Problem 1

1a

$$\alpha = 7.14 \text{ [rad/s]}, \quad \beta = 286.3 \text{ [rad/(V·s}^2\text{)]}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -7.14 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 286.3 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Table 1: LQR Gain and Closed-Loop Poles for $\rho = 1$ to 10

ρ	K_1	K_2	Pole 1	Pole 2
1	1.000	0.062	-12.486	-12.486
2	0.707	0.050	-10.676	-10.676
3	0.577	0.043	-9.767	-9.767
4	0.500	0.039	-9.183	-9.183
5	0.447	0.036	-8.761	-8.761
6	0.408	0.034	-8.437	-8.437
7	0.378	0.032	-8.176	-8.176
8	0.354	0.031	-7.960	-7.960
9	0.333	0.029	-7.776	-7.776
10	0.316	0.028	-7.617	-7.617

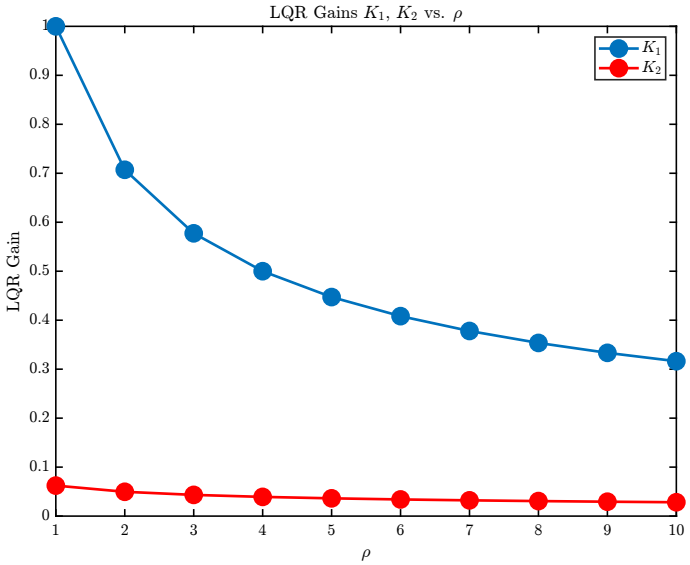
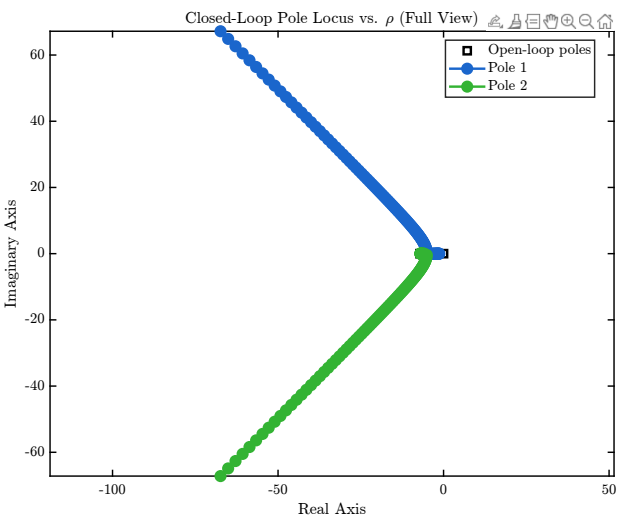


Figure 1: Variation of LQR Gains K_1 and K_2 as a Function of Control Weight ρ in State-Feedback Design

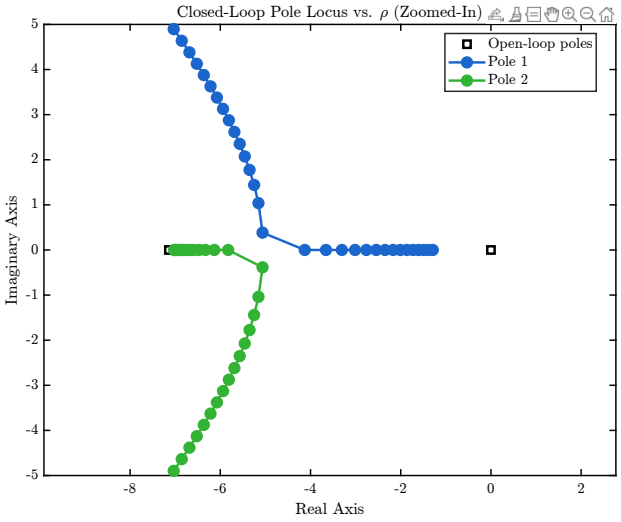
The model is designed to implement how the LQR state-feedback gains vary as a function of the control weighting parameter ρ, ranging from 1 to 10. It can be observed in Table 1 and Figure 1, both gains K_1 and K_2 decrease monotonically with increasing ρ, which indicates a more conservative control policy as higher emphasis is placed on minimizing control effort. The resulting closed-loop poles, which are listed in Table 1, also shift closer to the imaginary axis as ρ increases. Interestingly, the poles remain real and identical for all tested values of ρ, confirming critically damped behavior for this specific configuration of Q and system dynamics.

