Computers and Chemical Engineering 106 (2017) 501–511

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Development of local dynamic mode decomposition with control: Application to model predictive control of hydraulic fracturing

Abhinav Narasingam[a,b], Joseph Sang-Il Kwon[a,b,*]

[a] Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77845, USA
[b] Texas A&M Energy Institute, Texas A&M University, College Station, TX 77845, USA

ARTICLE INFO

Article history:
Received 18 April 2017
Received in revised form 17 June 2017
Accepted 3 July 2017
Available online 12 July 2017

Keywords:
Distributed parameter systems
Reduced-order model
Temporal clustering
Model predictive control
Hydraulic fracturing
Dynamic mode decomposition

ABSTRACT

Dynamic mode decomposition with control (DMDc) is a modal decomposition method that extracts dynamically relevant spatial structures disambiguating between the underlying dynamics and the effects of actuation. In this work, we extend the concepts of DMDc to better capture the local dynamics associated with highly nonlinear processes and develop temporally local reduced-order models that accurately describe the fully-resolved data. In this context, we first partition the data into clusters using a Mixed Integer Nonlinear Programming based optimization algorithm, the Global Optimum Search, which incorporates an added feature of predicting the optimal number of clusters. Next, we compute the reduced-order models tailored to each cluster by applying DMDc within each cluster. The developed models are subsequently used to compute approximate solutions to the original high-dimensional system and to design a feedback control system of hydraulic fracturing processes for the computation of optimal pumping schedules.

Published by Elsevier Ltd.

1. Introduction

Many chemical processes and fluid systems have models that apparently describe their dynamics to near-perfect accuracy. However, very often, these turn out to be high-dimensional complex models and the fully-resolved simulations necessary to capture their detailed nonlinear behaviors put a considerable strain on computational resources. This limits the capability to perform parameter estimation or design feedback control systems which require real-time computation of dynamic solutions. Nevertheless, despite the fact that they are governed by high-dimensional systems, very often the dominant behavior can be captured by modes that are many orders of magnitude smaller than the dimension of the original system. For instance, Noack et al. (2003) showed that as few as three ordinary differential equations can describe the essential features of a laminar flow past a 2D cylinder. Thus, practical engineering strategies for dealing with high-dimensional data require developing simplified (reduced-order) models that significantly reduce the dimension of the underlying system to remain computationally-tractable at the negligible expense of model accuracy.

The field of reduced-order modeling is large, and new methods are being developed at a fast pace. Among these, two of the most commonly used modal decomposition techniques are Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD). Both of them are based on analyzing information from a sequence of observational data arising from high-dimensional systems to identify coherent structures embedded in the system. POD determines the structures that capture the most energy to optimally reconstruct a data set arising from a linear or nonlinear dynamical process in the mean square sense (Holmes et al., 1996). However, the energy criterion may not be the relevant measure to precisely rank the flow structures in all the circumstances (Noack et al., 2008). DMD has originally been introduced in the fluids community (Schmid and Sesterhenn, 2008) to yield flow structures that accurately describe the motion of the flow. In contrast to POD, this method extracts modes that are dynamically relevant spatial structures rather than selecting the dominant modes that capture most of the flow energy. Ghommem et al. (2014) has shown that the DMD-based approach extracted modes that are more relevant for long-term dynamics compared to POD. Computationally, DMD assumes a linear model that best represents the underlying dynamics, even if those dynamics stem from a nonlinear process. Although it might seem equivocal describing a nonlinear system by superposition of modes whose dynamics are governed by the corresponding eigenvalues, DMD can be considered a numerical

* Corresponding author at: Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77845, USA.
E-mail address: kwonx075@tamu.edu (J.S.-I. Kwon).

http://dx.doi.org/10.1016/j.compchemeng.2017.07.002
0098-1354/Published by Elsevier Ltd.

approximation to Koopman spectral analysis providing theoretical justification for characterizing nonlinear systems (Rowley et al., 2009; Mezić, 2013; Bagheri, 2013). After gaining quick popularity, DMD has found successful implementation in many fluid mechanics applications to analyze both numerical (Schmid, 2010; Schmid et al., 2011; Seena and Sung, 2011; Mizuno et al., 2011; Muld et al., 2012) as well as experimental (Schmid, 2009, 2011; Schmid et al., 2009; Pan et al., 2011; Semeraro et al., 2012; Lusseyran et al., 2011; Duke et al., 2012) flow field data and help characterizing relevant physical mechanisms. Several efforts have been made to explore the connections of DMD with other methods, such as Eigensystem Realization Algorithm (ERA) (Tu et al., 2014), Fourier Analysis (Chen et al., 2012), POD (Schmid, 2010) and Koopman analysis (Rowley et al., 2009; Mezić, 2013; Bagheri, 2013). Several variants of the DMD algorithm have also been proposed, including optimized DMD (Chen et al., 2012), optimal mode decomposition (Goulart et al., 2012; Wynn et al., 2013), sparsity promoting DMD (Jovanović et al., 2014) and extended DMD (Williams et al., 2015).

Within this context, Proctor et al. (2016) extended the concepts of DMD and introduced the method of Dynamic Mode Decomposition with control (DMDc) to utilize both measurements of the system and applied external inputs in extracting the underlying dynamics. Additionally, DMDc also provides a description of how the control inputs affect the system, and with this understanding of the input-to-output behavior, a reduced-order model can be generated and used in the design of feedback control systems to regulate the original high-dimensional systems. DMDc inherits a number of advantages of DMD in that it is a completely data-driven framework and can be applied to nonlinear systems. Furthermore, there are a number of connections between DMDc and other popular system identification methods such as Observer Kalman Filter Identification (OKID) (Juang et al., 1991), Numerical Algorithms for Subspace State Space System Identification (N4SID) (Van Overschee and De Moor, 1994), Multivariable Output Error State Space (MOESP) (Van Overschee and De Moor, 1996), and Canonical Variate Analysis (CVA) (Katayama, 2005). Algorithmically, these methods involve regression, model reduction, and parameter estimation steps similar to DMDc. However, differences do exist in terms of the similarity transformation required for projection and the use of an orthogonal complement of control inputs to generate the approximate solution (Qin, 2006). Therefore, DMDc can be used in diverse engineering applications, one of which is presented in this manuscript, where the study of dynamics while simultaneously considering the applied control input to the complex systems is important.

However, for a highly nonlinear system, the assumption of linear relation might not work well, especially in the case of limited access to spatial and/or temporal measurements. Additionally, these global methods fail to capture the local dynamics when the process parameters change with space and time (e.g., permeability and Young's modulus in the rock formation are space-dependent constants). Based on these observations, in order to capture the local dynamics of a complex nonlinear system more effectively, the embedded coherent structures must be tailored to the temporally local behavior of every portion of the solution trajectory. This idea of local model reduction has been applied successfully in several applications. In Dihlmann et al. (2011), the time domain was partitioned into multiple subdomains and temporally local eigenfunctions were used to construct a reduced-order model. Local bases were also exploited by Anttonen (2001) in the context of aeroelastic applications according to a space domain partition. In Ghommem et al. (2013), a global-local model reduction approach was presented where a generalized multiscale finite element method (GMsFEM) was combined with DMD and/or POD and applied to flows in high-contrast porous media. In Efendiev et al. (2012), a balanced truncation based global model reduction was efficiently combined with the local model reduction tools introduced in Efendiev et al. (2011). Our previous work also exploited the idea of time-domain partitioning to develop temporally local reduced-order models to accurately approximate the fully-resolved data (Narasingam et al., 2017).

Our contribution in this work is integrating the idea of temporal-clustering to DMDc to develop tailored temporally local reduced-order models that reproduce the essential features of the underlying system and quantify the effect of control inputs on the process dynamics of the system. Because the accuracy of the local approach depends on the number of clusters, our motivation is to obtain inexpensive decomposition problems while preserving the accuracy of the approximated models. In the proposed framework, we achieve this goal by using the Global Optimum Search algorithm (GOS) which predicts the optimal number of clusters based on the similarity and dissimilarity of the cluster configurations (Tan et al., 2007). Since the developed temporally local reduced-order models are computationally inexpensive, yet fairly accurate dynamic models, they can be used to design feedback control systems based on a model predictive control (MPC) framework. The proposed model reduction technique is different from other earlier works in that it is a completely data-driven approach and does not require any knowledge in terms of the system model. For example, Christofides and Daoutidis (1997) used an elegant singular perturbation based approximate inertial manifolds (AIMs) for the construction of low-dimensional ordinary differential equations (ODEs) that are subsequently used in the synthesis of non-linear output feedback controllers that guarantee stability. Baker and Christofides (2000) extended this concept to nonlinear parabolic PDE systems by first computing a set of empirical eigenfunctions to be used as basis functions in the Galerkin's framework. In Armaou and Christofides (2001), a mathematical transformation was used to construct a finite-dimensional model of a nonlinear parabolic PDE system with time-varying spatial domains. Izadi and Dubljevic (2013) presented a systematic approach to compute time-varying empirical eigenfunctions by an appropriate mapping onto the time-varying spatial domain. However, the aforementioned methods require mathematical expressions for system models and their time-varying spatial domains to construct low-dimensional approximate models. In contrast, the proposed approach requires no such knowledge and approximate models can simply be obtained using the data snapshots.

