INTERPOLATING LOCAL MODELS OF POD USING FUZZY DECISION MECHANISMS

Mehmet Önder Efe[1], **Xin Yuan**[2], **Hitay Özbay**[3] and **Mo Samimy**[4]

[1]Dept. of Mechatronics Eng., Atilim University, Incek, TR-06836, Ankara, Turkey, e-mail: onderefe@ieee.org (Corresponding Author)

[2]Collaborative Center of Control Science, Dept. of Electrical Eng., The Ohio State Univ., Columbus, OH 43210, U.S.A., e-mail: yuanx@ee.eng.ohio-state.edu

[3]Collaborative Center of Control Science, Dept. of Electrical Eng., The Ohio State Univ., e-mail: ozbay@ee.eng.ohio-state.edu, part of the work was done at Dept. of Electrical & Electronics Eng., Bilkent Univ., Ankara TR-06800, Turkey

[4]Collaborative Center of Control Science, Dept. of Mechanical Eng., The Ohio State Univ., Columbus, OH 43210, U.S.A., e-mail: samimy.1@osu.edu.

Abstract

In aerospace applications Proper Orthogonal Decomposition (POD) is used to obtain a dynamical model of aerodynamic flow control systems from simulation or experimental data. For different external test inputs (with disjoint frequency contents) the models obtained via POD are different, i.e. these models are valid locally in a certain narrow frequency band. As the external input gets rich in the frequency content the model is unable to generate a good estimate of the states. Therefore, interpolation of these local models is necessary to capture a model that works in the whole frequency range of interest. We study a fuzzy system to perform a smooth transition from one model to another, and realize the transition scheme in the frequency domain. We illustrate the results on the one dimensional Burgers equation.

1 Introduction

Proper Orthogonal Decomposition (POD) is a modeling technique used for infinite dimensional systems, based on numerical simulation or experimental data, [1-5]. Application areas include flow control problems, where the underlying system dynamics are represented by Navier-Stokes equations. For this type of systems, POD generates a reduced order model. Though the method is powerful in expanding a given solution to a set of orthogonal eigenbasis, the resulting set of Ordinary Differential Equations (ODEs) is autonomous, i.e. the external excitation is implicitly contained. In [6], it is demonstrated that a suitable partitioning of the spatial domain illustrates the control input explicitly in the new system of equations. This approach has been used with Singular Value Decomposition (SVD) in [6], and with POD in [7]. The latter also identifies a substantial problem of modeling procedure: that is the model can give a good estimate of the states only when the external input's frequency content is similar to the boundary excitation used when the original data was collected. This fact can be viewed as an analogy between the situation at hand and learning of exposed data. In other

words, the system is unable to respond correctly to the situations that are not encountered in the basis generation phase. This paper discusses the problem from the point of merging several models utilizing expert systems such as fuzzy inference systems.

Assume that a set of solutions obtained under several excitation conditions is available. For high frequency excitations, the spatial distribution turns out to be such that the activity takes place only around the boundary, however, for low frequencies, the boundary excitation is well distributed over the spatial domain. This fundamental fact is relevant to the stiffness (or viscosity) of the Partial Differential Equation(s) (PDE) characterizing the process under investigation, and it appears as a gradual change as the frequency spectrum is swept. An exemplar case of this for Burgers Equation is depicted in Figure 1. Therefore, it is reasonable to observe that the coefficients of the models will display tiny differences thereby leading to behaviorally different systems of ODEs. Nevertheless, one can use fuzzy logic to obtain a semiglobal model, which works in a wider band of frequency range than those of its constituents. The method is based on the fuzzification in the frequency domain, and quantification of excitation levels of each local model (individual boundary conditions) from an inner product defined in frequency domain multiplied by the given boundary condition. The defuzzification stage outputs a crisp value, which is the reconstructed output of the PDE. The parameters of the defuzzification stage are adjustable thereby enabling the time-variable weighted mixtures of the individual model outputs to form the total output.

