Application of Neural Network to Turbulence Control for Drag Reduction

Changhoon Lee*, John Kim*, David Babcock** & Rodney Goodman**

** Department of Mechanical and Aerospace Engineering, UCLA, USA*

*** Department of Electrical Engineering, Caltech, USA*

Abstract

A new adaptive controller based on a neural network was constructed and applied to turbulent channel flow for drag reduction. A simple control network, which employs blowing and suction at the wall based only on quantities measured at the wall, was shown to reduce the skin friction by as much as 20% in direct numerical simulations of low-Reynolds number turbulent channel flow. Also, a pattern was observed in the distribution of weights associated with the neural network. This allowed us to derive a simple control scheme that produced the same amount of drag reduction more efficiently.

1 Introduction

The ability to control turbulent flows is of significant economic interest. Successful control of turbulent boundary layers by reducing drag, for example, can result in a substantial reduction in operational cost for commercial aircraft and marine vehicles. Recent studies show that near-wall streamwise vortices are responsible for high skin-friction drag in turbulent boundary layers (Kim 1992; Choi *et al.* 1994). Some attempts have been made to reduce the skin-friction drag by controlling the interactions between these vortices and the wall. Choi *et al.* (1994), for example, used blowing and suction at the wall equal and opposite to the wall-normal component of velocity at $y^+ = 10$, giving approximately 25% drag reduction in a turbulent channel flow. Although the method employed in their work is impractical, since the information at $y^+ = 10$ is usually not available, it demonstrates a control scheme by which the skin-friction drag can be reduced by manipulation of the near-wall streamwise vortices.

A systematic approach using suboptimal control theory has also been tried in the past. This approach, which attempts to minimize a cost function, was applied to the stochastic Burgers equation and was found to be successful (Choi *et al.* 1993). Moin & Bewley (1995) applied a similar approach to a turbulent channel flow to achieve up to 50% drag reduction. This approach, however, requires information from the entire velocity field inside the flow domain and excessive computational time, making it impractical to implement such a scheme in real situations. For practical implementation,

a control scheme should be based only on quantities that are easily measurable at the wall, and should be fast enough to be applied in real time. Our goal is to find such a scheme.

We seek wall actuations, in the form of blowing and suction at the wall, dependent on the wall-shear stress to achieve a substantial skin-friction reduction. This requires knowledge of how the wall-shear stresses respond to wall actuations, i.e., the correlation between the wall-shear stresses and the wall actuations. Because of the complexity of solutions to the Navier-Stokes equations, however, it is not possible to find such a correlation in closed form or to approximate it in simple form. We use instead a neural network to approximate the correlation and then to find the optimal wall actuations that will give the minimum value of the skin-friction drag. Neural networks have been used to obtain complicated, nonlinear correlations without *a priori* knowledge of the system that is to be controlled. Jacobson & Reynolds (1993), for instance, used a neural network to obtain about 7% drag reduction in their simulation of an artificial flow. In this paper, we describe how we construct and train a neural network, and then constructed a control scheme for drag reduction based on that neural network. We then applied this control scheme to direct numerical simulations of turbulent channel flow at low Reynolds number, and observed about 20% drag reduction. We then describe how examination of the weight distribution from the trained neural network led to a very simple control scheme that worked equally well while being more efficient.

In section 2, a brief description of the architecture of the neural network used in the present work is given. In section 3, results obtained from control using an off-line trained network are presented, while results obtained from control using a network with continuous on-line training are given in section 4. In section 5, a simple control scheme derived from the weight distribution in a successful neural network control is presented. A few turbulence statistics are given in section 6, followed in section 7 by a discussion of practical implementation and the conclusions.

In this paper we use (x, y, z) for the streamwise, wall-normal, and spanwise coordinates, respectively, and (u, v, w) for the corresponding velocity components.

2 Neural Network

In this section we describe the construction of a neural network to learn the correlation between wall-shear stresses and wall actuations from given data set. Although a neural network generally requires no prior knowledge of the system (or "plant"), knowledge about the near-wall turbulence structures provides a guideline for the design of the network architecture. Initially $\partial u/\partial y$ and $\partial w/\partial y$ at the wall were used as input data fields and v at the wall was used for the output data of the network. Later, it was found that using only $\partial w/\partial y$ at the wall from one instance of time

was sufficient for proper training, and that a controller based on this network performed very well. Subsequently, we used only $\partial w/\partial y$ at the wall as input. Because we wanted the output to be based on a local input area, we designed our network using shared weights. The network had a single set of weights that is convolved over the entire input space to generate output values. The template (i.e. the set of network weights) extracts spatially invariant correlations between input and output data. The size of the template was initially chosen to include information about a single streak and streamwise vortex, and then was varied to find an optimal size. As for the network architecture, we used a two-layer network with hyperbolic tangent hidden units and a linear output unit (see figure 1). The functional form of the neural network is:

$$v_j = W_a \tanh\left(\sum_{i=j-N/2}^{j+N/2} W_i \left.\frac{\partial w}{\partial y}\right|_{w_i} + W_b\right) + W_c \ , \tag{1}$$

where the W's denote weights, N is the total number of input weights, and the subscript j indicates discrete wall locations. The summation is done over the spanwise direction. Seven neighboring points, including the point of interest, in the spanwise direction (corresponding to approximately 90 wall units with our numerical resolution) were found to provide enough information to adequately train and control the near-wall structures responsible for the high skin friction. A scaled conjugate gradient learning algorithm (Moller 1993) was used to produce rapid training. The sum of a weighted-squared error given by

$$Error = \frac{1}{2}\sum_j e^{k|v_{des,j}|}(v_{des,j} - v_{net,j})^2 \tag{2}$$

was minimized during training, where v_{des} is the desired output value and v_{net} is the network output value. Note that the error exponentially emphasizes (proportional to k) large actuations. Usually within 100 training epochs, the error reached its asymptotic limit.

3 Off-line Training and Control

As an initial experiment we investigated whether we could train a neural network to predict the velocity at $y^+ = 10$ from only the wall-shear stresses. The rationale behind this experiment was that if a neural network could be trained to predict v at $y^+ = 10$ based on the wall-shear stresses, the output from the network could be used as input to the actuator. The network should thus yield a similar amount of drag reduction to that obtained by Choi *et al.* (1994) without the knowledge of the velocity at $y^+ = 10$. The training data consisted of 100 time steps of output obtained from a numerical simulation of channel flow under the control employed by Choi *et al.* (1994), i.e., using the wall-normal velocity at $y^+ = 10$. The flow regime is turbulent channel flow with $Re_\tau = 100$,

where Re_τ is the Reynolds number based on the wall-shear velocity, u_τ, and the channel half-width, δ. All numerical simulations presented in this paper were obtained using a modified version of Kim *et al.*'s (1987) spectral code with the computational domain $(4\pi, 2, 4\pi/3)\delta$, and a grid resolution of (32, 65, 32) in the (x, y, z) directions, respectively. Each time step contained a 32×32 array of input values $(\partial w/\partial y|_w)$ and corresponding actuations, $-v$ at $y^+ = 10$.

