutation of the entire data of dimension  $N$ . We note that it is equally effective to adopt ideas such as K-means clustering as was employed in [63]. Since the experiments performed in this study use highly sparse measurements relative to the total number of grid points, i.e.,  $P \ll N$  to mimic practical data acquisition, we focus on random single-pixel measurements. To better assess the effectiveness of such sensor placement methods, we generate multiple realizations by changing the seed of the random number generator. The outcomes are then quantified in terms of the mean as well as the outliers. This particular choice of sensor placement is designed to serve as an inexpensive benchmark to compare against other greedy sampling methods.

#### 2.4.2. Minimization of Matrix Condition Number (MCN)

As shown in sec. 2.1.1, the success of the reconstruction effort for  $K = N_b$  is tied to the accuracy of the inverse computation of  $\mathbf{M} = \boldsymbol{\Theta}^T \boldsymbol{\Theta}$  or  $\mathbf{M} = \boldsymbol{\Theta} \boldsymbol{\Theta}^T$  (depending on whether the system is over determined ( $P > K = N_b$ ) or under determined ( $P < K = N_b$ )). Therefore, if  $\mathbf{M} = \boldsymbol{\Theta}^T \boldsymbol{\Theta}$  or  $\mathbf{M} = \boldsymbol{\Theta} \boldsymbol{\Theta}^T$  has full column or row rank respectively along with a reasonable condition number, then the estimated inverse has a chance to be accurate. This approach focuses on sensor placement (through the construction of  $C$ ) that minimizes the condition number of  $\mathbf{M}$  or  $\kappa(\mathbf{M})$ . The condition number is directly related to the orthogonality of  $\boldsymbol{\Theta}$  and the presence of significant diagonal entries in  $\mathbf{M}$ . Therefore, this algorithm can be viewed as placing sensors at locations that display significant flow dynamics and also preserve the orthogonality of the POD modes. From a mathematical perspective, the condition number represents the ratio of maximum and minimum singular values of  $\boldsymbol{\Theta}$  or  $\mathbf{M}$ . Therefore, for large  $\kappa(\mathbf{M})$ , the errors tend to be amplified with respect to the signal. As shown by Willcox [13], and Yildrim *et al.* [62] such a method compares favorably to more physics-based approaches [57] where sensors are placed at the extrema of dominant POD modes although the latter method is computationally efficient. Such a heuristic approach has been used by Bayon de Noyer [64] to locate effective sensor locations for feedback control to alleviate tail buffeting of a high performance twin tail craft. However, the formulation based on condition number minimization of  $\mathbf{M}$  (see eqn. (26)) is considered more reliable. The key steps of this MCN algorithm is listed below for completeness.

- (i) Consider all possible placement points, evaluate  $\mathbf{M}$  for each point, choose the point that minimizes condition number of  $\mathbf{M}$  i.e.  $\kappa(\mathbf{M})$ .
- (ii) With the previous sensor locations set, loop over all possible remaining placement points. For each point, update the mask vector, evaluate  $\mathbf{M}$ , and choose the point that minimizes  $\kappa(\mathbf{M})$ .
- (iii) In same way as (ii) find all remaining sensor locations.

A slightly more efficient version of this algorithm is presented by Willcox [60] where the first sensor point is chosen as the location that maximizes the sum of the difference between the diagonal and off-diagonal entries of  $\mathbf{M}$ . The rest of the algorithm is similar to that presented above.

#### 2.4.3. QR Factorization with Column Pivoting

The reduced matrix QR factorization [31] decomposes any given real matrix  $\mathbf{A} \in \mathbb{R}^{S \times T}$  with full column rank into a unitary matrix  $\mathbf{Q} \in \mathbb{R}^{S \times T}$  and an upper triangular matrix  $\mathbf{R} \in \mathbb{R}^{T \times T}$ . Therefore, it follows that  $|\det(\mathbf{A})| = |\det(\mathbf{Q}) \cdot \det(\mathbf{R})| = |\det(\mathbf{R})| = \left| \prod_i r_{ii} \right| = \left| \prod_i \lambda_i \right|$  where  $r_{ii}$  are the diagonal entries of  $\mathbf{R}$  and  $\lambda_i$  are the eigenvalues. It is easy to show that minimizing the condition number of  $\mathbf{A}$  is related to optimizing the spectral characteristics of the matrix such as the determinant or spectral radius, i.e., maximize  $\left| \prod_i r_{ii} \right|$ . In general, the  $\mathbf{R}$  from a reduced matrix QR factorization has diagonal values,  $r_{ii}$  in no particular sequence. However, when combined with column pivoting, we have  $\mathbf{A}\mathbf{D} = \mathbf{Q}\mathbf{R}$ , where  $\mathbf{D} \in \mathbb{R}^{T \times T}$  is a square column permutation matrix containing ones and zeros.

The resulting QR decomposition outcome can be controlled through the pivoting procedure such that the diagonal values of  $\mathbf{R}$ ,  $r_{ii}$  form a decreasing sequence. Therefore, pivoting provides a smart approach for ‘submatrix volume maximization’ and in turn maximize the absolute value of the determinant [32] by reordering the columns of  $\mathbf{A}$ . This approach can easily be extended for sensor placement by leveraging the connections between the permutation matrix  $\mathbf{D}$  and the point sensor measurement matrix  $\mathbf{C}$  in fig. 2 and eqn. (4).

For the case with  $P = K$ , the reconstruction problem in eqn. (4) requires inversion of the square matrix  $\Theta = \mathbf{C}\Phi_k$ . For improved reconstruction, the determinant of  $\Theta$  needs to be maximized through sensor placement which in turn is expected to reduce (and maybe minimize) the condition number. One can see that for a square matrix the following relationship is true.

$$\det(\Theta) = \det(\Theta^T) = \det(\Phi_k^T \mathbf{C}^T). \quad (14)$$

Therefore, reduced matrix QR factorization of  $\Phi^T \in \mathbb{R}^{K \times N}$  with column pivoting will yield

$$\Phi_k^T \mathbf{D} = \mathbf{Q} \mathbf{R} \quad (15)$$

where  $\mathbf{D} \in \mathbb{R}^{N \times N}$  is a square permutation matrix. The right hand side of eqn. (14) will be maximized for a given sensor quantity  $P$  if  $\mathbf{C}$  is chosen as the first  $P$  rows of  $\mathbf{D}^T$ . The index locations of ones in each row of  $\mathbf{C}$  are denoted by  $[\varrho_1, \varrho_2, \varrho_3, \dots, \varrho_P]$ .

For the case of oversampled case with  $P > K$ ,  $\Theta$  is a tall and slender matrix whose inversion (section 2.1.1) is typically handled using a least squares minimization. The left (Moore-Penrose) pseudoinverse requires computing the inverse of the square matrix.  $\mathbf{M} = \Theta^T \Theta \in \mathbb{R}^{K \times K}$ . Therefore, the sensor placement that increases the probability of accurate reconstruction should maximize  $\det(\Theta^T \Theta)$  so that condition number of  $\mathbf{M}$  is bounded. Specifically, we will have the following:

$$\det(\mathbf{M}) = \det(\Theta^T \Theta) = \prod_i \sigma_i(\Theta^T \Theta) = \prod_{i=1}^K \sigma_i(\Theta \Theta^T) = \prod_{i=1}^K \sigma_i(\mathbf{C} \Phi_K \Phi_K^T \mathbf{C}^T). \quad (16)$$

Leveraging the relationships in eqn. (16), we see that maximizing the determinant of  $\mathbf{M} = \Theta^T \Theta$  can be realized by maximizing  $\det(\Phi_K \Phi_K^T \mathbf{C}^T)$  by using a reduced matrix QR factorization as shown below

$$(\Phi_k \Phi_k^T) \mathbf{D} = \mathbf{Q} \mathbf{R}, \quad (17)$$

and once again choosing  $\mathbf{C}$  as the first  $P$  rows of the  $N \times N$  square matrix given by  $\mathbf{D}^T$ . The index locations of ones in each row of  $\mathbf{C}$  are denoted by  $[\varrho_1, \varrho_2, \varrho_3, \dots, \varrho_P]$ .

The algorithm of greedy sensor selection for an oversampled case using a given tailored basis  $\Phi_K$  and number of sensors  $P$  is summarized below:

#### 2.4.4. Discrete Empirical Interpolation Method (DEIM)

All the above smart sensor placement methods focus on minimizing the condition number directly or indirectly controlling it through the determinant of the matrix to the inverted. The discrete empirical interpolation method or DEIM is a discrete variant of the Empirical Interpolation Method (EIM) originally presented by Barrault et al. [65] and subsequently extended to nonlinear model order

**Algorithm 1:** Greedy Sensor Selection using QR Factorization with Column Pivoting

---

**input** : Data-driven basis,  $\Phi_K$   
 Number of sensors,  $P$   
**output:** Measurement Matrix  $C$

1 **if** ( $P = K$ ) **then**  
 2   |  $[\varrho_1, \varrho_2, \dots, \varrho_P] \leftarrow$  Reduced Matrix QR Column Pivoting of  $\Phi_k^T$  ;  
 3 **else if** ( $P > K$ ) **then**  
 4   |  $[\varrho_1, \varrho_2, \dots, \varrho_P] \leftarrow$  Reduced Matrix QR Column Pivoting of  $\Phi_k \Phi_k^T$  ;  
 5  $C \leftarrow [e_{\varrho_1}, e_{\varrho_2}, \dots, e_{\varrho_P}]^T$  where  $e_{\varrho_i} = [0, \dots, 0, \underbrace{1}_{\varrho_i}, 0, \dots, 0]^T$

---

reduction applications by Chaturantabut and Sorensen [18,59]. Here, one recursively learns the interpolation points (with indices  $\varrho_j$ ) at locations carrying maximum linear dependence error estimated at previously estimated interpolation points.

The primary idea behind DEIM is to estimate a high-dimensional state using information at sparsely sampled interpolation points. Such techniques (other examples being Gappy POD [17] and missing point estimation or MPE [20]) are popular as hyper-reduction tools that bypass expensive nonlinear term computations in model order reduction. Naturally, one can adopt these interpolation points for sensor placement in sparse reconstruction applications. To illustrate this, the POD-based DEIM approximation of order  $M$  (number of interpolation points) for  $\mathbf{u}(t)$  in the space spanned by the basis  $\Psi \in \mathbb{R}^{N \times M}$  is given by

$$\mathbf{u} = \Psi b(t) \quad (18)$$

where  $b(t)$  is coefficient vector,  $b(t) \in \mathbb{R}^M$ . When using POD bases,  $\Psi$ , one can simply estimate  $b(t) = \Psi^T \mathbf{u}(t)$ , but this requires dealing with the higher dimensional state vectors  $\in \mathbb{R}^N$  that are computationally comparable to high-fidelity models even when using projection-based reduced order models. Hyper-reduction strategies [19] bypass this issue by estimating the approximate coefficients  $b(t)$  using carefully chosen set of  $M$  interpolation points instead of the full ( $N$ -dimensional) state so that computational cost scales with  $M$ . Specifically, one chooses  $M$  interpolation points corresponding to indices  $[\varrho_1, \dots, \varrho_M]$ ,  $\varrho_i \in \mathbb{N}$  to form a  $M$ -by- $M$  linear system  $\mathbf{D}^T \Psi b(t) = \mathbf{D}^T \mathbf{u}(t)$ , where the interpolation or measurement matrix is given by  $\mathbf{D} = [e_{\varrho_1}, \dots, e_{\varrho_M}] \in \mathbb{R}^{N \times M}$  with columns  $e_{\varrho_i} = [0, \dots, 0, \underbrace{1}_{\varrho_i}, 0, \dots, 0]^T \in \mathbb{R}^N$ . The DEIM approximation of  $\mathbf{u}(t)$  then becomes

$$\mathbf{u}_{DEIM}(t) = \underbrace{\Psi(\mathbf{D}^T \Psi)^{-1}}_{N \times M} \underbrace{\mathbf{D}^T \mathbf{u}(t)}_{M \times 1}, \quad (19)$$

where  $\Psi(\mathbf{D}^T \Psi)^{-1}$  can be precomputed once to yield a  $N \times M$  matrix while  $\mathbf{D}^T \mathbf{u}(t)$  represents  $M$ -dimensional representation of the state denoting the interpolation points. This way, one avoids repeated computation of the high-dimensional  $\mathbf{u}(t)$ . One can easily see the connections between DEIM approximation and the sparse reconstructed state  $\mathbf{x}_{SR} = \Phi a$  with  $a$  obtained using eqns. (5)-(8) using  $C = \mathbf{D}^T$ . Therefore, estimating the interpolation points ( $\mathbf{D}^T$ ) is similar to estimating the sparse measurement locations in  $C$ . The indices  $\varrho_1, \dots, \varrho_M$  are estimated recursively from the input basis  $\{\Phi_j\}_{j=1}^M$  using the algorithm 2 from [18]. The process starts from selecting the first interpolation

**Algorithm 2:** Discrete Empirical Interpolation Method (DEIM)

---

**input :**  $\{\Phi_j\}_{j=1}^M \subset \mathbb{R}^N$  linearly independent  
**output:**  $\vec{\varrho} = [\varrho_1, \dots, \varrho_M]^T \in \mathbb{R}^M$

