Climate Control

Chao Ni

April 2020

Question 1

$$A^c = \begin{bmatrix} \frac{-\alpha_{1,2}-\alpha_{1,o}}{m_1} & \frac{\alpha_{1,2}}{m_1} & 0 \\ \frac{\alpha_{1,2}}{m_2} & \frac{-\alpha_{1,2}-\alpha_{2,3}-\alpha_{2,o}}{m_2} & \frac{\alpha_{2,3}}{m_2} \\ 0 & \frac{\alpha_{2,3}}{m_3} & \frac{-\alpha_{2,3}-\alpha_{3,o}}{m_3} \end{bmatrix} = 10^{-3} \begin{bmatrix} -0.4250 & 0.3750 & 0 \\ 0.1875 & -0.9375 & 0.5000 \\ 0 & 0.4000 & -1.000 \end{bmatrix} \tag{1}$$

$$B^c = \begin{bmatrix} \frac{1}{m_1} & 0 \\ 0 & \frac{1}{m_2} \\ 0 & 0 \end{bmatrix} = 10^{-5} \begin{bmatrix} 0.5000 & 0 \\ 0 & 0.2500 \\ 0 & 0 \end{bmatrix} \tag{2}$$

$$d^c = \begin{bmatrix} w_1 + \alpha_{1,o} T_o \\ w_2 + \alpha_{2,o} T_o \\ w_3 + \alpha_{3,o} T_o \end{bmatrix} = \begin{bmatrix} 120 \\ 1200 \\ 3600 \end{bmatrix} \tag{3}$$

Question 2

Use exact discretization and MATLAB c2d function:

$$A = e^{A^c T_s} = \begin{bmatrix} 0.9749 & 0.0216 & 0.0003 \\ 0.0108 & 0.9458 & 0.0283 \\ 0.0001 & 0.0226 & 0.9421 \end{bmatrix} \tag{4}$$

$$B = \int_0^{T_s} e^{A^c (T_s - \tau) B^c \tau} = 10^{-3} \begin{bmatrix} 0.2962 & 0.0016 \\ 0.0016 & 0.1459 \\ 0.0000 & 0.0017 \end{bmatrix} \tag{5}$$

$$B_d = \int_0^{T_s} e^{A^c (T_s - \tau) B_d^c \tau} = 10^{-3} \begin{bmatrix} 0.2962 & 0.0016 & 0.0000 \\ 0.0016 & 0.1459 & 0.0017 \\ 0.0000 & 0.0017 & 0.1165 \end{bmatrix} \tag{6}$$

Question 3

Set $\dot{T} = 0$ and plug in the desired temperature T_1 and T_2 as well as all other coefficients, solving the differential equation leads to the steady-state $T_{sp} = [-21; 0.3; 7.32]$ and $p_{sp} = [-1927.5; -976.5]$.
Define $\Delta x(k) = T(k) - T_{sp}$ and $\Delta u(k) = p(k) - p_{sp}$, and formulate the Delta-Formulation.

$$\Delta x(k+1) = A \Delta x(k) + B \Delta u(k) \tag{7}$$

apply the system with input $p(k) = p_{sp} + \Delta u(k)$.

Question 4

$$Ucons = Pcons - p_{sp} \tag{8}$$

$$Xcons = Tcons - T_{sp} \tag{9}$$

Question 5

We design the LQR controller by setting

$$Q = \begin{bmatrix} 500 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

and

$$R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.001 \end{bmatrix} \tag{11}$$

The resulting plot is shown as Fig (1). All input and state constraints are satisfied, and $||T_sp - T(30)|| = 0.406 < 0.2||x_0^{(1)}|| = 0.632$.

