REAL TIME OPTIMIZATION AND MODEL PREDICTIVE CONTROL OF A CSTR

Evan Casper, Srinivas Gudipudi, Saaksshi Jilhewar, Srikar Devarakonda

DATE : 04/25/2023

Contents

Project Summary

The primary objective of this project was to optimize the heat input for a Continuous Stirred Tank Reactor (CSTR) while minimizing the heat price deviation over time by utilizing Dynamic Matrix Control (DMC) and Real-time Optimization. The project was conducted in several stages, beginning with the modeling and simulation of the Ordinary Differential Equation (ODE) system for the CSTR. The CSTR model was then obtained and solved using solve-ivp, followed by the transformation of the continuous model into a discrete model and the derivation of a step response model. This step response model was utilized to formulate the DMC controller, and the implementation of the DMC controller was carried out using relevant code.

To determine the optimal combination of prediction horizon (P), control horizon (M), and weight (w) parameters, ablation studies were conducted by varying these parameters. The results were analyzed to identify the best combination that resulted in optimal heat price control. Finally, the DMC controller was employed to perform real-time optimization on the CSTR, and the obtained results were compared to those obtained without DMC control. The analysis indicated that DMC control significantly improved heat price control.

Overall, this project successfully demonstrated the effectiveness of DMC control in optimizing the heat price of a CSTR. The ablation study helped to identify the best combination of parameters for the DMC controller, and real-time optimization further validated the effectiveness of this approach. These findings hold significant potential for improving the performance and efficiency of industrial processes involving heat transfer, such as chemical reactors.

The findings of this project could have significant implications in various industrial sectors, including the chemical and petrochemical industries, where efficient heat transfer is a crucial factor in determining the process's overall efficiency and cost-effectiveness. By optimizing the heat input of CSTR using DMC control, industries could significantly reduce energy consumption and, consequently, their carbon footprint. This project's approach could be further developed to optimize heat transfer in other industrial processes and contribute to the ongoing global efforts to reduce carbon emissions and mitigate climate change.

Project Objectives

- **Modeling and simulation of ODE system -** Creating a mathematical model of a dynamic system using ordinary differential equations and simulating its behavior.

- **Converting ODE model to discrete-time state-space representation -** Transforming the continuous-time ODE model into a discrete-time state-space model, which is suitable for analysis and control design.

- **Obtaining Step response model -** Generating the step response model of the system to study its dynamic behavior under a sudden change in the input.

- **Formulating the optimization problem for heat price variation -** Defining the optimization problem that aims to minimize the energy cost of the system while satisfying performance and operational constraints, considering the fluctuating energy prices.

- **Developing the DMC algorithm -** Designing a control strategy that uses a predictive model of the system and an optimization algorithm to calculate the optimal control actions over a finite horizon.

- **Performing the ablation study -** Analyzing the performance of the control model by systematically removing or modifying its components to understand their contribution to the overall performance.

- **Performing Real Time Optimization -** Updating the setpoints of the system in real-time to minimize the energy cost while accounting for the time-varying energy prices.

- **Comparing Results -** Evaluating the performance of different control strategies and optimization algorithms by comparing their results in terms of energy cost, system performance, and robustness to disturbances.

Literature Review

Model predictive control (MPC) refers to a class of computer control algorithms that utilize an explicit process model to predict the future response of a plant[2]. One of the main attributes that separate it from more conventional process control like PID is its ability to optimize future behavior by calculating variable adjustments that take place in the future, of which the first sequence is used to determine control action.

Of the several nonlinear MPC algorithms that exist, dynamic matrix control (DMC) uses a linear step response model that calculates corrected process outputs as a function of the past input changes multiplied by the step response coefficients. These coefficients are collected in a matrix known as the Dynamic Matrix, whose shape is determined by the prediction and control horizons, and is what allows the outputs to be calculated analytically. DMC algorithms feature smaller input moves, and as a result these moves are typically less aggressive[2].

Our study is a modification of a study done by Zhang et al[1]. Their goal was to apply machine learning methods to RTO and MPC control of nonlinear processes. Most notably, they used a feed-forward neural network in order to obtain the nonlinear reaction rates, as opposed to the first principles model that gives the rates as the Arrhenius expressions $k_A e^{\frac{-E_A}{RT}} C_A$ and $k_B e^{\frac{-E_B}{RT}} C_B$ (Refer to Table 1 and Eqns 1 & 2). The neural network serves to give the reaction rate as a function of C_A, C_B, and T. This function is then used in both the calculation of control action and RTO. However, we will forgo the neural network model in favor of the ODE model based on first principles.

Unlike our planned study, Zhang et al uses a Lyapunov-based MPC (LMPC), to control C_A, T, and additionally C_B using only $u = Q - Q_s$ as an input. The system is initially operating at steady state values calculated from the ODE model. Once a step change to a new setpoint is implemented, which is calculated from the RTO algorithm, the LMPC solves an minimization problem based on Lyapunov functions in order to determine future control action. This does not require the use of a discrete time model of the system, unlike DMC, and therefore may be preferable when less is known about the system.