1  $[\|\rho\|, \varrho_1] = \max\{|\Phi_1|\}$   
2  $\Phi = [\Phi_1], \mathbf{D} = [e_{\varrho_1}], \vec{\varrho} = [\varrho_1]$   
3 **for**  $j = 2$  to  $M$  **do**

4     Solve  $(\mathbf{D}^T \Phi) a = \mathbf{D}^T \Phi_j$  for  $a$   
5      $\mathbf{r} = \Phi_j - \Phi a$   
6      $[\|\rho\|, \varrho_j] = \max\{|\mathbf{r}|\}$   
7      $\Phi \leftarrow [\Phi \Phi_j], \mathbf{D} \leftarrow [\mathbf{D} e_{\varrho_j}], \vec{\varrho} \leftarrow \begin{bmatrix} \vec{\varrho} \\ \varrho_j \end{bmatrix}$

8 **end**

---

index  $\varrho_1 \in \{1, 2, \dots, N\}$  using the first input POD basis  $|\Phi_1|$ . The remaining interpolation indices  $\{\varrho_j, j = 2, 3, \dots, M\}$  are selected such that they correspond to the largest magnitude of  $|\mathbf{r}|$  where  $\mathbf{r} = \Phi_j - \Phi a$  is the residual error between current input basis and its interpolation  $\Phi a$  obtained using  $\{\Phi_1, \Phi_2, \Phi_3 \dots \Phi_{j-1}\}$  at the indices  $\{\varrho_1, \varrho_2, \varrho_3 \dots \varrho_{j-1}\}$ . This step is given by line 5 of the algorithm 2. In lines 1 & 6, the ‘max’ function is similar to that available in MATLAB and  $|\rho| = |r_{\varrho_j}|$ . The residual represents a measure of the linear independence of  $\Phi_j$  with respect to the basis vectors ahead of this in the sequence and the interpolation point is at the maximum absolute value of this measure. Naturally, the realized  $\varrho_j$  depends on the choice of basis  $\Phi_j$  and their sequence. The ordering of the input basis is not critical for the QR-pivoting based approach. The linear independence of the input basis ensures the above procedure is well-defined, i.e.  $\mathbf{D}^T \Phi$  is non singular and  $\rho \neq 0$  for all iterations. By using POD basis as the input, the linear independence and hierarchy (i.e. basis ordered in terms of decreasing singular values) characteristics are guaranteed which in turn ensures that the sparse interpolation indices are hierarchical and non repeating.

## 2.5. Sparse Recovery (SR) Framework

The reconstruction algorithm used in this work based on the Gappy POD or GPOD framework [17] and can be viewed as an  $l_2$  minimization solution of the sparse recovery problem summarized through eqns. (4),(5) and (6) with  $\Phi$  composed of  $K \leq M$  basis vectors, i.e. dimension of  $a$  is  $K \leq M$ . At this point, we remind the reader of naming conventions adopted in this paper: the instantaneous  $j^{th}$  full flow state is denoted by  $x_j \in \mathbb{R}^N$ , whereas the entire set consisting of  $M$  snapshots is assembled into a matrix form denoted by  $X \in \mathbb{R}^{N \times M}$ . This discussion focuses on single snapshot reconstruction as the extension to multiple snapshots is trivial, i.e. each snapshot can be reconstructed sequentially or in some cases be grouped together as a batch. This allows for such algorithms to be parallelized easily.

The primary difference between the SR framework in eqn. (4) as used in compressive sensing or DEIM-based approaches and GPOD [12,13,17,66] as shown in eqn. (21) is the construction of the measurement matrix  $\mathbf{C}$  and the sparse measurement vector  $\tilde{x}_j$ . In SR (eqn. (4)), the down sampled state  $\tilde{x}_j \in \mathbb{R}^P$  is a *compressed* version containing only the measured data, whereas in GPOD,  $\tilde{x}_j \in \mathbb{R}^N$

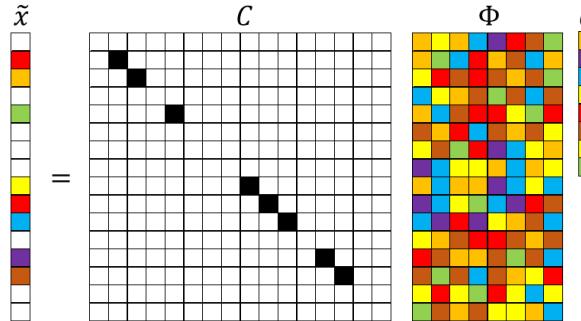
is a *masked* version of the full state vector, i.e. values outside of the  $P$  measured locations are zeroed out to generate a filtered version of  $x_j$ . For high resolution data  $x_j \in \mathbb{R}^N$  with chosen basis  $\Phi_j \in \mathbb{R}^N$ , the low-dimensional features,  $a^j \in \mathbb{R}^K$  are obtained from the relationship shown below:

$$x_j = \sum_{i=1}^K \Phi_i a_i^j. \quad (20)$$

The masked (incomplete) data  $\tilde{x}_j \in \mathbb{R}^N$ , corresponding measurement matrix  $\mathbf{C}$  and mask vector  $m \in \mathbb{R}^N$  are related by:

$$\tilde{x}_j = \langle m \cdot x_j \rangle = \mathbf{C}x_j, \quad (21)$$

where  $\mathbf{C} \in \mathbb{R}^{N \times N}$ . Therefore, the GPOD algorithm results in a larger measurement matrix with numerous rows full of zeros as shown in fig. 3 (compare with fig. 1). To bypass the complexity of handling this  $N \times N$  matrix, a mask vector,  $m \in N \times 1$  with 1s and 0s operates on  $x_j$  through a point-wise multiplication operator  $\langle \cdot \rangle$ . To illustrate, the point-wise multiplication is represented as  $\tilde{x}_j = \langle m_j \cdot x_j \rangle$  for each snapshot  $j = 1..M$  where each element of  $x_j$  multiplies with the corresponding element of  $m_j$ . This is applicable when each data snapshot,  $x_j$  has its own measurement mask  $m_j$  which is a useful way to represent the evolution of sparse sensor locations over time. The SR formulation in eq. (4) can also support time varying sensor placement, but would require a compression matrix,  $\mathbf{C}_i \in \mathbb{R}^{P \times N}$  that changes with each snapshot. The goal of the SR procedure is to recover the full



**Figure 3.** Schematic of GPOD formulation for sparse recovery. The numerical values represented by the colored blocks: black (1), white (0), color (other numbers).

data from the masked data given in eqn. (22) by approximating the coefficients  $\bar{a}^j$  (in the  $l_2$  sense) with basis,  $\Phi_K$ , learned offline using training data (snapshots of the full field data).

$$\tilde{x}_j \approx m_j \sum_{i=1}^K \bar{a}_i^j \Phi_i. \quad (22)$$

The coefficient vector  $\bar{a}^j$  cannot be computed by direct projection of  $\tilde{x}_j$  onto  $\Phi$  as these are not designed to optimally represent the sparse data. Instead, one needs to obtain the “best” approximation of the coefficient  $\bar{a}^j$ , by minimizing the error  $\mathcal{E}_j$  in the  $l_2$  sense as show in eqn. (23).

$$\mathcal{E}_j = \left\| \tilde{x}_j - m_j \sum_{i=1}^K \bar{a}_i^j \Phi_i \right\|_2^2 = \left\| \tilde{x}_j - m_j \cdot \Phi \bar{a}^j \right\|_2^2 = \left\| \tilde{x}_j - \mathbf{C} \Phi \bar{a}^j \right\|_2^2. \quad (23)$$

In eqn. (23) we see that  $m_j$  acts on each column of  $\Phi$  through a point-wise multiplication operation which is equivalent to masking each basis vector  $\Phi_i$ . The above formulation is valid for a single snapshot reconstruction where the mask vector,  $m_j$  could change with every snapshot  $\tilde{x}_j$  for  $j = 1..M$  and the error  $\mathcal{E}_j$  represents the single snapshot reconstruction error to be minimized in order to estimate the approximate coefficients  $\bar{a}^j$ . It can easily be seen from below that one will have to minimize the different  $\mathcal{E}_j$ 's sequentially to learn the entire coefficient matrix,  $\bar{a} \in \mathcal{R}^{K \times M}$  for all the  $M$  snapshots. Denoting the masked basis functions as  $\tilde{\Phi}_i = \langle m_j \cdot \Phi_i \rangle$ , eq. (23) is rewritten as in eqn. (24).

$$\mathcal{E}_j = \left\| \tilde{x}_j - \sum_{i=1}^K \bar{a}_i^j \tilde{\Phi}_i \right\|_2^2. \quad (24)$$

In the above,  $\tilde{\Phi}$  is analogous to  $\Theta = C\Phi$  in eq. (4). To minimize  $\mathcal{E}_j$ , one computes the derivative with respect to  $\bar{a}_j$  and equated to zero as below:

$$\frac{\partial}{\partial \bar{a}^j} (\mathcal{E}_j) = 0. \quad (25)$$

The result is the linear normal equation given by eqn. (26)

$$\mathbf{M}\bar{a}^j = f^j, \quad (26)$$

where  $\mathbf{M}_{k1,k2} = \langle \tilde{\Phi}_{k1}, \tilde{\Phi}_{k2} \rangle$  or  $\mathbf{M} = \tilde{\Phi}^T \tilde{\Phi}$  and  $f_i^j = \langle \tilde{x}_j, \tilde{\Phi}_i \rangle$  or  $f^j = \tilde{\Phi}^T \tilde{x}_j$ . The reconstructed solution is given by eqn. (27) below:

$$\bar{x}_j = \sum_{k=1}^K \phi_k \bar{a}_k^j. \quad (27)$$

Algorithm 3 summarizes the above SR framework assuming the basis functions ( $\Phi_i$ ) are known. The above solution procedure for sparse recovery is the same as that described in section 2.1 except for the dimensions of the  $\tilde{x}$  and  $C$ .

---

**Algorithm 3: Least Squares ( $l_2$ ) Sparse Reconstruction with basis  $\Phi$ .**


---

**input :** Full data ensemble  $X \in \mathbb{R}^{N \times M}$   
           Incomplete data vector  $\tilde{X} \in \mathbb{R}^{N \times M}$   
           the mask vector  $m \in \mathbb{R}^N$ .

**output:** Approximated full data vector  $\bar{X} \in \mathbb{R}^{N \times M}$

**1 for** each snapshot index  $j \leq M$  **do**

2    Build a least squares problem:  $\mathbf{M}\bar{a}^j = f^j$  (eqn. (26)) as below;

3      Compute masked basis function:  $\tilde{\Phi} = m\Phi$ ;

4      Compute matrix  $\mathbf{M} = \tilde{\Phi}^T \tilde{\Phi}$ ;

5      Compute vector:  $f^j = \tilde{\Phi}^T \tilde{x}_j$ ;

6      Solve  $\bar{a}^j$  from the least squares problem:  $\mathbf{M}\bar{a}^j = f^j$  (eqn. (26));

7      Reconstruct the approximated solution  $\bar{x}_j = \Phi \bar{a}^j$  (eqn. (27)).

**8 end**

---

## 2.6. Algorithmic Complexity

The cost of computing the POD basis is  $O(N \times M^2)$  where  $N$  is the full state dimension and  $M$ , the number of snapshots. The cost of sparse reconstruction turns out to be  $O(N \times K \times M)$  for both the methods, where the  $K \leq M$  is the system dimension chosen for reconstruction. Naturally, for many practical problems with an underlying low-dimensional structure, POD is expected to result in a smaller  $K$  than any other choice of basis. This helps reduce the sensor quantity requirement and reconstruction cost. Further, since the number of snapshots ( $M$ ) is also tied to the desired basis dimension ( $K$ ), a larger  $K$  requires a larger  $M$  and in turn a higher cost to generate the basis.

The complexity of the sensor placement depends on the choice of method adopted. Using QR factorization with column pivoting, the cost of sensor placement for a  $N \times N$  matrix is  $O(N^3)$  and  $O(NM^2)$  for a  $N \times M$  matrix. The DEIM method carries a complexity of  $O(NM^3)$  when retaining  $M$  POD modes and identifying  $M$  sensors with a state dimension of  $N$ . The MCN algorithm requires an expensive combinatorial search to find the sensor location that minimizes the condition number of  $M \in \mathbb{R}^{M \times M}$  in each possible sweep. This results in a computational cost of  $O(N^2M^3)$  for identifying  $M$  sensors. These estimates are consistent with our observation that the DEIM and QR-pivoting approaches yield results in very short time.

