17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

Obtaining Regional Dynamical Models for Flow Control using Wavelet Transform

Türker Nazmi ERBİL
Electrical and Electronics Engineering Department
TOBB Economics and Technology University
Söğütözü, Ankara 06560, Turkey
tnerbil@etu.edu.tr

Coşku KASNAKOĞLU
Electrical and Electronics Engineering Department
TOBB Economics and Technology University
Söğütözü, Ankara 06560, Turkey
kasnakoglu@etu.edu.tr

Abstract— **In this paper, we utilize wavelet transform to obtain dynamical models describing the behavior of fluid flow in a local spatial region of interest. First, snapshots of the flow are obtained from experiments or from computational fluid dynamics (CFD) simulations of the governing equations. A wavelet family and decomposition level is selected by assessing the reconstruction success under the resulting inverse transform. The flow is then expanded onto a set of basis vectors which are constructed from the wavelet function. The wavelet coefficients associated with the basis vectors capture the time variation of the flow within the spatial region covered by the support of basis vectors. A dynamical model is established for these coefficients by using subspace identification methods. The approach developed is applied to a sample flow configuration on a square domain where the input affects the system through the boundary conditions. It is observed that there is good agreement between CFD simulation results and the predictions of the dynamical model. A controller is designed based on the dynamical model and is seen to be successful in regulating the velocity of a given point within the region of interest.**

Keywords-flow control; regional dynamic modeling; wavelet transform;

I. INTRODUCTION

The term fluid flow refers to the motion of liquids and gases, which is an important part of everyday life. The air flow over the wings of an airplane, crude oil flow in a pipeline or water flow around the body of a submarine are all examples fluid flow. Thus, from a scientific and technological point of view, modeling and understanding of fluid flow is an issue of high importance [1, 2]. One of the most common methods in dynamical modeling of fluid flow is the Proper Orthogonal Decomposition/Galerkin Projection (POD/GP) method. In this approach, one obtains a set of modes called POD modes, which capture a sufficiently large amount of energy of the flow. The flow is then expanded in terms of these modes and this expansion is substituted into the partial differential equations (PDEs) representing the flow, resulting in a set of ordinary differential equations (ODEs) in the time coefficients of the modes [3, 4, 5]. While this process does indeed result in finite dimensional dynamical models, it is still very difficult to perform analysis and design as these models as they are non-linear in nature. Another issue is that the POD modes do not have a compact support but instead they are spread out to the entire flow domain. Hence the time coefficients associated with the modes do not provide direct information regarding changes in a local spatial domain of interest. In many cases one is interested in the dynamical behavior in a given local region only, so it is of interest to build models whose states can directly be associated with a spatial region.

In this paper we utilize wavelet transform methods [6, 7, 8, 9] to develop a dynamical local modeling approach for flow control, which addresses the aforementioned problems. The paper is organized as follows: Section II presents an overview of the wavelet transform and the Navier-Stokes equations. Next, the approach proposed in the paper is explained in Section III and it is illustrated with a flow control example in Section IV, where the task is to regulate the velocity of a given point inside a square. The paper ends with Section V, which provides conclusions and future work ideas.

II. MOTIVATION AND BACKGROUND

A. Wavelet Transform and Reconstruction, Multilevel Decomposition and Thresholdolding

The wavelet transform is among the most common methods in signal processing on which a large numbers of sources and studies exist [6, 7, 8, 9]. The wavelet transform is the representation of a function by wavelets, where the wavelets are scaled and translated versions of a finite-length fast-decaying oscillating waveform called the wavelet function. Wavelet transforms are advantageous over traditional Fourier transforms for representing functions that have discontinuities and sharp peaks, and for accurately deconstructing and reconstructing finite, non-periodic and/or non-stationary signals. The wavelet transform can be expressed mathematically as the integration of scaled and shifted versions of a wavelet function over time, i.e.

