State Feedback and Observer Design for Inverted Pendulum

Homework

Notes:

- The homework should reflect your own work!
- The deadline of submission: Midnight, December 10, 2023 via Teams assignment
- Deadline of late submission: Midnight, December 17, 2023 via Teams assignment (with paying late submission fee in NEPTUN)

Problem: The system inverted pendulum consists of a cart which can be moved along a metal guiding bar and a metal rod is fixed to the cart by an axis. The scheme of the inverted pendulum is given in Figure 1. The state vector consists of the position of cart, the velocity of cart, the angle of pendulum and the angular velocity of pendulum.

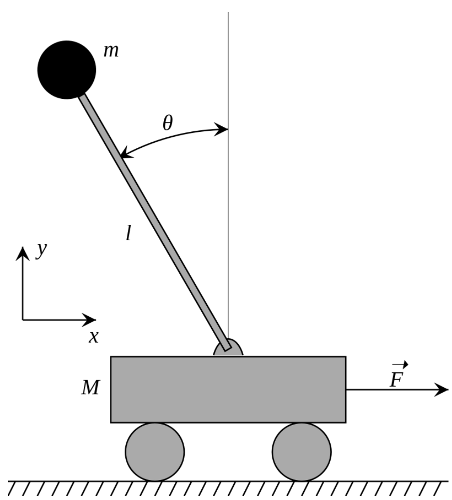


Figure 1: Scheme of the inverted pendulum

The actual values of the cart mass, M, pendulum tip mass, m and pendulum length, l can be found from the attached table based on your NEPTUN code. There is a viscous damping $b = 0.2\,Ns/m$ that acts on the cart in the x direction. The inertia of the pendulum, J_s around its end point can be given as $J_s = \frac{1}{3}ml^2$.

Equations of motion: The differential equations governing the motion of the inverted pendulum can be given in the following way.

The inverted pendulum can be modeled as a 2 DoF system. Let us choose the generalized coordinates as

$$q = \begin{bmatrix} \phi \\ x \end{bmatrix} \tag{1}$$

The equations of motions based on the Euler-Lagrange equations have the form

$$\frac{d}{dt}\frac{\partial \mathcal{T}}{\partial \dot{q}_i} - \frac{\partial \mathcal{T}}{\partial q_i} + \frac{\partial \mathcal{U}}{\partial q_i} + \frac{\partial \mathcal{D}}{\partial \dot{q}_i} = Q_i \tag{2}$$

The kinetic energy of the system is

$$\mathcal{T} = \mathcal{T}_1 + \mathcal{T}_2 \tag{3}$$

where

$$\mathcal{T}_1 = \frac{1}{2}M\dot{x}^2 \tag{4}$$

$$\mathcal{T}_2 = \frac{1}{2}mv_s^2 + \frac{1}{2}J_s\dot{\phi}^2 \tag{5}$$

The velocity of the c.g. of the pendulum is

$$v_s = \begin{bmatrix} \dot{x} + l\dot{\phi}cos\phi & -l\dot{\phi}sin\phi & 0 \end{bmatrix}^T \tag{6}$$

The potential energy of the system $\mathcal{U}$ is

$$\mathcal{U} = \mathcal{U}_2 = mglcos\phi \tag{7}$$

The dissipated energy $\mathcal{D}$ is

$$\mathcal{D} = \mathcal{D}_1 = \frac{1}{2}b\dot{x}^2 \tag{8}$$

The generalized forces can be obtained based on the power of the external forces

$$\mathcal{P} = F^T v_a = Q_1\dot{\phi} + Q_2\dot{x} = 0 + u\dot{x} \tag{9}$$

The nonlinear equation of motions are

$$\begin{bmatrix} J_s + ml^2 & mlcos\phi \\ mlcos\phi & M + m \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} -mglsin\phi \\ b\dot{x} - ml\dot{\phi}^2 sin\phi \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix} \tag{10}$$

The equations can be linearized around the trim point $\bar{\phi} = \bar{\dot{\phi}} = \bar{x} = \bar{\dot{x}} = 0$ which means that the pendulum is in upright position. The resulting linearized equations of motion are

$$\begin{bmatrix} J_s + ml^2 & ml \\ ml & M + m \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} -mgl & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix} \tag{11}$$

Tasks:

1. Implement the nonlinear model in Matlab/Simulink.

2. Linearize the model in Matlab/Simulink and compare the resulting linear model with the analytical linearization. Make sure that the ordering of the states is the following: $x = \begin{bmatrix} \phi & \dot{\phi} & x & \dot{x} \end{bmatrix}$. This is important because details in later tasks assume this ordering of the states and it also makes the correction easier. If the linearization returns different ordering, use Matlab's *xperm* function.

3. Give the poles, corresponding damping factors and time constants of the linearized system. Include the pole/zero map in the documentation.

4. Show that the system is controllable and observable.

5. Design a state feedback controller with pole placement. Start with $p = \begin{bmatrix} -1; -1.1; -1.2; -1.3 \end{bmatrix}$. Test the controller with the nonlinear system with initial conditions: $x_0 = 1.1$, $\dot{x}_0 = 0.4$, $\phi_0 = 0.1$, $\dot{\phi}_0 = 0.3$. Try to increase the poles until the controller brakes down and decrease them until the controller has not enough output to drive the system into zero states. Figures of all these responses are mandatory in the documentation, including the control signal u.

6. Design an LQR state feedback.
   Start with $Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$, $R = 1$. Try out different weighting schemes and see their effect on the control performance. Figures of all these responses are mandatory in the documentation, including the control signal u. Explain how the change of the Q and R matrices effect the results.

7. Design a full state observer based on the measurement of x (the state which indicates the displacement of the card). Start with $\sigma_{W_d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$, $\sigma_{W_n} = 0.1$. Test the LQR controller and Kalman filter with the nonlinear system with initial conditions: $x_0 = -0.01$, $\dot{x}_0 = 0.01$, $\phi_0 = 0.05$, $\dot{\phi}_0 = 0.1$. Try out different covariances as well and simulate the resulting observers. Include measurement noise in the simulation (Variance= 0.0001, mean = 0, seed = 10000, sample time = 0.001s).

8. In this part you are asked to use square cart position trajectories and examine the trajectory tracking of the LQG controlled system (with nonzero reference signal compensation).

MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Mechanical Engineering

Advanced Control and Informatics Homework

Gergő Kelle

December 15, 2023

Contents

Initial data and equations

The homework focuses on studying the dynamics of an inverted pendulum system, including the cart's position, velocity, pendulum angle, and angular velocity. As for initial data the homework description defined a viscous damping $b = 0.2$ [Ns/m] along the x axis. While the inertia of the pendulum, $J_s = \frac{1}{3} m\ L^2$.
Further initial data is showed in Table 1.

Table 1: Initial Data based on *Homework Codes.pdf*

m [kg]	M [kg]	L [m]
1.53	7.37	3.05

Based on the given parameters in Table 1 we can calculate J_s as:

$$J_s = \frac{1}{3} m\ L^2 = 4.744275\ \ \left[\mathrm{kgm}^2\right] \tag{1}$$

The homework description also defined the generalized coordinates that should be used during calculation, which is:

$$\mathbf{q} = \begin{bmatrix} \phi \\ x \end{bmatrix} \quad , \quad \dot{\mathbf{q}} = \begin{bmatrix} \dot{\phi} \\ \dot{x} \end{bmatrix} \quad , \quad \ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\phi} \\ \ddot{x} \end{bmatrix} \tag{2}$$

Regarding to this generalized coordinates the nonlinear and the linear equation of motion is also given as:

$$\text{Nonlinear:} \quad \begin{bmatrix} 0 \\ u \end{bmatrix} = \begin{bmatrix} J_s + m\ L^2 & m\ L\ \cos(\phi) \\ m\ L\ \cos(\phi) & M + m \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} -m\ g\ L \sin(\phi) \\ b\ \dot{x} - m\ L\ \dot{\phi}^2 \sin(\phi) \end{bmatrix} \tag{3}$$

$$\text{Linearization:} \quad \sin\phi = \phi \qquad \text{and} \qquad \cos\phi = 1 \qquad \textit{for small } \phi$$

$$\text{Linearized:} \quad \begin{bmatrix} 0 \\ u \end{bmatrix} = \begin{bmatrix} J_s + m\ L^2 & m\ L \\ m\ L & M + m \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \dot{\mathbf{q}} + \begin{bmatrix} -m\ g\ L & 0 \\ 0 & 0 \end{bmatrix} \mathbf{q} \tag{4}$$

Task 1 - Nonlinear model

Rearranging the equation to get the second derivatives of the generalized coordinates we get the following equation:

$$\ddot{\mathbf{q}} = \begin{bmatrix} J_s + m\ L^2 & m\ L\ \cos(\phi) \\ m\ L\ \cos(\phi) & M + m \end{bmatrix}^{-1} \begin{bmatrix} m\ g\ L \sin(\phi) \\ u - b\ \dot{x} + m\ L\ \dot{\phi}^2 \sin(\phi) \end{bmatrix} \tag{5}$$

With the help of two integrators the generalized coordinates can be determined for the nonlinear model. The first derivatives can be produced by using a loop back from after the first integrator. The created Simulink model for the nonlinear system case can be viewed in Figure 1 while the system response to a unit step function is visualized on Figure 2.

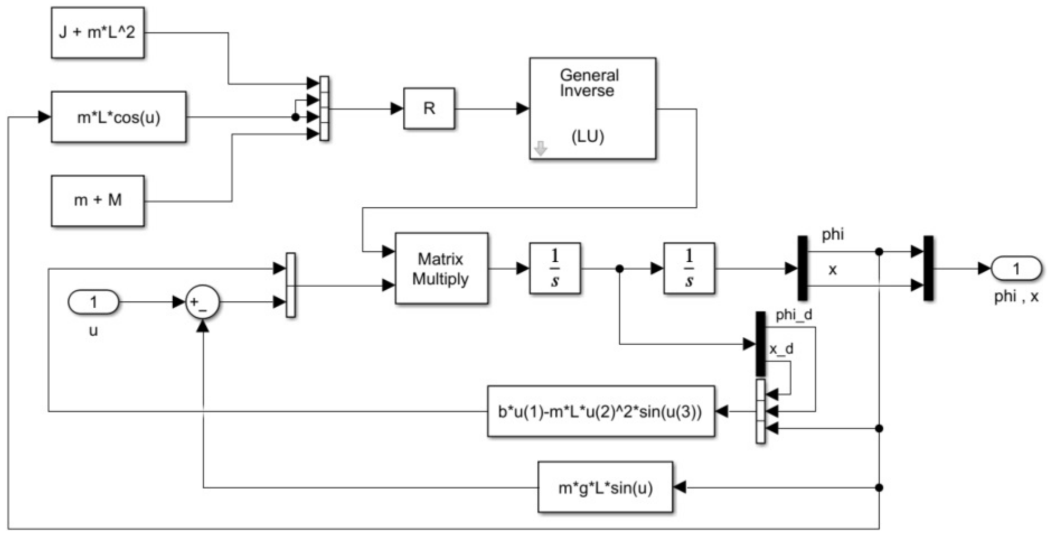


Figure 1: Simulink implementation of the nonlinear model (subsystem)

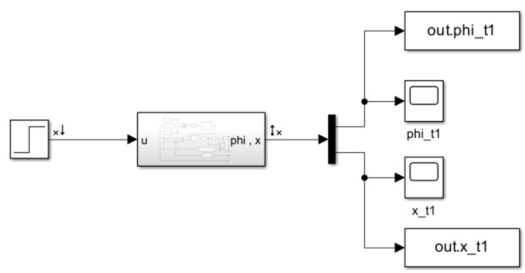


Figure 2: Unit step function excitation

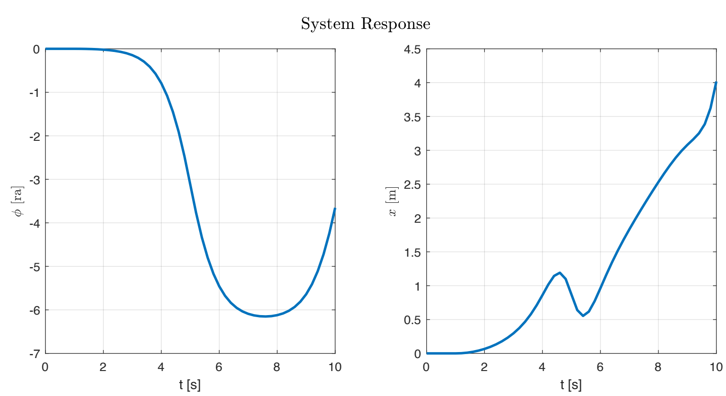


Figure 3: Results for generalized coordinates

Based on the results seen in Figure 3 it can be stated that the system is unstable, thus a control mechanism is needed.