## 3. Data Generation for Numerical Experiments

### 3.1. Low-dimensional Cylinder Wake Flows

Studies of cylinder wakes [24,67–69] have attracted considerable interest from the model reduction and dynamical systems communities for its particularly rich flow physics content, encompassing many of the complexities of nonlinear flow systems, while easy to simulate accurately on the computer using established CFD tools. In this study, we explore data-driven sparse reconstruction for the cylinder wake flow at Reynolds numbers with unsteady wakes in the range of  $Re = 100 - 1000$ . To generate two-dimensional cylinder flow data, we adopt the spectral Galerkin method [70] to solve incompressible Naiver-Stokes equations, as shown in Eq. (28) below:

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \quad (28a)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} + \nu \nabla^2 u, \quad (28b)$$

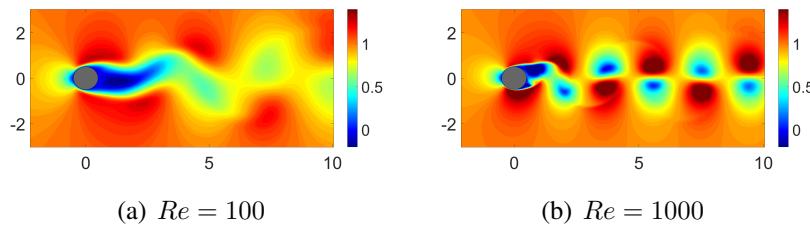
$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} + \nu \nabla^2 v, \quad (28c)$$

where  $u$  and  $v$  are horizontal and vertical velocity components.  $P$  is the pressure field, and  $\nu$  is the fluid viscosity. The rectangular domain used for this flow problem is  $-25D < x < 45D$  and  $-20D < y < 20D$ , where  $D$  is the diameter of the cylinder. For the purposes of this study, data from a reduced domain, i.e.,  $-2D < x < 10D$  and  $-3D < y < 3D$ , is used. The mesh was designed to sufficiently resolve the thin shear layers near the surface of the cylinder and transit wake physics downstream. For the case of  $Re = 100$  the grid includes 24,000 points whereas for  $Re = 1000$  the grid is refined to

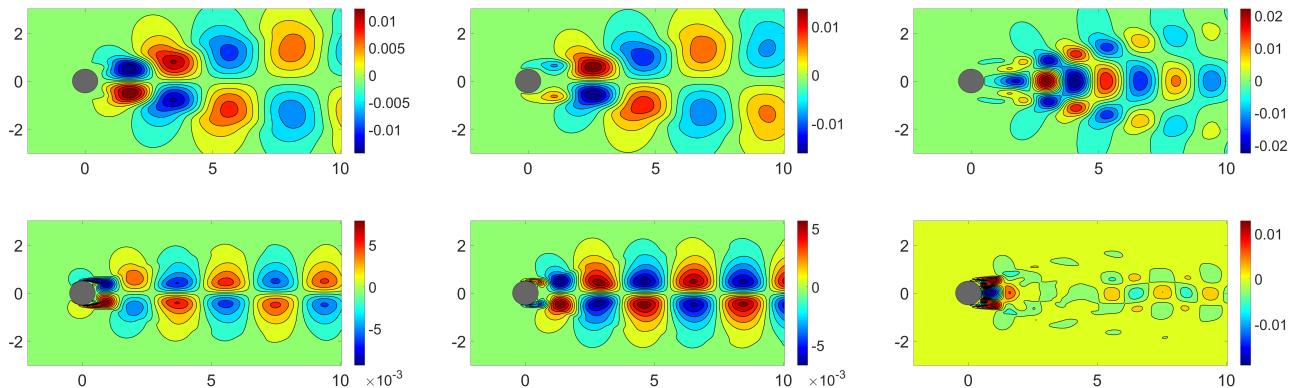
include approximately 95,000 points for the sampled flow region. The computational method employed is a fourth order spectral expansion within each element in each direction. The sampling rate for each snapshot output is chosen as  $\Delta t = 0.2$  seconds.

### 3.1.1. Cylinder Wake Limit-cycle Dynamics

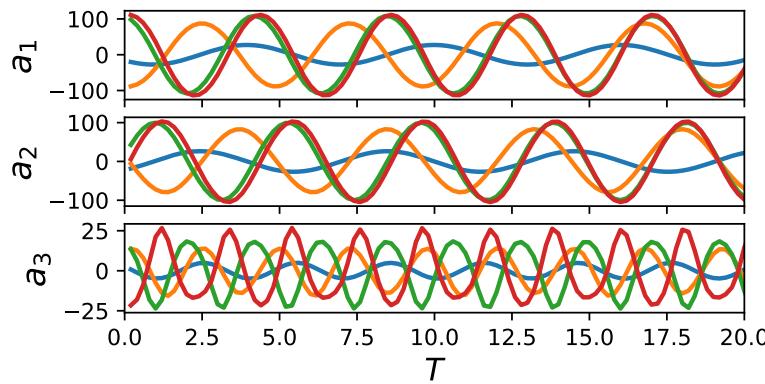
In this section, we explore sparse reconstruction of unsteady wake flows with well-developed periodic vortex shedding behavior. The GPOD type algorithm is chosen over the traditional Compressive sensing-based SR formulations to bypass the need for maintaining a separate measurement matrix, especially when employing point sensors to mimic realistic data acquisition. The first three POD modes and coefficients are computed and shown in figure 5 and 6, respectively. Similar to the velocity field visualized in figure 4, the dominant POD modes (mode 1 and mode 2) capture the symmetric vortex shedding patterns at various length scales for all the cases. The vestiges of onset of instability at higher  $Re$  numbers ( $Re = 1000$ ) is observed from the asymmetry in mode 3. This is consistent with the observations in [71], where the laminar vortex shedding happens at around  $Re = 47$ , and becomes unstable at  $\sim 190$  which deforms the small-scale structures at higher POD modes. On the other hand, the temporal evolution of POD coefficients show periodic limit-cycle behavior for all the  $Re$  numbers with the characteristic frequencies increasing with  $Re$  as shown in fig. 6. The dependence of sparse reconstruction performance on  $Re$  is beyond the scope this article although we expect the system dimensionality to increase with  $Re$  as the shear layers become thinner and consequently, require increased spatial sampling to capture all the relevant scales of motion. Instead, in this work we operate with narrower focus, i.e., explore and quantify how sparse reconstruction performance (in terms of error metrics) depend on sensor quantity, their placement and model dimensionality. With this in mind, we focus the analysis in the rest of the paper to a single Reynolds number,  $Re = 100$ . In this study, we choose 300 snapshots of data corresponding to a Reynolds number,  $Re = 100$ , corresponding to a non-dimensional time ( $T = \frac{U_t}{D}$ ) of  $T = 60$  with uniform temporal spacing of  $dT = 0.2s$ .  $T = 60$  corresponds to multiple ( $\approx 10 - 15$ ) cycles of periodic vortex shedding behavior.



**Figure 4.** Color contour snapshots of the instantaneous stream-wise velocity component for the cylinder flow at  $Re = 100$  and  $Re = 1000$ .



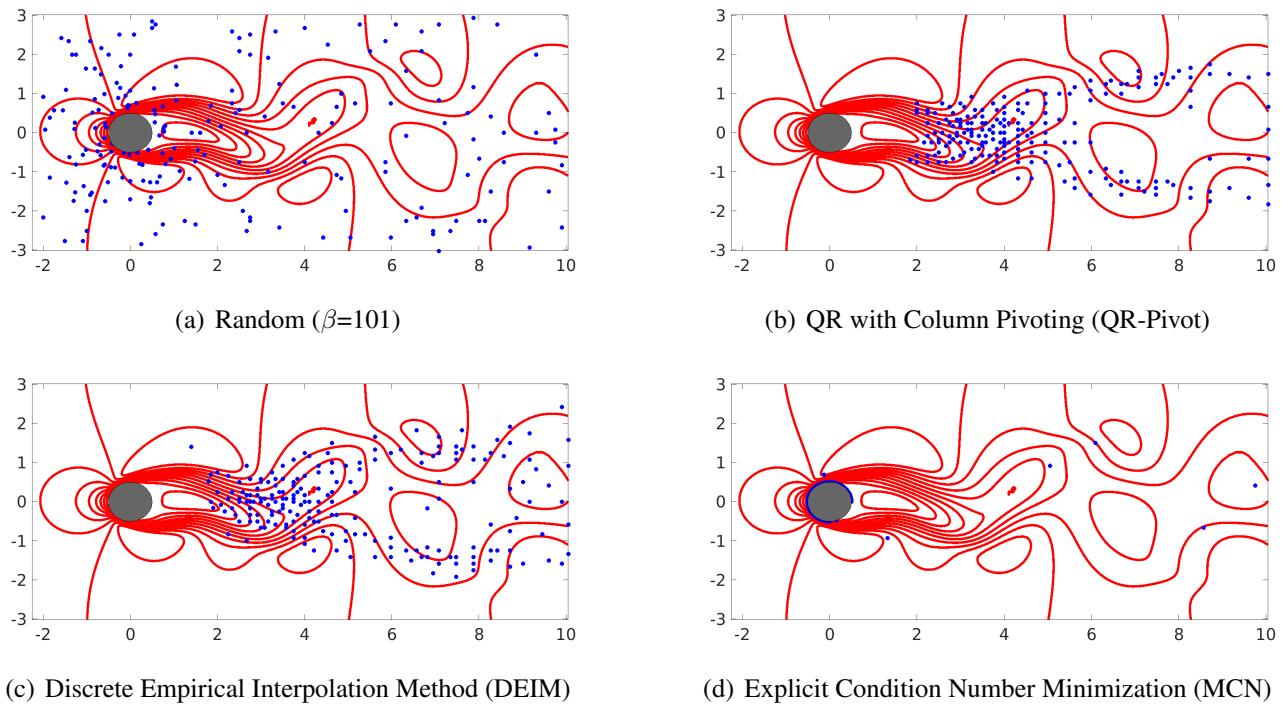
**Figure 5.** Isocontours of the three most energetic modes (from left to right) for the cylinder flow at  $Re = 100$  and  $Re = 1000$ .



**Figure 6.** Time evolution of the first three POD coefficients (Top:  $a_1$ ; Middle:  $a_2$ ; Bottom:  $a_3$ ) for the cylinder flow at different  $Re$  number ( $Re = 100$  (Blue), 300 (Orange), 800 (Green), 1000 (Red)) in the periodic vortex shedding regime.

### 3.1.2. Smart Sensor Placement in Cylinder Wake

Figure 7 summarizes the realized sensor placement with a budget of 200 sensors using four different algorithms: (i) random algorithm in sec. 2.4.1; (ii) QR factorization with column pivoting that maximizes the determinant capture for a given  $P$  as discussed in sec. 2.4.3; (iii) discrete empirical interpolation method (DEIM) in sec. 2.4.4 and (iv) explicit condition number minimization through an iterative combinatorial search (sec. 2.4.2). All these sensor placement algorithms use POD bases computed using the 300 snapshots of data containing both  $u$  and  $v$  velocity components. As expected, the random sensor placement in fig. 7(a) samples the entire domain without bias although the regions with higher grid density tend to be sampled more frequently. This includes the region close to the cylinder surface where the mesh resolution is high to capture the thin shear layers accurately. On the other hand, the three ‘smart’ sensor placement algorithms and shown fig. 7(b)-7(d) identify locations either in the wake and also on the cylinder surface. The QR-Pivot and the DEIM methods identify a cluster of centers about two diameters downstream of the cylinder. Further downstream, the computed centers split into two narrow ‘tail’-like regions on either side of the wake centerline. On the other hand, the MCN framework (fig. 7(d)) identifies only five sensors in the wake, mostly downstream while a overwhelming number are



**Figure 7.** Sensor locations learned from different choices of methods considered in this work including both random as well as smart algorithms for the cylinder wake flow ( $Re = 100$ ). These images show the most relevant 200 sensors, i.e. ( $P = 200$ ).

predicted on the cylinder surface. This is potentially indicative of the low-dimensional characteristic of the wake flow where only a few POD modes are needed to capture a significant portion of the energy.

#### 4. Sparse Reconstruction of Canonical Fluid Flows

In this section, we explore sparse reconstruction of cylinder wake flow at  $Re = 100$  using the above SR infrastructure based on the GPOD formulation as against the traditional SR formulation. This choice is purely a matter of convenience and helps bypass the need for maintaining a separate measurement matrix (resulting in memory savings). In all the oversampled cases reported in this section, Tikhonov regularization is employed to reduce overfitting.

##### 4.1. Sparse Reconstruction (SR) Experiments and Analysis

For this *a priori* assessment of SR performance we reconstruct sparse data from simulations where the full field representation is available. The sparse sensor locations are chosen as single point measurements using a variety of sensor placement methods as discussed in section 3.1.2 and these locations are fixed for the ensemble of snapshots used for the reconstruction (i.e. we do not consider dynamic sensor placement). Reconstruction performance is evaluated by comparing the original simulated field with that from SR using POD as the choice of basis across the entire ensemble of numerical experiments. Of course, using such a data-driven basis requires availability of data so that one can compute the basis vectors *a priori*. In practice, one would have to build a library of basis vectors from data that can in turn be used for sparse recovery and is currently being developed within our team. In this we undertake

this approach in order to focus on assessing the relative roles of system dimensionality ( $K$ ), sensor sparsity ( $P$ ) and sensor placement ( $C$ ) for the POD-based SR. In particular, we aim to accomplish the following through this study: (i) check if oversampling relative to desired system dimension ( $P > K$ ) is sufficient condition for accurate reconstruction of the fluid flow for this POD-based  $l_2$  reconstruction; (ii) understand how sensor placement impacts reconstruction quality for different choice of basis.

