Transactions of the Institute of Measurement and Control 0, 0 (2009) pp. 1–21

Modelling flow control problems under diverse operating conditions by identification and blending of multiple systems

Coşku Kasnakoğlu

Department of Electrical and Electronics Engineering, TOBB University of Economics and Technology, 06560 Ankara, Turkey

In this paper, a dynamical modelling procedure for fluid flow control problems is proposed. The resulting model is simple in that it consists of a number of linear time-invariant (LTI) systems, but powerful in that it is capable of representing diverse operating conditions within a given flow envelope. The procedure makes use of snapshots of the flow process, which are obtained from experiments or computational fluid dynamics (CFD) simulations. A proper orthogonal decomposition (POD) expansion of the flow is computed from snapshots, and the time coefficients of the expansion are coupled with the input values to form the estimation data. A linear state-space system representing the time coefficients is obtained using subspace system-identification methods. The procedure is repeated for a number of operating points, called breakpoints, which are characterized by one or more flow parameters of interest. The dynamical models obtained at the breakpoints are fused using the output-blending technique. The modelling procedure is illustrated with a flow control case study, where the flow dynamics are governed by the Navier–Stokes equations, the flow parameter of interest is the kinematic viscosity, and the control goal is to regulate the velocity of a given point inside the flow domain.

Key words: dynamical modelling; flow control; fluid flow; multiple-operating conditions.

Address for correspondence: Coşku Kasnakoğlu, Department of Electrical and Electronics Engineering, TOBB University of Economics and Technology, 06560 Ankara, Turkey.
E-mail: kasnakoglu@etu.edu.tr
Figures 2–8 appear in color online: http://tim.sagepub.com

 10.1177/0142331209347650

1. Introduction

The term fluid refers to substances that continually flow under shear stress, and these substances are a crucial part of our daily life. All gases, liquids and plasmas are fluids, and even some solids exhibiting plasticity can be classified as fluids. The flow of fluids is observed continuously around us, including the air flow over the body of an aircraft or an automobile, petroleum and natural gas flow in pipelines, water flow around the hull of a ship or submarine, and the motion of clouds in the atmosphere. The interdisciplinary field concerned with understanding and influencing the flowing behaviour of fluids is referred to as flow control, and is an area of high significance from scientific, industrial and technological perspectives (Gad-el Hak, 2000). Among extensive studies on the subject, one can find research on the control of: airfoil and airplane-related flows (Joslin, 1998), channel flows (Baramov *et al*., 2004), boundary layers (Kim, 2003), combustion instability (Banaszuk *et al*., 2004), bluff-body flow
2 (Dohen *et al*., 2005), cylinder wakes (Noack *et al*., 2005), cavity flows (Caraballo *et al*., 2008) and vortex shedding (Singh *et al*., 2001).

The behaviour of fluids is most commonly described by the Navier–Stokes equations, which are a set of partial differential equations (PDEs) based on continuity (conservation of mass), conservation of momentum and conservation of energy (Acheson, 1990; Batchelo, 2000). The Navier–Stokes PDEs describe the fluid flow quite accurately and are indispensable for the field of computational fluid dynamics (CFD), a key branch of fluid mechanics, which uses numerical methods and algorithms to conduct simulations for fluid flow problems (Hirsch, 2007; Versteeg and Malalasekera, 2007). However these PDEs are of very high complexity, involve both time and spatial derivatives, contain multiple non-linear terms, and the effect of the control input is not explicit in the equations. These issues make any mathematical analysis and manipulation on the Navier–Stokes PDEs extremely difficult, and to date, the direct use of these PDEs is mostly limited to carrying out CFD simulations. Given that the Navier–Stokes PDEs offer little assistance when it comes to theoretical analysis and control design, a fundamental research direction in flow control is to obtain low-order and control-oriented mathematical models to describe the dynamics of flow process.

The most common low-order mathematical models used in flow control are the so-called Galerkin models (or Galerkin systems), which are obtained by a series of simplification steps on the Navier–Stokes PDEs (Holmes *et al*., 1996; Sirovich, 1987). Important enhancements to these techniques have also been proposed recently based on the concept of input separation (IS) (Efe and Ozbay, 2004; Kasnakoğlu *et al*., 2008). Galerkin models have been used in various flow control applications, including cavity flow (Caraballo *et al*., 2008), vortex shedding (Singh *et al*., 2001), cylinder wake (Noack *et al*., 2005) and flow over obstacles (Kasnakoğlu *et al*., 2009). Despite their widespread use in flow control, Galerkin models suffer important disadvantages and difficulties. Although these models are much simpler that the Navier–Stokes equations, they are still non-linear systems. It is therefore very challenging to perform analysis and control design on Galerkin models. Consequently one must either resort to linearization as in Caraballo *et al*. (2008), or employ intricate non-linear control design tools as in Kasnakoğlu *et al*. (2008, 2009). An additional difficulty is associated with the derivation of Galerkin systems, where one has to substitute the POD expansion into the Navier–Stokes equations and then project onto the POD modes. These computations require complex calculations, manipulations and approximations on the Navier–Stokes PDEs. As a result of these actions, the Galerkin model behaviour may deviate significantly from that of the flow process described by the original Navier–Stokes PDEs, including the loss of stability properties (Noack *et al*., 2005; Rempfer, 2000).

In this paper, an alternative modelling procedure for flow control problems, which addresses the above-mentioned difficulties is outlined. The course of action differs from standard Galerkin modelling in that once the POD modes are obtained from flow snapshots, the IS and Galerkin projection (GP) steps are replaced by subspace system-identification methods (Ljun, 1999; Van Overschee and De Moor, 1996). The time coefficients obtained from the snapshots are used as the system output, and a linear-state space model that best represents the time coefficient data is obtained. No direct manipulations, substitutions or approximations are carried out on the Navier–Stokes PDEs and hence the errors from these operations are eliminated. In addition, subspace system identification is much easier to implement than IS and GP, owing to the readily available routines in commonly used scientific computing packages (eg, MATLAB). It is also possible to assure the stability of the system resulting from subspace identification (Lacy and Bernstein, 2003), which is not always the case for GP (Rempfer, 2000).

An additional issue regarding dynamical models for flow control is that the models are typically built at around a single operating point. The term operating point refers to any flow parameter of interest, eg, the density or viscosity of the fluid, the temperature, and the Reynolds numbers can all be regarded as flow parameters. In certain situations, a dynamical model describing operation for multiple values of the flow parameter is of interest. For instance, the fluid viscosity may vary during the operation, or the process may take place under different temperatures. Dynamical modelling for flow control under multiple-operating conditions is not a topic that has been researched extensively, although one finds studies utilizing non-equilibrium modes (Jørgensen *et al*., 2003; Noack *et al*., 2003, and snapshots sets encompassing data from multiple operating points or transients (Bergman *et al*., 2005; Ma and Karniadakis, 2002).

In this paper, modelling at multiple operating conditions is considered, which differs from pre-existing literature in that after the POD modes are obtained from snapshots comprising multiple operating conditions, a separate linear state-space model is obtained via subspace identification for each of the flow parameters, which are termed breakpoints for the model. These outputs of these models are then fused using output-blending (Hyde, 1995), which enables the calculation of the time coefficients continuously over the entire range of the flow parameters.

The ideas in the paper are developed in the following: Section 2 introduces the problem; Section 3 outlines the dynamical modelling approach; Section 4 presents a flow control case study and Section 5 ends the paper with conclusions and future work possibilities.