We trained several networks with one hidden unit and different sized input templates: 7×1, 7×3, 7×5, 9×1, 9×3, 11×1, and 11×3 (the number of input units in the spanwise direction by the number of input unit in the streamwise direction). After training was completed, the distribution of input weights was examined to see whether there was a discernible pattern in the input template. The weight distributions in the spanwise direction at the same streamwise location for different input templates are shown in figure 2. The same pattern for all 7 input template sizes was observed, and is similar to a finite difference approximation of spanwise differentiation. Increasing the number of hidden units improved the performance marginally, whereas increasing the template size reduced the final training error.

We then applied a control scheme to a regular channel flow using the fixed input weights obtained from the off-line training. This was perhaps a somewhat naive approach since the weights were obtained from fully controlled flow data and the shear stresses used for the training were already altered by the actuations. Nevertheless, two cases were tested: a control scheme based on 7 weights in the spanwise direction, and another based on the same 7 weights plus 3 more in the immediate downstream location. These 10 weights were chosen because among all weights obtained from the off-line training they had non-negligible values. In figure 3, the mean shear stress variations at the wall (i.e., drag) obtained with these two controls are plotted along with the no-control case. Nearly 18% drag reduction was achieved, with slightly better performance from the 10-weight network.

These results demonstrate that a correlation exists between the shear stresses at the wall and the desired actuations, and that control based on this correlation produces a significant amount of drag reduction. This fixed-weight control scheme, however, was deduced from fully controlled flow data; thus it does not guarantee the same performance for other flow situations.

We conclude this section by remarking that our initial network structures, which employed $\partial u/\partial y|_w$ as well as $\partial w/\partial y|_w$ using a bigger template (7×5), did not produce a fixed pattern of weights. This suggests that the large number of weights, together with the additional information from $\partial u/\partial y|_w$, reduced the capability of the network to identify relevant flow structures.

4 On-line Control

In the previous section we showed how successful control based on off-line trining can be obtained. However, since the system we are trying to control is time-varying and nonlinear, this approach is not likely to be successful in general. Continuous on-line training allows a controller to adapt to the evolution of the system. In this section, we describe an adaptive controller for on-line training and control.

There are various schemes for on-line neural network control. The most direct scheme is adaptive-inverse model control (Widrow 1986). A schematic representation of this approach is shown in figure 4. Here the plant denotes the numerical solver of the Navier-Stokes equations. This configuration employs a neural network to model the (possibly time-varying) inverse plant mapping the wall shear stress $\partial w/\partial y|_w$ to the wall-normal actuations, and then uses a copy of the model as the controller with the desired shear stresses as input. One restriction of this technique is that it usually requires an initial model training phase using random plant inputs and corresponding plant outputs. This, however, caused no serious problem since usually one timestep was enough for the model training in our application. Once the model represents a reasonably close approximation to the actual plant inverse, a copy is then implemented as a feedforward controller.

The desired inputs to the controller are a fractional reduction in the shear stress from the previous step, i.e.,

$$\left(\overline{\frac{\partial w}{\partial y}}\right)_{t+1} = \eta \left(\frac{\partial w}{\partial y}\right)_t \tag{3}$$

where $0 < \eta < 1$. This indirect suppression of $\partial w/\partial y|_w$, instead of $\partial u/\partial y|_w$, turns out to be more efficient in achieving drag reduction. The output of the controller, which is the input to the plant, is the predicted actuation necessary to produce this shear stress reduction. The quantity η should be chosen such that sets of the desired large amplitude outputs are among the sets that are well represented by the training sets. Good performance was achieved for the range of $\eta = 0.8 \sim 0.85$.

A turbulent channel flow at Reynolds number $Re_\tau = 100$ was used to test the neural network. We allowed all the weights in the network to adapt and examined the input template pattern after each time step. As the control began, the weight distribution immediately assumed a fixed pattern. There was no appreciable change in the relative magnitudes of the template weights over time, indicating that the pattern is preserved. The absolute magnitudes, however, did vary indicating that the need for gain and bias adaptation for each layer. The number of hidden layers, the size of the hidden feature detector, the size of the input template, and the error scale of the training error were 1, 1×1, 7×1, and 5, respectively. We also varied the values of the error scale, and found 5 to be optimum, with larger values causing an instability in training. The converged weight

distribution for 20 consecutive time steps after an asymptotic state was reached are shown in figure 5. The pattern is only slightly different from the one obtained from the off-line training (see figure 2). It should be noted that this pattern emerged immediately after the on-line control began.

Time histories of the wall-shear stress for 3 different input template sizes are shown in figure 6. As the control began, the drag quickly drops to about 80% of that observed without control, for the two cases with template sizes 7×1 and 9×1. The template size of 5×1, however, did not produce as much reduction, implying that at least 7 spanwise points (which extended about 90 wall units, with our grid resolution) should be used for good performance. At the initial stage of our study, we tested a control using both $\partial w/\partial y|_w$ and $\partial u/\partial y|_w$ as input, with a 7×5 input template. It produced about the same amount of reduction, but the training time was excessive due to the large number of weights and it did not produce a coherent pattern in the weight distribution.

Since the seven weights show a fixed pattern, we fixed the input template weights to this pattern and used a single hidden unit network, giving only four adaptable parameters (a bias and gain for each layer). This simplified network had the following functional form:

$$v_w = W_a \tanh(W_b g + W_c) + W_d \tag{4}$$

with

$$g = \sum_{j=-3}^{3} W_j \left.\frac{\partial w}{\partial y}\right|_j \tag{5}$$

where W_j's are the fixed-weight pattern obtained from the previous on-line control. On-line control using this network also produced the same amount of drag reduction. The weight variation with time was also monitored. The bias weights (W_c and W_d) were negligibly small, but those controlling the gain (W_a and W_b), which had finite values, changed in time significantly, although the product of the two gain-weights remained almost constant. This suggests that effective control can be achieved by simply using g with an adjustable amplitude. This will be discussed in the following section.

