2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Hybrid Model Reduction for Compressible Flow Controller Design

Xiaoqing Ge* and John T. Wen
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract— **Computational fluid dynamics (CFD) has been a powerful simulation tool to gain insight and understanding of fluid dynamic systems. However, it is also extremely computationally intensive and thus unsuitable for control design and iteration. Various model reduction schemes have been proposed in the past to approximate the Navier-Stokes equation with a low-dimensional model. There are essentially two approaches: input/output model identification and proper orthogonal decomposition (POD). The former captures mostly the local behavior near a steady state and the latter is highly dependent on the snapshots of the flow state used to extract the projection. This paper presents a hybrid model reduction approach that attempts to combine the best features of the two approaches. We first identify an input/output linear model by using the subspace identification method. We next project the difference between CFD response and the identified model response onto a set of POD basis. This trajectory is then fit to a nonlinear dynamical model to augment the input/output linear model. The resulting hybrid model is then used for control design with the controller evaluated with CFD. The proposed methodology has been applied to a 2D compressible flow passing a contraction geometry. The result indicates that near the steady state used for linear system identification, the linear system based design works well. However, far away from the steady state, the hybrid system shows much better performance.**

I. Introduction

Efficient numerical techniques are now commonly used to efficiently simulate physical systems governed by partial differential equations, such as fluid, thermal, and structural systems. Due to the high dimensionality of such systems, the simulation is computationally expensive and time consuming. For system analysis and control design, low-dimensional models that can capture key attributes of the high order systems are needed.

A commonly used tool for complex nonlinear system model reduction is the method of proper orthogonal decomposition (POD) and Galerkin projection [1]–[3]. It provides a systematic way to develop reduced order models from a series of "snapshots" of experimental or computational data. Through an orthogonalization procedure, characteristic information of the system is extracted from the solution snapshots and used to construct a reduced basis set. A low-dimensional approximation is obtained by projecting the full nonlinear system to this reduced basis. The attraction of the POD/Galerkin method is that it can capture nonlinear behavior of the system (if appropriate snapshots are used), but may lose key system properties such as stability near a stable equilibrium [4]. Some extended approaches have been proposed based on the POD method. [5] establish a connection between POD and balanced truncation theory. The method of snapshots is used to obtain low-rank approximations to the system controllability and observability grammians, which are then used to develop a balanced reduced order model. A trust-region POD (TRPOD) method is proposed in [6]. By embedding the POD based reduced order modeling technique into a trust-region framework, this approach updates the reduced order models to represent the flow dynamics during an optimization process.

For linear time invariant systems, a rich set of tools exist, mostly based on the singular value decomposition (SVD) of some input/output map. In particular, if only input/output data are available, such tools, e.g., subspace algorithm, may be used to construct a low order dynamical model [7]–[9]. This type of tool is attractive as it usually preserves system stability and has explicit error bounds. Furthermore, it does not require the knowledge of the governing equation. However, it is unable to predict the nonlinear behavior of the system nor provide insight into the flow field.

One way of modeling the dynamics of a nonlinear system based on input/output data is the nonlinear autoregressive moving average model with exogenous inputs (NARMAX) method. In [10], NARMAX is employed to develop a nonlinear model representing the coupling between jet actuation and measured pressure for a boundary layer flow separation control problem. However, this method only generates an input/output model, thus providing no information about the internal system dynamics.

This paper presents a reduced order modeling technique we refer to as *hybrid* method, which combines ideas from subspace-based system identification and POD method. It first identifies a stable linear dynamical model. Then it represents the difference between the full flow field and the identified model by a POD expansion. Thus, the full system is decomposed into a dominant linear subsystem and a small-scale nonlinear subsystem. The dynamics of the nonlinear subsystem can be obtained by projecting the difference between linear model and CFD data onto the POD modes. We apply the hybrid approach to a 2D compressible flow that passes a contraction geometry and compare it with three other modeling methods. Computation fluid dynamics (CFD) results suggest that the hybrid model accurately predicts the nonlinear behavior of the system. It also performs the best in terms of feedback control and learning control.

