Programming Exercise for Model Predictive Control

Fenglong Song, Wenbo Wang, Hui Zhang

May 18, 2021

Part I
System Modeling

1 Continuous System Modeling

We can write the system dynamics in following form:

$$\begin{bmatrix} \dot{T}^c_{VC}(t) \\ \dot{T}^c_{F1}(t) \\ \dot{T}^c_{F2}(t) \end{bmatrix} = A^c \begin{bmatrix} T^c_{VC}(t) \\ T^c_{F1}(t) \\ T^c_{F2}(t) \end{bmatrix} + B^c \begin{bmatrix} p^c_{VC}(t) \\ p^c_{F1}(t) \\ p^c_{F2}(t) \end{bmatrix} + B^c_d d^c$$

where:

$$A^c = B^c_d \begin{bmatrix} -\alpha_{F1,VC} - \alpha_{F2,VC} - \alpha_{Env,VC} & \alpha_{F1,VC} & \alpha_{F2,VC} \\ \alpha_{VC,F1} & -\alpha_{VC,F1} - \alpha_{F2,F1} & \alpha_{F2,F1} \\ \alpha_{VC,F2} & \alpha_{F1,F2} & -\alpha_{VC,F2} - \alpha_{F1,F2} \end{bmatrix}$$

$$B^c = B^c_d \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} \\ \beta_{21} & \beta_{22} & \beta_{23} \\ \beta_{31} & \beta_{32} & \beta_{33} \end{bmatrix}, \quad d^c = \begin{bmatrix} d_{VC} + \alpha_{Env,VC} T_{Env} \\ d_{F1} \\ d_{F2} \end{bmatrix}, \quad B^c_d = \begin{bmatrix} \frac{1}{m_{VC}} & 0 & 0 \\ 0 & \frac{1}{m_{F1}} & 0 \\ 0 & 0 & \frac{1}{m_{F2}} \end{bmatrix}$$

2 System Discretization

Using Euler's Discretization Method:

$$\dot{T}^c_{VC}(t) = \frac{T^c_{VC}(t + T_s) - T^c_{VC}(t)}{T_s}$$

we get:

$$T(k+1) = AT(k) + Bp(k) + B_d d$$

where

$$A = T_s A^c + I, B = T_s B^c, B_d = T_s B^c_d$$

3 Steady State and Delta Formulation

The steady state T_{sp} and its corresponding input p_{sp} satisfy:

$$\begin{bmatrix} I - A & -B \\ C_{ref} & 0 \end{bmatrix} \begin{bmatrix} T_{sp} \\ p_{sp} \end{bmatrix} = \begin{bmatrix} B_d d \\ b_{ref} \end{bmatrix}$$

where

$$C_{ref} = I, \quad b_{ref} = \begin{bmatrix} 25 \\ -42 \\ -18.5 \end{bmatrix}$$

Solve the equation, we get the value of steady state and input:

$$T_{sp} = \begin{bmatrix} 25 \\ -42 \\ -18.5 \end{bmatrix}, \quad p_{sp} = \begin{bmatrix} 8237.3 \\ -4911.1 \\ -241.5 \end{bmatrix}$$

The Delta-Formulation of the system is

$$\Delta T(k+1) = A\Delta T(k) + B\Delta p(k)$$

4 State Constraints

The Original constraints are:

$$T_{min} \leq T \leq T_{max}, \quad p_{min} \leq p \leq p_{max}$$

In Delta formulation, state constraints are:

$$T_{min} - T_{sp} \leq \Delta T \leq T_{max} - T_{sp}, \quad p_{min} - p_{sp} \leq \Delta p \leq p_{max} - p_{sp}$$

Take values in, we get:

$$\begin{bmatrix} -3 \\ -3 \\ -0.5 \end{bmatrix} \leq \begin{bmatrix} \Delta T_{VC} \\ \Delta T_{F1} \\ \Delta T_{F2} \end{bmatrix} \leq \begin{bmatrix} 2 \\ 3 \\ 1.5 \end{bmatrix}, \quad \begin{bmatrix} -8237.3 \\ -2588.9 \\ -1258.5 \end{bmatrix} \leq \begin{bmatrix} \Delta p_{VC} \\ \Delta p_{F1} \\ \Delta p_{F2} \end{bmatrix} \leq \begin{bmatrix} 6762.7 \\ 4911.1 \\ 241.5 \end{bmatrix}$$

Part II
Unconstrained Optimal Control

5 Autonomous System Simulation

The evolution of the autonomous system start from T_{sp}, without any controller is as figure 1. Since there is no controller at all, the temperatures change according to the law of heat transfer. T_{VC} decreases since it's higher than temperature of the environment T_{Env}. T_{F1} and T_{F2} increase since they're both lower than their surrounding temperature T_{VC}.

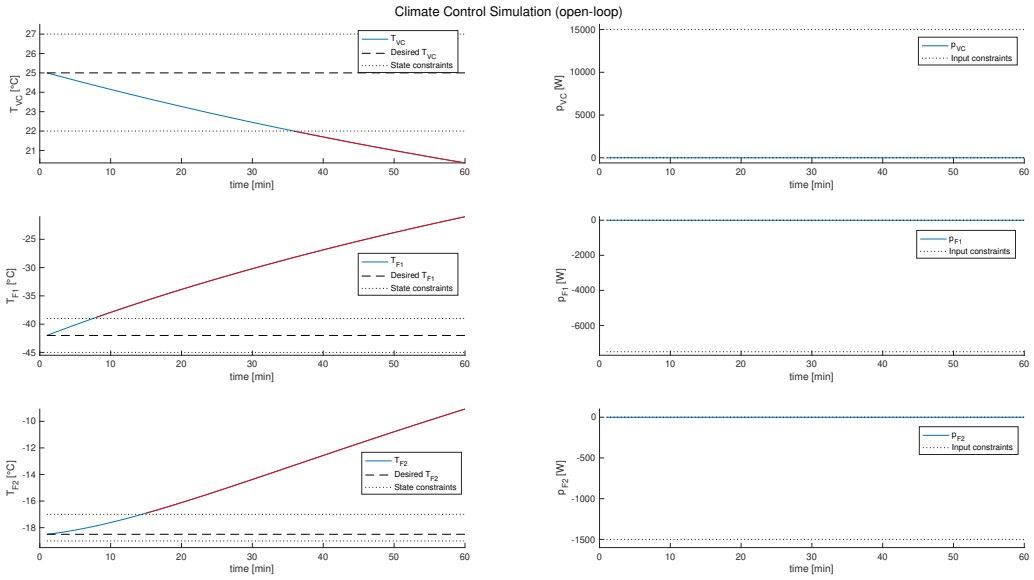


Figure 1: Autonomous system evolution

6 Heuristic LQR Tuning

The heuristic approach to find the best Q for T_{01} is shown as figure 2. The best Q point is filled with black and its corresponding value is:

$$Q_1 = \begin{bmatrix} 4973679 & 0 & 0 \\ 0 & 5427908 & 0 \\ 0 & 0 & 4949349 \end{bmatrix}$$

It's quite natural that the more energy it consumes in certain amount of time, the better temperature control it can achieve. That is, larger power_sum cooresponds to smaller dT_relative, which matches the bottom blue curve in figure 2.