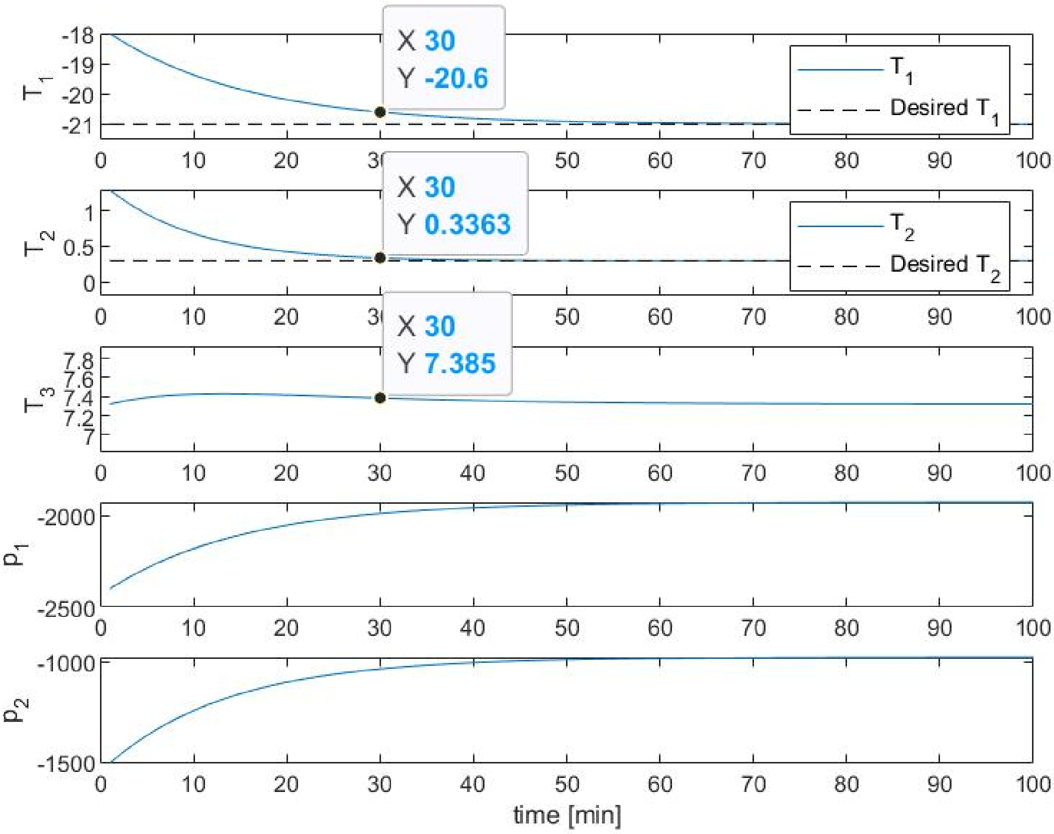


Figure 1: closed-simulation plot with $x_0 = (3, 1, 0)'$

Question 6

$$J_{LQR}^{\infty}(x(0)) = x(0)^T P_\infty x(0) \tag{12}$$

With the chosen controller designer, the corresponding matrix solved by Algebraic Riccati equation (ARE).

$$P_\infty = 10^3 \begin{bmatrix} 5.4131 & 0.8696 & 0.1811 \\ 0.8696 & 1.1151 & 0.2665 \\ 0.1811 & 0.2665 & 0.1176 \end{bmatrix} \tag{13}$$

with eigenvalues $10^3[0.0505, 1.0032, 5.5920]$.

Question 7

After change of initial point, the state constraints are violated during certain period as shown in Fig (2). Because LQR controller doesn't consider the feedback, therefore it is possible the state constraints are violated.

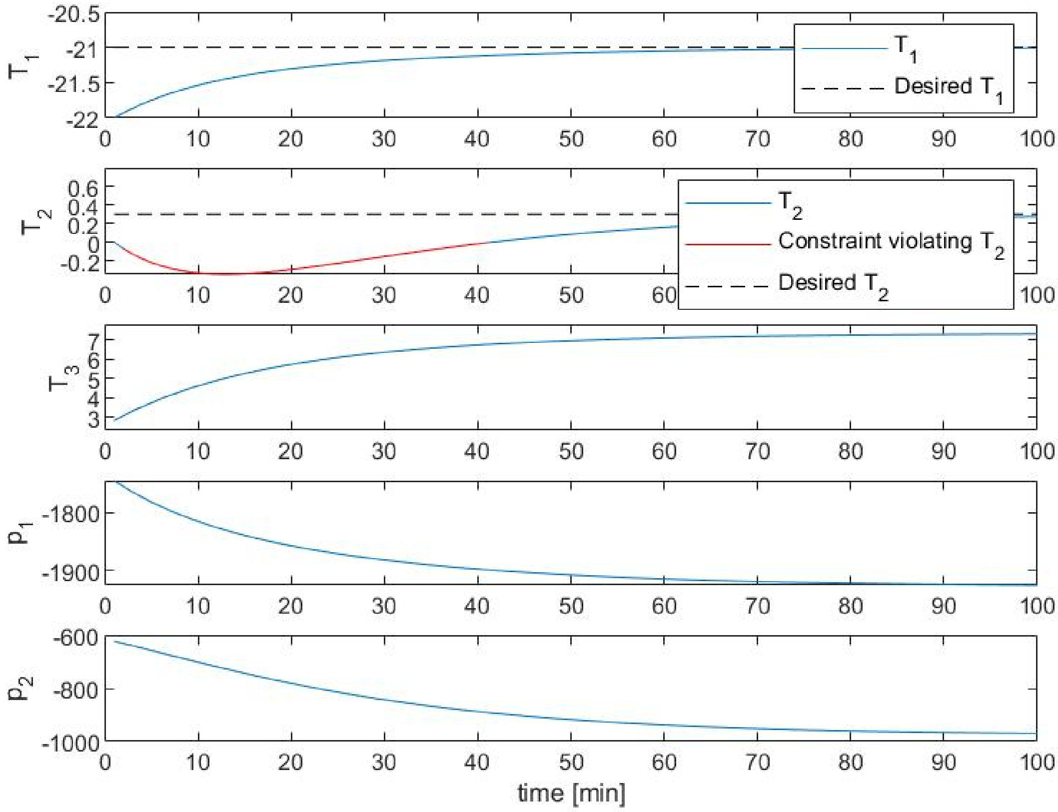


Figure 2: closed-simulation plot with $x_0 = (-1, -0.3, -4.5)'$

Question 8

Because we need to provide a polyhedron form of the invariant set, we need to compute how many steps we are going ahead (therefore providing the number of constraints). So every time we increase the horizon by 1, we will compare the new polytope and the old one. If it is same, then our set converges to the invariant set. As shown in Fig. 3 the polytope is the feasible initialization set for my controller. The blue point is the infeasible initialization point $(-1, -0.3, -4.5)$, which is clearly out of the polytope.