To learn the data-driven POD basis we employ the method of snapshots [52] as shown in eqns. (9)-(12). For the numerical experiments described here, the data-driven basis and coefficients are obtained from the full data ensemble, i.e.,  $M = 300$  snapshots corresponding to  $T = 60$  non-dimensional times. This gives rise to at most  $M$  basis for use in the reconstruction process in eqn. (3), i.e. a candidate basis dimension of  $N_b = M$ . As shown in table 1, the choice of algorithms depend on the choice of system dimensionality ( $K$ ), data sparsity ( $P$ ) and dimension of the candidate basis space,  $N_b$ . Recalling from the earlier discussion in section 2, we see that  $P \geq K$  would require an  $l_2$  method for a desired reconstruction sparsity  $K$  as long as  $P \geq N_b$ . In case of POD-based SR, the basis are energy optimal for the training data and hence, contain *built-in sparsity*. That is, as long as the basis is relevant for the flow to be reconstructed, retaining only the most energetic modes (basis) should generate the best possible reconstruction for the given sensor placement and quantity. Therefore, the POD basis need to be generated once and the reconstructed system dimension is determined by just choosing to retain the first few modes in the sequence. For generic basis with no built-in ordering of when using a library of basis with no clear classification available *a priori*, one requires to search for the  $K$  most significant basis amongst the maximum possible dimension of  $N_b = M$  using sparsity promoting  $l_1$  methods.  $l_1$  methods such as the iterative thresholding algorithms [27,72] require multiple solutions of the least-squares problem before a sparse solution is realized. This results in necessary, but increased computational cost.

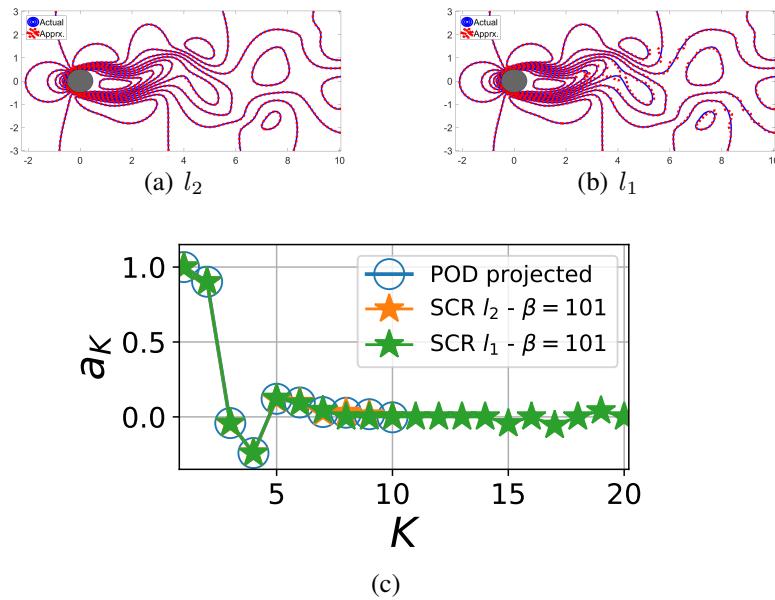
#### 4.2. Comparison of $l_2$ and $l_1$ Sparse Reconstruction using Energy-ordered POD Basis

In this subsection, we verify the equivalence between  $l_2$  reconstruction using energy ordered POD basis and sparsity promoting  $l_1$  – minimization reconstruction for the cylinder wake data when presented the same sparse sensor data. The success of this verification will imply that the chosen POD basis ordered in terms of energy of the training data is also relevant to the data being reconstructed and also validate the accuracy of our  $l_1$  implementation. In such cases, choosing the  $K$ -most relevant basis for reconstruction is trivial (and efficient) as one chooses the first or last  $K$  modes (as against searching for the sparsest  $K$  modes using  $l_1$ ) as the candidate basis for  $l_2$  reconstruction provided sufficient sparse data,  $P > K$ , is available. To verify this, we consider two cases of reconstruction for the limit-cycle cylinder wake at  $Re = 100$ . Case (a):  $P > K$ ,  $K = N_b$  and using least squares reconstruction; Case (b):  $P > K$ ,  $K < N_b \leq M$  and employs  $l_1$  reconstruction using optimal matching pursuit. Figure 8 and table 2 compares the reconstructed fields and the  $K$  predicted weights for the two cases for a single snapshot corresponding to  $T = 0.2$ . In this study we use a greedy optimal matching pursuit algorithm for the  $l_1$  norm minimization, CoSAMP as described by Needell and Tropp in [11].

The above results confirm the equivalence of these two approaches for POD-based reconstruction of cylinder wake flows considered in this study. It is worth noting that  $l_1$  should be the preferred choice

**Table 2.** caption

	$Re$	$K$	$P$	$N_b$	$\beta$
$l_2$	100	10	40	10	101
$l_1$	100	10	40	20	101

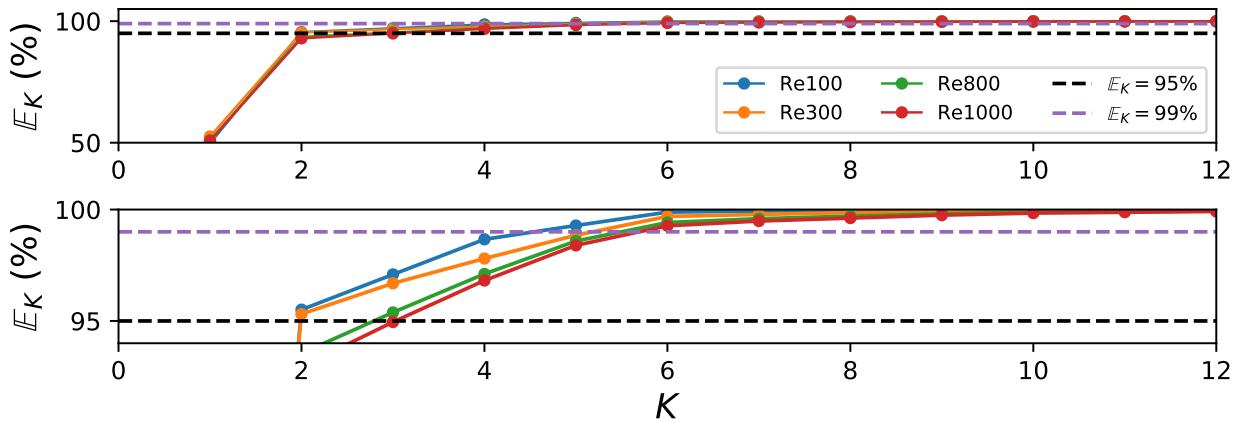


**Figure 8.** Comparison of the sparse reconstruction using both  $l_1$  and  $l_2$  minimization methods for basis that is ordered in terms of energy content. The reconstructed and actual flowfields at  $T = 0.2$  are compared in (a) for  $l_2$  and (b)  $l_1$ . The corresponding POD features from both methods are shown in (c).

when the relevance of sparse data to the candidate basis space is unclear. For the rest of this paper, we will leverage energy hierarchy of the candidate POD basis and adopt the SR problem formulation using a  $l_2$  minimization algorithm with  $K = N_b$ . In other words, the candidate basis space is chosen based on the desired reconstructed field dimension and not fixed.

#### 4.3. Sparsity and Energy Metrics

For this SR study, we explore the conditions for accurate recovery of information in terms of data sparsity ( $P$ ) and system sparsity ( $K$ ) which also represents the dimensionality of the system in a given basis space. In other words, sparsity represents the size of a given basis space needed to capture a desired amount of energy. As long as the measurements are incoherent with respect to the basis  $\Phi$  and the system is overdetermined, i.e.,  $P > K$ , one should be able to invert  $\Theta$  to recover the higher dimensional state,  $X$ . From earlier discussions in section 2, we know  $P > K$  is a sufficient condition for accurate reconstruction using  $l_0$  minimization. Thus, both interpretations require a minimum quantity of sensor data for accurate reconstruction and is verified through numerical experiments in section 4.4. In this section, we describe how the different system sparsity metrics,  $K = N_b$ , are chosen for the numerical experiments with both POD and ELM basis. Since the basis spaces are different, a good way to compare system dimensions is through the energy captured by the respective basis representations.



**Figure 9.** Schematic shows the cumulative energy capture corresponding to various system sparsity levels,  $K$ , (i.e. the number of POD modes) for cylinder flow at  $Re = 100$  to  $1000$ . The bottom figure is a magnified version of the top figure for  $\mathbb{E}_K$ . For  $Re = 100, 300, 800, 1000$ :  $K_{95} = [2, 2, 3, 4]$  and  $K_{99} = [5, 6, 6, 6]$ , respectively.

For POD one easily defines a cumulative energy fraction captured by the  $K$  most energetic modes,  $\mathbb{E}_K$ , using the corresponding singular values of the data as shown in eqn. (29).

$$\mathbb{E}_K = \sum_{k=1}^K \frac{\lambda_k}{(\lambda_1 + \lambda_2 + \dots + \lambda_M)} \times 100. \quad (29)$$

In the above, the singular values,  $\lambda$  are computed from eqn. (10), and  $M$  is the total number of possible eigenvalues for  $M$  snapshots. As a result, the energy fraction  $\mathbb{E}_K$  corresponding to sparsity  $K$  for the different flow Reynolds numbers,  $Re$ , is shown in Figure 9. For the cylinder flow case with  $Re = 100$ , one requires two and five POD modes to capture 95% and 99% of the energy content respectively, indicative of the low dimensional dynamics. We denote the mode numbers corresponding to 95% and 99% of the energy capture as  $K_{95}$  and  $K_{99}$  respectively. In this case, we compute  $err_K^{POD} = \|X - \Phi_{1..K}^{POD} a_{1..K}^{POD}\|_2$ , where  $\Phi_{1..K}^{POD}$ , and  $a_{1..K}^{POD}$  represent the matrix comprising of  $K$  POD bases and the corresponding coefficients for the different snapshots respectively. To assess SR performance across different flow regimes (that have different  $K_{95}$ ) with different values of  $K$  we define a normalized system dimension,  $K^* = K/K_{95}$  and a normalized sensor sparsity,  $P^* = P/K_{95}$ . This allows us to design an ensemble of numerical experiments in the discretized  $P^* - K^*$  space and the outcomes can be generalized beyond the current study. In this work, the design space is populated over the range  $1 < K^* < 6$  and  $1 < P^* < 12$  for POD-based SR with  $K$  bounded by the total number of snapshots,  $M = 300$ . The lower bound of one is chosen such that the minimally accurate reconstruction captures 95% of the energy. If one desires a different reconstruction norm, then  $K_{95}$  can be changed to  $K_{xx}$  without loss of generality and the corresponding  $K$ -space modified accordingly. Alternately, one can choose  $\mathbb{E}_K$ , the normalized energy fraction metric to represent the desired energy capture as a fraction of  $\mathbb{E}_{K_{95}}$ , but is not used in this study.