$$C(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} f(t)\psi(\text{scale}, \text{position}, t)dt \qquad (1)$$

where C represents the Wavelet Transform coefficients and ψ represents the wavelet function which depends on the wavelet family being used for the process. There are numerous families available for wavelet transform, including BNC, Coiflet-Daubechies-Feauveau, Daubechies, Haar, Mathieu, Legendre,

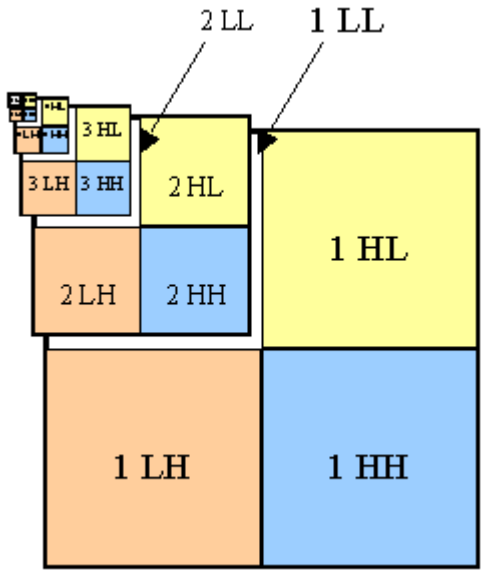

Figure 1. Multilevel 2D wavelet decomposition

Villasenor, and Symlet. The reconstruction of the function f is obtained by the summation of the coefficients C multiplied by the wavelet function ψ that is scaled and shifted properly.

In numerical analysis and functional analysis problems, a sampled version of the continuous wavelet transform described above is used more commonly, which is called the discrete wavelet transform (DWT). This is also the method that we employ in this paper. In DWT, the signal to be analyzed is filtered into high-pass and low-pass filters with certain cutoff frequencies, and the resulting signal is downsampled to obtain an equal number of data as the original signal. The inverse transform for rebuilding the signal from wavelet coefficients is also done in a similar but backwards fashion: After upsampling, one applies reconstruction low-pass and high-pass filters to approximation and detail coefficients respectively, and combines the two to obtain the reconstructed signal.

The wavelet transform can also be applied to two-dimensional signals, by applying filtering and downsampling first to the columns, and then to the rows. This results in four matrices containing the wavelet coefficients; one for the approximation coefficients, and three for the detail coefficients in horizontal, vertical and diagonal directions. This procedure can be repeated on the approximation coefficients to obtain a second level of approximation and detail coefficients, and then on the second level approximation coefficients to obtain a third level of coefficients, and so on. This process is termed the *multilevel* DWT and is illustrated in Figure 1.

Also worth mentioning is the procedure of *thresholding*, which is a common post-transform operation to apply to the wavelet coefficients. The thresholding process can be described as follows

$$Y = \begin{cases} X, & \text{for } |X| > T \\ 0, & \text{for } |X| \le T \end{cases} \tag{2}$$

where X represents the detail coefficients, Y represents the thresholded detail coefficients and $T \in \mathbb{R}_+$ is the threshold value. The expression shown above states that if the absolute value of a coefficient is greater than the threshold value, this coefficient is saved; otherwise it set to zero. It is quite common that one can pick a very small value for T and still achieve an acceptable reconstruction from the thresholded coefficients. Since a small value for T implies that most detail coefficients will be set to zero, one can store the thresholded coefficients in a sparse matrix to save space, which is the basic idea behind using wavelet transform for the compression of images and videos.

B. Navier-Stokes (NS) Equations

The Navier-Stokes PDEs are the most commonly used equations to describe the behavior of fluid flow, and have the following form

$$\frac{\partial q}{\partial t} + (q \cdot \nabla)q = \nabla p + \nu \Delta q \tag{3}$$

where $\nu \in \mathbb{R}$ is the viscosity, $p(x, y, t) \in \mathbb{R}$ is the pressure and $q(x, y, t) = (u(x, y, t), v(x, y, t)) \in \mathbb{R}^2$ is the flow velocity with u and v being the components in the longitudinal and lateral directions [1]. The equations arise from applying Newton's second law to fluid motion, under the assumption that the fluid stress is the sum of a diffusing viscous term plus a pressure term. It is also common to assume that the flow is incompressible; meaning that equation (3) is subject to the divergence condition $\nabla \cdot q = 0$.