The remainder of this work is organized as follows. Section 2 provides a theoretical background on DMD and presents mathematical formulations for DMDc. Section 3 discusses the proposed methodology and outlines an algorithm for computing temporally local reduced-order models. Application of the proposed temporally local DMDc method on a hydraulic fracturing process is described in Section 4, and a series of numerical simulation results including (a) a performance assessment of the proposed local method over the global method, (b) a study of factors influencing the accuracy of the proposed method and (c) a comprehensive design of a model predictive controller to achieve a specified control objective in the hydraulic fracturing process are presented.

2. Theoretical background

2.1. Dynamic mode decomposition

DMD is a method that can extract dynamically relevant spatial structures solely from the data of a high-dimensional complex system. These structures, called dynamic modes, are equivalent to a linear tangent approximation and describe the dominant dynamic behavior of a nonlinear data sequence. In this section, we present a short mathematical description of DMD and the formulations required for DMDc.

We start with a general description of state vector fields $\mathbf{x}$ collected at regular time intervals Δt, either by direct numerical simulations or from experimental data, $\mathbf{x}_i = \mathbf{x}(t_i) \in \mathbb{R}^n$ where $t_i = i\Delta t$ and n represents the number of available spatial measurements. Through the course of this paper, we will refer to these measurements as snapshots. The fundamental assumption in DMD is that we seek a linear operator $\mathbf{A}$ that approximately connects the snapshot $\mathbf{x}_i$ to the subsequent snapshot $\mathbf{x}_{i+1}$, which is given by

$$\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i, \tag{1}$$

and that this mapping is approximately the same over the full sampling interval $t \in [0, (m-1)\Delta t]$. We collect an ensemble of trajectories containing m snapshots and arrange them into two matrices,

$$\mathbf{X}_1 = [\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_{m-1}],\ \mathbf{X}_2 = [\mathbf{x}_2\, \mathbf{x}_3 \ldots \mathbf{x}_m] \tag{2}$$

Based on the above description the relation between the two snapshot matrices is given by

$$\mathbf{X}_2 = \mathbf{A}\mathbf{X}_1 \tag{3}$$

The primary objective now is to solve for an approximation of the system matrix $\mathbf{A}$. Rowley et al. (2009) presented a mathematical expression for computing the matrix $\mathbf{A}$ using a companion matrix based on a variant of the Arnoldi algorithm. A more well-conditioned algorithm, which we will follow here, based on a similarity transformation achieved using a singular value decomposition (SVD) is presented by Schmid (2010). The computation of $\mathbf{A}$ follows from applying the pseudo-inverse of $\mathbf{X}_1$ to both sides of Eq. (3).

$$\mathbf{A} = \mathbf{X}_2\mathbf{X}_1^{\dagger} \tag{4}$$

where $\mathbf{X}_1^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\mathbf{X}_1$. Computationally, the most efficient way to perform the pseudo-inverse is via SVD.

$$\mathbf{X}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \approx \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^* \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n\times m-1}$, $\mathbf{V}^* \in \mathbb{R}^{m-1\times m-1}$, $\hat{\mathbf{U}} \in \mathbb{R}^{n\times r}$, $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{r\times r}$, $\hat{\mathbf{V}}^* \in \mathbb{R}^{r\times m-1}$ and * denotes the complex conjugate transpose.

We recognize that in the above decomposition the left singular vectors, $\mathbf{U}$, contains proper orthogonal modes of the snapshot set and hence amounts to a projection of the linear operator $\mathbf{A}$ onto a set of POD bases. Furthermore, Eq. (5) results in a low-dimensional representation of $\mathbf{A}$ by accounting for the rank-deficiency in the snapshot sequence via a limited number of projection modes given by r. The appropriate value of r can be given by non-zero singular values (or more practically singular values above a prescribed threshold) (Eckart and Young, 1936; Mirsky, 1960; Golub and Reinsch, 1970; Chen et al., 2012). The approximation for the linear operator $\mathbf{A}$ can thus be computed from SVD of the snapshot set by plugging Eq. (5) into Eq. (4)

$$\mathbf{A} \approx \mathbf{X}_2\mathbf{X}_1^{\dagger} = \mathbf{X}_2\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}^* \tag{6}$$

The reduced-order model can therefore be derived by taking a basis transformation, $\hat{\mathbf{U}}^*\mathbf{X} = \hat{\mathbf{X}}$,

$$\begin{aligned}\hat{\mathbf{X}}_2 &= \hat{\mathbf{U}}^*\mathbf{A}\hat{\mathbf{U}}\hat{\mathbf{X}}_1\\ &= \hat{\mathbf{U}}^*\mathbf{X}_2\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{X}}_1\\ &= \hat{\mathbf{A}}\hat{\mathbf{X}}_1\end{aligned} \tag{7}$$

where $\hat{\mathbf{X}} \in \mathbb{R}^r$ represents the states in the reduced-order model obtained by projecting the original states onto a new subspace with a reduced dimension r, and $\hat{\mathbf{A}}$ denotes an approximation to the linear operator $\mathbf{A}$.

The eigendecomposition of $\hat{\mathbf{A}}$ defined by $\hat{\mathbf{A}}\mathbf{W} = \mathbf{\Lambda}\mathbf{W}$ yields the eigenvalues and eigenvectors. According to the property of similarity transformation, the eigenvalues of $\hat{\mathbf{A}}$ are also the eigenvalues of the full matrix $\mathbf{A}$. The eigenvectors of the full matrix $\mathbf{A}$ are called dynamic modes, denoted by $\mathbf{\Phi}$, and are related to the eigenvectors of $\hat{\mathbf{A}}$ via the following transformation,

$$\mathbf{\Phi} = \mathbf{P}\mathbf{W} = \mathbf{X}_2\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W} \tag{8}$$

where $\mathbf{P} = \mathbf{X}_2\mathbf{V}\mathbf{\Sigma}^{-1}$ is the required linear transformation.

2.2. Dynamic mode decomposition with control (DMDc)

Building on the DMD theory, Proctor et al. (2016) developed the DMDc method in order to ascertain the dynamic characteristics of a system that depend both on inherent dynamics (e.g., states) as well as applied external inputs. In addition to the sequence of state snapshots in DMD, we now collect the sequence of input snapshots as

$$\mathbf{\Gamma} = [\mathbf{u}_1\, \mathbf{u}_2 \ldots \mathbf{u}_{m-1}] \tag{9}$$

where $\mathbf{u}_i \in \mathbb{R}^l$ and l denotes the number of input variables. The system description can therefore be rewritten in an augmented form as

$$\mathbf{X}_2 = \mathbf{A}\mathbf{X}_1 + \mathbf{B}\mathbf{\Gamma} = [\mathbf{A}\,\mathbf{B}]\begin{bmatrix}\mathbf{X}_1\\ \mathbf{\Gamma}\end{bmatrix} = \mathbf{G}\mathbf{\Omega} \tag{10}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times l}$, $\mathbf{\Gamma} \in \mathbb{R}^{l\times(m-1)}$, $\mathbf{\Omega} \in \mathbb{R}^{(n+l)\times(m-1)}$ and $\mathbf{G} \in \mathbb{R}^{n\times(n+l)}$.

In contrast to DMD, SVD here, is performed on the augmented data matrix $\mathbf{\Omega} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^*$ and the approximations for $\mathbf{A}$ and $\mathbf{B}$ are derived as follows:

$$\begin{aligned}\mathbf{A} &\approx \mathbf{X}_2\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}_1^*\\ \mathbf{B} &\approx \mathbf{X}_2\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}_2^*\end{aligned} \tag{11}$$

where $\hat{\mathbf{U}}_1^* \in \mathbb{R}^{n\times p}$, $\hat{\mathbf{U}}_2^* \in \mathbb{R}^{l\times p}$, $[\hat{\mathbf{U}}_1\ \ \hat{\mathbf{U}}_2]^T = \hat{\mathbf{U}}$ and p denotes the reduced-order of the augmented system.

In the next step, we project the state onto a subspace on which it evolves using a basis transformation. However, unlike DMD, in DMDc, the truncated left singular vectors $\hat{\mathbf{U}}$ of the augmented matrix, $\mathbf{\Omega}$, can not be used to define this transformation because they include the inputs as well as states. To find a linear transformation for the states alone, we utilize a reduced-order subspace of the shifted snapshot sequence, $\mathbf{X}_2$. This fundamental observation allows for DMDc to discover a reduced-order representation of the dynamics $\mathbf{A}$ and the input matrix $\mathbf{B}$. Computationally, SVD of the shifted snapshot sequence ($\mathbf{X}_2 = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*$ where $\tilde{\mathbf{U}} \in \mathbb{R}^{n\times r}$, $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{r\times r}$, $\tilde{\mathbf{V}} \in \mathbb{R}^{m-1\times r}$ and r denotes the dimension of the subspace) is used to define the linear transformation required for projecting the state. Using the new transformation, a low-dimensional representation of the system matrices can be computed as follows:

$$\begin{aligned}\hat{\mathbf{A}} &\triangleq \tilde{\mathbf{U}}^*\mathbf{A}\tilde{\mathbf{U}} = \tilde{\mathbf{U}}^*\mathbf{X}_2\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}_1^*\tilde{\mathbf{U}}\\ \hat{\mathbf{B}} &\triangleq \tilde{\mathbf{U}}^*\mathbf{B} = \tilde{\mathbf{U}}^*\mathbf{X}_2\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}_2^*\end{aligned} \tag{12}$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{r\times r}$ and $\hat{\mathbf{B}} \in \mathbb{R}^{r\times l}$. In general, the dimension p of the reduced SVD for augmented data is greater than that of the dimension r of the subspace onto which the states are projected. Please note that one could equivalently use the singular value decomposition of $\mathbf{X}_1$ to obtain the required basis transformation. Depending on the choice of $\mathbf{X}_1$ or $\mathbf{X}_2$ the corresponding linear transformation will be unique.