Question 9

For the first initial point, because with LQR controller the state and input constraints are always satisfied and the terminal cost defined in MPC controller is just the same as the cost to go in time step N in the LQR controller. So these two solutions match exactly. For the second initial point, because with LQR the state constraints are violated some time, and MPC controller helps with this issue and provides a feasible solution. The results are shown in Fig (4 and 5)

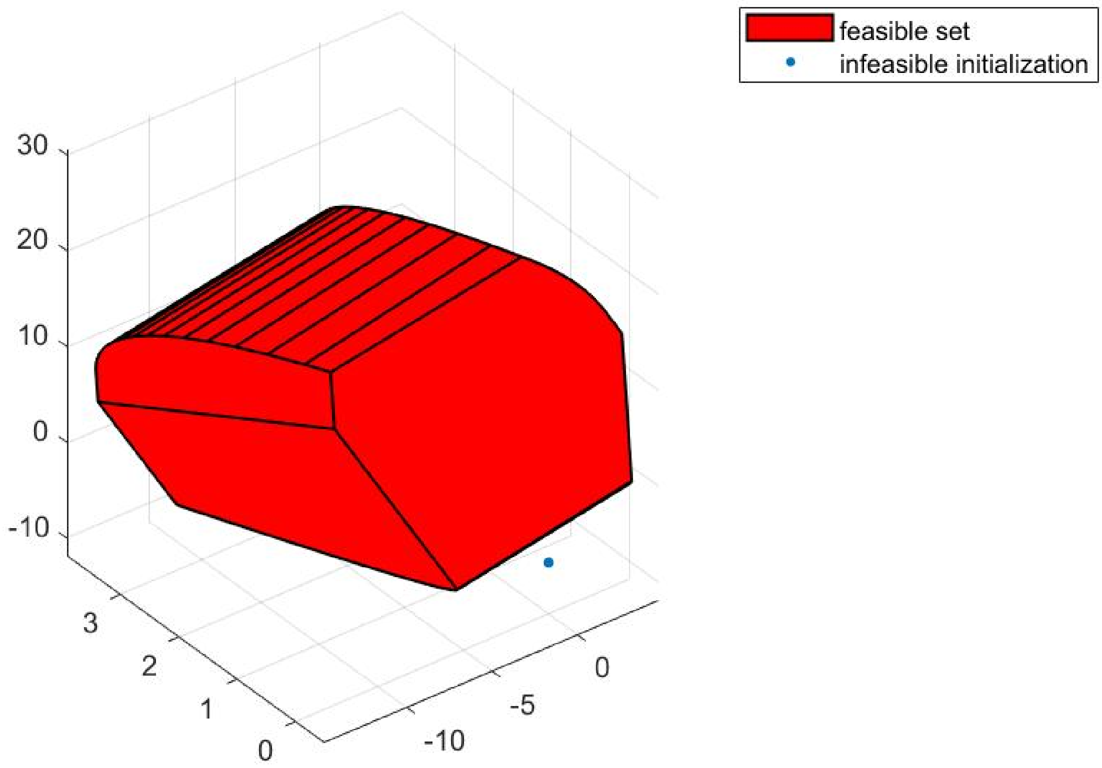
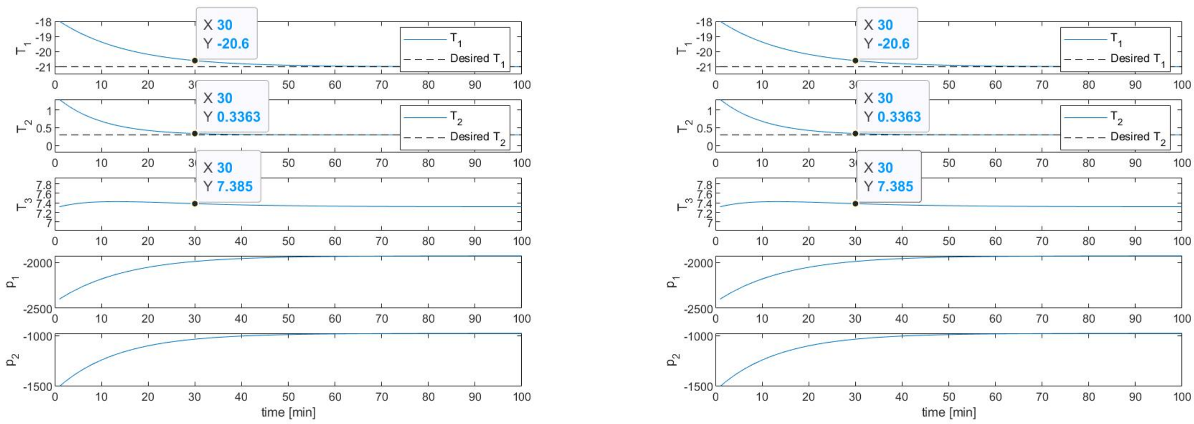