To quantify the  $l_2$  reconstruction performance, we define the mean squared error as shown in eqn. (30) below,

$$\epsilon_{K^*, P^*}^{SR} = \frac{1}{M} \frac{1}{N} \sum_{j=1}^M \sum_{i=1}^N (X_{i,j} - \bar{X}_{i,j}^{SR})^2, \quad (30)$$

where  $X$  is the true data, and  $\bar{X}^{SR}$  is the reconstructed field from sparse measurements as per algorithm 3.  $N$  and  $M$  represent the state and snapshot dimensions and related to indices  $i$  and  $j$ , respectively. Similarly, the mean squared error  $\epsilon_{K_{95}^*}^{FR}$  and  $\epsilon_{K^*}^{FR}$  for the *full reconstruction* from the POD basis can be computed as:

$$\epsilon_{K_{95}^*}^{FR} = \frac{1}{M} \frac{1}{N} \sum_{j=1}^M \sum_{i=1}^N (X_{i,j} - \bar{X}_{i,j}^{FR, K_{95}^*})^2, \quad (31)$$

$$\epsilon_{K^*}^{FR} = \frac{1}{M} \frac{1}{N} \sum_{j=1}^M \sum_{i=1}^N (X_{i,j} - \bar{X}_{i,j}^{FR, K^*})^2, \quad (32)$$

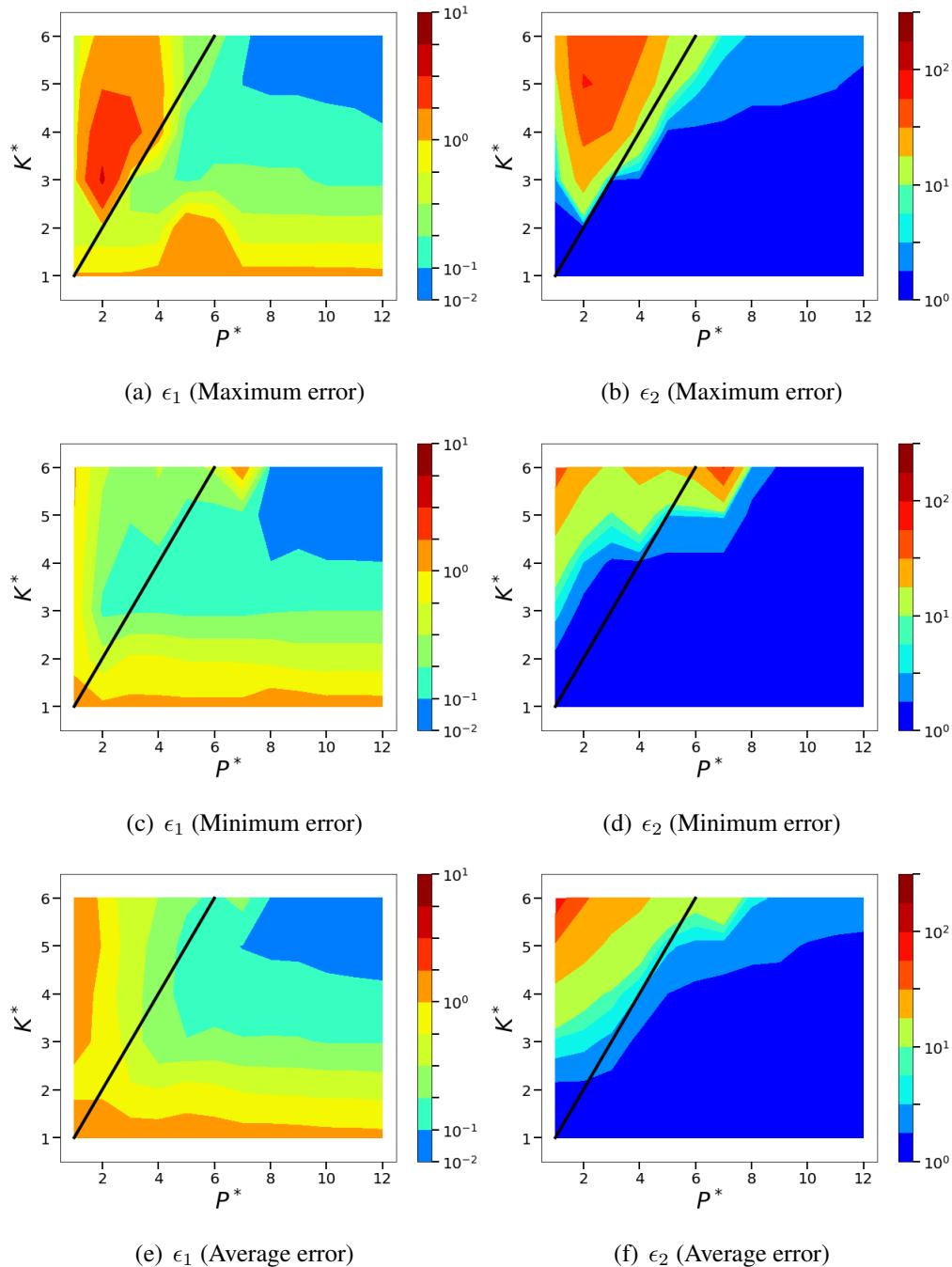
where  $\bar{X}^{FR}$  is the full field reconstruction using *exactly computed POD coefficients a*. For POD this is simply  $a = \Phi^T X$  as per eqn. (12).  $K_{95}^*$  is the normalized system dimension (i.e. number of POD modes normalized by  $K_{95}$ ) corresponding to 95% energy capture and  $K^* = K/K_{95}$  represents the desired reconstructed system dimension. Note that  $K_{95}^* = K_{95}/K_{95} = 1$  is trivially seen to be one for this case.

Using the above definitions, we can now generate normalized versions of the absolute ( $\epsilon_1$ ) and relative ( $\epsilon_2$ ) errors as shown in eqn. (33).  $\epsilon_1$  represents the SR error normalized by the corresponding full reconstruction error for 95% energy capture.  $\epsilon_2$  represents the normalized error relative to the desired reconstruction accuracy for the chosen system reconstruction dimension,  $K$ . These two error metrics are chosen so as to achieve the twin goals of assessing the overall quality of the SR in a normalized sense ( $\epsilon_1$ ) and the best possible reconstruction accuracy for the chosen problem set-up (i.e  $P, K$ ). Thus, if the best possible reconstruction for a given  $K$  is realized then  $\epsilon_2$  will take the same value across different  $K^*$ . This error metric is used to assess relative dependence of  $P^*$  on  $K^*$  for the chosen flow field. On the other hand,  $\epsilon_1$  provides an absolute estimate of the reconstruction accuracy for a given flow system so that minimal values of  $P^*, K^*$  needed to achieve a desired accuracy can be identified.

$$\epsilon_1 = \frac{\epsilon_{K^*, P^*}^{SR}}{\epsilon_{K_{95}^*}^{FR}}, \quad \epsilon_2 = \frac{\epsilon_{K^*, P^*}^{SR}}{\epsilon_{K^*}^{FR}}. \quad (33)$$

#### 4.4. Sparse Reconstruction of Cylinder Wakes

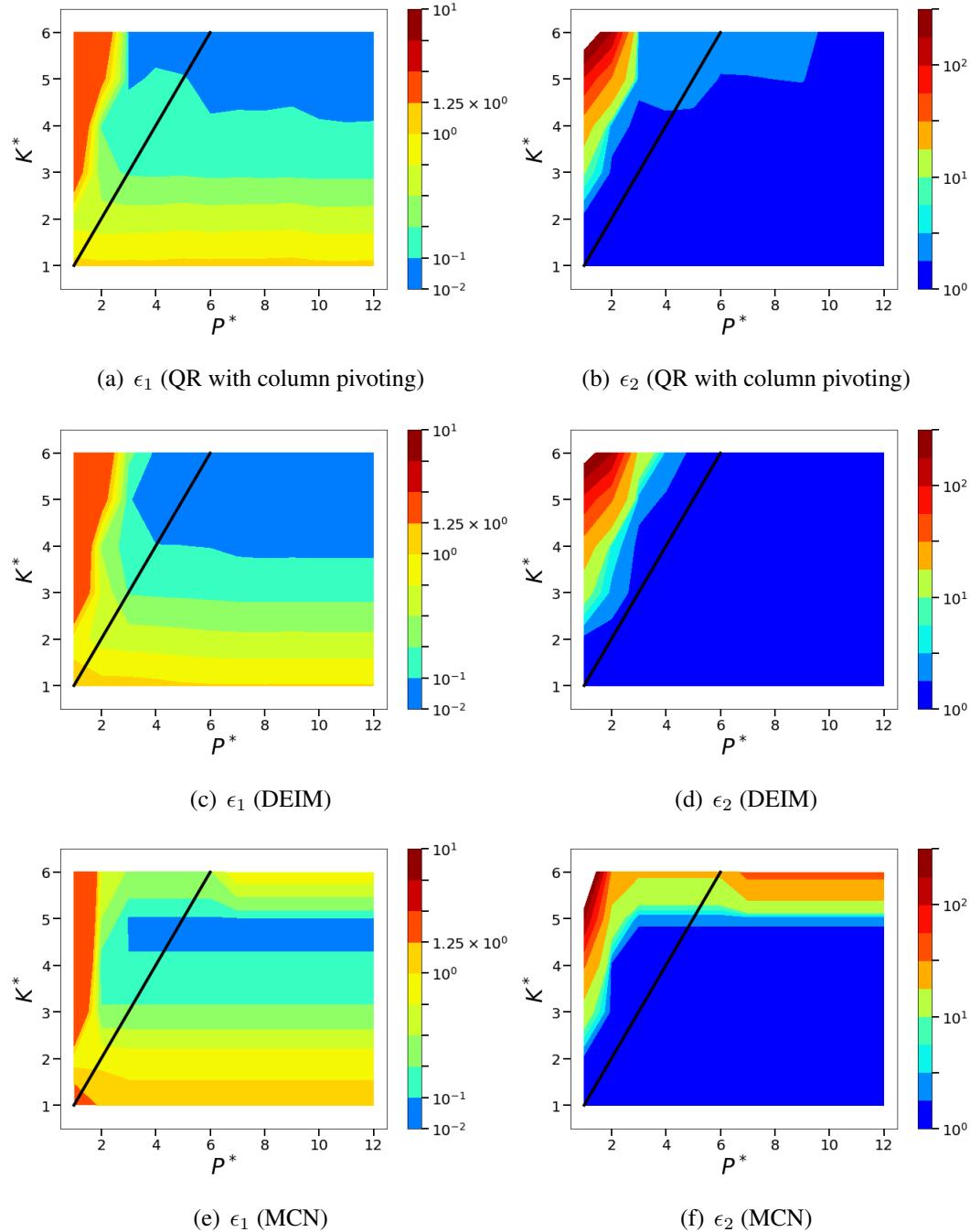
As one of the goals of this study is to establish the baseline SR quality using POD basis we performed a series of nearly one thousand SR experiments corresponding to different points in the  $P^* - K^*$  design space and spread over the different sensor placements. The underlying goal through this work is to bare the aspects of interplay between the various SR control variables, namely, reconstruction dimension,  $K$ , sensor quantity,  $P$  and the choice of sensor placement method (which affects  $C$ ). In all these numerical experiments, the sensor placement is fixed across the range of snapshots as is the case in stationary sensing.



**Figure 10.** Isocontours of the normalized mean squared POD-based sparse reconstruction errors ( $l_2$  norm) corresponding to the sensor placement with maximum and minimum errors from the chosen ensemble of random sensor arrangements. The average error across the entire ensemble of ten random sensor placements is also shown. Left: normalized absolute error metric,  $\epsilon_1$ . Right: normalized relative error metric,  $\epsilon_2$ .

#### 4.4.1. Sparse Reconstruction Accuracy

By computing the errors as described in section 4.3 across the  $K^* - P^*$  space, the contours of  $\epsilon_1$  and  $\epsilon_2$  are generated for both the random as well as greedy sensor placements in figures 10 and 11. As the random sensor placement can lead to high variability between realizations, we estimate multiple sets



**Figure 11.** Isocontours of the normalized mean squared POD-based sparse reconstruction errors ( $l_2$  norm) corresponding to the different greedy sensor placement methods. Left: normalized absolute error metric,  $\epsilon_1$ . Right: normalized relative error metric,  $\epsilon_2$ . (MCN: Minimum Condition Number)

of sensor locations corresponding to different seeds (as denoted by  $\beta$  in this article). Specifically, we compute the SR errors from ten different random seeds and the corresponding reconstruction errors are presented in terms of the mean, maximum and minimum (based on the average over the  $K^* - P^*$  space) in fig. 10. For the greedy ‘smart’ sensor placements a single realization is representative of the method and corresponding error contours for  $\epsilon_1$  and  $\epsilon_2$  are shown (fig. 11). For ease of interpretation, the contour levels in both these figures are made consistent along with appropriately normalizing the error metrics.