3. Local dynamic mode decomposition with control

In this work, we apply the concept of DMDc to approximate a nonlinear system. Recall, the basic assumption that drives DMDc is that the snapshots are approximately related by a linear operator. Therefore, DMDc may fail to capture the local dynamics if there is significant nonlinearity in the local dynamics. To handle this issue, in this work, we suggest a two-step process where, in the first stage, the snapshots are partitioned into clusters based on the local dynamics exhibited by the solution, and in the second stage, DMDc is applied to each cluster to compute the local dynamic modes that describe the dominant local dynamic behavior within each cluster.

3.1. Temporal clustering

We first partition the ensemble of generated snapshots into clusters by applying the GOS algorithm. For a complete understanding of the GOS algorithm, we refer the readers to (Tan et al., 2007). Let us suppose we want to group n snapshots into m clusters. Without loss of generality, we can assume that these snapshots are sampled with uniform time intervals, say $\mathbf{x}_j$ for $j = 1, \ldots, n$ where n is the total number of snapshots. We then seek to minimize the distances between the snapshots and the cluster centers, which can be formulated as the following Mixed-Integer Nonlinear Programming (MINLP) problem:

$$\begin{aligned}
\underset{c_{ki}, y_{jk}}{\text{Minimize}} \quad & \sum_{i=1}^{s}\sum_{j=1}^{n} x_{ij}^2 - \sum_{i=1}^{s}\sum_{j=1}^{n}\sum_{k=1}^{m}(x_{ij} y_{jk} c_{ki}) \\
\text{s.t} \quad & c_{ki}\sum_{j=1}^{n} y_{jk} - \sum_{j=1}^{n} x_{ij} y_{jk} = 0, \ \forall i, k \\
& \sum_{k=1}^{m} y_{jk} = 1, \ \forall j = 1, \ldots, n \\
& 1 \le \sum_{j=1}^{n} y_{jk} \le n - m + 1 \\
& y_{jk} \in \{0, 1\}, \ \forall j, k \\
& c_{ki}^L \le c_{ki} \le c_{ki}^U, \ \forall i, k \\
& c_{ki}^L = \min\{x_{ij}\}, \ \forall i = 1, \ldots, s \\
& c_{ki}^U = \max\{x_{ij}\}, \ \forall i = 1, \ldots, s
\end{aligned} \tag{13}$$

where i denotes the spatial points, y_{jk} are binary variables which indicate whether a snapshot j falls within the kth cluster, c_{ki} are continuous variables representing the cluster centers, and c_{ki}^L, c_{ki}^U denote the lower and upper bounds of c_{ki}, respectively. The necessary optimality condition is included in the first set of constraints. The second requires that each snapshot be placed in only one cluster. The third specifies that there is at least one and no more than $(n - m + 1)$ snapshots in a cluster. We apply the GOS algorithm and solve the above MINLP problem to obtain the clusters.

$$\mathcal{C}^k = \{\mathbf{x}_j | (\|\mathbf{x}_j - \mathbf{c}_k\|_2^2) \le (\|\mathbf{x}_j - \mathbf{c}_\ell\|_2^2), \quad \forall \ell \ne k\}, \ k = 1, \ldots, m \tag{14}$$

where $\mathcal{C}^k$ is the kth cluster, $\mathbf{c}_k$ is the center of the kth cluster and m is the optimal number of clusters which is determined from the clustering balance curve (Tan et al., 2007). The clustering balance, ϵ is defined as:

$$\epsilon = \frac{1}{2}\left(\sum_{i=1}^{s}\sum_{j=1}^{n}\sum_{k=1}^{m} y_{jk}\|x_{ij} - c_{ki}\|_2^2 + \sum_{i=1}^{s}\sum_{k=1}^{m}\|c_{ki} - c_i^\circ\|_2^2\right) \tag{15}$$

where the first and second terms are the intra and inter-cluster error sums, respectively, and $c_i^\circ = \frac{1}{n}\sum_{j=1}^{n} x_{ij}$ is the global cluster center.

Remark 1. The GOS algorithm does not guarantee a global optimum. Providing a good initial guess for clustered data is necessary to ensure that we obtain a good (in a sense that it is better than other heuristic approaches) result. A good initial guess will depend on the nature of the data being clustered. Therefore, we suggest implementing a pre-clustering strategy which will incorporate any available knowledge of the data.

3.2. Capturing the local dynamics

In this stage, we apply DMDc to each cluster and compute the low-dimensional representations of **A** and **B** matrices respectively. We first compute SVD of the augmented and the shifted snapshot sequences respectively, $\Omega^k = \hat{\mathbf{U}}^k \hat{\mathbf{\Sigma}}^k \hat{\mathbf{V}}^{k^*}$ and $\mathbf{X}_2^k = \tilde{\mathbf{U}}^k \tilde{\mathbf{\Sigma}}^k \tilde{\mathbf{V}}^{k^*}$, where k denotes the current cluster number. Next, using the transformation basis $\tilde{\mathbf{U}}^k$, i.e., the left singular vectors of the shifted snapshot sequence, we apply Eq. (12) to identify the input-to-output relationship within each cluster. The algorithm detailing the application of the proposed local DMDc method is presented below.

Algorithm 1. Nonlinear model order-reduction using temporally local dynamic modes

$$\bigcup_{k=1}^{m} \mathcal{C}^k = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$$

1: Collect the state and input snapshots and construct the data matrices, $\mathbf{X}_1 = [\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_{m-1}]$, $\mathbf{X}_2 = [\mathbf{x}_2\, \mathbf{x}_3 \ldots \mathbf{x}_m]$ and $\mathbf{\Gamma} = [\mathbf{u}_1\, \mathbf{u}_2 \ldots \mathbf{u}_{m-1}]$.

2: Apply the GOS algorithm to generate clusters and assign the snapshots to each cluster. For the kth cluster, compute the cluster centers $\mathbf{c}_k$ and the optimal number of clusters m that minimizes the clustering balance ϵ. The clustered snapshots $\mathcal{C}^k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_{n^k}^k\}$ for $k = 1, \ldots, m$, where n^k denotes the number of snapshots in the kth cluster, will satisfy:
$\bigcup_{k=1}^{m} \mathcal{C}^k = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

3: For every cluster, stack the state and input matrices to construct the augmented matrix $\mathbf{\Omega}^k$
$\Omega^k = \begin{bmatrix} \mathbf{X}_1^k \\ \mathbf{\Gamma}^k \end{bmatrix}$

4: Compute the reduced SVD of the augmented matrix and the shifted snapshot sequence. Choose values for the reduced-orders p^k and r^k based on the singular values and the corresponding singular vectors to obtain
$\Omega^k = \hat{\mathbf{U}}^k \hat{\Sigma}^k \hat{\mathbf{V}}^{k^*}$
$\mathbf{X}_2^k = \tilde{\mathbf{U}}^k \tilde{\mathbf{\Sigma}}^k \tilde{\mathbf{V}}^{k^*}$

5: Compute the low-dimensional representations of the operators $\mathbf{A}^k$ and $\mathbf{B}^k$ as described in Eq. (12)
$\hat{\mathbf{A}}^k = \tilde{\mathbf{U}}^{k^*} \mathbf{X}_2^k \hat{\mathbf{V}}^k \hat{\mathbf{\Sigma}}^{k^{-1}} \hat{\mathbf{U}}_1^{k^*} \tilde{\mathbf{U}}^k$
$\hat{\mathbf{B}}^k = \tilde{\mathbf{U}}^{k^*} \mathbf{X}_2^k \hat{\mathbf{V}}^k \hat{\mathbf{\Sigma}}^{k^{-1}} \hat{\mathbf{U}}_2^{k^*}$

Remark 2. Recall, the fundamental form of dynamic mode decomposition with control is as follows:

$$\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i \tag{16}$$

The above equation is analogous to a general nth order discrete-time linear state-space model. Hence, we recognize that the DMDc method can be used for system identification of a linear time-invariant state-space model that captures the underlying dynamics in the data obtained from high-fidelity models.

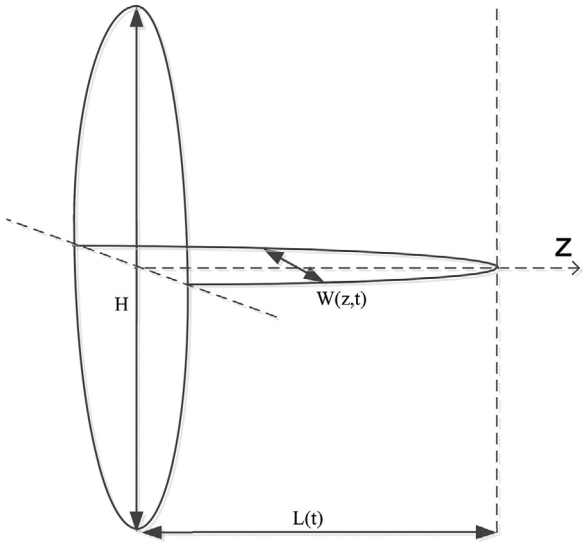


Fig. 1. The PKN fracture model considered in this work (Perkins and Kern, 1961; Nordgren, 1972).

