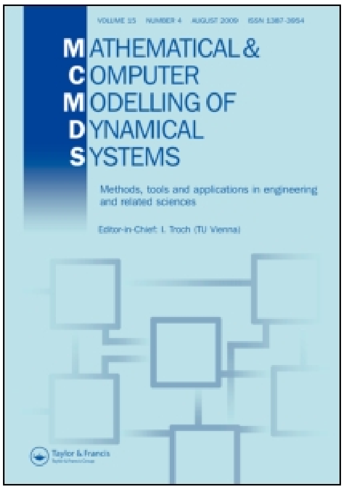

Mathematical and Computer Modelling of Dynamical Systems
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713682513

Feedback flow control employing local dynamical modelling with wavelets

Türker Nazmi Erbil [a]; Coku Kasnakolu [a]
[a] Department of Electrical and Electronics Engineering, TOBB University of Economicsand Technology, Ankara, Turkey

First Published on: 01 September 2009

To cite this Article Erbil, Türker Nazmi and Kasnakolu, Coku(2009)'Feedback flow control employing local dynamical modelling with wavelets',Mathematical and Computer Modelling of Dynamical Systems,99999:1,

To link to this Article: DOI: 10.1080/13873950903187253

URL: http://dx.doi.org/10.1080/13873950903187253

Mathematical and Computer Modelling of Dynamical Systems
i*First*, 2009, 1–21

Feedback flow control employing local dynamical modelling with wavelets

Türker Nazmi Erbil and Coşku Kasnakoğlu*

Department of Electrical and Electronics Engineering, TOBB University of Economics and Technology, Ankara, Turkey

(*Received 25 February 2009; final version received 24 June 2009*)

In this paper, we utilize wavelet transform to obtain dynamical models describing the behaviour 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 that 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 the 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 modelling; wavelet transform

1. 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 of fluid flow. Thus, from a scientific and technological point of view, modelling and understanding fluid flow is an issue of high importance [1,2]. Among extensive research on the topic one can find studies on flow control in aircrafts and airfoils [3,4], control of channel flows [5,6], control of turbulent boundary layers [7], control of combustion instability [8], stabilization of bluff-body flow [9], control of cylinder wakes [10,11], control of cavity flows [12–15] and optimal control of vortex shedding [16,17].

The most common technique in the dynamical modelling of fluid flow is the proper orthogonal decomposition (POD) 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 in the time coefficients of the modes [18–21]. Also worth mentioning are input separation (IS) techniques, which are important extensions to POD [22–24]. These methods

*Corresponding author. Email: kasnakoglu@etu.edu.tr

ISSN 1387-3954 print/ISSN 1744-5051 online

DOI: 10.1080/13873950903187253
http://www.informaworld.com

address the problem that the control input gets embedded into the system coefficients and remedy the issue by producing stand-alone control terms in the dynamics. POD-based methods have been used for the modelling and control of numerous flow applications, including feedback control of cylinder wakes [10,11], control of cavity flows [12–15], optimal control of vortex shedding [17] and the stabilization of the flow over obstacles [25].

Although the above-mentioned approaches do indeed result in finite-dimensional dynamical models, it is still very difficult to perform analysis and design as these models are nonlinear 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 concerned with the dynamical behaviour in a given local region only, so it is of interest to build models whose states can directly be associated with a given spatial region. In this paper, we utilize wavelet transform methods [26–29] to develop such a modelling approach. The ideas in the paper are developed and are organized as follows: Section 2 presents an overview of the wavelet transform and the Navier–Stokes (NS) equations. Section 3 describes the main modelling approach, which is based on obtaining spatially local basis modes using wavelet transform and using subspace identification to construct a model capturing the dynamics of the time coefficients associated with these modes. The proposed approach is illustrated with a flow control case study in Section 4, where the task is to regulate the velocity of a given point inside a square region in which the flow is governed by the NS PDEs. It is first observed that the modes built from wavelet transform can adequately represent the snapshots of the flow. Then the model capturing the dynamics of the time coefficients for these modes is built, and it is seen to produce trajectories sufficiently close to the actual trajectories associated with the snapshots. Next a controller design is carried out using the dynamical system and applied to the actual NS PDEs. It is seen that the controller is successful in achieving the desired regulation. The paper ends with Section 5, which provides conclusions and future work ideas.

2. Background information

2.1. Wavelet transform, reconstruction, multilevel decomposition and thresholding

The wavelet transform is among the most commonly used methods in signal processing on which a large numbers of resources and studies exist [26–29]. 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, that is,

$$C(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\psi^{*}\left(\frac{t-b}{a}\right)\mathrm{d}t \tag{1}$$

where $a \in \mathbb{R}_+$ is the *scale*, $b \in \mathbb{R}_+$ is the *translational value*, $C(a, b)$ are the *wavelet coefficients* and ψ is the *wavelet function* that depends on the wavelet family being used for the process. There are numerous families available for wavelet transform, including biorthogonal nearly Coiflet (BNC), Coiflet–Daubechies–Feauveau, Daubechies, Haar,

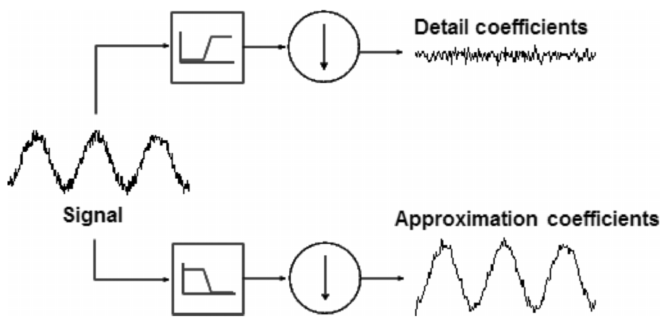

Figure 1. Discrete wavelet transform. The signal is split into approximation/detail coefficients by applying decomposition low-pass/high-pass filters, followed by downsampling.

Mathieu, Legendre, 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 analysed 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 process is illustrated in Figure 1. 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 (2D) 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 2.

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:

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

where X represents the detail coefficients, Y represents the thresholded detail coefficients and $T \in \mathbb{R}_+$ represents 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 is 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.

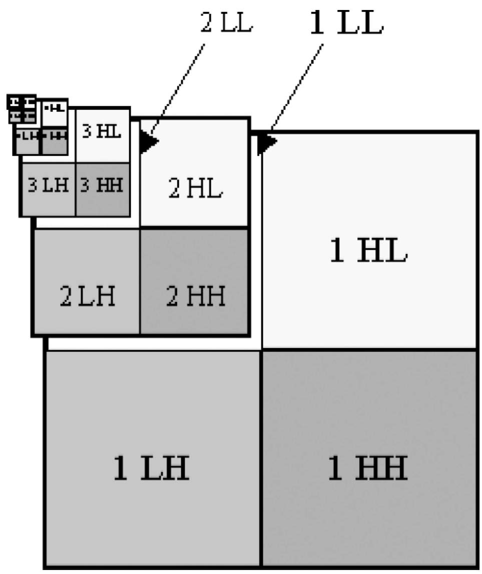

Figure 2. Multilevel 2D wavelet decomposition. The coefficients are labelled as 'N XY', where N is the level, and X, Y denote the filtering operation for columns and rows, respectively. For instance, 2 LH are the second-level wavelet coefficients obtained by applying low-pass filtering/downsampling to the columns of 1 LL, and then applying high-pass filtering/downsampling to the rows.

2.2. Navier–Stokes (NS) equations

The NS PDEs are among the most useful sets of equations to describe the behaviour of fluid flow. These 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. We shall consider the case of non-dimensional, incompressible NS equations

$$\frac{\partial q}{\partial t} + (q \cdot \nabla)q = -\nabla p + \frac{1}{\mathrm{Re}}\Delta q \tag{3}$$

$$\nabla \cdot q = 0 \tag{4}$$

where $\mathrm{Re} \in \mathbb{R}_+$ is the Reynolds number, $p(x, y, t) \in \mathbb{R}$ 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 streamwise and transverse directions. The interested reader is referred to Gad-el Hak [1] for details regarding the NS PDEs.