1b

Figure 2a illustrates the evolution of the closed-loop poles. Here, the LQR control weight ρ is modified over a wide range from 10^{-3} to 10^{3}. As ρ decreases, the poles move deeper into the left-half plane and gain larger imaginary components, which provides faster but potentially less stable responses. Conversely, as ρ increases, the poles approach the imaginary axis, resulting in slower system dynamics with more conservative control behavior. NOTE: Figure 2b highlights the region near the dominant poles for moderate values of ρ, and is for better visualization.



(a) Full-range pole locus as $\rho \in [10^{-3}, 10^{3}]$

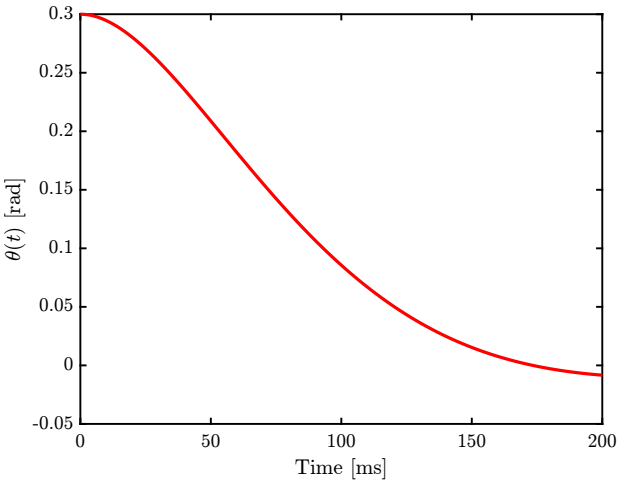


(b) Zoomed-in view of poles near the imaginary axis

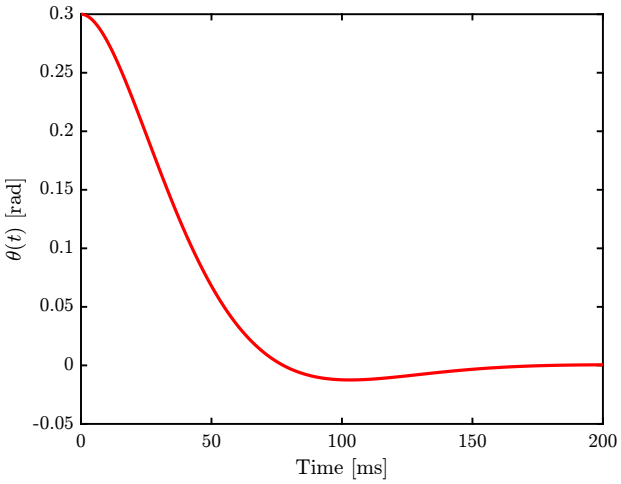
Figure 2: Closed-loop pole trajectories as ρ varies. (a) shows the full pole locus over a wide sweep of ρ, while (b) highlights the dominant poles near the imaginary axis.

1c

The effect assessment of the control weight ρ on system performance is applied, we analyzed the angular position $\theta(t)$ and control effort $u(t)$ for two values of ρ: a defaults value of 0.5 (which can be any other value) and an optimized value of 0.0231013 obtained from the Matlab code solution. That optimal ρ also satisfies the a 50 ms rise time criterion. Figure 3 displays that a reduced ρ enhances the system's reaction rate. The trajectory in subfigure 3a with $\rho = 0.5$ is noticeably slower and more gradual, while subfigure 3b indicates that with $\rho = 0.0231013$, the system achieves faster convergence and better damping. Nonetheless, this enhanced performance implies a greater control effort, as seen in Figure 4. Subfigure 4a illustrates the moderate voltage levels for $\rho = 0.5$, whereas subfigure 4b demonstrates the considerably greater control magnitude necessary for the optimum ρ.

