

METHODOLOGY TO ACHIEVE ROBUST CLOSED-LOOP TURBULENT FLOW CONTROL USING MACHINE LEARNING

Calin G. Gaina Ghiroaga¹, Matthew R. Singbeil, Chris Morton, Robert J. Martinuzzi

Department of Mechanical and Manufacturing Engineering
University of Calgary

2500 University Drive NW, Calgary, Alberta T2L 1N4, Canada

calin.gainaghiroaga@ucalgary.ca, mrsingbe@ucalgary.ca, chris.morton@ucalgary.ca, rmartinu@ucalgary.ca

ABSTRACT

A methodology to achieve robust closed-loop feedback control of a turbulent flow using machine learning is outlined. The chosen candidate system is a square cross-sectional cylinder with two moving surface actuators embedded in the windward face at the leading corners. A Long Short-Term Memory (LSTM) Neural Network is trained using motor actuation and pressure sensor data to forecast future system states. This LSTM model is then implemented with Model Predictive Control (MPC) in order to achieve closed-loop flow control. The derived controller performance was tested experimentally using three objective functions: recovery of mean-base pressure set-point from perturbation, and drag or wake fluctuation intensity optimizations. An adaptive learning algorithm, which adjusts the model to new Reynolds number (Re) conditions without user intervention, is implemented to extend controller performance and achieve robust control.

INTRODUCTION

Control of bluff body flows is an important research area impacting many industry applications such as efficiency of energy harvesting systems, stability of high-lift devices or drag reduction of high-speed ground vehicles (Choi *et al.*, 2008). Achieving robust control of fluid flows is challenging as the response to an actuation is non-linear, thus linear control methods are generally not successful. Additionally, the high dimensionality of the Navier-Stokes equations implies a large parameter space and thus real-time optimization is difficult. Finally, turbulence increases complexity as fluctuations are stochastic which effectively is equivalent to reducing signal quality, thus making model identification more difficult.

The use of machine learning in closed-loop flow control is appealing as it is a strategy used to identify patterns and relations within large stochastic datasets. Neural Networks (NN) are machine learning architectures that mimic the interconnected nature of neurons in the brain and are capable of forming input and output relationships through trial and error. Recurrent Neural Networks (RNN) are a type of NN that specialize at identifying and correlating patterns through time by recurrently using earlier outputs as inputs for subsequent computations. MPC is a control method that uses a predictive model of the system being controlled to optimize future actuations.

The predictive model is generally a dynamical system of non-linear ordinary differential equations which need to be numerically solved to achieve predictions of future states. While

MPC is able to achieve control, it can be a computationally expensive procedure. In particular, MPC must perform an iterative online optimization which involves numerically solving the model for several candidate actuations in series to determine the optimal actuation based on a cost function. This process is then repeated until the prediction horizon is achieved, which is the number of time steps the controller is set to predict.

An alternative approach can be to incorporate LSTM networks, a type of RNN whose architecture is more suitable for long term dependencies. LSTMs do not require numerical solving, can evaluate all candidate actuations in parallel and can forecast several future states, thus removing all areas of iteration within the MPC framework. In addition to reducing the computational complexity of MPC, LSTMs are a good candidate predictive models as they excel at forecasting time evolution data from smaller training sets when compared to traditional techniques (Siarni-Namini *et al.*, 2018). Therefore, by using a time series of past system states and actuations as an input, future system states can be predicted using the LSTM. Moreover, the LSTM is capable of learning dynamics associated with new Re conditions when they arise, without user intervention, allowing for the model to adapt and achieve robust control over a Re range.

Studies that have achieved flow control are usually restricted to a narrow range of Re , and application to turbulent flows have had limited success. For example, Bieker *et al.* (2019) applied a RNN in conjunction with MPC to a computational model of a fluidic pinball system which was able to achieve closed-loop control authority at low $Re = 100 - 200$, with performance lowering as Re increased. Additionally, the system controlled was laminar and computational. This novel approach aims to achieve robust real-time experimental control of turbulent fluid systems by reducing optimization time and using high performing predictive model architectures.