5 A Simple Control Scheme

Since the control based on the network given by equation (4) produced a substantial reduction in drag, this section develops a control scheme based only on the weighted sum of the wall shear stress. The distribution of the weights can be approximated by (see figure 5):

$$W_j = A\frac{1 - \cos(\pi j)}{j} \tag{6}$$

where $j = 0$ corresponds to the point of interest. Since only the relative values are important, the constant A has no special meaning. A highly resolved computation was run to confirm that

equation (6) gives the proper form of the weight distribution. It turns out that equation (6) is the inverse Fourier transform of $\mathrm{i}k_z/|k_z|$, where k_z is the wavenumber in the spanwise direction, for the finite maximum wavenumber, i.e.,

$$\int_{-k_m}^{k_m} \frac{\mathrm{i}k_z}{|k_z|} \exp(-ik_z z)\mathrm{d}k_z = 2\frac{1-\cos(k_m z)}{z} \tag{7}$$

where $k_m = \pi/\Delta z$ is the maximum wavenumber and Δz is the numerical grid spacing. Replacing z with $j\Delta z$ in the right-hand-side of equation (7) leads to equation (6). From this result and the convolution theorem, one can deduce the following simple control law for wall transpiration:

$$\widehat{v}_w = C\,\frac{\mathrm{i}k_z}{|k_z|}\widehat{\frac{\partial w}{\partial y}}\bigg|_w = C\frac{1}{|k_z|}\widehat{\frac{\partial^2 w}{\partial z \partial y}} \tag{8}$$

where the 'hat' denotes a Fourier transformed quantity and C is a positive scale factor determining the amplitude of the actuation. Equation (8) implies that the optimum blowing and suction at the wall is proportional to $\partial^2 w/\partial z\partial y$ with the high wavenumber component suppressed by $1/|k_z|$. Note that the weights at even numberd grid spacing away from the center point vanish. This is beneficial for physical implementation, since sensors and actuators cannot be placed at the same location. Because the Fourier integral is computed for a finite value of k_m, the values of the weight at non-integer j in equation (6) have no meaning. The above control law is equivalent to:

$$v_w = C\sum_{j=-\infty}^{\infty} W_j\,\frac{\partial w}{\partial y}\bigg|_j \tag{9}$$

where W_j is given by equation (6). The magnitude of the weights decays with increasing distance from the center, which allows for good approximation with only a small number of weights; i.e., successful control requires only local values of the shear stress. A natural concern is how changing grid spacing will affect the control. Since the above control law (equation (9)) is simply another expression of equation (8), which is a good approximation as long as k_m is large enough, the control should be relatively independent of resolution.

Control based on the above scheme (equation (9)) with only 7 points produced the same amount of drag reduction (20%) as the neural network control did (see figure 7). The constant C is chosen so that the root-mean-squared (rms) value of the actuation is kept at 0.15 u_τ. Blowing and suction of this magnitude at the wall suppress the near-wall streamwise vortices by counteracting the up-and-down motions associated with these vortices. This result is consistent with Choi *et al.*'s (1994) results, which were obtained by active control based on flow information near the wall.

6 Turbulence Statistics

The computed flow fields for a no-control case and a successful control case, based on the 7-point weighted sum of $\partial w/\partial y|_w$ (equation (9)), were examined to investigate the mechanism by which the drag reduction is achieved. The most salient feature of the controlled case was that the strength of the near-wall streamwise vortices was drastically reduced. In figure 8, contours of streamwise vorticity in a cross plane are shown. This result further substantiates the notion that a successful suppression of the near-wall streamwise vortices leads to a significant drag reduction. Note that for the controlled case the wall actuations were applied at both walls.

The probability-density function of the wall-shear stress in the streamwise direction (figure 9) indicates that the control is effective in suppressing large fluctuations, thus reducing the mean skin-friction as well as its variance. Furthermore, the rms values of turbulent fluctuations in the wall region are also reduced, as shown in figure 10. The same trend was observed by Choi *et al.* (1994). Finally, an actuation distribution at the wall used in our control compared with that from Choi *et al.*'s (1994) v-control using the information at $y^+ = 10$ for the same wall-shear stress distribution is shown in figure 11. They show a strikingly similar distribution to each other, even though the wall actuation of our control is based only on the wall-shear stress $\partial w/\partial y|_w$.

7 Discussion and Conclusion

We have presented a successful application of a neural network to turbulence control for drag reduction. First we were able to construct and train a neural network off-line to find a correlation between the wall-shear stress and the desired wall actuations. Based on the optimal network structure from the off-line training, we successfully implemented an on-line inverse model controller in numerical experiments of a turbulent channel flow, resulting in about 20% drag reduction. Finally, we were able to deduce a simple control scheme (equation (9)) for drag reduction based on observation of the weight distribution from a successful control case. This control scheme uses a minimal amount of wall-shear stress information and requires only simple operations, thus rendering a scheme whose actual implementation would be relatively easy.

There are still several issues that must be addressed. First, our numerical experiments were performed in a very low Reynolds number flow. It remains to be seen whether the same control scheme extends to higher Reynolds numbers. If the main cause of high-skin friction in turbulent boundary layers at higher Reynolds numbers is also due to the near-wall streamwise vortices, the same scheme should work equally well. Detection of these vortices through the wall-shear stress

would become more difficult, however, because the scales associated with these vortices decrease as the Reynolds number increases.

Another important issue is the influence of the spatial resolution of sensors and actuators on performance. We showed that a simple control scheme with 7 neighboring points in the spanwise direction, which corresponds to 90 wall units, performed very well. This suggests that the distance between sensors should be about 10 wall units. As the Reynolds number increases, the physical separation between sensors must decrease. We note in passing that recent advances in micromachined sensors and actuators make those scale feasible (Liu *et al.* 1994), and the present work is part of a joint research project aimed at integrating micron-sized sensors and actuators with control algorithms implemented in analog VLSI.

A third issue is determining an appropriate scale factor C in equation (9). We tested a range of C and found that the value that yielded an actuation rms value between $0.1u_\tau$ and $0.15u_\tau$ gave the best performance. Smaller values resulted in less reduction, while larger ones caused rapid fluctuations of the wall-shear stress in time. The experiments were only performed for $Re_\tau = 100$ and 180, from which the optimum value was deduced, but we expect that the same amplitude level should produce similar reductions for higher Reynolds number flows. We further note that for this amplitude the required power input to produce the actuation was negligible compared to the power saved due to the reduced drag.

One final practical issue worth mentioning is the time delay between sensing and actuation. None was included in any of our numerical experiments. In a real situation, however, there will be a finite time delay between sensing and actuation. A related issue is the location of sensors and actuators. Ideally, sensors should register the response of the flow to be controlled due to wall actuations. An actuation signal, however, might produce an immediate spurious response at nearby sensors. One possible remedy to this problem is an underrelaxation of the actuation signal with past signals. For example, an underrelaxation using the following formula:

$$v_w^{t+1} = \xi C \sum_j W_j \left.\frac{\partial w}{\partial y}\right|_j^t + (1-\xi) v_w^t \tag{10}$$

could take into account all past signals with a single parameter ξ that has a value between 0 and 1. We used this approach successfully in our numerical experiment for $Re_\tau = 180$ and found that there is an optimum ξ depending on temporal resolution.

