

NEURAL NETWORKS FOR ACTIVE DRAG REDUCTION IN FULLY TURBULENT AIRFLOWS

David Babcock, Rodney Goodman

Department of Electrical Engineering,
California Institute of Technology
Pasadena, CA 91125
babcock@micro.caltech.edu

Changhoon Lee, John Kim

MAE, University of California Los Angeles
Los Angeles, CA 90024

ABSTRACT

This paper presents the application of a neural network controller to the problem of active drag reduction in a *fully turbulent* 3D fluid flow regime. Based on a successful yet infeasible previous active control scheme, we trained a neural network to mimic the control law using only surface spanwise shear stress measurements. We then demonstrate the ability of a neural controller implemented in an adaptive inverse model scheme to maintain a drag-reduced flow in a fully turbulent fluid simulation. By observing the weights of the on-line controller, a simple control law that predicts actuations proportional to the spanwise derivative of the spanwise shear stress is derived. Finally we examine the amount of parameter variation that may be required for a physical implementation of linear and non-linear neural controllers.

INTRODUCTION

In today's competitive air transportation industry, fuel costs are a substantial economic concern. Through drag reduction, airlines can increase fuel efficiency thereby reducing overall costs. It is estimated that reductions of as little as 5% can translate into millions of dollars in annual fuel cost savings.

Neural networks have been used to actively reduce drag in 2D flow simulations [3]. We have successfully applied neural network control techniques to fully turbulent 3D flow simulations. In this paper, we first introduce the problem and an infeasible active control law. We then present a neural network that mimics this control through learning a correlation between surface measurable variables and the desired actuation. We then show a successful on-line adaptive inverse model neural controller. From the resulting weight patterns, we then discuss a simple analytic control law that performs as well as the neural controller yet is computationally more efficient. Finally, we men-

tion some practical hardware issues related to the control schemes.

The cause of drag

Large skin friction drag has recently been linked to organized structures in turbulent flows which play an important role in turbulence transport. The main cause of high drag in turbulent flows is randomly occurring near-wall streamwise vortices, see Figure 1. High local regions of surface shear stress are created when these vortices interact with the viscous layer near the surface. Passive control schemes, such as riblets, attempt to reduce drag by minimizing the surface area of vortex interaction. Such schemes are usually effective only in the particular flow regimes for which they are designed, i.e. where the scale of the turbulent structures matches the physical size of the passive elements.

Active control schemes attempt to reduce drag in turbulent flows by mitigating the strength of the vortices through surface blowing and suction. These control schemes have a wider range of application but require some form of signal processing usually through sensors, control logic, and actuators. Until recently, the small size of these vortices, which decreases as the Reynolds number of the flow increases, has limited physical experimentation. Furthermore, the inherent complexity of the governing Navier-Stokes equations has likewise limited analytical approaches.

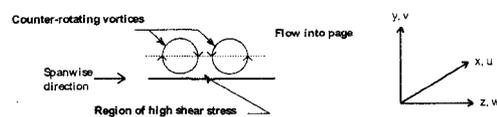


Figure 1: Diagram of the interaction between a vortex pair and a surface. x , y , and z represent the streamwise, normal, and spanwise directions with u , v , and w the respective velocity components.

Development of accurate numerical algorithms have revolutionized the study of

the basic physics of turbulence through computer simulations of turbulent flows. Time-averaged statistics and instantaneous turbulence structures obtained from these simulations agree well with existing experimental results, validating these computer-generated databases [5]. These databases can provide training data for off-line neural network training as well as a platform to implement on-line controllers in fully turbulent 3D flow regimes.

Previous active control work

Other researchers have demonstrated significant drag reduction ($\approx 25\%$) in simulation using active control [1]. The control law used in these experiments involves blowing and suction at the surface according to the normal component of the velocity field sensed in the near-wall region directly above the surface, see Figure 2. The actuation is thus given by:

$$v_{wall} = -v(y^+ = 10) \quad (1)$$

We will refer to equation 1 as Near Wall Control (NWC). The drag reduction results from suppression of the interaction of the streamwise vortices with the surface. Choi, et. al. [1] also discovered that approximately 80% of the drag reduction is due to large actuations (i.e. those with magnitude greater than v_{rms}) which account for about 15% of all actuations.