Burgers equation is a good example to study these issues as it is nonlinear, yet simple enough to devise strategies for modeling and control design. In [8], the issues of uncertainty on the viscosity parameter are discussed in the context of adaptive control system design. A backstepping boundary control procedure for Burgers equation is described in [9]. Park *et al*, [10], focus on the controller synthesis for Burgers equation. Having obtained the POD based reduced order model, a cost function is described and conjugate gradient technique is used to minimize it. Burns *et al* study the problem with design in Hilbert spaces and demonstrate the efficacy of the design by finite element based simulations, [11]. For a more detailed discussion the reader is referred also to the references given in [8-11]

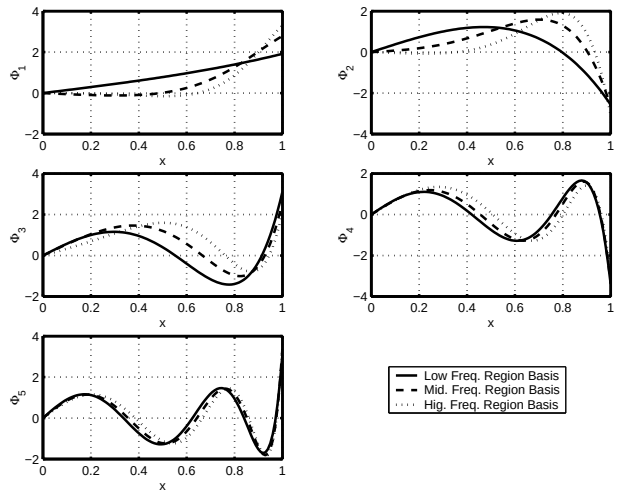

Figure 1: A closer look at the basis functions generated for three different boundary excitation conditions, namely a unit magnitude sinusoidal signal with frequency 0.01 Hz (LOW), 5 Hz (MIDDLE) and 10 Hz (HIGH).

This paper is organized as follows. We describe the POD scheme briefly in the Section 2 and discuss model reduction and control separation in Section 3. A detailed discussion on these topics can be found in [6] and [7]. In the Section 4, the frequency domain partitioning

is explained and the wide-band interpolation is explained. Simulation results are presented in Section 5. Finally, concluding remarks are given in Section 6.

2 Proper Orthogonal Decomposition

Consider the ensemble $U_k(x)$ for $k = 1, 2, \ldots, N$, where $x \in \Omega := [0, 1]$, and the index k corresponds to the observed "snapshots" (i.e. numerical simulation result, or experimental data collected, at kth time instant) from a process, say for example, the Burgers equation $u_t(x,t) = \epsilon u_{xx}(x,t) - u(x,t)u_x(x,t)$, where ϵ is a known constant.

The goal is to find an orthogonal basis set letting us to write the solution as

$$u(x,t) \approx \sum_{i=1}^{N} \alpha_i(t)\phi_i(x), \tag{1}$$

where $\alpha_i(t)$ is the temporal part, and $\phi_i(x)$ is the spatial part. It will later be clear that if the basis set $\{\phi_i(x)\}_{i=1}^N$ is an orthogonal set, then the modeling task can exploit Galerkin projection technique.

Let us summarize the POD procedure.

Step 1. Start calculating the $N \times N$ correlation matrix L, the $(ij)^{th}$ entry of which is $L_{ij} := \langle U_i, U_j \rangle_\Omega$, where $\langle ., . \rangle_\Omega$ is the inner product operator defined over the spatial domain (Ω) of the process.

Step 2. Find the eigenvectors (denoted by v_i) and the associated eigenvalues (λ_i). Sort them in a descending order in terms of the magnitudes of λ_i. Note that every v_i is an $N \times 1$ vector satisfying $v_i^\intercal v_i = \frac{1}{\lambda_i}$, here, for simplicity of the exposition we assume that the eigenvalues are distinct.