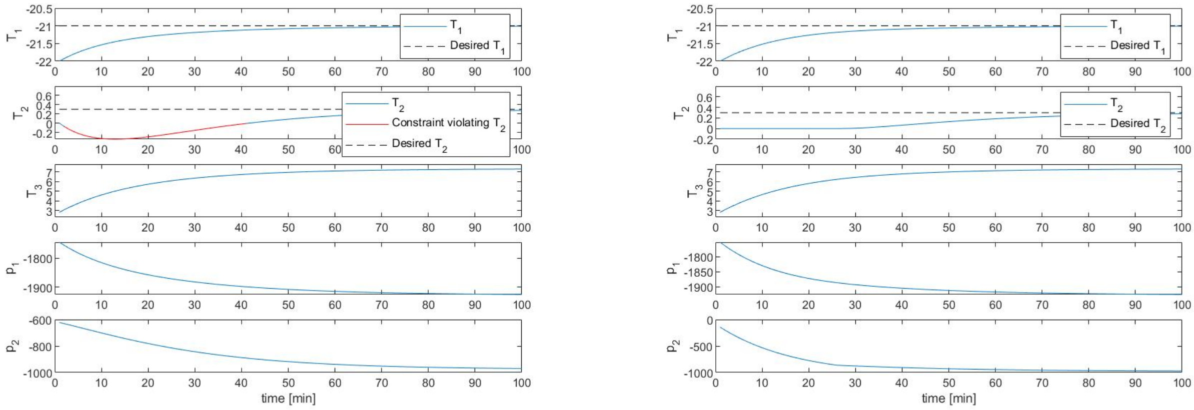
Figure 3: Feasible initialization set: X_{LQR}



(a) Plot with LQR controller

(b) Plot with MPC1 controller

Figure 4: With $x_0 = (3, 1, 0)'$



(a) Plot with LQR controller

(b) Plot with MPC1 controller

Figure 5: With $x_0 = (-1, -0.3, -4.5)'$

Question 10

We prove $J^{\infty}_{MPC1}(x(0)) = J^{\infty}_{LQR}(x(0))$ by showing that the optimal controller derived by MPC1 problem is exactly the LQR controller and showing that $J_{MPC1}(x(0)) = J^{\infty}_{MPC1}(x(0))$ and $J^{\infty}_{LQR}(x(0)) = J_{MPC1}(x(0))$.

We rewrite MPC_1 as

$$J_{MPC1}(x(k)) \tag{14}$$

$$= \sum_{i=1}^{29} x(i)^T Q x(i) + u(i)^T R u(i) + \tag{15}$$

$$\sum_{i=30}^{\infty} x_{LQR}(i)^T Q x_{LQR}(i) + u_{LQR}(i)^T R u_{LQR}(i) \tag{16}$$

Compare it with J^{∞}_{LQR} we know they have same structure. For LQR problem, there is no constraints, but for MPC problem, there are constraints. Therefore, we have $J_{MPC_1}(x(0)) \geq J^{\infty}_{LQR}(x(0))$. Or more strongly, for any controller q, we have $J_q(x(0)) \geq J^{\infty}_{LQR}(x(0))$

On the other hand, Plugging in the LQR controller K in MPC problem, because $x(0) \in X_{LQR}$, we have $x_{K,MPC}(N) = x_{LQR}(N)$, and by definition of I_f, we have $I_f(x_{K,MPC}(N)) = \sum_{i=N}^{\infty} x_K(i)^T Q x_K(i) + u_K(i)^T R u_K(i)$, therefore, LQR controller is feasible for MPC initialized with $x(0)$ and the corresponding cost $J_{K,MPC1}(x(0)) = J^{\infty}_{LQR}(x(0))$.

Moreover, from knowledge of LQR we know the optimal solution of LQR is unique and is given by the controller K where

$$K = -(B^T P_{\infty} B + R)^{-1} B^T P_{\infty} A \tag{17}$$

if R is positive definite, which is the case in our problem. Therefore, we conclude that the optimal controller of MPC problem is also given by this LQR controller.

Thus, we plug in the LQR controller K to J^{∞}_{MPC1} and by definition of I_f we know $J_{K,MPC1}(x(0)) = J^{\infty}_{K,MPC}(x(0))$.

Question 11

Given $x(0)$ is feasible, we assume the corresponding optimal control sequences are $U^* = \{u^*_0, ..., u^*_{N-1}\}$ and the state sequences are $X^* = \{x^*_0, ...x^*_{N-1}, x^*_N = 0\}$. We then propose a feasible candidate control