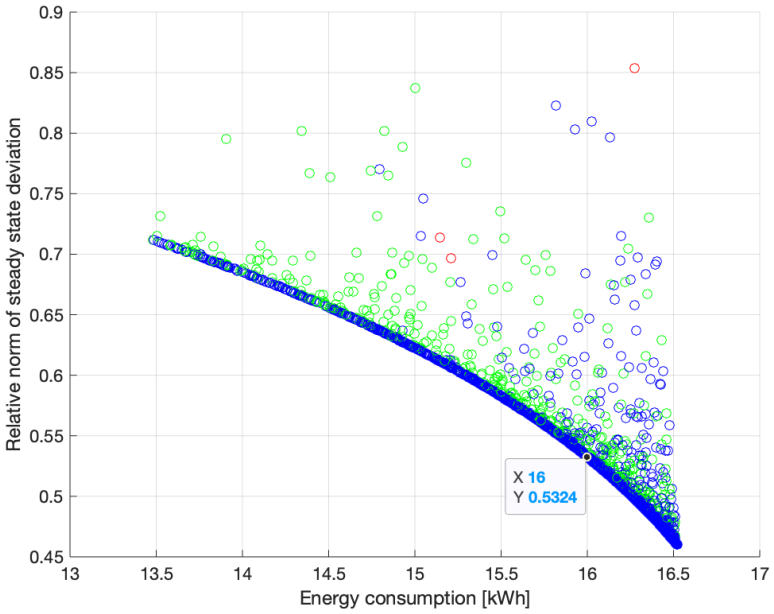


Figure 2: Heuristic approach for T_01

7 LQR Simulation for T_{01} and influence of Q and R

System evolution with LQR controller starting from T_01 is as figure 3.

Change of diagonal elements in Q and R means different penalty weight on corresponding state or input.

1. The larger R, more heavily the input is penalized. The controller will be more energy-saving and it will take longer time for the closed-loop system to converge to the desired state, i.e., to reach the target temperature.
2. The larger Q, more heavily the state is penalized. The controller may consume more energy but it will take shorter time for the closed-loop system to converge to the desired state, i.e., to reach the target temperature.

8 LQR Simulation for T_{02}

Starting from $T(0) = T_{init}^{(2)}$, we've recalculated the best Q with heuristic approach in Section 6. Best choice of Q_2 is:

$$Q_2 = \begin{bmatrix} 5836090 & 0 & 0 \\ 0 & 1342679 & 0 \\ 0 & 0 & 3555303 \end{bmatrix}$$

The plot of heuristic approach is as figure 4.

System evolution under LQR controller is shown in figure 5.