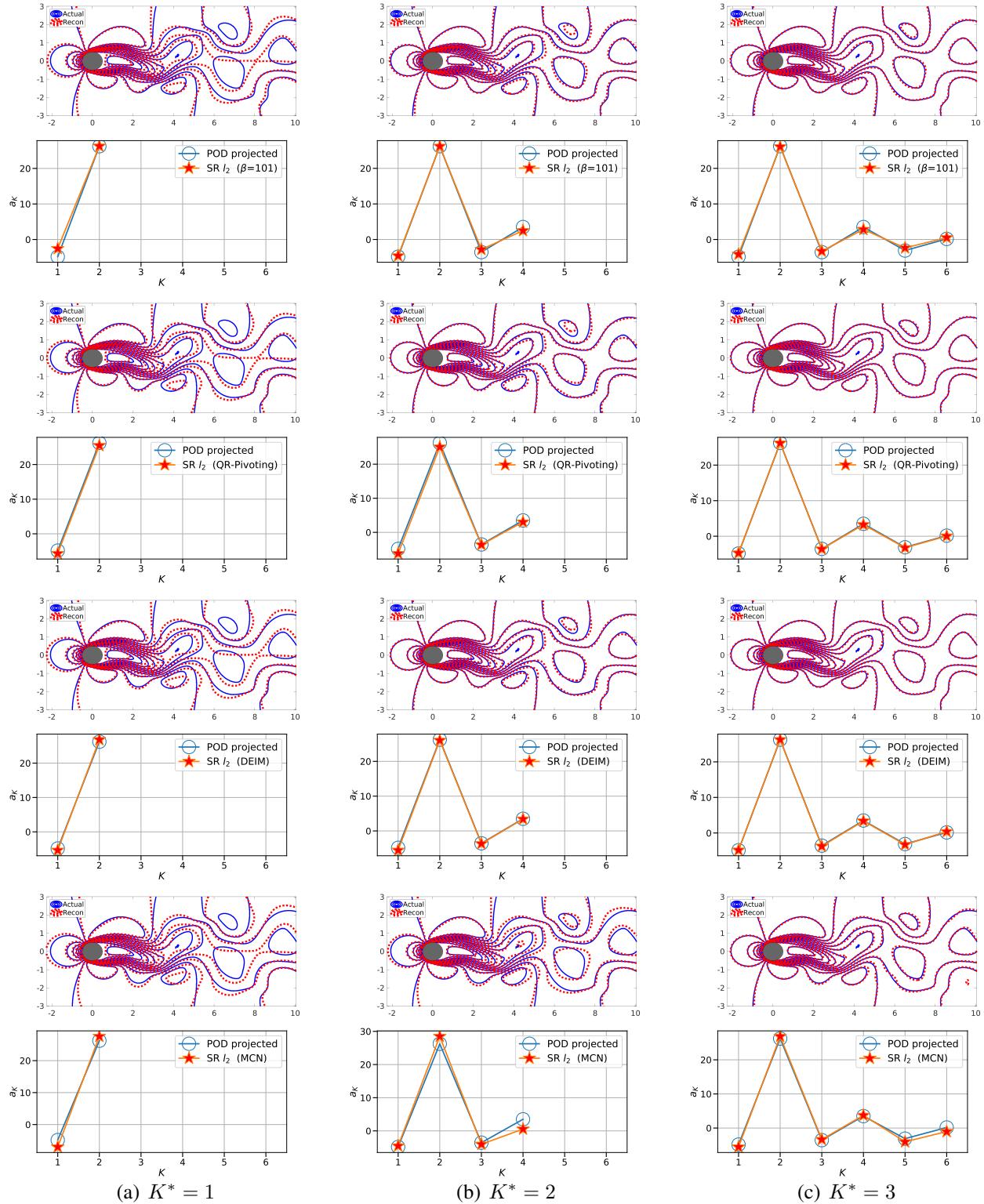
The relative error metric  $\epsilon_2$  (the right column in Figure 10 and 11), shows that the smaller errors (predominantly blue regions) are located in the region where  $P^* > K^*$  and approximately separated from the rest of the  $K^* - P^*$  space using a black diagonal line. This indicates that the oversampled SR problem with  $P > K$ , i.e. having more sensors than the dimensionality chosen to reconstruct the system yields good results in terms of  $\epsilon_2$  while for small  $P^*$ , the normalized relative error can reach as high as  $\mathcal{O}(10^1 - 10^2)$ . Since  $\epsilon_2$  is normalized by the error contained in the exact  $K$ -dimensional POD reconstruction, this metric represents how effectively the sparse sensor data can approximate the  $K$ -dimensional solution using  $l_2$  minimization for the given sensor quantity and placement. In principle, the exact  $K$ -sparse POD reconstruction is the best possible outcome to expect irrespective of how much sensor data is available as long as  $K = N_b$ . Another way of looking at this is that by constraining the SR problem by choosing a relevant POD basis with knowledge of the energy hierarchy, the  $l_2$  reconstruction approaches the  $l_0$  minimization solution, i.e. the  $K$ -sparsest basis are selected by default. Of course, this is not feasible in practical situations where the basis relevance (and its ordering) to the data is not known *a priori*. We also observe that as expected  $\epsilon_1$  contours adhere to a *L*-shaped structure indicating that absolute normalized error reduces as both  $P$  and  $K$  increase due to oversampling and increased system dimension. In practice,  $\epsilon_1$  is the more useful metric for planning and designing the sparse recovery framework for a given flow system, provided the system dimension, i.e.  $K_{95}$  or  $K_{xx}$  is known.

While qualitatively accurate reconstruction is almost always observed for the higher values of  $P^*$  and  $K^*$  for the different sensor placements, there appear to be exceptions in the form of higher reconstruction errors even when  $P^* > K^*$ . This is observed for both the random sensor placement as well as smart sensing approaches. In fact, for the random sensor placement method, a small portion of  $\epsilon_2$  in the region abutting the  $P^* = K^*$  line shows nearly an order of magnitude higher error,  $\mathcal{O}(10^1)$  (colored as yellow in Figure 10) as compared to rest of the region with expected values of  $\mathcal{O}(1)$ . This trend is observed for the greedy sensor placement methods as well, particularly QR-pivoting and MCN. In general, the greedy sensor placement methods show better reconstruction performance when compared to random approaches in the regions surrounding  $P^* \approx K^*$ . This is not surprising given that oversampled ( $P^* \gg K^*$ ) or under-sampled ( $P^* \ll K^*$ ) reconstruction will invariably generate low and high errors respectively. It is only when operating in the transition region that separates under-sampling from over-sampling that sensor placement becomes important.

#### 4.4.2. Assessment of Sensor Placement

Among the three greedy sensor placement methods experimented in this work, DEIM provides the most reliable reconstruction (fig. 11 (c,d)) and closely followed by QR factorization with column pivoting (fig. 11 (a,b)). MCN method which explicitly minimizes the condition number of  $M$  shows good reconstruction accuracy for smaller values of  $K^*$  until about  $K^* \sim 4 - 5$  (see fig. 11 (e,f)). The anomalous inaccuracy that is observed for reconstruction dimensions beyond  $K^* \approx 5$  is due to very few sensors being generated in the wake downstream of the cylinder as seen in fig. 7(d). For this fluid flow, the dominant dynamics occur in the wake and MCN produces very few sensors in this region which under-samples the flow field for higher dimensional reconstruction.

Although the error metrics serve as a useful indicator of performance, we show the exact comparison of the instantaneous sparse reconstructed flow contours (along with the exact field) and the estimated



**Figure 12.** 1<sup>st</sup> row (Random  $\beta = 101$ ), 3<sup>rd</sup> row (QR-Pivot), 5<sup>th</sup> row (DEIM) and 7<sup>th</sup> (MCN) row: we show the line contour comparison of streamwise velocity between the actual CFD solution field (blue) and the POD-based SR reconstruction (red) for  $Re = 100$  at  $P^* = 10$  and  $K^* = 1, 2, 3$ . 2<sup>nd</sup> row (Random  $\beta = 101$ ), 4<sup>th</sup> row (QR with column pivoting), 6<sup>th</sup> row (DEIM) and 8<sup>th</sup> row (MCN) show the corresponding projected (full reconstruction) and sparse recovered coefficients  $a$  from the SR algorithm.

**Table 3.** Sparse reconstruction performance quantification for different sensor location selection method for periodic cylinder flows at  $Re = 100$ .  $\epsilon_1$  is the SR error normalized by the exact reconstruction error corresponding to a dimension of  $K_{95}$ .  $\epsilon_2$  is the SR error normalized by the exact reconstruction error corresponding to a dimension of  $K$ .

Method	$K$	$P$	$K^*$	$P^*$	$\mu_u$	$\mu_v$	$\epsilon_1$	$\epsilon_2$
Random( $\beta = 101$ )	2	20	1.0	10.0	2.548	2.306	1.08E+00	1.08E+00
	4	20	2.0	10.0	2.548	3.247	6.71E-01	1.23E+00
	6	20	3.0	10.0	2.548	4.186	3.17E-01	1.96E+00
QR-Pivoting	2	20	1.0	10.0	2.520	3.794	1.04E+00	1.04E+00
	4	20	2.0	10.0	3.917	3.794	6.05E-01	1.11E+00
	6	20	3.0	10.0	3.917	4.506	1.88E-01	1.16E+00
DEIM	2	20	1.0	10.0	2.323	3.720	1.00E+00	1.00E+00
	4	20	2.0	10.0	3.685	3.867	5.56E-01	1.02E+00
	6	20	3.0	10.0	3.685	4.562	1.69E-01	1.05E+00
MCN	2	20	1.0	10.0	1.090	2.213	1.15E+00	1.15E+00
	4	20	2.0	10.0	1.476	3.502	8.70E-01	1.59E+00
	6	20	3.0	10.0	1.476	3.725	2.76E-01	1.71E+00

POD weights, a comparison with that for the full data reconstruction in fig. 12. In particular, we show results for an over-sampled case  $P^* = 10$  for different reconstruction dimension,  $K^* = 1, 2, 3$  where all the different error contours in figs. 10 and 11 show low error metrics. As expected, the quality of the isocontour reconstruction improves with increasing  $K^*$ . Further, amongst the different experiments for a given over-sampled  $P^*, K^*$ , the DEIM and QR with column pivoting provide the most accurate sparse estimation of the POD coefficients ( $a$ ), especially for the higher  $K^*$ . The relative inaccuracy of the MCN framework in the estimation of  $a$  is observed even for these carefully chosen design points with low average error metrics although the discrepancies are barely noticeable. For such low-dimensional wake flows, these small errors in  $a$  amplify the discrepancy in the full field reconstruction. The relevant quantifications including the sensor quantity, placement method, reconstruction dimension and error metrics for these select dissection cases are summarized in table 3. In addition, we also estimate the coherency parameter for each of these cases. The computed  $\mu$  values are  $\mathcal{O}(1)$  indicating that the rows of the measurement matrix are sufficiently incoherent with respect to the POD basis on average. On careful examination, we observe that the coherency parameters for the QR-pivoting and DEIM placements are higher than the random sensor placement. This again is not surprising as these approaches take advantage of the underlying physical structure as contained in the POD modes to determine sensor placement.

#### 4.4.3. Sparse Reconstruction with Marginally Oversampled Sensors

We observed earlier that the importance of data-driven sensor placement will be felt most when the sensor quantity is comparable to the targeted reconstruction dimension, i.e.  $P^* \approx K^*$ . To verify this, we dissect the instantaneous snapshot reconstruction corresponding to  $K^* = 5$  and  $P^* = 6$  as shown in fig. 13. The plots in the left column of this figure denote the sensor locations, the middle column

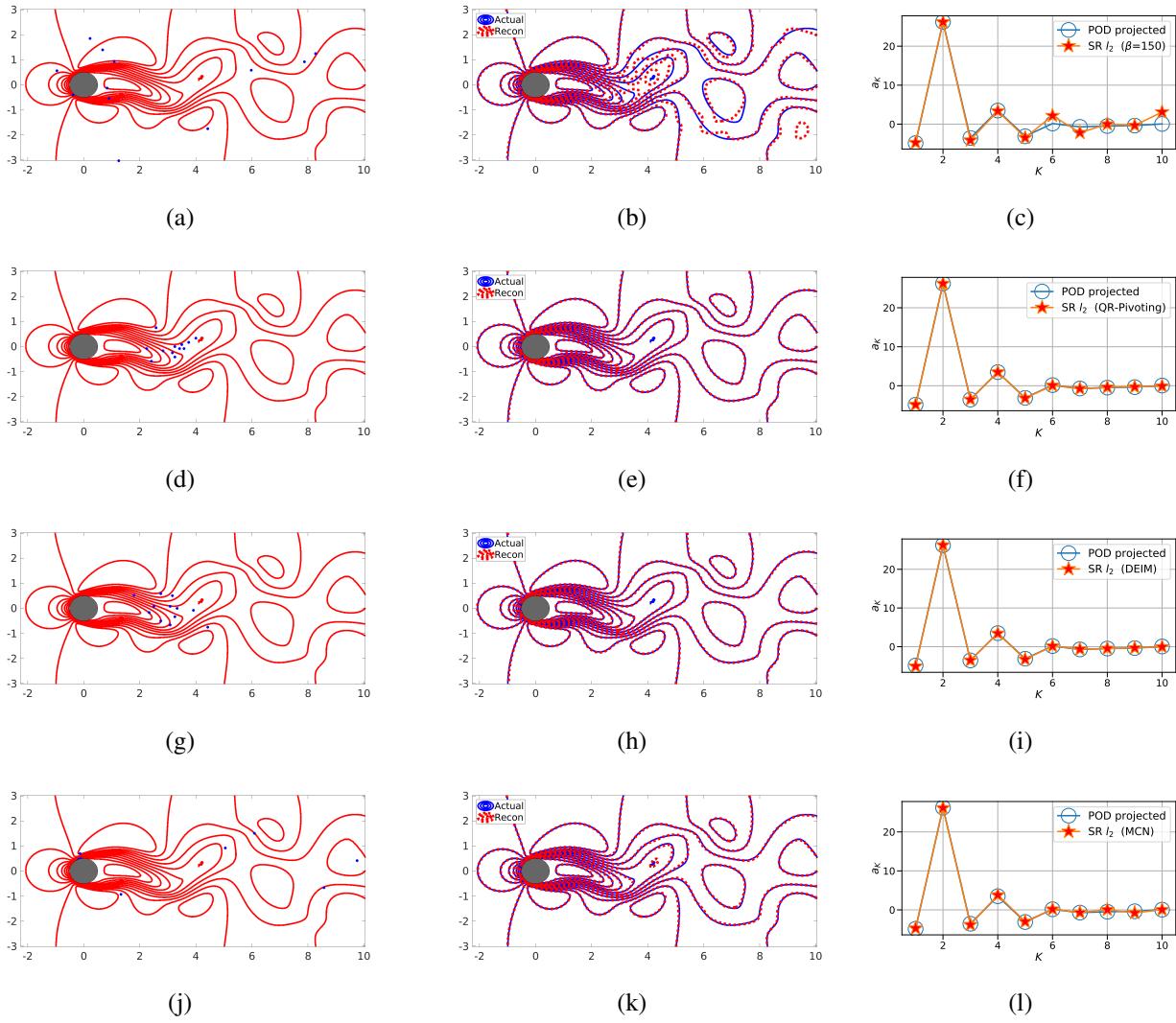
represents the instantaneous reconstructed fields (compared with the exact contours) and the right column shows the predicted and exact POD coefficients. As expected, the greedy smart sensor placements perform better at reconstruction when compared to random sensor placement which is evident from inaccuracy in the reconstructed coefficients (see figs. 13(c,f,i,l)). This is because, the smart sensors are placed at locations that are physically relevant while the random placement as shown in fig. 13 results in under-sampling of the wake region. While not all random sensor placement will generate bad reconstructions, we see strong variability in the error metrics between realizations as a consequence. This is clearly evident from fig. 10 where the variability from different choices of random sensor placement is captured primarily in the under-sampled and marginally oversampled regions of the  $P^* - K^*$  design space.

