Optimal Control

"The Stability Analysis of Inverted Pendulum using LQR approach"

2022-2023

Name: Mohammad Eqbal Balaghi

Registration: 242602

DIMES - Univ. della Calabria

blgmmm95e06z200t@studenti.unical.it

July 2023

Contents

Introduction

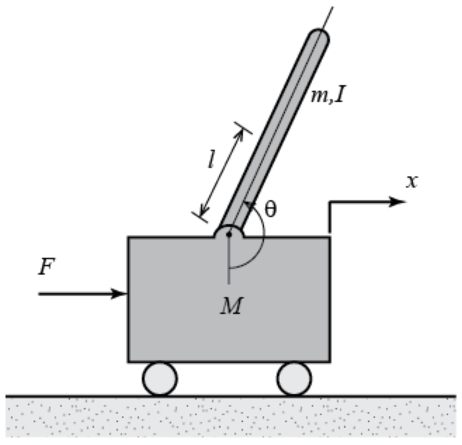
An inverted pendulum refers to a pendulum with its center of mass positioned above its pivot point. Due to its inherent instability, it will topple over without external assistance. However, it can be stabilized in the inverted position by utilizing a control system that monitors the pole's angle and adjusts the pivot point horizontally to maintain balance when it starts to fall. This setup serves as a classic problem in dynamics and control theory and serves as a standard for evaluating control strategies. Typically, the inverted pendulum is implemented with the pivot point attached to a controllable cart that moves horizontally through an electronic servo system, forming what is known as a cart and pole apparatus.

To ensure stability, the pendulum's movement is often limited to just one degree of freedom by fixing the pole to a rotational axis. While a regular pendulum is stable when hanging down, the inverted pendulum inherently requires active balancing to remain upright. This balance can be achieved by applying torque at the pivot point, moving the pivot point horizontally with a feedback system, altering the rotation rate of a mass mounted on the pendulum's axis parallel to the pivot axis to generate a net torque, or oscillating the pivot point vertically. A simple demonstration of moving the pivot point using a feedback system can be seen when balancing an upturned broomstick on the tip of one's finger.

Chapter 1: Model

Physical setup and system equations:

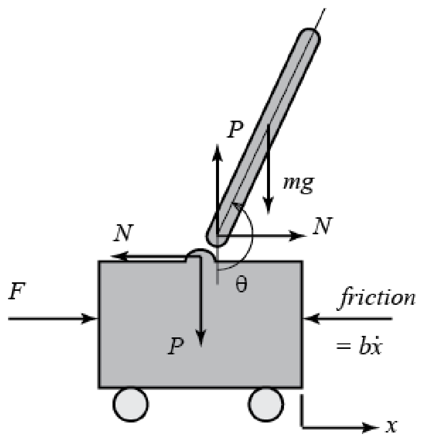
In this instance, we'll examine a two-dimensional variation of the inverted pendulum setup involving a cart. The pendulum's motion is restricted to the vertical plane, as depicted in the figure below. In this system, the control input is represented by the force denoted as F, which propels the cart horizontally. Meanwhile, the outputs of interest are the angular position of the pendulum, theta, and the horizontal position of the cart, x.



For this example, let's assume the following quantities:

(M)	mass of the cart	0.5 kg
(m)	mass of the pendulum	0.2 kg
(b)	coefficient of friction for cart	0.1 N/m/sec
(l)	length to pendulum center of mass	0.3 m
(I)	mass moment of inertia of the pendulum	0.006 kg.m^2
(F)	force applied to the cart	
(x)	cart position coordinate	
(theta)	pendulum angle from vertical (down)	

If we consider the free body diagram of the case we are analyzing, then we have:



We will generate the differential equations for these degrees of freedom from first principles employing Newton's second law ($F = ma$) as shown below:

$$\ddot{x} = \frac{1}{M}\sum_{cart} F_x = \frac{1}{M}(F - N - b\dot{x})$$

$$\ddot{\theta} = \frac{1}{I}\sum_{pend} \tau = \frac{1}{I}(-Nl\cos\theta - Pl\sin\theta)$$

To comprehensively represent the dynamics of the system, it becomes imperative to account for the interaction forces N and P between the cart and the pendulum. By incorporating these forces, the modeling must encompass both the x and y components of the translation of the pendulum's center of mass, in addition to its rotational dynamics.