* Corresponding author: gex@rpi.edu

II. Contraction Section Example

Fig. 1 shows the geometry of the contraction section in a subsonic wind tunnel facility that motivated this research [11]. The contraction, with boundary shaped by a fifth order polynomial curvature, settles the flow transitions from the settling chamber and feeds uniform flow to the test section. In computational simulations, throat Mach number is attained to precisely match the experimental setup by adjusting the mass flux rate at the inlet. The adjustment can be formulated as a classical control problem: first model the dynamics from the inlet mass flux to the throat Mach number, then design control strategies to achieve a desired output.

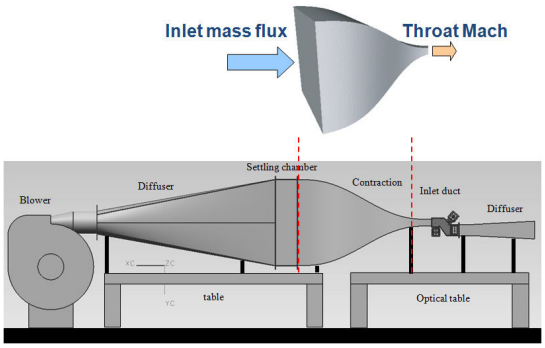


Fig. 1. Contraction geometry

The current work uses PHASTA, a numerical CFD code, to simulate the flow dynamics on a multiprocessor UNIX cluster. Flow computations are performed using the Streamline Upwind Petrov-Galerkin (SUPG) stabilized finite element method [12], which has been proven stable and higher order accurate. The numerical simulation uses a 2D slab geometry which approximates the flow in a $40mm$ slice along the span of the contraction. Full flow field data are collected at a sampling rate of $5 \times 10^{-4}s$, based on a fine 2D mesh.

III. Model Reduction

A. POD and Galerkin Projection

The essential idea of POD is to generate a set of basis that "optimally " spans a given sample set of data. Consider a Hilbert space H with an inner product $\langle \cdot, \cdot \rangle$. The goal of POD is, given an ensemble of data $y(t) \in H$, to find an optimal subspace S, such that the averaged error of the projection $E(\|y - P_S y\|)$ is minimized. Here $\|\cdot\|$ denotes the norm on H, P_S denotes the projection onto the subspace S, and E is an average operator over t. Let $\{\varphi_j \in H | j = 1, \cdots, n\}$ be an orthogonal basis for S, then the projection of $y(t)$ onto S is

$$P_S y(t) = \sum_{j=1}^{n} \langle y(t), \varphi_j \rangle \varphi_j. \tag{1}$$

The basis functions φ can be obtained directly from the SVD of the snapshot matrix $[y(t_1), \cdots, y(t_N)]$.

Suppose the dynamics of a system are described by a high order ODE (or PDE) of the form:

$$\dot{y} = D_x(y), \tag{2}$$

where D_x is a nonlinear differential operator depending on some parameter x. We may approximate solutions to (2) by projecting the equations onto a finite-dimensional subspace such as the one spanned by POD modes. Inserting the POD expansion

$$y(t) = \sum_{j=1}^{n} \alpha_j(t) \varphi_j, \tag{3}$$

into (2) and taking the inner product with φ_k yield a set of ODEs for α_k:

$$\dot{\alpha_k} = \langle D_x(y), \varphi_k \rangle, \qquad k = 1, \cdots, n. \tag{4}$$

B. Hybrid Model Reduction Method

The first step is to use the input/output CFD data (in the contraction section example, inlet mass flux is the input u_k, throat Mach number is the output z_k) to identify a linear time invariant system (e.g., using the subspace identification method).

$$\begin{aligned} x_{k+1} &= A_{11} x_k + B_1 u_k \\ z_k &= C_1 x_k. \end{aligned} \tag{5}$$

We next try to relate the state x_k of this (typically low order) system to the full flow field. Denote the k^{th} snapshot of the full flow field by Y_k. A mapping Φ from the reduced state x_k to the full state Y_k may be obtained by a least-squares fit:

$$\Phi \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix} = \begin{bmatrix} Y_1 & \dots & Y_N \end{bmatrix}. \tag{6}$$

The linear model can approximate well the local behavior (near the steady state), but has increasing error far away from the steady state. To capture the nonlinear behavior, we approximate the difference between Y_k and Φx_k as a linear combination of POD modes Ψ:

$$Y_k = \Phi x_k + \Psi a_k = \sum_{i=1}^{n_1} \varphi_i x_k^i + \sum_{j=1}^{n_2} \psi_j a_k^j, \tag{7}$$

where Ψ is obtained from the SVD of the difference snapshot matrix $[Y_1 - \Phi x_1, \cdots, Y_N - \Phi x_N]$.