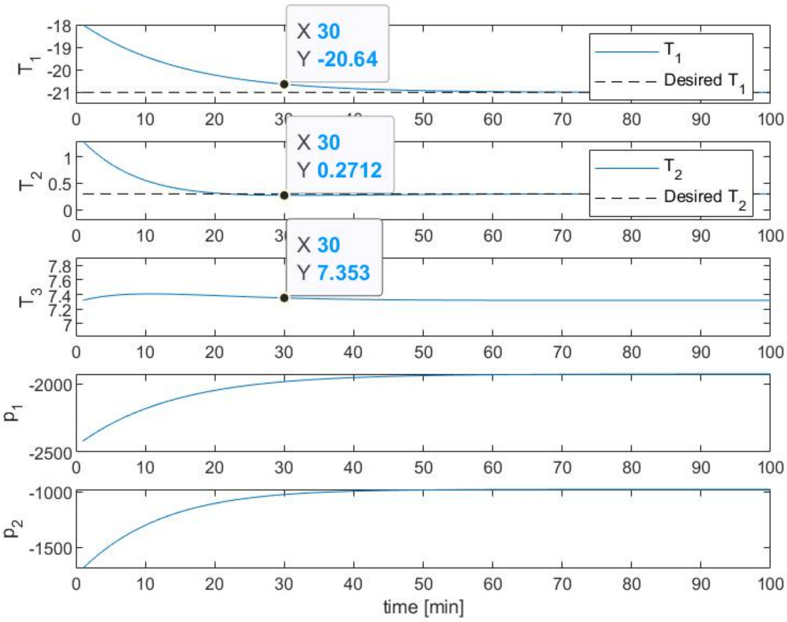


Figure 6: MPC2: $x_0 = (3, 1, 0)'$

sequences for the next step $\tilde{U} = \{u_1^*, ..., u_{N-1}^*, 0\}$, and then we know the corresponding state are also feasible as $x_{N+1} = Ax_N^* = 0$, we have $\tilde{X} = \{x_1^*, ..., x_N^* = 0, x_{N+1}\}$, we prove origin is an asymptotically stable point by showing that $J^*_{MPC2}(x(k+1)) < J^*_{MPC2}(x(k))$.
With the constructed control sequence for step $k+1$, we know it is feasible as shown above, and

$$\begin{aligned}
J^*_{MPC2}(x(k+1)) &\leq \tilde{J}_{MPC2}(x(k+1)) \\
&= \sum_{i=1}^{N} (x_i^*)^T Q x_i^* + (u_i^*)^T R u_i \\
&= \sum_{i=0}^{N-1} (x_i^*)^T Q x_i^* + (u_i^*)^T R u_i - ((x_0^*)^T Q x_0^* + (u_0^*)^T R u_0) \\
&< \sum_{i=0}^{N-1} (x_i^*)^T Q x_i^* + (u_i^*)^T R u_i \\
&= J^*_{MPC2}(x(k))
\end{aligned}$$

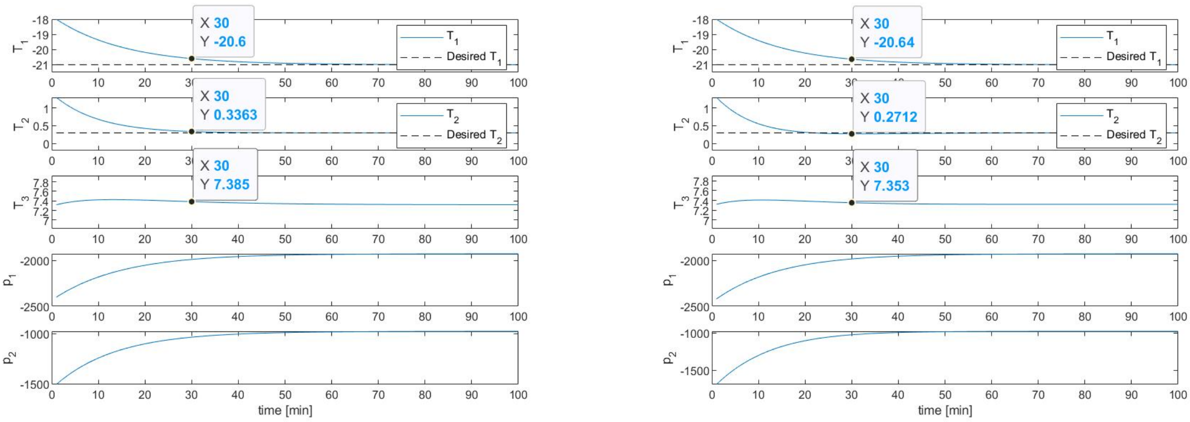
Question 12

The plot for MPC2 with initial value $x_0 = (3, 1, 0)'$ is as shown in Fig (6)

Question 13

The comparison between MPC1 and MPC2 wrt initial point $x_0 = (3, 1, 0)$ is shown as Fig (7). And we have $J_{MPC2}(T_{init}^{(1)}) = 5.8200e+04$ and $J_{MPC1}(T_{init}^{(1)}) = 5.5051e+04$ And we will find $J_{MPC1}(T_{init}^{(1)}) < J_{MPC2}(T_{init}^{(1)})$ because any feasible solution of MPC2 problem is also feasible to MPC1.