When the heat price increases and then decreases to simulate real life changing prices, the optimized set points for C_A also increase and decrease proportionally, while C_B's behavior is inversely proportional. T had a similar trend to C_B, both of which make sense when considering that the reaction is exothermic, less heat is being used due to higher prices, so less of A is being converted to B. However, when looking at the graph of the magnitude of the manipulated input Q vs. time, the control action features large overshoots when a step change is initiated, up to about 2500 cal/s in the direction of the step change. Depending on the process, this may be unacceptable if the deviations cause the instrument to fall outside of safe operating limits. Since improving the control action is one of the main focuses of our study, this is one area that we believe a tuned DMC controller can provide more optimal control action, avoiding large deviations.

The RTO algorithm takes into account both the desire to minimize total costs and the desire to maximize reactant conversion in order to calculate improved set points. This was done by solving an optimization problem using the neural network model. It is worth noting that Zhang et al also showed that the setpoints calculated using the first principles model were the same as using the neural network model for this process, which justifies our use of the former when applying the DMC controller.

The setpoints calculated by the RTO algorithm also showed an improvement in total cost compared to holding the initial steady state values with the fluctuating heat price. While it is inevitable that the operating cost increase to some extent when increasing the heat price, RTO was able to reduce the cost increment by a factor of approximately ⅕ compared to not using RTO.

When comparing the control action of the first principles model and the neural network model for the temperature curves, the accumulated relative error between the two was found to be 4.98×10^{-6}, meaning that the neural network did not offer a significant improvement over the first principles model, further justifying our decision to use it. However, they note that this may not be the case for systems with more complex reaction rates or unknown reaction mechanisms.

Methodology

Building the model and simulating the ODE system

The rate of change of reactant concentration and the rate of change of temperature serve as the two functions that define the CSTR system. Two inputs, two outputs, and two states make up the system. The reactant's concentration and temperature will both be controlled variables in this process. We model and simulate a Continuous Stirred Tank Reactor (CSTR) with a reversible, exothermic reaction . Using mass balance and energy balance equations for the reactants A and the temperature T, the CSTR is mathematically represented as *Eqn 1* and *Eqn 2*. Reversible, first-order kinetics is used to represent the reaction A ⇋ B, and the heat of reaction is taken into consideration as follows. The parameters are defined in Table 1.

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Ao} - C_A) - k_A e^{\frac{-E_A}{RT}} C_A + k_B e^{\frac{-E_B}{RT}} C_B \qquad \text{Eqn. 1}$$

$$\frac{dT}{dt} = -\frac{\Delta H}{\rho C_P}\left(k_A e^{\frac{-E_A}{RT}} C_A - k_B e^{\frac{-E_B}{RT}} C_B\right) + \frac{F}{V}(T_0 - T) + \frac{Q}{\rho C_p V} \qquad \text{Eqn. 2}$$

Table 1: CSTR model parameters

$T_0 = 400$	$\rho = 1$ kg/L
$F_s = 1.67$ L	$C_p = 1000$ cal/(kg K)
$k_A = 5000$ /s	$C_{A0} = 1$ mol/L
$k_B = 1E6$ /s	$V = 100$ L
$E_A = 1E4$ cal/mol	$C_{As} = 0.4977$ mol/L
$E_B = 1.5E4$ cal/mol	$C_{Bs} = 0.5023$ mol/L
$R = 1.987$ cal/(mol K)	$T_s = 426.743$ K
$\Delta H = -5000$ cal/mol	$Q_s = 40{,}386$ cal/s

It can be seen from this model that concentrations of species A (C_A) and temperature of the reactor T are our state variables. From this, we define the inlet volumetric flow rate *F* and heat input to the reactor *Q* as inputs to be varied, while C_A and T are the output variables to be controlled. Furthermore, we assume that the inlet flow only contains A at concentration C_{Ao}. Shown mathematically, this takes the following deviation variable matrix form where *x*, *u*, *y* are system states, system inputs and system outputs respectively.

$$x = \begin{bmatrix} C_A - C_{As} \\ T - T_s \end{bmatrix} \qquad u = \begin{bmatrix} Q - Q_s \\ F - F_s \end{bmatrix} \qquad y = \begin{bmatrix} C_A - C_{As} \\ T - T_s \end{bmatrix}$$

The *solve_ivp* function from the *scipy.integrate* library is used to find the solution of the model's system of ODEs numerically. The code specifies the variables and parameters utilized in the model, such as the reactor parameters, reaction rate constants, and initial conditions. The CSTR function, which takes the present state of the system (C_A, C_B, and *T*) and returns the temporal derivatives of those variables, is used to define the ODE function. The ODE system is then solved, and temperature and concentration vs. time plots are produced using the *solve_ivp* function.

The aforementioned graphs plot the temperature *T* and the concentration of the reactants *A* and *B* as a function of time. As time passes, we see that the concentration of A falls while the concentration of *B* rises to a steady state value. The results from solving these ODE's are discussed in the results section.

