MPC Project Report

Yuliang Zhong, Yueshan Li, Aobo Yang

May 2021

1 System Modeling

1.1 Question 1

According to the system dynamic equations given in the handout, we defined the continuous-time state-space description as follows:

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

where:

$$A^c = \begin{bmatrix} -\frac{\alpha_{F_1,VC}+\alpha_{F_2,VC}+\alpha_{Env,VC}}{m_{VC}} & \frac{\alpha_{F_1,VC}}{m_{VC}} & \frac{\alpha_{F_2,VC}}{m_{VC}} \\ \frac{\alpha_{F_1,VC}}{m_{F_1}} & -\frac{\alpha_{F_1,VC}+\alpha_{F_2,F_1}}{m_{F_1}} & \frac{\alpha_{F_1,F_2}}{m_{F_1}} \\ \frac{\alpha_{F_2,VC}}{m_{F_2}} & \frac{\alpha_{F_1,F_2}}{m_{F_2}} & -\frac{\alpha_{F_2,VC}+\alpha_{F_2,F_1}}{m_{F_2}} \end{bmatrix}$$

$$B^c = \begin{bmatrix} \frac{\beta_{11}}{m_{VC}} & \frac{\beta_{12}}{m_{VC}} & \frac{\beta_{13}}{m_{VC}} \\ \frac{\beta_{21}}{m_{F_1}} & \frac{\beta_{22}}{m_{F_1}} & \frac{\beta_{23}}{m_{F_1}} \\ \frac{\beta_{31}}{m_{F_2}} & \frac{\beta_{32}}{m_{F_2}} & \frac{\beta_{33}}{m_{F_2}} \end{bmatrix}$$

$$d^c = \begin{bmatrix} d_{VC} + \alpha_{Env,VC} T_{ENV} \\ d_{F_1} \\ d_{F_2} \end{bmatrix}$$

Please refer to **compute_controller_base_parameters.m** for numerical results.

1.2 Question 2

With Euler discretization, we get:

$$\begin{aligned} A &= I + T_s A^c \\ B &= T_s B^c \\ B_d &= T_s B^c_d \end{aligned}$$

Please refer to **compute_controller_base_parameters.m** for numerical results.

1.3 Question 3

The steady-state temperatures are given in the system description, which are

$$T_{sp} = \begin{bmatrix} 25 \\ -42 \\ -18.5 \end{bmatrix}$$

At steady state, we have $T(k+1) = T(k) = T_{sp}$. Therefore we can compute the steady-state control inputs p_{sp} by substituting T_{sp} into the discrete-time state-space description.

$$\begin{aligned} T_{sp} &= AT_{sp} + Bp_{sp} + B_d d \\ p_{sp} &= B^{-1}((I - A)T_{sp} - B_d d) \end{aligned}$$

Given the steady-state temperatures and control inputs, we can formulate the system in Delta-Formulation as follows,

$$\begin{bmatrix} T_1(k+1) - T_{sp} \\ T_2(k+1) - T_{sp} \\ T_3(k+1) - T_{sp} \end{bmatrix} = A \begin{bmatrix} T_1(k) - T_{sp} \\ T_2(k) - T_{sp} \\ T_3(k) - T_{sp} \end{bmatrix} + B \begin{bmatrix} p_1(k) - p_{sp} \\ p_2(k) - p_{sp} \\ p_3(k) - p_{sp} \end{bmatrix}$$

Please refer to **compute_controller_base_parameters.m** for numerical results.

1.4 Question 4

The constraints for the Delta-Formulation are given by

$$T_{min} - T_{sp} \leq T(k) - T_{sp} \leq T_{max} - T_{sp}$$
$$p_{min} - p_{sp} \leq p(k) - p_{sp} \leq p_{max} - p_{sp}$$

Please refer to **compute_controller_base_parameters.m** for numerical results.

2 Unconstrained Optimal Control

2.1 Question 5

Figure 1 shows the open-loop simulation without any control inputs. We can see that the T_{vc} kept decreasing and violated the state constraint after 35 minutes. Meanwhile, T_{F1} and T_{F2} kept increasing and violated the state constraints after 8 and 15 minutes, respectively.