3. Modelling approach

The regional modelling procedure proposed in this paper consists of the following steps:

The first step in the modelling process is to record 2D instantaneous images, that is snapshots, of the flow. The snapshots can be obtained either from actual physical experiments using techniques, such as particle image velocimetry [30], or from computer data that result from computational fluid dynamics (CFD) simulations of the NS Equations (3) and (4).

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. A 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 is too low, these coefficients will not be able to capture a sufficient amount of the energy in the snapshots, and hence the quality of the representation will degrade. 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 $\Phi_i(x, y)$ in terms of which the flow variable q will be expressed as an expansion of the following form:

$$q(x, y, t) = \sum_{t=1}^{N} a_i(t)\Phi_i(x, y) \tag{5}$$

where a_i are the time coefficients and $N \in 2\mathbb{R}$ is the number of basis functions. Each $\Phi_i(x, y)$ 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 streamwise and the transverse directions, and have the following form:

$$\Phi_i(x, y) = \begin{bmatrix} \Phi_{i,u}(x, y) \\ \Phi_{i,v}(x, y) \end{bmatrix}, \quad i = 1, \ldots, N. \tag{6}$$

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

$$\Phi_{i,u}(x, y) = \begin{cases} \Upsilon_i(x, y), & i = 1, \ldots \frac{N}{2} \\ 0, & i = \frac{N}{2} + 1, \ldots, N \end{cases} \tag{7}$$

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

$$\Phi_{i,v}(x, y) = \begin{cases} 0, & i = 1, \ldots, \frac{N}{2} \\ \Upsilon_{i-\frac{N}{2}}(x, y), & i = \frac{N}{2} + 1, \ldots, N \end{cases} \tag{8}$$

where the functions $\Upsilon_i : \mathbb{R}^2 \to \mathbb{R}$ for $i = 1, \ldots, N/2$ are simply the wavelet function shifted and scaled appropriately, which can be obtained by taking a coefficient matrix that has a value of 1 at the coefficient of interest and is 0 elsewhere, and then inverse transforming. Depending on the location of the wavelet coefficient, the oscillating part of the function Υ_i will be located in a different region of the spatial domain. One must therefore pick a number of suitable Υ_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 Equations (7) and (8). If the wavelet function is orthogonal, then

$$\langle \Phi_i(x, y), \Phi_j(x, y) \rangle = 0, \quad \text{for } i \neq j \tag{9}$$

and the wavelet coefficient $a_i(t)$ in Equation (5) 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 that 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 Equation (5), it is seen that the time variation of the flow is dictated by the coefficients a_i, since the vectors Φ_i are constant with respect to time. Thus the modelling 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:

$$\xi(t + T_s) = A\xi(t) + B\gamma(t), \tag{10}$$

$$y(t) = C\xi(t) + D\gamma(t), \tag{11}$$

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 Equations (10) and (11) using system identification techniques. To construct the data for system identification, various input signals, for example, sine waves, ramp functions and chirp signals, are applied to the system at a sampling period of T_s, and the resulting snapshots are recorded. Applying wavelet transform to these snapshots yields the system output, which consists of the N wavelet coefficients representing the region of interest, that is,

$$y(t) = a(t) = [a_1(t)\ a_2(t) \dots a_N(t)]^T. \tag{12}$$

From the input data $\{\gamma_k\}_{k=1}^M$ and the output data $\{y_k\}_{k=1}^M$, subspace system identification method (N4SID) is used for obtaining the A, B, C and D matrices in Equations (10) and (11). The main idea behind the subspace method is to first estimate the extended observability matrix:

$$O_r = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} \tag{13}$$

for the system from input–output data by direct least-squares-like projection steps. In particular, it is possible to show that an expression of the form

$$Y_r(t_k) = O_r\,\xi(t_k) + S_r\Gamma_r(t_k) + V(t) \tag{14}$$

can be obtained from Equations (10) and (11), where

$$Y_r(t_k) = \begin{bmatrix} y(t_k) \\ y(t_{k+1}) \\ \vdots \\ y(t_{k+r-1}) \end{bmatrix}, \quad \Gamma_r(t_k) = \begin{bmatrix} \gamma(t_k) \\ \gamma(t_{k+1}) \\ \vdots \\ \gamma(t_{k+r-1}) \end{bmatrix} \tag{15}$$

$$S_r = \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{r-2}B & CA^{r-3}B & \cdots & CB & D \end{bmatrix} \tag{16}$$

and $V(t)$ is the contribution because of output noise. The extended observability matrix O_r can then be estimated from Equation (14) by correlating both sides of the equality with quantities that eliminate the term $S_r\Gamma(t_k)$ and make the noise influence from $V(t)$ disappear asymptotically. Once O_r is known, it is possible to determine C and A by using the first block row of O_r and the shift property, respectively. Once A and C are at hand, B and D are estimated using linear least squares on the following expression:

$$y(t_k) = C(zI - A)^{-1}Bu(t_k) + Du(t_k) \tag{17}$$

where Equation (17) is a representation of system Equations (10) and (11) in terms of the time-shift operator z. Details of the subspace method for estimating state-space models can be found in Ljung [31], Van Overschee [32] and Larimore [33].

Remark: Note that the system identification approach considered above differs from the calibration techniques commonly used in flow modelling [34–37]. In the calibration approaches, one first obtains a POD-based reduced order model (ROM) and then adjusts its coefficients to minimize the error between the time coefficients and the states of the model (or their derivatives). In the approach considered in this section, a linear discrete-time model is obtained directly from the input–output data using general-purpose subspace system identification tools (N4SID), without going through an intermediate ROM.

The dynamical regional modelling approach described in this section is best illustrated by means of a case study, which will be presented in the Section 4.

4. Case study: dynamical modelling of a boundary-controlled flow governed by the 2D Navier–Stokes equations

In this example we consider the fluid flow over a 2D square region $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$, where the fluid dynamics is governed by the NS Equations (3) and (4) 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. This choice of Ω_R is without loss of any generality, and the proposed approach can be applied in an identical manner to any other region of interest. After the dynamical model is at hand, we will also illustrate how this model can be used to realize a control task within the region. Let us first rewrite the NS Equations (3) and (4) in two dimensions as

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}u + \frac{\partial u}{\partial y}v = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{18}$$

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}u + \frac{\partial u}{\partial y}v = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{19}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{20}$$

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 streamwise and transverse directions. We take the parameter Re to be Re $= 10$, the initial conditions as

$$u(x,y,0) = v(x,y,0) = 0 \tag{21}$$

and the boundary conditions as

$$u(x,0,t) = u(x,1,t) = 1, \tag{22}$$

$$v(x,0,t) = v(x,1,t) = 0, \tag{23}$$

$$u(0,y,t) = 0, \quad y \in [0, 0.0918) \cup (0.1735, 0.8265)(0.9082, 1], \tag{24}$$

$$\frac{\partial p}{\partial x}(0,y,t) = 0, \quad y \in [0.0918, 0.1735] \cup [0.8265, 0.9082], \tag{25}$$

$$\frac{\partial v}{\partial x}(0,y,t) = 0, \tag{26}$$

$$u(1,y,t) = \begin{cases} 0, & y \in [0, 0.4184) \\ \gamma(t), & y \in [0.4184, 0.5816) \\ 0, & y \in (0.5816, 1], \end{cases} \tag{27}$$

$$v(1,y,t) = 0, \tag{28}$$

where $\gamma \in \mathbb{R}$ is the control input. This example was chosen because it is relatively simple to implement, but at the same time it contains challenges for modelling and control: the problem contains a mixture of Dirichlet- and Neumann-type boundary conditions (corresponding to constant flowing, no-slip, stress-free and outflow-at-fixed-pressure-type boundaries) and the control input γ can induce changes to the system through only a limited segment of the right-hand-side boundary.