Step 3. Construct the basis set by using

$$\phi_i(x) = \sum_{j=1}^{N} v_{ij} U_j(x), \tag{2}$$

where v_{ij} is the j^{th} entry of the eigenvector v_i, and $i = 1, 2, ..., \text{rank}(L)$. It can be shown that $\langle \phi_i(x), \phi_j(x) \rangle_\Omega = \delta_{ij}$ with δ_{ij} being the Kronecker delta function.

Notice that the basis functions are admixtures of the snapshots.

Step 4. Calculate the temporal coefficients. Taking the inner product of both sides of (1) with $\phi_i(x)$, the orthogonality lets us have

$$\begin{aligned} \alpha_i(t_0) &= \langle \phi_i(x), u(x,t_0) \rangle_\Omega \\ &= \langle \phi_i(x), U_{t_0} \rangle_\Omega, \end{aligned} \tag{3}$$

Without loss of generality, an element of the ensemble $\{U_i(x)\}_{i=1}^N$ may be $U(x,t_0)$. Therefore, to generate the temporal gain ($\alpha_i(t)$) of the spatial basis $\phi_i(x)$, one would take the inner product with the elements of the ensemble with the basis functions.

A standing assumption of this procedure is that the solution is dominated by coherent structures letting us write the solution as a sum given in (1), so the both sides of (1) is assumed to be equal, [1-7].

In the next section, we demonstrate how the boundary condition is transformed to an explicit control input in the ODEs.

3 Reduced Order Modeling and Separation of the Boundary Control

In this section, we apply the POD technique to the viscous Burgers equation described by

$$u_t(x,t) = \epsilon u_{xx}(x,t) - u(x,t)u_x(x,t), \tag{4}$$

where $\epsilon = 1$ is a known process parameter, $x \in \Omega$ and $\Omega = [0,1]$. The problem is specified with the initial condition $u(x,0) = 0\ \forall x$, the homogeneous boundary condition at $x = 0$ as $u(0,t) = 0$ and Dirichlét boundary condition at $x = 1$ as $u(1,t) = \gamma(t)$, where $\gamma(t)$ is the external input of the system. Since the POD scheme yields the decomposition in (1), choosing the most dominant M modes ($M \leq \texttt{min}(\texttt{rank}(L), N)$), and inserting this into (4) results in

$$\begin{aligned}\sum_{i=1}^{M} \dot{\alpha}_i(t)\Phi_i(x) &= \sum_{i=1}^{M} \alpha_i(t)\epsilon\frac{\partial^2\Phi_i(x)}{\partial x^2} \\ &- \sum_{i=1}^{M}\sum_{j=1}^{M} \alpha_i(t)\alpha_j(t)\Phi_i(x)\frac{\partial\Phi_j(x)}{\partial x}.\end{aligned} \tag{5}$$

Taking the inner product of both sides of (5) with $\Phi_k(x)$, which corresponds to the Galerkin projection, results in the equality in (6).

$$\begin{aligned}\dot{\alpha}_k(t) &= \sum_{i=1}^{M} \alpha_i(t)\epsilon\langle\Phi_k(x), \zeta_i(x)\rangle \\ &- \sum_{i=1}^{M}\sum_{j=1}^{M} \alpha_i(t)\alpha_j(t)\langle\Phi_k(x), \Phi_i(x)\beta_j(x)\rangle,\end{aligned} \tag{6}$$

