Modern Control Theory MPC Project

Suraj Y
CH11B063

Shararaghawardhan B
ME11B121

Process Description

The MPC framework is applied to a **Fluidized Catalytic Cracking(FCC)**, which is used for the production of light hydrocarbons from a heavy feed stock. The process is simulated in MATLAB with the following variables:

Inputs

There are two inputs - dF_a and dF_{rc}. They are deviation variables, where the deviation is from their respective nominal values.

Table 1: Input Description

Variable	Desctiption	Nominal Value
F_a	Mass flow rate of air to the generator (kg/s)	25.7267
F_{rc}	Mass flow rate of regenerated catalyst (kg/s)	283.7317

Outputs

Three outputs are being measured in the process.

Table 2: Output Description

Variable	Desctiption	Nominal Value	Restriction
O_d	Oxygen mole fraction in regenerator dense bed	0.00472	Low
C_{rc}	Weight-fraction of coke deposited on catalyst (kg/kg)	0.0069	Low
T_{rg}	Temperature of Catalyst in Regenerator dense bed (K)	972.1	920 - 1040 K

States

Five states are being generated from the data by the **n4sid** routine. These are not unique and may not have any physical meaning. They are strictly used for the purpose of state estimation and MPC, and are not be interpreted.

Files Used

The zip file submitted has the following code files

- *mpc_template.m*: The template to be used for the MPC application.
- *mpc_main.m*: Functional Block which performs the MPC operations
- *mpc_block.m*: Functional Block which performs the MPC optimization
- *kalman_filter.m*: Functional Block which performs the Kalman Filter Operations
- *costFunction.m*: Cost Function for the MPC block
- *coefEst.m*: Function used to generate the augmented matricies
- *inital_filter_block.m*: Functional Block which performs the k^{th} state estimation

The zip file also has the following '.mat' files:

- *results.mat*: Data pertaining to the PRBS input
- *sine.mat*: Data pertaining to the sinusoidal input
- *rbs.mat*: Data pertaining to the RBS input
- *rgs.mat*: Data pertaining to the RGS input

MPC starting from initial state

We are fixing the Prediction Horizon to be 100. We have used this intial problem to get more insights into the control horizon and hence tune it for the next set of problems.

Observations

We have tested the system for M = 5,15,20.

1. If M is small(5), both the **initial repsonse is better and the optimization process is faster** as there are fewer variables. However optimal control isn't achieved after stabilization.
2. If M is large(50), **the initial response is slightly worse off**. **The optimization process is slow and expensive** as there are more variables to control. However the output error is highly reduces after stabilization and hence better control is achieved.
3. As a trade-off between operation expenses and the final control, we have chosen **M = 20** for the rest of the operations.