#### 4.4.4. Sparse Reconstruction with Highly Oversampled Sensors

In the limit of oversampling, i.e.  $P^* > K^*$ , one expects all these different methods to perform accurately as it is not hard even using purely random placement to populate the physically relevant regions. However, we observe surprising inaccuracies with the MCN method in the limit of large  $P$ . We dissect this observation using a single design point,  $K^* = 6, P^* = 10$  (as shown in fig. 14) for which the sparse recovery should be close to exact when applied to this low-dimensional wake flow. Figure 14 shows the sensor locations for each of the different algorithms in the left column, instantaneous reconstructed fields (compared with the exact contours) in the middle column and the predicted POD coefficients compared with the exact values in the right column. We easily see that the sensor combinations corresponding to random, DEIM and QR with column pivoting produce identical results while the sensors based on condition number minimization generates erroneous outcomes. The reasons for this is not hard to decipher as the MCN estimates only six sensors in the wake of the cylinder and any additional sensors that are generated happen to be located on the cylinder surface. This in turn effectively under samples the system although  $P = 20$ , i.e. 20 sensors were used. This is clearly illustrated in fig. 15 where the sensor locations identified for different budgets with  $P^* = 4, 7, 10$  are shown for QR with column pivoting, DEIM and MCN. These plots show that MCN generates all the sensors in the wake region for  $P^* = 4$  and subsequent increase in the budget does not produce meaningful additions. For this low-dimensional system,  $K = 5$  modes captures 99% of the energy, the least condition number of  $M$  appears to be estimated when sensors are placed at locations where the field values are zero. More investigation is needed to identify the causes that underlie this observation.

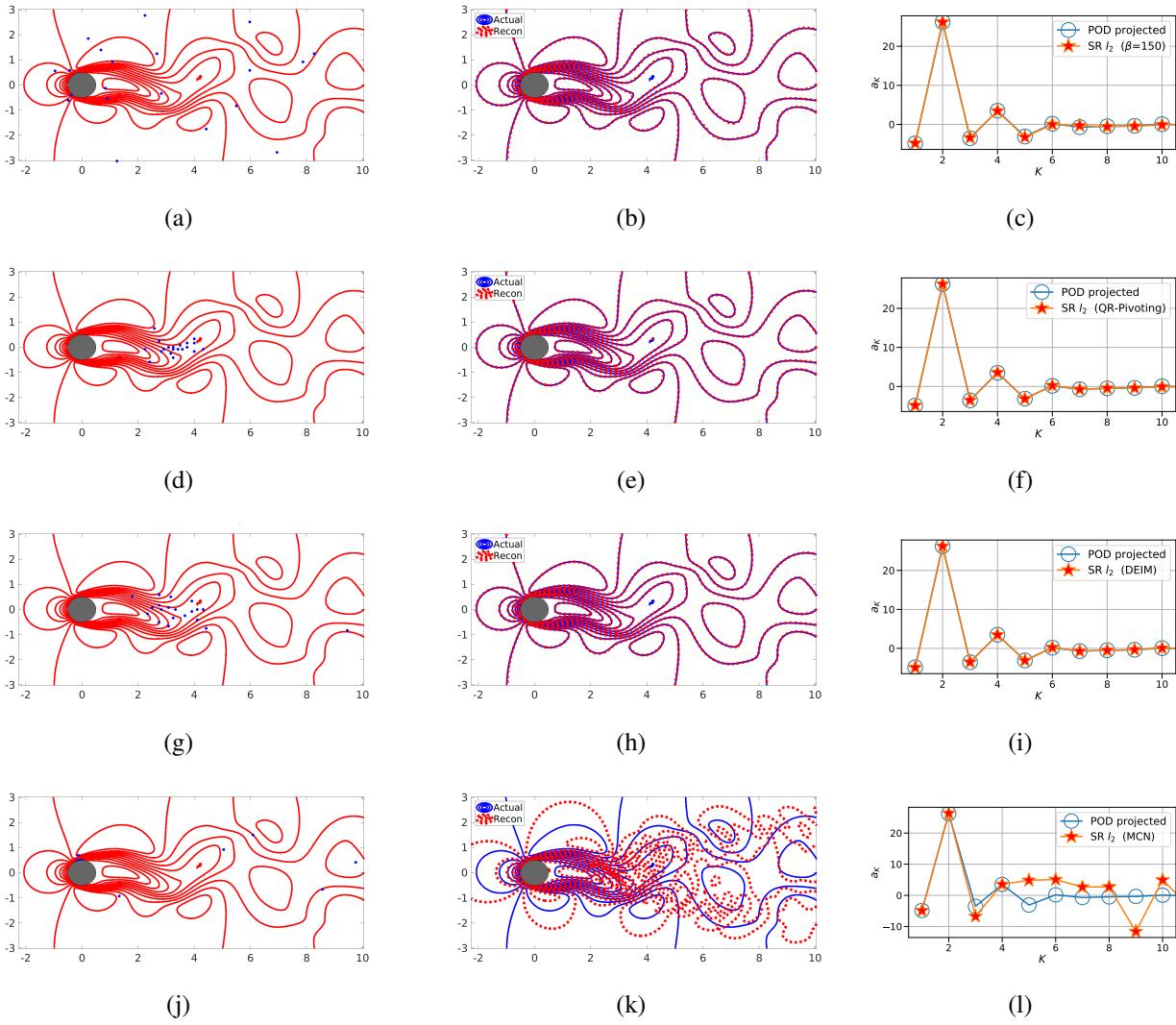
## 5. Conclusion

In this work, we have developed a framework for systematic assessment of sparse reconstruction (SR) performance using flow normalized error metrics and basis/sensor dimension. Using this, we explore the interplay of sensor quantity, their placement and system dimension using optimal data-driven basis from proper orthogonal decomposition for reconstruction from sparse data of canonical low-dimensional wake flows. It is well known that sparse solutions are sought in such SR problems as the candidate basis is typically generic and exhaustive. While such generic candidate basis have a good probability of representing the data, the hierarchy of its relevance to the flow system of interest is not known *a priori*.



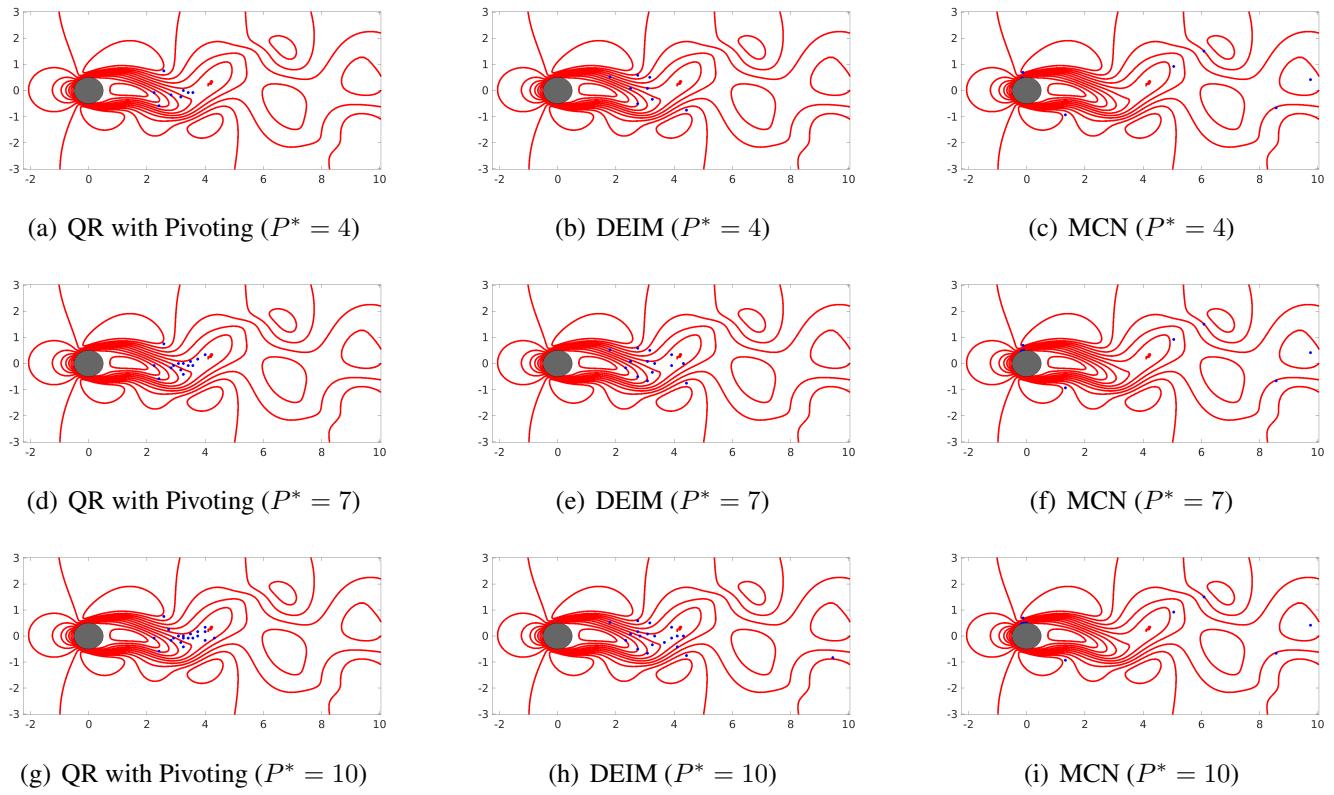
**Figure 13.** Dissection of instantaneous snapshot reconstruction for a marginally oversampled case ( $K^* = 5, P^* = 6$ ). The figure shows the different sensor locations (left column), overlaid true and reconstructed solutions (middle column), and the reconstructed coefficients  $a$  (right column) using POD-based SR for  $Re = 100$ . The different rows correspond to the different sensor placement: random sensor placement with seed  $\beta = 150$  ( $1^{st}$  row), QR factorization with column pivoting ( $2^{nd}$  row), DEIM ( $3^{rd}$  row) and Minimum condition number (MCN) sensor placement ( $4^{th}$  row). The corresponding error quantifications are as follows.  $1^{st}$  row:  $\epsilon_1 = 2.46E-01$  and  $\epsilon_2 = 8.18E+00$ .  $2^{nd}$  row:  $\epsilon_1 = 5.20E-02$  and  $\epsilon_2 = 2.37E+00$ .  $3^{rd}$  row:  $\epsilon_1 = 4.44E-02$  and  $\epsilon_2 = 1.47E+00$ .  $4^{th}$  row:  $\epsilon_1 = 8.04E-02$  and  $\epsilon_2 = 2.67E+00$ .

In this study, we used *a priori* generated POD-basis along with knowledge of its *energy hierarchy* and *relevance* to the given (full state) flow system for sparse recovery. This allows us to bypass the need for more expensive  $l_1$  norm minimization with  $l_2$  approaches of complexity  $\mathcal{O}(NKM)$  as long as the same relevance exists for the sparse measurements. To verify this, we explicitly show that given basis relevance and ordering for the full flow state,  $l_1$  methods using optimal matching pursuit and  $l_2$  minimization (with the first  $K$  basis) using least squares yield the same  $K$ -sparse reconstruction from very few random



**Figure 14.** Dissection of instantaneous snapshot reconstruction for a highly oversampled case ( $K^* = 6, P^* = 10$ ). The figure shows the different sensor locations (left column), overlaid true and reconstructed solutions (middle column), and the reconstructed coefficients  $a$  (right column) using POD-based SR for  $Re = 100$ . The different rows correspond to the different sensor placement: random sensor placement with seed  $\beta = 150$  ( $1^{st}$  row), QR factorization with column pivoting ( $2^{nd}$  row), DEIM ( $3^{rd}$  row) and Minimum condition number (MCN) sensor placement ( $4^{th}$  row). The corresponding error quantifications are as follows.  $1^{st}$  row:  $\epsilon_1 = 5.73E-02$  and  $\epsilon_2 = 3.43E+00$ .  $2^{nd}$  row:  $\epsilon_1 = 3.01E-02$  and  $\epsilon_2 = 1.80E+00$ .  $3^{rd}$  row:  $\epsilon_1 = 1.98E-02$  and  $\epsilon_2 = 1.19E+00$ .  $4^{th}$  row:  $\epsilon_1 = 9.77E-01$  and  $\epsilon_2 = 58.60E+00$ .

measurements. In addition, we also observe from systematic analysis of error metrics over a carefully designed parameter ( $P^* - K^*$ ) space that even marginal oversampling, i.e.  $P \gtrsim K$  is sufficient for accurate full state recovery using  $l_2$  SR with high probability which is akin to the requirement for  $l_0$  minimization. We reiterate here that this study is not intended to advocate the use  $l_2$  over sparsity promoting  $l_1$  methods and acknowledge that the latter is the sensible choice in practical situations where knowledge of the underlying system is scarce.