4. Application of local DMDc based MPC to hydraulic fracturing

Hydraulic fracturing is a process of introducing water and viscosifying agent such as proppant (e.g., sand or ceramic materials) to create fractures in a rock formation. This process will create a highly-conductive channel through which oil can be produced from the reservoir. Therefore, producing fractures with the desired geometry and uniform proppant concentration at the end of pumping is essential to enhance the productivity of stimulated wells. In this section, we applied the proposed local DMDc methodology to develop an approximate model to design a feedback control system for the computation of optimal pumping schedules to accomplish the two objectives simultaneously.

4.1. Dynamic modeling of hydraulic fracturing systems

We consider a fracturing process characterized by the time-dependent spatial domain, adopting the following standard assumptions: (1) the fractures are confined vertically within a single horizontal rock layer due to sufficiently large stresses of the formation layers above and below the fracture; (2) the rock properties remain constant with respect to time and space; (3) the fracture length is much greater than its width, and thus, the fluid pressure across the vertical direction remains constant; and (4) the fracture propagation is described using the Perkins, Kern, and Nordgren (PKN) model which is shown in Fig. 1.

The dynamic modeling of hydraulic fracturing processes involves fracture propagation and proppant transport. A brief description of the equations governing the fracture propagation is presented below. We assume that the fluid is incompressible and the sectional area of the fracture is elliptical. By taking into account the fracture volume changes and fluid leak-off into the reservoir, the continuity equation is given by:

$$\frac{\partial A}{\partial t} + \frac{\partial q_z}{\partial z} + HU = 0 \tag{17}$$

where $A = \pi WH/4$ is the cross-sectional area of the elliptic fracture, H is the fracture height, $z \in [0, L(t)]$ is the time-dependent spatial coordinate in the lateral direction, $W(z, t)$ is the fracture width, $U(t)$ is the fluid leak-off rate and $q_z(z, t)$ is the local flow rate in the horizontal direction, which can be related to the fracture width using the lubrication theory and the elasticity equation as:

$$q_z = -\frac{\pi E W^3}{128\mu(1-\nu^2)}\frac{\partial W}{\partial z} \tag{18}$$

where E is the Young's modulus of the formation, μ is the fracture fluid viscosity, and ν is the Poisson ratio of the formation. The model for the fluid leak-off rate per unit height during fracture propagation is given by (Howard and Fast, 1957; Economides and Nolte, 2000)

$$U = \frac{2C_{\text{leak}}}{\sqrt{t-\tau(z)}} \tag{19}$$

where C_{leak} is the overall leak-off coefficient, t is the elapsed time since fracturing was initiated, and $\tau(z)$ is the time at which the fracture has arrived at the spatial coordinate z for the first time. Plugging Eq. (18) into Eq. (17) results in the following partial differential equation:

$$\frac{\pi H}{4}\frac{\partial W}{\partial t} - \frac{\pi E}{128\mu(1-\nu^2)}\left[3W^2\left(\frac{\partial W}{\partial z}\right)^2 + W^3\frac{\partial^2 W}{\partial z^2}\right] + HU = 0 \tag{20}$$

At the wellbore, the flow rate q_z is specified, and at the fracture tip $L(t)$, the fracture is always closed (i.e., the width of the fracture is zero). These lead to the following two boundary conditions (Gu and Hoo, 2014, 2015):

$$q_z(0, t) = Q_0 \; W(L(t), t) = 0, \tag{21}$$

where Q_0 is the fluid injection rate at the wellbore (i.e., the manipulated input). Initially, the fracture is closed leading to the following initial condition:

$$W(z, 0) = 0 \tag{22}$$

The transport of the injected proppant is modeled assuming simultaneous advection in the lateral direction, at the velocity equal to the carrier fluid, and settling toward the fracture bottom forming a proppant bank. Additionally, the proppant particles are assumed to be sufficiently large resulting in negligible diffusivity of the solid phase. Therefore, only convective flux is considered which can be computed using,

$$\begin{aligned} &\frac{\partial(WC)}{\partial t} + \frac{\partial}{\partial z}(WCV_p) = 0 \\ &C(0, t) = C_0(t) \, and \, C(z, 0) = 0 \end{aligned} \tag{23}$$

where $C(z, t)$ is the proppant concentration inside the fracture, $C_0(t)$ is the injected proppant concentration at the wellbore.

The relationship between the velocity of an individual proppant particle, V_p, the velocity of the fluid, V, and gravitational settling velocity, V_s, is given by (Adachi et al., 2007):

$$V_p = V - (1 - C)V_s \tag{24}$$

Daneshy (1978) provided a way to compute the gravity-induced settling velocity as

$$V_s = \frac{(1-C)^2(\rho_{sd} - \rho_f)gd^2}{10^{1.82C}18\mu} \tag{25}$$

where ρ_{sd} is the proppant particle density, ρ_f is the pure fluid density, g is the gravitational acceleration constant, d is the proppant diameter, and μ is the fracture fluid viscosity where its relationship with concentration can be modeled through the following empirical expression (Barree and Conway, 1995):

$$\mu(C) = \mu_0\left(1 - \frac{C}{C_{\max}}\right)^{-\alpha} \tag{26}$$

where μ_0 is the pure fluid viscosity, α is an exponent in the range of 1.2 to 1.8, and C_{max} is the maximum theoretical concentration determined by $C_{max} = (1-\phi)\rho_{sd}$ where ϕ is the proppant bank porosity.

The evolution of proppant bank height, δ, by the settling flux is described by Gu and Hoo (2014), Novotny (1977),

$$\frac{d(\delta W)}{dt} = \frac{CV_s W}{(1-\phi)} \tag{27}$$

where there is initially no proppant bank, so the initial condition is that $\delta(z, 0) = 0$. Please note that due to dilute suspension, as long as the operation is carried out for short periods of times the proppant bank height will remain much smaller than the fracture height ($\delta \ll H$).

4.2. Numerical simulations

In this section, we solve the dynamic model of a hydraulic fracturing process described in Section 4.1. To deal with the nonlinearity and moving boundary of the high-fidelity model, arising from coupled flow, reaction and geomechanics mechanisms, a novel in-house numerical scheme previously developed by Yang et al. (2017) is used. The values of various process parameters used in our calculations are obtained from Gu and Hoo, 2014: $H = 20$ m, $\mu = 0.56$ Pa·s, $E = 5 \times 10^3$ MPa, $\nu = 0.2$, and $C_{\text{leak}} = 6.3 \times 10^{-5}$ m/s$^{1/2}$. The simulation has been carried out for $t_f = 1220$ s. A constant flow rate of $Q_0 = 0.03$ m^3/s has been used for the entire process duration. The high-order discretization scheme resulted in a total of $n = 501$ spatial points, and a total of $m = 12201$ snapshots were collected to describe the fracking process. Out of the 501 spatial locations, six were selected with uniform spacing and assumed the real-time concentration measurements are available at these locations.

Remark 3. We want to note that due to the nature of the hydraulic fracturing process, the availability of concentration measurements at many different spatial locations is not guaranteed. Therefore, in this work, we assumed the measurements can be carried out for proppant concentrations at six spatial locations across the fracture along with the fracture width at the wellbore and the fracture length.

4.3. Reduced-order model

Due to the large computational requirement of the dynamic model described in Eqs. (17)–(27), it cannot be directly used for controller design. As an alternative, we generate an appropriate linear time-invariant state-space model by applying the proposed local DMDc algorithm (within each cluster):

$$\begin{aligned} \hat{\mathbf{x}}(t_{k+1}) &= \hat{\mathbf{A}}\hat{\mathbf{x}}(t_k) + \hat{\mathbf{B}}u(t_k) \\ \hat{\mathbf{y}}(t_k) &= \mathbf{H}\hat{\mathbf{x}}(t_k) \end{aligned} \tag{28}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^{r\times 1}$ denotes the vector of states in the reduced-order model, $\hat{\mathbf{A}} \in \mathbb{R}^{r\times r}$ and $\hat{\mathbf{B}} \in \mathbb{R}^{r\times 1}$ denote the system matrices and $\hat{\mathbf{y}}(t_k)$ denotes the vector of output variables. In this work, we assume outputs are equal to the states giving $\mathbf{H} = \mathbf{I}^{r\times r}$.

For the example presented in the manuscript, we first compute the reduced-order models using the proposed local in time model order-reduction approach. Once the approximate system matrices, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$, are available we apply the inverse transformation, $\hat{\mathbf{U}}\hat{\mathbf{x}} = \mathbf{x}$, and project the obtained reduced-order model back onto the original full-order space. This provides us an approximate (full-order) linear model for the original system that is used in the design of a model based controller. Since the states lose their physical meaning in the reduced subspace, we project them back to retain their original physical meaning. Please note that all the calculations of the system matrices are still performed in the r-dimensional subspace and hence model reduction is established. In general, one can as well work in the reduced-order subspace without projecting back onto the original full-order space, given the set-point is transformed accordingly.

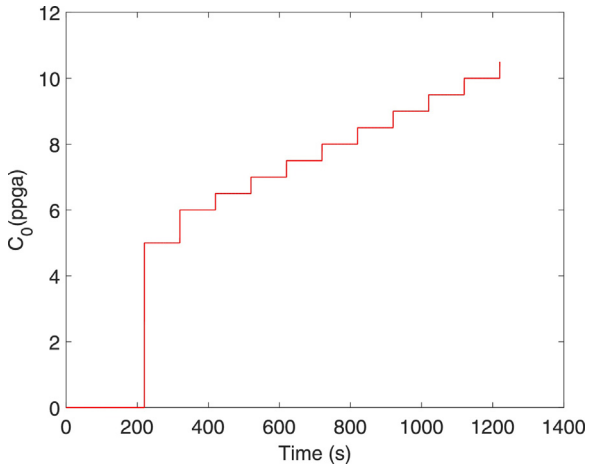


Fig. 2. The input profile applied to generate the open-loop simulation data of the process.