Task 2 - Linearized model

The linerization can be done via *Simulink* automatic linearization and with analical calculations. In order to have a proper documentation about the homework I've choosed to do analytic linearization. With the help of the second order equation of motion the linearized state and input matrices can be calculated as equation (6) and (7) shows. *The resulting numerical solution can be viewed below.*

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \tag{6}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{F} \end{bmatrix} \tag{7}$$

$$\text{Where} \left| \begin{array}{llll} \text{State matrix} : \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 2.7694 & 0 & 0 & 0.0063 \\ 0 & 0 & 0 & 1 \\ -1.4520 & 0 & 0 & -0.0258 \end{bmatrix} & \text{Input matrix} : \mathbf{B} = \begin{bmatrix} 0 \\ -0.0317 \\ 0 \\ 0.129 \end{bmatrix} \\ \text{Mass matrix} : \mathbf{M} = \begin{bmatrix} 18.9771 & 4.6665 \\ 4.6665 & 8.9 \end{bmatrix} & \text{Damping matrix} : \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0.2 \end{bmatrix} \\ \text{Stiffness matrix} : \mathbf{K} = \begin{bmatrix} -45.7784 & 0 \\ 0 & 0 \end{bmatrix} & \text{Excitation vector} : \mathbf{F} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{array} \right.$$

Before visualization the aoutmatic linearization can be done using the `linearize` function in MATLAB, since the ordering is not what we desire I've also implemented an `xperm` function.
The two method gave very similar results, but to be exact in case of state and input matrices the following numerical values were determined as a result of the difference between the two method.

$$\mathbf{A}_{analytic} - \mathbf{A}_{automatic} = 10^{-10} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.4616 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.2420 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

$$\mathbf{B}_{analytic} - \mathbf{B}_{automatic} = \mathbf{0} \tag{9}$$

The system response to a step function for both the linearized and nonlinear models is illustrated in Figure 5.

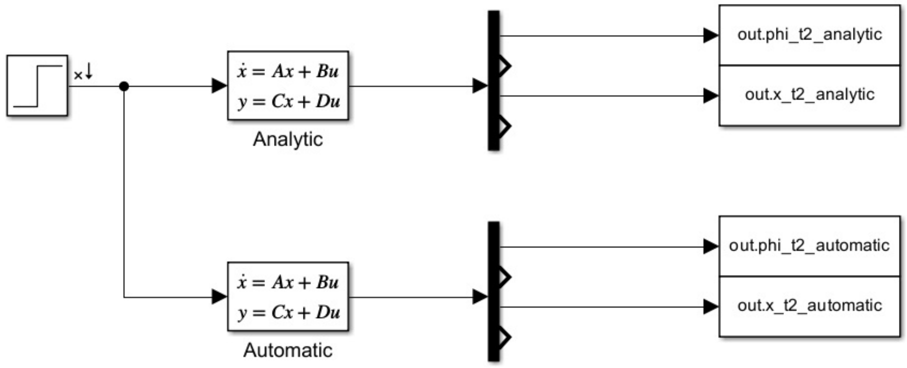


Figure 4: Simulink Model for Solution Method Comparison

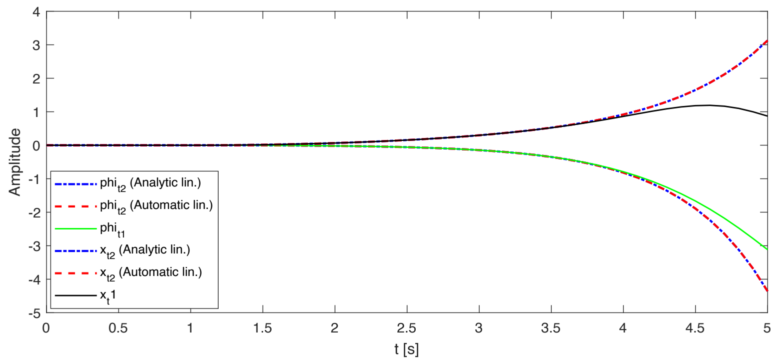


Figure 5: Comparison of Analytic and Automatic Linearization Results With The Nonlinear Solution

Task 3 - Pole/zero map

In this task I've calculated the poles with the corresponding damping factors and time constants of the linearized system determined in Task 2 to extract key parameters and visualize its behavior.

As for the pole-zero parameters the natural frequencies (ω_n), damping factors (ζ), and poles of the linearized system were obtained using the `damp` function in MATLAB:

```
[omega_n, damping_factors, poles] = damp(SYS);
```

The time constants (τ) associated with each pole were calculated using the formula $\tau = \dfrac{1}{\omega_n \cdot \zeta}$:

```
time_c = 1 ./ (omega_n .* damping_factors);
```

With all theese parameters the pole-zero map of the linearized system was generated using the `pzmap` function in MATLAB. The resulting pole-zero map is showed in Figure 6 while the poles with the corresponding damping factors and time constants is summarized in Table 2.

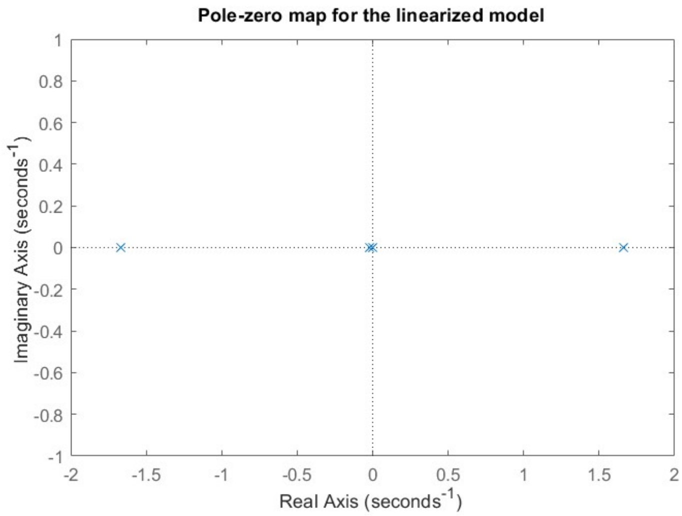


Figure 6: Pole-zero map

Table 2: Pole-zero map key parameters

Natural frequency	Poles	Damping factors	Time constants
0	0	-1	$-\infty$
0.022471	-0.022471	1	44.501
1.6625	1.6625	-1	-0.6015
1.6658	-1.6658	1	0.6003