Developing Continuous and Converting to discrete state space model
Calculations are made of the system equations' partial derivatives with regard to the state variables C_A and T with the simplification of Taylor's series expansion. The symbols *df1_dA, df2_dA, df1_dT, and df2_dT*, are used to denote the partial derivatives denoted in *A, B, C, D* matrices.
The matrices are converted from continuous to discrete using *scipy.signal.cont2discrete* technique with the *'zoh'* approximation. The continuous-time system matrices *(A, B, C,* and *D)*, the sampling time (*delta_t*), and the approximation method (in this case, 'zoh') are all inputs to the algorithm. The *cont2discrete* method with 'zoh' approximation is used in the code to calculate the phi(and gamma matrices. The implementation and results are discussed in detail in the code file and results section subsequently.

Developing Step Response model
The step response model is developed in order to observe how the system responds to changes in inputs, the step response of the model was analyzed for two different inputs, *Q* and *F*. A discrete-time model was created from the continuous-time model using the zero-order hold method and a sampling period of one second. The step response model is built from scipy.signal.dtsep, by running the model through simulation iterations, the system's step response was determined, with outputs C_A and T plotted against time for both input steps. It was observed that the outputs approached their steady-state values after hundred seconds. The results of the step response are discussed in the results section.

Developing DMC
The DMC function is an implementation of the Dynamic Matrix Control algorithm for controlling a process system. It takes in system matrices *A, B, C,* and *D*, the number of inputs and outputs, the prediction horizon, the control horizon, the sampling time, the setpoint, and the initial conditions.

The function first converts the continuous-time system to a discrete-time system and calculates the step response of the system. It then creates matrices for the past and future values of the output and input signals. The function then initializes the control inputs and calculates the predicted output values and control inputs based on the DMC algorithm. The function then updates the state of the system and stores the output and input values. Finally, the function returns the output values and control inputs.

The DMC algorithm uses a model of the system to predict the future behavior of the system by using the current state and past values of the system and calculates the control inputs that minimize a cost function that penalizes deviations from the setpoint and changes in the control inputs.

Performing Ablation Study

The DMC algorithm was tested with three different values of *P* (number of past control moves used to calculate the control action), *M* (number of past measurements used in the prediction model), and *W* (weighting factor of the control moves in the objective function). The results show that the DMC algorithm was able to control the reactor outputs within acceptable ranges for all tested values of *P, M,* and *W.*

Performing Real Time Optimization(RTO)

The optimization objective of the RTO algorithm is to minimize a cost function that is a combination of the conversion and heat cost. A simple linear function has been taken as represented in the equation below:

$$Total\ cost = \frac{C_A}{C_{A0}} + heat\ price * Q$$

The results of the RTO has been discussed below, A situation is considered where heat price is changing for a RTO period of every 1000 seconds and the above algorithm is implemented in Pyomo using GAMS/IPOPT to find the optimal set point within each period , then the DMC algorithm is implemented within each interval to drive the system towards optimality each time to keep the cost low .

Results and Discussion

Step response model:

The step response model was obtained from the linearized State Space representation as described, and the output looked as below. The deviation variables have been considered for this. As can be observed the variables start at an initial value and then converge to a steady state.s the dynamic behavior of the output has been plotted and we can observe that both variables converge towards the steady state.