As the first step of the procedure described in Section 3, the NS equations above were simulated using *Navier2d*, an NS CFD solver for MATLAB [38]. 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. A chirp signal input is shown in Figure 3, and few snapshots obtained from the CFD simulation with this input are shown in Figures 4 and 5.

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 decompositions, we have decided to use a two-level decomposition using the Daubechies 4 wavelet (db4) for the rest of the modelling procedure. This wavelet function is asymmetric with 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

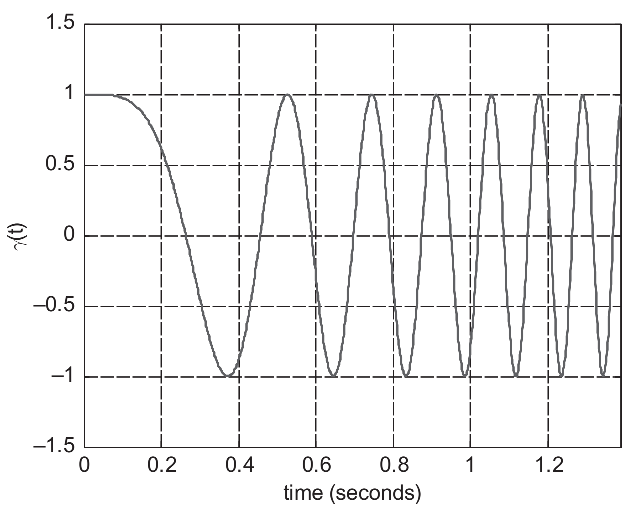

Figure 3. Chirp signal excitation used to obtain the flow snapshots in Figures 4 and 5.

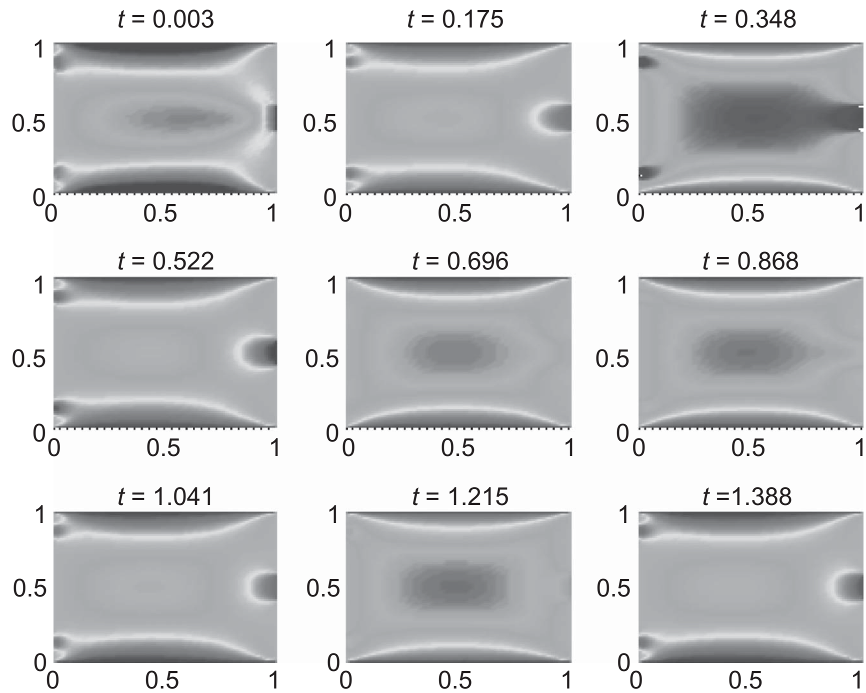

Figure 4. u-Component of the flow snapshots obtained under the chirp excitation shown in Figure 3.

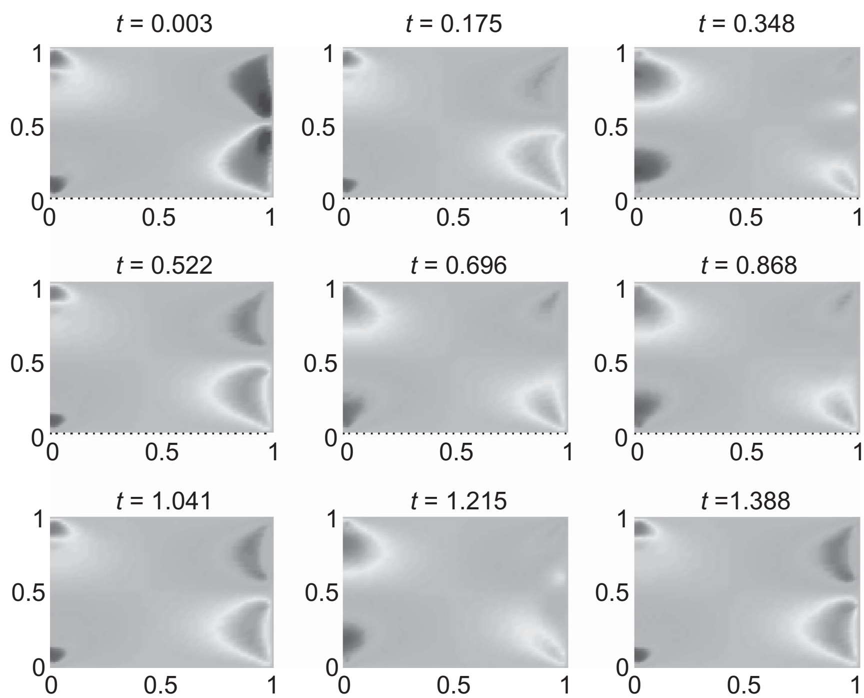

Figure 5. v-Component of the flow snapshots obtained under the chirp excitation shown in Figure 3.

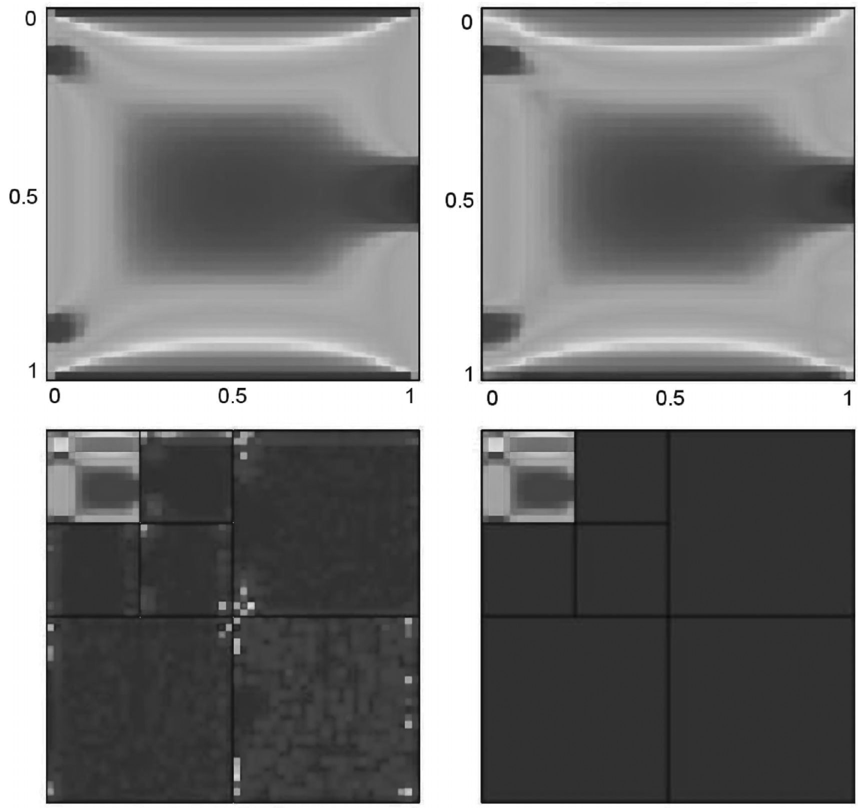

Figure 6. Original snapshot (top left), wavelet coefficients resulting from two-level decomposition using Daubechies 4 wavelet (bottom left), thresholded wavelet coefficients (bottom right), snapshot reconstructed from thresholded coefficients (top right).

