

Figure 1: Control policy for the training data. The control policy comes from a proportional controller measuring the y-velocity at a point in the wake

Goal

The goal of this work is to suppress vortex shedding of a circular cylinder using model predictive control, with the model coming from Dynamic Mode Decomposition with control (DMDc) and the control coming from the rotation of the cylinder.

Broadly speaking, DMDc uses singular value decomposition to create a low-dimensional linear model for the dynamics of the system of the form

$$\begin{aligned}\tilde{x} &= U^T x \\ \tilde{x}_{t+1} &= A\tilde{x}_t + Bu_t\end{aligned}$$

where the number of state variables, x is much larger than the number of state variables \tilde{x} .

I implemented DMDc in julia at the following repository (which includes a patch for running MPC with DMDc in PyFR). Below is an update on what is working with DMDc and what is currently not working.

What works

I've have shown that DMDc works for suppressing vortex shedding in a very narrow set of circumstances (see fig 2). Namely, when the dynamics and control matrices (A and B) are trained from snapshots taken from the proportional control of the same system (see figure 1 for the control input of the data that A and B were trained from). In this case, the DMDc + MPC algorithm produces essentially the same control policy and successfully suppresses the vortex shedding.

This success, is only a modest one, however, because a working control policy was used to create the DMDc model. Unfortunately, as will be shown in the next section, the creation of a DMDc model that successfully works to suppress vortex shedding is difficult and fails in many other cases that I have tried.

What Doesn't work

When A and B are found from training data that deviates from the proportional control data, MPC doesn't seem to work at all (see figure 3 for an example). I have tracked the failure back to the fact that the predictive capabilities of DMDc seem limited to the training data that produced to the DMDc model

In order to evaluate the predictive capabilities of DMDc, I did the following:

- From a sample PyFR with MPC run, I copied the dynamics matrices, A and B , the projection matrix U , and all of the simulation data from the run.

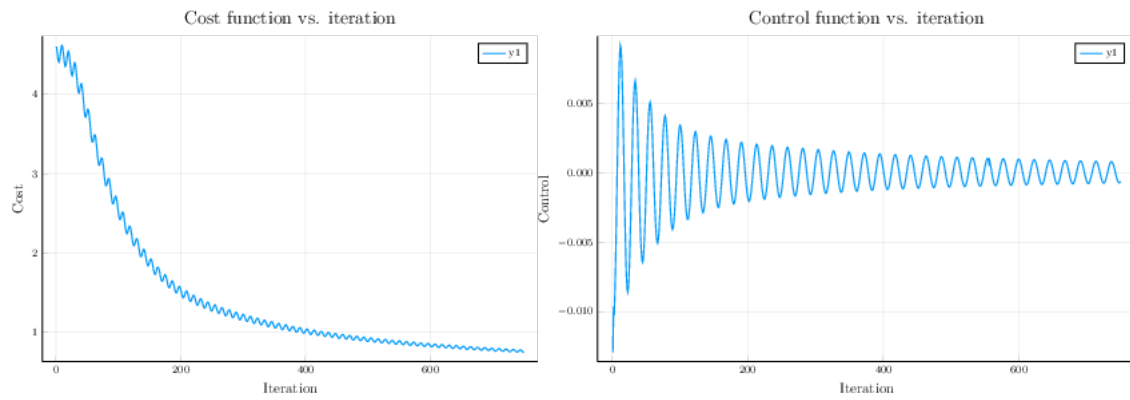


Figure 2: Control policy from DMDc dynamics with MPC controller.

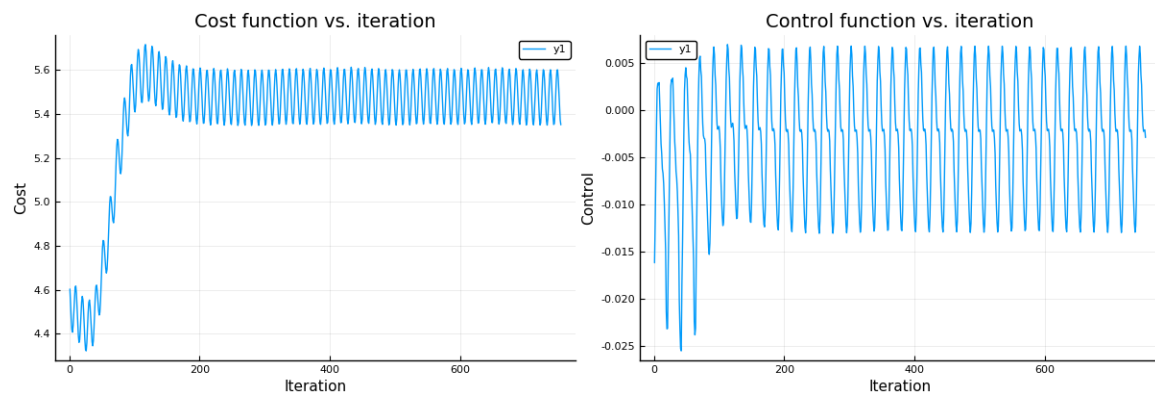


Figure 3: Example of DMDc + MPC failing to suppress vortex shedding. This failure is due to a failure in predictive capabilities of the model