If we set the initial condition as $T_{init}^{(2)}$, then the program prompts MPC infeasible warning. Because from this initialization point, the system cannot track to the terminal set $X_f = \{0\}$



(a) Plot with MPC1 controller

(b) Plot with MPC2 controller

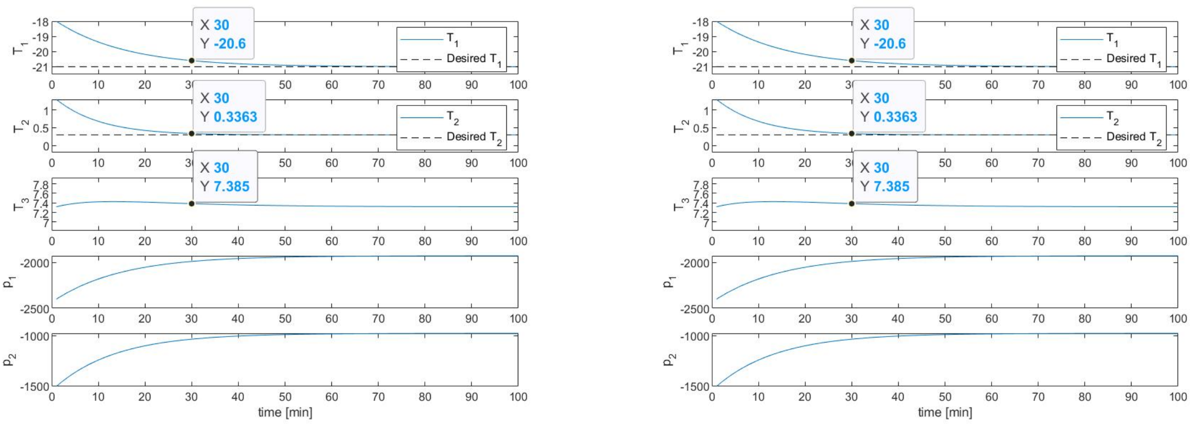
Figure 7: With $x_0 = (3, 1, 0)'$

Question 14

To guarantee the origin is asymptotically stable, we need make sure that $I_f(x_{k+1}) - I_f(x_k) \leq -I(x_k, \kappa_f(x_k))) = -(x_k^T Q x_k + (\kappa_f(x_k))^T R \kappa_f(x_k))$ and I_f is a Lyapunov function in the terminal set X_{LQR}. One way is simply to set the terminal cost as $x(k)^T P x(k)$, where P is the the solution to the Algebraic Riccati Equation.

Question 15

Because we set the terminal cost as $I_f(x) = x^T P_\infty x$, which exactly equals to the LQR infinite horizon cost J_∞^{LQR}, the only difference is that (5) has one more constraint on terminal state. For the given two initialization, the terminal state constraints are satisfied in the MPC1 controller. Therefore, the result of MPC3 and MPC1 is the same in term of numerical accuracy. The comparision results are shown in Fig (8 and 9).



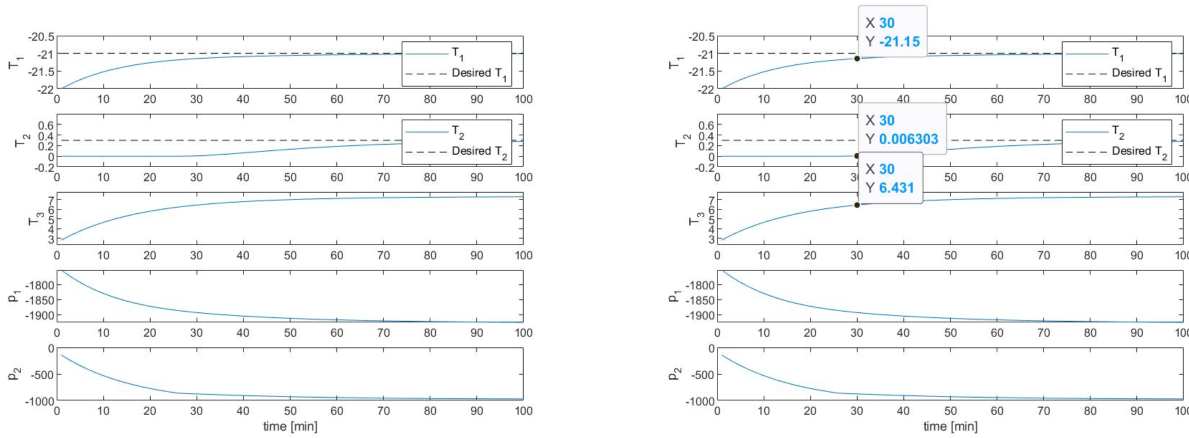
(a) Plot with MPC1 controller

(b) Plot with MPC3 controller

Figure 8: With $x_0 = (3, 1, 0)'$

Question 16

Because (4) enforces a stronger terminal condition which requires the terminal state is exactly zero, which leads to a smaller initial feasible set; on the contrary, (5) enforces that the terminal state should



(a) Plot with MPC1 controller

(b) Plot with MPC3 controller

Figure 9: With $x_0 = (-1, -0.3, -4.5)'$

lie in the set L_{LQR} which is much larger then zero state condition, therefore, the corresponding feasible initial set is larger.

Question 17

If the initial temperature is set to be $[12, 12, 12]'$, then I noticed that the MPC will become infeasible, therefore return NaN in MATLAB, and the corresponding computation is meaningless.

Question 18

In order that the controller will provide the same control as (5) if feasible, we introduce linear Penalty for the slack variables. The plot is shown as Fig (10).