In this work, the LSTM-MPC approach is implemented to control the turbulent wake of a square cross-sectional cylinder. The actuators are rotating cylinders at the leading edges which are modulated to achieve forcing frequencies which excite the separated shear layer. These shear layer excitations can manipulate the circulation generation and transport to exploit flow instabilities and alter the wake dynamics. Singbeil *et al.* (2021) showed that frequency-based actuation achieves authority of the flow at lower surface velocity to free-stream ratios ($v_{surf}/U_\infty = 0.2$ to 0.4) compared to no forcing frequency ($v_{surf}/U_\infty = 3$) (Munshi *et al.*, 1997). A model relating cylinder RPM and obstacle surface pressure will be deter-

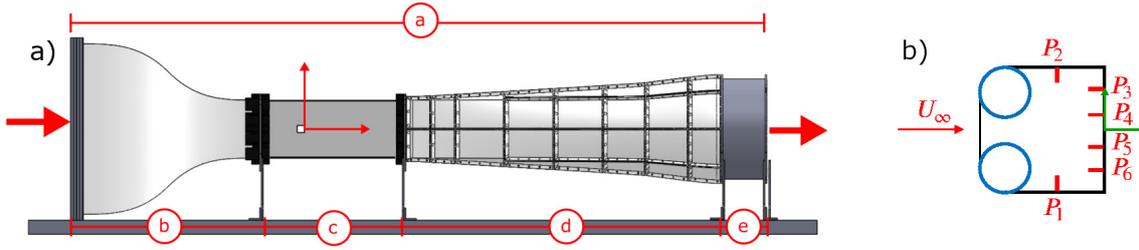


Figure 1. a) Side View of Wind Tunnel Facility. b) Obstacle with surface mounted pressure taps (P1 through P6).

mined experimentally and used to achieve flow control using MPC.

METHODOLOGY

Experimental Facility

The experiments were performed in a small-scale suction type wind tunnel shown schematically in Fig. 1a. The wind tunnel is $a = 2510$ mm in length consisting of a square cross sectional inlet ($b = 690$ mm), interchangeable test-section ($c = 510$ mm), square to the circular cross sectional expansion ($d = 1150$ mm), and fan shroud ($e = 160$ mm). Air is drawn through a honeycomb and four conditioning grids ($d_{screen} = 0.22\text{mm}$, $\Delta_{screen} = 1.18\text{mm}$) at the inlet before undergoing a 9:1 area contraction to the square test section with a wetted cross-sectional area of $203.2\text{ mm} \times 203.2\text{ mm}$.

The obstacle is a square cross-section cylinder 3D printed from PLA (Polylactic Acid) with a characteristic length $D = 25.4\text{mm}$ and thickness $T = 25.4\text{mm}$ (Fig 1b). The obstacle was printed with the layers being oriented in the freestream direction, and the surface was sanded to minimize variation in surface roughness. Installed 127.0 mm downstream of the test section inlet, the obstacle spans the test section for an aspect ratio of 8:1 and blockage of 12.5%.

The actuators are two $d_{act} = 10$ mm diameter rotating cylinders, embedded as shown in Fig. 1b, at the leading edges of the square cylinder. Each cylinder was printed from PLA and a tight tolerance tool steel shaft was inserted through the center of the cylinders. Each cylinder was sanded to have surface roughness similar to the obstacle. The cylinders have nine span wise sinusoidal grooves that have an amplitude of 0.5mm. The grooves create a forcing frequency equal to nine times the rate of rotation. One end of each cylinder was connected to the drive shaft of a brushless servo-motor mounted external to the test section. The other end of the steel shaft is supported externally from the test-section using radially loaded ball bearings. Additionally, the cylinders experience a whirling motion which introduces a forcing frequency and harmonics of the rate of rotation ($f_w = \text{RPM}/60$) (Baek & Sung, 2000). The whirling arises due to the center of inertia not coinciding with the axis of rotation. The whirling is beneficial as it introduces new forcing frequencies that can be exploited.