Traditional POD/Galerkin method finds the dynamics of a_k by projecting the governing equation onto the POD basis. The projection is computational expensive when there is a large number of sampling points. Moreover, the result is highly dependent on the choice of inner product for projection. One needs to take care of different units of flow variables (pressure, velocity, temperature, etc.), as well as the weighting ratio of each point if the mesh is nonuniform. Therefore, instead of projecting the governing equation, we directly compute a_k by projecting the linear system error onto the POD basis:

$$a_k = \Psi^T (Y_k - \Phi x_k). \tag{8}$$

Then we use $[a_1, \cdots, a_N]$ as a "training" data set to fit a nonlinear discrete dynamical system

$$a_{k+1} = f_a(x_k, a_k, u_k) \tag{9}$$

to approximate the dynamics of a_k. Combining (5), (7) and (9), we have a hybrid dynamical model:

$$\begin{aligned} x_{k+1} &= A_{11}x_k + B_1 u_k \\ a_{k+1} &= f_a(x_k, a_k, u_k) \\ z_k &= g(Y_k) = g(\Phi x_k + \Psi a_k) := h(x_k, a_k) \end{aligned} \quad (10)$$

where $g(\cdot)$ represents a nonlinear output function.

C. Application to Contraction Section Example

We have applied the hybrid model reduction method described above to the flow modeling of the contraction section. Using the input/output data, a second-order linear model of the form (5) is identified by using the subspace algorithm. As shown in Fig. 2, it captures well the input/output relationship for the training data set. The linear model is also validated by a different data set, where the input range is much larger than the training data. We shall see later that when the input level is far from the training set, the linear model yields a poor prediction of the throat Mach number.

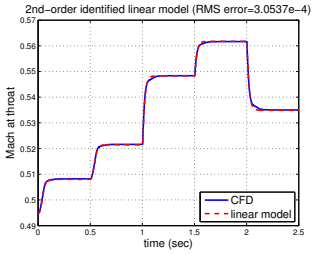


Fig. 2. Output response comparison of the training set between CFD and linear model (input range: $[0.7, 0.8]$)

We next use the POD expansion to represent the difference between CFD data and the linear model. Fig. 3 plots the energy captured by the first 10 POD modes. In the expansion we use the first 4 modes, which capture more than 99% of the total energy.

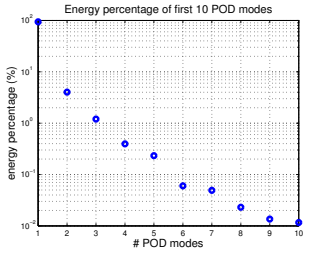
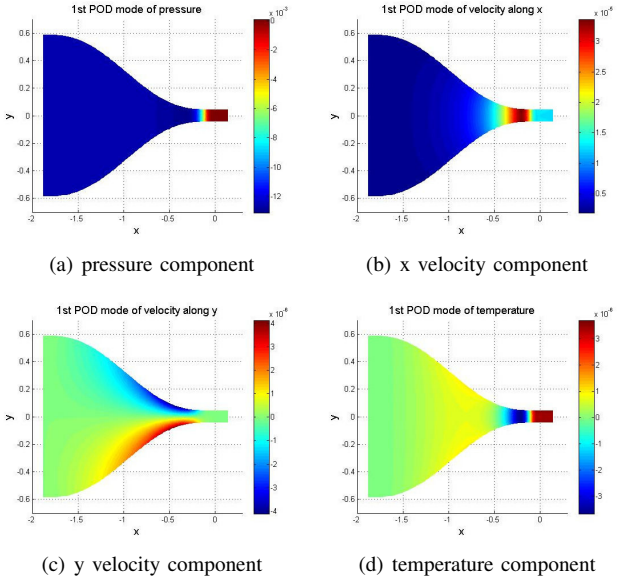