Task 4 - Controllability and observability

To determine the controllability of the system, the controllability matrix **Crtb** was formed using the `ctrb` function in MATLAB. The determination of controllability is based on the following equation:

$$\underbrace{\text{length}(\mathbf{A})}_{n} - \text{rank}(\mathbf{Crtb}) = \begin{cases} \text{controllable}, & \text{if } \dots = 0 \\ \text{NOT controllable}, & \text{if } \dots \neq 0 \end{cases} \tag{10}$$

The difference $n - \text{rank}(\mathbf{Crtb})$ was computed, where n is the number of states. If this difference is zero, the system is considered controllable. My result showed that the system is **controllable**.

To assess the observability of the system, the observability matrix **Obsv** was formed using the `obsv` function:

$$\underbrace{\text{length}(\mathbf{A})}_{n} - \text{rank}(\mathbf{Obsv}) = \begin{cases} \text{observable}, & \text{if } \dots = 0 \\ \text{NOT observable}, & \text{if } \dots \neq 0 \end{cases} \tag{11}$$

The system's observability is determined by computing the difference $n - \text{rank}(\mathbf{Obsv})$. The system is considered observable if this difference is zero because the value of $n - \text{rank}(\mathbf{Obsv})$ indicates the number of unobservable states in the system. If this quantity is zero, all states are observable. My result showed that the system is **observable**.

Task 5 - Nonlinear system state feedback controller with pole placement

In this task, a state feedback controller was designed with pole placement, starting with the initial poles $p = [-1; -1.1; -1.2; -1.3]$ and initial conditions of $\phi_0 = 0.1,\ \dot{\phi}_0 = 0.3,\ x_0 = 1.1,\ \dot{x}_0 = 0.4$. The controller was tested on the nonlinear system showed in Figure 7 with varying initial conditions.

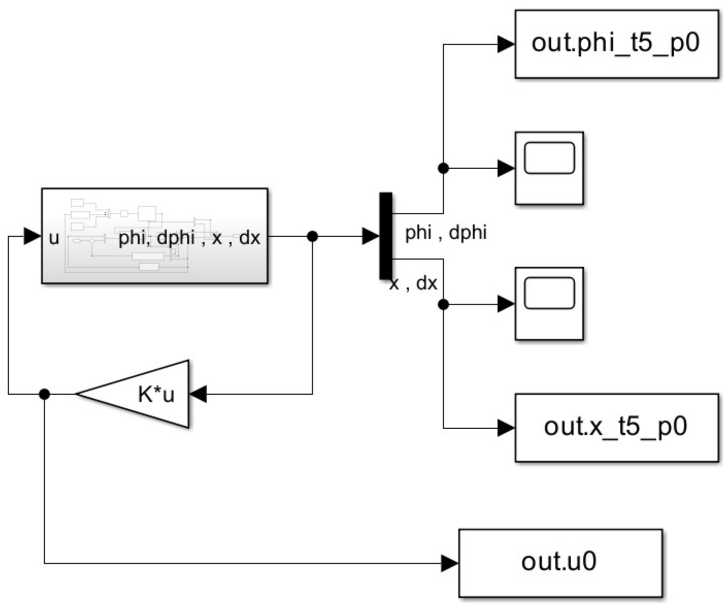


Figure 7: Simulink Model for Task 5

Pole Increase

In case of pole increasing for each iteration, new pole values were calculated, and state feedback gains were computed using the `place` function. Simulink models were run, and figures were generated to visualize the system's response, including plots for ϕ, x and the control signal u.

The different $\mathbf{p}$ values can be seen in Table 3 while the resulting solutions is showed in Figure ??.

Table 3: Pole increasing case

i	$\mathbf{p}_{\text{inc}}$
initial	$[-1; -1.1; -1.2; -1.3]$
1	$[-0.75; -0.85; -0.95; -1.05]$
2	$[-0.5; -0.6; -0.7; -0.8]$
3	$[-0.25; -0.35; -0.45; -0.55]$
4	$[0; -0.1; -0.2; -0.3]$
5	$[0.25; 0.15; 0.05; -0.05]$

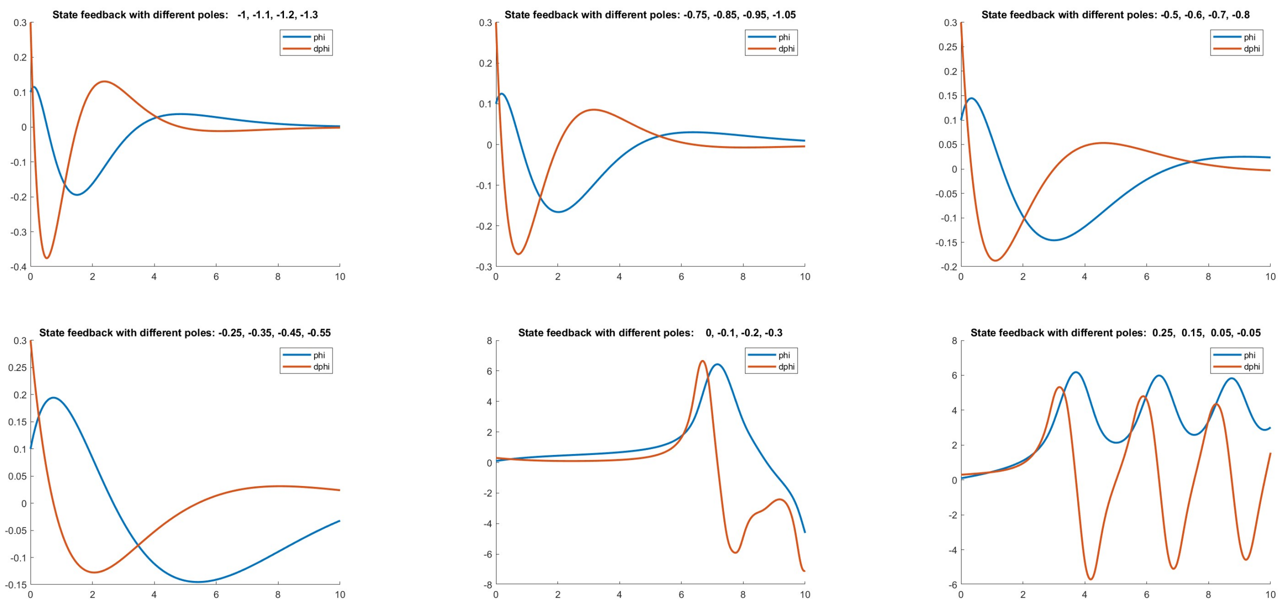


Figure 8: System's Respons with Different $\mathbf{p}$

Pole Decrease

As for pole decresing similarly to the pole increasing case, new pole values were calculated (as Table 4 shows), and state feedback gains were computed. Simulink models were run, and figures were created for system response showed in Figure **??**.