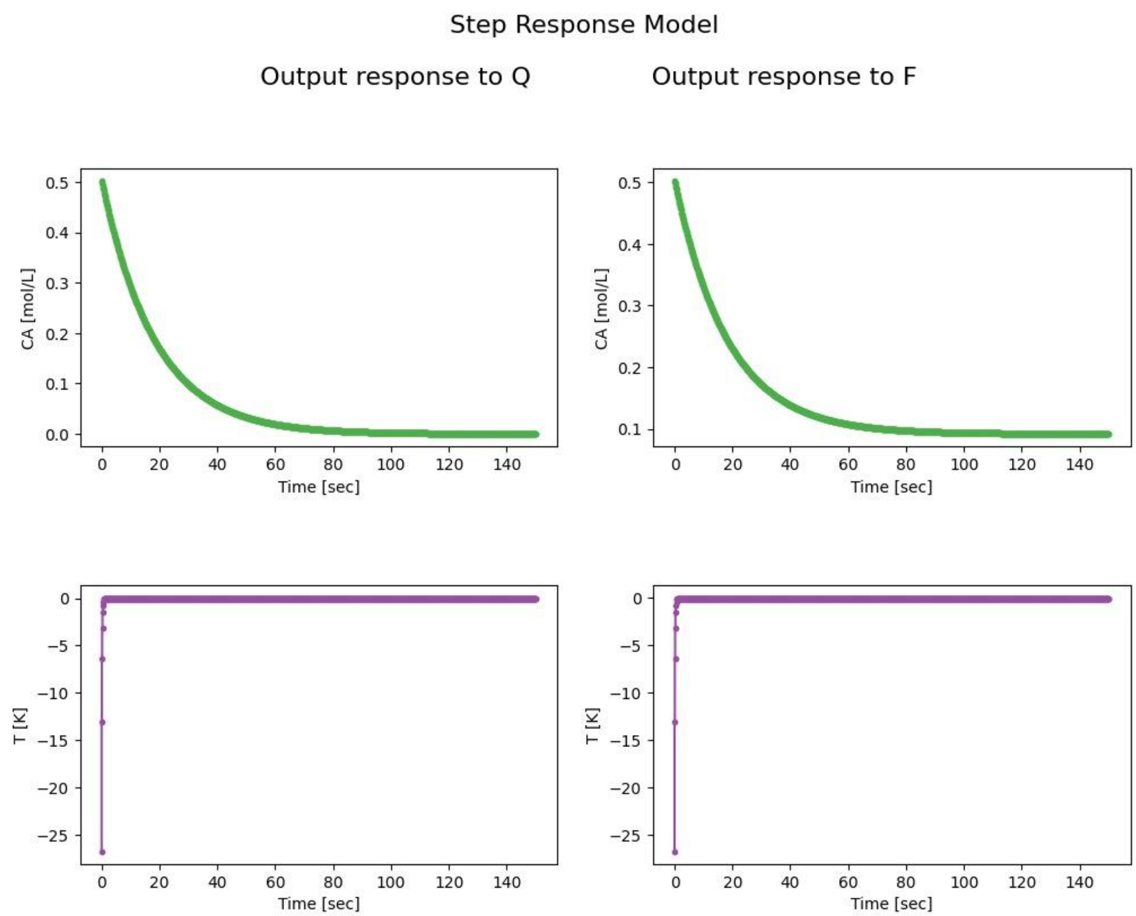

Figure 1: Step responses for outputs CA and T responding to Q and F

DMC output behavior and ablation study

After implementing the DMC algorithm with tuning parameters of P=8, M=2, and w=1, we observed that all input and output variables achieved stabilization except for Q. This variable did not show proper convergence due to its magnitude being significantly higher than other variables. As a result, it underwent negligible changes per step change, leading to minimal changes in its value. Consequently, Q did not show much variation during the stabilization process.