Fig. 3. Energy captured by the first 10 POD modes

We denote the i^{th} power of a vector y_k by $(y_k)^i = [y_{k,1}^i, \cdots, y_{k,n}^i]^T$, where $y_{k,j}, j = 1, \cdots, n$ is the j^{th} element of y_k. The dynamics of a_k is fit to a nonlinear function of the form:

$$\begin{aligned} a_{k+1} &\approx \hat{f}_a(x_k, a_k, u_k) \\ &= \sum_{i=1}^{m_1} \Gamma_i (x_k)^i + \sum_{i=1}^{m_2} \Lambda_i (a_k)^i + \sum_{i=1}^{m_3} \beta_i (u_k)^i. \end{aligned} \quad (11)$$



(a) pressure component (b) x velocity component

(c) y velocity component (d) temperature component

Fig. 4. First POD mode

Let n_a, n_x denote the dimension of x_k, a_k, respectively. Then Γ_i is a $n_a \times n_x$ matrix and Λ_i is a $n_a \times n_a$ matrix. Here β_k is a $n_a \times 1$ vector since u_k is a scalar. If $\hat{f}_a(\cdot)$ has a large number of terms that increases the computational cost of the approximation, one can extract dominant basis by performing SVD of

$$\begin{bmatrix} x_1 & \cdots & x_N \\ \vdots & \ddots & \vdots \\ (x_1)^{m_1} & \cdots & (x_N)^{m_1} \\ a_1 & \cdots & a_N \\ \vdots & \ddots & \vdots \\ (a_1)^{m_2} & \cdots & (a_N)^{m_2} \\ u_1 & \cdots & u_N \\ \vdots & \ddots & \vdots \\ (u_1)^{m_3} & \cdots & (u_N)^{m_3} \end{bmatrix}. \quad (12)$$

Fig. 5 shows a good approximation of $f_a(x_k, a_k, u_k)$ with $m_1 = 4$, $m_2 = 1$, $m_3 = 1$, which yields a hybrid model with state nonlinearity and output nonlinearity:

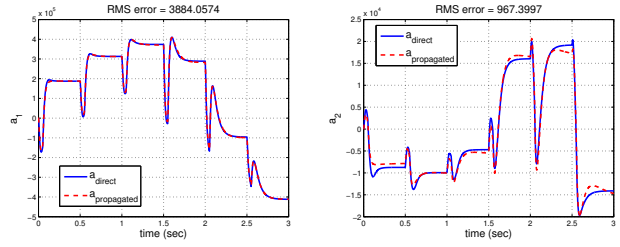
$$\begin{aligned} x_{k+1} &= A_{11}x_k + B_1 u_k \\ a_{k+1} &= A_{21}x_k + A_{22}a_k + B_2 u_k + \sum_{i=2}^{m_1} \Gamma_i (x_k)^i \\ z_k &= h(x_k, a_k). \end{aligned} \quad (13)$$

The hybrid model accurately predicts the throat Mach number (Fig. 6) for a wide input range.

IV. Control Design for Output Regulation

The goal of obtaining a reduced model is to use it for control design. In this section, we design LQG control and learning control for the inlet mass flux to regulate the output throat Mach number based on the hybrid model and compare its performance with three other reduced order models:

1) Identified linear model (5).