Question 19

With Fig.11, we know with soft constraints and linear penalty, if the initial condition is feasible, then MPC will return the same solution as in the hard constraints case. (with negligible numerical difference).

Question 20

The temperature can be measure during evolution, therefore we choose $y(k) = I \cdot x(k) = Cx(k)$. The augmented model is as follows:

$$A_{aug} = \begin{bmatrix} A & B_d \\ o & I \end{bmatrix} \quad B_{aug} = \begin{bmatrix} B \\ \mathbf{0} \end{bmatrix} \quad C_{aug} = [C, \mathbf{0}] \quad D_{aug} = \mathbf{0} \tag{18}$$

Question 21

In the steady-state, we have $z_s = Hy_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} y = r$

$$\begin{bmatrix} A - I & B \\ HC & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} -B_d \hat{d} \\ r \end{bmatrix} \tag{19}$$

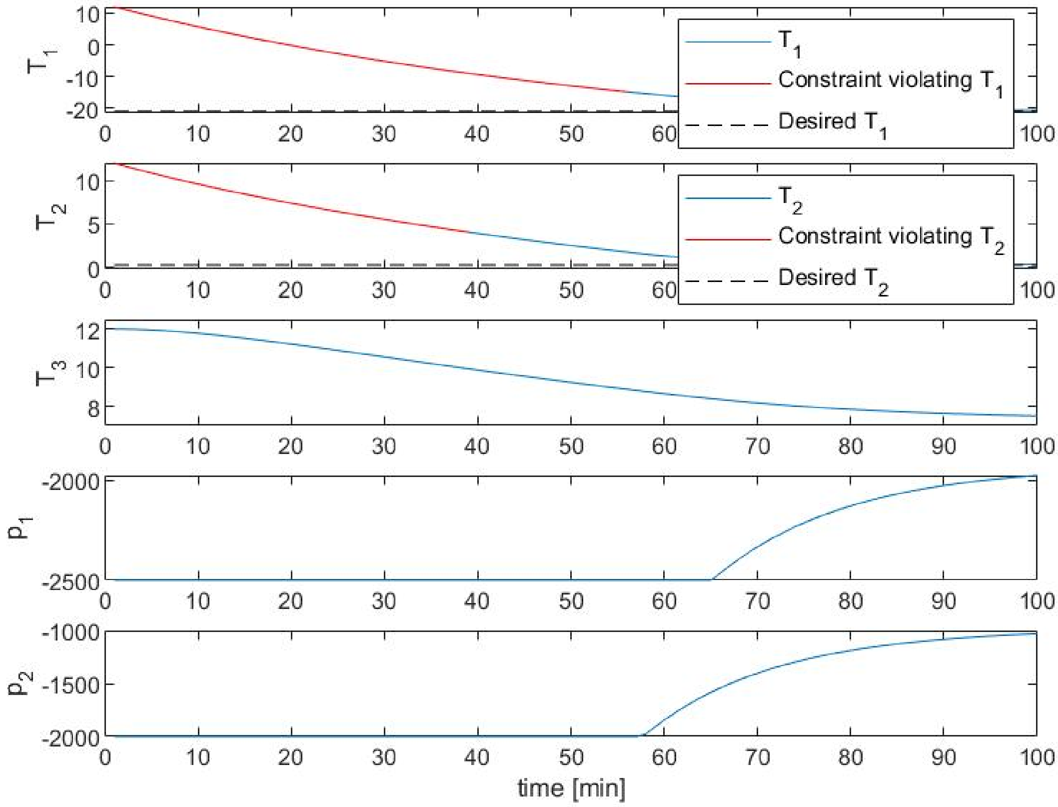
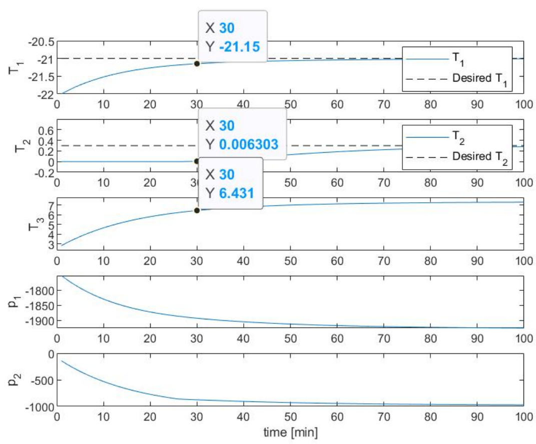
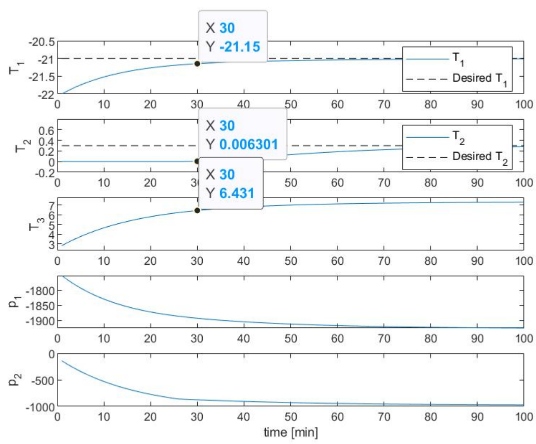