where $\zeta_i(x) := \frac{\partial^2\Phi_i(x)}{\partial x^2}$ and $\beta_i(x) := \frac{\partial\Phi_i(x)}{\partial x}$. As mentioned earlier, the effects of the external stimulus is implicit in the above equation. For this reason, define the grid as $\underline{x} = (\bigcup_{i=0}^{S-1} i\Delta x)$, where Δx is the spatial step size and S is the number of grid points considered for the numerical solution satisfying $(S-1)\Delta x = 1$. Partitioning the grid as $\underline{x} = \left(\bigcup_{i=0}^{S-2} i\Delta x\right) \bigcup 1 = \begin{pmatrix} \underline{x}^{\circ\intercal} & 1 \end{pmatrix}^{\intercal}$, one can calculate the values of the functions $\Phi_k(x)$ and $\zeta_i(x)$ at every grid point, and rewrite them in the vector form as $\Phi_k(\underline{x})$ and $\zeta_i(\underline{x})$ respectively. Since the external inputs are not seen explicitly in (6), one has to manipulate the expression above. The driving point is to notice that the solution in (1) must be satisfied at the boundaries as well. This gives the following information;

$$u(1,t) = \gamma(t) = \sum_{i=1}^{M} \alpha_i(t)\Phi_i(1). \tag{7}$$

Or $\alpha_k(t)\Phi_k(1) = \gamma(t) - \sum_{i=1}^{M}(1-\delta_{ik})\alpha_i(t)\Phi_i(1)$, which is included by both terms of (6). Manipulating the equations would introduce the control term and would let us have the following set of ODEs with $\underline{\alpha} = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_M \end{pmatrix}^{\intercal}$ being the state vector,

$$\dot{\underline{\alpha}} = \boldsymbol{A}\underline{\alpha} - \underline{B}(\underline{\alpha}) + (\underline{C} - \boldsymbol{D}\underline{\alpha})\gamma, \tag{8}$$

where

$$(\boldsymbol{A})_{ki} = \frac{1}{N}\epsilon\left(\Phi_k^{\intercal}(\underline{x})\zeta_i(\underline{x}) - \Phi_i(1)\zeta_k(1)\right), \tag{9}$$

$$\underline{B}(\underline{\alpha}) = \begin{pmatrix} \underline{\alpha}^{\intercal}\boldsymbol{B}_1\underline{\alpha} & \underline{\alpha}^{\intercal}\boldsymbol{B}_2\underline{\alpha} & \dots & \underline{\alpha}^{\intercal}\boldsymbol{B}_M\underline{\alpha} \end{pmatrix}^{\intercal}, \tag{10}$$

where $(\boldsymbol{B}_k)_{ij} = \frac{1}{N}\Phi_k^{\intercal}(\underline{x}^{\circ})(\Phi_i(\underline{x}^{\circ}) \star \beta_j(\underline{x}^{\circ}))$.

$$(\underline{C})_k = \frac{1}{N}\epsilon\zeta_k(1), \tag{11}$$

and

$$(\boldsymbol{D})_{ki} = \frac{1}{N}\beta_i(1)\Phi_k(1). \tag{12}$$

One should remember that the model above is valid only for a limited range of frequencies contained in the boundary conditions letting us to produce the ensemble U.

4 Fuzzy Decision Mechanism for Interpolating Multiple Models

Fuzzy systems have extensively been used for the applications in which the prescribed task can better be fulfilled by a suitably integrated local decisions than global mechanisms which are typically hard to achieve. Furthermore, since fuzzy inference systems are universal approximators, a given mapping can be realized with an arbitrary degree of accuracy given enough number of rules over the universe of discourse. A more detailed treatment on fuzzy logic and its applications can be found in [12].

In this paper, we use a SISO fuzzy decision mechanism, which utilizes triangular membership functions, singleton fuzzifier and weighted average dynamical defuzzifier. We use a simple fuzzy system which is intended to interpolate three models generated at different frequencies. We use only three rules in the rule base, which quantifies its argument as LOW, MIDDLE and HIGH.