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 6 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.

As an additional justification for choosing the db4 wavelet and a two-level decomposition, we have applied the same operations of transforming, thresholding and reconstruction using other compactly supported orthogonal wavelets and decomposition levels. The wavelets tested are Coiflet (coif) 1, 2, 3, 4, 5; Daubechies (db) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10; and Symlet (sym) 2, 3, 4, 5, 6, 7, 8; and the decomposition levels tested are 1, 2, 3 and 4. The numbers next to the wavelet name indicate the *order* of the wavelet, which determines certain characteristics such as the support width, the filter length and the number of vanishing moments. For illustration

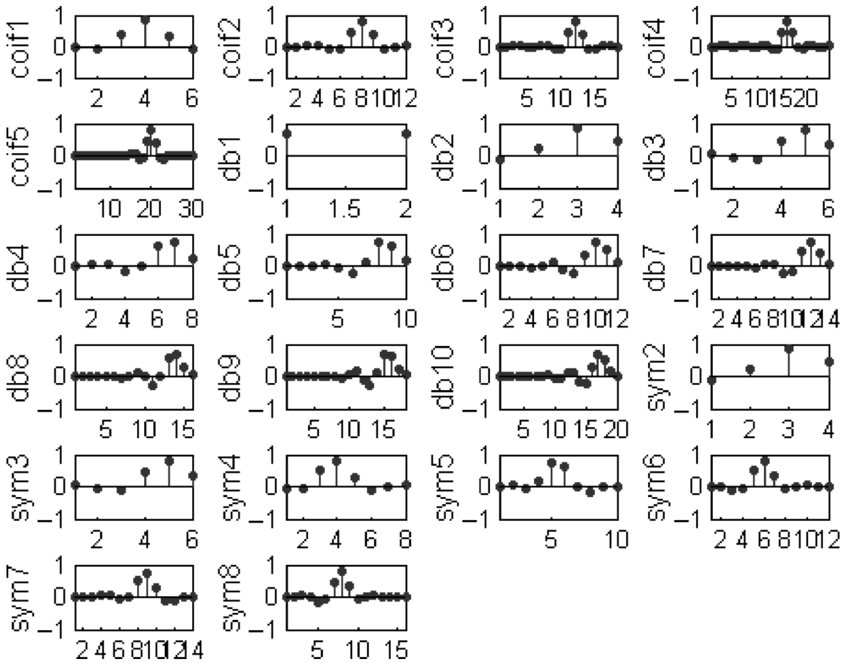

Figure 7. Decomposition low-pass filters (L) for different wavelet functions used.

purposes, the coefficients (i.e. impulse responses) of the decomposition low-pass and high-pass filters for the wavelets utilized are shown in Figures 7 and 8. The reader interested in full details of these wavelets is referred to Daubechies [26].[1] Tables 1 and 2 show various metrics to evaluate the performance of the wavelet functions and decomposition levels. The columns denote the name of the wavelet, decomposition level, number of coefficients after thresholding out the details, average percent energy over all snapshots of the energy captured in the u-direction, average percent energy in the v-direction, average mean squared error (MSE) in the u-direction and average MSE in the v-direction, respectively. Recall that the desirable result is to have a good reconstruction (i.e. high percent energy and low MSE) with a small number of approximation coefficients. Hence we classify the result of each metric into three classes and mark the cells of the tables with symbols to serve as visual aids: ✓ denotes a desirable value, ! denotes a value that is borderline tolerable and × denotes an unacceptable value. For the number of approximation coefficients we set the class boundaries as 300 and 600, for energy percentage we set the boundaries as 93 and 97% and for MSE we set the boundaries as 0.1 and 0.2. Observing the tables, one can see that the values for the two-level wavelet decomposition with the db4 wavelet (shown highlighted in Table 1) are within desired limits for all metrics. The performances of coif1, db2, db5, db6, db7, db8, sym5, sym6, sym7 and sym8 wavelets with two-level decomposition are also acceptable but not as good as those of the two-level db4 decomposition.

Once the wavelet type and the level of decomposition are 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 Equations (6) and (8). The functions γ_i for $i = 1,\ldots,4$ are shown in Figure 9, where the region of interest Ω_R is contained within the support of these functions.

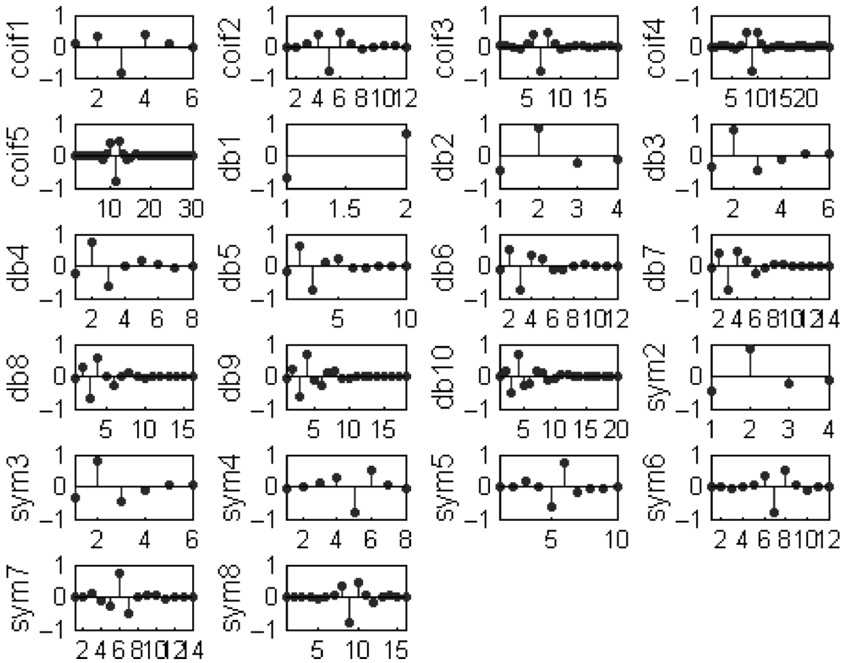

Figure 8. Decomposition high-pass filters (H) for different wavelet functions used.

Table 1. Performance in reconstructing flow snapshots for different wavelet functions and levels of decomposition.