Figure 10: With soft constraint



(a) Plot without soft constraints: MPC3



(b) Plot with soft constraints: MPC4

Figure 11: With $x_0 = (-1, -0.3, -4.5)'$

Where $r = [-21; 0.3]$. Therefore, we can solve for the steady-state based on the estimated disturbance:

$$\begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} A - I & B \\ HC & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} -B_d \hat{d} \\ r \end{bmatrix} \tag{20}$$

Actually, because we don't have target temperature of the third zone and based on the fact that we can actually measure the temperature, i.e. $y(k) = x(k)$, the computation process is exactly the same as we do in Question 3. Except that now we use the estimated disturbance.
The estimator error dynamics is:

$$\begin{bmatrix} x - \hat{x}_{k+1} \\ \bar{d} - \hat{d}_{k+1} \end{bmatrix} = (A_{aug} - LC_{aug}) \begin{bmatrix} x - \hat{x}_k \\ \bar{d} - \hat{d}_k \end{bmatrix} \tag{21}$$

Using pole placement method, we choose the gain matrix for the second scenery as follows:

$$L = 1e3 \cdot \begin{bmatrix} 0.0010 & 0.0000 & 0.0000 \\ 0.0000 & 0.0009 & 0.0000 \\ 0.0000 & 0.0000 & 0.0008 \\ 0.8589 & -0.0088 & 0.0004 \\ -0.0097 & 1.5807 & -0.0914 \\ 0.0000 & -0.1123 & 1.7570 \end{bmatrix} \tag{22}$$

and the resulting eigenvalues for the error dynamics are $\{0.47, 0.59, 0.52, 0.55, 0.48, 0.51\}$.

Question 22

To make it clear, we use explicit delta formulation of MPC problem. The offset-free MPC at each step k can be conclude as follows:

1. Estimate state and disturbance $\hat{x}_k$ and $\hat{d}_k$;
2. Obtain x_s and u_s based on Equ. (20);
3. Solve the following MPC problem:

$$J_{MPC5}(x(k)) = \min_{\Delta u_i} \sum_{i=0}^{30-1} \Delta x_i^T Q \Delta x_i + \Delta u_i^T R \Delta u_i + I_f(\Delta x_{30}) \tag{23}$$

$$s.t. \quad \Delta x_0 = \hat{x}(k) - x_s \tag{24}$$

$$\Delta x_{i+1} = A\Delta x_i + B\Delta u_i \tag{25}$$

$$T_{min} - x_s \leq \Delta x_i \leq T_{max} - x_s \tag{26}$$

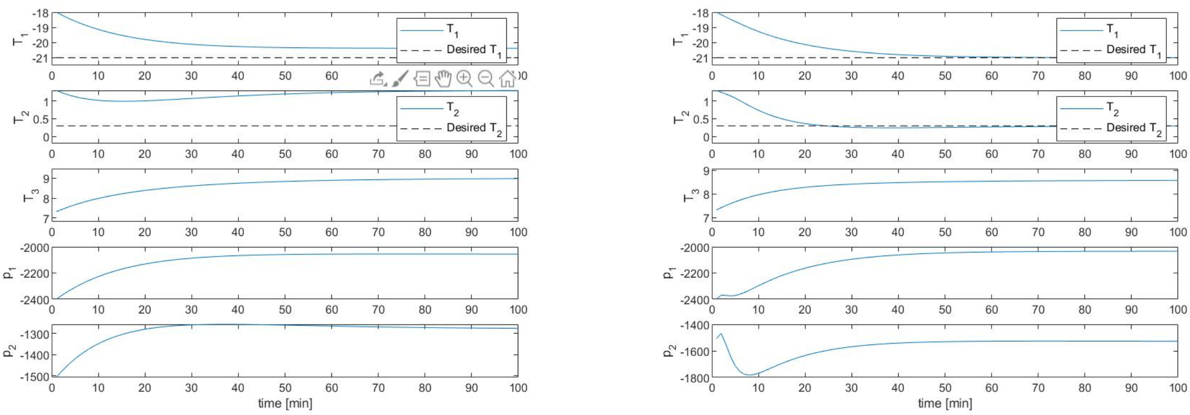
$$U_{min} - u_s \leq \Delta u_i \leq U_{max} - u_s \tag{27}$$

$$\Delta x_{30} \in X_{LQR} \tag{28}$$

For the initial condition, we estimate $\hat{x}_0 = T_0$ and $\hat{d}_0 = (120, 1200, 3600)'$. The comparison results are shown in Fig. 12. We can see, with the estimator-tracking mechanism, the tracking target is achieved while with normal MPC controller, there is an offset.

Question 23

I ran 20 times for both controllers and the average solver running time are roughly $0.48s$ for FORCE Pro and $0.75s$ for Matlab Controller. Clearly FORCE Pro runs much faster than normal Matlab Program.



(a) With Offset MPC: MPC3

(b) Offset-free MPC: MPC5

Figure 12: With Scenario 2 and $x_0 = (3, 1, 0)'$