Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

CFD-Based Adaptive Flow Control for Steady Flow Field Modification

Mario A. Santillo, Jesse B. Hoagg, Dennis S. Bernstein, Kenneth Powell

Department of Aerospace Engineering,
The University of Michigan,
Ann Arbor, MI 48109-2140
{santillo, jhoagg, dsbaero, powell}@umich.edu

I. Introduction

In this paper we demonstrate adaptive flow control for an incompressible viscous fluid through a two-dimensional channel without the use of an analytical model. An adaptive algorithm is implemented within a CFD simulation to control the performance variables, which are velocity components of the flow at downstream locations, to desired steady-state values. The algorithm requires minimal knowledge of the system, specifically, the numerator coefficients of the transfer functions from the control inputs to the performance variables. System identification, based on CFD simulations prior to the closed-loop simulations, is used to identify the required parameters.

Steady flow field modification is achieved using the ARMARKOV disturbance rejection algorithm [1], [2] implemented within Fluent, a commercial CFD software package available from Fluent, Inc [3]. The ARMARKOV algorithm has been applied to various testbeds in [4]–[12], and has been demonstrated to work for CFD-based adaptive disturbance rejection in [13].

An alternative objective in active flow control is to modify the steady-state flow field. Applications include modifying the flow field around obstacles to reduce drag and delay stall. To demonstrate the concept, we consider 2D channel flow without a disturbance actuator. The control objective is to command the flow field at the location of the performance sensors to independently achieve desired transverse and longitudinal flow velocities.

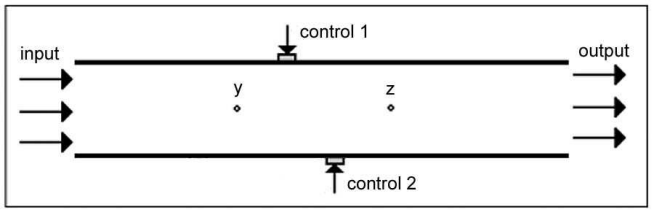

Fig. 1. 2D channel geometry for adaptive flow control.

We consider a two-dimensional channel geometry with velocity sensors y and z and two control jets u_1 and u_2 as in Figure 1. The goal is to control the performance variables to desired steady-state values using the control jets positioned between the measurement and performance sensors. Traditionally, the goal of the ARMARKOV algorithm has been to reduce the effects of an unknown disturbance on the performance sensor. Here, we locate two performance sensors downstream such that one is aligned with the longitudinal (downstream) flow direction and the other is aligned with the transverse flow direction. Independently controlling the desired longitudinal and transverse flow velocities at the performance sensor generates steady flow field modification.

An overview of flow control is given by [14]. Many standard control techniques have been applied to active flow control. For example, PI and LQG controllers have been considered in [15]–[17] based on a reduced-order model of the linearized Navier-Stokes equation, obtained by a Galerkin procedure. LQG/LTR control of the streamfunction formulation of the Navier-Stokes equation is used in [18], [19] to achieve drag reduction below the laminar level. An LQG controller based on a reduced-order model obtained from a finite element code is used in [20]. In addition, a nonlinear control law is developed in [21], where a Galerkin method is used to derive a reduced-order model of the two-dimensional Navier-Stokes equation. Alternative methods presented in [22]–[24] include PDE-based control, robust control, and predictive control with direct numerical simulation.

Flow control using the ARMARKOV adaptive algorithm differs from model-based control methods since a detailed model of the flow is not required. Instead, a CFD simulation is used to generate input/output data of the flow while band-limited white noise is applied at the control jets. The numerator coefficients of an ARMARKOV model are then identified off-line using a system identification algorithm. The combination of CFD simulation and the ARMARKOV adaptive algorithm thus constitutes a technique for controlling a fluid without analytical modeling. An alternative method to identify fluid-dynamic systems is given by [25].

To apply this technique to a physical system one would first use a CFD simulation coupled with system identification to obtain the numerator coefficients of an ARMARKOV model of the system. Alternatively, measurements obtained from the physical system can be used for identification assuming that the disturbance levels are manageable compared to the control input. Adaptive steady flow control using the ARMARKOV algorithm could then be implemented in real time on the physical system.

Supported by the National Aeronautics and Space Administration under a Graduate Researchers Program Fellowship

II. Problem Statement and Description

The goal of this paper is to demonstrate the use of the ARMARKOV adaptive algorithm for steady flow field modification directly on a CFD simulation of 2D channel Poiseuille (viscous) flow. Consider the 2D channel geometry shown in Figure 1. Let $u_1(t) \in \mathbb{R}$ be the control signal applied to the flow on the upper wall, $u_2(t) \in \mathbb{R}$ be the control signal applied to the flow on the lower wall, $y(t) \in \mathbb{R}^2$ the measurement variables, and $z(t) \in \mathbb{R}^2$ the performance variables. Here $y(t)$ and $z(t)$ are velocity components of the flow at the sensor locations. Both $u_1(t)$ and $u_2(t)$ represent mass flow into (or out of) the system.