Name	Level	#Coefs	Energy (U)	Energy (V)	MSE (U)	MSE (V)
sym8	1	× 1024	✓ 99.0023563	✓ 99.00235625	✓ 0.02579	✓ 0.025794
coif1	2	✓ 256	! 96.2290678	! 96.22906781	! 0.05945	! 0.059446
coif2	2	! 400	! 93.6503965	! 93.65039646	! 0.05605	! 0.056054
coif3	2	× 625	! 95.6123898	! 95.61238981	✓ 0.04772	✓ 0.047715
coif4	2	× 841	! 96.2467795	! 96.24677947	✓ 0.04742	✓ 0.047421
coif5	2	× 1156	✓ 97.0655349	✓ 97.06553487	✓ 0.04652	✓ 0.046525
db1	2	✓ 169	! 94.3143386	! 94.31433858	! 0.09489	! 0.094891
db2	2	✓ 196	! 96.3309065	! 96.33090651	! 0.05446	! 0.054457
db3	2	✓ 256	! 95.7813833	! 95.78138326	✓ 0.04933	✓ 0.049332
db4	2	✓ 289	✓ 97.1604309	✓ 97.16043094	✓ 0.04842	✓ 0.048418
db5	2	! 361	! 96.0196462	! 96.01964617	! 0.05994	! 0.05994
db6	2	! 400	✓ 98.3429988	✓ 98.34299883	✓ 0.04055	✓ 0.040552
db7	2	! 484	✓ 98.2234596	✓ 98.22345962	✓ 0.0415	✓ 0.041497
db8	2	! 529	✓ 97.6434158	✓ 97.64341578	! 0.05399	! 0.053988
db9	2	× 625	✓ 98.2710506	✓ 98.27105062	✓ 0.04474	✓ 0.044737
db10	2	× 676	✓ 97.5612652	✓ 97.56126522	✓ 0.04269	✓ 0.042694
sym2	2	✓ 196	! 96.3309065	! 96.33090651	! 0.05446	! 0.054457
sym3	2	✓ 256	! 95.7813833	! 95.78138326	✓ 0.04933	✓ 0.049332
sym4	2	✓ 289	× 92.9023109	× 92.90231093	! 0.06068	! 0.060682
sym5	2	! 361	! 93.1454346	! 93.14543458	! 0.05686	! 0.056865
sym6	2	! 400	! 93.4253015	! 93.42530154	! 0.06135	! 0.061347
sym7	2	! 484	! 94.0364659	! 94.03646592	! 0.05543	! 0.055425
sym8	2	! 529	! 94.3909896	! 94.39098957	✓ 0.0459	✓ 0.045901

Table 2. Performance in reconstructing flow snapshots for different wavelet functions and levels of decomposition (continued).

Name	Level	#Coefs	Energy (U)	Energy (V)	MSE (U)	MSE (V)
coif1	3	✓ 100	! 93.061951	! 93.06195097	! 0.08709	! 0.087092
coif2	3	✓ 225	× 83.1642388	× 83.1642388	× 0.10508	× 0.105079
coif3	3	! 441	× 79.8074672	× 79.80746719	× 0.11197	× 0.11197
coif4	3	× 676	× 83.9565327	× 83.95653271	× 0.117	× 0.116996
coif5	3	× 961	× 89.9811197	× 89.98111972	! 0.09655	! 0.096546
db1	3	✓ 49	× 88.0438509	× 88.04385093	× 0.1613	× 0.161305
db2	3	✓ 64	× 92.1584426	× 92.1584426	! 0.09112	! 0.09112
db3	3	✓ 100	! 93.1415122	! 93.14151223	! 0.09269	! 0.092691
db4	3	✓ 144	× 84.5527122	× 84.55271219	! 0.09891	! 0.098912
db5	3	✓ 196	× 87.6134052	× 87.61340515	× 0.10896	× 0.108965
db6	3	✓ 225	! 93.2901932	! 93.29019319	! 0.06912	! 0.069116
db7	3	✓ 289	× 90.8154361	× 90.81543614	× 0.10576	× 0.105764
db8	3	! 361	! 95.4861172	! 95.48611721	! 0.06996	! 0.069962
db9	3	! 441	× 92.3276191	× 92.32761906	! 0.0939	! 0.0939
db10	3	! 484	! 94.3985082	! 94.39850818	! 0.08374	! 0.083738
sym2	3	✓ 64	× 92.1584426	× 92.1584426	! 0.09112	! 0.09112
sym3	3	✓ 100	! 93.1415122	! 93.14151223	! 0.09269	! 0.092691
sym4	3	✓ 144	! 93.463259	! 93.46325901	! 0.09018	! 0.090178
sym5	3	✓ 196	× 83.5776873	× 83.57768732	! 0.08246	! 0.082458
sym6	3	✓ 225	× 78.0425283	× 78.04252827	× 0.12285	× 0.122845
sym7	3	✓ 289	× 77.4950346	× 77.49503457	! 0.09273	! 0.09273
sym8	3	! 361	× 74.4244992	× 74.42449923	× 0.12285	× 0.122847
coif1	4	✓ 49	× 90.0126135	× 90.01261346	× 0.14948	× 0.149485
coif2	4	✓ 169	× 86.5875342	× 86.58753422	× 0.16425	× 0.164251
coif3	4	! 361	× 63.4987778	× 63.4987778	× 0.19879	× 0.198786
coif4	4	! 576	× 55.9462443	× 55.94624432	× 0.15478	× 0.154776
coif5	4	× 900	× 64.7796876	× 64.77968763	× 0.19514	× 0.l95143
db1	4	✓ 16	× 80.2178489	× 80.21784894	× 0.26108	× 0.261078
db2	4	✓ 25	× 85.2158431	× 85.2158431	× 0.16762	× 0.167617
db3	4	✓ 49	× 91.525582	× 91.52558201	× 0.14782	× 0.147821
db4	4	✓ 81	× 86.9190587	× 86.9190587	× 0.15327	× 0.153271
db5	4	✓ 121	× 84.7852631	× 84.78526308	× 0.13724	× 0.137236
db6	4	✓ 169	× 75.2976566	× 75.29765657	× 0.17034	× 0.170345
db7	4	✓ 225	× 69.3504698	× 69.35046981	× 0.17272	× 0.172719
db8	4	✓ 289	× 60.667004	× 60.667004	× 0.15913	× 0.159134
db9	4	! 361	× 69.183353	× 69.18335297	× 0.19911	× 0.199112
db10	4	! 400	× 77.3505764	× 77.35057637	× 0.16699	× 0.166987
sym2	4	✓ 25	× 85.2158431	× 85.2158431	× 0.16762	× 0.167617
sym3	4	✓ 49	× 91.525582	× 91.52558201	× 0.14782	× 0.147821
sym4	4	✓ 81	! 93.0473145	! 93.0473145	× 0.13496	× 0.134957
sym5	4	✓ 121	× 87.6403668	× 87.64036679	× 0.13749	× 0.137493
sym6	4	✓ 169	× 88.3161985	× 88.31619853	× 0.15165	× 0.151654
sym7	4	✓ 225	× 77.0244315	× 77.62443152	× 0.16308	× 0.163077
sym8	4	✓ 289	× 67.6138562	× 67.61385625	× 0.17221	× 0.172209

At this point it will be helpful to take a digression and present a comparison with the basis vectors that would be obtained using the POD method [18–21], which is the most common approach used in the literature for fluid flow modelling. Recall from Equation (7) that $\Phi_{i,u} = \gamma_i$ for $i = 1,\ldots,4$ and note from Figure 9 that the support of each γ_i is a compact spatial region. Hence the coefficient a_i provides time information regarding this compact spatial region only. If the basis vectors had been obtained using POD (e.g. as it was done in

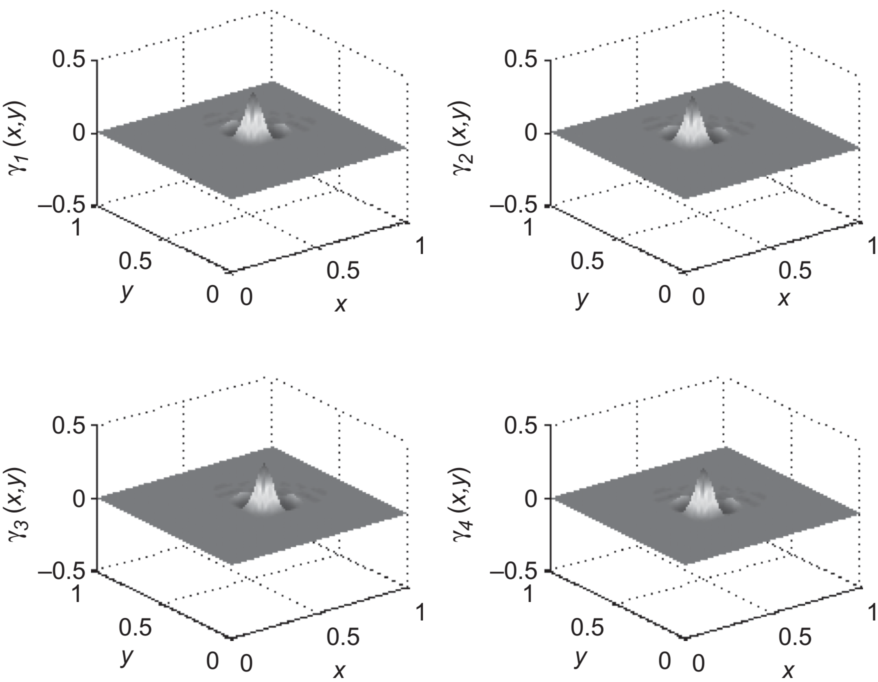

Figure 9. The functions $\{\Upsilon_i\}_1^4$ for constructing the basis vectors $\{\Phi_i\}_1^8$.