(a) a_1 (b) a_2

(c) a_3 (d) a_4

Fig. 5. Comparison between projected a_k (solid) and propagated a_k (dash) using $\hat{f}_a(x_k, a_k, u_k)$

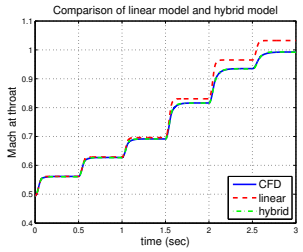


Fig. 6. Comparison of linear and hybrid models (input range: $[0.7, 1.5]$)

2) Semilinear model with input nonlinearity [13]:

$$\begin{aligned} x_{k+1} &= Ax_k + Bf(u_k) \\ z_k &= Cx_k. \end{aligned} \tag{14}$$

3) Purely nonlinear model (in the rest of the paper, we call it "nonlinear model" for simplicity): expand $Y_k = \Psi b_k$ and approximate the dynamics of b_k by a nonlinear function $f_b(\cdot)$

$$\begin{aligned} b_{k+1} &= f_b(b_k, u_k) \\ z_k &= g(Y_k) = g(\Psi b_k) := p(b_k). \end{aligned} \tag{15}$$

This method is very similar to POD method, except that it directly fits the dynamics of b_k to a nonlinear function, instead of projecting the Navier-Stokes equations to propagate b_k. This is also the same as the identification of the a_k dynamics without combining it with the linear time invariant system.

A. LQG Control

LQG regulator is a combination of a Kalman filter with an Linear-Quadratic-Regulator (LQR). It uses state estimate from the Kalman filter and optimized LQR state-feedback gain to generate the control signal. To eliminate steady-state error, an integrator is added to the LQG regulator. The extended state space realization of a linear system with an error integrator is given by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \\ \dot{e} &= r - y = r - Cx - Du \end{aligned} \tag{16}$$

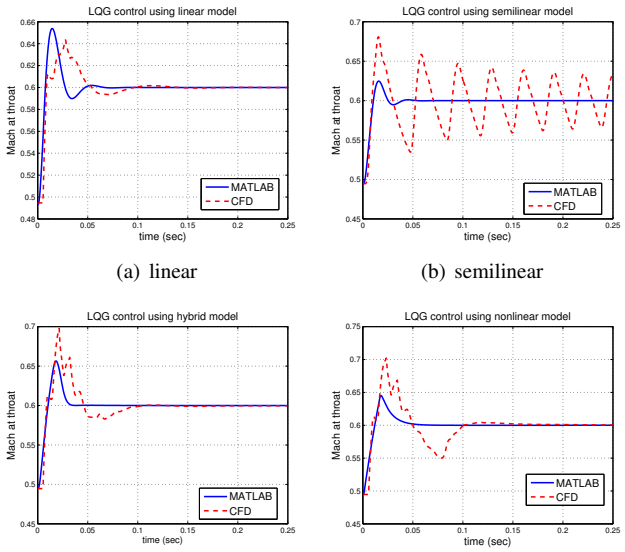
Let $\Delta x = x - x^*$, $\Delta u = u - u^*$, $\Delta y = y - y^*$, $(x^*, y^*, u^*$ are steady state values), then (16) is rewritten as

$$\begin{aligned} \dot{x}_e = \begin{bmatrix} \Delta\dot{x} \\ \dot{e} \end{bmatrix} &= \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ e \end{bmatrix} + \begin{bmatrix} B \\ -D \end{bmatrix} \Delta u \\ y_e = \begin{bmatrix} \Delta y \\ e \end{bmatrix} &= \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Delta x \\ e \end{bmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} \Delta u. \end{aligned} \tag{17}$$

The full control input is

$$u = u^* + \Delta u = u^* - K_e \hat{x}_e = u^* - [K | K_i] \begin{bmatrix} \hat{x} \\ e \end{bmatrix}, \tag{18}$$

where $\hat{x}$ represents the state estimate from a Kalman filter, K_e is computed by solving the Riccati differential equation. K_e can be further tuned to improve the system transient response by adjusting the weights on input, output and state.



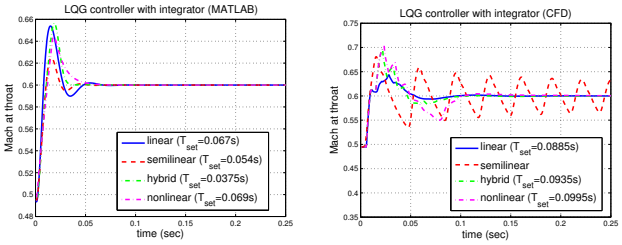
(a) linear (b) semilinear

(c) hybrid (d) nonlinear

Fig. 7. MATLAB and CFD simulation results (targe Mach number=0.6)