Table 4: Pole decreasing case

i	$\mathbf{p}_{\text{dec}}$
initial	$[-1; -1.1; -1.2; -1.3]$
1	$[-1.4; -1.5; -1.6; -1.7]$
2	$[-1.8; -1.9; -2; -2.1]$
3	$[-2.2; -2.3; -2.4; -2.5]$
4	$[-2.6; -2.7; -2.8; -2.9]$
5	$[-3; -3.1; -3.2; -3.3]$

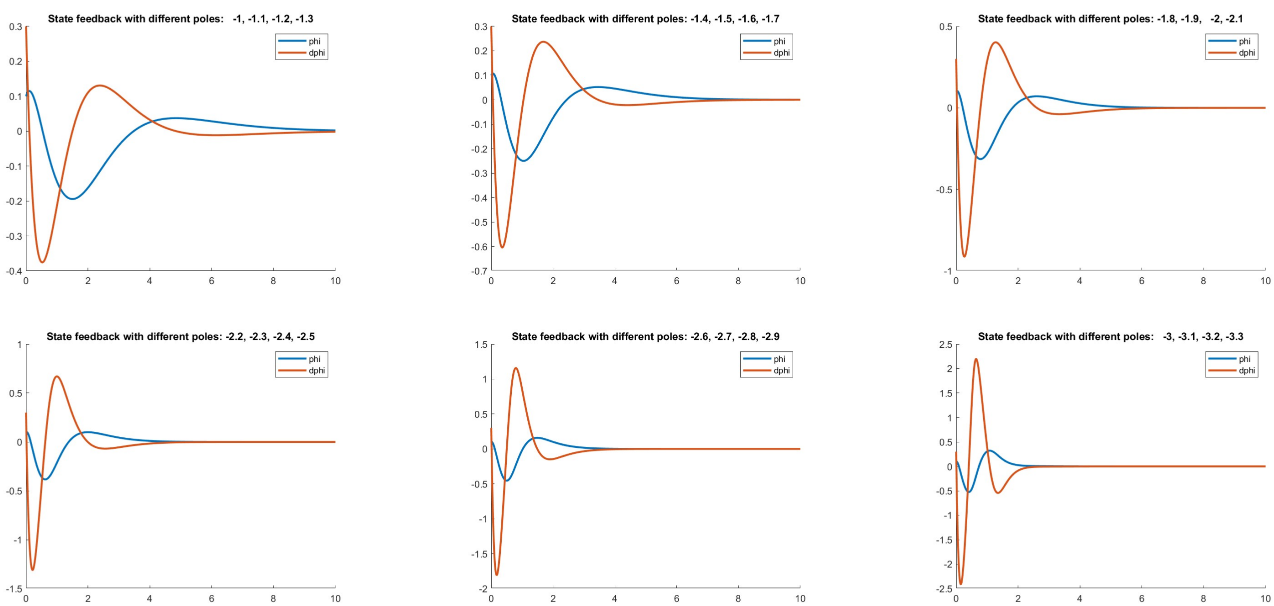
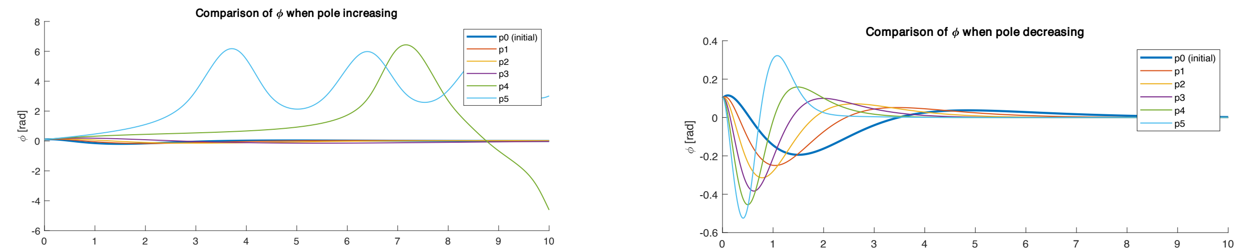


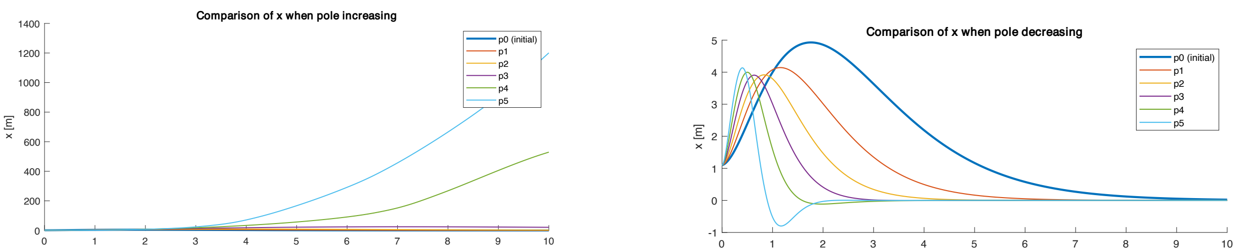
Figure 9: System's Respons with Different $\mathbf{p}$

Comparison

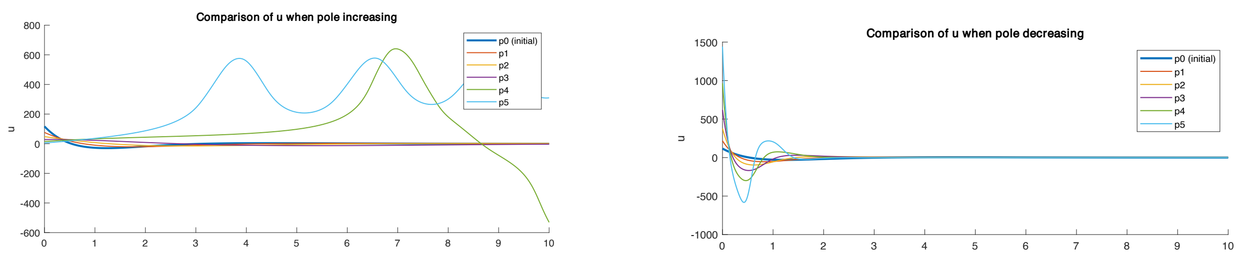
Comparison figures for ϕ and x were created to observe the system's behavior with different pole values. Additionally, the state feedback gains for each iteration were displayed.