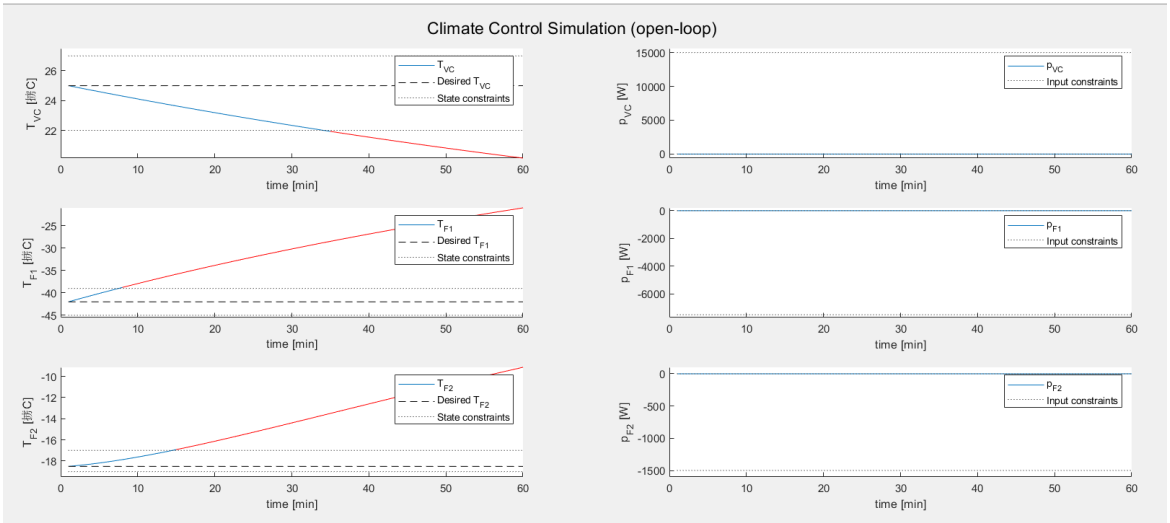

Figure 1: Open loop simulation without control inputs

2.2 Question 6

Check our solution to task 6 in **heuristic_LQR_tuning.m**. Figure 2 shows the tuning results. We can see that only three Q matrices lead to state constraint violation. Most Q matrices that cause input constraint violation form a blue lower bound in the figure. To satisfy the energy constraint and minimize the relative deviation from the steady-state, we choose Q as follows:

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

2.3 Question 7

Figure 3 shows the closed-loop simulation plot of the system starting from the initial state $T0^{(1)}$. We applied two different LQR controllers with different Q to the system. Performance of the controller with the best Q mentioned above is shown in the blue line, while the red line is the performance of another $Q' = 10^6 * I_3$. By changing Q, the time cost to reach the equilibrium increases.

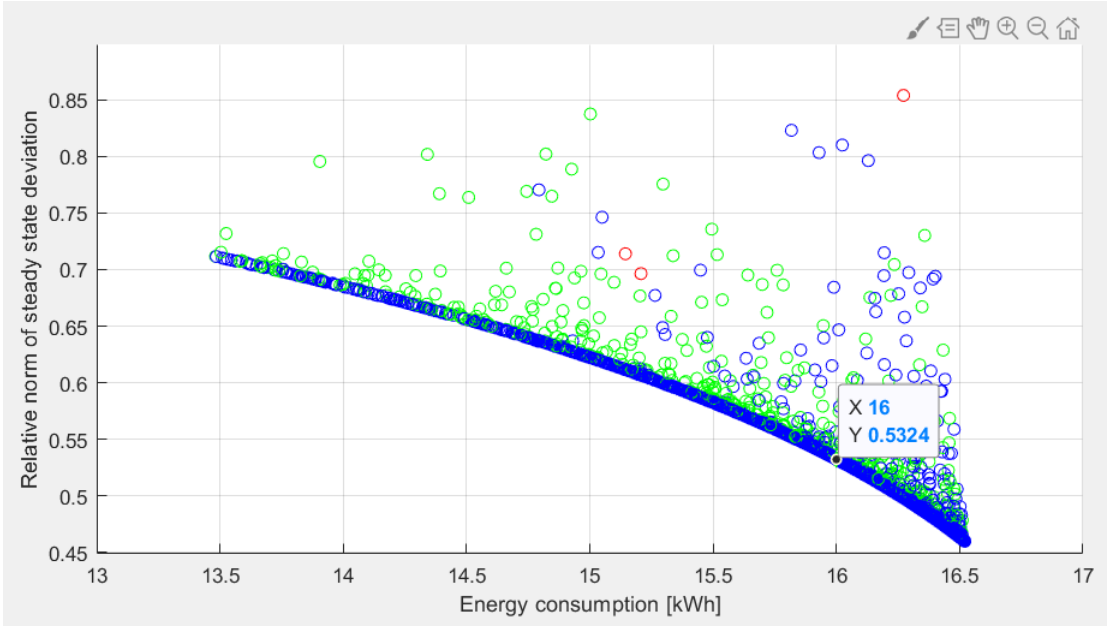

Figure 2: Heuristic LQR tuning result

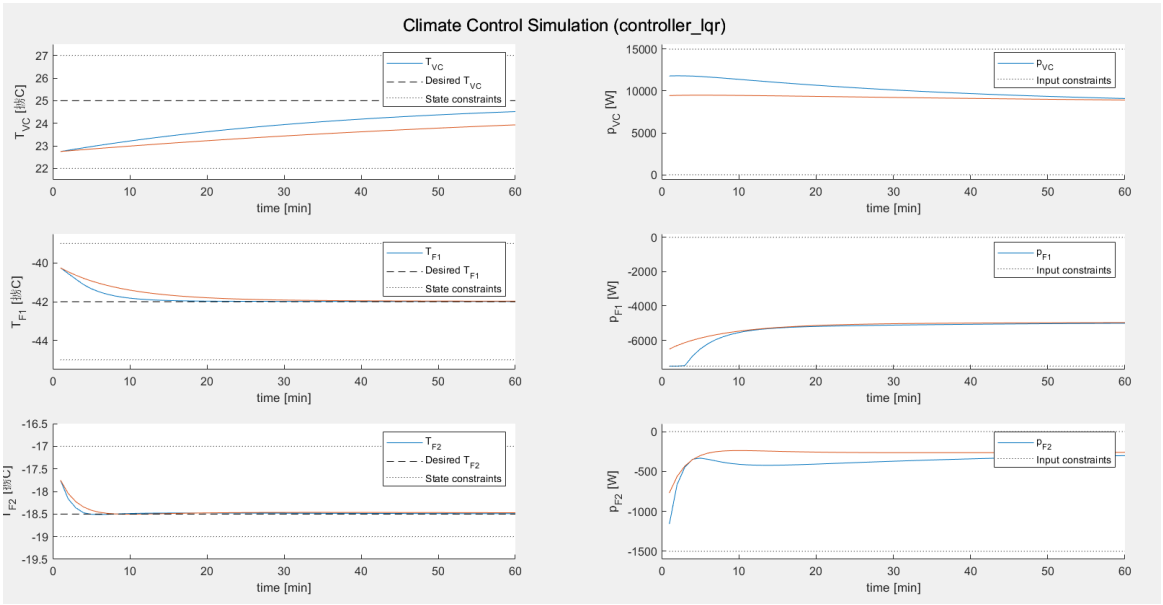

Figure 3: LQR controller from initial condition $T0^{(1)}$ (different Q)

2.4 Question 8

Figure 4 shows the closed-loop simulation plot of the system under LQR controller, starting from initial state $T0^{(2)}$. The state constraint of T_{F2} is violated from 4 to 12 minutes, because the LQR controller fails to take the state and input constraints into account.