The most significant finding of the present study is that a single spanwise strip of information of $\partial w/\partial y|_w$ is enough to achieve significant drag reduction. Additional information about $\partial w/\partial y|_w$ in the streamwise direction, or $\partial u/\partial y|_w$, reduced the efficiency of our neural-network based control. We also tested different sized input templates and found that as the template size increased, the

fixed pattern in the weight disappeared, indicating that the feature-detecting capability of the neural network decreases when too many weights are used.

ACKNOWLEDGMENTS

We are grateful to Dr. Gary Coleman for comments on a draft of this manuscript. This work is supported by AFOSR grants F49620-93-1-0332 and F49620-95-1-0263 (Dr. James M. McMichael, Technical Monitor). It is also supported in part by the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation Engineering Research Center Program under grant EEC-9402726. Computer time has been supplied by the San Diego Supercomputer Center and by the NAS program at NASA Ames Research Center.

REFERENCE

Choi, H., Moin, P. & Kim J. 1994 Active turbulence control for drag reduction in wall-bounded flows. *J. Fluid Mech.* **262,** 75.

Choi, H., Temam, R., Moin P. & Kim J. 1993 Feedback control for unsteady flow and its application to the stochasitic Burgers equation. *J. Fluid Mech.* **253,** 509.

Jacobson, S. L. & Reynolds, W. C. 1993 Active control of boundary layer wall shear stress using self-learning neural networks. *AIAA Shear Flow Conference,* 1, Orlando.

Kim, J. 1992 Study of turbulence structure through numerical simulations: The perspective of drag reduction. *AGARD Report R-786, AGARD FDP/VKI Special Course on "Skin Friction Drag Reduction,"* March 2–6, VKI, Brussels, Belgium.

Kim, J., Moin, P. & Moser, R. 1987 Turbulence statistics in fully-developed channel flow at low Reynolds number. *J. Fluid Mech.* **177,** 133.

Liu, C., Tai, Y. C., Huang, J. B. & Ho, C.M. 1994 Surface micromachined thermal shear stress sensor. In *ASME Application of Microfabrication to Fluid Mechanics,* pp. 9-15, Chicago.

Moin, P. & Bewley T. 1995 Application of control theory to turbulence. In *Twelfth Australasian Fluid Mechanics Conference,* 109, December 10-15, Sydney.

Moller, M. 1993 Efficient Training of Feed-Forward Neural Networks. Ph.D. Thesis, Aarhus University, Denmark.

Widrow, B. 1987 Adaptive inverse control. In *Second IFAC Workshop on Adaptive Systems in Control and Signal Processing,* pp. 1-5, Lund, Sweden.

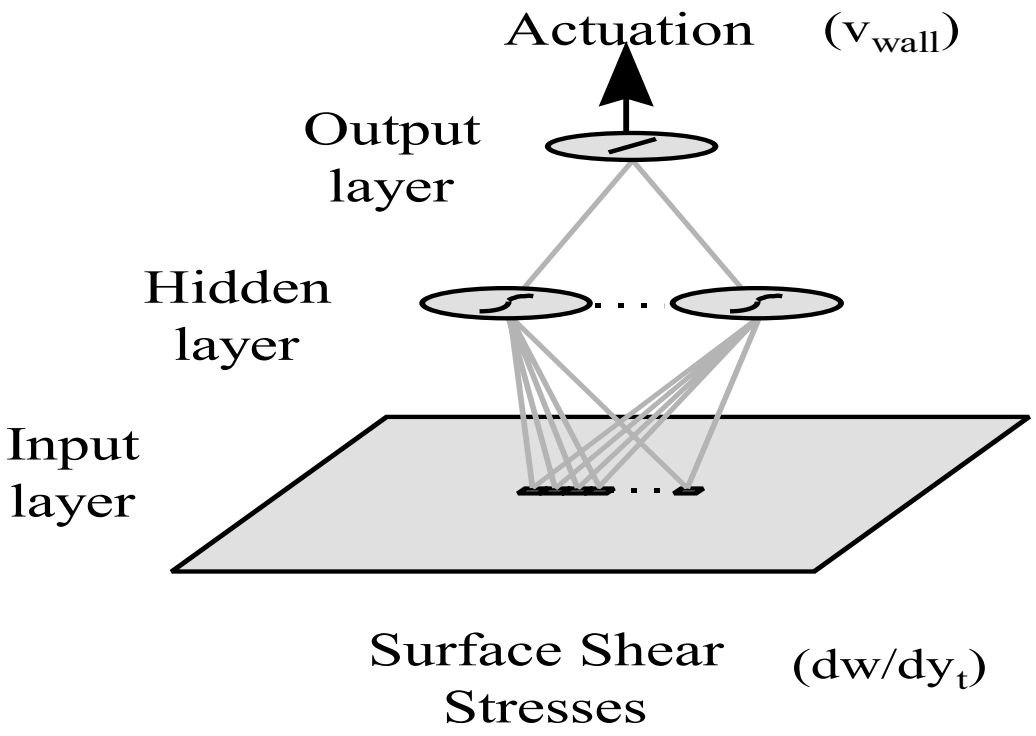


Figure 1: Neural network architecture.

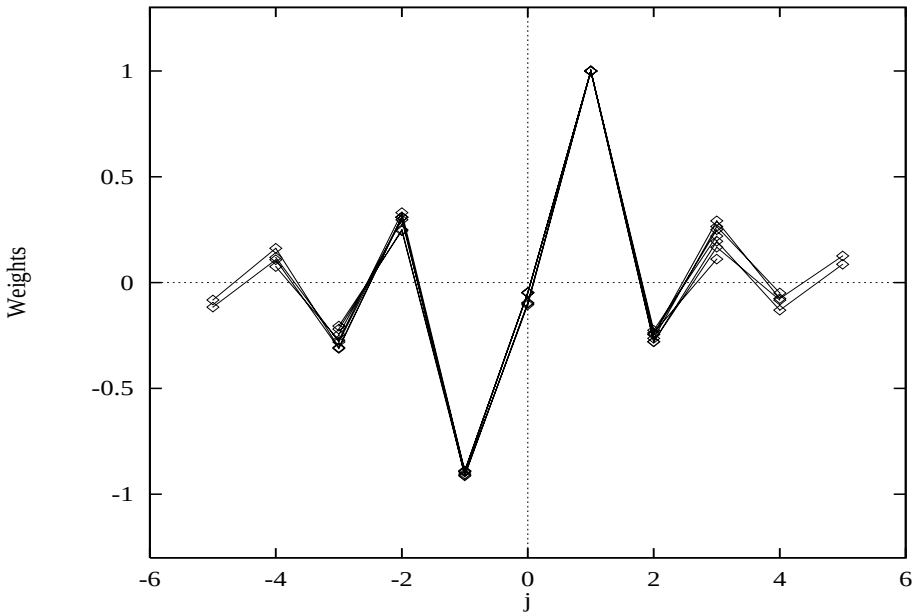


Figure 2: Weight distribution from off-line trainings for various sizes of the input template: 7x1, 7x3 7x5, 9x1, 9x3, 11x1, 11x3. Weights are normalized by W_1.