Clearly this control law requires information from within the flow field ($v(y^+ = 10)$). In real systems it is often difficult and/or impractical to place sensors away from surfaces to measure the fluid velocities associated with the streamwise vortices. A practical version of the control law could be constructed, however, if a correlation exists between the near-wall normal velocity and surface pressure and shear stress measurements.

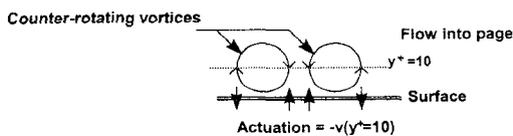


Figure 2: Near-Wall control scheme.

NEURAL NETWORK CONTROL

The above result demonstrates the existence of a successful active control scheme for drag reduction in fully turbulent flow simulations

yet is infeasible for physical implementation. One approach to overcome this limitation is to "learn" a correlation between available surface measurements and the desired actuations using a neural network. The initial training data is obtained from simulations employing NWC, while subsequent data can be obtained on-line during simulations using the neural controller.

Off-line training

Initially we trained a neural network off-line using the spanwise surface shear stress as input and the actuation from the NWC scheme as the desired output. This approach basically attempts to mimic NWC using only surface measurable quantities. The flow simulation was run using NWC with a computational domain resolution of $32 \times 64 \times 32$ points (representing $x, y,$ and z). The training data obtained from this simulation consisted of the 32×32 array of surface spanwise shear stress values ($\partial w / \partial y|_{wall}$) and the corresponding 32×32 array of actuations ($-v(y^+ = 10)$) from 100 consecutive timesteps.

We utilize a shared weight network in order to predict outputs based only on a *local* input area. The network has a single set of weights (template) that is convolved over the entire input space to generate the output values. The template attempts to find spatially invariant correlations between the input data and the desired output. We selected a standard two layer network with hyperbolic tangent hidden units and a linear output unit, see Figure 3.

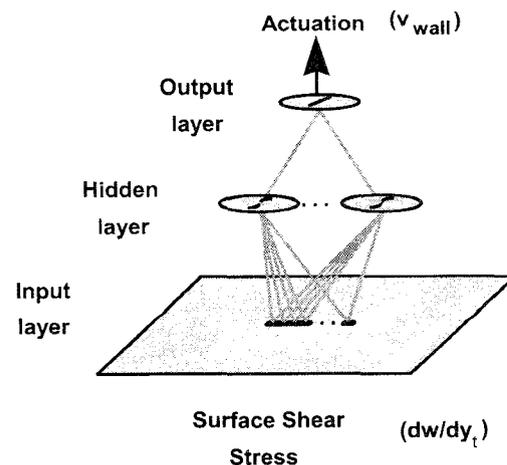


Figure 3: Neural network architecture.

The input template shape can be arbitrary, however based on experimentation we found that adequate performance could be

obtained using only a single row of inputs oriented in the spanwise direction (see Figure 3). Hence the functional form of the neural network is given by:

$$v_{wall} = \sum_{i=1}^{N_h} W_i^1 \tanh \left(\sum_{j=-N_i/2}^{N_i/2} W_{ij}^2 \left(\frac{\partial w}{\partial y} \right)_j - W_{ib}^2 \right) - W_b^1 \quad (2)$$

where N_h is the number of hidden units, N_i is the length of the input template, and W^l are the weights for the l^{th} layer. We then trained several networks with different length input templates and different numbers of hidden units using the scaled conjugate gradient method presented by Moller [8]. In order to emphasize accurate prediction of the large actuations (which as mentioned earlier are responsible for the majority of the drag reduction), we chose to add an exponential prefactor to the standard total sum squared error measure:

$$E = \frac{1}{2} \sum_{\nu} e^{k|\bar{v}_{wall}^{\nu}|} (\bar{v}_{wall}^{\nu} - v_{wall}^{\nu})^2 \quad (3)$$

This error measure exponentially weights the errors contributed by the larger actuations based on the scale parameter k . Empirically we found that $k = 5$ produced the best results.

The final input templates (normalized by the maximum weight in the template) for one hidden unit networks with input template lengths 5 and 7 are shown in Figure 4. These weights are similar to the coefficients of a centered finite difference approximation to the spanwise derivative. Increasing the number of hidden units only marginally improves performance, however increasing template length significantly reduced the final training error.