(a) ϕ Plots for Pole Increase and Decrease



(b) x Plots for Pole Increase and Decrease



(c) u Plots for Pole Increase and Decrease

Figure 10: Effect Comparison of Pole Increase and Decrease

In the analysis, it was observed that varying the pole values had a significant impact on the system's response to a unit step function. By increasing the poles, from $\mathbf{p}_{\text{inc},4}$ the system lost its stability. On the contrary, reducing the pole values did not lead to a loss of stability. However, further decreasing the poles caused MATLAB simulations to become unresponsive, suggesting large $\mathbf{K}$ matrices just below $\mathbf{p}_{\text{dec},5}$.

In summary whenever a pole is a positive real value the system loses its stability and on the other hand whenever a pole is decreased to a certain degree the MATLAB could not calculate the resulting solution due to large $\mathbf{K}$ values.

Task 6 - LQR State Feedback

In this task, an LQR (Linear Quadratic Regulator) state feedback controller was designed with varying weighting schemes for the state and control input. The effects of changing the $\mathbf{Q}$ matrices and R value on control performance were analyzed using a similar Simulink model showed in Figure 7.

In the MATLAB I've defined different weighting matrices starting from the initial $\mathbf{Q}_1$ to $\mathbf{Q}_5$ and calculated the corresponding LQR controllers.

By only increasing the value of the (1,1) element in case of $\mathbf{Q}_2$, the system will place more emphasis on minimizing the cost associated with the angular position ϕ. This adjustment can result in the controller putting more effort into controlling and reducing deviations in ϕ. This weighting scheme is practical if we want to prioritize the control of ϕ. To make practical sense about the used weighting schemes I've summarized them in Table 5.

Table 5: Different Q Matrices and Their Practical Meanings

Iteration	**Q Matrix**	**Practical Meaning**
1	$\text{diag}\begin{bmatrix}100 & 1 & 10 & 100\end{bmatrix}$	Initial $\mathbf{Q}$ matrix
2	$\text{diag}\begin{bmatrix}100 & 1 & 1 & 1\end{bmatrix}$	Emphasis on ϕ
3	$\text{diag}\begin{bmatrix}1 & 1 & 100 & 1\end{bmatrix}$	Emphasis on x
4	$\text{diag}\begin{bmatrix}100 & 1 & 100 & 1\end{bmatrix}$	Emphasis on both ϕ and x
5	$\text{diag}\begin{bmatrix}1 & 100 & 1 & 100\end{bmatrix}$	Emphasis on both $\dot{\phi}$ and $\dot{x}$

The system response for different weighting schemes are showed in Figure 12.

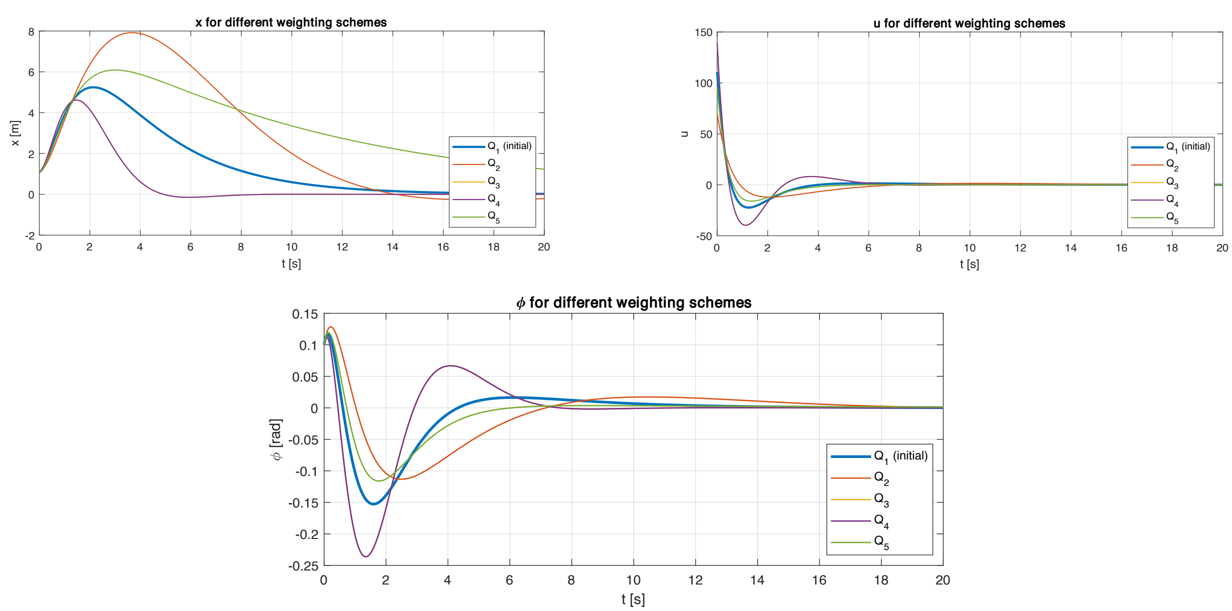


Figure 11: Effect Comparison of Different Weighting Schemes (only $\mathbf{Q}$ is changed)

In the end I've tried out different R values to see the cahnge of the results with the initial $\mathbf{Q}$ matrix. The choice of R influences the trade-off between achieving desired control performance and minimizing control effort. Therefore I've defined one with higher and one with lover value than the initial value to see it's effect.