2. Problem description

In an inertial frame of reference, the general form of the Navier–Stokes PDEs describing the flow can be expressed as

$$\rho\left(\frac{\partial \mathbf{q}}{\partial t} + \mathbf{q} \cdot \nabla \mathbf{q}\right) = -\nabla p + \nabla \cdot \mathbb{T}(\mu) + \mathbf{f} \tag{1}$$

where $\mathbf{q}$ is the flow velocity, ρ is the fluid density, p is the pressure, $\mathbb{T}(\mu)$ is the (deviatoric) stress tensor, μ is the dynamic viscosity, $\mathbf{f}$ represents body forces per unit volume acting on the fluid and ∇ is the del operator (see Batchelor, 2000, for a complete discussion on the Navier–Stokes PDEs). The flow is subject to the initial condition

$$\mathbf{q}(x, 0) = \mathbf{q}_{\text{init}}(x) \tag{2}$$

where $\mathbf{q}_{\text{init}} \in \mathbb{H}$, and to the boundary conditions

$$\big(B_i(\mathbf{q}, \gamma)\big)(x, t) = b_i(x, t), \quad i = 1, \ldots, N_b \tag{3}$$

where $x \in \partial\Omega$, $t \in \mathbb{R}_+$, $B_i : \mathbb{H} \times \mathcal{C}^k \to \mathbb{H}$, $b_i \in \mathbb{H}$, $N_b \in \mathbb{N}$, $\Omega \subset \mathbb{R}^2$ is the flow domain, and $\mathbb{H}$ is the real Hilbert space $\mathbb{H} = \mathcal{L}_2(\Omega, \mathbb{R}^2)$. A control input $\gamma \in \mathbb{R}$ acts through the boundary conditions. The operators B_i may include spatial derivatives. The problem studied in this work is to develop a dynamical model for the fluid flow process in a form that can be used in standard control system design. In particular, we would like to obtain a discrete-time linear state-space system of the form

$$\xi(t + T_s) = A\xi(t) + B\gamma(t) \tag{4}$$

$$y(t) = C\xi(t) + D\gamma(t) \tag{5}$$

where $T_s \in \mathbb{R}_+$ is the sampling period, $\xi \in \mathbb{R}^n$ is the state vector, $n \in \mathbb{N}$ is the degree of the system and $y \in \mathbb{R}^N$ is the output signal, in such a way that (4)–(5) represents the fluid flow process in the sense that the time variation of the flow is captured by these dynamics. In addition, it is also desirable to study the operation of the flow process under different values of one or more flow parameters of interest (eg, density, viscosity, temperature, Reynolds number etc.). This implies obtaining multiple systems of the form (4)–(5), and implementing a means to fuse these systems together.

3. Modelling approach

The first step of the modelling procedure is to record two-dimensional instantaneous images, ie, snapshots, of the flow. The snapshots can be obtained either from actual physical experiments using techniques such as particle image velocimetry (Raffel *et al.*, 1998), or from computer data recorded from CFD simulations of the Navier–Stokes equations. Then a proper orthogonal decomposition (POD) of the flow is obtained using procedures outlined in Holmes *et al*. (1996) and Sirovich (1987), which results in the flow variable $\mathbf{q}$ being expressed as

$$\mathbf{q}(x,t) \approx \sum_{i=1}^{N} a_i(t)\phi_i(x) \tag{6}$$

The vectors ϕ_i, $i = 1, \ldots, N$, are called the POD modes and $a_i \in \mathbb{R}$ are termed the time coefficients. The time coefficients are obtained by projecting the snapshots onto the POD modes

$$a_i(t) = \left\langle \mathbf{q}(x,t), \phi_i(x) \right\rangle \tag{7}$$

Each POD mode ϕ_i has an associated eigenvalue λ_i that measures the amount of kinetic energy captured by that mode. Thus, the total percentage energy of the snapshots captured by the POD expansion (6) can be computed as

$$\%\mathrm{E} = 100 \times \frac{\sum_{i=1}^{N} \lambda_i}{\sum_{i=1}^{M} \lambda_i} \tag{8}$$

Having obtained a POD expansion of the flow as in (6), 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

$$\xi(t_{k+1}) = A\xi(t_k) + B\gamma(t_k) \tag{9}$$

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

which is a discrete-time model, since the flow snapshots are available at discrete time values $t_k \in \mathbb{R}_+$ separated by a sampling period of $T_s \in \mathbb{R}_+$ s. 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 the control input and $y \in \mathbb{R}^N$ is the output signal, which is the vector of time coefficients

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

The matrices A, B, C and D defining the dynamical system (9)–(10) are obtained by using the subspace system-identification techniques (Ljung, 1999). The input data for identification is $\{\gamma_k\}_{k=1}^{M}$ and the output data is $\{y_k\}_{k=1}^{M}$, where $y_k = y(t_k)$.

The model (9)–(10) produced by the steps above is a linear state-space model. Linear models can represent the flow behaviour satisfactorily around a given operating condition, but may fail to characterize the flow adequately if some flow parameters, eg, density or viscosity, differ considerably from the design conditions. To cope with this situation, numerous models of the form (9)–(10) are derived from snapshot sets obtained at different values of the flow parameters across the expected flow envelope. For simplicity, assume that a single flow parameter will be varying during the operation, although the results can be generalized straightforwardly to multiple varying parameters. Let θ denote this parameter, and let $\Theta \subset \mathbb{R}$ denote the flow envelope in which the parameter varies, ie, $\theta \in \Theta$. Let $\theta_i \in \Theta$ for $i = 1, \ldots, p$, where $p \in \mathbb{N}$, be the values of the parameters at which the modelling process outlined above will be carried out. The parameter values θ_i are termed breakpoints, and repeating the modelling steps above for each breakpoint results in p linear state-space models

$$\xi(t_{k+1}) = A(\theta)\xi(t_k) + B(\theta)\gamma(t_k) \tag{12}$$

$$y(t_k) = C(\theta)\xi(t_k) + D(\theta)\gamma(t_k) \tag{13}$$

for $\theta = \theta_1, \ldots, \theta_p$. To obtain the system behaviour for the values of θ that lie in between the breakpoints, we utilize output blending (Hyde, 1995). In this method the linear state-space models (12)–(13) for the breakpoints $\theta_i = \theta_1, \ldots, \theta_p$ are run in parallel and their outputs are interpolated according to the current value of the flow parameter θ. More specifically, if $\theta_i \le \theta < \theta_{i+1}$ then the blended system output is

$$y(t) = (1 - \lambda)\, y_i(t) + \lambda y_{i+1}(t) \tag{14}$$

where

$$0 \le \lambda = \frac{\theta - \theta_i}{\theta_{i+1} - \theta_i} < 1 \tag{15}$$

and y_i, y_{i+1} denote the output of the state-space model corresponding to breakpoint θ_i and θ_{i+1} respectively. A schematic illustration of the process is given in Figure 1. The process illustrated in the figure can implemented in MATLAB by augmenting the vector fields of all p state-space models and calling the appropriate ODE solver (eg, ode23, ode45, ode15s, etc.) to obtain the state trajectories and outputs $y_i(t)$ for all the models at once. The current value of the flow parameter θ can then be used to compute the blended output $y(t)$ as given in (14).

4. Case study: dynamical modelling of a boundary controlled flow governed by the two-dimensional Navier–Stokes Equations

In this example, 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 and the control input affects the system through the boundary conditions.