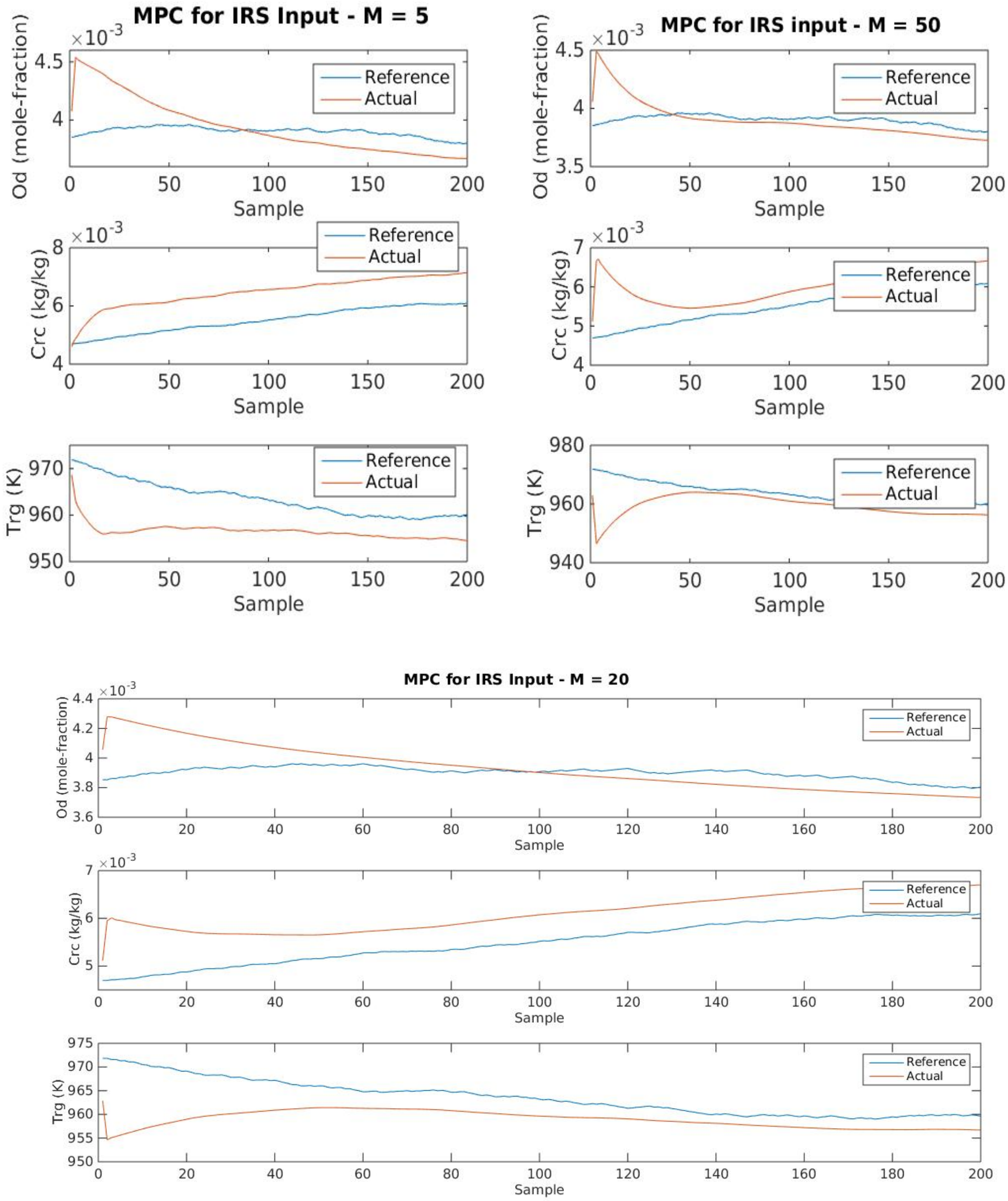
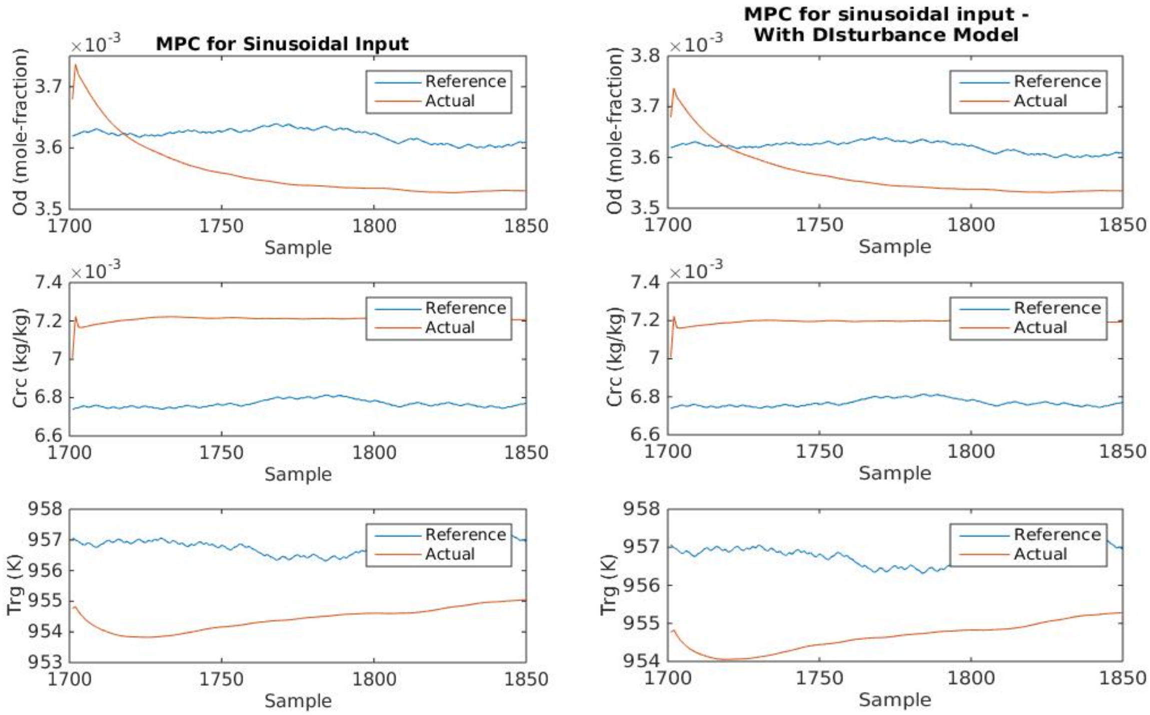


Figure 1: MPC Estimates for the IRS input

MPC Starting from k^{th} instant

Rather than using an initial guess state for the MPC problem, we estimate the state upto a certain instant (first 1700 observations) and then perform MPC for the remaining time samples. All the systems are subjected to two regimes

Sinusoidal Input



(a) Without Disturbance Model

(b) With Disturbance Model

Figure 2: MPC for the Sine Input

RGS Input

The estimation of the state-space model for the RGS input fails miserably as shown in the figure to the right. Since the linear model obtained here, MPC will definitely fail in achieving optimal control for this system.