III. MODELING APPROACH

The regional modeling process proposed in this paper consists of the following steps:

The first step in the modeling process is to record 2D instantaneous images, i.e. snapshots, of the flow. The snapshots can either be obtained from actual physical experiments using techniques such as particle image velocimetry (PIV), or from computer data that results from CFD simulations of the Navier-Stokes equations (3).

The next step is the selection of a wavelet function to be used. The selection criterion is that the wavelet function must be able to represent the flow snapshots with adequate accuracy, in the sense that the reconstructed snapshots formed from the wavelet coefficients are close to the original snapshots.

The third step is to determine the number of levels for the wavelet transform. Higher number of levels will result in the approximation coefficients getting decomposed further and will enable to flow snapshots to be represented with a fewer number of approximation coefficients. However if the number of approximation coefficients are too low, each coefficient will have to represent a larger spatial region, so the model resolution will decrease. Thus one must take these factors into account when determining a suitable level of decomposition.

The next step is the construction of a set of basis vectors $\Phi_i(\mathrm{x}, \mathrm{y})$ in terms of which the flow snapshots will be expressed as an expansion of the following form

$$q(x,y,t) = \sum_{i=1}^{N} a_i(t)\Phi_i(x,y) \tag{4}$$

where $N \in 2\mathbb{N}$ is the number of basis functions. Each $\Phi_i(\mathrm{x},\mathrm{y})$ captures the contribution of a local spatial region of the flow process. The basis vectors are to cover the spatial region of interest in both the longitudinal and lateral directions, and have the following form

$$\Phi_i(x,y) = \begin{bmatrix}\Phi_{i,u}(x,y)\\ \Phi_{i,v}(x,y)\end{bmatrix}, \qquad i = 1,\dots,N \tag{5}$$

Here the longitudinal component $\Phi_{i,u}$ is defined as

$$\Phi_{i,u}(x,y) = \begin{cases} \Upsilon_i(x,y), & i = 1,\dots,\dfrac{N}{2} \\ 0, & i = \dfrac{N}{2}+1,\dots,N \end{cases} \tag{6}$$

and the lateral component $\Phi_{i,v}$ is defined as

$$\Phi_{i,v}(x,y) = \begin{cases} 0, & i = 1,\dots,\dfrac{N}{2} \\ \Upsilon_{i-\frac{N}{2}}(x,y), & i = \dfrac{N}{2}+1,\dots,N \end{cases} \tag{7}$$

where the functions $\Upsilon_i : \mathbb{R}^2 \to \mathbb{R}$ for $i = 1,\dots,N/2$ are simply the wavelet function shifted and scaled appropriately, which can be obtained numerically by the cascade algorithm. This basically consists of taking a coefficient matrix that has the value one at the coefficient of interest and is zeros elsewhere, and then inverse transforming iteratively. Depending on the location of the wavelet coefficient, the oscillating part of the function ϕ_i will be located in a different region of the spatial domain. One must therefore pick a number of suitable ϕ_i functions whose support in $\mathbb{R}^2$ covers the spatial area of interest. The value N is then twice this number, as seen from (6) and (7). If the wavelet function is orthogonal, then it holds that

$$\langle \Phi_i(\mathrm{x},\mathrm{y}), \Phi_{\mathrm{j}}(\mathrm{x},\mathrm{y})\rangle = 0, \text{ for } i \neq j \tag{8}$$

and the wavelet coefficient $a_i(t)$ in (4) becomes the projection of the flow snapshots onto the basis function Φ_i. This allows for interpreting the basis vectors Φ_i as a set of coordinate axes which create an N dimensional subspace, and the coefficients a_i as the components of the flow variable q on these axes.