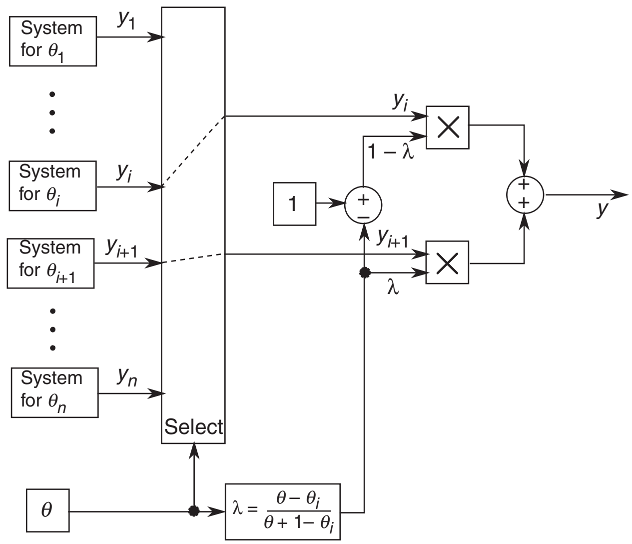

Figure 1 Blending the outputs of multiple systems based on a parameter θ

The flow parameter θ of interest is the kinematic viscosity (v), which is ratio of 1
dynamic viscosity (μ) to fluid density (ρ). The goal of this case study is to build a dynamical model describing the flow, where the range of the flow parameter is $\Theta = [0.000001, 1]\, m^2/s$; in other words, $0.000001\, m^2/s \leq v \leq 1\, m^2/s$, or equivalently $1\, cSt \leq v \leq 1000000\, cSt$. The selected kinematic viscosity range covers numerous common fluids under various temperature conditions. The lower end of the range corresponds to fluids where inertial forces are dominant and consequently the flow characteristics are turbulent. The higher end of the range corresponds to fluids where the viscous forces are dominant and hence the flow characteristics are laminar. To construct the desired dynamical model, we carry out the steps outlined in Section 3 by employing the scientific computing software MATLAB. In particular, for solving the Navier–Stokes PDEs, the MATLAB CFD solver Navier2D was used (Engwirda, 2005). Navier2D is a two-dimensional flow solver, which is adequate for the purposes of the example considered here; however, one should keep in mind that this is only an approximation, as the flows in real-life are three-dimensional. Navier2D is based on direct numerical simulation (DNS) and uses a finite volume method (FVM) for discretization. These choices are sufficient for the sample problem considered here; however, for different and more complex flow configurations one might need to employ turbulence models (eg, Reynolds-averaged Navier–Stokes (RANS) equations, Large eddy simulation (LES) or vortex method) or an alternate model for discretization (eg, finite element method (FEM) or finite difference method (FDM)).

To implement the subspace system identification methods, we make use of the System Identification Toolbox built into MATLAB. After the dynamical model is at hand, we also demonstrate how this model can be used to realize a control task related to the flow problem of interest, for which we employ functions from the Control Systems Toolbox.

Let $\mathbf{q}(x,y,t) = [u(x,y,t)\ v(x,y,t)] \in \mathbb{R}^2$ denote the flow velocity, where u and v are components in the longitudinal and lateral directions. The flow is governed by the Navier–Stokes PDEs where the initial conditions for the flow are

$$u(x,y,0) = v(x,y,0) = 0 \tag{16}$$

The boundary conditions are

$$u(x,0,t) = u(x,1,t) = 1 \tag{17}$$

$$v(x,0,t) = v(x,1,t) = 0 \tag{18}$$

$$u(0,y,t) = 0, \quad \frac{\partial v}{\partial x}(0,y,t) = 0 \tag{19}$$

$$u(1,y,t) = \begin{cases} 0, & y \in [0,0.42) \\ \gamma(t), & y \in [0.42,0.58] \\ 0, & y \in (0.58,1] \end{cases} \tag{20}$$

$$v(1,y,t) = 0 \tag{21}$$

and $\gamma \in \mathbb{R}$ is the control input. This flow problem was proposed in our earlier works (see for instance Kasnakoğlu *et al.*, 2008) as a relatively simple yet challenging benchmark problem to test modelling and control approaches for flow control problems. The problem contains both Dirichlet- and Neumann-type boundary conditions, which correspond to constant flowing, no-slip and stress-free walls around the flow domain Ω. The control input γ can induce changes to the system through only a limited segment of the right boundary. The first step of the procedure described in Section 3 is to simulate the flow configuration described above using Navier2D. 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 $T_s = 0.0069$ s for 1000 time steps on a 50×50 uniform grid of the spatial domain. These simulations were repeated for 10 different values of the operating point, ie, for 10 different values of the kinematic viscosity v. The selected breakpoints are $v_1 = 00000100$, $v_2 = 0.00001931$, $v_3 = 0.00037276$, $v_4 = 0.00719686$, $v_5 = 0.07142950$, $v_6 = 0.21428650$, $v_7 = 0.37275937$, $v_8 = 0.57142900$,

$v_9 = 0.78571450$ and $v_{10} = 1.00000000$. There are no hard and fast rules about the selection of these breakpoints in general, but for the problem at hand the breakpoints are concentrated near the lower limit of Θ, given that the sensitivity of the flow characteristics to viscosity are more pronounced at lower v values. For illustration, a few snapshots obtained from the CFD simulations under no input and under chirp signal input are shown in Figure 2, for selected values of the viscosity at the breakpoints $v_1 - v_{10}$ [only the longitudinal (u) components are shown as the sole purpose of these figures is to give a pictorial idea about the flow behaviour]. In all four figures, it can be seen that the flow satisfies the boundary conditions given in (16)–(21), including a constant velocity flow on the top and bottom walls, stress free-behaviour on the left wall, and flow induced by the actuation on the right wall. While these same boundary conditions are satisfied for all cases, the behaviour inside the flow domain Ω is significantly different for each viscosity value. Specifically, the flow behaviour is significantly oscillatory and exhibits turbulent tendencies for low values of v, whereas for high values of v, the flow characteristics are steadier and laminar. The numerical instabilities within the CFD solver may also be a factor contributing to the oscillatory and random-like behaviour at lower viscosity values.

The second step is obtaining the POD modes ϕ_i in terms of which the flow variable $\mathbf{q}$ will be expanded, by applying the POD procedure to the flow snapshots. Examining the eigenvalues corresponding to the POD modes it can be computed using (8) that these POD modes capture about 96% of the energy contained in the snapshots and hence it is acceptable to truncate the POD expansion (6) at $N = 4$. Once the POD modes are obtained, the snapshots recorded for each breakpoint value v_i are projected onto the POD modes as in (7) to obtain the time coefficients a_i. The vector (11) built from these time coefficients constitutes the output data for system identification. The output data is augmented with the input values to form the input–output data, and this data is then randomly partitioned into two parts, where the first part constitutes the data for estimation and the second part the data for validation. Applying subspace system identification to the input–output estimation data yields a linear state-space model for each viscosity value v_i, resulting in a total of 10 linear dynamical models. The order of each system was selected as $n = 4$, which was determined after experimenting with different orders and evaluating the relative importance of the dimension of the state vector by inspecting the singular values of the Hankel matrices. Figure 3 shows the time coefficients a_i obtained from the snapshots versus the outputs y_i of the dynamical models. Two forcing conditions are shown, namely, the unforced case ($\gamma = 0$), and the case when the input is a chirp signal. It can be observed that the models produce responses very close to the actual time coefficients. In fact, the responses are almost identical to the time coefficients for the higher end of the viscosity spectrum. For lower viscosity values, the model responses do not match the time coefficients perfectly; however, they still capture the general trend satisfactorily. A difference of this sort between low and high viscosity flow is to be expected since the flow characteristics tend to be more turbulent for lower v values. An additional problem regarding lower