We calculate the 1024-point Fast Fourier Transform (FFT) of the boundary excitation (γ), and consider the frequency content of the signal. On the other hand, we have already generated the dynamic models (Refer to equations (1) and (8)) for 0.01 Hz, 5 Hz and 10 Hz sinusoidal excitations. Every model is driven by a portion of the boundary excitation. The level of excitation is determined by the firing strengths of each rule, i.e. w_i. Since we have only one input, $w_i = \mu_i$, where μ_i is the i-th membership function. In Figure 2, the choice of the membership functions are depicted.

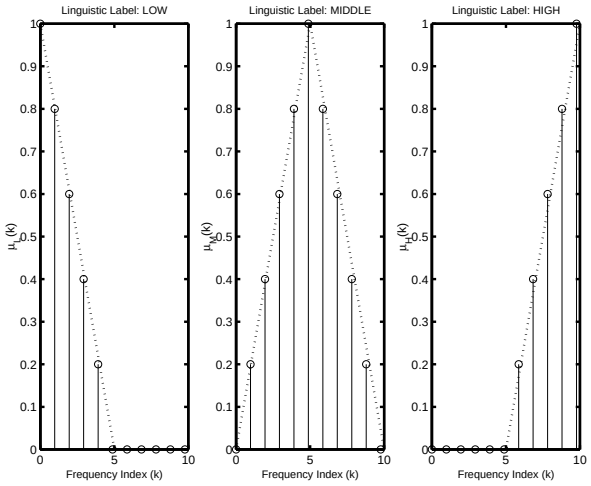

Figure 2: Membership functions in frequency domain, which has been partitioned into three fuzzy subspaces and considered only at the frequencies set by FFT algorithm.

With $w_i = \mu_i$ in mind, the selection of the membership functions in Figure 2 becomes more comprehensible. A low frequency signal should significantly excite the first model (LOW), and a high frequency signal should significantly excite the third model (HIGH). The transition

is smooth and is handled by the second model (MIDDLE). The sampling period (T_s) is 1 msec, therefore, every entry of the membership functions in Figure 2 occurs at the integer multiples of $\frac{1}{1024T_s}$ Hz. Since we are interested in frequencies up to 10 Hz, the universe of discourse for fuzzy model interpolation is [0 Hz, $\approx$ 10 Hz] band. It should be noted that the excitations around 10 Hz are termed as high frequencies but 10 Hz might not give the feeling of high frequency from a technical perspective. One should see this as the upper bound of the frequency range of interest. If a wider band of frequency domain is to be addressed, then more local models (more fuzzy levels) should have been used to capture the necessary dynamical information. Considering the change depicted in Figure 1, a fuzzy scheme interpolating the models generated for example at 0.01 Hz, 50 Hz and 100 Hz would fail since these models do not carry any information for 25 Hz excitations.

In the view of all these, first the frequency picture of the boundary excitation is constructed, then its first 11 elements are masked. The 11-tuple vector (denoted by Γ) is normalized to unit sum. Then the unit sum vector and each membership function are passed through an inner product to get the scaling factor for corresponding model. This process can be described compactly as

$$w_j = \sum_{k=1}^{11} \mu_j(k) \left(\frac{\Gamma(k)}{\sum_{p=1}^{11} \Gamma(p)} \right). \tag{13}$$

where k is the frequency index seen on Figure 2 and $j = 1, 2, 3$; or equivalently LOW, MIDDLE and HIGH respectively.

Next the input signal for each model is computed as $\gamma_j(t) = w_j \gamma(t)$ and the subsystems are left for state and output evolution. The total output of the system is calculated through a weighted summation of the outputs of the local models:

$$\hat{u}(x,t) = \sum_{n=1}^{3} y_n(t) u_n(x,t), \tag{14}$$

where $y_n(t)$ is the weight used to determine the contribution of nth local model on the reconstructed output.

Clearly a comparison of the result obtained through the numerical solution of the PDE and the proposed model can be made by setting some boundary excitation, solving the PDE, choosing a test location and picking up the values computed at that location as the *desired output* ($u_d(x,t)$) and analyzing the difference between this value and $\hat{u}$. In the next section, simulation results are presented.