Six differential pressure transducers monitor surface pressure at locations P_1 to P_6 indicated in Fig. 1b. On the top/bottom surfaces, transducers P_1 and P_2 are 1-inch water column (*All Sensors D-4V* - Sensitivity = 124 Pa/V) and on the leeward face P_3 to P_6 are 5-inch (*All Sensors Dx-4v Mini* - Sensitivity = 622 Pa/V) transducers. An Advantech PCIE-1812 DAQ was used to measure all the pressure signals, RPM signals and output RPM signals.

MPC Methodology

MPC was used to achieve closed-loop feedback control of the system. Closed-loop control works by measuring the system output or current system state and feeding it back into the controller, allowing the controller to improve future actuations. A flow diagram of the entire closed loop control system and the MPC controller is shown in Fig. 4.

Traditional MPC uses an optimizer which suggests several actuation strategies for the next time step based on the current actuation and the actuation constraints. The suggested actuations and previous system states are input into a model, generally a system of ODEs, which gets numerically solved for each potential actuation. The entire process gets repeated until the prediction horizon, N , has been reached. By using a pre-trained LSTM as the predictive system model, the traditional MPC framework can be altered to reduce computational complexity since all areas of iteration are eliminated. For this work, as an input to the LSTM, the optimizer suggests a library of actuations, created a-priori, containing all possible actuations discretized between minimum and maximum actuations. The LSTM then computes (in parallel) all of the candidate actuations inside this library and outputs the predicted system state for the entire prediction horizon. Afterwards, the optimizer, through the use of a cost function, locks in an actuation. The cost function is a relationship comparing the systems state predictions to a desired state which is optimized in order to choose the optimal future actuation.

The quantity of interest which will be controlled will be the leeward face base pressure on the obstacle ($Cp_{base} = (Cp_3 + Cp_4 + Cp_5 + Cp_6)/4$). The reason for this is to influence important characteristics that are linked to the leeward face base pressure such as drag and fluctuating loads on the obstacle. Thus, the output of the system will be Cp_{base} , and it will be controlled with the two rotating cylinders at the leading edges of the obstacle. In order to use MPC to control this quantity, a model that relates the system actuation, cylinder RPM, and the system state, Cp_{base} , must be developed. Cp_{base} will be treated as the system state that will be measured and fed back into the controller, where it will be used as an input to an LSTM, and the output will be future values of Cp_{base} .

LSTM Methodology

An LSTM was used to create a model of the system which will be used inside the MPC controller. The LSTM was chosen since it has the capability of making use of history when given an input sequence to better inform the prediction. A diagram of an LSTM showing how inputs, outputs are handled is shown in Fig. 5. The diagram shows the flow of information through time (horizontally) and how information is remembered and forgotten through forget and input gates. The