Two suction/blowing jets are placed in the walls of the channel, one to apply control along the upper wall (located 1.25 m from the inlet on the lower wall) and the second to apply control along the lower wall (located 1.5 m from the inlet on the upper wall). Measurement and performance velocity sensors (y and z) are located in the center of the channel at 1.0 m and 1.5 m, respectively, from the inlet. The wall separation distance is 0.25 m. For the purpose of simulation, we take the channel centerline velocity to be 2 m/s and the fluid to be incompressible. Taking the fluid as air at standard temperature and pressure and wall separation as the characteristic length, the Reynolds number is 3.3×10^4.

A segregated, 2D, unsteady, implicit, second-order solver is used for the CFD simulation, along with User Defined Functions (UDFs) and function hooks to integrate the custom C-programming code into the simulation. ASCII text files are used to transfer data to and from the simulation environment.

III. Adaptive Disturbance Rejection Algorithm Description

We now provide a brief overview of the key components of the ARMARKOV adaptive disturbance rejection algorithm. Consider the n^{th}-order discrete-time finite-dimensional linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k) + D_1 w(k), \tag{1}$$
$$z(k) = E_1 x(k) + E_2 u(k) + E_0 w(k), \tag{2}$$
$$y(k) = Cx(k) + Du(k) + D_2 w(k), \tag{3}$$

with the transfer function representation

$$z = G_{zw} w + G_{zu} u, \tag{4}$$
$$y = G_{yw} w + G_{yu} u. \tag{5}$$

The controller G_c that produces the control signal $u(k)$ using feedback $y(k)$ is written as

$$u = G_c y. \tag{6}$$

The ARMARKOV model of (1) - (3) is the μ-step ahead model

$$\begin{aligned} z(k) = &\textstyle\sum_{j=1}^{n} -\alpha_j z(k-\mu-j+1) \\ &+ \textstyle\sum_{j=1}^{\mu} H_{zw,j-2} w(k-j+1) \\ &+ \textstyle\sum_{j=1}^{n} B_{zw,j} w(k-\mu-j+1) \\ &+ \textstyle\sum_{j=1}^{\mu} H_{zu,j-2} u(k-j+1) \\ &+ \textstyle\sum_{j=1}^{n} B_{zu,j} u(k-\mu-j+1), \end{aligned} \tag{7}$$

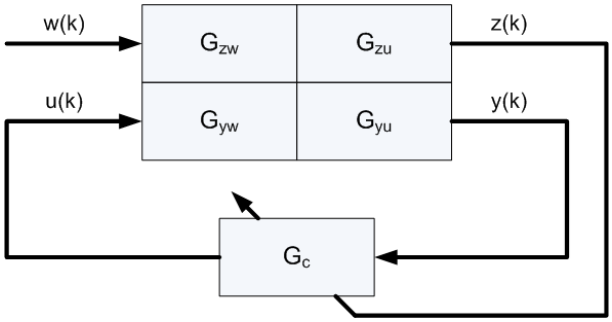

Fig. 2. Standard problem with controller adaptation

$$\begin{aligned} y(k) = &\textstyle\sum_{j=1}^{n} -\alpha_j y(k-\mu-j+1) \\ &+ \textstyle\sum_{j=1}^{\mu} H_{yw,j-2} w(k-j+1) \\ &+ \textstyle\sum_{j=1}^{n} B_{yw,j} w(k-\mu-j+1) \\ &+ \textstyle\sum_{j=1}^{\mu} H_{yu,j-2} u(k-j+1) \\ &+ \textstyle\sum_{j=1}^{n} B_{yu,j} u(k-\mu-j+1), \end{aligned} \tag{8}$$

with the Markov parameters

$$\begin{aligned} &H_{yu,-1} \triangleq D, \quad H_{yu,j} \triangleq CA^j B, \quad j \geq 0, \\ &H_{yw,-1} \triangleq D_2, \quad H_{yw,j} \triangleq CA^j D_1, \quad j \geq 0, \\ &H_{zu,-1} \triangleq E_2, \quad H_{zu,j} \triangleq E_1 A^j B, \quad j \geq 0, \\ &H_{zw,-1} \triangleq 0, \quad H_{zw,j} \triangleq E_1 A^j D_1, \quad j \geq 0. \end{aligned}$$