Having obtained an expansion of the flow as in (4), it is seen that the time variation of the flow is dictated by the coefficients a_i, since the vectors $\Phi_i$are constant with respect to time. Thus the modeling task for the flow is reduced to fitting a suitable dynamical model to the trajectories $a_i(t)$. For this purpose a state-space model of the following form will be sought

$$\xi(t+T_s) = A\xi(t) + B\gamma(t) \tag{9}$$

$$y(t) = C\xi(t) + D\gamma(t) \tag{10}$$

which is a discrete-time model since the flow snapshots are available at discrete time values separated by a sampling period of $T_s \in \mathbb{R}$ seconds. Here, $\xi \in \mathbb{R}^n$ is the state vector, $n \in \mathbb{N}$ is the degree of the system, $\gamma \in \mathbb{R}$ is control input and $y \in \mathbb{R}^N$ is the output signal. The matrices A, B, C and D determine the dynamical system and are to be obtained by constructing a model of the form (9)-(10) using system identification techniques. To construct the data for system identification, various input signals, e.g. sine waves, ramp functions and chirp signals, are applied to the system at a sampling period of T_s, and the snapshots resulting are recorded. Applying wavelet transform to these snapshots yields the system output, which consists of the N wavelet coefficients representing the region of interest, i.e.

$$y(t) = a(t) = [a_1(t)\ \ a_2(t) \dots a_N(t)]^T \tag{11}$$

From the input-output data obtained as described, subspace system identification methods (N4SID) are used for obtaining the A, B, C and D matrices in (9)-(10). These methods first construct the state trajectories termed the Kalman states, and then obtain the state matrices from these states using least squares techniques. The reader interested in further details regarding subspace system identification methods is referred to [10, 11].

The dynamical regional modeling approach described in this section is best illustrated by means of an example, as will be presented in the next section.

IV. Application Example

In this example we consider the fluid flow over a two-dimensional square region $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$, where the fluid dynamics is governed by the Navier-Stokes equations (3) and the control input affects the system through the boundary conditions. The main goal is to obtain a dynamical model for a region of interest $\Omega_R = [0.3878, 0.5102] \times [0.4694\ , 0.5918]$ located within Ω. After the model is at hand, we will also illustrate how this model can be used to realize a control task within the region.

First let us rewrite the Navier-Stokes equations (3) in two dimensions as

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}u + \frac{\partial u}{\partial y}v = \frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{12}$$

$$\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x}u + \frac{\partial v}{\partial y}v = \frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{13}$$

where $q(x,y,t) = [u(x,y,t)\ \ v(x,y,t)] \in \mathbb{R}^2$ is the flow velocity and u and v are components in the longitudinal and lateral directions. We take the viscosity value as $\nu = 0.1$, the initial conditions as

$$u(x,y,0) = v(x,y,0) = 0 \tag{14}$$

and the boundary conditions as

$$u(x,0,t) = u(x,1,t) = 1 \tag{15}$$

$$v(x,0,t) = v(x,1,t) = 0 \tag{16}$$

$$u(0,y,t) = 0, \quad \frac{\partial v}{\partial x}(0,y,t) = 0 \tag{17}$$

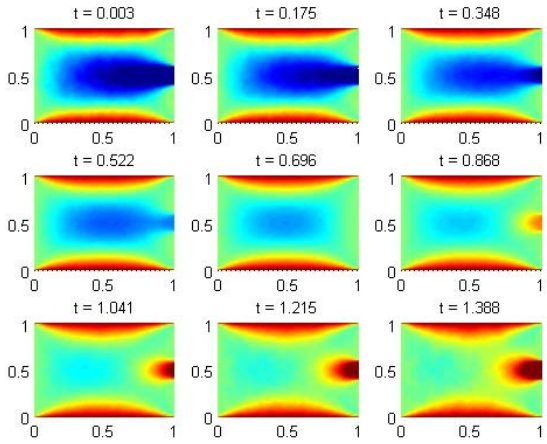

Figure 2. Flow snapshots under chirp excitation (u-component)