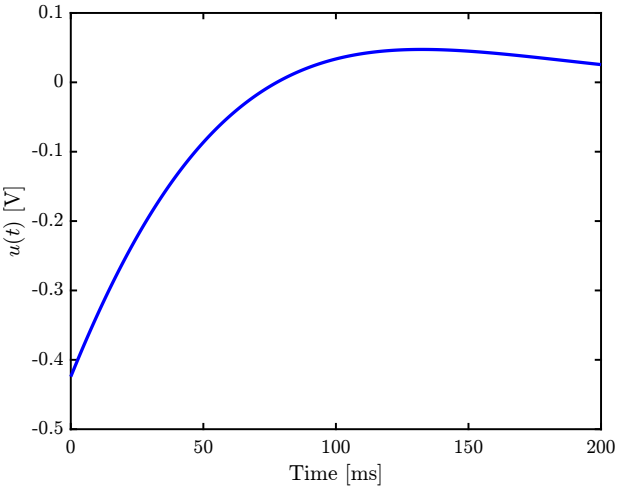


(a) $\theta(t)$ with $\rho = 0.5$

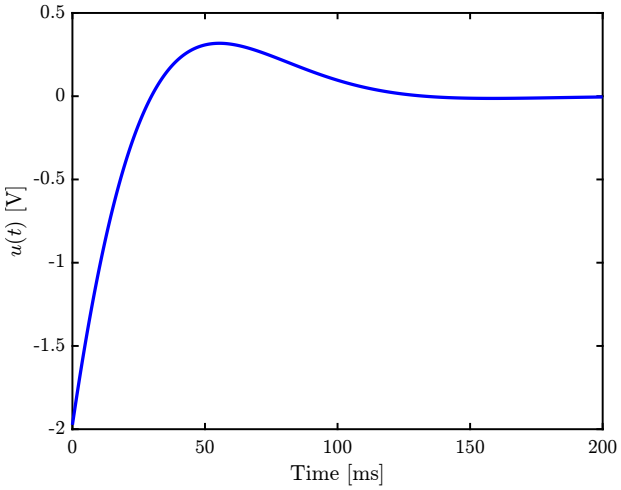


(b) $\theta(t)$ with optimal $\rho = 0.0231013$

Figure 3: Angular position $\theta(t)$ comparison for two values of ρ. (a) Higher $\rho = 0.5$ yields a slower response. (b) Optimized $\rho = 0.0231013$ achieves faster and well-damped behavior.



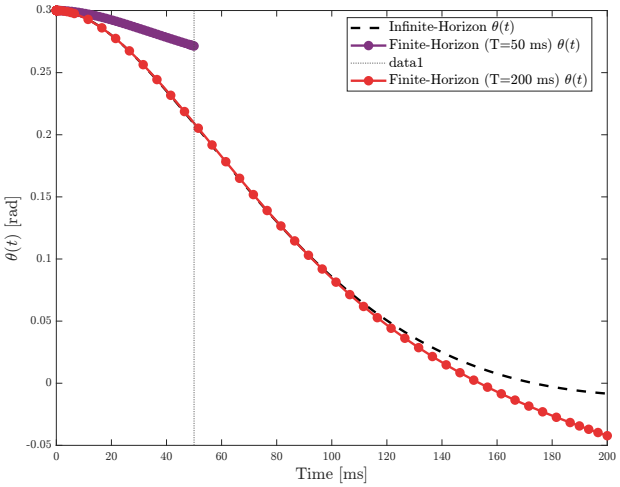
(a) $u(t)$ with $\rho = 0.5$



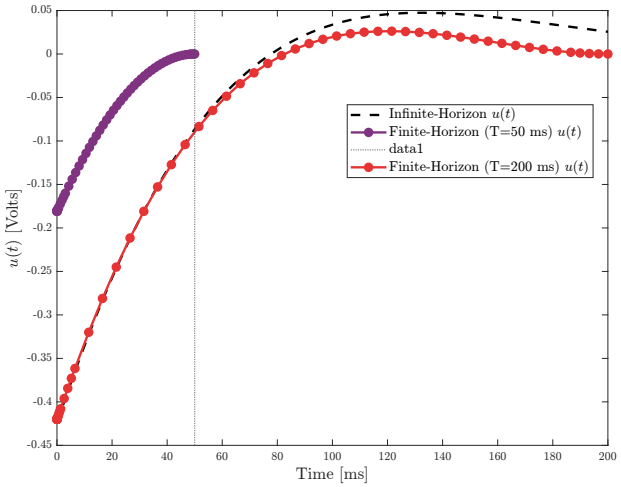
(b) $u(t)$ with optimal $\rho = 0.0231013$

Figure 4: Control input $u(t)$ comparison. (a) Lower effort for $\rho = 0.5$. (b) Optimized $\rho = 0.0231013$ requires higher initial control magnitude to meet the rise time constraint.