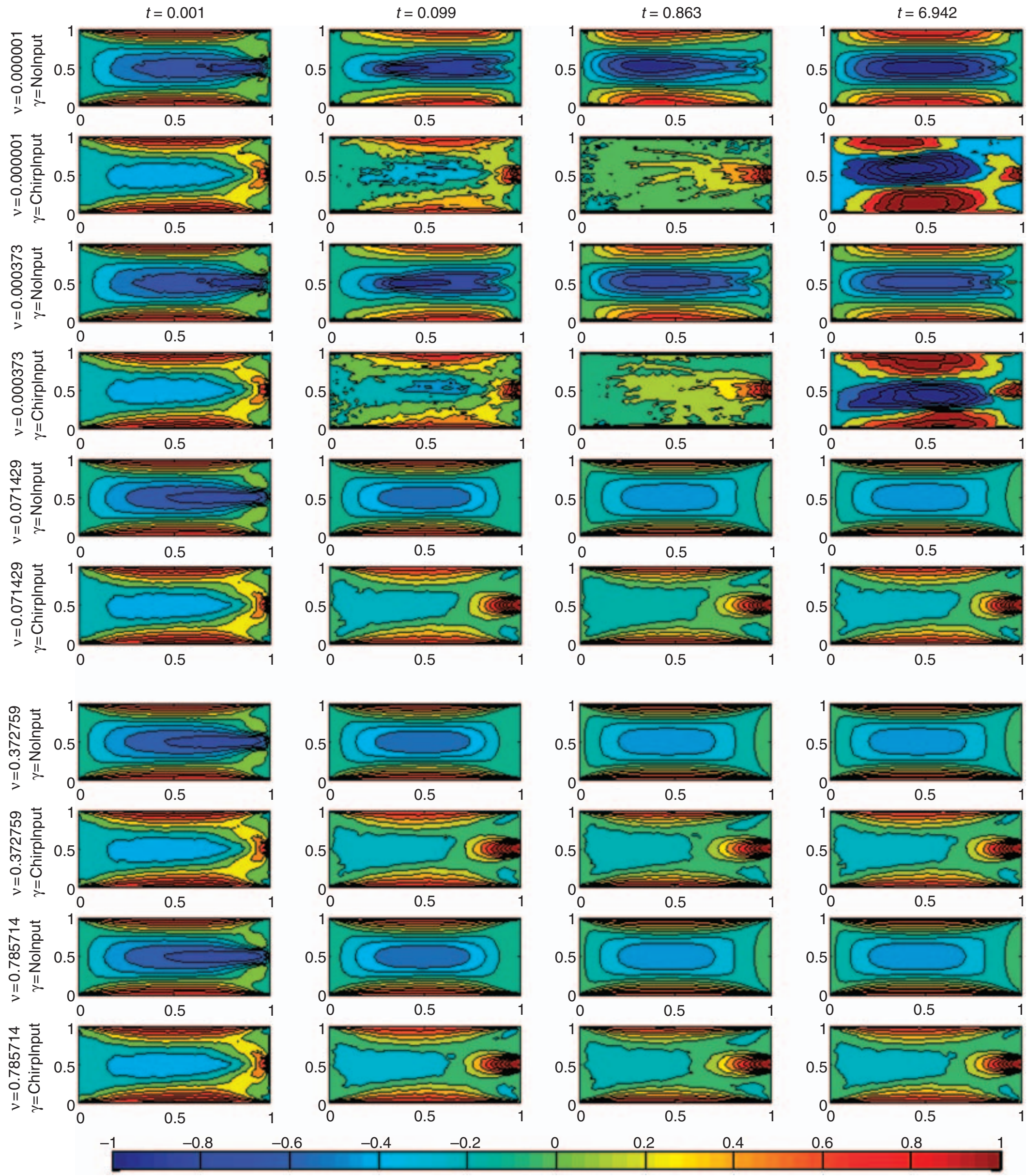

Figure 2 Snapshots of the longitudinal component of the flow velocity, at selected breakpoint values of the flow parameter v, under no input and chirp input forcing

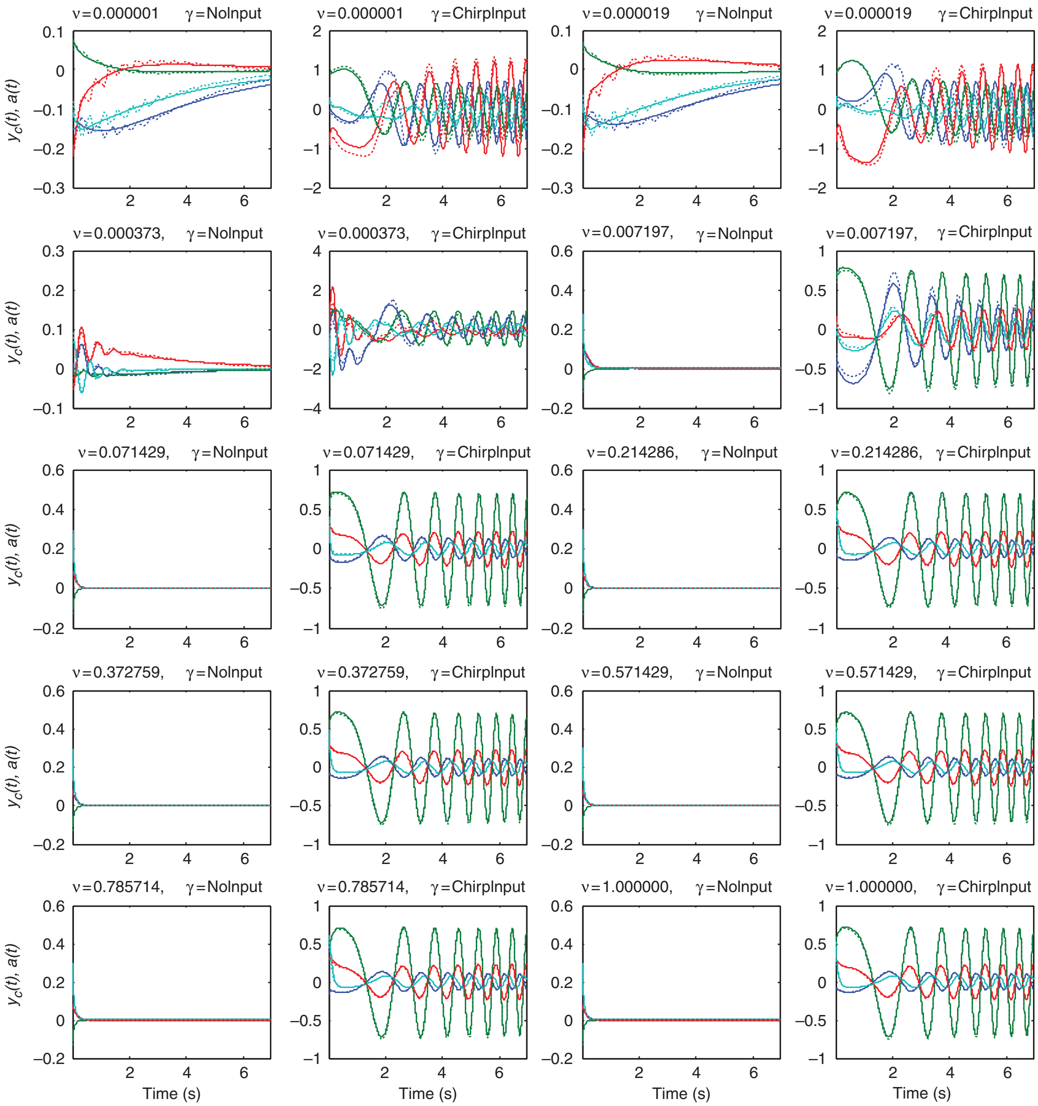

Figure 3 Time coefficients a obtained from snapshots (dashed) versus the outputs y_c of the dynamical models (solid), for breakpoint values of the flow parameter v. The colours blue, green, red and cyan correspond to the first, second, third and fourth elements of the vectors respectively