4.4. Comparison with global DMDc

To illustrate the superior performance of the proposed local model reduction methodology, a comparison with its global counterpart (i.e., applying DMDc over the entire time domain) has been performed. The data required for the comparison analysis has been obtained from the open-loop simulations of the hydraulic fracturing process. The test input profile has been designed to mimic the practically viable inlet proppant concentration profile in the field. Specifically, a simple uniformly increasing staircase input profile has been considered to solve the dynamics of the process, which is presented in Fig. 2. The corresponding response of the open-loop system is then obtained by solving the high-fidelity model and stored along with the manipulated input trajectory. The stored vectors then comprise the columnar entries in the data matrices defined in Section 2. The GOS algorithm, then, has been applied to the generated data to cluster the snapshot vectors based on the similarity between their local dynamic behaviors. The GOS algorithm has been implemented using the modeling language GAMS, which was interfaced with CPLEX (an MILP solver) to determine solutions to the primal and master problems. Fig. 3 shows how the clustering balance changes with the total number of clusters considered. We can predict the optimal number of clusters by selecting the turning point in the curve. The GOS algorithm determined the optimal number of clusters as $m = 100$ and the corresponding cluster configuration, $\mathcal{C}^k$, has been extracted. Once clustering has been completed, using the data matrices, the DMDc computation has been performed within each cluster to find an approximation of the system matrices. These reduced system matrices are subsequently used to compute approximate solutions to the full-order model.

Similarly, a reduced-order model has also been generated by applying global DMDc over the entire snapshot sequence, and it is compared against the full-order model. Fig. 4 illustrates one such comparison arising from the numerical realizations of the fracture width at the wellbore, fracture length and the proppant concentration at two different spatial locations. It can be observed from the plot that the approximate solution computed using the proposed local model reduction method mimics the full-order solution more

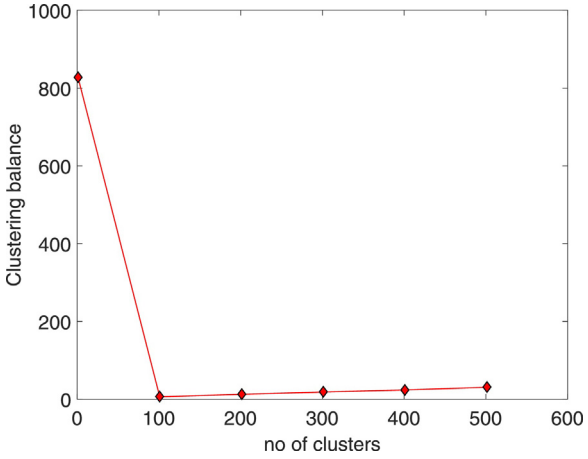


Fig. 3. Clustering balance curve predicting the optimal number of clusters, which is $m = 100$.

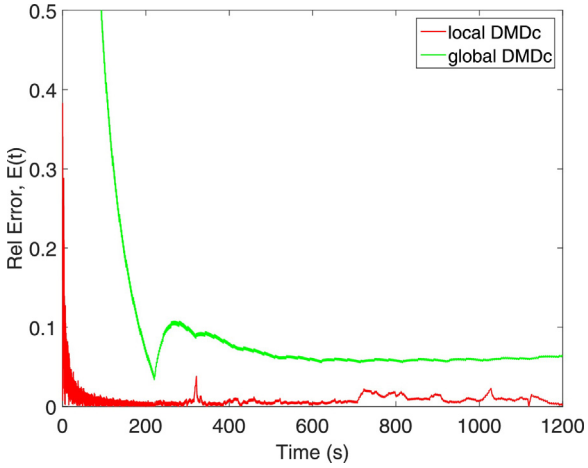


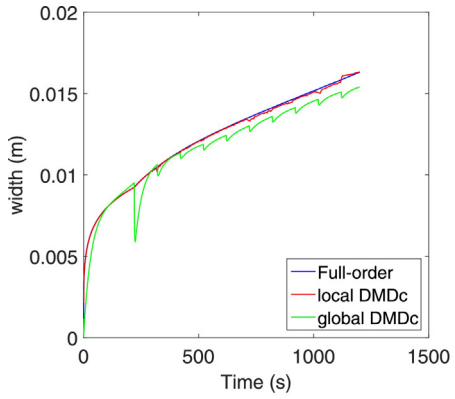
Fig. 5. Profiles of the relative error with time for approximate solutions constructed from the reduced-order models based on the proposed local and global DMDc methods.

accurately compared to the one obtained from its global counterpart.

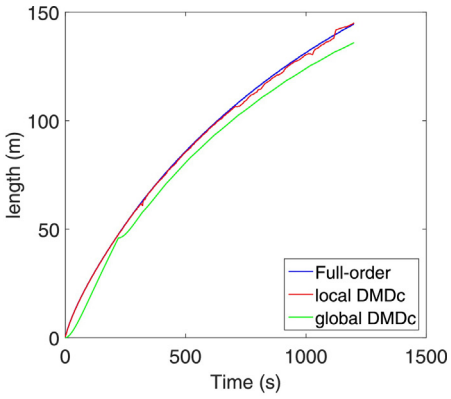
To quantify the significant performance difference between the two reduced-order models, we compared the two methods with respect to their relative errors. The relative error has been calculated using the Frobenius norms of the state vectors in the following manner:

$$E(t) = \frac{\|\mathbf{x}_{full} - \mathbf{x}_{rom}\|_{fro}}{\|\mathbf{x}_{rom}\|_{fro}} \tag{29}$$

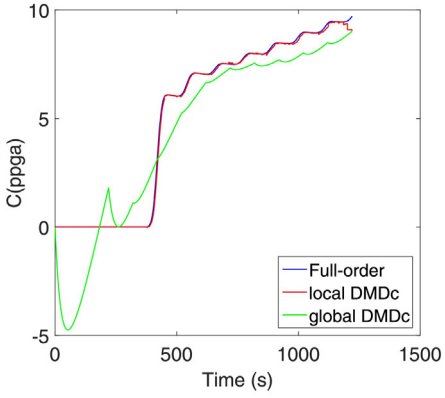
where $\|.\|_{fro}$ denotes the Frobenius norm, $\mathbf{x}_{full}$ and $\mathbf{x}_{rom}$ are state vectors obtained from the high-order discretization of the PDE and the reduced-order model, respectively. The relative error profiles for approximate solutions constructed from the local and global DMDc are presented in Fig. 5. From the plot we observe that the proposed local reduced-order model provides a precise approximation to the original solution, thus exhibiting its superior performance and warranting its application in the design of closed-loop operation of the hydraulic fracturing process.



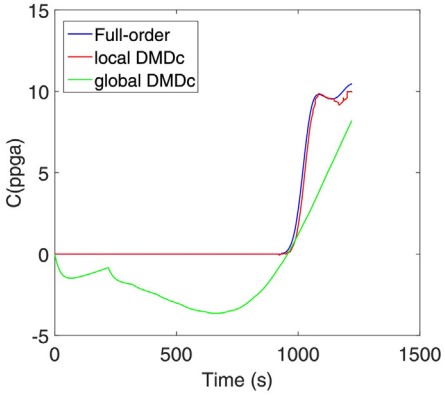
(a) W(t) at the wellbore



(b) L(t) of the fracture



(c) C(t) at z = 36 m



(d) C(t) at z = 108 m

Fig. 4. Comparison of the approximate solutions computed using local and global DMDc methods.

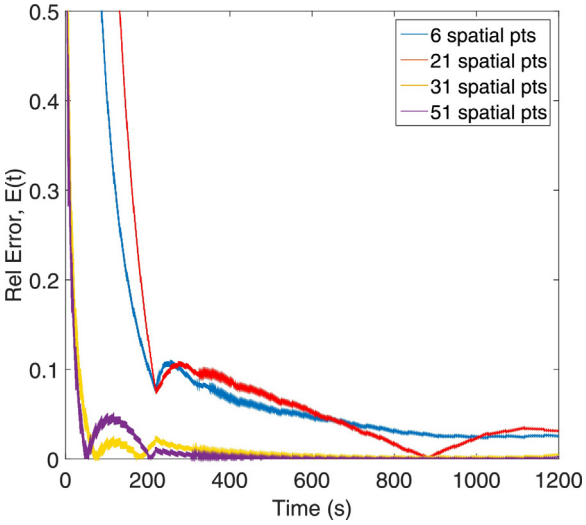


Fig. 6. Effect of the number of spatial points on the convergence of approximate solutions (to the full-order solution) computed using the global DMDc method. The relative error measures are shown.

4.5. Convergence behaviour

Before attempting to design feedback control systems, in this section, we study various factors that influence the accuracy of the reduced-order models developed based on global and local DMDc methods. Such a study is essential in gaining knowledge that will help us identify reduced-order models that accurately capture the dynamically relevant spatial behavior contained in the process data sequence. For the case of global DMDc, as the number of spatial points (i.e., the number of concentration measurements at different locations) is increased, we expect convergence of the reduced-order model towards the high-fidelity model. This is observed in Fig. 6 which shows that the relative error decreases with the number of spatial measurements.