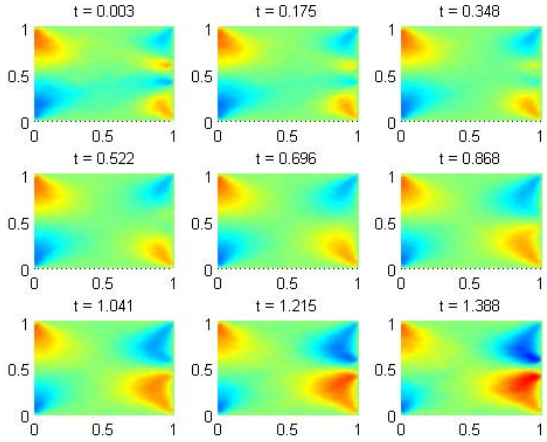

Figure 3. Flow snapshots under chirp excitation (v-component)

$$u(1,y,t) = \begin{cases} 0, & y \in [0, 0.42) \\ \gamma(t), & y \in [0.42, 0.58] \\ 0, & y \in (0.58, 1] \end{cases} \quad (18)$$

$$v(1,y,t) = 0 \quad (19)$$

where $\gamma \in \mathbb{R}$ is the control input.

As the first step of the procedure described in Section III, the Navier-Stokes equations above were simulated using *Navier2d*, a Navier-Stokes CFD solver for MATLAB [12]. Several simulations were carried out under different inputs, including zero-input, chirp signal, square wave, ramp function and white noise. Each simulation was carried out with a time step of $T_s = 0.0014$ seconds for 1000 time steps on a 50×50 uniform grid of the spatial domain. The snapshots for the simulation with chirp signal input are shown in Figures 2 and 3. Next a wavelet decomposition of the snapshots was performed at various levels using different wavelet functions with the help of MATLAB Wavelet Toolbox. Evaluating these decomposition, we have decided to use a two level decomposition using the Daubechies wavelet for the rest of the modeling procedure. This wavelet function is asymmetric with

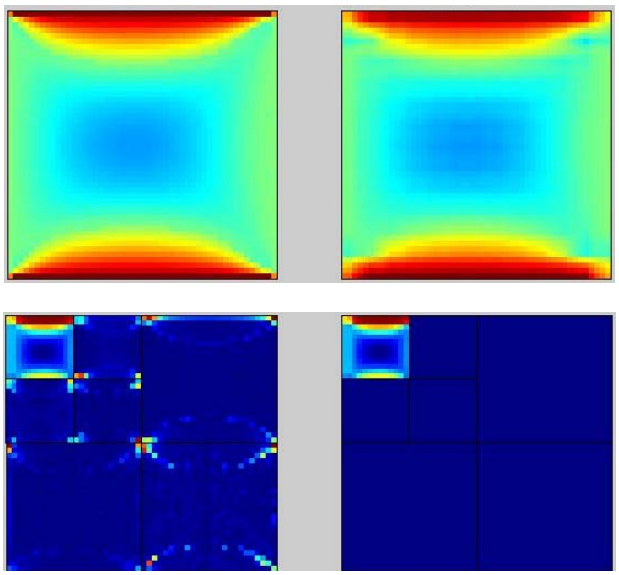

Figure 4. Original snapshot (top left), wavelet coefficients resulting from two level decomposition using Daubechies wavelet (bottom left), thresholded wavelet coefficients (bottom right), snapshot reconstructed from thresholded coefficients (top right).

a near random structure, is orthogonal, produces exact reconstruction, has a finite support area, and the highest number of vanishing moments for a given support width. These properties make the Daubechies wavelet a suitable candidate for representing snapshots taken from fluid flow processes. In addition the availability of fast and efficient methods for obtaining DWT and inverse DWT with the Daubechies wavelet makes it possible to process a high number of snapshots in a short time. Figure 4 shows the u-component of a sample snapshot together with its two level decomposition using the Daubechies wavelet. Also shown in the figure is the result of applying thresholding to the wavelet coefficients. Different values for the threshold T were tested, and it was observed that under the selected level and wavelet function, the thresholded coefficients produce good reconstructions, even for very small values of T. In fact, the reconstruction is satisfactory even for $T = 0$, which is the case shown in the figure. This implies that even if all the detail coefficients are omitted, the approximation coefficients are adequate to reconstruct the snapshot. The results for the v-component of the snapshot were equally satisfactory, and so were the results for the other 999 snapshots. This result is a further justification for the selected wavelet function and the level of decomposition.