v values is that the mesh used for CFD simulations may be not be dense enough, which in turn might cause numerical instabilities. As a result of these phenomena, the time coefficient trajectories tend to exhibit oscillatory and random behaviour, which are difficult to capture in the modelling process. To quantify the performances of the dynamical models one can make use of the average percentage root mean squared error: For a given breakpoint v_i, the average percentage root mean square (%RMSE) error is computed as

$$\%\mathrm{RMSE} = 100\sqrt{\frac{1}{MN}\sum_{k=1}^{M}\sum_{i=1}^{N}\left(\frac{a_i(t_k) - y_i(t_k)}{a_i(t_k)}\right)^2} \tag{22}$$

Though a single %RMSE value can be computed for the entire set of snapshots, for comparison purposes it is helpful to compute separate %RMSE values for unforced and forced operation. For these cases the values a_i and y_i in (22) are the time coefficients and outputs for the particular case of interest, and M is the number of data for that case. Table 1 shows the %RMSE values computed at the breakpoints for unforced operation and also when the system is forced with a chirp signal input. Examining the table one observes that for the unforced case there is good agreement between the time coefficients and the outputs of the models for all 10 breakpoints, and the error is less than 1% for all cases. For the forced case, the error is also acceptable and is around 1–2% for the most part, except for the lower end of the viscosity range, where it rises to about 16%. As explained earlier, such a rise in error value towards the lower end of Θ is to be expected; for low v values the flow exhibits oscillatory behaviour caused by turbulent flow characteristics, and the effect of numerical instabilities as well as unmodelled/unsimulated dynamics (eg, because of a coarse grid) become more apparent.

The subsequent step is to combine the 10 dynamical models so that the predictions of the flow characteristics are obtained continuously over the flow envelope Θ. By continuously, we mean the ability to model the flow for any $v \in \Theta$, where v is not

Table 1 Percentage root mean squared error between actual time coefficients and the outputs of the dynamical models, for breakpoint values of v

v (m^2/s)	%RMSE (no input)	%RMSE (chirp input)
0.00000100	0.631659247	16.06740648
0.00001931	0.554085603	16.18678612
0.00037276	0.202137261	16.1782188
0.00719686	0.002742377	2.789811998
0.07142950	0.003737121	1.061357123
0.21428650	0.003268267	0.941987259
0.37275937	0.003302684	0.890139528
0.57142900	0.003097139	0.895443896
0.78571450	0.003098128	0.95728875
1.00000000	0.003162296	0.87916504

necessarily equal to a breakpoint value v_i. For this purpose, we make use of the output blending technique mentioned in Section 3 (Figure 1). This step results in a dynamical model describing the flow process for all values of the flow parameter of interest within the desired range Θ. To evaluate its performance, additional CFD simulations are carried out for several values of v between the breakpoints, and time coefficients are obtained from the resulting snapshots. The non-breakpoint values of v used for evaluation are $v'_1 = 0.00000268$, $v'_2 = 0.00000720$, $v'_3 = 0.00005179$, $v'_4 = 0.00013895$, $v'_5 = 0.00100000$, $v'_6 = 0.00268270$, $v'_7 = 0.01930698$, $v'_8 = 0.05179475$, $v'_9 = 0.13894955$, $v'_{10} = 0.14285800$, $v'_{11} = 0.28571500$, $v'_{12} = 0.35714350$, $v'_{13} = 0.42857200$, $v'_{14} = 0.50000050$, $v'_{15} = 0.64285750$, $v'_{16} = 0.71428600$, $v'_{17} = 0.85714300$ and $v'_{18} = 0.92857150$. Figure 4 shows a comparison of the time coefficients for these v values with the results obtained from output-blending, where it is seen that the trajectories are reasonably close to one another. The %RMSE for each case is also tabulated in Table 2, which are also acceptable. (Again, for low v values the %RMSE is higher because of the oscillatory behaviour caused by turbulent flow characteristics, as well as numerical instabilities and unmodelled/unsimulated dynamics.) In summary, it can be stated that the proposed modelling procedure produces satisfactory results in representing the flow process over the entire range of the flow parameter v.

As stated in the beginning, the main motivation for constructing a dynamical model is to carry out a control design task for the flow process under consideration. For the sake of example, let us presume that the task is to control the longitudinal velocity of the centre point of the domain $(x_o, y_o) := (0.5, 0.5)$. For controller design, we will follow a similar approach to modelling, in the sense that we build individual controllers for the linear state-space models obtained at the breakpoints, and then blend the controller outputs based on the current value of the flow parameter. When applied to the controller outputs, the blending procedure is usually termed input blending rather than output blending, since the blended signal is the input to the flow process. Let us denote the quantity to be controlled as y_c and express it in terms of the POD expansion in (6), which yields

$$y_c(t) = u(x_o, y_o, t) = \sum_{i=1}^{4} a_i(t)\phi_{i,u}(x_o, y_o) =: C'a(t) \tag{23}$$

where C' is the 1×4 matrix

$$C' := [\,\phi_{1,u}(x_o, y_o) \quad \phi_{2,u}(x_o, y_o) \quad \phi_{3,u}(x_o, y_o) \quad \phi_{4,u}(x_o, y_o)\,] \tag{24}$$

Recall that for a given breakpoint v_i, a dynamical model of the form (9)–(10) has already been obtained following the steps described earlier. Then from (10) and (23) one obtains

$$\begin{aligned} y_c &= C'a = C'(C\xi + D\gamma) \\ &= C_c\xi + D_c\gamma \end{aligned} \tag{25}$$