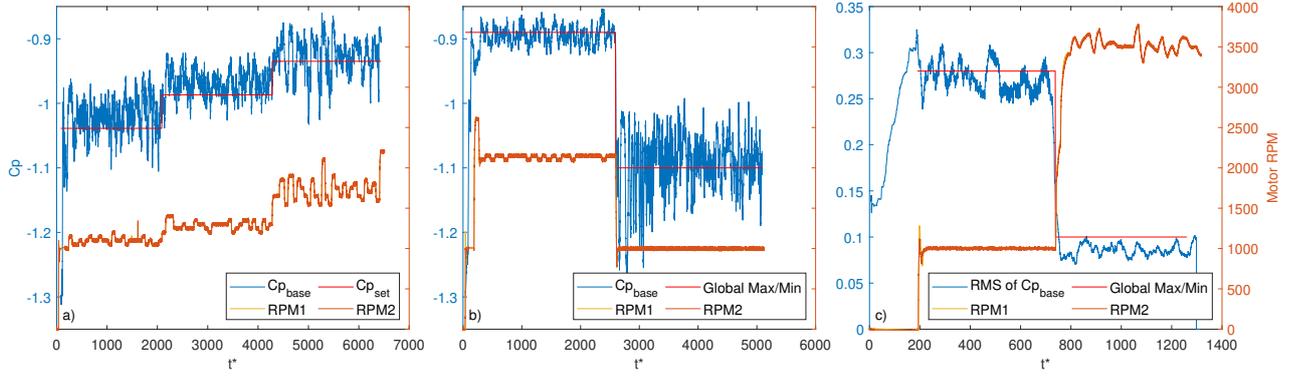


Figure 2. Motor Actuation Speed (RPM1 and RPM2) and Pressure Response for a) Recovery of mean-base pressure set-point b) Optimization of Drag through mean Cp_{base} c) Optimization of wake fluctuation through RMS of Cp_{base} . Cp_{base} is defined as $(Cp_3 + Cp_4 + Cp_5 + Cp_6)/4$. t^* is the time non-dimensionalized by the shedding period. The RMS is computed as a windowed average of 17 shedding cycles.

goal of the LSTM model is to accurately forecast future system states, $Cp_{base}(t)$, in order for the controller to better optimize the next actuation (RPM). The LSTM model predicts future system states, $Cp_{base}(t)$, which are represented as a vector array of size $1 \times N$, where N is the prediction horizon. The input to the LSTM is an array of size $3 \times M$, where M is the input sequence length. The three rows of the input array are the past system states ($Cp_{base}(t)$), the past actuation (RPM), and the future actuation (Candidate RPM). Past system states and past actuation were decided upon as inputs as they provide the necessary history and feedback to predict future system states in a closed loop control system. The future actuation input is used in order for the optimizer to suggest future actuations, and thus allow the controller optimize the future actuation in order to achieve the control goal.

The LSTM model is trained offline using time series of pressure and actuation data. To begin, input and output data are collected, where the system is randomly actuated between the minimum and maximum $RPM = [1000, 3800]$ for a set time interval of 10 minutes, changing RPM every 2 seconds, at a constant $Re = 12500$. Pressure and RPM data are collected at a sampling frequency of 3000Hz. 50% of the data obtained from the data collection are used for training and 50% are saved for model validation. This split was empirically tested to be suitable and was not optimized. The training data are split into inputs ($3 \times M$) and outputs ($1 \times N$). The choice of N and M is important as the LSTM needs sufficient past information to model the characteristic time-scale of the flow. The relevant time scale is that of the vortex shedding cycle-to-cycle variations and it is found that this requires at least two shedding periods. Each input and output are normalized to be within $[-1, 1]$, by their minimum and maximum values.

Adaptive Learning

A strategy is developed so that the model trained for a given Re can adapt autonomously to flows at different Re conditions. The algorithm is split into two main functions; a continuous adaptation, and a transfer learning adaptation. The purpose of the continuous adaptation is to improve controller performance for small Re changes, and for when the model has not perfectly captured the dynamics. In order to improve the controller performance in these scenarios, the model is continually adapted and trained using feedback data, as seen in Fig. 4, before the feedback data enters the controller. This quickly corrects any discrepancies between model prediction

and actual pressure by correcting model errors at the current conditions.

The purpose of transfer learning adaptation is to improve controller performance for large Re changes. The algorithm begins by collecting a small set of data at new conditions which span the entire actuation space. The model is re-trained using the pre-trained data as a starting point and adapting to the new conditions. Through this transfer learning, a higher accuracy can be obtained in a shorter amount of time if the data set used to pre-train the model exhibit similar system dynamics to the new data set (Torrey & Shavlik, 2010).

RESULTS

The performance of the closed-loop control was measured by subjecting the controller to: (i) Recovery of mean-base pressure set-point from perturbation, (ii) drag or wake fluctuation intensity optimization, and (iii) robustness test at small and large Re changes, and a difficult control scenario.

(i) Recovery of Mean-base Pressure Set-point from Perturbation

This experiment tests the controller ability to recover the system state to a specific value when started at a random condition, and with naturally occurring perturbations which come from the turbulent flow development. Additionally, it tests whether the LSTM is capable of accurately predicting the system dynamics in real time. To confirm these two capabilities, the controller optimizes the cost function $(\min_{u \in R^n} \|Cp_{future}(t) - Cp_{set}(t)\|)$. Figure 2a shows how the controller is able to keep the short term average of the base pressure $Cp_{base}(t)$ (blue) to the prescribed set-point (red). The optimized cylinder actuation is shown through RPM1 (yellow) and RPM2 (orange). This confirms that the controller is capable of recovering the mean-base pressure to a set-point with constant perturbations and that the LSTM accurately predicts system dynamics real time.