Next, for a data window p, define the data vectors

$$Z(k) \triangleq \begin{bmatrix} z(k) \\ \vdots \\ z(k-p+1) \end{bmatrix},$$

$$Y(k) \triangleq \begin{bmatrix} y(k) \\ \vdots \\ y(k-p+1) \end{bmatrix},$$

$$U(k) \triangleq \begin{bmatrix} u(k) \\ \vdots \\ u(k-p_c+1) \end{bmatrix},$$

where $p_c \triangleq \mu + n + p - 1$, and the regressor vectors

$$\Phi_{zw}(k) \triangleq \begin{bmatrix} z(k-\mu) \\ \vdots \\ z(k-\mu-p-n+2) \\ w(k) \\ \vdots \\ w(k-\mu-p-n+2) \end{bmatrix},$$

$$\Phi_{yw}(k) \triangleq \begin{bmatrix} y(k-\mu) \\ \vdots \\ y(k-\mu-p-n+2) \\ w(k) \\ \vdots \\ w(k-\mu-p-n+2) \end{bmatrix}.$$

Furthermore, define the block-Toeplitz ARMARKOV *weight matrices* W_{zw}, W_{yw} and block-Toeplitz ARMARKOV *control matrices* B_{zw}, B_{yw} as in [1]. Then (7) and (8) can be written as the ARMARKOV/Toeplitz model

$$Z(k) = W_{zw}\Phi_{zw}(k) + B_{zu}U(k), \tag{9}$$
$$Y(k) = W_{yw}\Phi_{yw}(k) + B_{yu}U(k). \tag{10}$$

The adaptive disturbance rejection algorithm for (9) and (10) uses a strictly proper controller in ARMARKOV form of order n_c and μ_c Markov parameters, so that, analogous to (7) and (8), the control $u(k)$ is given by

$$\begin{aligned} u(k) = &\sum_{j=1}^{n_\mathrm{c}} -\alpha_{\mathrm{c},j}(k)u(k-\mu_\mathrm{c}-j+1) \\ &+ \sum_{j=1}^{\mu_\mathrm{c}-1} H_{\mathrm{c},j-1}(k)y(k-j+1) \\ &+ \sum_{j=1}^{n_\mathrm{c}} B_{\mathrm{c},j}(k)y(k-\mu_\mathrm{c}-j+1), \end{aligned} \tag{11}$$

where $H_{\mathrm{c},j} \in \mathbb{R}$, $j = 1, 2, \ldots$, are the Markov parameters of the controller. Next, define the *controller parameter block vector* $\theta(k)$ by

$$\begin{aligned} \theta(k) \triangleq [&-\alpha_{\mathrm{c},1}(k) \ \cdots - \alpha_{\mathrm{c},n_\mathrm{c}}(k) \ \ H_{\mathrm{c},0}(k) \\ &\cdots \ H_{\mathrm{c},\mu_\mathrm{c}-2}(k) \ \ B_{\mathrm{c},1}(k) \ \ \cdots \ B_{\mathrm{c},n_\mathrm{c}}(k)]. \end{aligned}$$

Now from (11) it follows that $u(k)$ and $U(k)$ are given by

$$u(k) = \theta(k)R_1\Phi_{uy}(k), \tag{12}$$
$$U(k) = \sum_{i=1}^{p_\mathrm{c}} L_i\theta(k-i+1)R_i\Phi_{uy}(k), \tag{13}$$

where L_i and R_i are zero-one matrices defined in [1], and

$$\Phi_{uy}(k) \triangleq \begin{bmatrix} u(k-\mu_\mathrm{c}) \\ \vdots \\ u(k-\mu_\mathrm{c}-n_\mathrm{c}-p_\mathrm{c}+2) \\ y(k-1) \\ \vdots \\ y(k-\mu_\mathrm{c}-n_\mathrm{c}-p_\mathrm{c}+2) \end{bmatrix}.$$

Next, we define the *retrospective performance*

$$\hat{Z}(k) \triangleq W_{zw}\Phi_{zw}(k) + B_{zu}\sum_{i=1}^{p_\mathrm{c}} L_i\theta(k)R_i\Phi_{uy}(k),$$

which has the same form as (9) but with $\theta(k-i+1)$ replaced by the *current* controller parameter block vector $\theta(k)$. We thus define the *retrospective performance cost function*

$$J(k) \triangleq \frac{1}{2}\hat{Z}^\mathrm{T}(k)\hat{Z}(k). \tag{14}$$

Finally, the update law for the controller parameters is given by

$$\theta(k+1) = \theta(k) - \eta(k)\frac{\partial J(k)}{\partial \theta(k)}, \tag{15}$$

where $\eta(k)$ is the *adaptive step size*, and the gradient of $J(k)$ with respect to $\theta(k)$ is given by

$$\frac{\partial J(k)}{\partial \theta(k)} = \sum_{i=1}^{p_\mathrm{c}} L_i^\mathrm{T}B_{zu}^\mathrm{T}\hat{Z}(k)\Phi_{uy}^\mathrm{T}(k)R_i^\mathrm{T}. \tag{16}$$