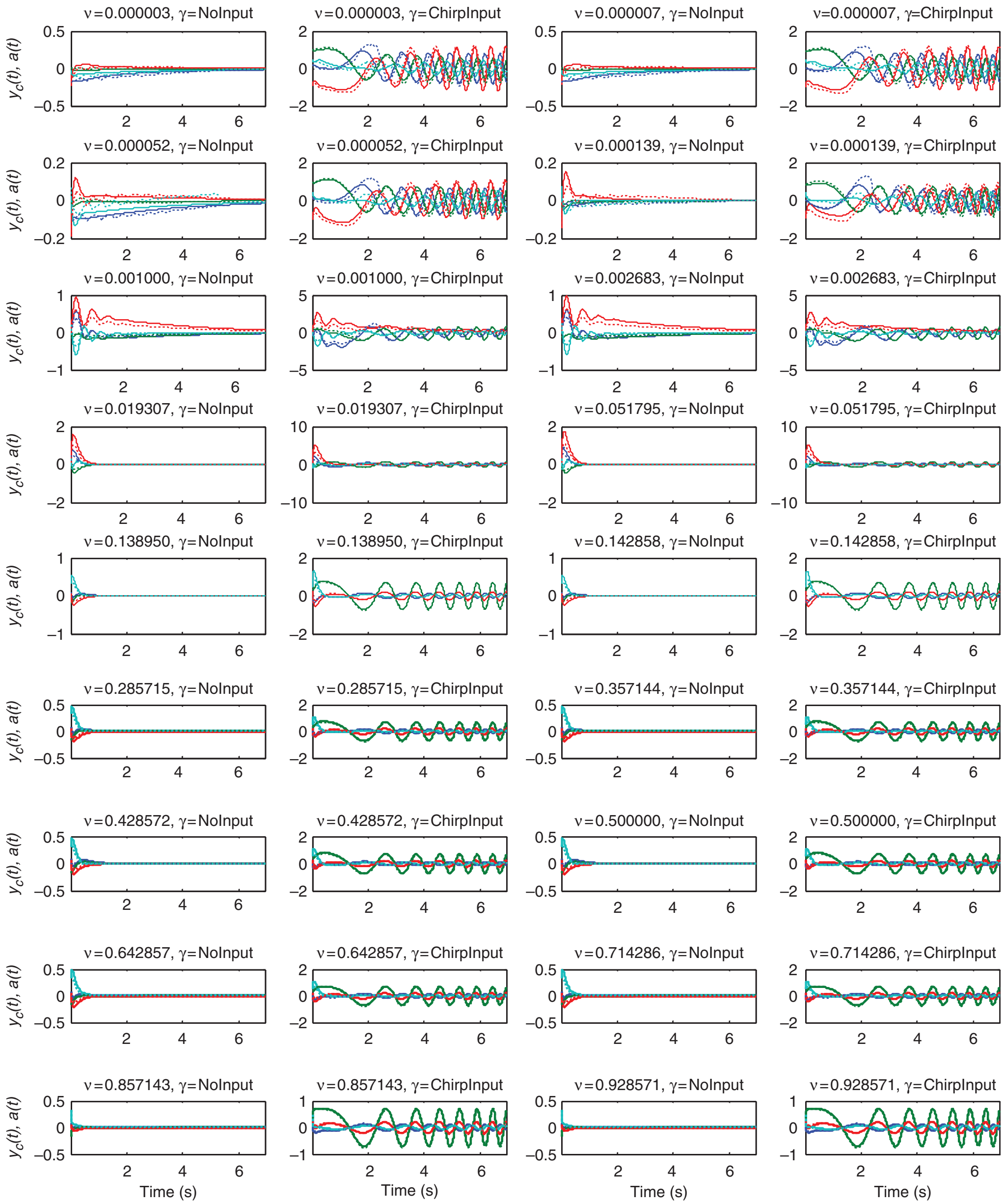

Figure 4 Time coefficients a obtained from snapshots (dashed) versus the outputs y_c of the dynamical models (solid), for non-breakpoint values of the flow parameter v. The colours blue, green, red and cyan correspond to the first, second, third and fourth elements of the vectors respectively

Table 2 Percentage root mean squared error between actual time coefficients and the outputs of the dynamical models, for non-breakpoint values of v

v (m^2/s)	%RMSE (no input)	%RMSE (chirp input)
0.00000268	1.154424943	17.79804256
0.00000720	1.183815725	17.03198281
0.00005179	0.852570529	16.28663879
0.00013895	0.970341921	15.7536293
0.00100000	4.327768622	17.98535088
0.00268270	4.454212301	15.74610951
0.01930698	1.228373634	5.685199342
0.05179475	1.372493217	5.222336112
0.13894955	0.36887619	1.836132481
0.14285800	0.369076022	1.842522767
0.28571500	0.307725256	1.646235896
0.35714350	0.308298056	1.636625119
0.42857200	0.349282282	1.742876644
0.50000050	0.349598762	1.740751512
0.64285750	0.318717516	1.656126996
0.71428600	0.318849569	1.65020463
0.85714300	0.078611544	1.055233326
0.92857150	0.078628958	1.05628866

where $C_c := C'C$ and $D_c := C'D$. Stacking the state dynamics (9) with the output y_c to be regulated yields

$$\xi(t_{k+1}) = A\xi(t_k) + B\gamma(t_k) \tag{26}$$

$$y_c(t_k) = C_c\xi(t_k) + D_c\gamma(t_k) \tag{27}$$

which is a single-input single-output (SISO) system from γ to y_c. Denote by y_{ref} the reference signal to be tracked by y_c. To accomplish the desired tracking, design a compensator K with state dynamics

$$\chi(t_{k+1}) = A_k\chi(t_k) + B_k e(t_k) \tag{28}$$

$$\gamma(t_k) = C_k\chi(t_k) + D_k e(t_k) \tag{29}$$

where A_k, B_k, C_k and D_k are the state matrices of the controller, χ is the controller state, γ is the input to the system and $e = y_{ref} - y_c$ is the tracking error. A selection of standard and automatic design methods exist for obtaining K, including PID tuning methods, internal model control (IMC) design techniques, LQG synthesis and optimization-based design. Experimenting with numerous compensators using various methods, it was seen that the most satisfactory results are obtained for compensators built using IMC design methods (Chien and Fruehauf, 1990). Ten compensators were designed for the 10 linear state-space models using IMC design techniques; we shall denote by K_i the compensator designed for the linear state-space model corresponding to breakpoint v_i. A key design parameter for IMC is the desired dominant closed-loop time constant (τ), which has been selected as $\tau = 0.1$ s, to achieve a settling time less than 0.5 s. To evaluate the performance of the designed controllers, closed-loop CFD simulations of the flow process were carried out at the breakpoints. Figure 5 shows the trajectory of the point $(x_o, y_o) = (0.5, 0.5)$ of interest when the reference signal y_{ref} is the unit step signal. To make the situation more realistic, disturbances were also added to the inputs and outputs of the system. The disturbances applied are in the form of white noise signals with magnitude 0.1, which is 10% of the reference signal. It is seen from Figure 5 that for all breakpoints v_i, the closed-loop system successfully tracks the unit step, and the settling time is less than 0.5 s as desired. That is, despite the very different open-loop behaviours of the flows at different viscosity values (Figure 2), the closed-loop responses can be rendered to be quite similar to each other (Figure 5) thanks to a separate controller design at each breakpoint.

The next step is to fuse the controllers K_i using the blending procedure, so that the desired tracking can be achieved over the entire flow envelope Θ. If the current value
1 for the flow parameter is v, where $v_i \leq v < v_{i+1}$, then input signal γ to be applied to the flow process is constructed as

$$\gamma(t) = (1 - \lambda)\gamma_i(t) + \lambda\gamma_{i+1}(t) \tag{30}$$

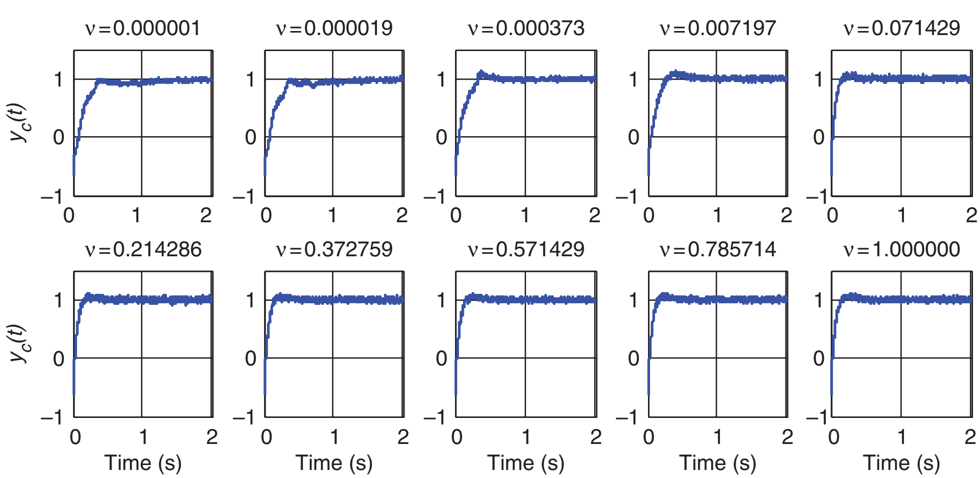

Figure 5 Closed-loop step response of the point $(x_o, y_o) := (0.5, 0.5)$ obtained from CFD simulations, for breakpoint values of the flow parameter