(ii) Drag and Wake Fluctuation Intensity Optimization

This experiment attempts to determine the global optima of the drag or wake fluctuation intensities through $Cp_{base}(t)$. For drag intensity, the mean of $Cp_{base}(t)$ is optimized through the cost function: $(\min_{u \in R^n} \text{ OR } \max_{u \in R^n} \|Cp_{future}(t) - Cp_{current}(t)\|)$. Similarly, for wake fluctuation intensity, the RMS of $Cp_{base}(t)$ is optimized through the cost function:

$$(\min_{u \in R^n} \text{ OR } \max_{u \in R^n} \|CpRMS_{future}(t) - CpRMS_{current}(t)\|).$$

This problem is traditionally difficult since iteration is necessary to find global optima, which is computationally expensive (Choi *et al.*, 2008). Sub-optimal control, which stops iteration early, is faster and less computationally expensive but can get trapped in local minima and relies heavily on a cost function. Iteration and sub-optimal solutions are eliminated by the LSTM since the prediction horizon is computed in one step and the entire actuation space can be evaluated in one input. By evaluating the entire actuation space, the cost function is easily optimized to determine the global optima.

The test requires the controller to maximize and then minimize the mean/RMS of $Cp_{base}(t)$. Figure 2b shows the max/min ($t^* = 0 - 2500/2500 - 5000$) mean of $Cp_{base}(t)$. Next, Figure 2c shows the max/min ($t^* = 200 - 750/750 - 1300$) RMS of $Cp_{base}(t)$. These two figures show the controller's capability to find the global max/min for both the mean and RMS of $Cp_{base}(t)$. Figure 6 shows the results of the grid search, confirming the locations of the optima and proving that the global optima were found and accurately captured in the model.

(iii) Control Robustness

One important challenge for robust control is the sensitivity of the system dynamics to changes in Re . The challenge is resolved by adapting the LSTM to unseen conditions. Three tests are done to show how effective the adaptive learning algorithm is. The first test is to use continuous adaptation in a region in the parameter space that is difficult to control, such as high gradients of pressure with respect to RPM. Discerning small pressure differences in large pressure gradient regions is difficult in real experimental systems due to the compounding effects of sensor noise and model inaccuracy. Fig. 7 shows how the controller performs when asked to discern small pressure differences without adaptation on the left and with adaptation on the right. The adaptive learning algorithm improves controller accuracy in challenging situations as the controller is better able to regulate the pressure response to the control goal.

The second test is to use continuous adaptation when making small Re changes to see if the controller can recover mean-base pressure set-point to a constant value. Without adaptation, this is difficult as dynamics change when Re changes, and thus the LSTM model will be inaccurate. Figure 8 shows how the controller quickly reacts to a 10% change in Re and recovers the pressure back to the desired set-point, at a different Re . With a 10% change in Re , a 100% change in RPM was needed to maintain the same pressure, which shows the system sensitivity to Re changes and the necessity for continuous adaptation in real-world systems when Re is not constant.

Finally, a recovery of mean-base pressure set-point test was run at $Re = 22250$, a significantly different Re compared to $Re = 12500$ which the model was previously trained on, to see how transfer adaptive learning performs. This procedure includes collecting a small set of data and using it to retrain the model on the new conditions. The set-point test is run before and after re-training to show the difference in performance. Figure 3 shows how the model initially performs poorly at the new Re when the controller is asked to regulate $Cp_{base}(t)$. After adaptive learning, the controller performs well at regulating the pressure to the desired set-point. This shows that the model has indeed learned the new dynamics associated with the unseen conditions. The time to train depends on the integral scales of the flow, the size of the actuation space, and hardware implementation. In this implementation, with an un-