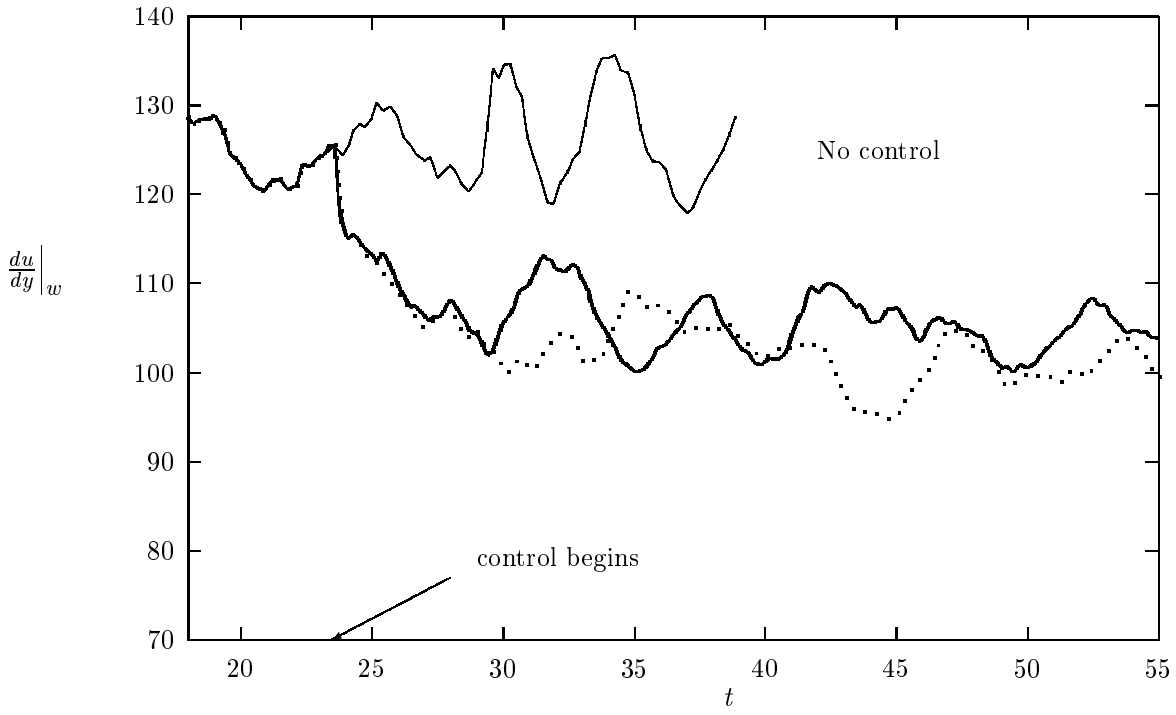


Figure 3: Mean wall-shear stress histories for various control laws compared to the no-control case: ——— , no control; ——— , control with 7 fixed weights obtained from off-line control; ····· , control with 10 fixed weights from the same off-line control.

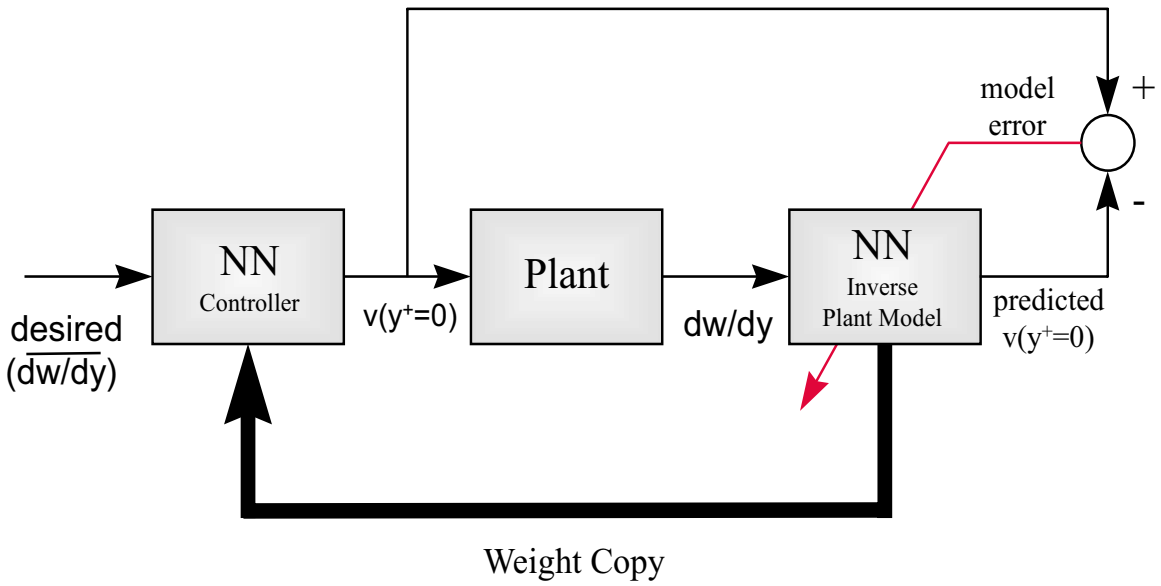


Figure 4: Schematic representation of adaptive inverse model control.