By defining a *desired performance* $\hat{Z}^*(k)$ we can derive the *optimal step size*

$$\eta_\mathrm{opt}(k) = \frac{\|\hat{Z}^*(k) - \hat{Z}(k)\|_2^2}{\left\|\frac{\partial J(k)}{\partial \theta(k)}\right\|_\mathrm{F}^2}. \tag{17}$$

Since $\hat{Z}^*(k)$ is unknown, $\eta_\mathrm{opt}(k)$ is not computable. Hence we define the *implementable adaptive step size*

$$\eta_\mathrm{imp}(k) \triangleq \frac{1}{p_\mathrm{c}\|B_{zu}\|_\mathrm{F}^2\|\Phi_{uy}(k)\|_2^2}, \tag{18}$$

which satisfies

$$0 < \eta_\mathrm{imp}(k) < \eta_\mathrm{opt}(k). \tag{19}$$

The use of $\eta_\mathrm{imp}(k)$ guarantees a decrease in the cost.

IV. System Identification for a Linearized Flow Model

To obtain the required controller parameters for the ARMARKOV model, a CFD simulation is used with band-limited white noise signals injected into a near nominal (undisturbed) laminar flow from the control jets. Both the input and performance velocity measurements are recorded at a sampling rate of 100 Hz.

Following the open-loop CFD simulation, we implement a time domain system identification technique using ARMARKOV/Toeplitz Models [4], [5] to extract the necessary controller parameters, specifically the numerator coefficients of the transfer functions from the control inputs to the performance variables. These coefficients are subsequently used by the adaptive controller during a closed-loop simulation. The combination of CFD simulation and the adaptive disturbance rejection algorithm thus constitutes a technique for controlling a fluid without analytical modeling and with limited empirical (identification) modeling.

Now, we provide a brief review of the recursive ARMARKOV/Toeplitz algorithm for system identification [5]. Consider the n^{th}-order discrete-time finite-dimensional linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k), \tag{20}$$
$$y(k) = Cx(k) + Du(k). \tag{21}$$

Defining the ARMARKOV *regressor vector* $\Phi_{yu}(k)$ by

$$\Phi_{yu}(k) \triangleq \begin{bmatrix} y(k-1) \\ \vdots \\ y(k-\mu_c-n_c-p_c+2) \\ u(k-\mu_c) \\ \vdots \\ u(k-\mu_c-n_c-p_c+2) \end{bmatrix}, \qquad (22)$$

it follows that $Y(k) = W_{yu}\Phi_{yu}(k)$, where the ARMARKOV/Toeplitz *weight matrix* W_{yu} is the block-Toeplitz matrix defined in [5].

Let $\widehat{W}(k)$ denote an estimate of the ARMARKOV/Toeplitz weight matrix W_{yu} at time k, where $\widehat{W}(k)$ has the same block-zero structure as W_{yu}. Define the *estimated output vector* $\widehat{Y}(k) \triangleq \widehat{W}(k)\Phi_{yu}(k)$, the *output error* $\varepsilon(k) \triangleq Y(k) - \widehat{Y}(k)$, and the *output error cost function* $J(k) \triangleq \frac{1}{2}\varepsilon^{\mathrm{T}}(k)\varepsilon(k)$.

The gradient of $J(k)$ with respect to the estimated weight matrix $\widehat{W}(k)$ is given by

$$\frac{\partial J(k)}{\partial \widehat{W}(k)} \triangleq -U \circ [\varepsilon(k)\Phi_{yu}^{\mathrm{T}}(k)], \qquad (23)$$

where U is a zero-one matrix defined in [5].