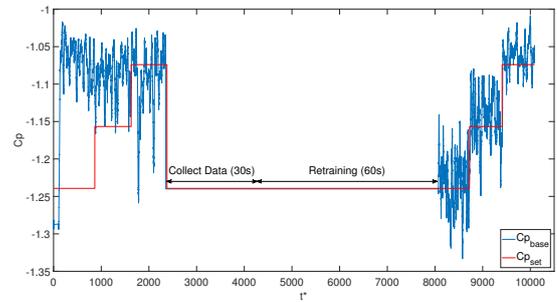


Figure 3. Closed-Loop Controller Adaptive Learning Test. Cp_{base} is defined as $(Cp_3 + Cp_4 + Cp_5 + Cp_6)/4$. t^* is the time non-dimensionalized by the shedding period.

actuated vortex shedding frequency of 60Hz, the model learns the new dynamics quickly (60s) based on its experience learning from the training set at a different Re . If trained without prior learning, it would need at least one order of magnitude larger training time.

CONCLUSIONS

A methodology has been presented for achieving closed-loop turbulent flow control using Machine Learning. MPC optimization time was improved through the use of LSTMs, by eliminating all sources of iteration in traditional MPC. Recovery of mean-base pressure set-point from perturbation, and drag or wake fluctuation intensity optimizations were performed. In addition, an adaptive learning algorithm was introduced to make the control robust to changing Re and increase accuracy in difficult to control scenarios. The novelty of this work is the development of a methodology which achieves robust closed-loop feedback control of an experimental high Re turbulent fluid system. This can help advance the field towards the control of complex systems in industry.

Future challenges to be addressed include increasing actuation degrees of freedom, and increasing MPC cost function complexity by including actuator power.

REFERENCES

- Baek, S. & Sung, H. J. 2000 Quasi-periodicity in the wake of a rotationally oscillating cylinder. *Journal of Fluid Mechanics* **408**, 275–300.
- Bieker, K., Peitz, S., Brunton, S. L., Kutz, J. N. & Dellnitz, M. 2019 Deep model predictive control with online learning for complex physical systems. *arXiv preprint arXiv:1905.10094*.
- Choi, H., Jeon, W. & Kim, J. 2008 Control of flow over a bluff body. *Annu. Rev. Fluid Mech.* **40**, 113–139.
- Munshi, S. R., Modi, V. J. & Yokomizo, T. 1997 Aerodynamics and dynamics of rectangular prisms with momentum injection. *Journal of Fluids and Structures* **11** (8), 873–892.
- Siami-Namini, S., Tavakoli, N. & Namin, A. S. 2018 A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401. IEEE.
- Singbeil, M., Ghiroaga, C., Morton, C. & Martinuzzi, R. 2021 Moving surface actuation, effects of frequency-based shear layer excitation on the response of a bluff body wake. In *14th International Symposium on Particle Image Velocimetry*, vol. 1, No. 1.
- Torrey, L. & Shavlik, J. 2010 Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264. IGI global.