Once the wavelet type and the level of decomposition is determined, it is possible to construct the basis vectors Φ_i. To cover the domain of interest Ω_R, it turns out that one needs to use four vectors per direction, making a total of eight basis vectors, which can be defined as in (5)-(7). The functions Υ_i for $i = 1, \ldots, 4$ are shown in Figure 5 where it can be seen that the region of interest Ω_R is contained within the support of these functions. Having obtained the basis vectors Φ_i, it is possible to expand the flow as

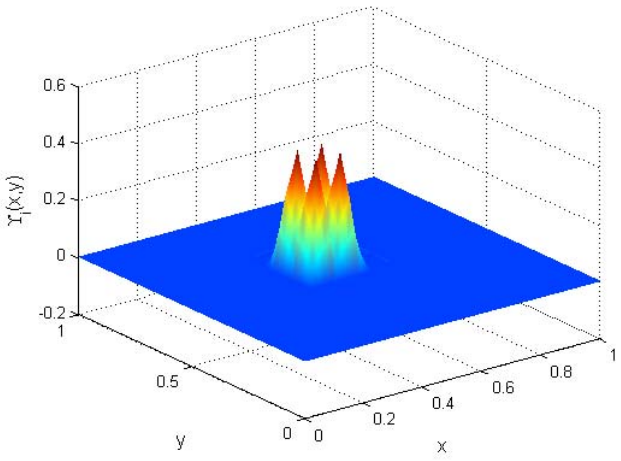

Figure 5. The functions $\{\boldsymbol{\Upsilon}_i\}_1^4$ for constructing the basis vectors $\{\boldsymbol{\Phi}_i\}_1^8$

$$q(x,y,t) = \sum_{i=1}^{8} a_i(t)\Phi_i(x,y) \tag{20}$$

where a_i are the approximation coefficients.

The step after obtaining the basis functions Φ_i is the generation of input-output data for the identification of a state-space dynamical model. Recall that the system output for identification purposes is

$$y(t) = a(t) = [a_1(t)\ \ a_2(t) \ldots a_8(t)]^T \tag{21}$$

which can obtained by wavelet transforming the snapshots of the system under various test inputs and recording the coefficients of interest. The output data resulting from the zero-input case and the chirp signal case are shown in Figure 6. Output data under other input trajectories including square waves, ramp functions and white noise signals have also been obtained and recorded. We use these input-output data to obtain a dynamical system of the form (9)-(10) using subspace system identification methods (N4SID) available through the MATLAB System Identification Toolbox. For this purpose we split the first half of the data for estimation, whereas the second