We now consider the *estimated weight matrix update law*

$$\widehat{W}(k+1) = \widehat{W}(k) - \eta(k)\frac{\partial J(k)}{\partial \widehat{W}(k)}, \qquad (24)$$

where $\eta(k) \geq 0$ is the *adaptive step size*. Furthermore, define the *estimated weight matrix error* $E(k) \triangleq W_{yu} - \widehat{W}$, and the *estimated weight matrix error cost function* $J_W(k, \eta(k)) \triangleq \|E(k+1)\|_{\mathrm{F}}^2 - \|E(k)\|_{\mathrm{F}}^2$. Then it follows from the estimated weight matrix update law (24) that

$$E(k+1) = E(k) + \eta(k)\frac{\partial J(k)}{\partial \widehat{W}(k)}, \qquad (25)$$

$$\varepsilon(k) = E(k)\Phi_{yu}(k). \qquad (26)$$

For the flow simulations using one control input, an identified system of order $n = 20$ is used with a data window $p = 1$ step. For the flow simulations using two control inputs, a system of order $n = 50$ is used with a data window $p = 5$ steps. Figure 3 shows the predicted error norm generated by the Toeplitz system identification algorithm for the two control input case.

V. Results

Three types of CFD simulations are used to test the adaptive algorithm for steady flow field modification. First, we consider the case of only one actuator (u_1) and its ability to control the longitudinal flow direction at z, which is a velocity sensor aligned parallel to the nominal flow direction. Next, we again consider the case of one actuator (u_1) and its ability to control the transverse flow direction at z, which is now a velocity sensor aligned tangent to the nominal flow direction. Finally, we consider the case of two actuators (u_1, u_2) and their ability to control both the longitudinal and transverse flow directions at z, which are now two velocity sensors aligned perpendicular to one another. Independently controlling both the longitudinal and transverse flow directions at z constitutes modifying the nominal flow angle at that particular location. In all three cases, the objective is to control the flow field velocity, at the location of the performance sensor, to a desired steady-state value.

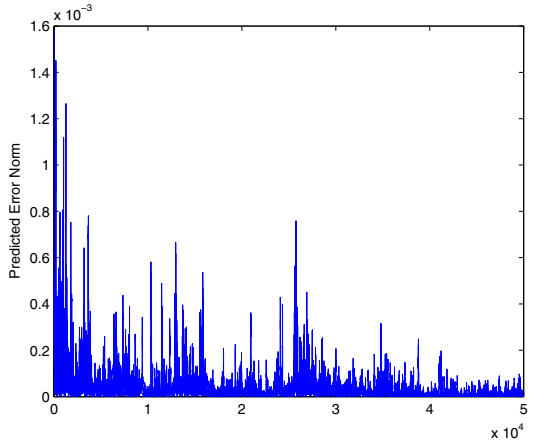

Fig. 3. Predicted error norm for the case of two control inputs, system order $n = 50$, and recursive data window $p = 5$ steps.

The CFD mesh is a rectangular grid with cell size 0.01 m $\times$ 0.01 m, resulting in 300 streamwise cells and 25 wall normal cells. The simulation step size is 0.01 s. Multiple simulations were run with different space and time discretizations to determine appropriate values for maintaining the underlying fluid dynamics.

We first use only one actuator (u_1) and show its ability to control the longitudinal flow component of z. Figure 4 shows the longitudinal flow velocities at the location of the performance sensor for several cases of steady longitudinal flow. In each of the four cases depicted, the flow velocity reaches steady-state in under 20 seconds. It should be noted that the transverse velocities, which are not shown, are not controlled in these cases. For these simulations, the controller is of order $n_c = 20$ with recursive data window $p = 1$ step.

Similarly, Figure 5 shows a plot of the steady-state flow field obtained from Fluent while controlling the longitudinal flow velocity at z to exactly 0.0 m/s. We see that the flow is nonzero in a neighborhood of z, but the flow is exactly stopped at the desired location. Figure 6 shows a plot in which the longitudinal flow velocity is controlled to 5.0 m/s. It is interesting to note that the control actuator induces reversed flow along the upper wall of the channel downstream from the input.

Next, we continue to use one actuator (u_1) and show its ability to control the transverse flow component of z. Figure 7 shows the transverse flow velocities at the location of the performance sensor for several cases of steady transverse flow. In the four cases depicted, the flow velocity essentially

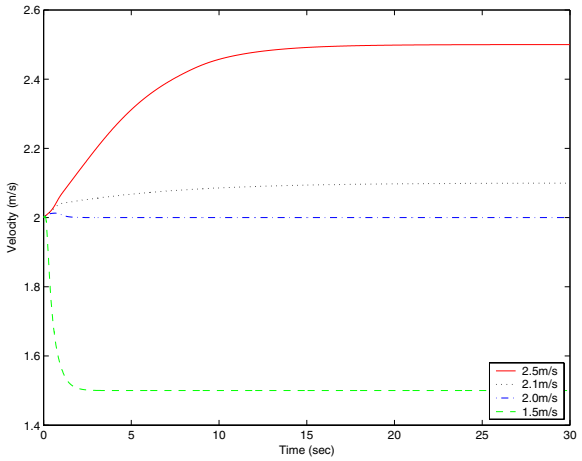

Fig. 4. Flow velocities at z-sensor for four cases of steady longitudinal flow. Nominal channel velocity is 2 m/s in the longitudinal direction.