Next, we studied the influence of clusters on the convergence behavior of the reduced-order models developed by the proposed local DMDc. Intuitively, we expect an increase in the accuracy of the approximate solutions computed using local reduced-order models with an increase in the number of clusters. In other words, as the number of partitions increases, the local dynamic behavior of the underlying system can be better captured by affine subspaces constructed by the selected dynamic modes and hence will result in more accurate approximations. A demonstration of the convergence of reduced-order models based on the proposed local DMDc methodology is presented in Fig. 7, where we see an overall decrease in the relative error with respect to an increase in the number of clusters.

Another critical factor that dictates the level of accuracy of the reduced-order models is the truncation value r in the SVD step of the DMDc algorithm. Again, one expects an increase in the r value should result in a more accurate reduced-order model. We compare the convergence behavior of the local reduced-order models (both partitioned into 100 clusters) with two different r values in Fig. 8. From the plot it is clear that the addition of the 3rd mode has greatly improved the accuracy of the low-order approximation. However, as we further increased the value of r, a "blow-up" of the numerical values occurred (i.e., infinite error). Fig. 9 shows that the magnitude of the eigenvalues of the reduced-order model describing cluster 1 is less than unity, $|\lambda| < 1$, implying that this "blow-up" is not due to the numerical instability but probably due to the inherent instability in the system alone. Please note that cluster 1 is used as a representative but this has been observed to be true for all the

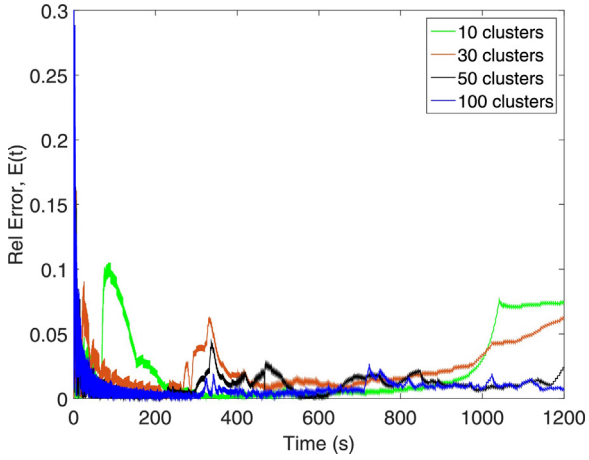


Fig. 7. Effect of the number of clusters on the convergence of approximate solutions (to the full-order solution) computed using the local DMDc method. The relative error measures are shown.

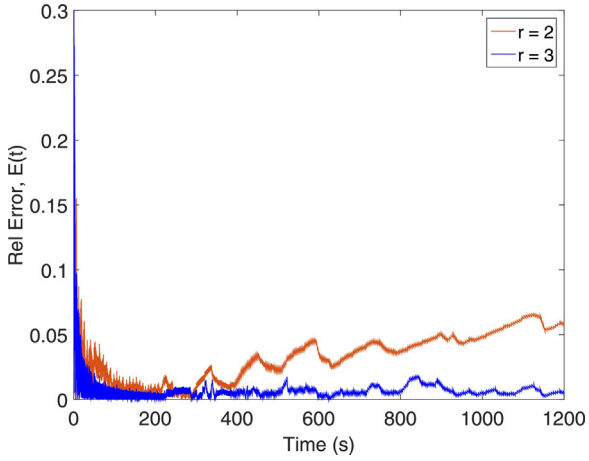


Fig. 8. Effect of the truncation value 'r' on the convergence of approximate solutions (to the full-order solution) computed using the local DMDc method. The relative error measures are shown.

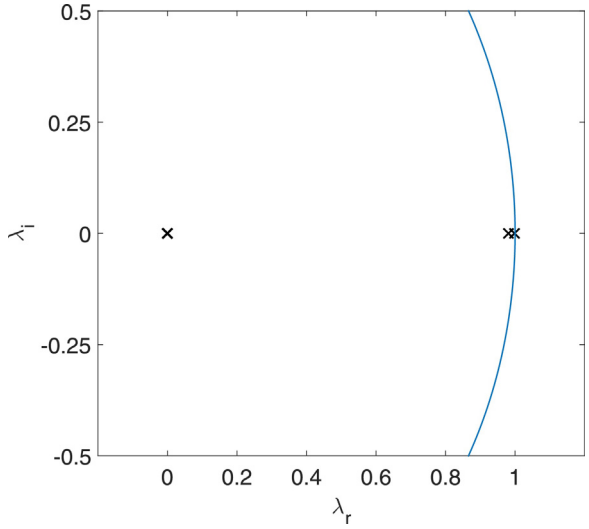


Fig. 9. The approximate eigenvalues of the system matrix $\hat{\mathbf{A}}$ describing the local dynamics in the 1st cluster, where λ_r and λ_i denote the real and imaginary parts. Note: $|\lambda| < 1$ as shown by a unit circle.

clusters. Therefore, we suspect as the r value is increased it results in the inclusion of very small singular values that in turn introduce fluctuations into the system and results in a "blow-up".

4.6. Controller design

Equipped with the results from the preceding sections, we used the local reduced-order models developed (with $m = 100$ clusters and truncation value $r = 3$) by applying the proposed methodology to design a model predictive control framework for the hydraulic fracturing process. The MPC formulation is introduced first, followed by closed-loop simulation results.

4.6.1. MPC formulation

In practice, the ultimate goal of hydraulic fracturing is to increase the productivity of a stimulated (i.e., fractured) well. The hydraulic fracturing process begins with perforation, and these initial paths are extended by injecting a high-pressure fluid (consisting of water, additives and proppant) into the perforations. After pumping is stopped and the remaining fluid leaks off to the reservoir, the fracture wall will trap the proppant inside the fracture. The trapped proppant increases fluid conductivity by creating conduits through which oil and gas can be transported easily. Therefore, achieving uniform proppant concentration across the fracture at the end of pumping, which strongly influences the propped fracture height and geometry, is paramount for effective oil and gas extraction. We formulate the following MPC problem to achieve this objective:

$$\underset{C_{0,k}}{Minimize}\,(\bar{\mathbf{C}}(t_f) - C_{target})^T Q_c(\bar{\mathbf{C}}(t_f) - C_{target}) \tag{30a}$$

$$\text{s.t}\,\bar{\mathbf{x}}(t_{k+1}) = \bar{\mathbf{A}}_i\bar{\mathbf{x}}(t_k) + \bar{\mathbf{B}}_i C_{0,k},\ \forall t_k \in \mathcal{C}^i \tag{30b}$$

$$\bar{\mathbf{y}}(t_k) = \bar{\mathbf{x}}(t_k) \tag{30c}$$

$$\bar{\mathbf{x}}(t_k) = \mathbf{x}(t_k) \tag{30d}$$

$$\mathbf{C}_{\min} \le \bar{\mathbf{C}}(t_k + j\Delta) \le \mathbf{C}_{\max},\ \forall j = 0, \ldots, 10 - k \tag{30e}$$

$$C_{0,k-1+m} \le C_{0,k+m} \le C_{0,k-1+m} + 4,\ m = 1, \ldots, 10 - k \tag{30f}$$

$$2Q_0\Delta\left(\sum_k C_{0,k}\right) = M_{prop} \tag{30g}$$

$$\bar{L}(t_f) = L_{opt},\ \bar{W}_0(t_f) \ge W_{opt} \tag{30h}$$

where $\bar{(\cdot)}$ indicates the predicted state trajectory, $\mathbf{Q}_c$ is a positive definite matrix used to compute the weighted norm, t_f denotes the total treatment time, Δ is the sampling time, t_k is the current time, $\bar{\mathbf{C}}$ is the vector of proppant concentrations at 6 locations and $C_{0,k}$ is the inlet proppant concentration (i.e., manipulated input) corresponding to kth time interval i.e., $t \in [t_k, t_{k+1})$.

In the above optimization problem, Eq. (30a) is a quadratic function which penalizes the squared deviation of the proppant concentration from the set-point at 6 different locations at the end of pumping. The predicted state trajectory $\bar{\mathbf{x}}(t)$ is computed using an approximate linear model, Eq. (30b), under the piecewise constant input profile computed by the MPC framework. This approximate model is available to us by applying the proposed local DMDc methodology and inverse transformation as detailed in Section 4.3. Since temporal clustering is performed on the process data offline, we have to use different local in time state-space models at different time instants. Therefore, the corresponding approximate model, Eq. (30b), is used to predict the state trajectory at a particular time interval in which the calculation is being performed. More specifically, at any time $t_k \in \mathcal{C}^i$, $\bar{\mathbf{A}}_i$ and $\bar{\mathbf{B}}_i$ matrices corresponding to the ith cluster (i.e., ith temporal subdomain) will be selected to predict the state trajectory. Eq. (30c) specifies that the outputs of the process are equal to the states as stated before. The initial conditions are given in Eq. (30d) which are obtained at each sampling time from the current full-state measurements. Eq. (30e) imposes limits on the concentration profiles to avoid premature termination of the process. Eq. (30f) demands a monotonic increase in the input concentration profiles with a maximum increase of 4 ppga/stage, where ppga is a concentration unit used in petroleum engineering that refers to 1 pound of proppant added to a gallon of water. Lastly, Eq. (30g) specifies the total mass of the proppant to be injected, M_{prop}, and Eq. (30h) imposes the terminal constraints on the fracture geometry at the end of pumping. Please note that the performance index in the above MPC formulation does not include a penalty term on the process inputs. In the specific example presented here, this is implicitly addressed by introducing the input constraint, Eq. (30f), specifying that the rate of change of inlet proppant concentration at the wellbore should not exceed 4 ppga/stage. Additionally, in practice, there is no need for penalizing control actions on proppant concentration, because it is not the major source of high production expenses required for hydraulic fracturing.