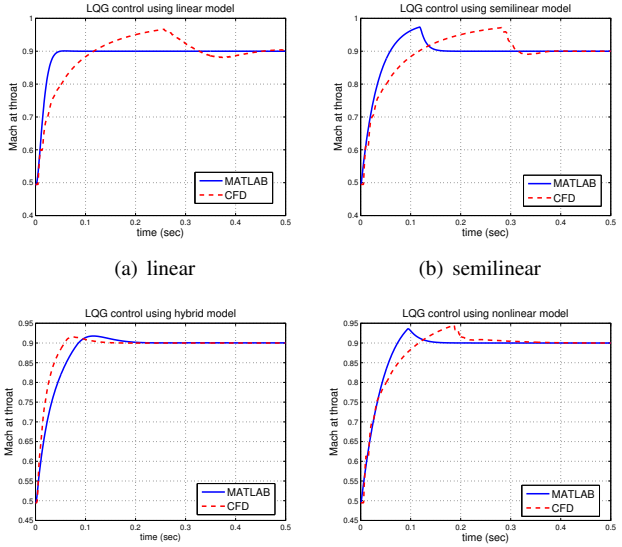
We designed an LQG regulator based on each model (linear, semilinear, hybrid, nonlinear) and tune K_e in MATLAB such that they have similar settling time. For semilinear, hybrid and nonlinear models, the control design uses linearization of the model and the state estimate $\hat{x}$ is generated by an extended Kalman filter. The Kalman filter and LQG controller are then implemented in CFD simulations.

Fig. 7 and 8 show the performances of four LQG controllers which are designed to drive the throat Mach number to 0.6. Controllers based on linear, hybrid and nonlinear models generate a similar output to MATLAB simulation,



(a) MATLAB results (b) CFD results

Fig. 8. Comparison of four models (targe Mach number=0.6)



(a) linear (b) semilinear

(c) hybrid (d) nonlinear

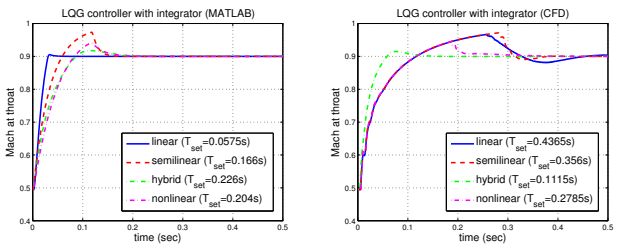
Fig. 9. MATLAB and CFD simulation results (targe Mach number=0.9)

while the semilinear model yields a very oscillatory output. We also test the four controllers with a larger target Mach number of 0.9. Although MATLAB simulations (Fig. 10 (a)) suggest that they have similar performances (linear-model-based controller performs even better), only hybrid-model-based controller settles the output within a reasonable time of $0.1115s$. The other three controllers have a much longer settling time of more than $0.27s$.

To sum up, when the input is small, linear model is good enough to predict the output as well as to guide feedback control design. However, it shows inaccuracy at high input level. Semilinear model can not provide a reasonable prediction of the output in both cases. Hybrid model is shown to be the most accurate and reliable model for a wide input range.

B. Iterative POD Approach

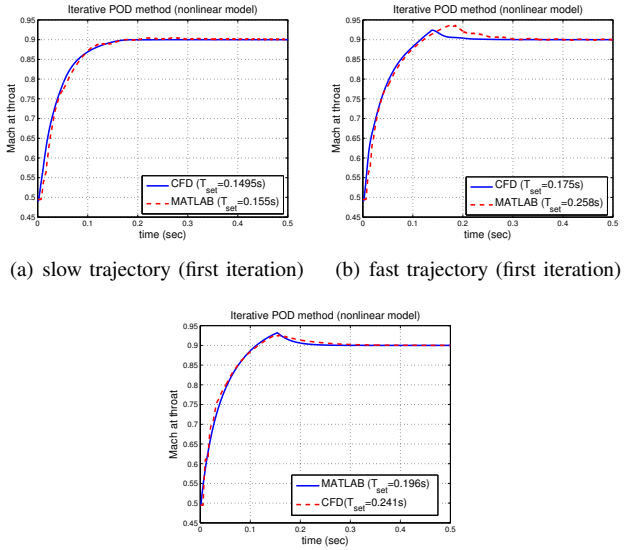
As can be seen from Fig. 7 and 9, there is still a discrepancy between MATLAB and CFD trajectories. This suggests an iterative POD approach to correct the discrepancy, based on CFD closed loop simulation results. Take the nonlinear model in Fig. 9 (d) for example, we extract POD basis from the CFD closed loop response (red line) and compute a new