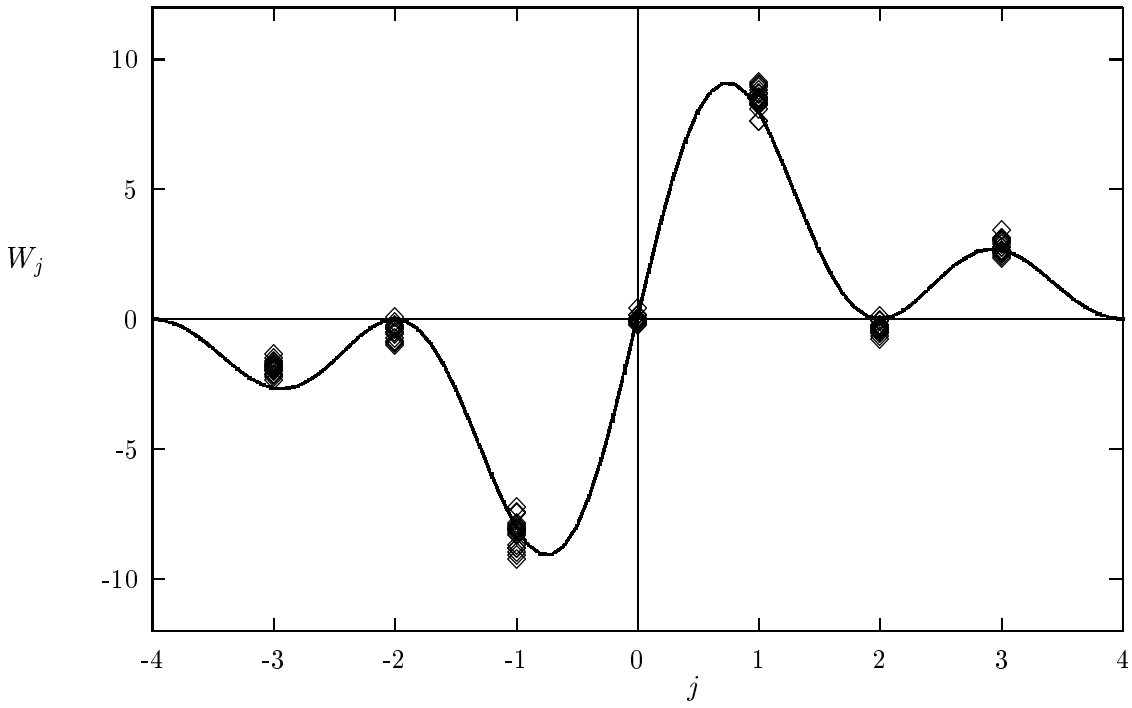


Figure 5: Distribution of the weights:$\diamond$, from on-line training; ——— , $A(1-\cos(\pi j))/j)$.

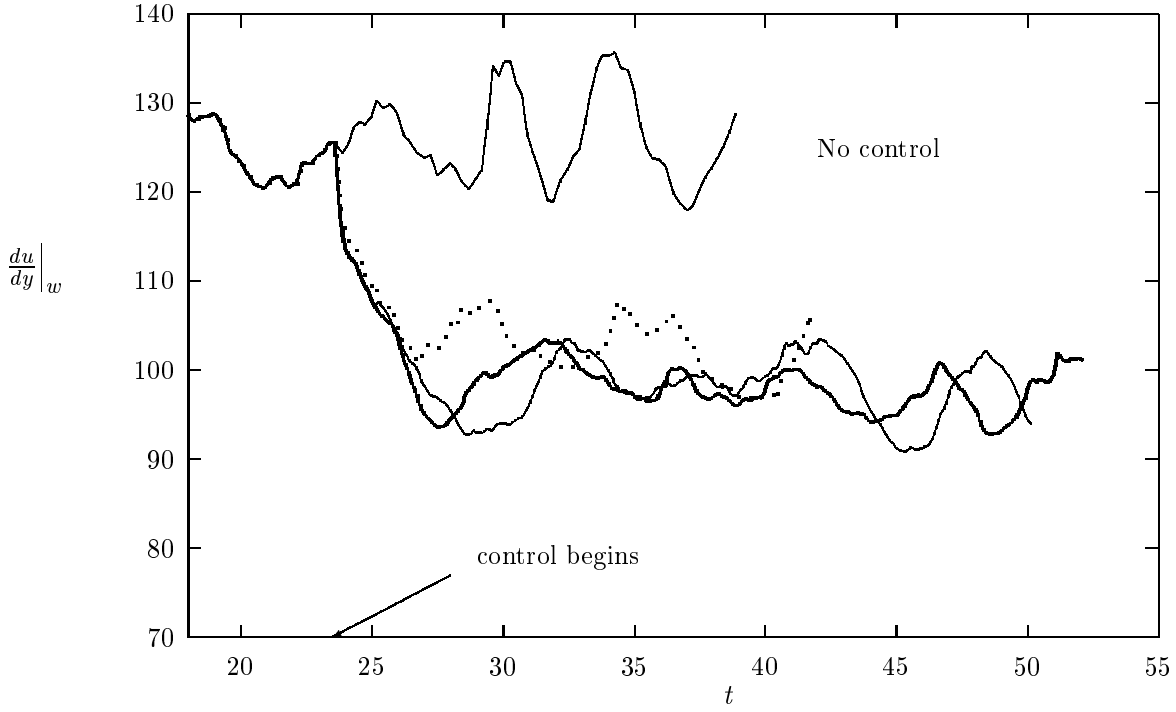


Figure 6: Mean wall-shear stress histories for on-line control with different input template sizes: ——— , no control; ····· , control with 5x1; ——— , control with 7x1; ——— , control with 9x1.

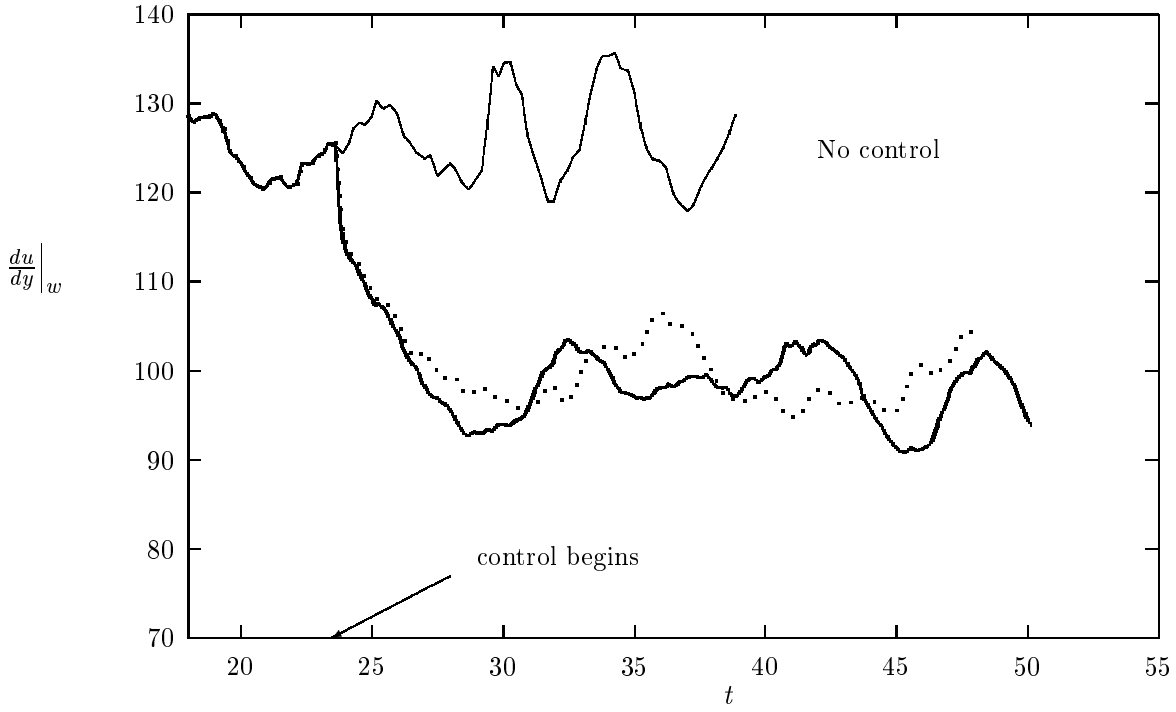
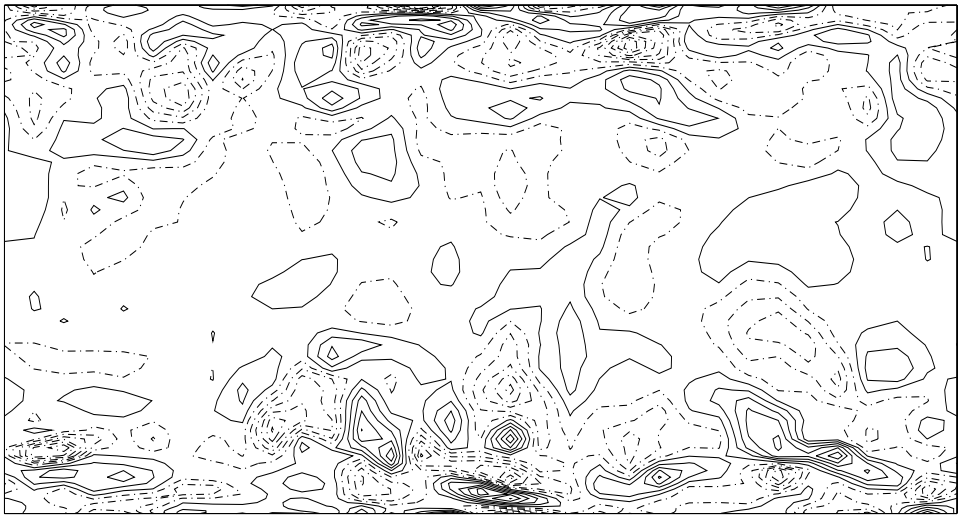