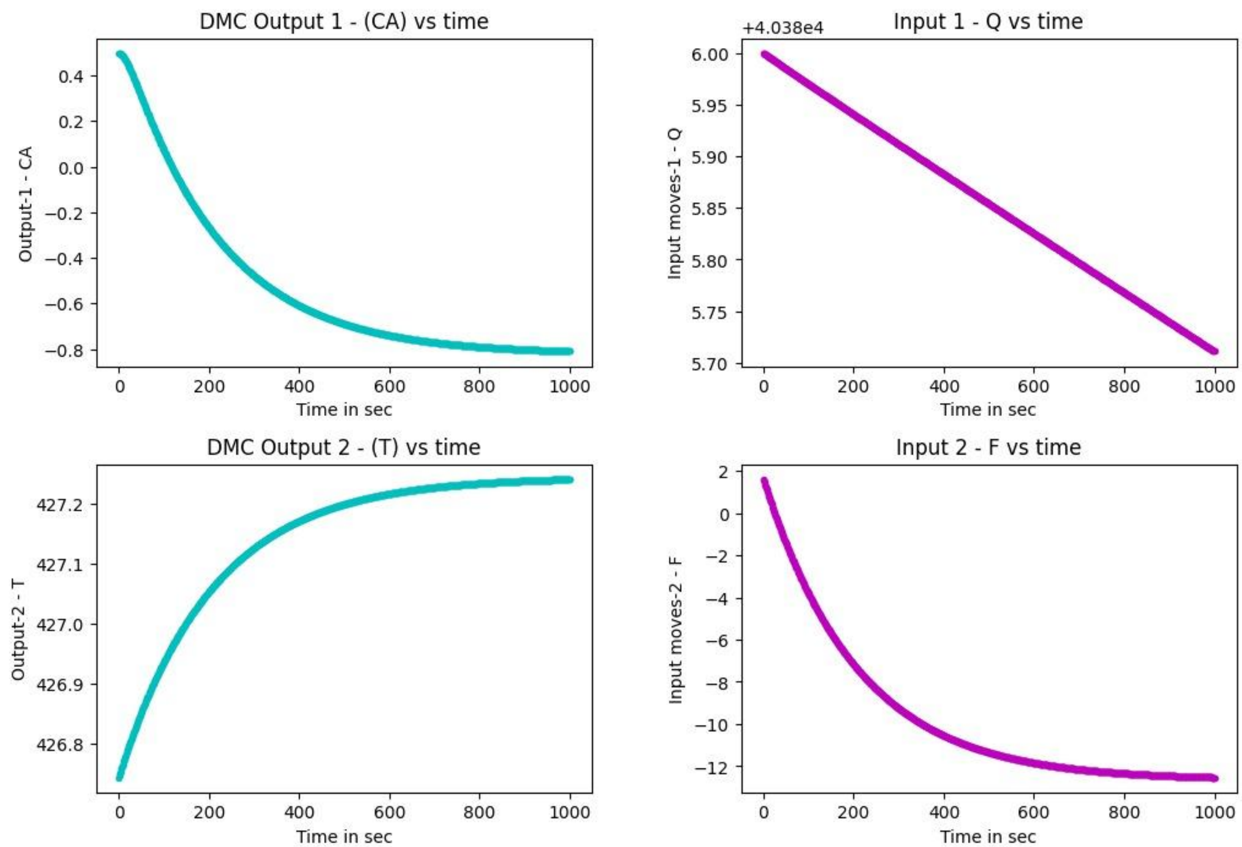

Figure 2: DMC outputs and inputs with P=8, M=2, w=1

Ablation study

The parameters P,M and w were varied in an attempt to optimize the DMC control behavior

Variation of the prediction horizon (P): (Figure 3)

We can observe that as P increases, we can see better control behavior here and the output variables reaching the set point faster. Theoretically, that is because increasing the prediction horizon means that the algorithm can make better predictions, as it accounts for disturbances better, and this helps generate more stable control inputs as well.

Variation of Control horizon (M): : (Figure 4)

The variation of control horizon (M) doesn't affect the control behavior a lot in this case, as we can see that the plots nearly coincide. This generally means that the chosen control horizon is appropriate and increasing it further doesn't affect the system as much.

Variation of weight(w): (Figure 5)

The factor w represents the weight applied to past inputs, and it works in slowing down the reaction times of the system and smoothing the response behavior. This can be observed in the above plots, as w increases we get a less abrupt response and slower reaction times.