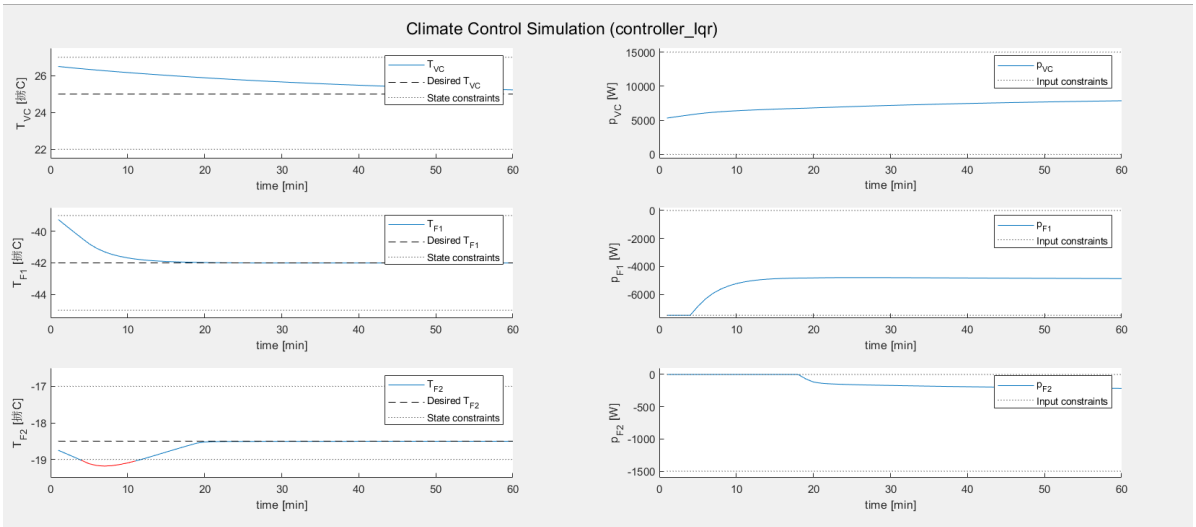

Figure 4: LQR controller from initial condition $T0^{(2)}$

3 From LQR to Model Predictive Control

3.1 Question 9

Figure 5 and 6 show the invariant set X_{LQR} of our LQR controller. All closed-loop control trajectories start from X_{LQR} do not violate the state and input constraints. Figure 6 also shows the positions of $T0^{(1)}$ and $T0^{(2)}$ with respect to X_{LQR}. Both of the two initial conditions are outside of X_{LQR}. In the heuristic LQR tuning algorithm, we ignore input constraints violations during the optimal Q selection. Therefore, even though the state constraints are satisfied throughout the simulation for $T0^{(1)}$, the input constraints are violated, resulting in the initial state outside the invariant set. On the other hand, as shown in Figure 4, starting from $T0^{(2)}$ leads to state constraints. So $T0^{(2)}$ is also not contained in X_{LQR}.

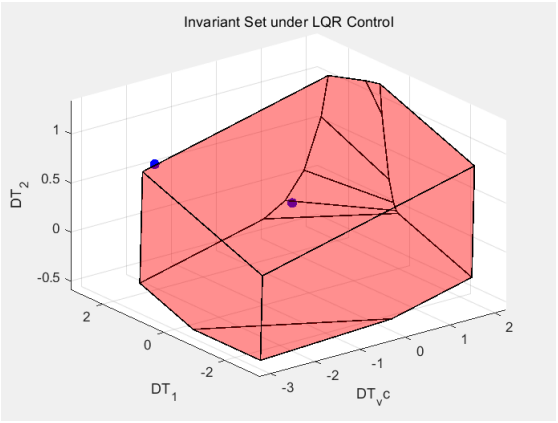

Figure 5: Invariant set (view 1)

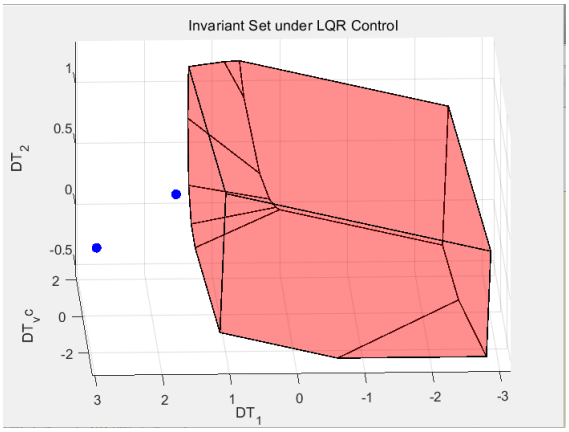

Figure 6: Invariant set (view 2)

3.2 Question 10

The infinite horizon cost under the LQR control law is given by

$$J^{\infty}_{LQR}(x(0)) = x(0)^T P_{\infty} x(0)$$

where P_∞ is the positive definite solution of Discrete Algebraic Riccati Equation.