1d [EC]



(a) State response $\theta(t)$ for finite and infinite horizons



(b) Control input $u(t)$ for finite and infinite horizons

Figure 5: Comparison between finite-horizon and infinite-horizon LQR solutions. (a) shows the angular position $\theta(t)$ under three controllers: one infinite-horizon and two finite-horizon designs (50 ms and 200 ms). (b) shows the corresponding control inputs. Shorter horizons yield more time-varying and aggressive control.

To compare finite-horizon and infinite-horizon LQR performance, we solved the time-varying Riccati differential equation backward in time from t_f for two horizon lengths: 50 ms and 200 ms. Figure 5a shows that shorter finite horizons (e.g., 50 ms) produce responses that initially diverge from the infinite-horizon behavior via aggressive control efforts. Longer horizons such as 200 ms mathc more closely with the infinite-horizon solution over time. Figure 5b highlights the control effort $u(t)$: with a 50 ms horizon, the controller acts more forcefully early on, while the 200 ms solution closely tracks the infinite-horizon strategy.

Problem 2

2a and 2b

Here 2a and 2b solutions are provided together, and the final result is in the conclusion section (Table 3). A simulation-based optimization approach was implemented using `fmincon`, with spline-interpolated control discretized over varying numbers of time points n_t. For each case, the numerical values of the cost J, final control $u(t_f)$, and final state $x(t_f)$ were computed via the code and presented in Table 2. Before analyzing that let's write the Session 9 results with the given fixed values:

$$J^* = -\frac{2}{3} \approx -0.66667, \quad u^*(1) = \frac{4}{9} \approx 0.4444, \quad x^*(1) = \left(\frac{2}{3}\right)^2 = 0.4444.$$

Table 2 introduces the simulation results for $n_t = 3$ to 35. A combined absolute error metric I used is:

$$\text{Error} = |J - J^*| + |u(t_f) - u^*(1)| + |x(t_f) - x^*(1)|.$$

Emphasizing just on Objective function value or providing different weights to the J,$u(t_f)$,$x(t_f)$ would be another approach. I just did not want to make it easier or more complicated.

Table 2: Simulation results for varying time discretization points n_t.

n_t	**J**	$u(t_f)$	$x(t_f)$	**Total Error**
3	−0.66897	0.39644	0.44099	0.05376
4	−0.66737	0.41151	0.44338	0.03470
5	−0.66696	0.41914	0.44400	0.02603
11	−0.66669	0.43383	0.44441	0.01066
12	−0.66670	0.43461	0.44435	0.00996
13	−0.66670	0.43625	0.44443	0.00824
34	−0.66986	0.44746	0.44512	0.00688
35	−0.67023	0.44867	0.44527	0.00861

From this table, it can be seen that the error becomes minimal after the $n_t = 11$, and I chose the discretization level of $n_t = 11$. The combined error is relatively low, and the values of J, $u(t_f)$, and $x(t_f)$ are all in close agreement with its Session 9 results (Table 3). The Figure 6 visualizes how the objective value, final control input, and final state evolve as a function of n_t. We see more monotone-level plots in the top-left (the objective J) and similarly, the final state $x(t_f)$. Bottom-top right plots have more exponential changes in them. I believe these visual trends reinforce the numerical results in Table 2.