5 Simulation Results

A. Obtaining the Local Models

According to the procedure described in the second and third sections, the PDE has been solved by using Crank-Nicholson method (See [13] for details), with a step size of 1 msec. The initial conditions are taken as zero everywhere and the process parameter is set as $\epsilon = 1$. In order to form the solution, a linear grid having $S = 100$ points is chosen. According to the above parameter values, a total of 1001 snapshots embody the entire numerical solution, among which a linearly sampled $N = 101$ snapshots have been used for the POD scheme. Clearly the simulation end time is 1 sec. Although one might use the entire set of snapshots, it has been shown that a reasonably descriptive subset of them can be used. In the literature, this approach is called *method of snapshots*, which significantly reduce the computational intensity of the overall scheme, [1,2,7]. Once the modes have been obtained, we have truncated the solution at $M = 5$, which represents almost %100 of the total energy which is described as $E = \left(\sum_{i=1}^{M} \lambda_i\right) / \left(\sum_{i=1}^{N} \lambda_i\right)$.

The low frequency local model is obtained with the boundary condition $\gamma(t) = \sin(2\pi 0.01t)$, similarly, for the middle frequency region and the high frequency region we utilize $\gamma(t) =$

$\sin(2\pi 5t)$ and $\gamma(t) = \sin(2\pi 10t)$ respectively. The basis functions for these three cases have already been depicted in Figure 1, which demonstrate the slight change as the frequency content of the boundary condition changes.

For each case we have reconstructed the temporal variables ($\underline{\alpha}$) and verified that the models function properly.

B. Nonadaptive Case - $y_n(t) = 1$

An intuitive selection of the weights is to set them all to unity. Since all the models are synthesizing the response relevant frequency domain, the output should be built up collectively. Based on this idea, we observed the results seen in Figure 3.

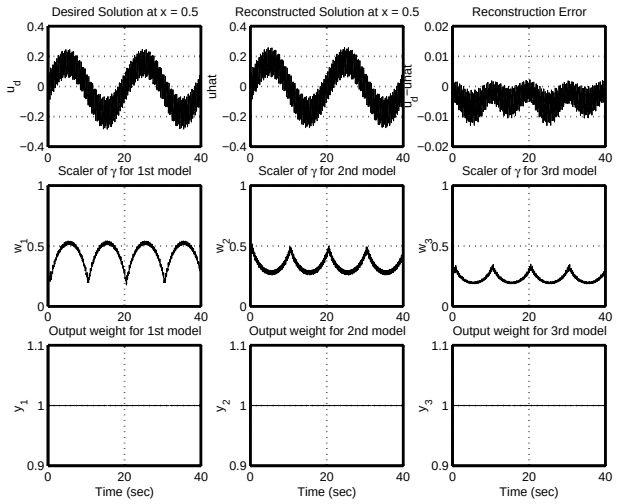

Figure 3: Simulation results for the nonadaptive case.

It is seen that the desired signal taken from the numerical solution has both low and high frequency components. The boundary condition leading to the observation of this signal at $x = 0.5$ was

$$\gamma(t) = \frac{1}{3}\left(\sin(2\pi 0.05t) - \sin(2\pi 3t) + \cos(2\pi 8t)\right), \tag{15}$$

which has enough spectral richness to excite all three membership functions depicted in Figure 2. The time variation of the quantity described in (13) is depicted in the second row of Figure 3. The soft switching strategy excites a model a the level read from these values.

It is apparent that the observed error is admissibly small in magnitude and justifies the intuitive selection of $y_n(t) = 1$ $(n = 1, 2, 3)$ depicted also in the last row of Figure 3.