3.3 Question 11

Figure 7 shows the closed-loop trajectories under the MPC controller. The blue line starts from initial state $T0^{(1)}$ while the red line starts from initial state $T0^{(2)}$. Compared with task 7 and 8, the state constraint is not violated anymore when we start from $T0^{(2)}$. This is because the state and input constraints are considered in MPC controller during optimization.

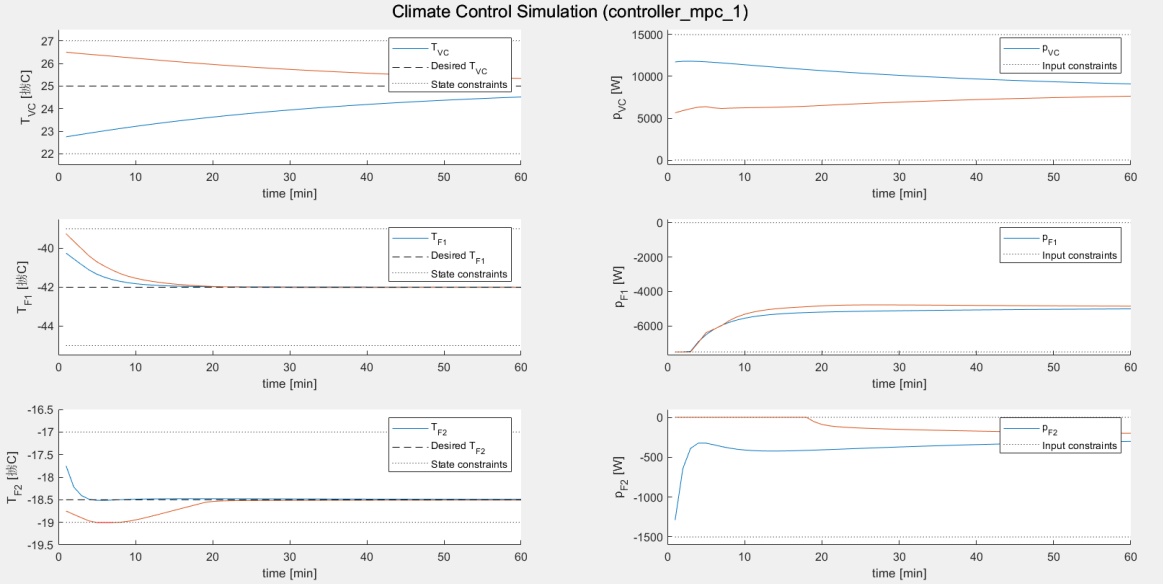

Figure 7: MPC controller 1 from initial condition T01 (blue), T02 (red)

4 MPC with theoretical closed-loop guarantees

4.1 Question 12

Proof: The cost function $J(x(k))$ of the MPC controller in given by

$$J(x(k)) = \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i$$

Assuming u_i^* and x_i^* are the optimal control input and its corresponding state at time i, we get

$$\begin{aligned} J^*(x(k+1)) &\leq J(x(\tilde{k}+1)) \\ &= \sum_{i=1}^{N} {x_i^*}^T Q x_i^* + {u_i^*}^T R u_i^* \\ &= \sum_{i=0}^{N-1} {x_i^*}^T Q x_i^* + {u_i^*}^T R u_i^* + {x_N^*}^T Q x_N^* + {u_N^*}^T R u_N^* - {x_0^*}^T Q x_0^* + {u_0^*}^T R u_0^* \\ &\leq J^*(x(k) \end{aligned}$$

Also notice that $J^*(x(k)) = 0$ when $x(k) = 0$ and $J^*(x(k)) > 0$ for any other $x(k)$. Therefore, the optimal cost function $J^*(x(k))$ is a Lyapunov function. According to the theorem of Lyapunov stability, $x = 0$ is a asymptotically stable equilibrium point for the system.

4.2 Question 13

Figure 8 shows the closed-loop trajectories under the MPC controller, with terminal state constraint $x_N = 0$. The blue line starts from initial state $T0^{(1)}$ and the red line starts from initial state $T0^{(2)}$. The state constraint of T_{F2} is violated from 4 to 12 minutes.

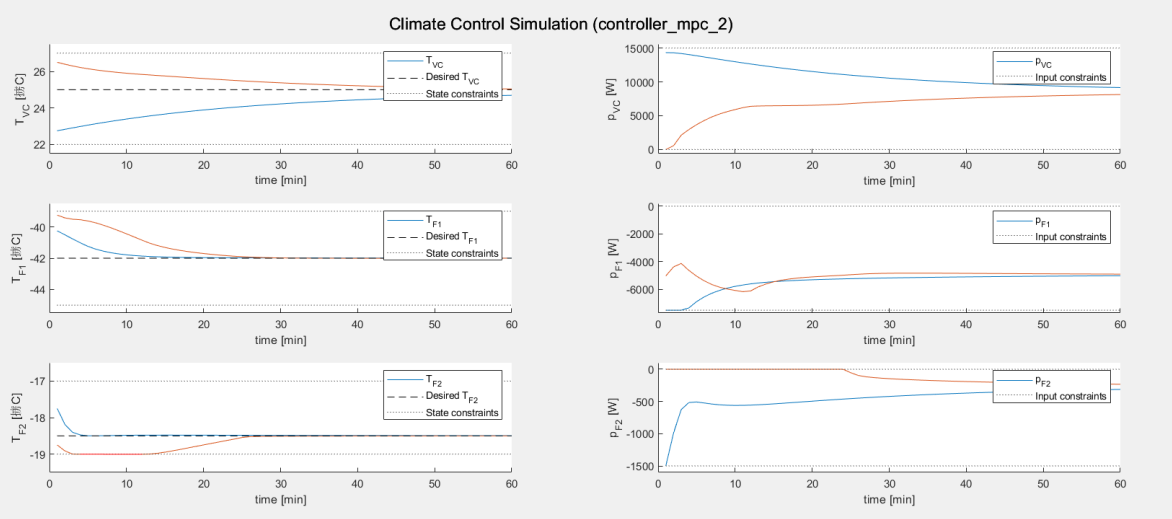

Figure 8: MPC controller 2 from initial condition T01 (blue), T02 (red)