**Figure 15.** Incremental evolution of sensor placement using the different greedy sensor placement methods (QR-pivoting (left), DEIM (middle) and MCN (right)) as  $P^* = P/K_{95}$  is increased. We show plots for three different cases corresponding to  $P^* = 4, 7, 10$  in each of the rows.

We further expand the  $P - K$  design space to include the effect of data-driven sensor placement with the following candidates: random sensing and greedy-smart sensing algorithms such as DEIM, QR with column pivoting and explicit condition number minimization or MCN. We observe that while the random sampling methods show highly variable errors for marginal oversampling, greedy-smart sensor placement show consistently accurate recovery under these conditions. In the limit of heavy oversampling, the MCN method produces diminishing returns as seen from the wake flow example reported in this study. This can be ascribed to the inability of MCN to generate meaningful sensors in dynamically relevant regions of the flow beyond a small threshold. More research is necessary to delineate the causes for this behavior. Considering the computational complexity of learning sensors from data and the propensity for reliable sparse reconstruction, QR factorization with column pivoting and DEIM (in that order) turn out to be the best alternatives to purely random sampling.

## Acknowledgments

We acknowledge computational resources from OSU High Performance Computing and start-up research funds from the Oklahoma State University.

## Author Contributions

BJ designed the study. CL and AAM developed the codes with input from BJ. CL and AAM carried out the analysis with input from BJ. CL and AAM developed a partial draft of the manuscript and BJ produced the final written manuscript.

### Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Friedman, J.; Hastie, T.; Tibshirani, R. *The elements of statistical learning*; Vol. 1, Springer series in statistics New York, 2001.
2. Holmes, P. *Turbulence, coherent structures, dynamical systems and symmetry*; Cambridge university press, 2012.
3. Berkooz, G.; Holmes, P.; Lumley, J.L. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics* **1993**, *25*, 539–575.
4. Taira, K.; Brunton, S.L.; Dawson, S.; Rowley, C.W.; Colonius, T.; McKeon, B.J.; Schmidt, O.T.; Gordeyev, S.; Theofilis, V.; Ukeiley, L.S. Modal analysis of fluid flows: An overview **2017**, *55*, 4013–4041.
5. Jayaraman, B.; Lu, C.; Whitman, J.; Chowdhary, G. Sparse Convolution-based Markov Models for Nonlinear Fluid Flows. *arXiv preprint arXiv:1803.08222b* **2018**.
6. Bai, Z.; Wimalajeewa, T.; Berger, Z.; Wang, G.; Glauser, M.; Varshney, P.K. Low-dimensional approach for reconstruction of airfoil data via compressive sensing. *AIAA Journal* **2014**.
7. Bright, I.; Lin, G.; Kutz, J.N. Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids* **2013**, *25*, 127102.
8. Candès, E.J.; others. Compressive sampling. Proceedings of the international congress of mathematicians. Madrid, Spain, 2006, Vol. 3, pp. 1433–1452.
9. Tropp, J.A.; Gilbert, A.C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory* **2007**, *53*, 4655–4666.
10. Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE signal processing magazine* **2008**, *25*, 21–30.
11. Needell, D.; Tropp, J.A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis* **2009**, *26*, 301–321.
12. Bui-Thanh, T.; Damodaran, M.; Willcox, K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA journal* **2004**, *42*, 1505–1516.
13. Willcox, K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids* **2006**, *35*, 208–226.
14. Venturi, D.; Karniadakis, G.E. Gappy data and reconstruction procedures for flow past a cylinder. *Journal of Fluid Mechanics* **2004**, *519*, 315–336.
15. Gunes, H.; Sirisup, S.; Karniadakis, G.E. Gappy data: To Krig or not to Krig? *Journal of Computational Physics* **2006**, *212*, 358–382.
16. Gunes, H.; Rist, U. On the use of kriging for enhanced data reconstruction in a separated transitional flat-plate boundary layer. *Physics of Fluids* **2008**, *20*, 104109.
17. Everson, R.; Sirovich, L. Karhunen–Loeve procedure for gappy data. *JOSA A* **1995**, *12*, 1657–1664.
18. Chaturantabut, S.; Sorensen, D.C. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing* **2010**, *32*, 2737–2764.

19. Dimitriu, G.; Ștefănescu, R.; Navon, I.M. Comparative numerical analysis using reduced-order modeling strategies for nonlinear large-scale systems. *Journal of Computational and Applied Mathematics* **2017**, *310*, 32–43.
20. Zimmermann, R.; Willcox, K. An accelerated greedy missing point estimation procedure. *SIAM Journal on Scientific Computing* **2016**, *38*, A2827–A2850.
21. Romberg, J. Imaging via compressive sampling. *IEEE Signal Processing Magazine* **2008**, *25*, 14–20.
22. Saini, P.; Arndt, C.M.; Steinberg, A.M. Development and evaluation of gappy-POD as a data reconstruction technique for noisy PIV measurements in gas turbine combustors. *Experiments in Fluids* **2016**, *57*, 1–15.
23. Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics* **2010**, *656*, 5–28.
24. Rowley, C.W.; Dawson, S.T. Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics* **2017**, *49*, 387–417.
25. Mallet, S. *A Wavelet Tour of Signal Processing*, 1998.
26. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Compressive sampling and dynamic mode decomposition. *arXiv preprint arXiv:1312.5186* **2013**.
27. Brunton, S.L.; Tu, J.H.; Bright, I.; Kutz, J.N. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems* **2014**, *13*, 1716–1732.
28. Bai, Z.; Brunton, S.L.; Brunton, B.W.; Kutz, J.N.; Kaiser, E.; Spohn, A.; Noack, B.R. Data-driven methods in fluid dynamics: Sparse classification from experimental data. In *Whither Turbulence and Big Data in the 21st Century?*; Springer, 2017; pp. 323–342.
29. Kramer, B.; Grover, P.; Boufounos, P.; Nabi, S.; Benosman, M. Sparse sensing and DMD-based identification of flow regimes and bifurcations in complex flows. *SIAM Journal on Applied Dynamical Systems* **2017**, *16*, 1164–1196.
30. Al Mamun, S.; Lu, C.; Jayaraman, B. Extreme Learning Machines as Encoders for Sparse Reconstruction. *Fluids* **2018**, *3*, 88.
31. Trefethen, L.N.; Bau III, D. *Numerical linear algebra*; Vol. 50, Siam, 1997.
32. Manohar, K.; Brunton, B.W.; Kutz, J.N.; Brunton, S.L. Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns. *IEEE Control Systems* **2018**, *38*, 63–86.
33. Tarantola, A. *Inverse problem theory and methods for model parameter estimation*; Vol. 89, siam, 2005.
34. Arridge, S.R.; Schotland, J.C. Optical tomography: forward and inverse problems. *Inverse Problems* **2009**, *25*, 123010.
35. Tarantola, A.; Valette, B. Generalized nonlinear inverse problems solved using the least squares criterion. *Reviews of Geophysics* **1982**, *20*, 219–232.
36. Neelamani, R. Inverse problems in image processing. PhD thesis, Rice University, 2004.
37. Khemka, A. Inverse problems in image processing. PhD thesis, Purdue University, 2009.

38. Donoho, D.L. Compressed sensing. *IEEE Transactions on information theory* **2006**, *52*, 1289–1306.
39. Baraniuk, R.G. Compressive sensing [lecture notes]. *IEEE signal processing magazine* **2007**, *24*, 118–121.
40. Baraniuk, R.G.; Cevher, V.; Duarte, M.F.; Hegde, C. Model-based compressive sensing. *IEEE Transactions on Information Theory* **2010**, *56*, 1982–2001.
41. Sarvotham, S.; Baron, D.; Wakin, M.; Duarte, M.F.; Baraniuk, R.G. Distributed compressed sensing of jointly sparse signals. Asilomar conference on signals, systems, and computers, 2005, pp. 1537–1541.
42. Candès, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory* **2006**, *52*, 489–509.
43. Candès, E.J.; Romberg, J.K.; Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics* **2006**, *59*, 1207–1223.
44. Candès, E.J.; Tao, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory* **2006**, *52*, 5406–5425.
45. Candès, E.J.; Romberg, J.K. Signal recovery from random projections. Computational Imaging III. International Society for Optics and Photonics, 2005, Vol. 5674, pp. 76–87.
46. Chen, S.S.; Donoho, D.L.; Saunders, M.A. Atomic decomposition by basis pursuit. *SIAM review* **2001**, *43*, 129–159.
47. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **1996**, pp. 267–288.
48. Candès, E.J.; Wakin, M.B.; Boyd, S.P. Enhancing sparsity by reweighted  $\ell_1$  minimization. *Journal of Fourier analysis and applications* **2008**, *14*, 877–905.
49. Kim, S.J.; Koh, K.; Lustig, M.; Boyd, S.; Gorinevsky, D. An Interior-Point Method for Large-Scale L1 Regularized Least Squares. *IEEE journal of selected topics in signal processing* **2007**, *1*, 606–617.
50. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **2016**, p. 201517384.
51. Lumley, J. Stochastic Tools in Turbulence (New York: Academic) **1970**.
52. Sirovich, L. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of applied mathematics* **1987**, *45*, 561–571.
53. Brunton, S.L.; Rowley, C.W.; Williams, D.R. Reduced-order unsteady aerodynamic models at low Reynolds numbers. *Journal of Fluid Mechanics* **2013**, *724*, 203–233.
54. Lu, C.; Jayaraman, B. Interplay of Sensor Quantity, Placement and System Dimensionality on Energy Sparse Reconstruction of Fluid Flows. *arXiv preprint arXiv:1806.08428* **2018**.
55. Candès, E.; Romberg, J. Sparsity and incoherence in compressive sampling. *Inverse problems* **2007**, *23*, 969.
56. Csató, L.; Opper, M. Sparse on-line Gaussian processes. *Neural computation* **2002**, *14*, 641–668.

57. Cohen, K.; Siegel, S.; McLaughlin, T. Sensor placement based on proper orthogonal decomposition modeling of a cylinder wake. 33rd AIAA Fluid Dynamics Conference and Exhibit, 2003, p. 4259.
58. Kubrusly, C.S.; Malebranche, H. Sensors and controllers location in distributed systems—A survey. *Automatica* **1985**, *21*, 117–128.
59. Chaturantabut, S.; Sorensen, D.C. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM Journal on numerical analysis* **2012**, *50*, 46–63.
60. Willcox, K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids* **2006**, *35*, 208–226.
61. Manohar, K.; Brunton, B.W.; Kutz, J.N.; Brunton, S.L. Data-driven sparse sensor placement for reconstruction. *arXiv preprint arXiv:1701.07569* **2017**.
62. Yildirim, B.; Chryssostomidis, C.; Karniadakis, G. Efficient sensor placement for ocean measurements using low-dimensional concepts. *Ocean Modelling* **2009**, *27*, 160–173.
63. Jayaraman, B.; Lu, C.; Whitman, J.; Chowdhary, G. Sparse Convolution-based Markov Models for Nonlinear Fluid Flows. *arXiv preprint arXiv:1803.08222* **2018**.
64. Hanagud, S.; de Noyer, M.B.; Luo, H.; Henderson, D.; Nagaraja, K. Tail buffet alleviation of high-performance twin-tail aircraft using piezostack actuators. *AIAA Journal* **2002**, *40*, 619–627.
65. Barrault, M.; Maday, Y.; Nguyen, N.C.; Patera, A.T. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* **2004**, *339*, 667–672.
66. Bui-Thanh, T.; Damodaran, M.; Willcox, K. Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. 21st AIAA Applied Aerodynamics Conference, 2003, p. 4213.
67. Roshko, A. On the development of turbulent wakes from vortex streets **1954**.
68. Williamson, C. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics* **1989**, *206*, 579–627.
69. Noack, B.R.; Afanasiev, K.; Morzynski, M.; Tadmor, G.; Thiele, F. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics* **2003**, *497*, 335–363.
70. Cantwell, C.D.; Moxey, D.; Comerford, A.; Bolis, A.; Rocco, G.; Mengaldo, G.; De Grazia, D.; Yakovlev, S.; Lombard, J.E.; Ekelschot, D.; others. Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications* **2015**, *192*, 205–219.
71. Rajani, B.; Kandasamy, A.; Majumdar, S. LES of Flow past Circular Cylinder at  $Re=3900$ . *Journal of Applied Fluid Mechanics* **2016**, *9*.
72. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* **2009**, *2*, 183–202.