Overall, the aim is to leverage Simulink's modeling capabilities to handle the algebraic aspects on our behalf. Consequently, we will construct the model to include the supplementary equations for the x and y components of the pendulum, as depicted below.

$$m\ddot{x}_p = \sum_{pend} F_x = N$$

$$\Rightarrow N = m\ddot{x}_p$$

$$m\ddot{y}_p = \sum_{pend} F_y = P - mg$$

$$\Rightarrow P = m(\ddot{y}_p + g)$$

Nonetheless, the position coordinates xp and yp are precise functions of theta. As a result, we can express their derivatives in relation to the derivatives of theta. Initially focusing on the x component equations, we obtain the following expressions:

$$x_p = x + l\sin\theta$$

$$\dot{x}_p = \dot{x} + l\dot{\theta}\cos\theta$$

$$\ddot{x}_p = \ddot{x} - l\dot{\theta}^2\sin\theta + l\ddot{\theta}\cos\theta$$

When we examine the y component equations, we obtain the following expressions:

$$y_p = -l\cos\theta$$

$$\dot{y}_p = l\dot{\theta}\sin\theta$$

$$\ddot{y}_p = l\dot{\theta}^2\cos\theta + l\ddot{\theta}\sin\theta$$

We can substitute these derived expressions into the previously mentioned equations for N and P in the following manner:

$$N = m(\ddot{x} - l\dot{\theta}^2\sin\theta + l\ddot{\theta}\cos\theta)$$

$$P = m(l\dot{\theta}^2\cos\theta + l\ddot{\theta}\sin\theta + g)$$

In addition, we can write the above formula as follows as well:

Summing the forces in the free-body diagram of the cart in the horizontal direction, you get the following equation.

$$M\ddot{x} + b\dot{x} + N = F$$

When analyzing the forces acting on the cart, summing the forces in the vertical direction wouldn't provide any valuable insights. Instead, by focusing on the free-body diagram of the pendulum and summing the forces in the horizontal direction, we can obtain the following expression for the reaction force N.

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta$$

Upon substituting the expression for the reaction force N into the first equation (which likely represents the dynamic equation for the cart and pole system), you will obtain one of the two governing equations that describe the behavior of this system. This equation is crucial in understanding and modeling the dynamics of the inverted pendulum system and its control.

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F$$

To derive the second equation of motion for the cart and pole system, we sum the forces perpendicular to the pendulum. Solving the system along this axis significantly simplifies the mathematical analysis. The resulting equation should be as follows:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta$$

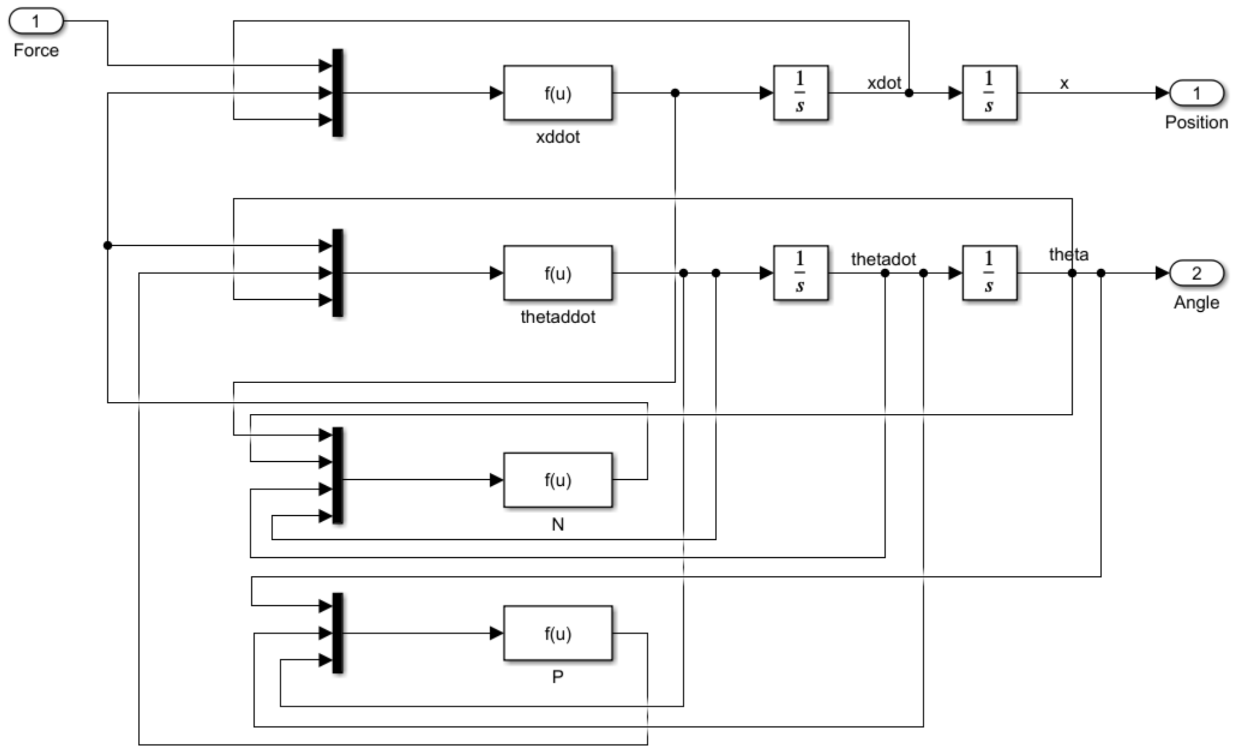
To eliminate the P and N terms from the equation derived earlier, we can sum the moments about the centroid of the pendulum. Then we combine the last two expression and we have the second governing equation:

$$(I+ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta$$

We can now represent these equations within Simulink.

Chapter 2: Building Simulink Model:

The resulting model according to the formulas above is as follow:



Linerization:

Linearization is the process of constructing a linear approximation of a nonlinear system, applicable within a small vicinity surrounding the operating point. The operating point represents a steady-state condition where all model variables remain constant. By employing linearization, it becomes possible to utilize tools developed for analyzing linear systems to study the behavior of nonlinear systems in the vicinity of a specific point. It is important to note that real-world dynamic systems, including the system being examined, are never completely linear.

we will linearize the equations about the vertically upward equillibrium position, $\theta = \pi$, and will assume that the system stays within a small neighborhood of this equillbrium. This assumption should be reasonably valid since under control we desire that the pendulum not deviate more than 20 degrees from the vertically upward position. Let ϕ represent the deviation of the pendulum's position from equilibrium, that is, $\theta = \pi + \phi$. Again presuming a small deviation (ϕ) from equilibrium, we can use the following small angle approximations of the nonlinear functions in our system equations:

$$\cos\theta = \cos(\pi + \phi) \approx -1$$

$$\sin\theta = \sin(\pi + \phi) \approx -\phi$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0$$

Upon replacing the aforementioned approximations into our nonlinear governing equations, we eventually arrive at the two linearized equations of motion.

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x}$$

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u$$

Transfer Functions:

To obtain the transfer functions of the linearized system equations, we need to take the Laplace transform of the system equations while assuming zero initial conditions. Then we are going to have:

$$P_{pend}(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}} \qquad [\frac{rad}{N}]$$

$$P_{cart}(s) = \frac{X(s)}{U(s)} = \frac{\frac{(I+ml^2)s^2 - gml}{q}}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \qquad [\frac{m}{N}]$$

Where:

$$q = [(M+m)(I+ml^2) - (ml)^2]$$

State Space Representation:

Absolutely, linearized equations of motion can be represented in state-space form, which is a common way to describe linear systems. By rearranging the equations into a series of first-order differential equations, the state-space representation can be achieved. The standard matrix form for the state-space representation is as follows:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

The reason why the matrix C consists of 2 rows is that it accounts for both the cart's position and the pendulum's position in the system's output. To be precise, the first element of the output vector Y corresponds to the cart's position, while the second element represents the pendulum's deviation from its equilibrium position.

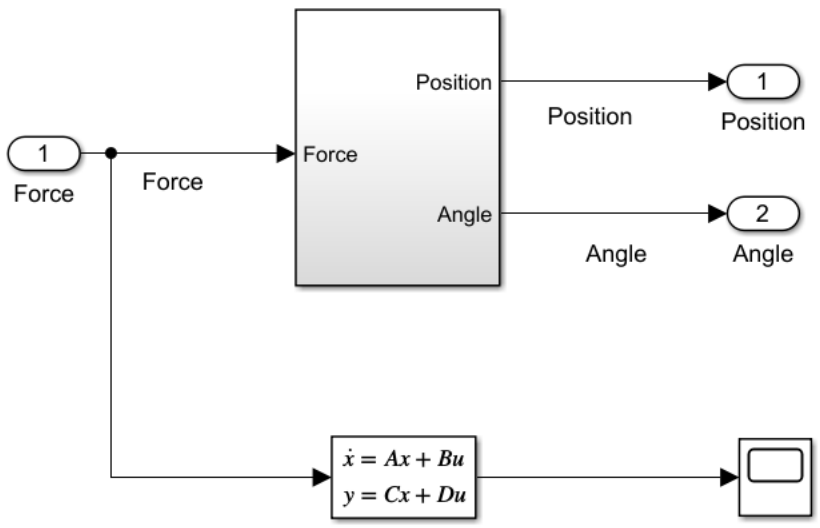
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.1818 & 2.6727 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.4545 & 31.1818 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