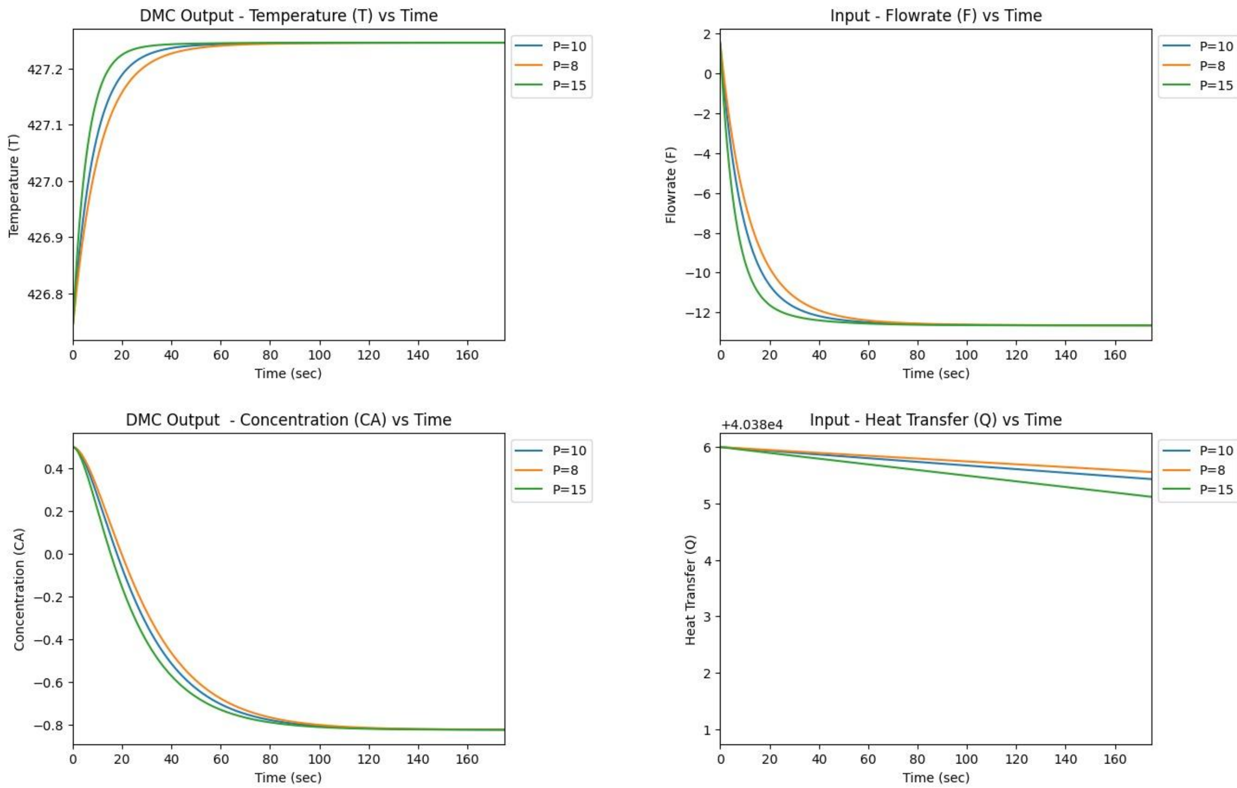

Figure 3: DMC outputs and inputs with P=8, 10, 15

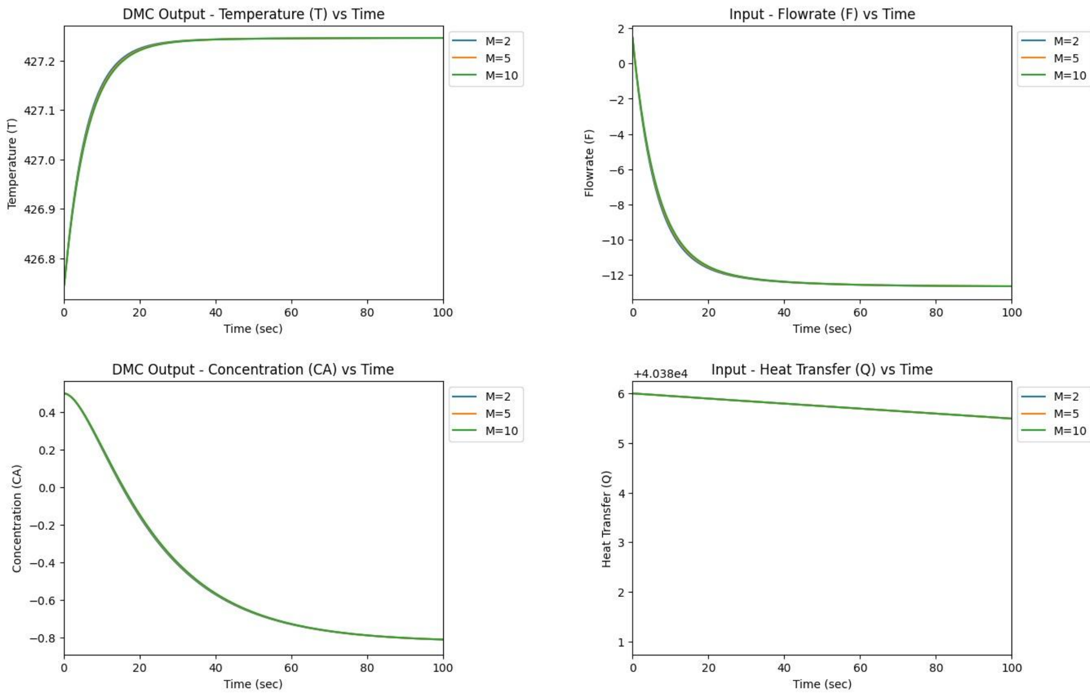

Figure 4: DMC outputs and inputs with M=2, 5, 10

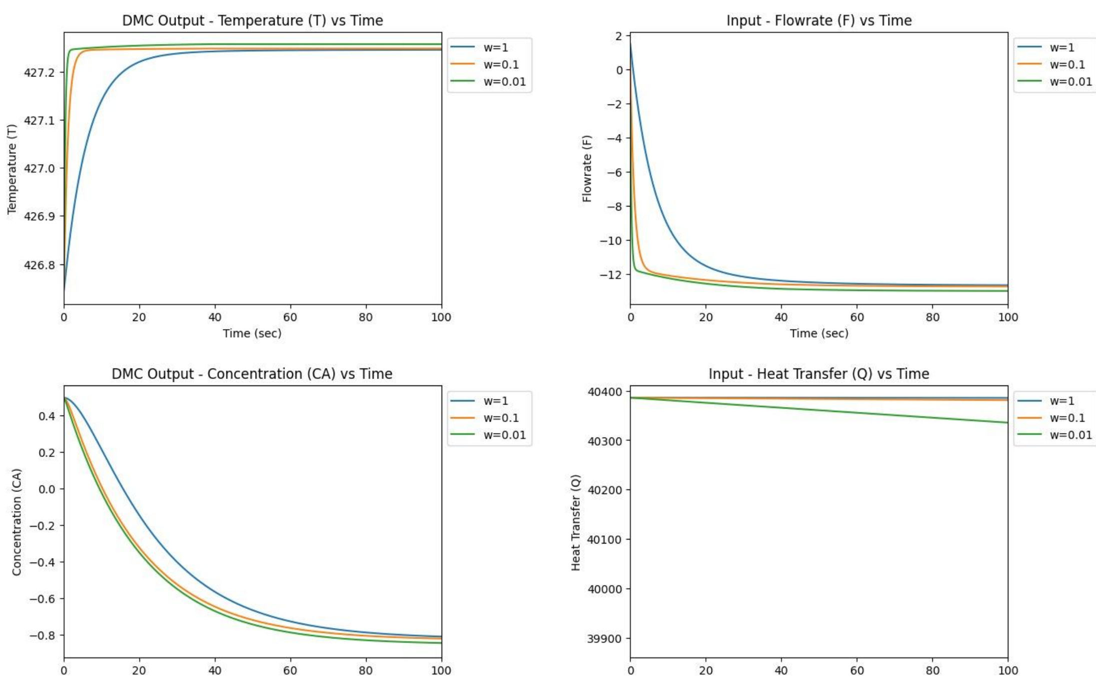

Figure 5: DMC outputs and inputs with w=1, 0.1, 0.01

Real-Time optimization:

The Real Time Optimization involved calculation of optimal set-points for an RTO period of 1000 sec. IPOPT was used for this purpose. The heat price variation over the time periods is given in the following figure.