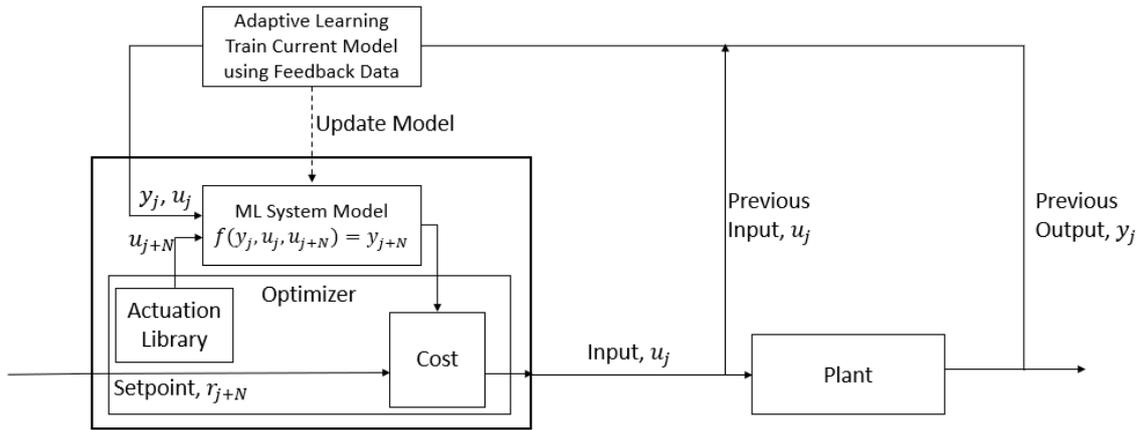


Figure 4. MPC Controller Diagram. The main feedback control loop works by measuring previous system states and actuations, and feeding them back into the controller, where the controller optimizes future actuations. The adaptive learning loop uses feedback data to update the model on current conditions to improve robustness and accuracy.

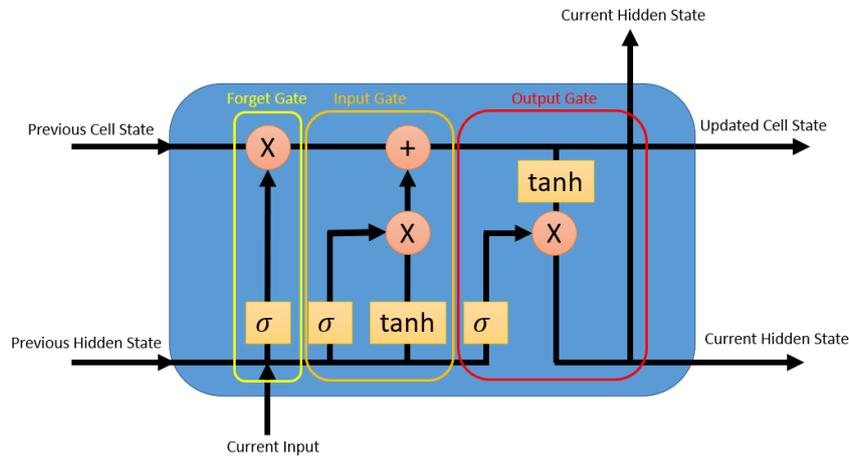


Figure 5. LSTM Architecture. The LSTM uses forget, input and output gates to control the flow of information.

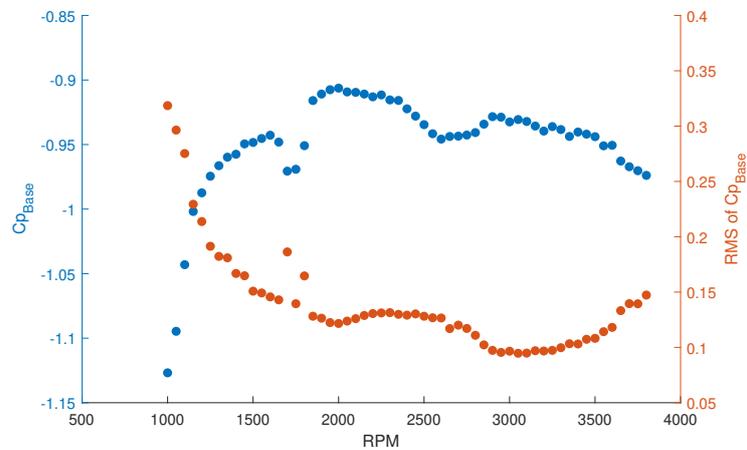


Figure 6. Grid Search Results for Mean and RMS Base Pressure, $Cp_{base} = (Cp_3 + Cp_4 + Cp_5 + Cp_6)/4$.