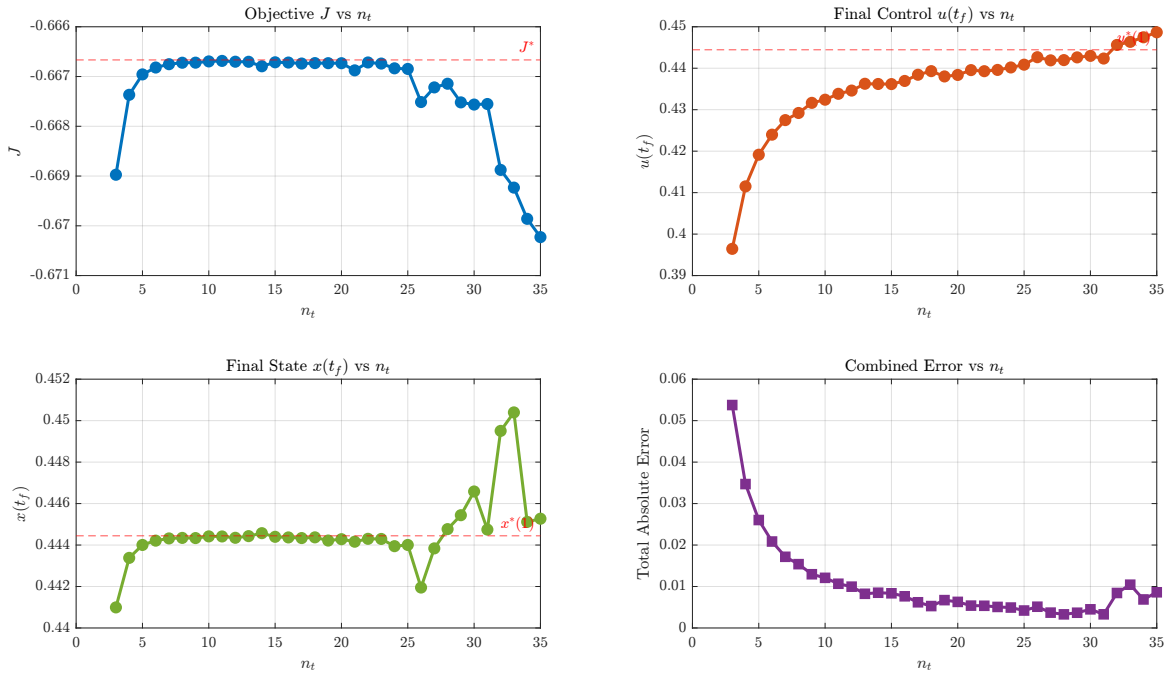
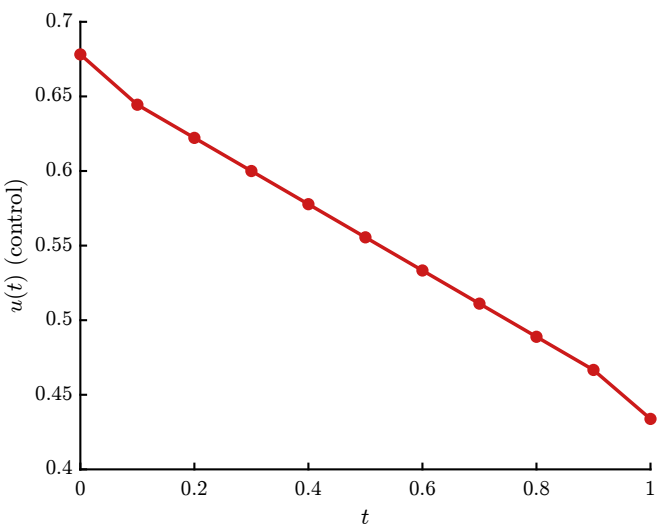
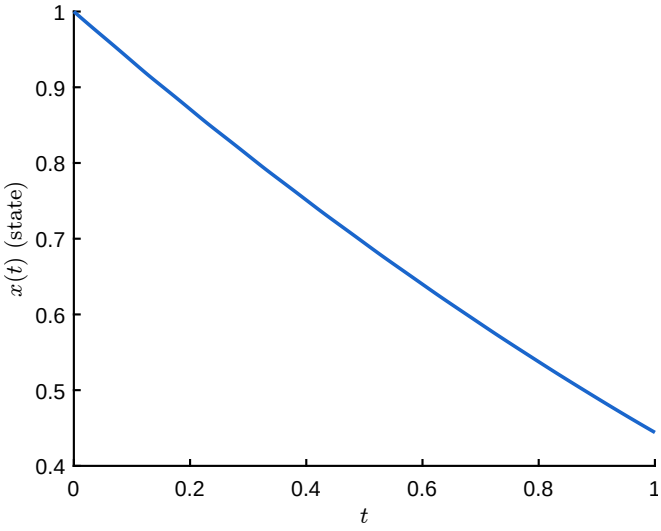


Figure 6: The figure shows how the objective J, final control $u(t_f)$, and state $x(t_f)$ change with discretization n_t, along with the total error. Best accuracy is chosen as $n_t = 11$.



(a) Optimal control signal $u(t)$



(b) Resulting state trajectory $x(t)$

Figure 7: Optimal control results computed using `fmincon` with $N = 11$ time points. (a) shows the control input $u(t)$; (b) shows the corresponding state trajectory $x(t)$.