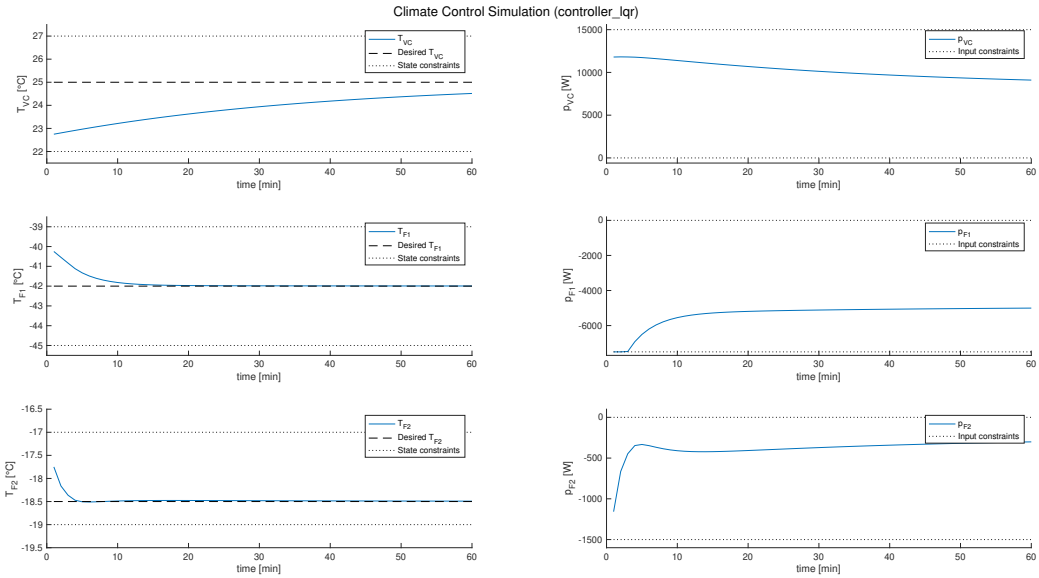


Figure 3: LQR controller for T_01

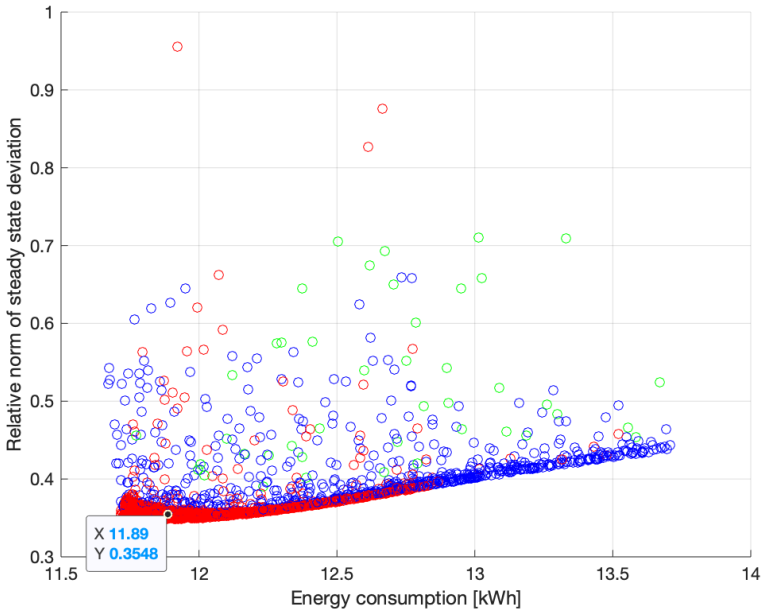


Figure 4: Heuristic approach for T_02

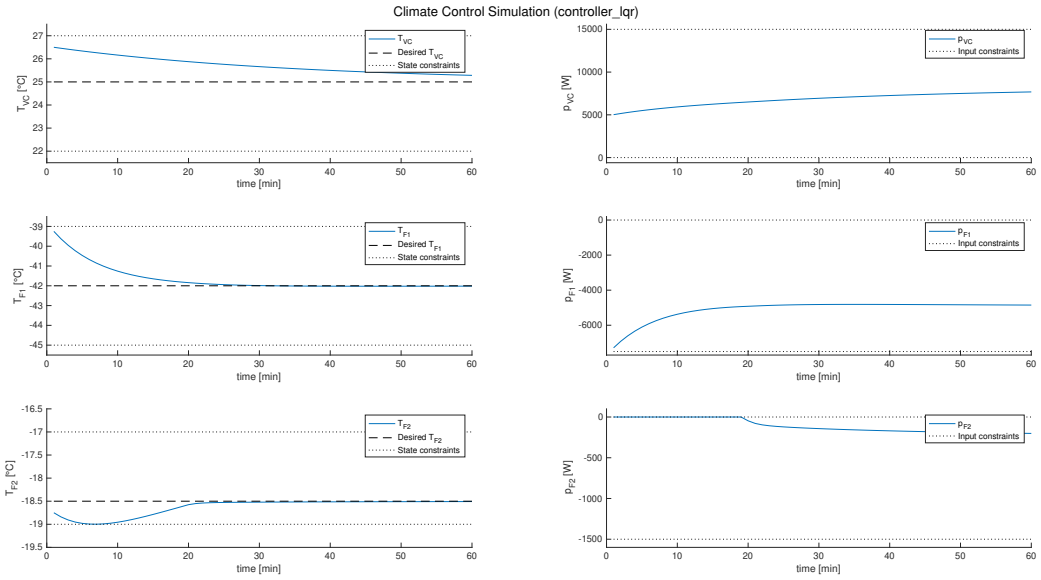


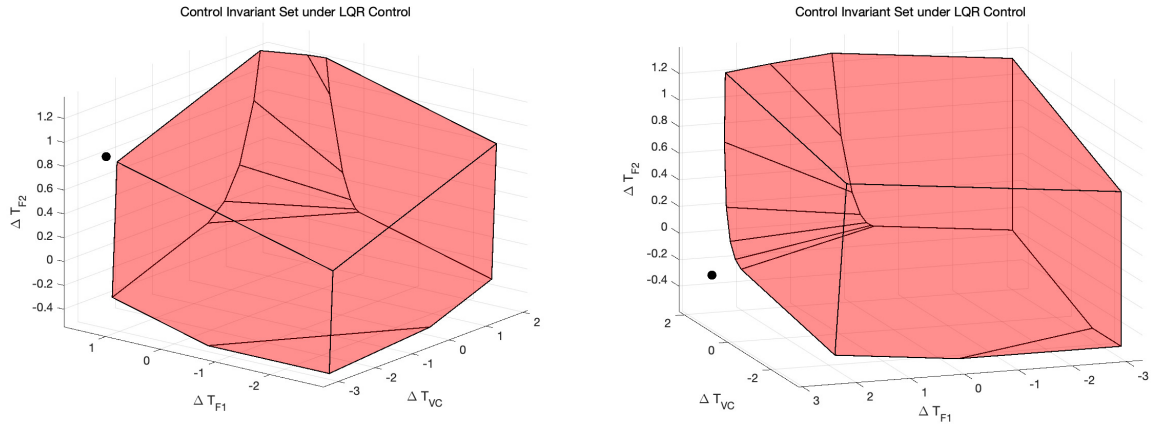
Figure 5: LQR controller for T_02

Part III
From LQR to model predictive controller

9 Compute invariant set

The resulting polyhedron with $T(0) = T_{init}^{(1)}$ and $T(0) = T_{init}^{(2)}$ are shown in figure 6(a) and 6(b), respectively. Initial conditions in Delta-Formulation $\Delta T_{init}^{(1)} = T_{init}^{(1)} - T_{sp}$ and $\Delta T_{init}^{(2)} = T_{init}^{(2)} - T_{sp}$ are drawn cooresponding in each figure as black dots.

From the plot we can see that neither of the two initial conditions $\Delta T_{init}^{(1)}$ and $\Delta T_{init}^{(2)}$ is in the Positive Invariant Set of LQR controller.



(a) Polyhedron for Q_1

(b) Polyhedron for Q_2

Figure 6: Polyhedron for Q_1 and Q_2 with initial conditions

10 Compute LQR infinite horizon cost

The infinite horizon cost under LQR control law is

$$J(x(0)) = x(0)^T P_\infty x(0)$$

where P_∞ is the solution of the Algebraic Ricatti Equation:

$$P_\infty = A^T P_\infty A + Q - A^T P_\infty B(B^T P_\infty B + R)^{-1} B^T P_\infty A \tag{1}$$

11 From LQR to MPC controller

System evolution with $T(0) = T_{init}^{(1)}$ and $T(0) = T_{init}^{(2)}$ are shown in figure 7 and 8, respectively.

The results in figure 7 and 8 under mpc_controller_1 are the same as LQR controller. That's because in Task 7 and Task 8 there is no state or input violation with LQR controller, so our MPC controller is giving exactly the same control inputs with LQR controller.

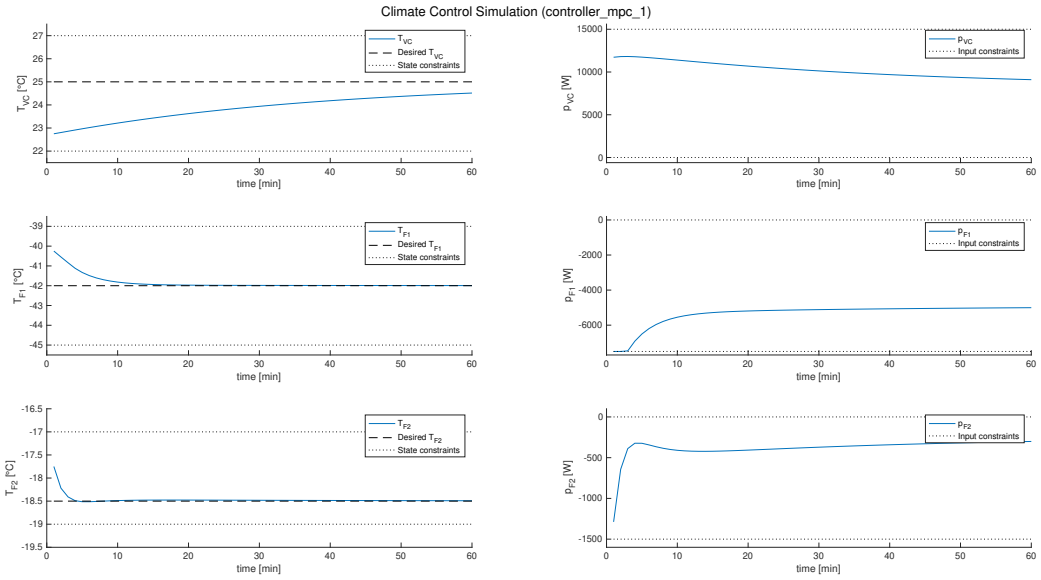


Figure 7: MPC controller 1 for T_01, scen1

Part IV
MPC with theoretical closed-loop guarantees

12 MPC with theoretical closed-loop guarantees

(8f) implies we have a terminal state constraint that $x_N \in \chi_f = \{0\}$. We will show that the optimal cost function $J^*(x)$ is a Lyapunov function.