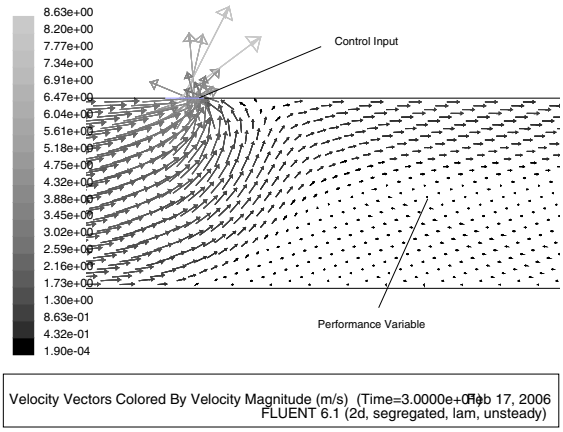

Fig. 5. Steady-state flow field velocities while controlling longitudinal z-sensor to 0.0 m/s. This plot shows the 2D channel zoomed in close to the control input and performance sensor

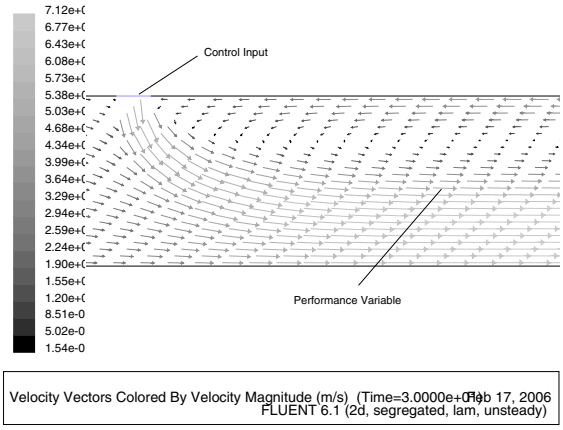

Fig. 6. Steady-state flow field velocities while controlling longitudinal z-sensor to 5.0 m/s. This plot shows the 2D channel zoomed in close to the control input and performance sensor

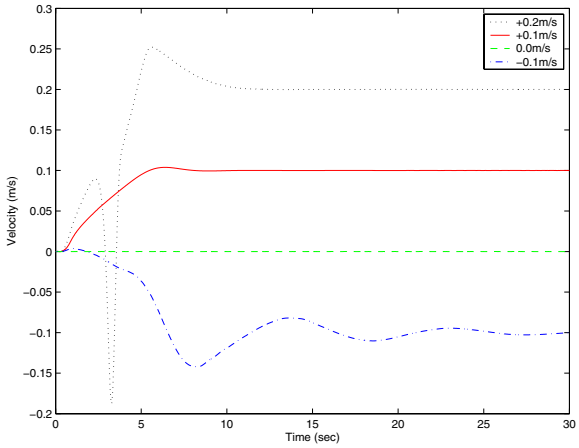

Fig. 7. Flow velocities at z-sensor for four cases of steady transverse flow. Nominal channel velocity is 2 m/s in the longitudinal direction.

reaches steady-state in about 25 seconds. It should be noted that the longitudinal velocities, which are not shown, are not controlled in these cases. For these simulations, the controller is of order $n_c = 20$ with recursive data window $p = 1$ step.

We see in Figure 7, for the case of controlling the transverse velocity to 0.2 m/s, the flow reverses direction before reaching its ultimate steady-state value. Throughout the simulations, using u_1 makes it difficult to arbitrarily control transverse velocities in the negative direction. We can see this in the case of controlling the transverse velocity to -0.1 m/s. Here, the flow oscillates before reaching steady-state. This is due to the dynamics of the channel and the location of u_1 with respect to z. Alternatively, using u_2 makes it difficult to control positive transverse velocities, as expected by symmetry.

We now present two-dimensional flow control results with the use of two control actuators. Figure 8 shows the flow field extracted from Fluent after a 50 second simulation. Here, the longitudinal z sensor is commanded to 2.0 m/s while the transverse z sensor is commanded to 0.2 m/s. This results in a 5.7 deg commanded deviation from the nominal flow. At 50 seconds, the flow hasn't quite reached steady state; the resulting angle at the performance sensors is 5.2 deg.