Figure 7 illustrates the optimized control input and resulting system response when $N = 11$. In Fig. 7a, the control signal $u(t)$ decreases smoothly over time in a stepwise fashion due to discretization, whereas Fig. 7b shows the corresponding state trajectory $x(t)$, which declines monotonically as expected.

[CONCLUSION] To conclude, in my opinion $N = 11$ time points are needed for an accurate solution, which can be seen from the comparison Table 3

Source	J	$x(1)$	$u(1)$
Session 9 Result	-0.66666	0.44444	0.44444
My Result	-0.66669	0.43383	0.44441

Table 3: Comparison of Session 9 Analytical Results and My Numerical Results at $t = 1$

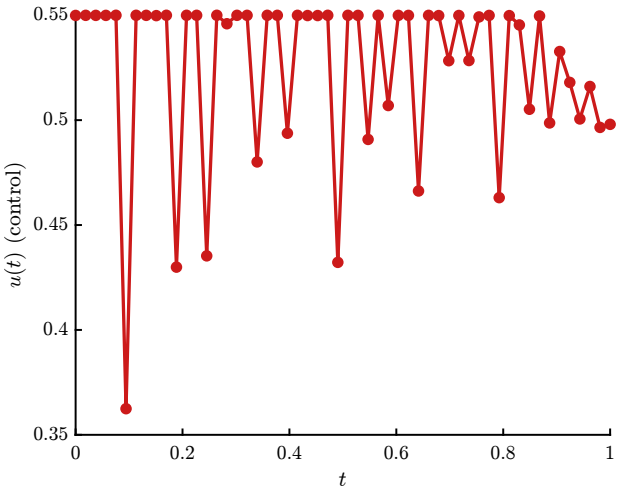
2c

I also did a similar approach here, like I applied in previous cases, and from the data analytics results, it seems in this constrained case it takes at least $N = 54$ time points for the model to find the a reasonably accurate and smooth solution.

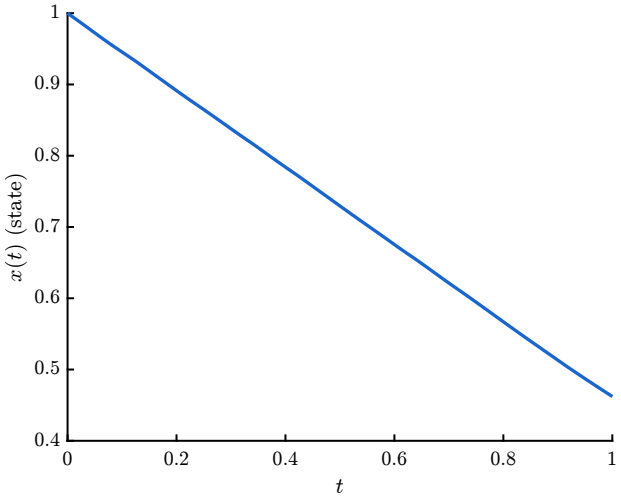
Table 4: Constrained Optimal Control Problem Results

n_t	**J**	$u(t_f)$	$x(t_f)$	**Total Error**
15	-0.66297	0.45690	0.46570	0.03741
16	-0.66299	0.45761	0.46558	0.03798
17	-0.66300	0.45936	0.46579	0.03993
		⋮		
50	-0.66513	0.48532	0.47463	0.07261
51	-0.66789	0.50225	0.48499	0.09958
52	-0.66657	0.49537	0.47061	0.07719
53	-0.66663	0.49676	0.46529	0.07320
54	-0.66660	0.49799	0.46214	0.07131
55	-0.66707	0.50144	0.45930	0.07226

Figure 8 illustrates this behavior. In Fig. 8a, the control signal maintains a constant maximum value before experiencing a swift decrease towards the conclusion. This transition introduces steep gradients in the control profile, which are harder to capture accurately with a coarse time discretization. Figure 8b illustrates the resultant state trajectory, which maintains a smooth and monotonic nature despite the more abrupt control changes Due to this change in problem structure, a finer discretization is needed to resolve the sharp change in control.



(a) Optimal control $u(t)$ with constraint $u(t) \leq 0.55$



(b) Resulting state trajectory $x(t)$

Figure 8: Constrained optimal control results with path constraint $u(t) \leq 0.55$. (a) shows the control profile with a flat segment due to the active constraint, followed by a sharp drop. (b) shows the corresponding state response.