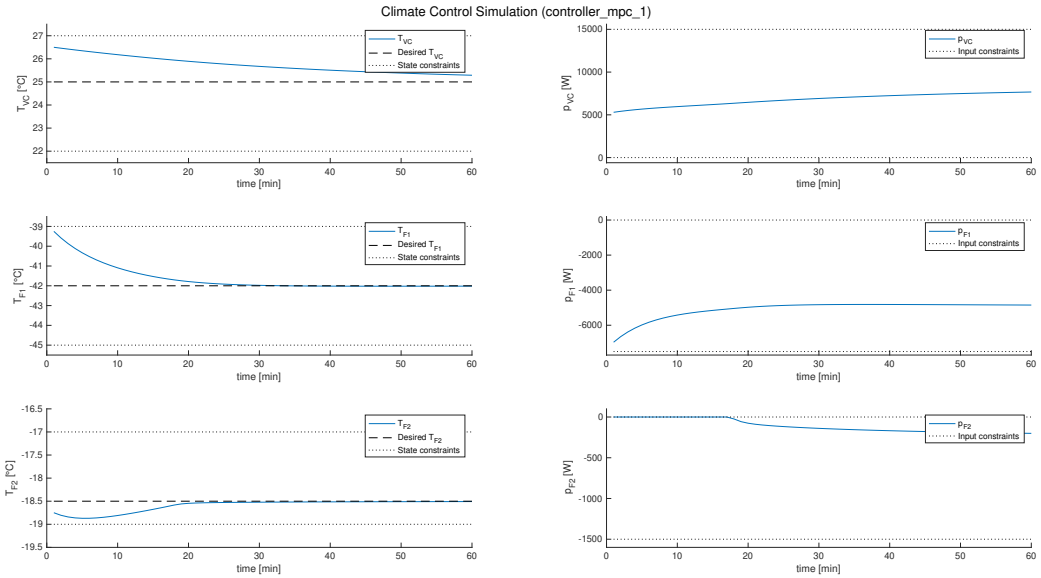


Figure 8: MPC controller 1 for T_02, scen1

$$
\begin{aligned}
J^*\left(x((k+1)\right) &\leq \sum_{i=1}^{N-1} l(x_i^*, u_i^*) + l(x_N^*, 0) \\
&= \sum_{i=0}^{N-1} l(x_i^*, u_i^*) - l(x_0^*, u_0^*) + l(x_N^*, 0) \\
&= J^*\left(x(k)\right) - l\left(x(k), u_0^*\right) + l(0,0) \\
&= J^*\left(x(k)\right) - l\left(x(k), u_0^*\right)
\end{aligned}
$$

Since $l\left(x(k), u_0^*\right) \geq 0$, $J^*(x)$ is a Lyapunov function. This implies asymptotically stability is ensured. So the origin is an asymptotically stable equilibrium point for closed-loop system.

13 MPC with zero terminal set

The closed-loop simulation for $T(0) = T_{init}^{(1)}$ and $T(0) = T_{init}^{(2)}$ under mpc_controller_2 are shown in figure 9 and 10, respectively.

14 MPC with larger terminal set

The terminal cost $l_f(x_N)$ is chosen as a quadratic function ${x_N}^T P_\infty x_N$, where P_∞ is the solution of equation 1, which is the Algebraic Ricatti Equation.

The closed-loop simulation for $T(0) = T_{init}^{(1)}$ and $T(0) = T_{init}^{(2)}$ under mpc_controller_3 are shown in figure 11 and 12, respectively.

15 Comparison of three MPC controllers

The comparison of costs starting from T_01 and T_02 are in table 1 and 2, repectively.