Figure 7: Mean wall-shear stress histories for various control schemes compared to the no-control case: ——— , no control; ——— , on-line control with neural network with 7x1 template; ····· , control with 7 fixed weights.

(a)



(b)

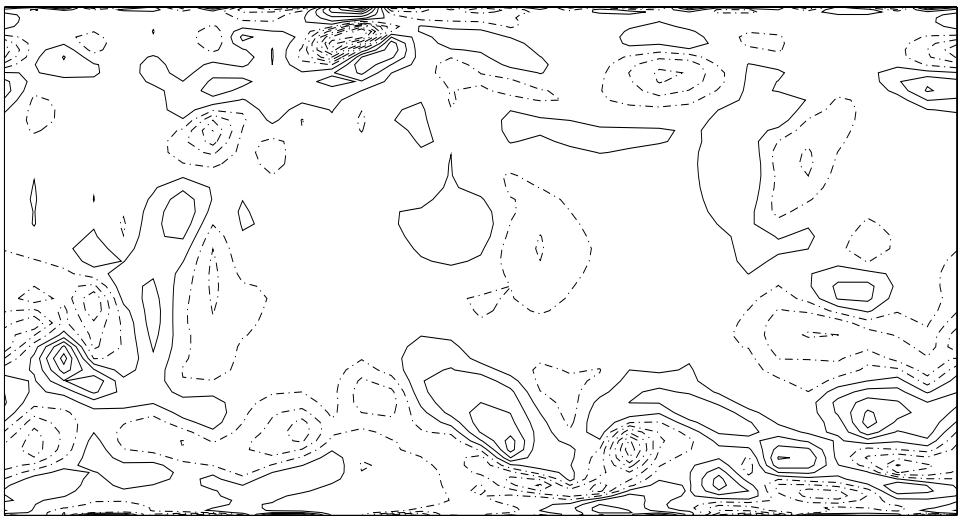


Figure 8: Contours of streamwise vorticity in a cross-flow plane: (a) no control; (b) control using 7 fixed weights. The contour level increment is the same for both figures. Negative contours are chain-dotted.

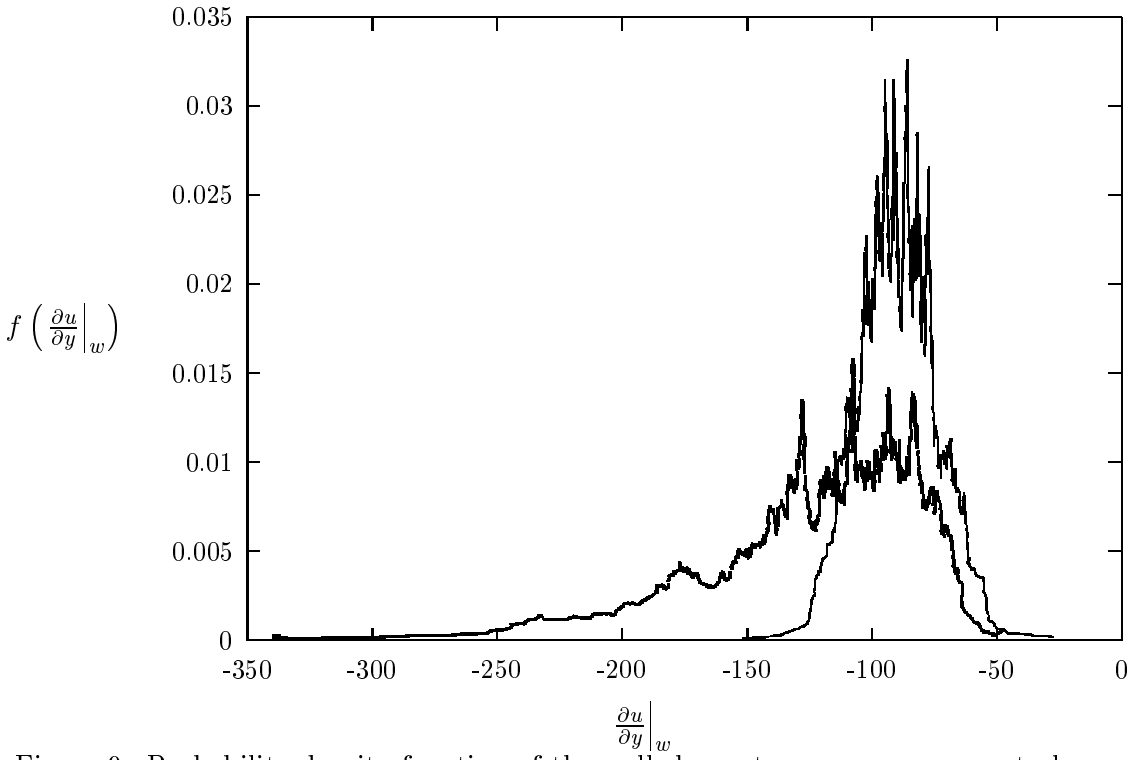


Figure 9: Probability-density function of the wall-shear stress: ——, no control; —— , control with 7 fixed weights. The area under each curve is normalized to one.