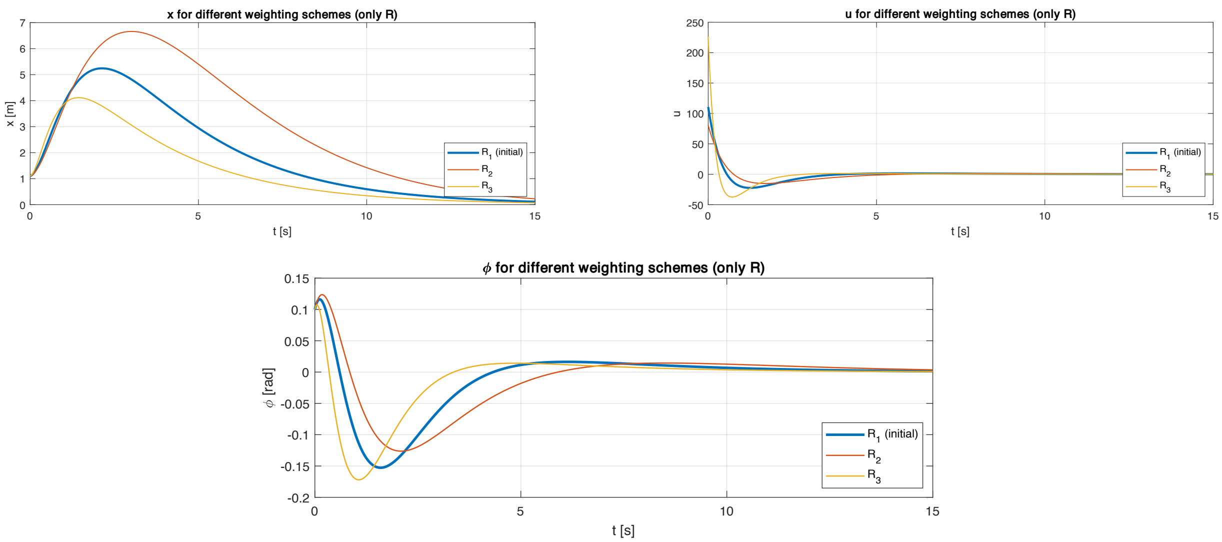


Figure 12: Effect Comparison of Different Weighting Schemes (only R is changed)

The increased $R_2 = 5$ resulted a smoother control by placing a higher penalty on the control effort while the decreased $R_3 = 0.1$ resulted opposite as expected.

Task 7 - Full State Observer Based on the Measurement of x

In this task, a full-state observer was designed based on the measurement of the displacement x, indicating the position of the cart. Different covariance matrices were experimented with to model process noise (σ_{W_d}) and measurement noise (σ_{W_n}). Three scenarios were considered:

- **Scenario 1:** $\sigma_{Wd_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$ (initial) and $\sigma_{Wn_1} = 0.1$ (initial)
- **Scenario 2:** $\sigma_{Wd_2} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and σ_{Wn_1} (initial)
  - *Expected Change:* Higher process noise in a different state (compared to Scenario 1) introduces increased uncertainty, influencing state estimation differently. Measurement noise remains consistent.
- **Scenario 3:** σ_{Wd_1} (initial) and $\sigma_{Wn_3} = 1.5$

- *Expected Change:* Higher measurement noise simulates conditions where measurement instruments have increased uncertainty, potentially leading to a less accurate state estimation.

Kalman filters were designed using the `lqe` and `ss` functions with different covariance matrices. The Kalman filters were implemented to estimate the state variables based on the available measurements.

The system (showed in Figure 13 with it's subsystem of Figure 14) response was simulated for each scenario, and plots were generated to visualize the performance of the observer.

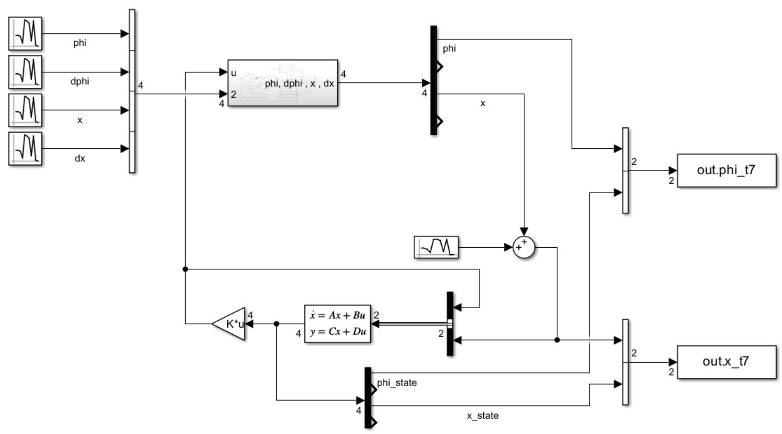


Figure 13: Simulink Model with Noise Input and Kalman Filter

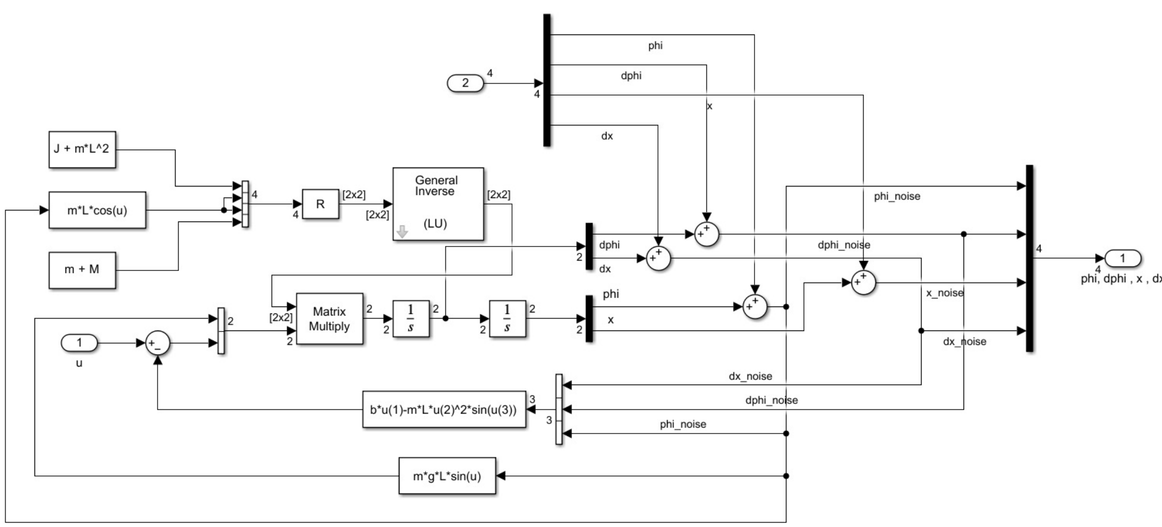


Figure 14: Nonlinear Simulink Subsystem with Noise Input

Plots included the true state and the estimated state for the angle signal (ϕ) and position signal (x) showed in Figure **??**.