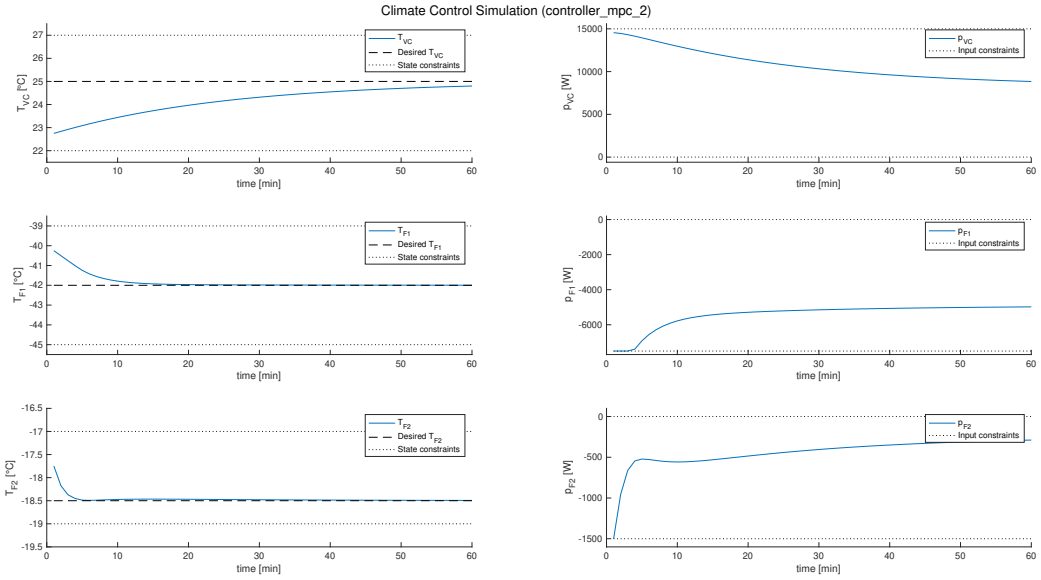


Figure 9: MPC controller 2 for T_01, scen1

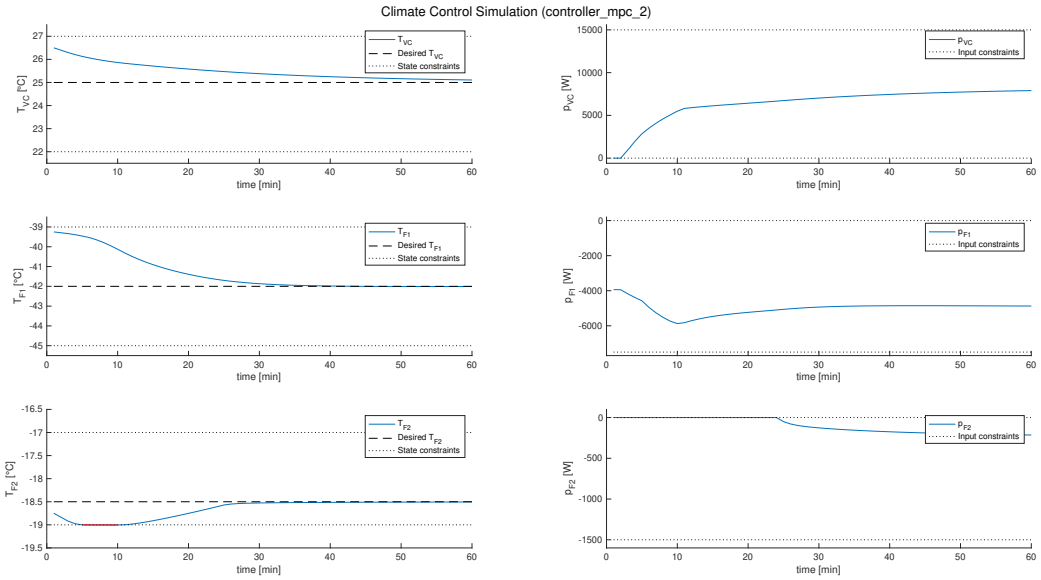


Figure 10: MPC controller 2 for T_02, scen1

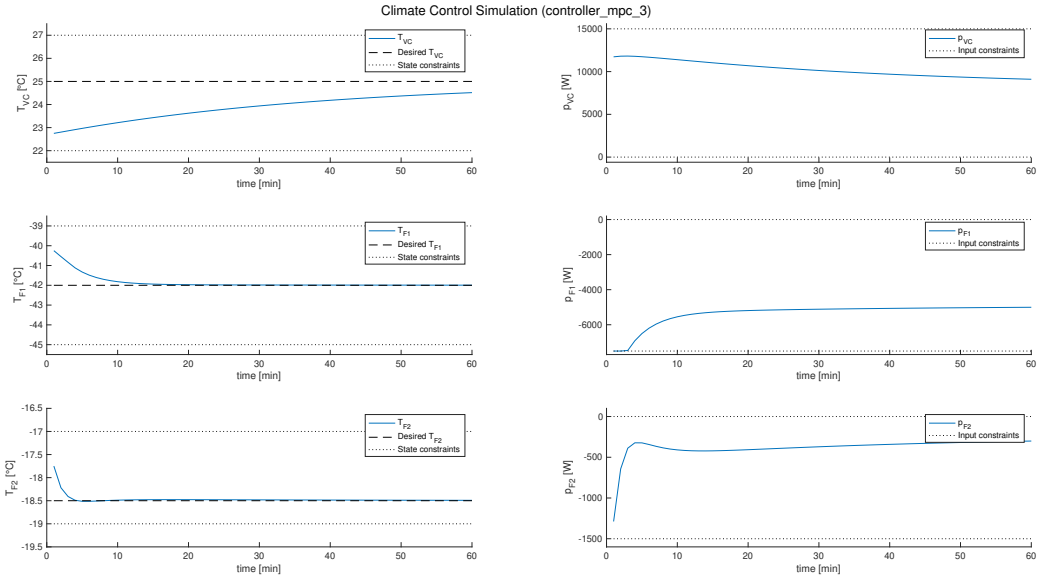


Figure 11: MPC controller 3 for T_01, scen1

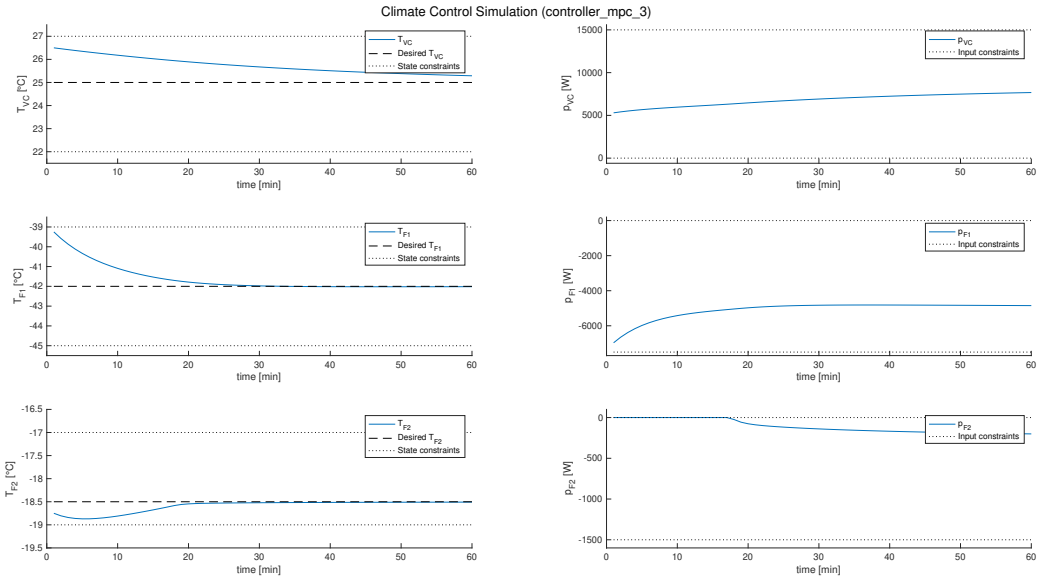


Figure 12: MPC controller 3 for T_02, scen1

Table 1: Costs with different controllers starting from T_01

	mpc controller 1	mpc controller 2	mpc controller 3
state cost J_x^*	4.827×10^8	3.277×10^8	4.827×10^8
input cost J_u^*	8.122×10^9	8.992×10^9	8.122×10^9
total cost J_{total}^*	8.605×10^9	9.320×10^9	8.605×10^9

Table 2: Costs with different controllers starting from T_02

	mpc controller 1	mpc controller 2	mpc controller 3
state cost J_x^*	2.650×10^8	2.166×10^8	2.650×10^8
input cost J_u^*	4.356×10^9	4.181×10^9	4.365×10^9
total cost J_{total}^*	4.621×10^9	4.398×10^9	4.621×10^9

By comparing the costs and plots, we noticed two things.

1. The mpc_controller_1 and mpc_controller_3 achieve almost identical performances, which means the terminal set constraints $x_N \in \chi_f$ are automatically satisfied, i.e. the terminal states given by mpc_controller_1 lie in χ_f.
2. The mpc_controller_2 gives larger input cost J_u^* and smaller state cost J_x^* than the other two controllers. This is because of the terminal set constraint $x_N \in \chi_f = \{0\}$, which is very strict. Achieving such a terminal set constraint of course reduces the state cost J_x^* which is a quadratic function of the state. However, to satisfy this demanding constraint also consumes larger energy, which means larger input cost J_u^*.

16 Using different terminal set

The intersection set $\chi_f \cap \chi_{LQR}$ is also an invariant set under LQR controller $u(k) = F_\infty x(k)$.

Proof: $\forall x(k) \in \chi_f \cap \chi_{LQR}$, we have $x(k) \in \chi_f$ and $x(k) \in \chi_{LQR}$ for $\forall k \in \{0, 1, \dots\}$. Both χ_f and χ_{LQR} are invariant of the closed-loop system, therefore $x(k+1) \in \chi_f$ and $x(k+1) \in \chi_{LQR}$, so $x(k+1) \in \chi_f \cap \chi_{LQR} \Rightarrow$ $\chi_f \cap \chi_{LQR}$ is an invariant set under this LQR controller.

Using $\chi_f \cap \chi_{LQR}$ as the terminal set still yields an asymptotically stable MPC controller.

Proof: We will show the three assumptions of asymptotically stable MPC are met in this case.

1. Stage cost is positive definite. This is obvious since the stage cost is quadratic function and both Q and R are positive definite.
2. Terminal set is invariant under the local control law and all state and input constraints are satisfied. This have been proved earlier in this task.
3. Terminal set is a continuous Lyapunov function in the terminal set χ_f and $l_f(x_{i+1}) - l_f(x_i) \leq -l(x_i, \kappa_f(x_i))$, $\forall x_i \in \chi_f$. Since $\chi_f \cap \chi_{LQR} \subseteq \chi_{LQR}$, this property still holds.

Therefore, taking $\chi_f \cap \chi_{LQR}$ as the terminal set still yields an asymptotically stable MPC controller.

Part V
Soft constraints

17

The closed-loop simulation for $T(0) = T_{init}^{(1)}$ in scen2 under mpc_controller_3 are shown in figure 13.

We noticed that the state constraints are strongly violated after the delivery process starts. Besides, all three inputs are saturated. This is because the disturbance is so strong that the optimization problem in solving MPC becomes infeasible.