On-line control

We (admittedly optimistically) fixed the final weights from off-line training and employed the network as a feedforward controller (with no further training) to evaluate the on-line effectiveness of the learned correlation. Figure 5 shows the results from a controller with 7 spanwise inputs and one with an additional 3 inputs in the streamwise direction. In simulations starting from identical initial flow conditions as the NWC simulation used to generate the training data, both controllers achieve roughly 18% drag reduction. This result indicates that a correlation exists between the surface shear stress

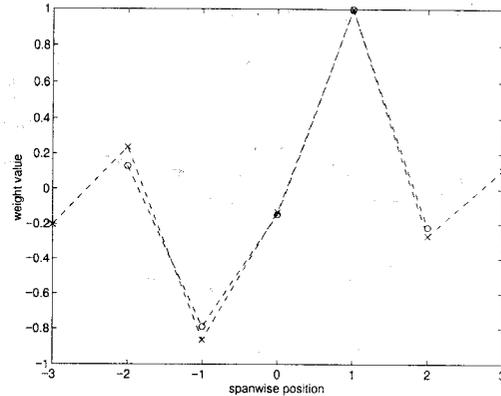


Figure 4: Final template weights. o - length 5 input array; x - length 7 input array. Note the template weights are normalized by W_1

and the desired actuation, though we do not expect the fixed weight network to generalize well for other flow conditions.

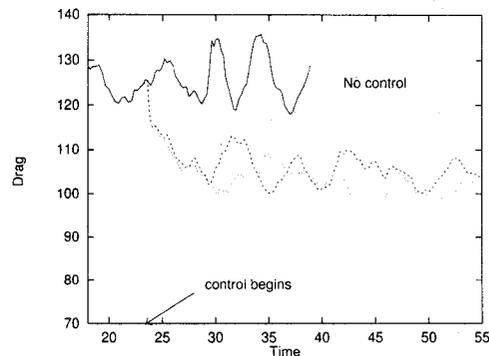


Figure 5: Results using fixed weight off-line trained neural networks as on-line feedforward controllers. - - - 7 fixed weights; ··· 10 fixed weights.

In order to develop a controller that adapts to different flow conditions, some form of on-line training is required. There are numerous implementation schemes for on-line neural control. Adaptive inverse model control is the most direct [10] and is presented pictorially in Figure 6. This configuration models the inverse plant (possibly time-varying) with a neural network and then uses a copy of the model as a feedforward controller. One restriction of this technique is that usually an initial model training phase using random plant inputs and corresponding plant outputs is required before the model can be used as a controller. We avoid this restriction by using the off-line trained network discussed above as an initial inverse model (and controller). We also take advantage of the shared weight structure which gives 1024 (32×32) new data

points at each step in time.

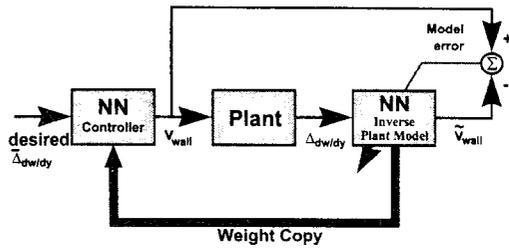


Figure 6: Schematic representation of adaptive inverse model control.

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

$$\left(\frac{\partial \bar{w}}{\partial y}\right)_{t+1} = \eta \left(\frac{\partial w}{\partial y}\right)_t \quad (4)$$

where $0 < \eta < 1$. The output is the predicted actuation necessary to produce this shear stress reduction. Good performance was achieved for $\eta = 0.85$.

The results for different length input templates is shown in Figure 7. The control scheme achieved approximately 20% drag reduction for input templates of length at least 7 spanwise units. We examined the template pattern after each timestep for the length 7 case and observed that the relative magnitudes remained nearly constant (and similar to the off-line pattern), see Figure 8. The absolute magnitudes did vary indicating that gain and bias adaptation are still required. Hence the current on-line implementation is a single hyperbolic tangent hidden unit network with a fixed input template based on the observed pattern. This network contains only four adaptable parameters (a bias and gain for each layer) and has the following functional form:

$$v_{wall} = W_0 \tanh \left(W_2 \left(\frac{\partial^2 \bar{w}}{\partial y \partial z} \right)_N - W_3 \right) - W_1 \quad (5)$$

where

$$\left(\frac{\partial^2 \bar{w}}{\partial y \partial z} \right)_N = \sum_{i=-N/2}^{N/2} \bar{W}_i \left(\frac{\partial w}{\partial y} \right)_i \quad (6)$$

where \bar{W}_i are the final weights from the off-line network. This simplified network achieved similar performance to the fully adaptable network discussed above, see Figure 9.