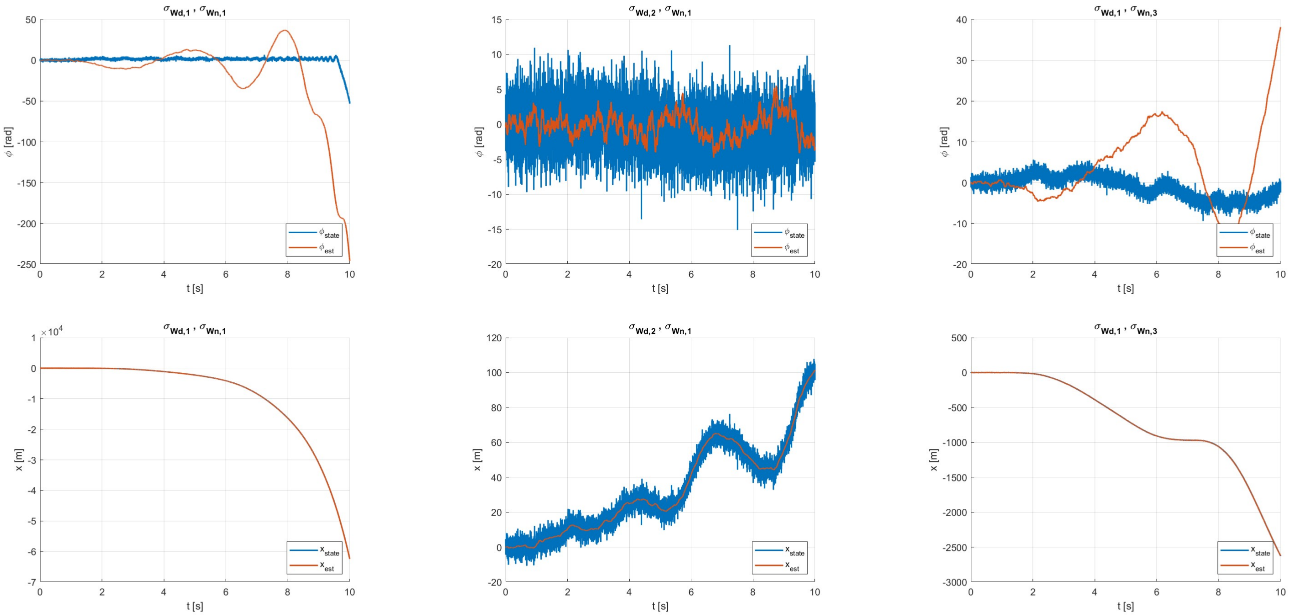


Figure 15: Full State Observer Signal with Different Noise Covariances

The impact of diverse covariance matrices can be succinctly described: employing different matrices alters process noise while maintaining identical estimation accuracy. Additionally, the simulation outcomes illustrated how variations in the covariance of measurement noise affected measurement precision, exerting a more significant influence on the estimation of ϕ due to its indirect measurement.

Task 8 - Trajectory tracking of the LQG controlled system

In this task, the trajectory tracking performance of an LQG-controlled system with nonzero reference signal compensation was analyzed with the help of a Simulink model showed in Figure 16. To adhere to a specified trajectory, a reference signal needs to be incorporated into the system. In this scenario, a square signal with time steps of 10 seconds is utilized, providing ample time for the system to respond effectively and approach a quasi-steady state between changes in the reference signal.

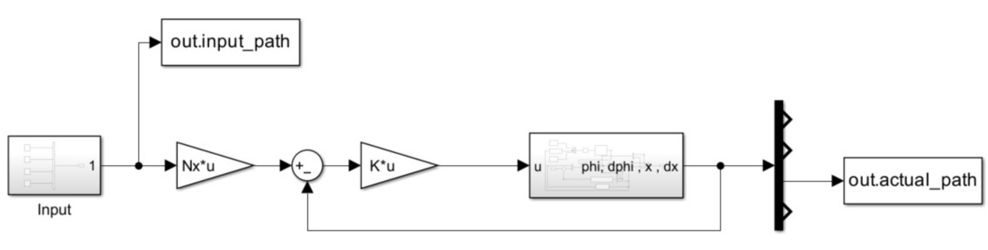


Figure 16: Simulink Model for Task 8

The code plots the square cart position trajectories, along with the actual system response using the LQG controller showed in Figure 17.

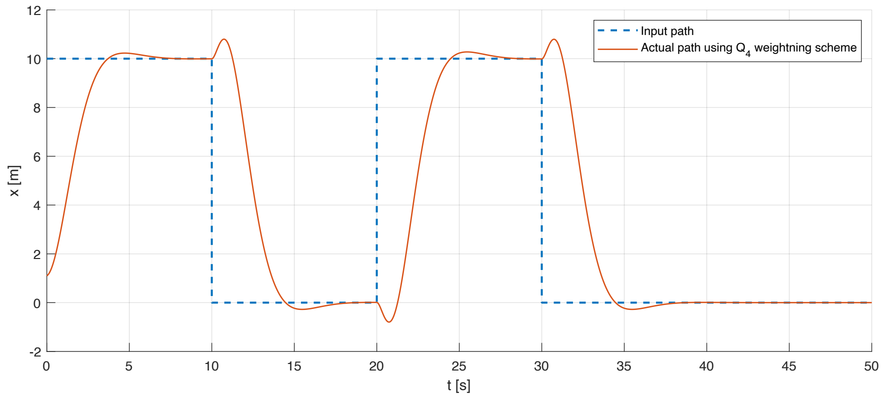


Figure 17: System's Trajectory Tracking

The trajectory tracking plot shows that the system is trying to follow the reference signal but it takes nearly 10 seconds to get back into the quasi-steady state. Other weighting schemes might produce more accurate trajectory tracking, however the task didn't clarified that as a goal.