where

$$0 \leq \lambda = \frac{v - v_i}{v_{i+1} - v_i} < 1 \tag{31}$$

and γ_i denotes to output of the ith IMC controller K_i. The blended controller was applied to the flow process and CFD simulations were carried out at the non-breakpoint values $v'_1 - v'_{18}$. Figure 6 shows the trajectory of the point $(x_o, y_o) = (0.5, 0.5)$ of interest. It is seen that closed-loop system settles to the reference value in less than 0.5 s for all cases, and successfully tracks the reference from thereafter.

As a more challenging test for the blended controller design, additional experiments were performed for the case when the flow parameter v is time-varying. For these simulations, the reference signal y_{ref} to be tracked is kept constant at 1 for until about $t = 1.0$ s, after which it is switched to -1. As before, white-noise disturbances of magnitude 0.1 were applied to the input and output. The snapshots resulting from this closed-loop operation are shown in Figure 7. In addition, the time trajectory of the point $(x_o, y_o) = (0.5, 0.5)$ is 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

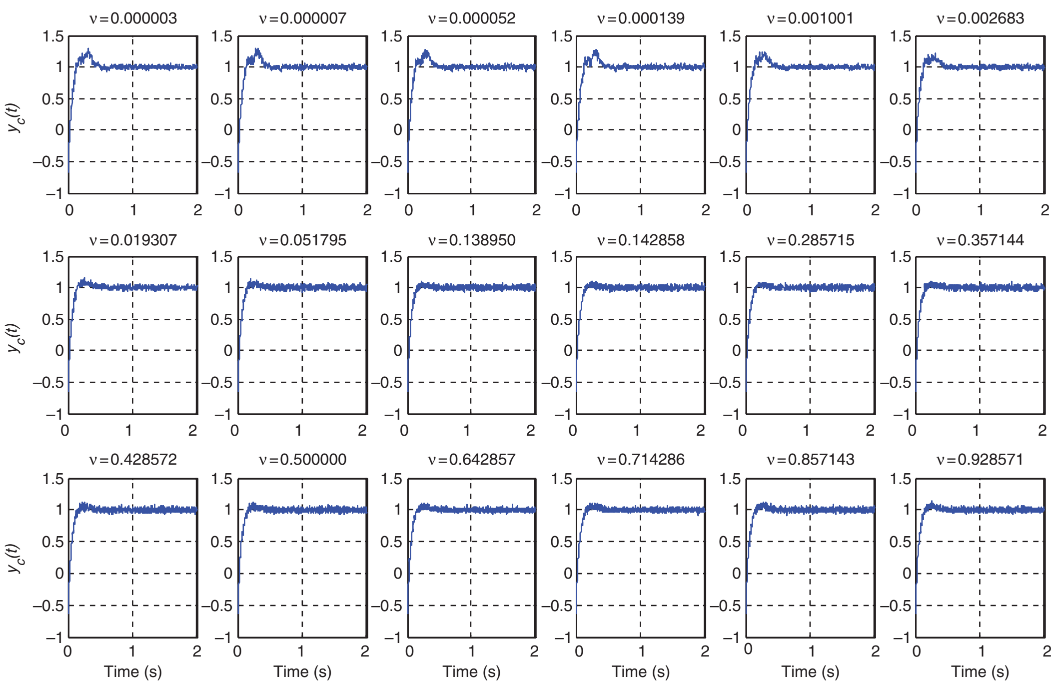

Figure 6 Closed-loop step response of the point $(x_o, y_o) := (0.5, 0.5)$ obtained from CFD simulations, for non-breakpoint values of the flow parameter

system accomplishes the desired tracking and keeps the velocity of the given point close to the reference signal. The small oscillation about the reference signal is tolerable and is related to input/output noises, in addition to the unmodelled dynamics resulting from operation away from the breakpoints and representing an infinite dimensional non-linear PDE system with a set of finite dimensional linear state-space models.

5. Conclusions and future works

In this paper, a systematic dynamical modelling procedure for fluid flow control problems is proposed. Snapshots of the flow process are obtained from experiments or CFD simulations, and a POD expansion of the flow is computed. The time coefficients of the expansion are merged with the input values to form the estimation data, and a linear state-space system representing the time coefficients is obtained using subspace system identification. The procedure is repeated for a number of operating points, called breakpoints, which are characterized by one or more flow parameters of interest. The dynamical models obtained at the breakpoints are then combined using the output-blending technique. The result of the procedure is a dynamical model, which can represent the flow process continuously within the flow envelope, ie, within a desired range of the flow parameters. The modelling approach developed

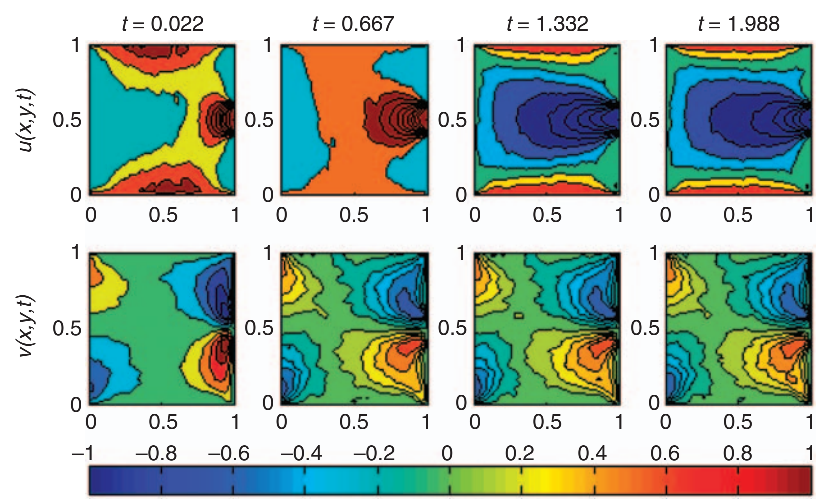

Figure 7 Flow snapshots obtained from CFD simulation under closed-loop operation, for the case when the flow parameter v is time-varying and the reference signal switches from 1 to -1 at $t = 1\,\mathrm{s}$

was seen to be successful on a flow control case study governed by the Navier–Stokes equations, where kinematic viscosity is regarded as the control parameter.

The main contributions and novelties put forward by this work can be summarized as follows: methodologically, the course of action differs from standard Galerkin modelling in that once the POD modes are obtained from flow snapshots, the IS and GP steps are replaced by subspace system-identification. Hence, no direct manipulations, substitutions or approximations are carried out on the complicated Navier–Stokes PDEs, which is a common source of numerical errors. An additional benefit is that subspace system identification is much easier to implement than IS and GP, because of readily available routines in commonly used scientific computing packages, eg, MATLAB. It is also possible to enforce the stability of the system resulting from subspace identification, which is not always the case for Galerkin models. Another important contribution of the proposed approach is the ability to model the flow process under different operating points, and in a continuous manner within a given flow envelope. Another advantage of the proposed method is that it is based on linear models, which is makes analysis and control design less complicated compared to non-linear Galerkin models. Such an approach was shown to be feasible and successful in the case study presented in Section 4.