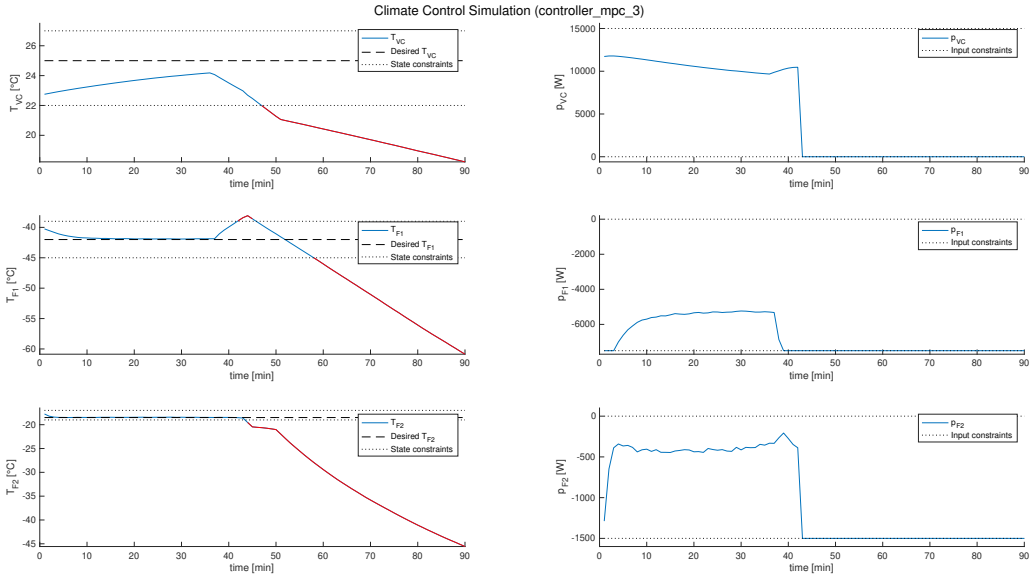


Figure 13: MPC controller 3 for T_01, scen2

18 MPC controller with soft constraints

The closed-loop simulation for $T(0) = T_{init}^{(1)}$ in scen2 under mpc_controller_4 are shown in figure 14.

19 MPC controller 4 under scenario 1

The plot for $T_{init}^{(1)}$ in scen1 under controller_mpc_3 is as figure 15 and with controller_mpc_4 is as figure 16.

20 Soft-constraint MPC with estimated future disturbance

We tuned our disturbance term d and used the following piece-wise constant $\bar{d}$ as estimated future disturbance (as figure 17). With our controller and estimated disturbance, the performance is significantly better than without it.

Part VI
Offset-free MPC

21 Augment system with disturbance

The augmented system dynamics is:

$$\begin{bmatrix} T(k+1) \\ d(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ O & I \end{bmatrix} \begin{bmatrix} T(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} B \\ O \end{bmatrix} p(k)$$