At some point, the flow dynamics reach physical limitations. From further simulation, it is clear that as the flow angle is commanded more forcefully, steady-state, if even achieved, takes a longer time to reach. Figure 9 shows the flow field extracted from Fluent after a 200 second simulation. Here, the longitudinal z sensor is commanded to 0.5 m/s while the transverse z sensor is commanded to 0.2 m/s. This results in a 21.8 deg commanded deviation from the nominal flow. At 200 seconds, the flow still hasn't reached steady state, and the resulting angle at the performance sensors is 12.0 deg. From a time-history plot, the longitudinal and transverse velocities at z are unresponsive to the control inputs, which continue to increase with positive slope at 200 seconds. Given longer simulation times with stressing cases, extreme values of the control inputs excite nonlinearities in the flow and affect the performance sensors' velocity. For these simulations, the controller is of order $n_c = 50$ with recursive data window $p = 5$ steps.

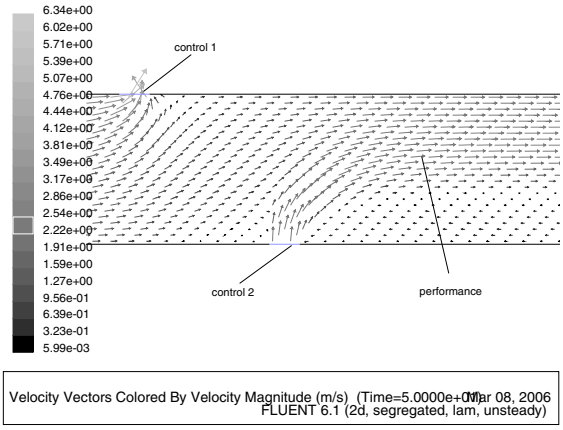

Fig. 8. Steady-state flow field velocities while controlling longitudinal z-sensor to 2.0 m/s and transverse z-sensor to 0.2 m/s. This corresponds to a 5.7 deg deviation from the nominal flow. This plot shows the 2D channel zoomed in close to the control inputs and performance sensors

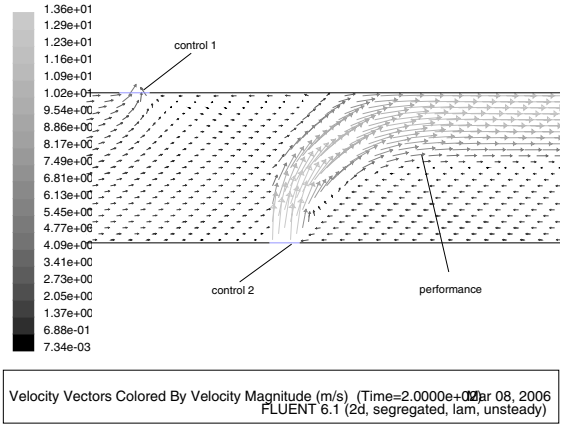

Fig. 9. Steady-state flow field velocities while controlling longitudinal z-sensor to 0.5 m/s and transverse z-sensor to 0.2 m/s. This corresponds to a 21.8 deg deviation from the nominal flow. This plot shows the 2D channel zoomed in close to the control inputs and performance sensors

As seen from the simulations, arbitrary flow specifications along the longitudinal or transverse directions are not difficult to achieve. Alternatively, specifying both the magnitude and angle of the flow at a point in the channel presents a more challenging problem.

VI. Conclusion

Adaptive flow control using the ARMARKOV adaptive algorithm for steady flow field modification was demonstrated by means of a 2D channel Poiseuille flow CFD simulation. System identification was used to estimate the numerator coefficients of the transfer functions from the control inputs to the performance variables. Steady flow field modification was demonstrated in both the 1D and 2D cases. No analytical model was derived or is needed for this adaptive algorithm, nor does the algorithm require constant tuning, as in PID control, for example. Future work will focus on refining the 2D control results for better performance as well as studying flow around objects in the flow field. This method can be used as a first step in applying adaptive algorithms within flow fields to modify their properties in a desired manner.

References

[1] R. Venugopal and D. S. Bernstein, "Adaptive disturbance rejection using ARMARKOV/Toeplitz models," *IEEE Trans. Contr. Syst. Technol.*, vol. 8, no. 2, pp. 257–269, 2000.

[2] S. Akhtar and D. Bernstein, "Discrete-time adaptive stabilization and disturbance rejection for minimum phase plants," in *Proc. Conf. Dec. Contr.*, Seville, Spain, 2005, pp. 2236–2241.

[3] "CFD Flow Modeling Software," Fluent Inc., February 2006, http://www.fluent.com/.

[4] J. C. Akers and D. S. Bernstein, "ARMARKOV least-squares identification," in *Proc. Amer. Contr. Conf.*, NM, June 1997, pp. 186–190.