Kasnakoglu [24]), then $\Phi_{i,u}$ for $i = 1,\ldots,4$ would be of the form shown in Figure 10. Note that the support of each $\Phi_{i,u}$ is spread out to the entire flow domain. Hence the time variation of coefficient a_i implies a change in the whole flow domain, and it is not possible to link a given coefficient a_i with a particular region of the flow domain. These arguments apply for $\Phi_{i,v}$ as well. This is an important shortcoming that makes POD unsuitable for building regional dynamical models and one of the major reasons for constructing the wavelet-based approach in this paper.

Having obtained the basis vectors Φ_i, it is possible to expand the flow as

$$q(x, y, t) = \sum_{i=1}^{8} a_i(t)\Phi_i(x, y), \tag{29}$$

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

$$y(t) = a(t) = [a_1(t)\ a_2(t) \ldots a_8(t)]^T, \tag{30}$$

which can be 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 11. 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 Equations (10) and (11) using subspace system identification methods (N4SID) available through the MATLAB System Identification Toolbox. For this purpose

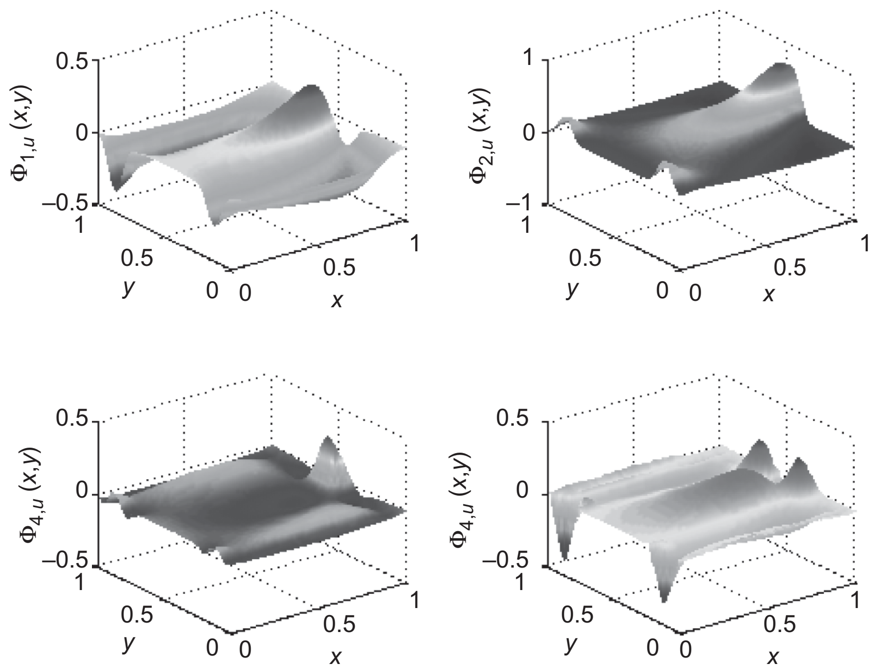

Figure 10. u-Components of the basis vectors $\{\Phi_i\}_1^4$ obtained by POD.

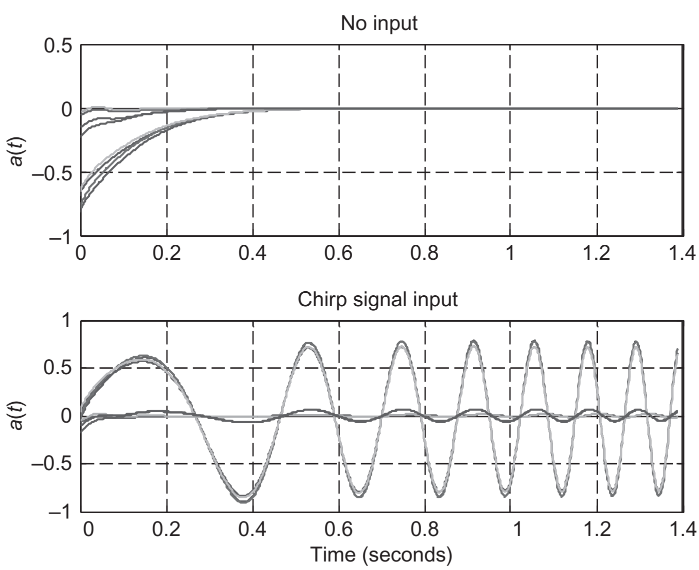

Figure 11. Coefficients obtained from snapshots under chirp excitation.

we split the first half of the data for estimation, whereas the second half is reserved for validation. Subsequent trials show that a satisfactory fit to the data can be obtained for an eight-order model, whose response under zero input and under chirp signal input is shown in Figure 12. Comparing with Figure 11, one can see that the responses are very close to

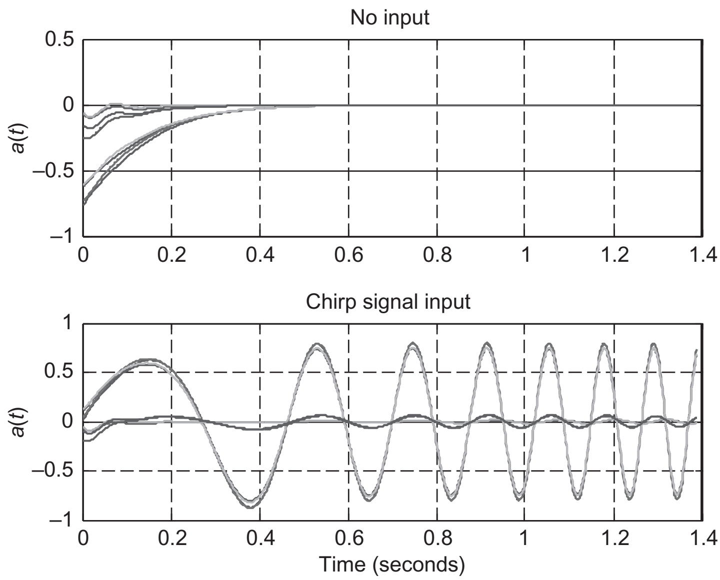

Figure 12. Coefficients obtained from the dynamical model under chirp excitation.