C. Adaptive Case - $\dot{y}_n(t) = \eta e(t) u_n(t)$

Define the cost function $J(t) = \frac{1}{2}e(t)^2$, where $e(t) := u_d(x,t) - u(x,t)$ at a given x. The gradient descent (or MIT rule) prescribes the following update law

$$\dot{y}_n = -\eta \frac{\partial J}{\partial y_n}, \tag{16}$$

where $\eta > 0$ is a design parameter. Calculating the derivative, we have

$$\dot{y}_n \quad = \quad -\eta e \frac{\partial e}{\partial y_n}$$

$$= \eta e \frac{\partial \hat{u}}{\partial y_n}$$
$$= \eta e u_n. \quad (17)$$

The results obtained through the adaptive case are illustrated in Figure 4, with $\eta = 1000$.

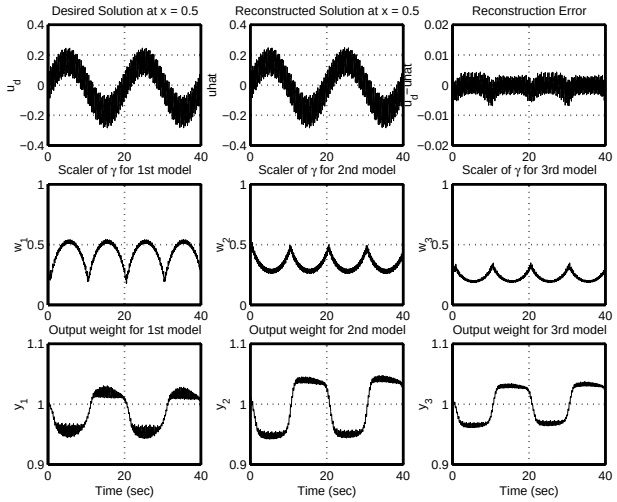

Figure 4: Simulation results for the adaptive case.

As the above figure suggests, under the same conditions, adaptive case removes the mean of the error signal (See Figure 3). The tiny variation in the parameters is also visible. Clearly the cost of the removal of error mean is a slight increase in the computational burden. However, in terms of parametric stability, there is no guarantee of parametric convergence or finite volume evolution for y_is.

We have repeated the tests for the following adaptation law too:

$$\dot{y}_n(t) = \eta u_n(t) sgn(e(t)), \quad (18)$$

which further reduces the error magnitude at the cost of introducing very fast parametric fluctuations taking place all around unity. A more detailed discussion on these issues can be found in any adaptive control textbook.

D. An Overall Assessment

Advantages:

i) The method is simple, and it works in a relatively large frequency region.

ii) There is no limit on the number of models to be involved in the procedure.

iii) If the number of local models increases, the increase in the computational complexity is affordable as the fuzzy system has a single input.

iv) As an alternative to the presented technique, one might propose filtering the signals before applying to the local models, however, this would introduce delay into the feedforward path. The method presented here does not introduce delays, furthermore filter based transitions do not provide as more flexible smoothness as fuzzy systems do.

v) The proposed technique is quite flexible to integrate with adaptive schemes.

vi) The partitioning can arbitrarily be done in the frequency domain. If the local models are valid on sub-domains which are not topologically similar, one can still use fuzzy decision mechanisms with modified membership functions.

vii) Time variations can be handled by utilizing parameterized membership functions.

viii) Rule number can be increased/decreased easily.

Disadvantages: Although the state of the art DSP hardware and computing tools provide extensive possibilities

i) The algorithm needs 1024-point FFT at every time step.

ii) When used with adaptive schemes more computational power is required.

iii) The adaptive cases entail proof of stability for bounded parameter evolution.

The presented technique aims to demonstrate the efficacy of the fuzzy decision mechanisms in interpolating local models of POD. Burgers system is a simple example for this, however, there is an ongoing research aiming to demonstrate the use of such techniques for aerodynamic flows, which are governed by Navier-Stokes equations. It is our understanding that the results presented in this paper are promising in the sense of applicability to the aerodynamic flow modeling problems.