For this problem the outputs are the cart's displacement (x in meters) and the pendulum angle (ϕ in radians) where ϕ represents the deviation of the pedulum's position from equilibrium, that is, $\theta = \pi + \phi$

The design criteria for this system for a 0.2-m step in desired cart position x are as follows:

- Settling time for x and θ of less than 5 seconds
- Rise time for x of less than 0.5 seconds
- Pendulum angle θ never more than 20 degrees (0.35 radians) from the vertical
- Steady-state error of less than 2% for x and θ



Chapter 3: Synthesis

State feedback controller

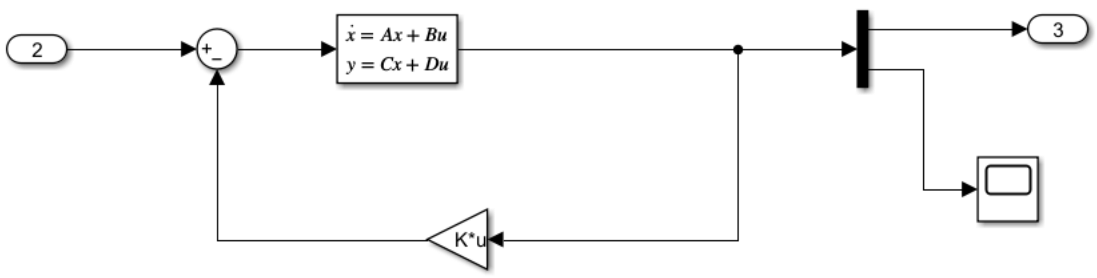
The first step in designing a full-state feedback controller is to determine the open-loop poles of the system. After execution in the MATLAB command window, the output will list the open-loop poles (eigenvalues of A) as shown below:

```
M = 0.5;

m = 0.2;

b = 0.1;

I = 0.006;

g = 9.8;

l = 0.3;



p = I*(M+m)+M*m*l^2; %denominator for the A and B matrices

A = [0      1              0           0;
     0 -(I+m*l^2)*b/p  (m^2*g*l^2)/p   0;
     0      0              0           1;
     0 -(m*l*b)/p       m*g*l*(M+m)/p  0];
B = [     0;
     (I+m*l^2)/p;
          0;
        m*l/p];
C = [1 0 0 0;
     0 0 1 0];
D = [0;
     0];

states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'u'};
outputs = {'x'; 'phi'};

sys_ss = ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);

poles = eig(A)
```

```
poles =

     0

  -0.1428

  -5.6041

   5.5651
```

The presence of a right-half plane pole at the value of 5.5651 is evident from the results. This observation aligns with your intuition that the system is unstable in the open-loop configuration.

The full-state feedback control technique can be employed to address this problem. The control system schematic, depicted below, incorporates a matrix of control gains denoted as K. It is essential to highlight that in this approach, all the states of the system are fed back for control, as opposed to utilizing the system's outputs for feedback.



Linear Quadratic Regulation (LQR)

In the subsequent phase of the design process, we aim to determine the vector of state-feedback control gains, denoted as K. This assumes that all four state variables are accessible, meaning they can be measured. There are several approaches to achieve this. If the desired locations of closed-loop poles are known, MATLAB offers the options of using the commands "place" or "acker." Alternatively, one can utilize the "lqr" command, which yields the optimal controller gain under the assumptions of a linear plant, a quadratic cost function, and a reference equal to zero.

Before we design our controller, we will first verify that the system is **controllable.**

the controllability matrix must have rank n where the rank of a matrix is the number of linearly independent rows (or columns). The controllability matrix of the system takes the form shown below:

$$\mathcal{C} = \left[B \ AB \ A^2B \ \cdots \ A^{n-1}B\right]$$

By using the command below, we get to this:

```
co = ctrb(sys_ss);

controllability = rank(co)

controllability =

    4
```

Hence, we have verified the controllability of our system, indicating that designing a controller to meet the specified requirements is feasible. To achieve this, we will employ the linear quadratic regulation (LQR) method to compute the state-feedback control gain matrix K. By using the MATLAB function "lqr," you can select two parameters, R and Q, which will influence the trade-off between the importance of control effort (U) and error (deviation from 0) in the cost function being optimized.