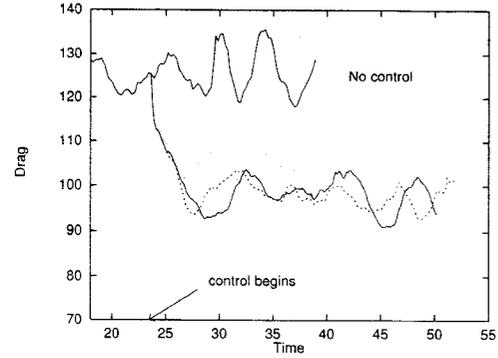


Figure 7: On-line simulation results for fully adaptive neural controllers. \cdots length 5 input template; $---$ length 7 input template; $—$ length 9 input template.

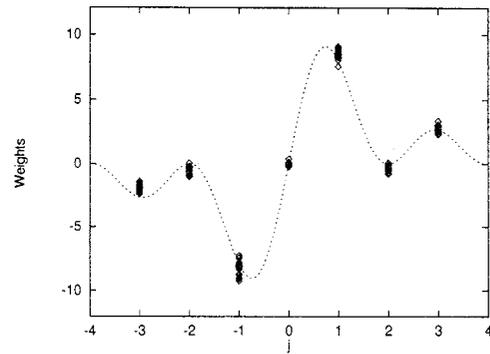


Figure 8: On-line input template weight distributions.

SIMPLE CONTROL LAW

The above results indicate that the streamwise vortices can be suppressed by wall blowing and suction, and that a correlation exists between the surface spanwise shear stress and these actuations. Based on the observation that the (stable) template patterns approximate spanwise differentiation, we used suboptimal control theory to derive the following simple control law for the actuations:

$$\hat{v}_{wall} = C \frac{1}{|k_z|} \frac{\partial^2 \bar{w}}{\partial z \partial y} \Big|_{wall} = C \frac{ik_z}{|k_z|} \frac{\partial \bar{w}}{\partial y} \Big|_{wall} \quad (7)$$

where k_z is the wave number in the spanwise direction, $\frac{\partial \bar{w}}{\partial y} \Big|_{wall}$ is the surface spanwise shear stress, C is a positive scale factor determining the amplitude of the actuation, and (\cdot) denotes a Fourier transformed quantity [6]. Equation 7 produces actuations proportional to the distribution of the *true spanwise derivative* of the spanwise surface shear stress. Taking the inverse Fourier transform of $ik_z/|k_z|$ for finite maximum wave number, equation 7 be-

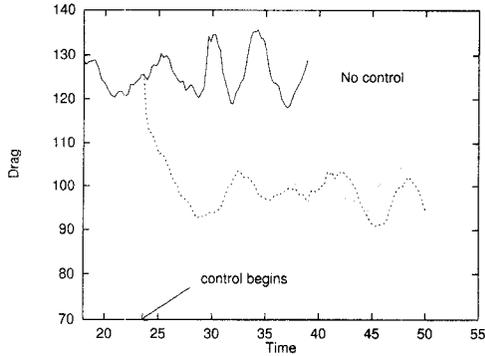


Figure 9: Simulation results for simplified controllers. - - - NN with fixed input template weights; ··· simple spanwise derivative control law.

comes:

$$v_{wall,k} = C \sum_{j=k-N/2}^{k+N/2} W_j \left(\frac{\partial w}{\partial y} \right)_j \quad (8)$$

where

$$W_j = \frac{A(\cos(\pi j) - 1)}{j} \quad (9)$$

N denotes the total number of weights, and k denotes the discretized location on the surface. A plot of equation 9 is shown in Figure 8. We note that W_j for $j = -3, -2, \dots, 3$ correspond well to the input template pattern of the on-line neural controller. Furthermore, this simple control law achieves similar drag reduction to the on-line neural controllers, see Figure 9.

PHYSICAL ISSUES

The ultimate goal of these control schemes is to implement them in a physical system. An active field of research, inspired by biology, is utilizing micromachine technology to build devices capable of operating at the scales necessary to actively control turbulent flows [4, 7, 9]. There is also an effort to integrate these devices into a complete system on a common substrate with analog VLSI control circuitry based on the results from the numerical simulations [2].