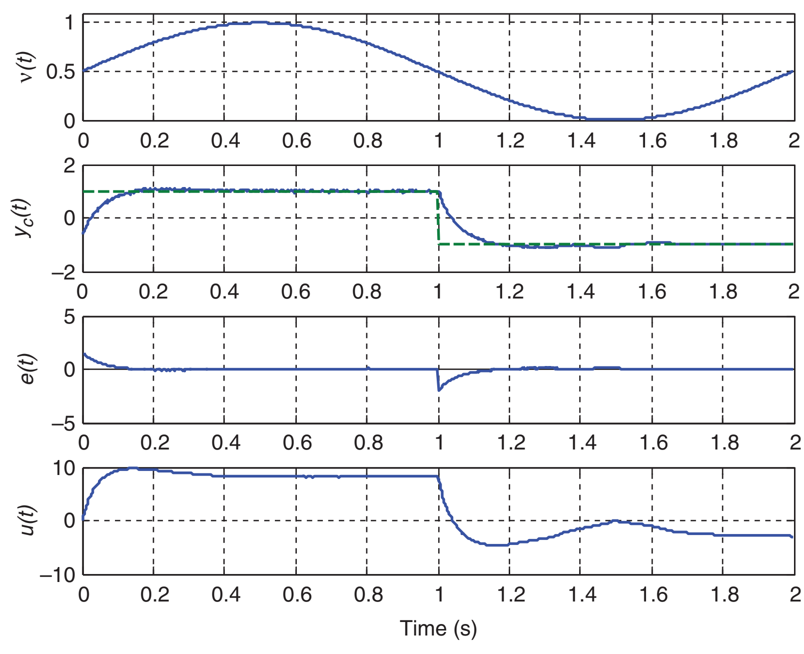

Figure 8 First (from top): Trajectory of the flow parameter v. Second: the trajectory of the point of interest (solid) and the reference signal (dashed). Third: Error signal $e=y_{ref}-y_c$. Fourth: Control signal

Future research directions include employing alternative schemes for identification, and applying the proposed technique to different flow control problems.

Acknowledgement

C. Kasnakoğlu would like to thank A. Serrani and M. Samimy for our collaborative works on flow control, and also the libraries of TOBB University of Economics and Technology for providing valuable resources used in this study.

References

Acheson, D.J. 1990: *Elementary fluid dynamics*. Oxford University Press.

Banaszuk, A., **Ariyur, K.B.**, **Krstic, M.** and **Jacobson, C.A.** 2004: An adaptive algorithm for control of combustion instability. *Automatica* 40, 1965–72.

Baramov, L., **Tutty, O.R.** and **Rogers, E.** 2004: H-infinity control of non-periodic two-dimensional channel flow. *IEEE Transactions on Control Systems Technology* 12, 111–22.

Batchelor, G.K. 2000: *An introduction to fluid dynamics*. Cambridge University Press.

Bergmann, M., **Cordier, L.** and **Brancher, J.P.** 2005: Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids* 17, 097101.

Caraballo, E., **Kasnakoğlu, C.**, **Serrani, A.** and **Samimy, M.** 2008: Control input separation methods for reduced-order model-based feedback flow control. *AIAA Journal* 46, 2306–22.

Chien, I.L. and **Fruehauf, P.S.** 1990: Consider IMC tuning to improve controller performance. *Chemical Engineering Progress* 86, 33–41.

Cohen, K., **Siegel, S.**, **McLaughlin, T.**, **Gillies, E.** and **Myatt, J.** 2005: Closed-loop approaches to control of a wake flow modeled by the Ginzburg–Landau equation. *Computers & Fluids* 34, 927–49.

Efe, M.O. and **Ozbay, H.** 2004: Low dimensional modelling and Dirichlet boundary controller design for Burgers equation. *International Journal of Control* 77, 895–906.

Engwirda, D. 2005: An unstructured mesh Navier–Stokes solver. Master's thesis, School of Engineering, University of Sydney. Also available at MATLAB Central.

Gad-el Hak, M. 2000: *Flow control - passive, active, and reactive flow management*. Cambridge University Press.

Hirsch, C. 2007: *Numerical computation of internal and external flows: fundamentals of computational fluid dynamics*. Butterworth-Heinemann.

Holmes, P., **Lumley, J.L.** and **Berkooz, G.** 1996: *Turbulence, coherent structures, dynamical system, and symmetry*. Cambridge University Press.

Hyde, R.A. 1995: *H-infinity aerospace control design – a VSTOL flight application*. Springer-Verlag.

Jørgensen, B.H., **Sørensen, J.N.** and **Brøns, M.** 2003: Low-dimensional modeling of a driven cavity flow with two free parameters. *Theoretical and Computational Fluid Dynamics* 16, 299–317.

Joslin, R.D. 1998: Aircraft laminar flow control. *Annual Review of Fluid Mechanics* 30, 1–29.

Kasnakoğlu, C. and **Serrani, A.** 2007: Attenuation of oscillations in Galerkin systems using center manifold techniques. *European Journal of Control* 13, 529–42.

Kasnakoğlu, C., **Camphouse, R.C.** and **Serrani, A.** 2009: Reduced-order model-based feedback control of flow over an obstacle using center manifold methods. *Journal of Dynamic Systems, Measurement, and Control* 131, 011011.

Kasnakoğlu, C., **Serrani, A.** and **Efe, M.O.** 2008: Control input separation by actuation mode expansion for flow control problems. *International Journal of Control* 81, 1475–92.

Kim, J. 2003: Control of turbulent boundary layers. *Physics of Fluids* 15, 1093.

Lacy, S.L. and **Bernstein, D.S.** 2003: Subspace identification with guaranteed stability using constrained optimization. *IEEE Transactions on Automatic Control* 48, 1259–63.

Ljung, L. 1999: *System identification: theory for the user*. PTR Prentice Hall.

Ma, X. and **Karniadakis, G.E.** 2002: A low-dimensional model for simulating three-dimensional cylinder flow. *Journal of Fluid Mechanics* 458, 181–90.

Noack, B.R., **Afanasiev, K.**, **Morzynski, M.**, **Tadmor, G.** and **Thiele, F.** 2003: A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics* 497, 335–63.

Noack, B.R., **Papas, P.** and **Monketwitz, P.A.** 2005: The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *Journal of Fluid Mechanics* 523, 339–65.

Raffel, M., **Willert, C.E.** and **Kompenhans, J.** 1998: *Particle image velocimetry: a practical guide*. Springer Verlag.

Rempfer, D. 2000: On low-dimensional Galerkin models for fluid flow. *Theoretical and Computational Fluid Dynamics* 14, 75–88.

Singh, S.N., **Myatt, J.H.**, **Addington, G.A.**, **Banda, S.** and **Hall, J.K.** 2001: Optimal feedback control of vortex shedding using proper orthogonal decomposition models. *Transactions of the ASME. Journal of Fluids Engineering* 123, 612–18.

Sirovich, L. 1987: Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics* XLV, 561–90.

Van Overschee, P. and **De Moor, B.** 1996: *Subspace identification for linear systems: theory, implementation, applications*. Kluwer Academic Publishers.

Versteeg, H.K. and **Malalasekera, W.** 2007: *An introduction to computational fluid dynamics: the finite volume method*. Prentice Hall.