4.3 Question 14

Figure 9 shows the closed-loop trajectories under MPC controller, with $x_N \in X_{LQR}$, which is computed in task 9. The blue line starts from initial state $T0^{(1)}$ and the red line starts from initial state $T0^{(2)}$.

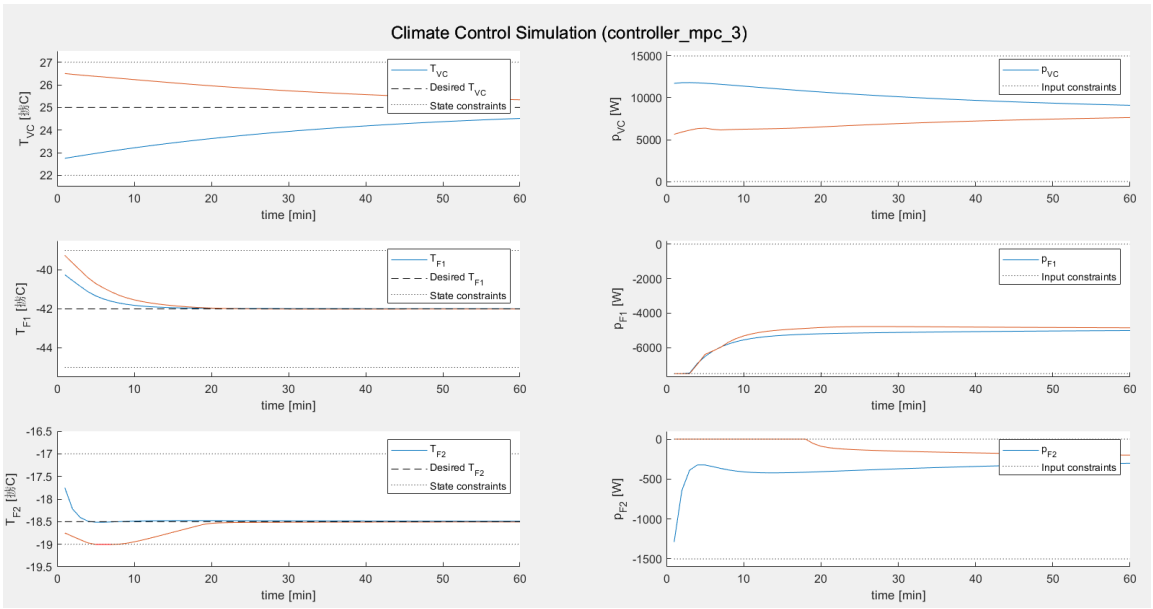

Figure 9: MPC controller 3 from initial condition T01 (blue), T02 (red)

4.4 Question 15

The optimization costs J_{MPC} for three MPC controllers from initial state $T0^{(1)}$ and $T0^{(2)}$ are shown below. From initial condition $T0^{(1)}$, MPC2 performed more aggressively than MPC1 and regulated the states more quickly to the steady-state. This is because MPC2 has more strict state constraints and naturally, this leads to a higher optimization cost. When the constraint on terminal state was relaxed in MPC3, the controller performed exactly the same as MPC1 did, resulting in same optimization costs and trajectories. From initial condition $T0^{(2)}$, MPC1 and MPC3 still performed similarly and gave the same optimization costs. However, MPC2 caused state constraints violation from 4 to 12 minutes, as it failed to find a feasible solution in the limited feasible sets.

	MPC1	MPC2	MPC3
$T0_1$	8604309929	9320036983	8604309929
$T0_2$	4772072812	4597281672	4772072812

4.5 Question 16

The intersection $\chi_f \cap X_{LQR}$ is an invariant set under the LQR controller. Assuming $x(0)$ is from $\chi_f \cap X_{LQR}$, then we have $x(0) \in \chi_f$ and $x(0) \in X_{LQR}$. Since both of the sets are invariant under the LQR controller, for any time step $k > 0$, we have $x(k) \in \chi_f$ and $x(k) \in X_{LQR}$. Therefore, $x(k) \in \chi_f \cap X_{LQR}$ and it is an invariant set under the LQR controller.
According to the lecture slides, using $\chi_f \cap X_{LQR}$ as terminal set yield an asymptotically stable MPC controller, given the following assumptions are satisfied: the stage cost is positive definite; all state and input constraints are satisfied and terminal cost is a continuous Lyapunov function.

5 Soft Constraints

5.1 Question 17

Figure 10 shows the closed-loop trajectory under MPC controller implemented in task14 from initial condition $T0^{(1)}$ under Scenario2. The controller stops functioning from about 40 minutes, because it can't find a feasible solution under the influence of unmodeled disturbance.