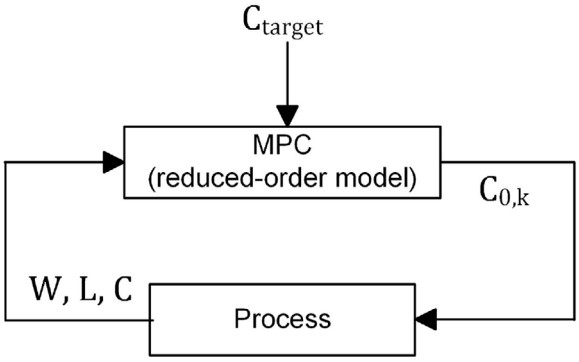


Fig. 10. Schematic representation of the closed-loop operation of hydraulic fracturing.

The MPC block diagram that describes the closed-loop operation of hydraulic fracturing processes is presented in Fig. 10. The process outputs are used as a feedback to the controller in which the developed local in time reduced-order models predict the state trajectory and compute the optimal input profile that will be applied to the process.

4.6.2. Closed-loop simulation results

The values used for various process parameters are the same as given earlier. In the closed-loop simulation, the sampling time Δ and the total time of operation t_f were chosen to be 100 s and 1220 s, respectively, during which a total of $M_{prop} = 48,000$ kg of proppant will be injected into the fracture. In the field, real-time measurements of the fracture width at the wellbore and the fracture length are available through downhole pressure analysis and microseismic monitoring techniques, respectively. This raw data will be analyzed to remove outliers and ensure they are chosen correctly. Therefore, the time required for the data to be prepared typically ranges from 1 to 5 min. Due to this reason, the sampling time was chosen to be around 2 min. During the initial stages of the hydraulic fracturing process, a high-pressure fluid (called pad) consisting mostly of water is pumped to break the rock and propagate fractures in the formation at perforated sites. The duration of this pad stage, t_p, has been fixed at 220 s to reach the desired fracture length of $L_{opt} = 135$ m without tip-screen out (premature termination of the process). The feedback control system was initialized after the injection of pad (i.e., $t_k \ge t_p$). The proppant pumping schedule was partitioned into 10 substages (i.e, $k = 1, \ldots, 10$) and the duration of each substage is given by the sampling time, Δ. We assumed that at the beginning of each substage, the full-state mea-

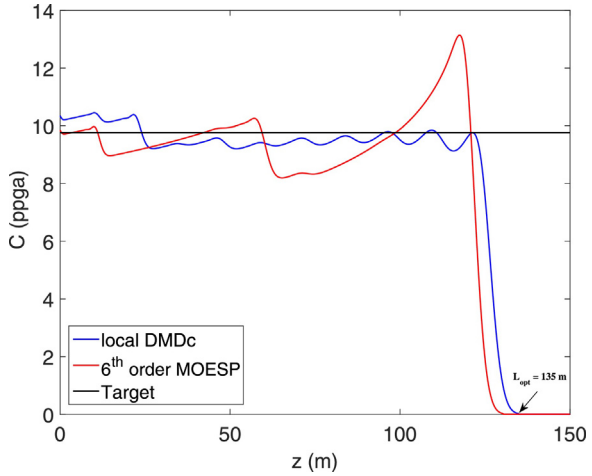


Fig. 11. Spatial profiles of the proppant concentration at the end of pumping under MPC based on the reduced-order models developed by the proposed local DMDc and the MOESP methods.

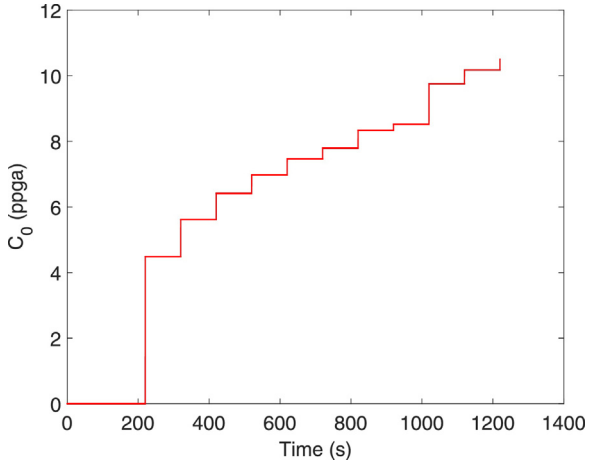


Fig. 12. Pumping schedule (inlet concentration profile) required for achieving uniform proppant concentration at the end of pumping under MPC based on the reduced-order models developed by the proposed local DMDc method.

surements of $\mathbf{x}(t_k)$ are available, which are used to predict the estimates of the future states via the developed approximate models. For illustration purposes, the approximate model for the 1st cluster computed by the proposed approach has been presented below.

$$\bar{\mathbf{A}}_1 = \begin{bmatrix} 0 & 0.00012 & 0.00059 & -0.00091 & 0.00042 & 0.00082 & -0.00220 & 0 \\ -0.00193 & 0.98919 & -0.15894 & 0.25095 & -0.11058 & -0.22025 & 0.61124 & 0 \\ 0.00061 & 0.00288 & 0.04775 & -0.07333 & 0.03343 & 0.06585 & -0.17973 & 0 \\ -0.00043 & 0.00233 & -0.07444 & 0.11438 & -0.05211 & -0.10268 & 0.28031 & 0 \\ 0.00042 & 0.00271 & 0.03331 & -0.05117 & 0.02333 & 0.04595 & -0.12539 & 0 \\ 0.00084 & 0.002945 & 0.06602 & -0.10141 & 0.04622 & 0.09106 & -0.24853 & 0 \\ -0.00231 & 0.00198 & -0.18183 & 0.27936 & -0.12729 & -0.25079 & 0.68460 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\bar{\mathbf{B}}_1 = \begin{bmatrix} 0.00088 \\ 3.56998 \\ 0.04725 \\ -0.04819 \\ 0.03557 \\ 0.06147 \\ -0.13161 \\ 0 \end{bmatrix} \quad (31)$$

The predicted state estimates are then used in the optimization problem to compute the control inputs and the corresponding process behavior that minimizes the squared deviation from its prespecified target value. The uniform target proppant concentration desired at the end of pumping is $C_{target} = 10$ ppga. Please refer to (Yang et al., 2017) for details on how to calculate this set-point value. The positive definite matrix, $\mathbf{Q}_c$, containing the weights on the penalty function of the MPC optimization problem is considered to be unity. However, one can use appropriate weights depending on the process nature. The first step, $C_{0,1}$, of the input profile, $C_{0,k}$, obtained by solving the optimization problem over a prediction horizon length of N_p was applied to the high-fidelity model in a sample-and-hold fashion, and this procedure was repeated at every sampling time until the end of treatment. In this application, we used a shrinking horizon which gives $N_p = t_f - t_k$. We solve an optimization problem to compute the control inputs while regulating the output values at the final time step, which is what we desire in this specific application.

In order to gauge the performance of the proposed model reduction method in designing a closed-loop control scheme, we compared it with MOESP, one of the well known subspace identification methods. We developed a 6th-order model by applying the MOESP algorithm to regress a linear time-invariant state-space model of the hydraulic fracturing process using the simulation results from the high-fidelity model. The developed reduced-order model was then used in the MPC framework to achieve the same control objective. Fig. 11 presents the generated spatial proppant concentration at the end of pumping. As can be observed from the plot, the controller based on the reduced-order model developed by local DMDc drives the concentration closer to the target value compared to the MOESP one, thus establishing the superiority of the proposed method (note that the order of the two reduced-order models is the same). The pumping schedule (i.e., the input concentration profile) for the corresponding process parameters and over the entire operation time considered is presented in Fig. 12.

Remark 4. Please note that due to the presence of moving boundaries and the highly nonlinear nature of the problem, an extra effort should be made in selecting the test input profile so that the overall range and trend of the test input profile is similar to the optimal input profile, especially because we are approximating the system using a linear model. Therefore, for significantly different process set-point values, the local in time reduced-order models should be recomputed carefully with different test input profiles to obtain good approximate models.

Overall, the results are very encouraging and illustrate the potential of the proposed method in approximating the prevalent dynamics and designing a feedback control system from just snapshot measurements alone, even though the entire spatiotemporal evolution data is partially captured by the measurements (the fact that we considered the concentration measurements at only 6 different locations).

5. Conclusions

In this article we proposed a method to obtain local reduced-order models from a set of solution snapshots of a nonlinear dynamic system. In this method, the solution snapshots were partitioned into the optimal number of clusters based on their local dynamic behavior by solving an MINLP problem. Then, the proposed local DMDc algorithm was applied to each cluster to extract a set of approximate linear models. In the simulation part, the developed local model reduction technique was applied to a hydraulic fracturing process governed by a set of highly-coupled nonlinear parabolic PDEs with a time-dependent spatial domain. The proposed method was compared with a system identification technique, MOESP, with respect to the accuracy of the developed reduced-order models (of same order) in designing a feedback controller. The results demonstrated that the proposed method is able to generate reduced-order models that can be used to represent high-dimensional complex data.