One concern with any physical system that employs adaptive schemes is the amount of parameter variation required for different operating regimes. Since the control law given by equation 8 is essentially linear, we tried a single linear neuron in the adaptive inverse model control scheme, i.e. removed the hyperbolic tangent non-linearity.

The performance was similar to the nonlinear network, and the relative weight patterns at each timestep again converged to the approximate derivative. However, the absolute size of the weights varied by up to an order of magnitude which was not observed for the nonlinear network, see Figure 10. This indicates that the saturating behavior of the nonlinearity is not crucial for good performance, but may simplify hardware implementation of the control (particularly considering the hyperbolic tangent function is a basic analog amplifier property).

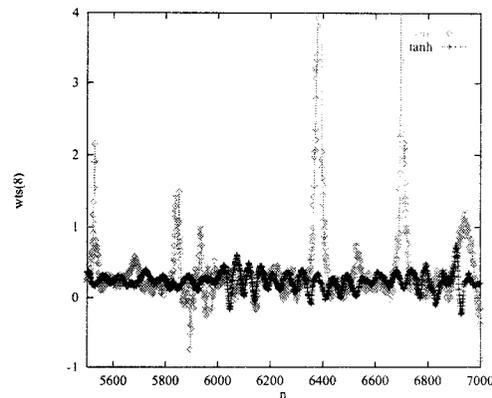


Figure 10: Example on-line weight variations for neural controllers. \diamond linear network; $+$ nonlinear network.

REMARKS

We have presented a neural network that learns a correlation between the surface spanwise shear stress and the desired actuation from an infeasible analytic control law. We have demonstrated the ability of a shared weight network based on the off-line result to achieve 20% drag reduction in an on-line adaptive inverse control scheme. Furthermore, the input template patterns converged to a common pattern which approximates spanwise differentiation. Similar drag reduction was achieved from a network with fixed input template weights and only gain and bias adaptation. From this pattern we derived an equally effective simple control law computing actuations proportional to the spanwise derivative of the spanwise shear stress. New technological advances provide the opportunity to implement such a controller in a real system with the network nonlinearity reducing the necessary dynamic range of the circuit.

ACKNOWLEDGMENTS

This work is 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; and by the California Trade and Commerce Agency, Office of Strategic Technology under grant C94-0165. This work is also supported in part by ARPA/ONR under grant no. N00014-93-1-0990, and by an AFOSR University Research Initiative grant no. F4962093-1-0332. Computer time has been supplied by the San Diego Supercomputer Center and by the NAS program at NASA Ames Research Center.

REFERENCES

- [1] H. Choi, P. Moin, and J. Kim. Turbulent drag reduction: Studies of feedback control and flow over riblets. Technical Report TF-55, Stanford University, 1992.
- [2] B. Gupta, R. Goodman, F. Jiang, T. Tsao, Y.C. Tai, S. Tung, and C.M. Ho. Wafer scale mems and analog vlsi system for active drag reduction. In *8th IEEE International Conference on Innovative Systems in Silicon*, pages 46–52, Austin, 1996.
- [3] S.A. Jacobson and W.C. Reynolds. Active control of boundary layer wall shear stress using self-learning neural networks. In *AIAA Shear Flow Conference*, pages 1–12, Orlando, 1993.
- [4] F. Jiang, Y.C. Tai, J.B. Huang, and C.M. Ho. Polysilicon structures for shear stress sensors. In *Tech. Digest IEEE TENCON'95*, Hong Kong, November 1995.
- [5] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully-developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 1987.
- [6] C. Lee, J. Kim, D. Babcock, and R. Goodman. Application of neural networks to turbulence control for drag reduction. In *Flow Control Workshop*, Cargese, Corsica, France, 1996.
- [7] C. Liu, Y.C. Tai, J.B. Huang, and C.M. Ho. Surface micromachined thermal shear stress sensor. In *ASME Application of Microfabrication to Fluid Mechanics*, pages 9–15, Chicago, 1994.
- [8] M. Moller. *Efficient Training of Feed-Forward Neural Networks*. PhD thesis, Aarhus University, Denmark, 1993.
- [9] T. Tsao, C. Liu, Y.C. Tai, and C.M. Ho. Micromachined magnetic actuator for active fluid control. In *ASME Application of Microfabrication to Fluid Mechanics*, pages 31–38, Chicago, 1994.
- [10] B. Widrow. Adaptive inverse control. In *Second IFAC Workshop on Adaptive Systems in Control and Signal Processing*, pages 1–5, Lund, Sweden, 1986.