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 streamwise velocity of the point $(x_c, y_c) := (0.5, 0.5) \in \Omega_R$. Let us denote this quantity to be regulated as y_2, which can be written from Equation (29) as

$$y_2(t) = u(x_c, y_c, t) = \sum_{i=1}^{8} a_i(t)\Phi_{i,u}(x_c, y_c) =: C'a(t), \tag{31}$$

where C' is the 1×8 matrix

$$C' := \left[\Phi_{1,u}(x_c, y_c)\ \Phi_{2,u}(x_c, y_c) \cdots \Phi_{8,u}(x_c, y_c)\right]. \tag{32}$$

Then from Equations (30) and (11)

$$y_2 = C'a = C'(C\xi + D\gamma) = C'C\xi + C'D\gamma = C_2\xi + D_2\gamma \tag{33}$$

where $C_2 := C'C$ and $D_2 := C'D$. Then, augmenting the state dynamics (10) with the output to be regulated we obtain

$$\xi(t + T_s) = A\xi(t) + B\gamma(t) \tag{34}$$

$$y_2(t) = C_2\xi(t) + D_2\gamma(t), \tag{35}$$

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

$$K(z) := \frac{\Gamma(z)}{E(z)} \tag{36}$$

where $\Gamma(z)$ is the z-transform of $\gamma(t)$ and $E(z)$ is the z-transform of the tracking error $e(t) := y_{\mathrm{ref}}(t) - y_2(t)$. A variety of standard and automated design methods exist for obtaining $K(z)$, including proportional integral derivative (PID) tuning techniques, internal model control (IMC) design methods, linear quadratic Gaussian (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 [39,40]

$$K(z) = \frac{0.1194z^3 - 0.1159z^2 - 0.1193z + 0.116}{z^3 - 2.968z^2 + 2.936z - 0.9681}. \tag{37}$$

This compensator was applied to the flow problem described by (18)–(28) 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 the 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 snapshots resulting from closed-loop operation are shown in Figures 13 and 14, and 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 15. It can be observed from the figures that the closed-loop system formed with the controller (37) is successful in accomplishing the desired tracking and keeping the velocity of the

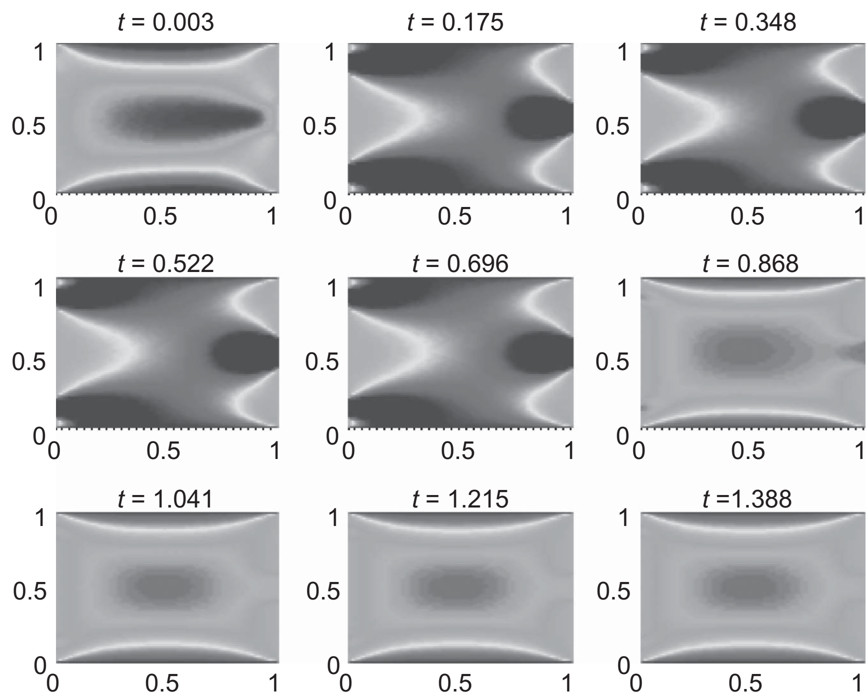

Figure 13. Flow snapshots of the system under closed-loop operation (u-component).

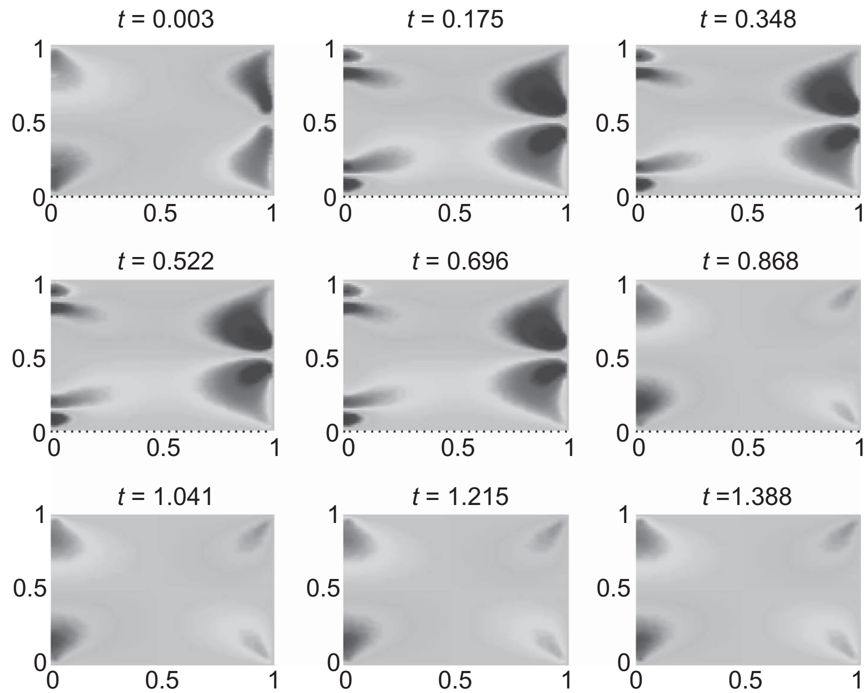

Figure 14. Flow snapshots of the system under closed-loop operation (v-component).

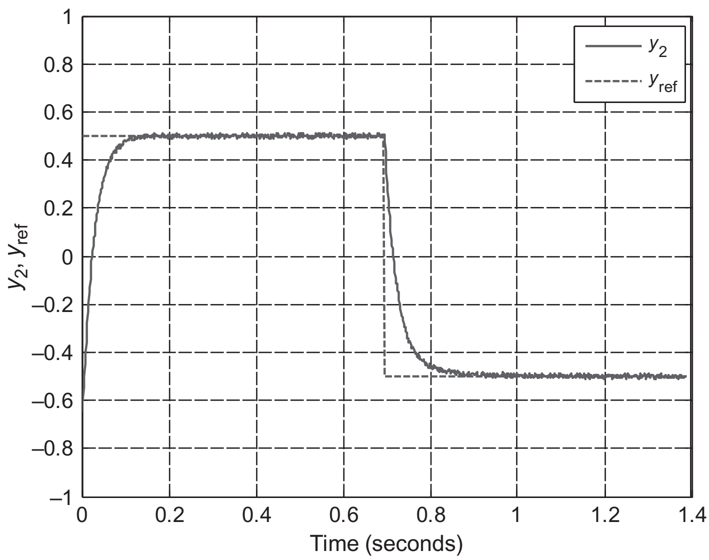

Figure 15. Streamwise velocity of the point (x_c, y_c) under closed-loop operation, and the reference signal y_{ref} to be tracked.

given point close to the reference signal. The minor oscillation about the reference signal is acceptable and is attributable to input and output noises, as well as the unmodelled dynamics resulting from representing an infinite-dimensional nonlinear PDE system with a finite-dimensional linear model.

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.

5. Conclusions and future works

In this study, a novel method for regional dynamical modelling of flow control problems using wavelet transform is proposed. First snapshots of the flow are collected, from where the wavelet family and the decomposition level to be used are determined. Next a set of basis vectors whose support cover the region of interest are constructed from the wavelet function. The flow snapshots are expanded in terms of these basis vectors, where the time variation is determined from the wavelet coefficients. Defining these coefficients as the system output, numerous input signals are applied to the system to construct a sufficient number of input–output data. Subspace identification methods are used to build a discrete-time state space model that best represents the data. The approach developed is illustrated on a sample flow control problem governed by the NS PDEs, where the input affects the system through the boundary conditions. A dynamical model for the given region of interest is built using the techniques proposed in the paper, and it is shown that it adequately represents the flow snapshots obtained from CFD simulations. Utilizing this model, a compensator is designed to regulate the streamwise velocity of a point within the region. It is seen through CFD simulations that the closed-loop system satisfactorily tracks a given reference in the presence of input and output noise signals.