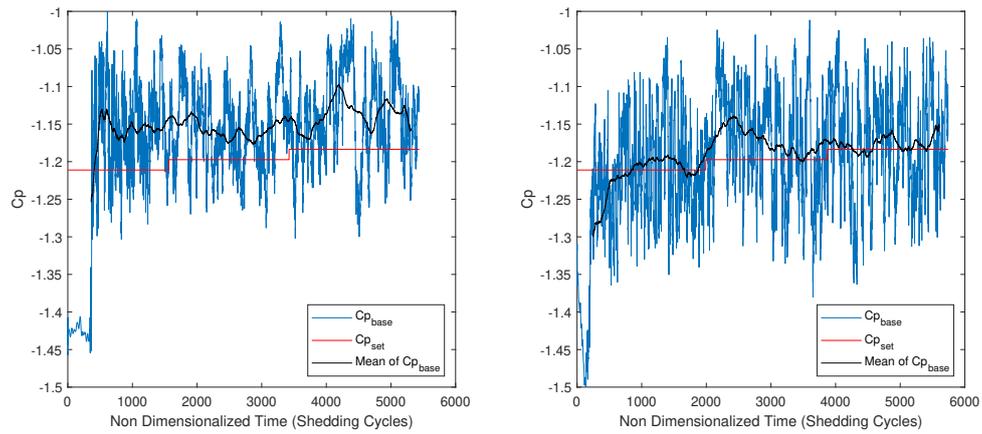


Figure 7. Pressure Response of Setpoint Test in High Gradient Location without (left) and with (right) Adaptive Learning. Cp_{base} is defined as $(Cp_3 + Cp_4 + Cp_5 + Cp_6)/4$. t^* is the time non-dimensionalized by the shedding period.

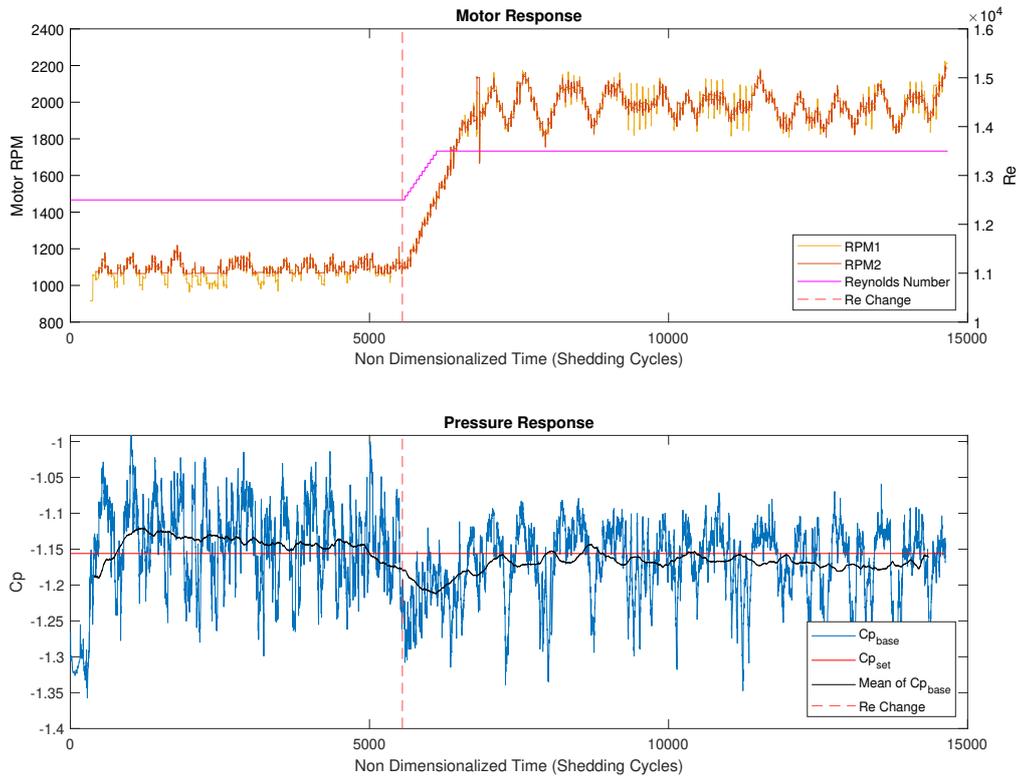


Figure 8. Motor Actuation Speed and Re (top) and Pressure Response (bottom) of Small Re Change Test with Adaptive Learning. Cp_{base} is defined as $(Cp_3 + Cp_4 + Cp_5 + Cp_6)/4$. t^* is the time non-dimensionalized by the shedding period.