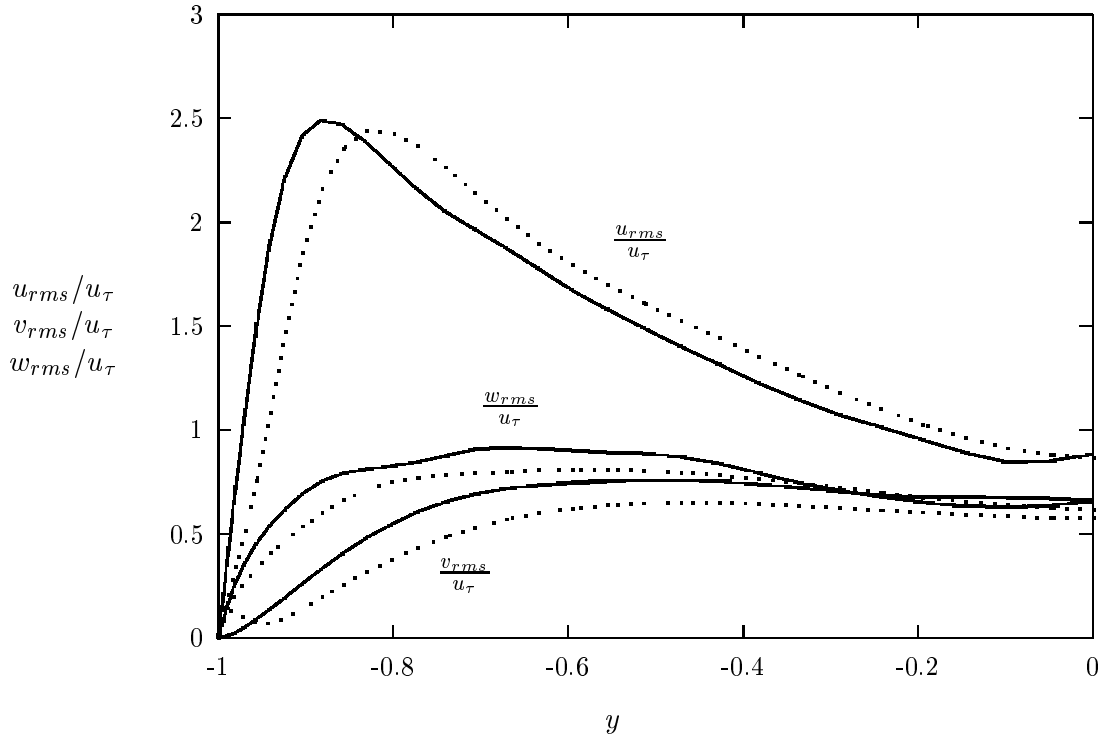
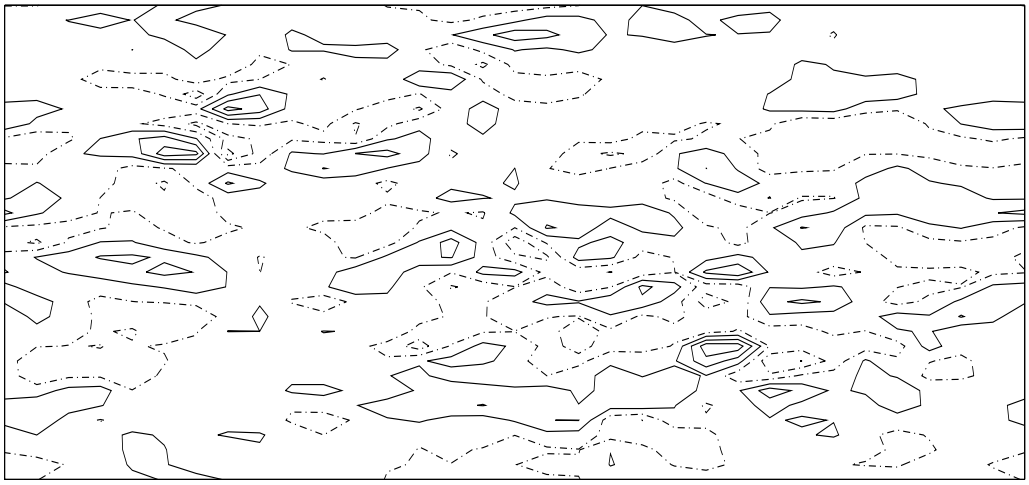


Figure 10: Root-mean-square fluctuations normalized by the wall-shear velocity: ——, no control; ····· , control with 7 fixed weights.

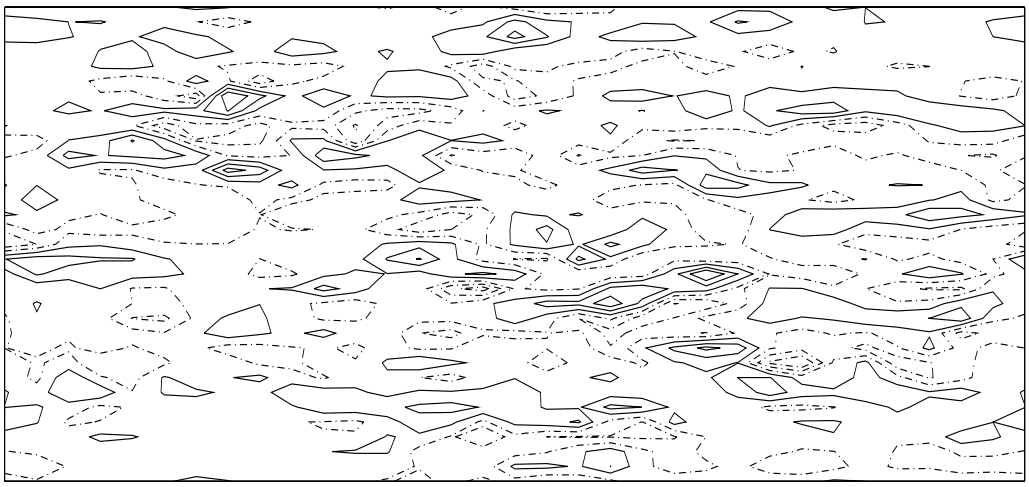
(a)



z

x

(b)



z

x

Figure 11: Contours of the wall actuation: (a) control using $\partial w/\partial y|_w$ with 7 fixed weights; (b) control using information at $y^+ = 10$. The contour level increment is the same for both figures. Negative contours are chain-dotted.