The main contribution of this work is to present a systematic method to construct dynamical models representing a local spatial region of interest for flow control problems. Currently there exist other approaches in the literature for dynamical modelling of flow processes, the most common of which are the POD-based methods. These approaches do produce dynamical models; however, it is hard to utilize these models in analysis and control design since the models resulting are nonlinear in nature. Another issue associated with these models is that the basis vectors are spread out to the entire flow domain; hence one cannot associate the time coefficients of these vectors with the dynamics of a specific region of interest. The approach suggested in this paper remedies both of these problems and therefore is of significance for flow control problems where linear and spatially local models are sought.

Future research directions include employing different identification schemes and application of the techniques to different flow control problems.

Acknowledgements

The authors thank the libraries of TOBB University of Economics and Technology for providing valuable resources used in this study.

Note

1. See Chapter 6 for Daubechies wavelets and Chapter 8 for Symlets and Coiflets.

References

[1] M. Gad-el Hak, *Flow Control – Passive, Active, and Reactive Flow Management*, Cambridge University Press, New York, NY, 2000.
[2] T.R. Bewley, *Flow control: new challenges for a new renaissance*, Prog. Aerosp. Sci. 37 (2001), pp. 21–58.
[3] R.D. Joslin, *Aircraft laminar flow control*, Annu. Rev. Fluid Mech. 30 (1998), pp. 1–29.
[4] J. Wu, X. Lu, A.G. Denny, M. Fan, and J.M. Wu, *Post-stall flow control on an airfoil by local unsteady forcing*, J. Fluid Mech. 371 (1998), pp. 21–58.
[5] O.M. Aamo, M. Krstic, and T.R. Bewley, *Control of mixing by boundary feedback in 2D channel flow*, Automatica 39 (2003), pp. 1597–1606.
[6] L. Baramov, O.R. Tutty, and E. Rogers, *H-infinity control of non-periodic two-dimensional channel flow*, IEEE Trans. Control Syst. Technol. 12 (2004), pp. 111–122.
[7] J. Kim, *Control of turbulent boundary layers*, Phys. Fluids 15 (2003), pp. 1093.
[8] A. Banaszuk, K.B. Ariyur, M. Krstic, and C.A. Jacobson, *An adaptive algorithm for control of combustion instability*, Automatica 40 (2004), pp. 1965–1972.
[9] K. Cohen, S. Siegel, T. McLaughlin, E. Gillies, and J. Myatt, *Closed-loop approaches to control of a wake flow modeled by the Ginzburg–Landau equation*, Comput. Fluids 34 (2005), pp. 927–949.
[10] J. Gerhard, M. Pastoor, R. King, B.R. Noack, A. Dillmann, M. Morzynski, and G. Tadmor. *Model-based control of vortex shedding using low-dimensional Galerkin models*. 33rd AIAA Fluid Conference and Exhibit, Jun 2003. AIAA Paper 2003–4262.
[11] B. Protas, *Linear feedback stabilization of laminar vortex shedding based on a point vortex model*, Phys. Fluids 16 (2004), pp. 4473.
[12] C.W. Rowley, T. Colonius, and R.M. Murray, *Model reduction for compressible flows using POD and Galerkin projection*, Phys. D 189 (2004), pp. 115–129.
[13] K. Fitzpatrick, Y. Feng, R. Lind, A.J. Kurdila, and D.W. Mikolaitis, *Flow control in a driven cavity incorporating excitation phase differential*, J. Guid. Control Dyn. 28 (2005), pp. 63–70.
[14] M. Samimy, M. Debiasi, E. Caraballo, A. Serrani, X. Yuan, J. Little, and J.H. Myatt, *Feedback control of subsonic cavity flows using reduced-order models*, J. Fluid Mech. 579 (2007), pp. 315–346.
[15] E. Caraballo, C. Kasnakoglu, A. Serrani, and M. Samimy, *Control input separation methods for reduced-order model-based feedback flow control*, AIAA J. 46 (2008), pp. 2306–2322.
[16] W.R. Graham, J. Peraire, and K.Y. Tang, *Optimal control of vortex shedding using low-order models. Part I-Open-loop model development*, Int. J. Numer. Methods Eng. 44 (1999), pp. 945–972.
[17] 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. ASME. J. Fluids Eng. 123 (2001), pp. 612–618.
[18] P. Holmes, J.L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical System, and Symmetry*, Cambridge University Press, Cambridge, 1996.
[19] L. Sirovich, *Turbulence and the dynamics of coherent structures*, Quart. Appl. Math. XLV (1987), pp. 561–590.
[20] B.R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, *A hierarchy of low-dimensional models for the transient and post-transient cylinder wake*, J. Fluid Mech. 497 (2003), pp. 335–363.
[21] B.R. Noack, P. Papas, and P.A. Monketwitz, *The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows*, J. Fluid Mech. 523 (2005), pp. 339–365.
[22] R.C. Camphouse, *Boundary feedback control using proper orthogonal decomposition models*, J. Guid. Control Dyn. 28 (2005), pp. 931–938.
[23] M.O. Efe and H. Ozbay, *Low dimensional modelling and Dirichlet boundary controller design for Burgers equation*, Int. J. Control 77 (2004), pp. 895–906.
[24] C. Kasnakoglu, A. Serrani, and M.O. Efe, *Control input separation by actuation mode expansion for flow control problems*, Int. J. Control 81 (2008), pp. 1475–1492.
[25] C. Kasnakoglu, R.C. Camphouse, and A. Serrani, *Reduced-order model-based feedback control of flow over an obstacle using center manifold methods*, J. Dyn. Syst. Meas. Control 131 (2009), pp. 011011.
[26] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial Mathematics, Philadelphia, PA, 1992.
[27] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, 1999.
[28] C.K. Chui, *An Introduction to Wavelets*, Academic Press, Boston, MA, 1992.

[29] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley Cambridge Press, Wellesley, MA, 1996.
[30] M. Raffel, C.E. Willert, and J. Kompenhans, *Particle Image Velocimetry: A Practical Guide*, Springer Verlag, Berlin, Heidelberg, 1998.
[31] L. Ljung, *System Identification: Theory for the User*, PTR Prentice Hall, Upper Saddle River, NJ, 1999.
[32] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, implementation, applications*, Kluwer Academic Publishers, Dordrecht, 1996.
[33] W.E. Larimore, *Statistical optimality and canonical variate analysis system identification*, Signal Process. 52 (1996), pp. 131–144.
[34] M. Bergmann, L. Cordier, and J.P. Brancher, *Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model*, Phys. Fluids 17 (2005), pp. 097101.
[35] M. Couplet, C. Basdevant, and P. Sagaut, *Calibrated reduced-order POD-Galerkin system for fluid flow modelling*, J. Comput. Phys. 207 (2005), pp. 192–220.
[36] B. Galletti, A. Bottaro, C.H. Bruneau, and A. Iollo, *Accurate model reduction of transient and forced wakes*, Eur. J. Mech. B Fluids 26 (2007), pp. 354–366.
[37] R. Bourguet, M. Braza, and A. Dervieux, *Reduced-order modeling for unsteady transonic flows around an airfoil*, Phys. Fluids 19 (2007), pp. 111701.
[38] D. Engwirda. *An unstructured mesh Navier-Stokes solver*. Master's thesis, School of Engineering, University of Sydney, 2005.
[39] I.L. Chien and P.S. Fruehauf, *Consider IMC tuning to improve controller performance*, Chem. Eng. Prog. 86 (1990), pp. 33–41.
[40] D.E. Rlvera, M. Morarl, and S. Skogestad, *Internal model control. 4. PID controller design*, Ind. Eng. Chem. Process Des. Dev. 25 (1986), pp. 252–265.