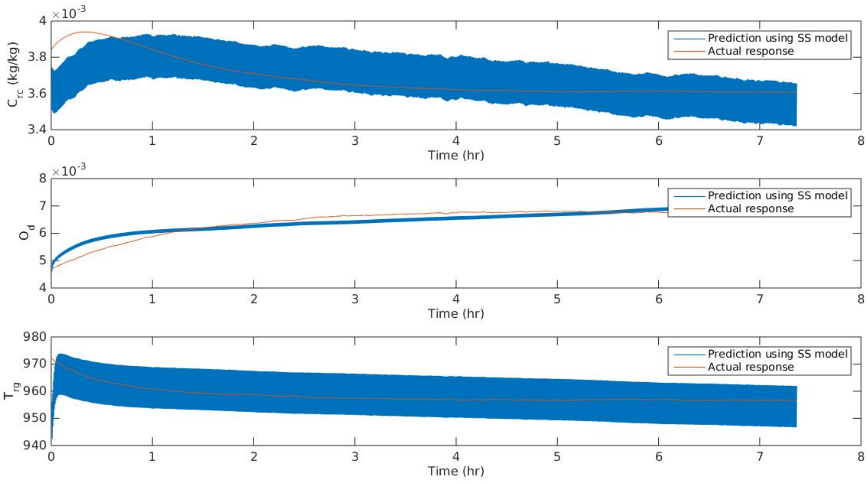
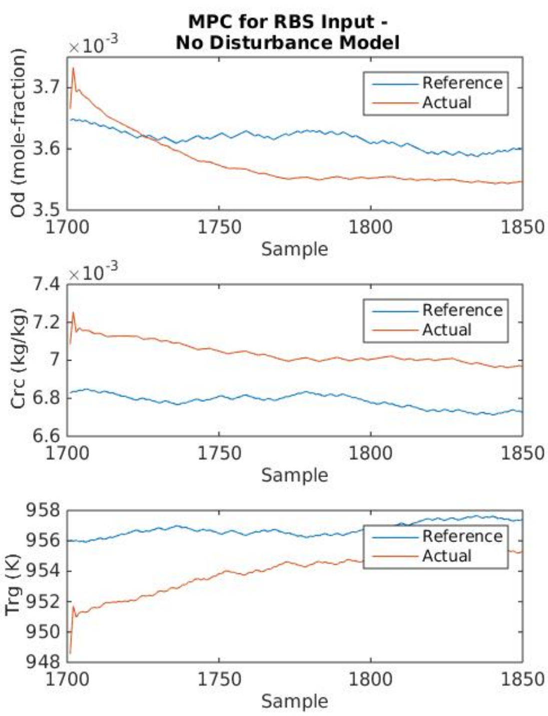
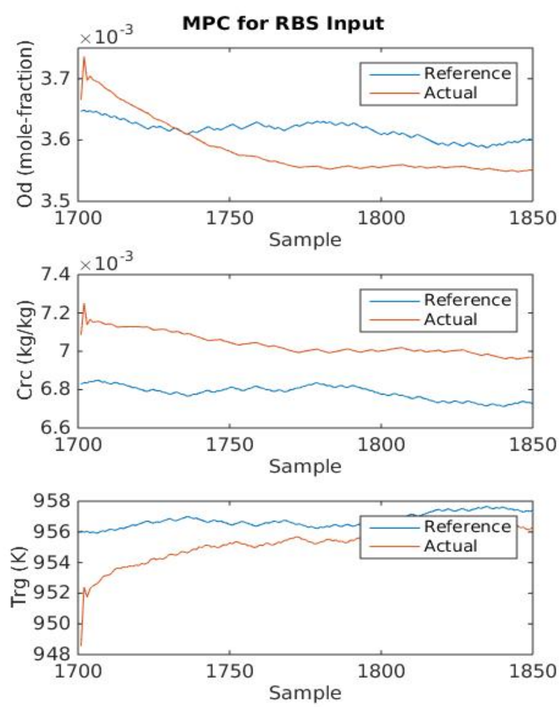


Figure 3: State-Space Model Prediction for RGS input

RBS Input



(a) Without Disturbance Model



(b) With Disturbance Model

Figure 4: MPC for the RBS Input

Observations

1. The controller does a better job here in keeping the difference between the reference and estimated outputs small. (Note that the scale used in figures 2 and 4 is smaller than that used in figure 1). This may be due to either a better state estimate or suitable properties.

2. Adding the disturbance model has little effect on the performance. This is not of a big concern as we have obtained reasonably good control performance.

Conclusions

In this project, we have implemented the MPC algorithm in MATLAB. It has beeen tested on a FCC process simulated using different (feasible) inputs and was found to be giving good results for all the systems. **The MPC algorithm is senstive to the initial state**, and hence it is advisable to get a better estimate usinng the Kalman filter and then implement the MPC block. **The addition of the disturbance model to the MPC algorithm has little impact on the estimates.**