References

Adachi, J., Siebrits, E., Pierce, A., Desroches, J., 2007. Computer simulation of hydraulic fractures. Int. J. Rock Mech. Min. Sci. 44, 739–757.

Anttonen, J.S.R., 2001. Techniques for Reduced Order Modeling of Aeroelastic Structures with Deforming Grids. PhD Thesis. Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.

Armaou, A., Christofides, P.D., 2001. Finite-dimensional control of nonlinear parabolic PDE systems with time-dependent spatial domains using empirical eigenfunctions. Int. J. Appl. Math. Comput. Sci 11, 287–317.

Bagheri, S., 2013. Koopman-mode decomposition of the cylinder wake. J. Fluid Mech. 726, 596–623.

Baker, J., Christofides, P.D., 2000. Finite-dimensional approximation and control of non-linear parabolic PDE systems. Int. J. Control 73, 439–456.

Barree, R., Conway, M., 1995. Experimental and numerical modeling of convective proppant transport. J. Pet. Technol. 47, 216–222.

Chen, K.K., Tu, J.H., Rowley, C.W., 2012. Variants of dynamic mode decomposition: connections between Koopman and Fourier analyses. J. Nonlinear Sci. 22, 897–915.

Christofides, P.D., Daoutidis, P., 1997. Finite-dimensional control of parabolic PDE systems using approximate inertial manifolds. J. Math. Anal. Appl. 216, 398–420.

Daneshy, A., 1978. Numerical solution of sand transport in hydraulic fracturing. J. Pet. Technol. 30, 132–140.

Dihlmann, M., Drohmann, M., Haasdonk, B., 2011. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning. In: Proceedings of the International Conference on Adaptive Modeling and Simulation, Paris, France, pp. 156–167.

Duke, D., Soria, J., Honnery, D., 2012. An error analysis of the dynamic mode decomposition. Exp. Fluids 52, 529–542.

Eckart, C., Young, G., 1936. The approximation of one matrix by another of lower rank. Psychometrika 1, 211–218.

Economides, M.J., Nolte, K.G., 2000. Reservoir Stimulation. Wiley, Chichester.

Efendiev, Y., Galvis, J., Gildin, E., 2012. Local-global multiscale model reduction for flows in highly heterogeneous media. J. Comput. Phys. 231, 8100–8113.

Efendiev, Y., Galvis, J., Wu, X., 2011. Multiscale finite element methods for high-contrast problems using local spectral basis functions. J. Comput. Phys. 230, 937–955.

Ghommem, M., Calo, V.M., Efendiev, Y., 2014. Mode decomposition methods for flows in high-contrast porous media. A Global approach. J. Comput. Phys. 257, 400–413.

Ghommem, M., Presho, M., Calo, V.M., Efendiev, Y., 2013. Mode decomposition methods for flows in high-contrast porous media. Global-local approach. J. Comput. Phys. 253, 226–238.

Golub, G.H., Reinsch, C., 1970. Singular value decomposition and least squares solutions. Numer. Math. 14, 403–420.

Goulart, P.J., Wynn, A., Pearson, D.S., 2012. Optimal mode decomposition for high dimensional systems. In: Proceedings of the 51st IEEE Conference on Decision and Control, Maui, Hawaii, pp. 4965–4970.

Gu, Q., Hoo, K.A., 2014. Evaluating the performance of a fracturing treatment design. Ind. Eng. Chem. Res. 53, 10491–10503.

Gu, Q., Hoo, K.A., 2015. Model-based closed-loop control of the hydraulic fracturing process. Ind. Eng. Chem. Res. 54, 1585–1594.

Holmes, P., Lumley, J.L., Berkooz, G., 1996. Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge University Press, New York.

Howard, G.C., Fast, C.R., 1957. Optimum fluid characteristics for fracture extension. Drill. Prod. Pract. 24, 261–270.

Izadi, M., Dubljevic, S., 2013. Order-reduction of parabolic PDEs with time-varying domain using empirical eigenfunctions. AIChE J. 59, 4142–4150.

Jovanović, M.R., Schmid, P.J., Nichols, J.W., 2014. Sparsity-promoting dynamic mode decomposition. Phys. Fluids 26, 024103.

Juang, J.N., Phan, M., Horta, L.G., Longman, R.W., 1991. Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments. Technical Memorandum 104069. NASA.

Katayama, T., 2005. Subspace Methods for System Identification. Springer-Verlag, London.

Lusseyran, F., Gueniat, F., Basley, J., Douay, C.L., Pastur, L.R., Faure, T.M., Schmid, P.J., 2011. Flow coherent structures and frequency signature: application of the dynamic modes decomposition to open cavity flow. J. Phys. Conf. Ser. 318, 042036.

Mezić, I., 2013. Analysis of fluid flows via spectral properties of the Koopman operator. Ann. Rev. Fluid Mech. 45, 357–378.

Mirsky, L., 1960. Symmetric gauge functions and unitarily invariant norms. Q. J. Math. Oxf. 11, 50–59.

Mizuno, Y., Duke, D., Atkinson, C., Soria, J., 2011. Investigation of wall-bounded turbulent flow using dynamic mode decomposition. J. Phys. Conf. Ser. 318, 042040.

Muld, T.W., Efraimsson, G., Henningson, D.S., 2012. Flow structures around high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. Comput. Struct. 57, 87–97.

Narasingam, A., Siddhamshetty, P., Kwon, J.S., 2017. Temporal clustering for order reduction of nonlinear parabolic PDE systems with time-dependent spatial domains: application to a hydraulic fracturing process. AIChE J., http://dx.doi.org/10.1002/aic.15733 (in press).

Noack, B.R., Afanasiev, K., Morzyński, M., Tandmor, G., Thiele, F., 2003. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. J. Fluid Mech. 497, 335–363.

Noack, B.R., Schlegel, M., Ahlborn, B., Mutschke, B., Morzyński, M., Comte, P., Tadmor, G., 2008. A finite-time thermodynamics formalism for unsteady flows. J. Non-Equilib. Thermodyn. 33, 103–148.

Nordgren, R., 1972. Propagation of a vertical hydraulic fracture. Soc. Pet. Eng. J. 12, 306–314.

Novotny, E.J., 1977. Proppant transport. In: Proceedings of the 52nd SPE Annual Technical Conference and Exhibition, Denver, CO. (SPE 6813).

Pan, C., Yu, D., Wang, J., 2011. Dynamical mode decomposition of gurney flap wake flow. Theor. Appl. Mech. Lett. 1, 012002.

Perkins, T.K., Kern, L.R., 1961. Widths of hydraulic fractures. J. Pet. Technol. 13, 937–949.

Proctor, J.L., Brunton, S.L., Kutz, J.N., 2016. Dynamic mode decomposition with control. SIAM J. Appl. Dyn. Syst. 15, 142–161.

Qin, J., 2006. An overview of subspace identification. Comput. Chem. Eng. 30, 1502–1513.

Rowley, C.W., Mezić, I., Bagheri, S., Schlatter, P., Henningson, D.S., 2009. Spectral analysis of nonlinear flows. J. Fluid Mech. 641, 115–127.

Schmid, P.J., 2009. Dynamic mode decomposition of experimental data. In: 8th International Symposium on Particle Image Velocimetry – PIV09.

Schmid, P.J., 2010. Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. 656, 5–28.

Schmid, P.J., 2011. Application of the dynamic mode decomposition to experimental data. Exp. Fluids 50, 1123–1130.

Schmid, P.J., Li, L., Juniper, M.P., Pust, O., 2011. Applications of the dynamic mode decomposition. Theor. Comput. Fluid Dyn. 25, 249–259.

Schmid, P.J., Meyer, K.E., Pust, O., 2009. Dynamic mode decomposition and proper orthogonal decomposition of flow in a lid-driven cylindrical cavity. In: 8th International Symposium on Particle Image Velocimetry – PIV09.

Schmid, P.J., Sesterhenn, J., 2008. Dynamic mode decomposition of numerical and experimental data. In: Bull. Amer. Phys. Soc. 61st APS Meeting, San Antonio, TX, p. 208.

Seena, A., Sung, H.J., 2011. Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations. Int. J. Heat Fluid Flow 32, 1098–1110.

Semeraro, O., Bellani, G., Lundell, F., 2012. Analysis of time-resolved PIV measurements of a confined turbulent jet using POD and Koopman modes. Exp. Fluids 53, 1203–1220.

Tan, M.P., Broach, J.R., Floudas, C.A., 2007. A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning. J. Glob. Optim. 39, 323–346.

Tu, J.H., Luchtenburg, D.M., Rowley, C.W., 2014. On dynamic mode decomposition: theory and applications. J. Comput. Dyn. 1, 391–421.

Van Overschee, P., De Moor, B., 1994. N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica 30, 75–93.

Van Overschee, P., De Moor, B., 1996. Subspace Identification for Linear Systems: Theory – Implementation – Applications. Springer, New York.

Williams, M.O., Kevrekidis, I.G., Rowley, C.W., 2015. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. J. Nonlinear Sci. 25, 1307–1346.

Wynn, A., Pearson, D.S., Ganapathisubramani, B., Goulart, P.J., 2013. Optimal mode decomposition for unsteady flows. J. Fluid Mech. 733, 473–503.

Yang, S., Siddhamshetty, P., Kwon, J.S., 2017. Optimal pumping schedule design to achieve a uniform proppant concentration level in hydraulic fracturing. Comput. Chem. Eng. 101, 138–147.