We can observe that the temperature is decreasing as the heat price increases and conversion decreases. This is because the optimization algorithm sacrifices heat input to obtain the optimal set point for each time period.

The objective values can be seen in Figure 7:

The black dotted line shows the variation of the cost while keeping initial steady state setpoints constant. The blue line is the total cost with the improved RTO setpoints.

Now, connecting the DMC algorithm along with RTO, we want to make sure that within each time period the DMC loop is working toward the optimal set-point in each case. The output behavior of the DMC loop can be observed in the following graphs:

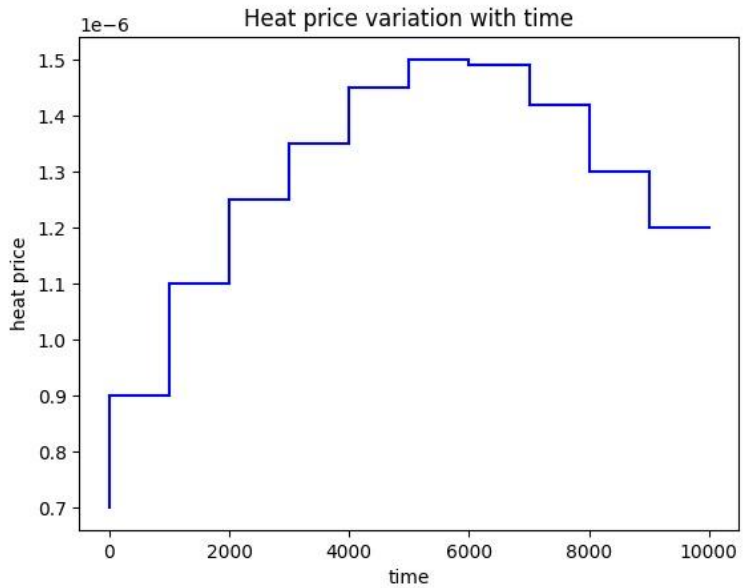

Figure 6: Heat price variation and CA setpoint variation from RTO algorithm

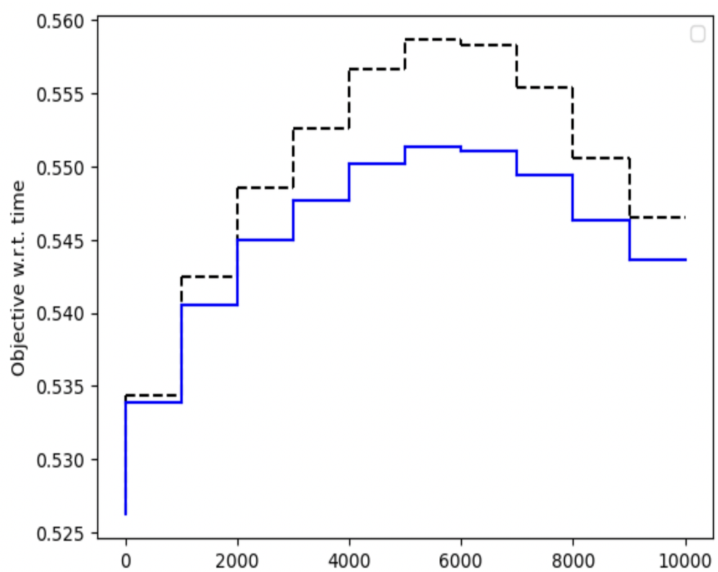

Figure 7: Total energy cost with and without RTO

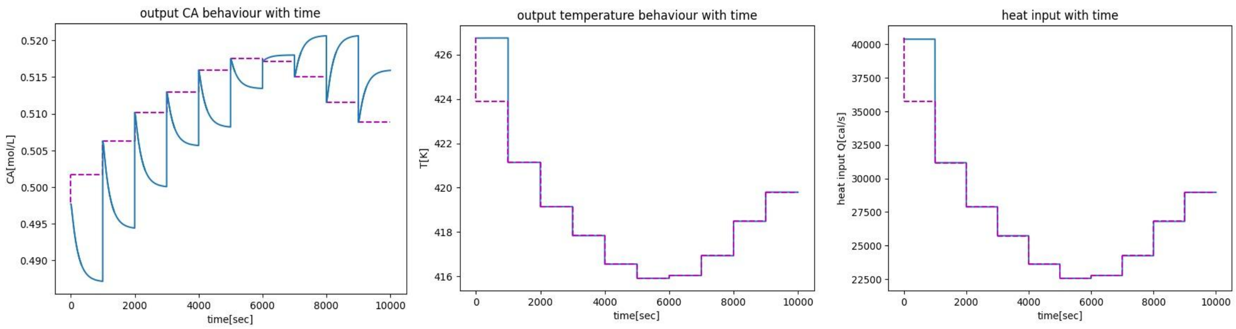

Figure 8: Output CA and T, and input Q response to RTO step changes

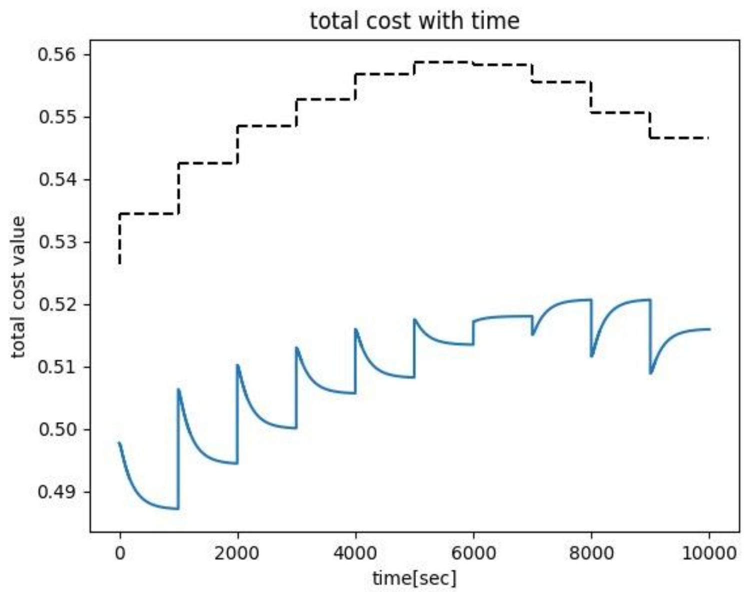

Figure 9: Total energy cost without RTO and with RTO + DMC control

In Figure 8, we can see that the blue line represents the output from DMC controlled with RTO. We can observe that our controller can mange the output temperature and heat input smae as, the given set points given by trh rto shown by magenta lines. The devaitaion can be observed in the Ca graph which might be a possibility of the scalability issue as mentioine dprior. We can also see in Figure 9 that, we have better coust value minnimixation represented in blue lines and the black line show the cost values without rto.

Result Comparison

We were able to replicate the results of a research paper on Real Time Optimization (RTO) and successfully control the time optimization of heat price. However, we encountered challenges when scaling the parameter Q, which resulted in sinusoidal plots for other parameters when observing the input and output step response. This may have contributed to the heavy deviation in the concentration of A, as shown in Figure 8. Although we achieved better results with cost optimization compared to the paper, we cannot claim full confidence due to the high error in stability. Therefore, further analysis and experimentation are required to increase the reliability of our findings.