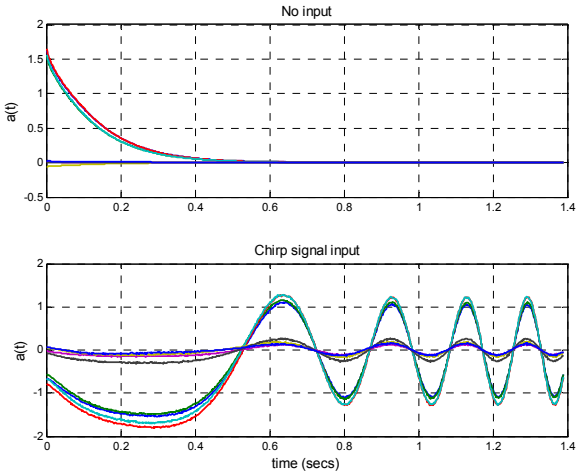

Figure 6. Coefficients obtained from snapshots

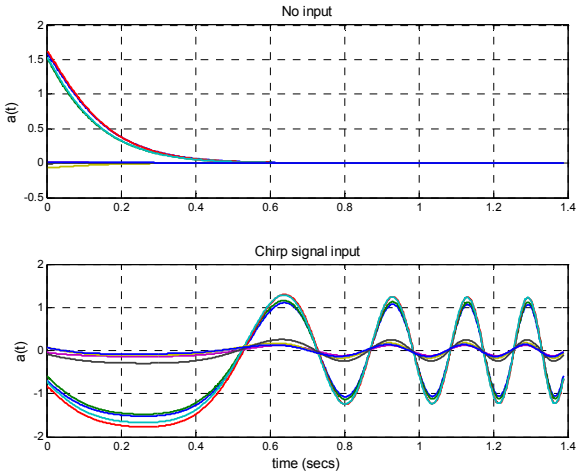

Figure 7. Coefficients obtained from the dynamical model

half is reserved for validation. The function N4SID is then called with its default parameter values. Subsequent trials show that a satisfactory fit to the data can be obtained for a tenth order model, whose response under zero input and under chirp signal input is shown in Figure 7. Comparing with Figure 6, one can see that the responses are very close to each other. The results were similar for other inputs tested as well; thus, one can state that the model constructed is satisfactory in representing dynamics of the spatial region Ω_R of interest.

Undoubtedly, the main purpose for building a dynamical model for the region of interest Ω_R is to carry out a control design task within the region. Let us assume, for the sake of illustration, that the control goal is to regulate the longitudinal velocity of the point $(x_c, y_c) := (0.5, 0.5) \in \Omega_R$. Let us denote this quantity to regulated as y_2, which can be written from (20) as

$$\begin{aligned} y_2(t) = u(x_c, y_c, t) &= \sum_{i=1}^{8} a_i(t)\Phi_{i,u}(x_c, y_c) \\ &=: C'a(t) \end{aligned} \tag{22}$$

where C' is the 1×8 matrix

$$C' := \begin{bmatrix} \Phi_{1,u}(x_c, y_c) & \Phi_{2,u}(x_c, y_c) & \ldots & \Phi_{8,u}(x_c, y_c) \end{bmatrix} \tag{23}$$

Then from (21) and (10)

$$\begin{aligned} y_2 &= C'a = C'(C\xi + D\gamma) \\ &= C'C\xi + C'D\gamma \\ &= C_2\xi + D_2\gamma \end{aligned} \tag{24}$$

where $C_2 := C'C$ and $D_2 := C'D$. Then, augmenting the state dynamics (9) with the output to be regulated we obtain

$$\xi(t + T_s) = A\xi(t) + B\gamma(t) \tag{25}$$

$$y_2(t) = C_2\xi(t) + D_2\gamma(t) \tag{26}$$

which is a single-input single-output system from γ to y_2. Let y_{ref} denote the reference signal to be tracked by y_2. To achieve the desired tracking one may design a compensator K with transfer function

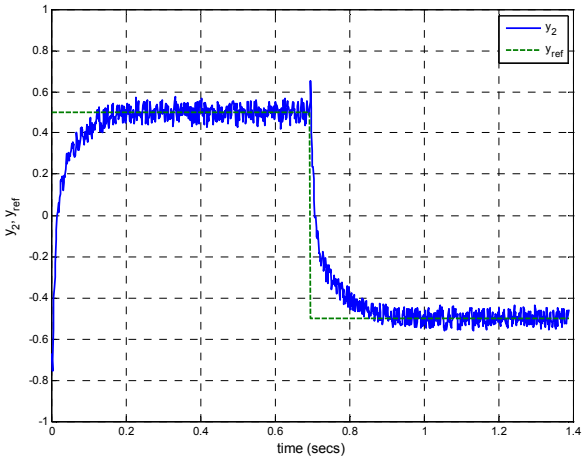

Figure 8. u-velocity of the point (x_c, y_c) and the reference signal y_{ref}

$$K(z) := \frac{\Gamma(z)}{E(z)} \tag{27}$$

where $\Gamma(z)$ is the z-transform of $\gamma(t)$ and $E(z)$ is the z-transform of the tracking error $e(t) := y_{ref}(t) - y_2(t)$. A variety of standard and automated design methods exist for obtaining $K(z)$, including PID tuning techniques, internal model control (IMC) design methods, LQG synthesis, and optimization based design. For the problem at hand, numerous compensators of different orders were designed using these methods with the help of MATLAB Control Systems Toolbox. The best results were obtained for the following third order compensator built using IMC design methods [13, 14]