6 Conclusions

Proper Orthogonal Decomposition based models are found to be useful in many applications displaying spatial continuum. However, these models are valid only under the conditions which have been used to generate them. Therefore, a model derived with low frequency boundary excitation is invalid for high frequency excitations and vice versa. A remedy to this problem is to find an interpolation technique which smoothly activates one model when the conditions leading to it are encountered. A fuzzy decision mechanism constitutes a good candidate for these problems. It allows a smooth transition from one region to another as those regions are characterized by fuzzy sets. Burgers equation is used as the test bed, and three models have been generated around 0.01 Hz, 5 Hz and 10 Hz points. A simple fuzzy model is used to quantify the frequency content of the boundary excitation, and the models have been excited by a scaled version of $\gamma(t)$. The coefficient is determined from the frequency picture and the membership functions covering the universe of discourse. It is observed that the use of fuzzy decision mechanisms in frequency domain provide good interpolation. A detailed discussion of advantages and disadvantages has been presented in the previous section. The method introduces significant improvements and a real-time version of it would be useful with high-power computing devices.

Future work in this field aims to demonstrate the efficient model interpolation techniques for aerodynamic flow modeling problems.

Acknowledgments

This material is based on research sponsored by Air Force Research Laboratory, Agreement no: F33615-01-2-3154.

The authors gratefully acknowledge the experimental facilities of Collaborative Center of Control Science Ohio State University.

References

[1] S.S. Ravindran, A reduced order approach for optimal control of fluids using proper orthogonal decomposition. *Int. J. for Numerical Methods in Fluids*, 34, 425-488 (2000).

[2] H.V. Ly and H.T. Tran, Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling of Dynamical Systems*, 33, 223-236 (2001).

[3] S.N. Singh, J.H. Myatt, G.A. Addington, S. Banda and J.K. Hall, Optimal feedback control of vortex shedding using proper orthogonal decomposition models. *Trans. of the ASME: J. of Fluids Eng.*, 123, 612-618 (2001).

[4] P.N. Blossey and J.L. Lumley, Reduced-order modeling and control of near-wall turbulent flow. Proc. of the 38th Conf. on Decision and Control, 2851-2856, Phoenix, Arizona, U.S.A. (1999).

[5] J.A. Atwell and B.B. King, Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and Computer Modelling of Dynamical Systems*, 33, 1-19 (2001).

[6] M.Ö. Efe and H. Özbay, Integral Action Based Dirichlét Boundary Control of Burgers Equation. Proc. of IEEE Int. Conf. on Control Applications, Istanbul, Turkey, June 2003, pp. 1267–1272.

[7] M.Ö. Efe and H. Özbay, Proper orthogonal decomposition for reduced order modeling: 2D heat flow. Proc. of IEEE Int. Conf. on Control Applications, Istanbul, Turkey, June 2003, pp. 1273–1277.

[8] W.-J. Liu and M. Krstic, Adaptive control of Burgers equation with unknown viscosity. *Int. J. of Adaptive Control and Signal Proc.*, 15, 745-766 (2001).

[9] W.-J. Liu and M. Krstic, Backstepping boundary control of Burgers equation with actuator dynamics. *Systems and Control Letters*, 41, 291-303 (2000).

[10] H.M. Park and Y.D. Yang, Control of Burgers equation by means of mode reduction. *Int. J. of Engineering Science*, 38, 785-805 (2000).

[11] J. Burns, B.B. King and L. Zietsman, Functional gain computations for a 1D parabolic equation using non-uniform meshes. Proc. of IEEE Int. Conf. on Control Applications (CCA-02), September 18-20, Glasgow, Scotland (2002).

[12] J.-S.R. Jang, C.-T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, Upper Saddle River, NJ, (1997).

[13] S.J. Farlow, *Partial Differential Equations for Scientists and Engineers*, Dover Publications Inc., New York, 317-322 (1993).