Conclusion

In conclusion, the findings of this project have significant implications in various industrial sectors and hold significant potential for improving the efficiency of heat transfer processes while reducing energy consumption and carbon emissions. The approach could be further developed to optimize other parameters and applied to larger and more complex industrial processes. From the performed experiment it is quite important to do the scaling of the parameters before performing DMC, for it to better update the step at every time interval.

However, it is important to note that this project's scope was limited to a single CSTR. Further research could expand the scope of this project by considering the effects of different disturbances on the CSTR's heat price control. One such disturbance that could be examined is the variation in the feed flow rate to the CSTR. Additionally, it would be interesting to investigate the scalability of this project to larger and more complex industrial processes.

References

1. Zhang, Zhihao, Zhe Wu, David Rincon, and Panagiotis D. Christofides. "Real-time optimization and control of nonlinear processes using machine learning." *Mathematics* 7, no. 10 (2019): 890.
2. S. Joe Qin, Thomas A. Badgwell, "A survey of industrial model predictive control technology", Control Engineering Practice 11 (2003) 733–764
3. Pyomo. (2023). Pyomo: Python Optimization Modeling Objects. Retrieved from https://pyomo.org/
4. Wächter, Andreas, and L.T. Biegler. "On the implementation of an interior-point algorithm for nonlinear programming with inexact step computations." Mathematical Programming 136, no. 1-2 (2012): 209-227. DOI: 10.1007/s10107-011-0470-1

$$\min_{C_A,T,Q} \ Total\ cost = \frac{C_A}{C_{A_0}} + Heat\ Price \times Q$$

$$s.t.\ \ 0 = \frac{1}{\tau}(C_{A_0} - C_A) - k_A C_A e^{\frac{-E_A}{RT}} + k_B C_B e^{\frac{-E_B}{RT}}$$

$$0 = -\frac{1}{\tau}(C_B) + k_A C_A e^{\frac{-E_A}{RT}} - k_B C_B e^{\frac{-E_B}{RT}}$$

$$0 = \frac{-\Delta H}{\rho C_p}(k_A C_A e^{\frac{-E_A}{RT}} - k_B C_B e^{\frac{-E_B}{RT}}) + \frac{1}{\tau}(T_0 - T) + \frac{Q}{\rho C_p V}$$

$$C_A \in [0, 1]$$

$$C_B \in [0, 1]$$

$$T \in [400, 500]$$

$$Q \in [0, 10^5]$$