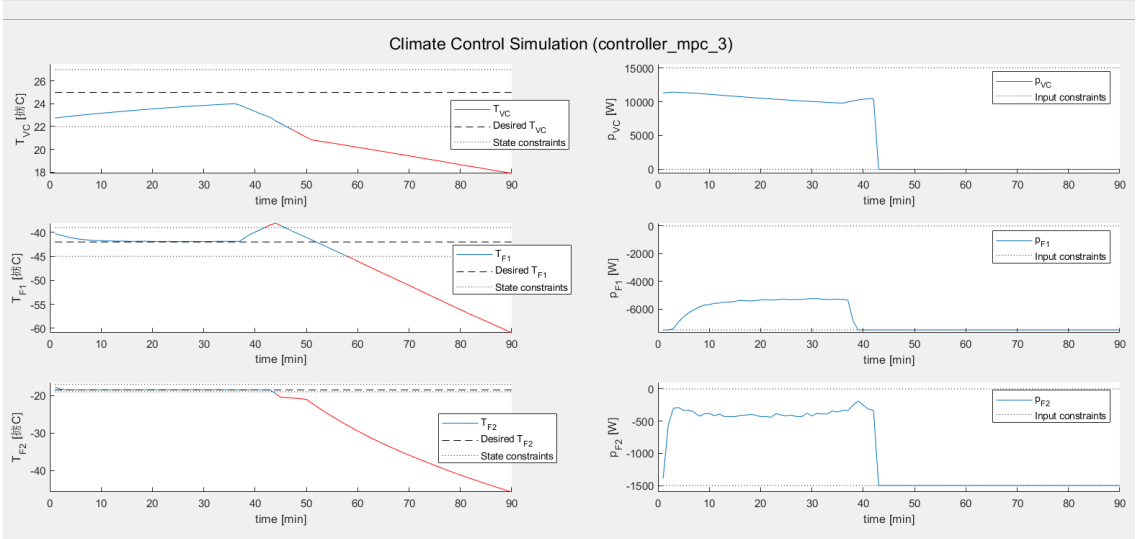

Figure 10: MPC controller 3 from initial condition T01 under Scen2

5.2 Question 18

We adds slack variables to the state variables during optimization and give quadratic and linear penalty to the slack variables. We set the quadratic penalty matrix to be $S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and the linear penalty $v = 1$.
Theoretically, the choice of penalty matrices effects the violation size and duration of the system. In our system, however, it has little effect because the control inputs reach the maximum no matter what penalty is given.
Figure 11 shows the closed-loop trajectory under MPC controller with soft-constraint under Scenario2 from initial condition $T0^{(1)}$. With soft-constraints, the MPC controller is able to find a feasible solution under unmodeled disturbance.

5.3 Question 19

Figure 12 shows the closed-loop trajectories under two MPC controllers, one with soft-constraints and one without. The blue line represents MPC3, which doesn't contain soft-constraints and the red line represents MPC4 with soft-constraints. The performances of the two controllers from initial state T01 under scenario1 are exactly the same, as the trajectory of MPC4 completely covers that of MPC3.