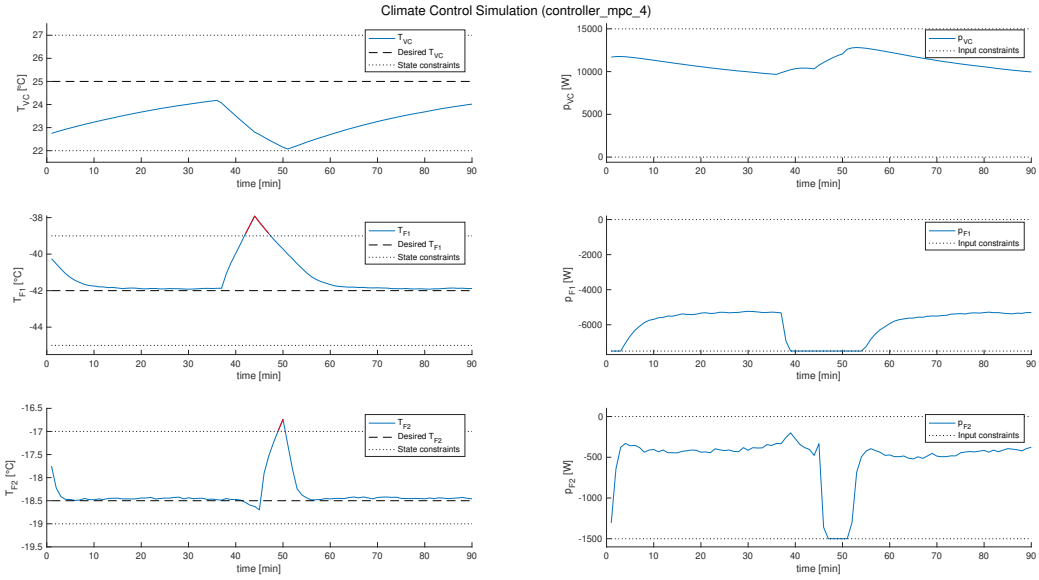


Figure 14: MPC controller 4 for T_01, scen2

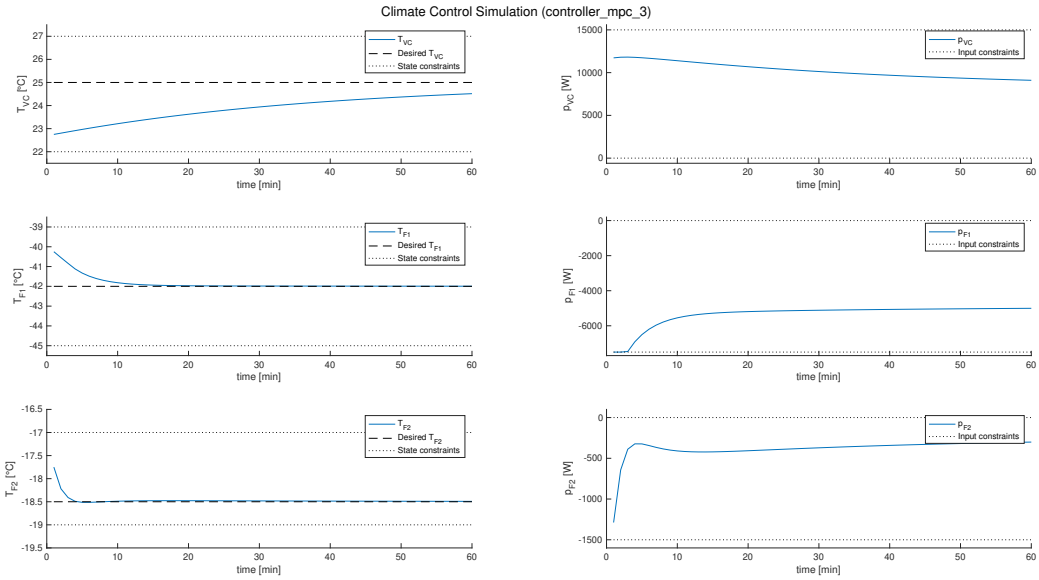


Figure 15: MPC controller 3 for T_01, scen1

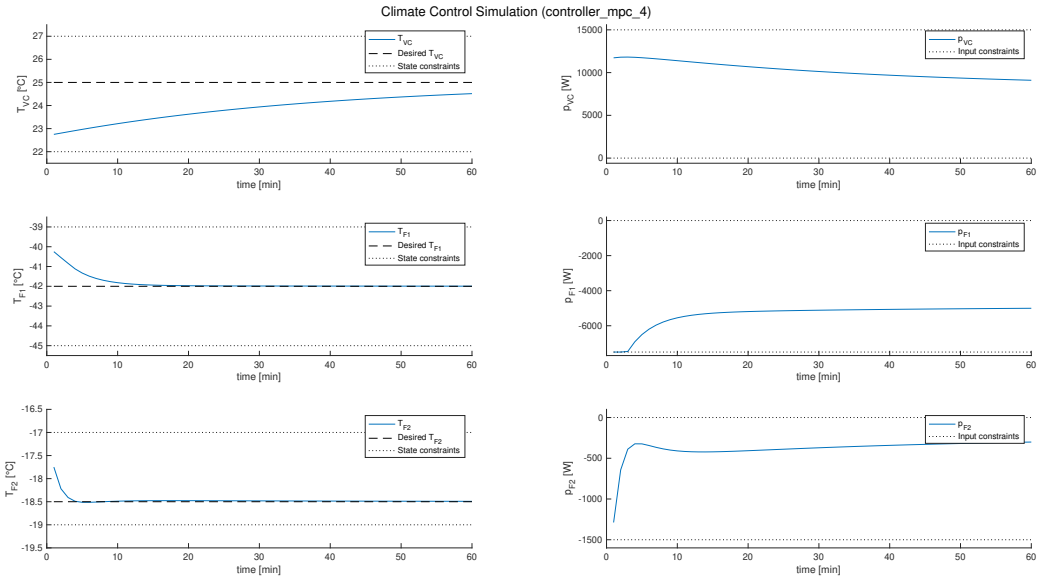


Figure 16: MPC controller 4 for T_01, scen1

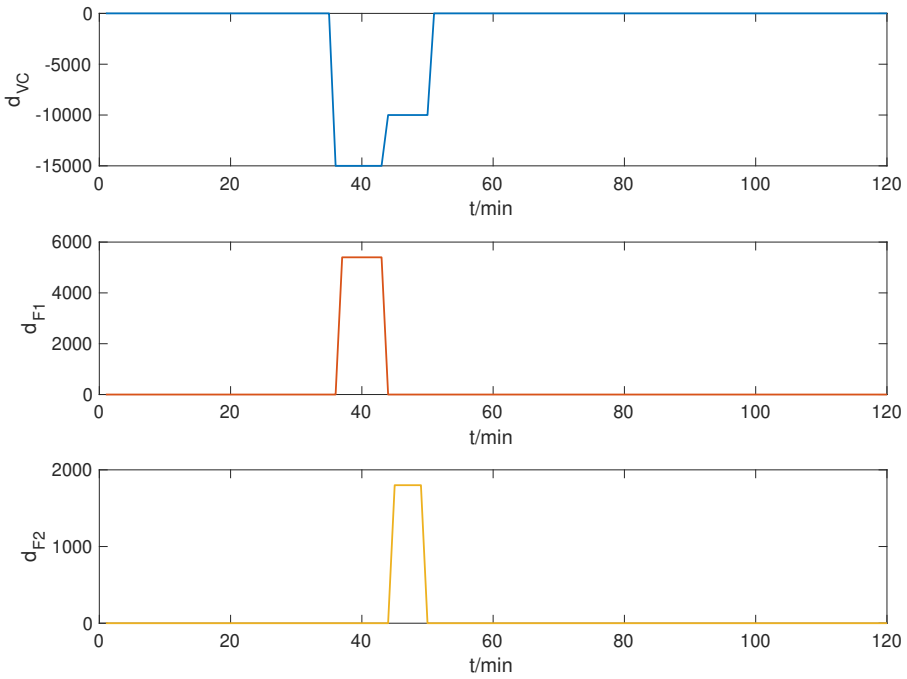


Figure 17: The estimated disturbance for secn2

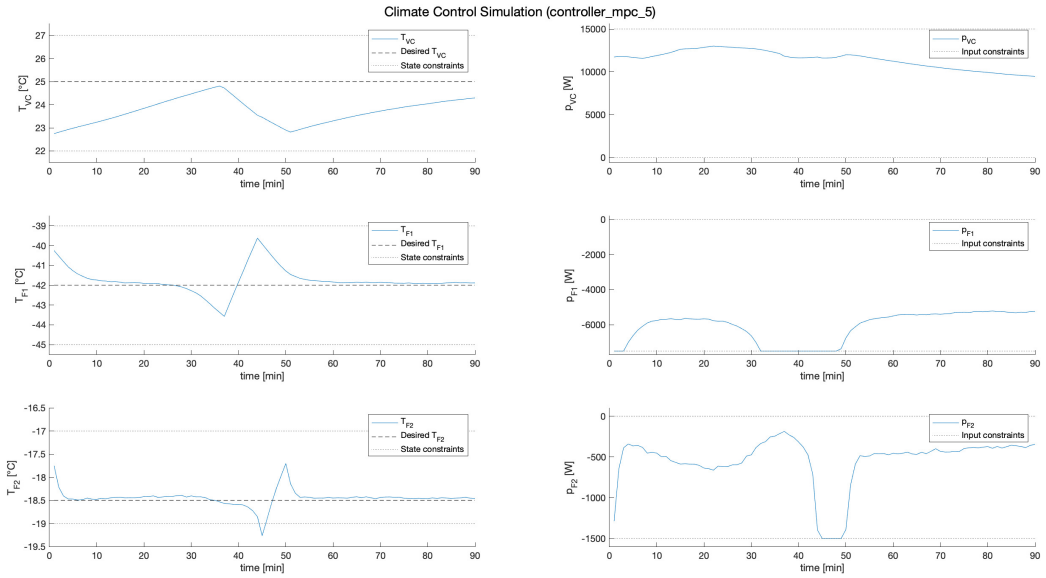


Figure 18: MPC controller 5 for T_01, scen2 with known estimated disturbance

$$y(k) = \begin{bmatrix} I & O \end{bmatrix} \begin{bmatrix} T(k) \\ d(k) \end{bmatrix}$$

Therefore, we have:

$$A_{aug} = \begin{bmatrix} A & B_d \\ O & I \end{bmatrix}, \quad B_{aug} = \begin{bmatrix} B \\ O \end{bmatrix}, \quad C_{aug} = \begin{bmatrix} I & O \end{bmatrix}, \quad D_{aug} = O$$

22 Observer design

The observer update is:

$$\begin{bmatrix} \hat{T}(k+1) \\ \hat{d}(k+1) \end{bmatrix} = A_{aug} \begin{bmatrix} \hat{T}(k) \\ \hat{d}(k) \end{bmatrix} + B_{aug} p(k) + L \left(T(k) - C_{aug} \begin{bmatrix} \hat{T}(k) \\ \hat{d}(k) \end{bmatrix} \right)$$

Observer error dynamics:

$$\begin{bmatrix} T(k+1) - \hat{T}(k+1) \\ d(k+1) - \hat{d}(k+1) \end{bmatrix} = (A_{aug} + LC_{aug}) \begin{bmatrix} T(k) - \hat{T}(k) \\ d(k) - \hat{d}(k) \end{bmatrix}$$

We need to choose L such that $A_{aug} + LC_{aug}$ has eigenvalues that are within the unit circle. Notes that $A_{aug} + LC_{aug}$ has eigenvalues P. Let $P = [0, 0, 0, 0.5, 0.5, 0.5]$. Then the observer gain matrix is:

$$L = \begin{bmatrix} -1.4948 & -0.0003214 & -0.0005714 \\ -0.0035 & -1.4860 & -0.01054 \\ -0.01846 & -0.03162 & -1.4500 \\ -35000 & 0 & 0 \\ 0 & -3250 & 0 \\ 0 & 0 & -1083 \end{bmatrix}$$

The resulting eigenvalues are verified and match with P.