(a) MATLAB results (b) CFD results

Fig. 10. Comparison of four models (targe Mach number=0.9)

set of $\hat{b}_k$ by projecting the full flow field onto the basis. We re-fit a nonlinear function that approximates the dynamics of $\hat{b}_k$ and use the new nonlinear model for Kalman filter and control design. Fig. 11 (a) and (b) show the result of using one POD iteration to improve the nonlinear model. We test two different LQG gains corresponding to a slow trajectory and a fast trajectory, respectively. Note that the LQG gains used here are re-tuned and different from that in Fig. 9 (d). It can be seen that the new model based on one POD iteration predicts the slow trajectory very well. It also shows improvement on the fast trajectory, though there is still a small discrepancy requiring further POD iteration. Fig. 11 (c) shows that through a second POD iteration, we obtain a nonlinear model that matches the CFD output very well.



(a) slow trajectory (first iteration) (b) fast trajectory (first iteration)

(c) fast trajectory (second iteration)

Fig. 11. Iterative POD approach applied to the nonlinear model

An alternative and more efficient way of performing iterative POD method is to augment existing POD basis based on predicted error, instead of regenerating a new set of POD basis in each iteration. Then the full flow state

$$Y_k = \Phi x_k + \sum_{\ell=1}^{n_\ell} \Psi^\ell a_k^\ell. \tag{19}$$

In ℓ^{th} iteration, Ψ^ℓ is generated from SVD of $Y_{CFD} - \Phi x_k - \sum_{i=1}^{\ell-1} \Psi^i a_k^i$, where Y_{CFD} is the true flow state from CFD simulation. This gives us a cascade system

$$\begin{aligned} x_{k+1} &= A_{11}x_k + B_1 u_k \\ a_{k+1}^1 &= f_1(x_k, a_k^1, u_k) \\ &\vdots \\ a_{k+1}^\ell &= f_\ell(x_k, a_k^1, \cdots, a_k^\ell, u_k). \end{aligned} \tag{20}$$

Based on the performance of the hybrid model, a well fitting nonlinear function f_a can significantly improve the linear model. Therefore, as long as we can find $f_1, ..., f_\ell$ that fit the dynamics of $a^1, \cdots, a^\ell$ reasonably well, the process should be able to decrease the predicted error in each iteration.

C. Learning Control

Iterative learning control (ILC) has gained great success in motion tracking control [14]. We consider the gradient approach which propagates the output trajectory error to the input using the gradient map which is just the linearized system [15]:

1) Set $u = u_0$, apply u to the physical system and obtain the output sequence y;
2) Update u by adding a corrective term

$$\Delta u = -\alpha G^*(y - y_{des}) \tag{21}$$

where G^* is the adjoint of G, the linearized system about u, and α may be set as a sufficiently small constant or found by a line search;
3) Iterate until $\|y - y_{des}\|$ of $\|\Delta u\|$ is sufficiently small.

Fig. 12 shows the responses of learning control based on linear model and hybrid model, respectively. Since the control design uses linearization of the hybrid model, two models generate similar gradient and, thus, have similar performances.

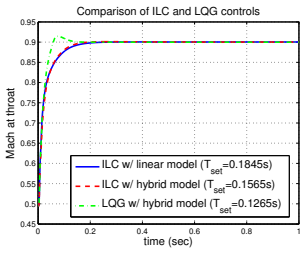


Fig. 12. Comparison of learning control and LQG control

V. Conclusion

This paper presents a new model reduction approach that combines ideas from subspace-based system identification and POD method. It first identifies a reasonable linear model using N4SID, and then introduces nonlinearity to the model by projecting the difference between CFD data and identified model onto a set of POD basis. The hybrid model is used for control design and shown to have consistently good performances, compared to linear, semilinear and nonlinear models.