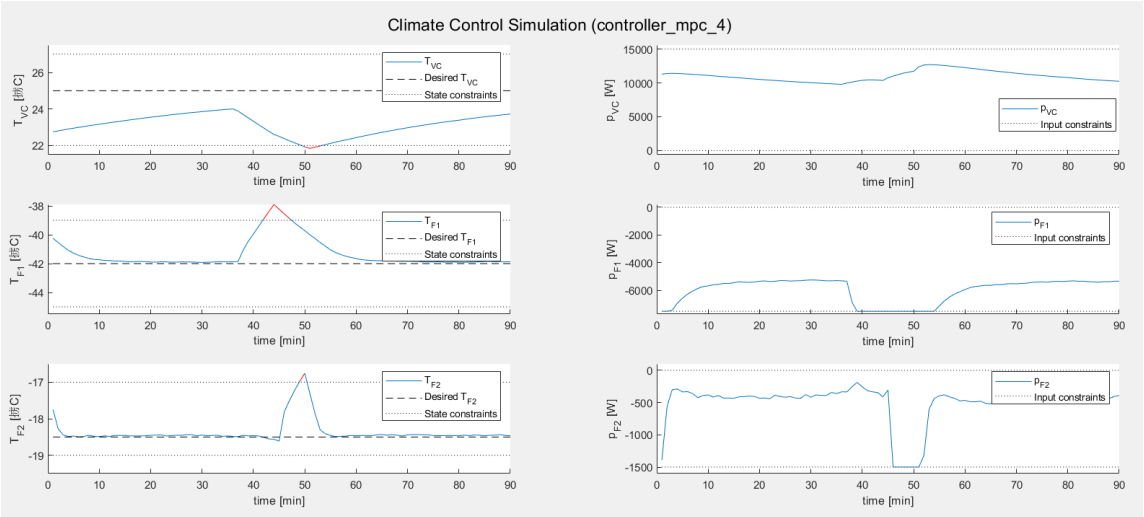

Figure 11: MPC controller 4 from initial condition T01 under Scen2

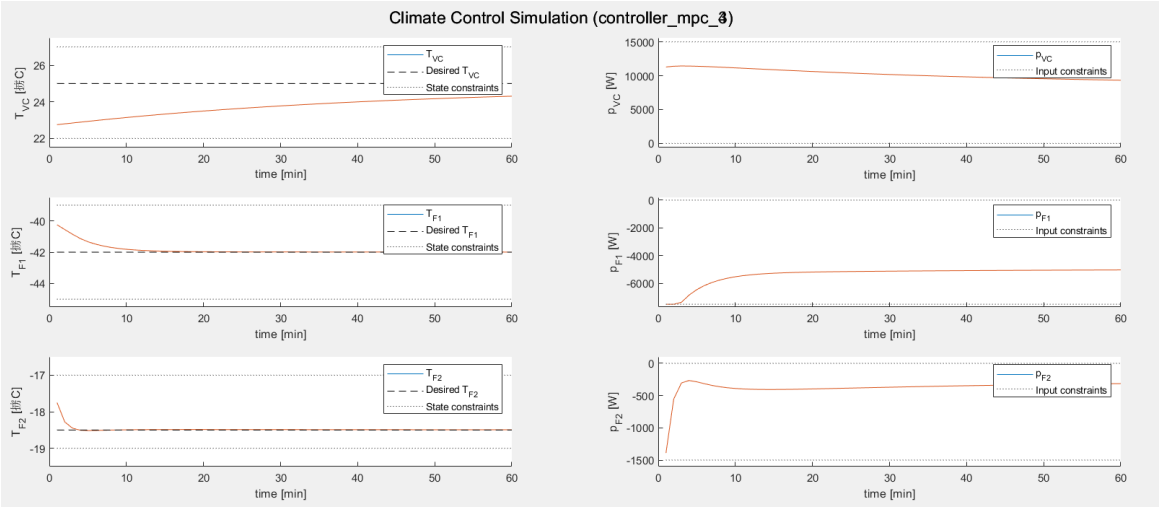

Figure 12: MPC controller 3 and 4 from initial condition T01 under Scen1

5.4 Question 20

By observing the occurring disturbances under scenario 2, we use three rectangular functions as the approximation for future disturbances on the three room temperatures. Specifically, we set $d_{VC} = -10^4$ from time step 36 to 50, $d_{F1} = 5 * 10^3$ from time step 37 to 43, $d_{F2} = 1.9 * 10^3$ from time step 45 to 49, and 0 for else elements.
Figure 13 shows the closed-loop trajectory under MPC controller with expected future disturbances under Scenario2 from initial condition $T0^{(1)}$. When the disturbance expectation is accurate enough, the MPC controller is able to satisfy the state constraints throughout the simulation under the temperature disturbances.

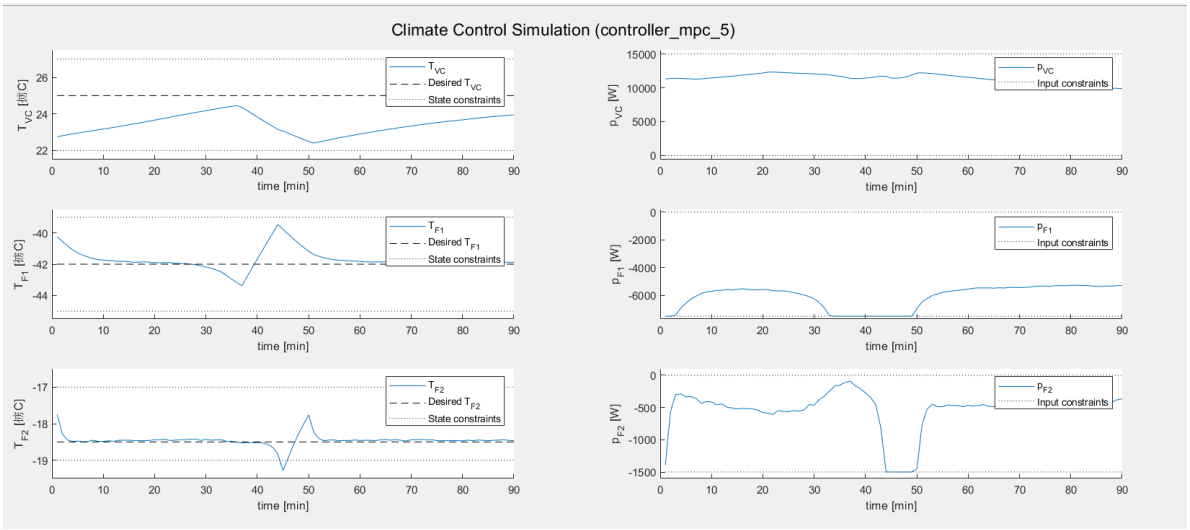

Figure 13: MPC controller 5 from initial condition T01 under Scen1

6 Offset-free MPC

6.1 Question 21

The matrices for augmented discrete-time system are,

$$A_{aug} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix}$$
$$B_{aug} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$
$$C_{aug} = \begin{bmatrix} I & 0 \end{bmatrix}$$
$$D_{aug} = 0$$

6.2 Question 22

Assuming the estimated disturbance is $\hat{d}$, the observer steady-state x_s and u_s satisfies

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} -B_d \hat{d} \\ r \end{bmatrix}$$

The estimator error dynamics is given by

$$e(k+1) = (A_{aug} - LC_{aug})e(k)$$

We choose $L = [L_x^T, L_d^T]^T$ such that the error dynamics are stable and converge to zero. That means the estimator is asymptotically stable. So we have $|\lambda_i| < 1$, for $\forall \lambda_i \in SPEC[A_{aug} - LC_{aug}]$. We tune L by pole placement and choose pole randomly by function rand(). $poles = [0.159; 0.439; 0.629; 0.925; 0.357; 0.819]$ works well.

6.3 Question 23

The offset-free MPC formulation is basically the same as MPC3, the MPC controller with terminal state $x(n) \in X_{LQR}$. The only difference is that the steady-state T_{sp} and p_{sp} we use in the Delta-Formulation is not constant anymore. We need to update it at every time step k using the estimated temperature $\hat{T(k)}$and disturbance $\hat{d(k)}$. The controller takes $\hat{T(k)} - T_{sp}$ as the initial condition and takes $\hat{d(k)}$, T_{sp} and p_{sp} as parameters for the optimization.
Figure 14 shows the closed-loop trajectories under MPC3 and the offset-free MPC controller(MPC6) from initial condition $T0^{(1)}$ under Scenario3. The blue line represents the performance of MPC3 and the red line represents MPC6. Due to the unmodeled disturbance, we observe a constant offset from the trajectory of MPC3. By contrast, MPC6 could successfully track the target despite the disturbances.