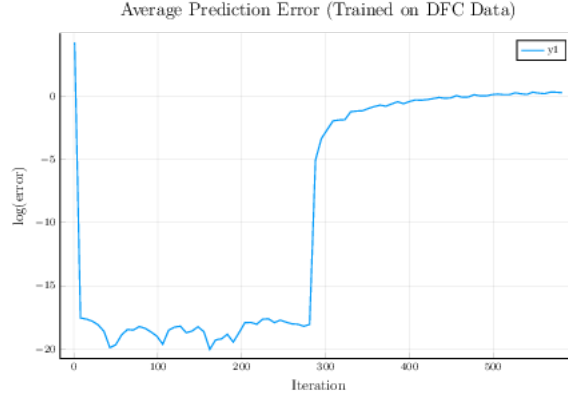


Figure 4: The prediction error of the DMDc model trained from the proportional control data, predicting the proportional control data

- For each snapshot of the simulation data, I predicted the following 16 timesteps using the DMDc model. If the t^{th} frame of the simulation is flattened into a state vector x_t , and the control input is u_t , then we predict via

$$\begin{aligned}\tilde{x}_t &= U^T x_t \\ \tilde{x}_{t+1}^{\text{pred}} &= A\tilde{x}_t + Bu_t \\ x_{t+1}^{\text{pred}} &= U\tilde{x}_{t+1}^{\text{pred}}\end{aligned}$$

- Then at each predicted timestep, the error in the prediction can be found via the norm $\|x_t^{\text{pred}} - x_t\|$, where x_t is the exact state, found from the simulation.

In the following examples, we show how well certain DMDc models predict the simulation dynamics of vortex suppression using proportional control. The dynamics of this process need to be accurately predicted if a similar control policy is going to be inferred with MPC.

First, consider the DMDc model that was trained on the first 300 iterations of the proportional control data (figure 4). In this case, we can see that the predictive capabilities of the model are excellent (to machine precision of exact) over the training period, and are significantly larger (although still remain small) after the training period has ended. This is the model that produced the working control policy.

Then consider a model that was trained off of the data whose control input is shown in figure ???. The predictive capability of this model on the proportional control data is order of magnitudes worse than the previous DMDc model as shown in figure 6. Correspondingly, this model failed to suppress vortex shedding.

Things I've tried

- Used many different types of input training data and checked the predictive power on many other sets of data
- tried online computation of the A and B matrices, for each timestep, or every interval of 20 timesteps. Due to the sensitivity of DMDs capability of predicting future dynamics, the constant recomputing of these matrices seems to cause the algorithm to become unstable and fail

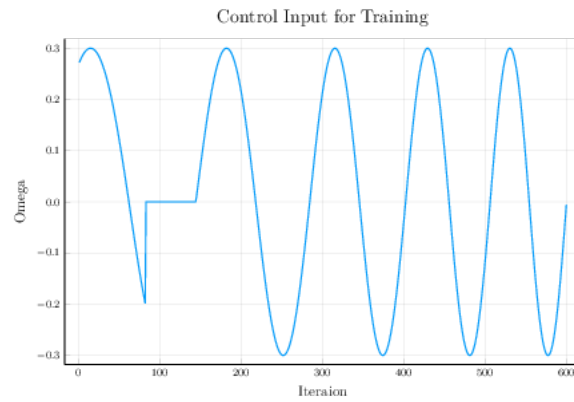


Figure 5: The control input for the dataset that trained the second model. The result was a model that did not accurately predict the proportional control data.

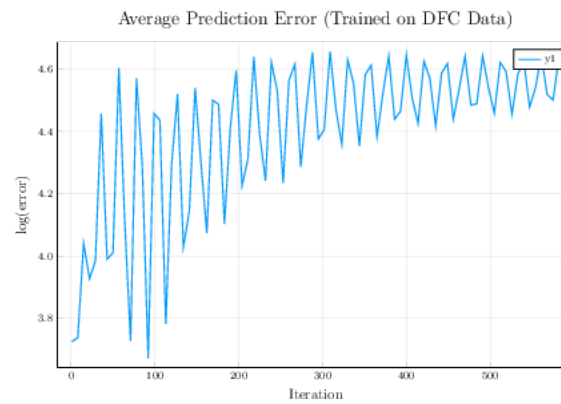


Figure 6: The prediction error of the DMDc model trained from a deep flow control data set, predicting the proportional control data