Future directions include using system trajectories that contain richer frequency components for model training and applying the hybrid approach to a more complex case (e.g., an inlet duct with settling chamber, contraction and downstream diffuser), which has a lot more sampling points and parallel computing issue.

VI. Acknowledgment

This work was supported in part by the Center for Automation Technologies and Systems (CATS) under a block grant from the New York State Foundation for Science, Technology and Innovation (NYSTAR). The authors would also like to thank Yi Chen, Onkar Sahni, Ken Jansen, Miki Amitay, John Vaccaro for their help and guidance on aerodynamic modeling and computational fluid dynamics.

References

[1] Lumley, J. L., "The Structure of Inhomogeneous Turbulence," Atmospheric Turbulence and Radio Wave Propagation, edited by Yaglom, A. M. and Tatarski, V. I., Nauka, Moscow, pp. 166-178, 1967.
[2] Sirovich, L., "Turbulence and the Dynamics of Coherent Structures," Parts I-III, Quart. Appl. Math. XLV (3), pp. 561-590, 1987.
[3] Rowley, C. W., Colonius, T., and Murray, R. M., "POD Based Models of Self-Sustained Oscillations in the Flow Past an Open Cavity," AIAA 2000-1969, 6th AIAA/CEAS Aeroacoustics Conference, 2000.
[4] Smith, T. R., "Low-Dimensional Models of Plane Couette Flow Using the Proper Orthogonal Decomposition," Ph.D. thesis, Princeton University, 2003.
[5] Willcox, K., and Peraire, J., "Balanced Model Reduction via the Proper Orthogonal Decomposition," AIAA Journal, vol. 40, no. 11, pp. 2323-2330, 2002.
[6] Fahl M., "Trust-Region Methods for Flow Control Based on Reduced Order Modeling," Ph.D. Thesis, University of Trier, 2000.
[7] Van Overschee, P., and De Moor, B., "N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems," Automatica, Special Issue on Statistical Signal Processing and Control, 30(1): pp. 75-93, 1994.
[8] Van Overschee, P., and De Moor, B., "N4SID: Numerical Algorithms for State Space Subspace System Identification," in Proceedings of the 12th IFAC World Congress, vol. 7, pp. 361-364, 1993.
[9] Kim, K., Debiasi, M., Schultz, R., Serrani, A., and Samimy, M., " Dynamic Compensator for a Synthetic-Jet-Like Compression Driver Actuator in Closed-Loop Cavity Flow Control," 45th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2007-880, 2007.
[10] Kim, K., Beskok, A., and Jayasuriya, S., "Nonlinear System Identification for the Interaction of Synthetic Jets with a Boundary Layer," in Proceedings of 2005 American Control Conferance, pp. 1313-1318, 2005.
[11] Vaccaro, J. C., Sahni, O., Olles, J., Jansen, K. E., and Amitay, M., "Experimental and Numerical Investigation of Active Control of Inlet Ducts," International Journal of Flow Control, vol. 1, pp. 133-154, 2009.
[12] Jansen, K. E., "A Stabilized Finite Element Method for Computing Turbulence," Computer Methods in Applied Mechanics and Engineering, vol. 174, pp. 299-317, 1999.
[13] Ge, X., Gressick, W., Wen, J. T., Sahni, O., and Jansen, K. E., "A Low-Order Nonlinear Models for Active Flow Control of a Low L/D Inlet Duct," in Proceedings of 2010 American Control Conference, 2010, pp. 2989-2994, 2010.
[14] Moore, K. L., "Iterative learning control - An expository overview," Applied and Computational Control, Signals, and Circuits, 1, pp.151–214, 1999.
[15] Potsaid, B., Wen, J. T., Unrath, M., Watt, D., and Alpay, M., "High Performance Motion Tracking Control for Electronic Manufacturing", Journal of Dynamic Systems, Measurement, and Control, vol. 129, pp. 767-776, 2007.