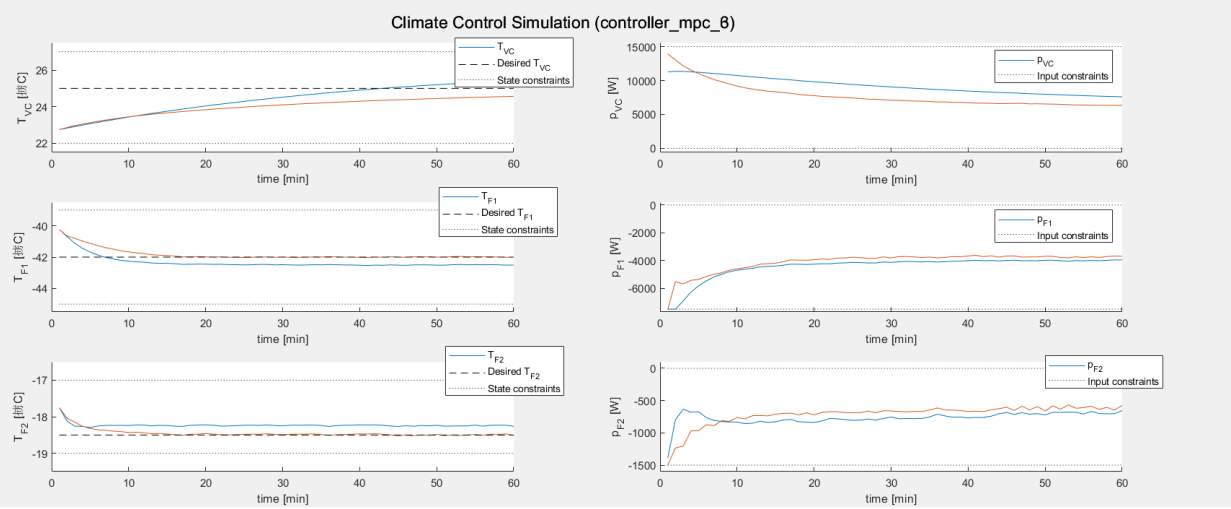

Figure 14: MPC controller 3(blue) and 6(red) from initial condition T01 under Scen3

7 FORCES Pro

7.1 Question 24

We put the authorized FORCES Pro client in our workspace and added relevant path. Then switch yalmip_optimizer to force_optimizer. The FORCES Pro client upload code to cloud and download results when initialization. Then if we do not clear controller and other variables, the $t\_sim\_forces = 0.43998s$, and $t\_sim = 1.2378s$. As the figure 15 shows, two controllers get the exact same trajectory. Such a short time cost makes it possible for real-world autonomous locomotive robots to move with MPC controllers.

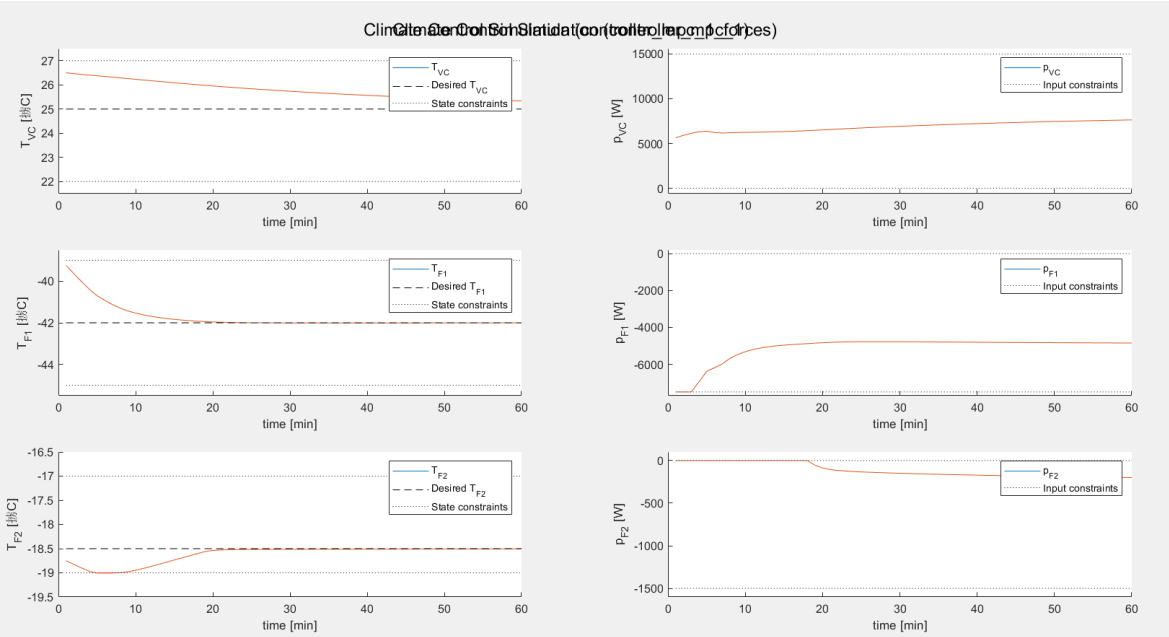

Figure 15: FORCES Pro and Yalmip