[5] ——, "Time-domain identification using ARMARKOV/Toeplitz models," in *Proc. Amer. Contr. Conf.*, NM, June 1997, pp. 191–195.

[6] S. Akhtar and D. S. Bernstein, "Optimal adaptive feedback disturbance rejection," in *Proc. ACTIVE 04*, Williamsburg, VA, September 2004.

[7] J. Chandrasekar, L. Liu, D. Patt, P. P. Friedmann, and D. S. Bernstein, "Active noise cancellation for systems with uncertain dynamics using adaptive harmonic steady state control," in *Proc. ACTIVE 04*, Williamsburg, VA, September 2004.

[8] J. B. Hoagg and D. S. Bernstein, "Discrete-time adaptive feedback disturbance rejection using a retrospective performance measure," in *Proc. ACTIVE 04*, Williamsburg, VA, September 2004.

[9] J. B. Hoagg, S. L. Lacy, R. Venugopal, and D. S. Bernstein, "Adaptive control of a flexible membrane using acoustic excitation and optical sensing," in *AIAA Guid. Nav. Contr. Conf.*, Austin, TX, August 2003, AIAA-2003-5430.

[10] S. L. Lacy, R. Venugopal, and D. S. Bernstein, "ARMARKOV adaptive control of self-excited oscillations of a ducted flame," in *Proc. Conf. Dec. Contr.*, Tampa, FL, December 1998, pp. 4527–4528.

[11] H. Sane and D. S. Bernstein, "Active noise control using an acoustic servovalve," in *Proc. Amer. Contr. Conf.*, Philadelphia, PA, June 1998, pp. 2621–2625.

[12] H. Sane, R. Venugopal, and D. S. Bernstein, "Disturbance rejection using self-tuning ARMARKOV adaptive control with simultaneous identification," *IEEE Trans. Contr. Sys. Tech.*, vol. 9, pp. 101–106, 2001.

[13] M. Rizzo, M. Santillo, A. Padthe, J. B. Hoagg, S. Akhtar, K. Powell, and D. S. Bernstein, "CFD-based adaptive flow control using ARMARKOV disturbance rejection," in *Proc. Amer. Contr. Conf.*, Minneapolis, MN, June 2006.

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

[15] O. M. Aamo and M. Krstić, *Flow Control by Feedback : Stabilization and Mixing*, 1st ed. Great Britain: Springer, 2003.

[16] S. S. Joshi, J. L. Speyer, and J. Kim, "A systems theory approach to the feedback stabilization of infinitesimal and finite-amplitude disturbances in plan poiseuille flow," *J. of Fluid Mechanics*, vol. 332, pp. 157–184, 1997.

[17] ——, "Finite dimensional optimal control of poiseuille flow," *J. Guid. Contr. Dynamics*, vol. 22, no. 2, pp. 340–348, 1999.

[18] L. Cortelezzi, K. Lee, J. Kim, and J. Speyer, "Skin-friction drag reduction via robust-order linear feeback control," *Int. J. Comput. Fluid Dyn.*, vol. 8, no. 1-2, pp. 79–92, 1998.

[19] L. Cortelezzi and J. Speyer, "Robust reduced-order controller of laminar boundary layer transitions," *Physical Review E*, vol. 58, no. 2, pp. 1906–1910, 1998.

[20] A. Emami-Naeini, S. A. McCabe, D. de Roover, J. L. Ebert, and R. L. Kosut, "Active control of flow over a backward-facing step," in *Proc. Conf. Dec. Contr.*, Seville, Spain, 2005, pp. 7366–7371.

[21] P. Christofides and A. Armaou, "Nonlinear control of navier-stokes equations," in *Proc. Amer. Cont. Conf.*, PA, 1998, pp. 1355–1359.

[22] R. Vazquez and M. Krstic, "A closed-form feedback controller for stabilization of linearized navier-stokes equations: The 2d poisseuille flow," in *Proc. Conf. Dec. Contr.*, Seville, Spain, 2005, pp. 7358–7365.

[23] J. A. Burns and J. Singler, "Feedback control of low dimensional models of transition to turbulence," in *Proc. Conf. Dec. Contr.*, Seville, Spain, 2005, pp. 3140–3145.

[24] T. R. Bewley, P. Moin, and R. Temam, "DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms," *Journal of Fluid Mechanics*, vol. 447, pp. 179–225, 2001.

[25] A. Pillarisetti and L. N. Cattafesta, "Adaptive identification of fluid-dynamic systems," in *Proc. AIAA Fluid Dynamics Conf.*, Anaheim, CA, June 2001, AIAA 2001-2978.