Based on the estimated disturbance $\hat{d}$, the steady-state satisfies:

$$\begin{bmatrix} A - I & B \\ I & O \end{bmatrix} \begin{bmatrix} T_s \\ p_s \end{bmatrix} = \begin{bmatrix} -B_d \hat{d} \\ r \end{bmatrix}$$

23 Track computed steady state

The closed-loop simulation plots for $T(0) = T_{init}^{(1)}$ in scenario 3 is as figure 19.

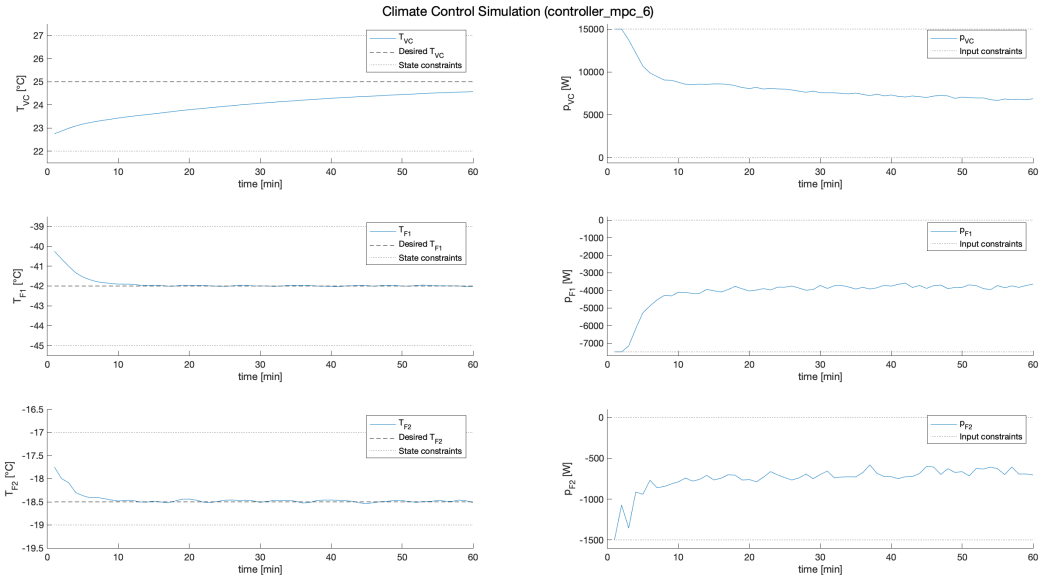


Figure 19: MPC controller 6 for T_01, scen3

Part VII
FORCES Pro

24 MPC controller using FORCES PRO

Closed-loop simulation for $T(0) = T_{init}^{(2)}$ in scenario 1 with our MPC controller implemented using FORCE Pro is shown in figure 20. The plot with controller_mpc_1 is as figure 8. From the plots we can see the system evolution with two implementation methods are identical.

Let's compare the running times for both controllers. On our computer, the controller with Yalmip takes 3.4847 seconds to solve the above MPC problem, while the one with FORCES Pro takes only 0.2462 seconds, which is only 7.07% of the former one. FORCES Pro shows great advantages in computation efficiency.

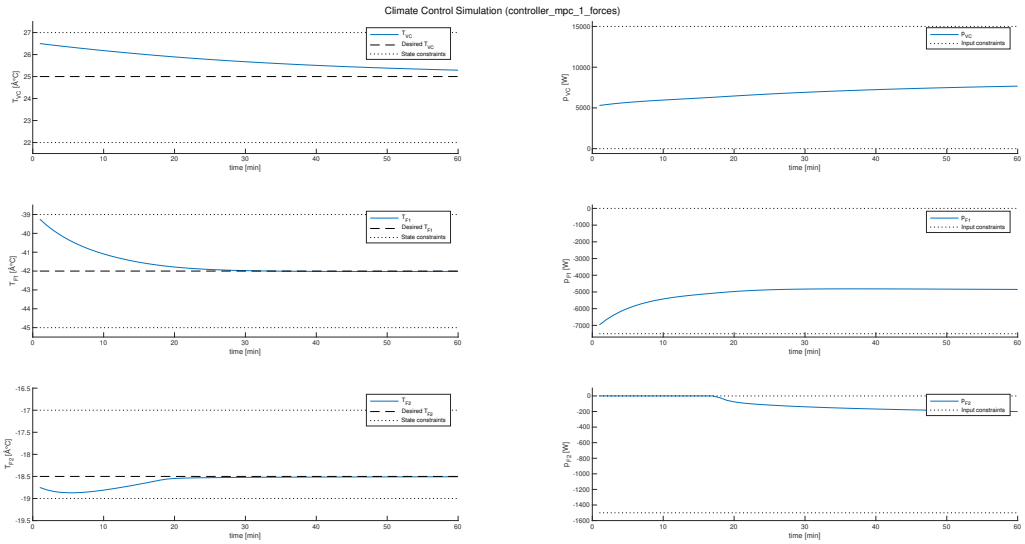


Figure 20: MPC controller with FORCES Pro for T_02, scen1

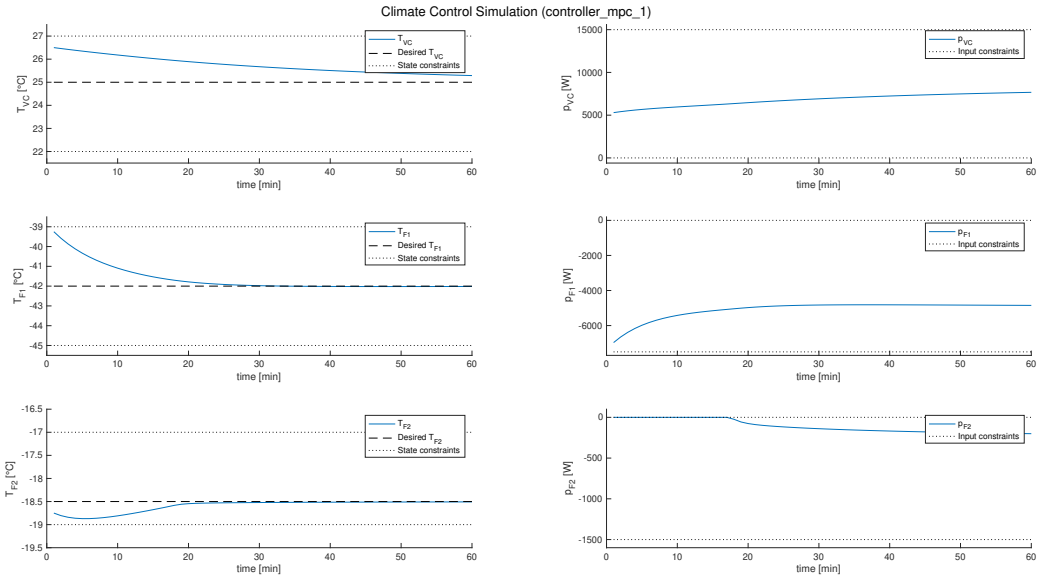


Figure 8: MPC controller 1 for T_02, scen1