$$K(z) = -0.649\frac{(z-1.79)(z-0.995)(z-0.976)}{(z-1)\,(z-0.994)\,(z-0.938)} \tag{28}$$

This compensator was applied to the flow problem described by (12)-(19) and CFD simulations were carried out. For the simulations, the reference signal y_{ref} was kept constant at 0.5 for until about $t = 0.7$ seconds, after which it was switched to -0.5. To make the situation more challenging and realistic, we also added disturbances to the input and output of the system. The disturbances applied were in the form of white noise signals with magnitude 0.05, which is 10% of the reference signal. The trajectory of the point $(x_c, y_c) = (0.5, 0.5)$ of interest, together with the reference signal y_{ref} is shown in Figure 8. It can be observed from the figures that the closed-loop system formed with the controller (28) is successful in accomplishing the desired tracking and keeping the velocity of the given point close to the reference signal.

In summary it can be stated that regional dynamical model built using the approach suggested in the paper represents the flow process adequately, and a control design carried out utilizing this model produces satisfactory results when applied to the complex PDE system governing the flow dynamics.

V. Conclusions and Future Works

In this study, a novel method for regional dynamical modeling of flow control problems using wavelet transform is proposed. The approach developed is illustrated on a sample flow control problem governed by the Navier-Stokes PDEs, where the input affects the system through the boundary conditions. The main contribution of this work is to present a systematic method to construct linear dynamical models representing a local spatial region of interest for flow control problems. Future research directions include employing different identification schemes and application of the techniques to different flow control problems.

Acknowledgment

The authors would like to acknowledge the libraries of TOBB University of Economics and Technology for providing valuable resources used in this study.

References

[1] M. Gad-el Hak, *Flow Control - Passive, Active, and Reactive Flow Management*. New York, NY: Cambridge University Press, 2000.

[2] T. Bewley, "Flow control: new challenges for a new Renaissance," *Progress in Aerospace Sciences*, vol. 37, no. 1, pp. 21–58, 2001.

[3] P. Holmes, J. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical System, and Symmetry*. Cambridge: Cambridge University Press, 1996.

[4] L. Sirovich, "Turbulence and the dynamics of coherent structures," *Quarterly of Applied Math.*, vol. XLV, no. 3, pp. 561–590, 1987.

[5] C. W. Rowley, T. Colonius, and R. M. Murray, "Model reduction for compressible flows using POD and Galerkin projection," *Physica D*, vol. 189, no. 1-2, pp. 115–29, 2004.

[6] I. Daubechies and B. Bates, "Ten Lectures on Wavelets," *The Journal of the Acoustical Society of America*, vol. 93, p. 1671, 1993.

[7] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[8] C. Chui, *An Introduction to Wavelets*. Academic Press, 1992.

[9] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley Cambridge Pr, 1996.

[10] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, implementation, applications*. Kluwer Academic Publishers, 1996.

[11] W. Larimore, "Statistical optimality and canonical variate analysis system identification," *Signal Processing*, vol. 52, no. 2, pp. 131–144, 1996.

[12] D. Engwirda, "An unstructured mesh navier-stokes solver," Master's thesis, School of Engineering, University of Sydney, 2005.

[13] I. Chien and P. Fruehauf, "Consider IMC tuning to improve controller performance," *Chem. Eng. Prog*, vol. 86, no. 10, pp. 33–41, 1990.

[14] D. Rlvera, M. Morarl, and S. Skogestad, "Internal Model Control. 4. PID Controller Design," *Industrial Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.