In the simplest case, we assume R=1 and *Q=C'C*, where *C* represents the output matrix. This choice of *R* and *Q* assigns equal significance to the control and the state variables, which, in this context, are the outputs, i.e., the pendulum's angle and the cart's position. Consequently, the LQR method allows control over both outputs, and the process is relatively straightforward. By modifying the nonzero elements in the *Q* matrix, the controller can be tuned to achieve the desired response.

To examine the structure of *Q*, input the following command into the MATLAB command window to observe the output displayed below.

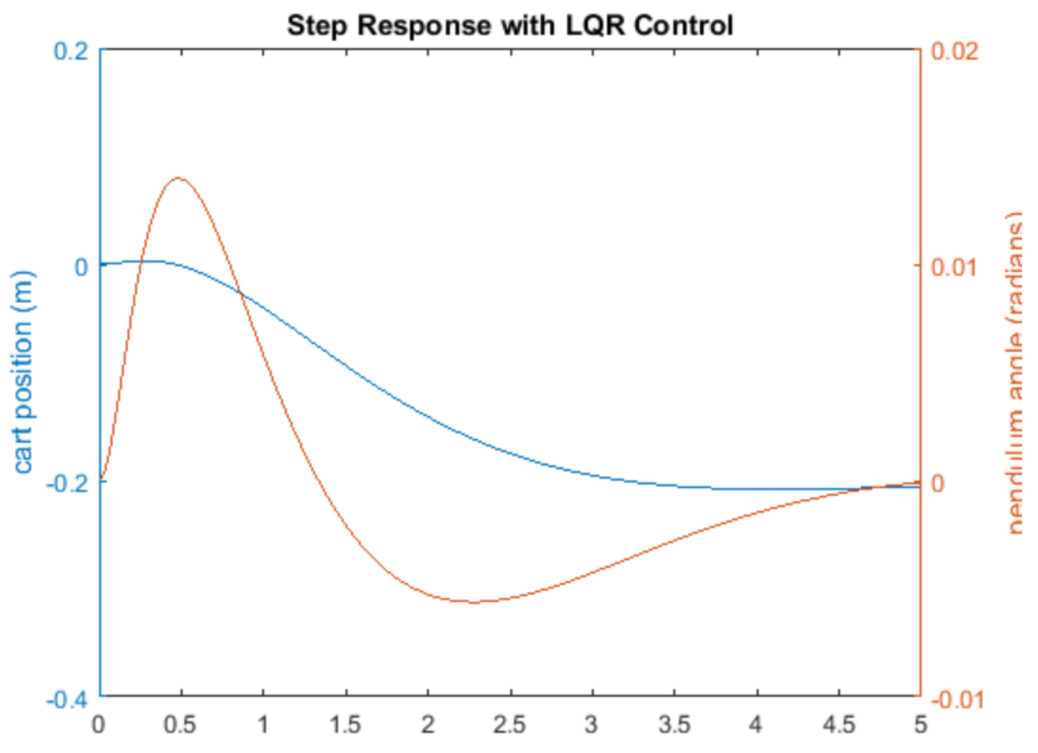
```
Q = C'*C

Q =

    1    0    0    0

    0    0    0    0

    0    0    1    0

    0    0    0    0
```

The element at position (1,1) of *Q* indicates the weight placed on the cart's position, while the element at position (3,3) represents the weight on the pendulum's angle. As for the input weighting *R*, it will remain constant at 1. What truly matters is the relative values of *Q* and *R*, rather than their absolute magnitudes. Having understood how to interpret the *Q* matrix, we can now proceed with experimentation to find the *K* matrix that will yield a "good" controller, effectively meeting our desired control objectives.

By using the code below we are going to have:

```
Q = C'*C;

R = 1;

K = lqr(A,B,Q,R)



Ac = [(A-B*K)];

Bc = [B];

Cc = [C];

Dc = [D];



states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'r'};
outputs = {'x'; 'phi'};

sys_cl = ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',inputs,'outputname',outputs);

t = 0:0.01:5;
r =0.2*ones(size(t));
[y,t,x]=lsim(sys_cl,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(get(AX(2),'Ylabel'),'String','pendulum angle (radians)')
title('Step Response with LQR Control')
```

```
K =

   -1.0000   -1.6567   18.6854    3.4594
```



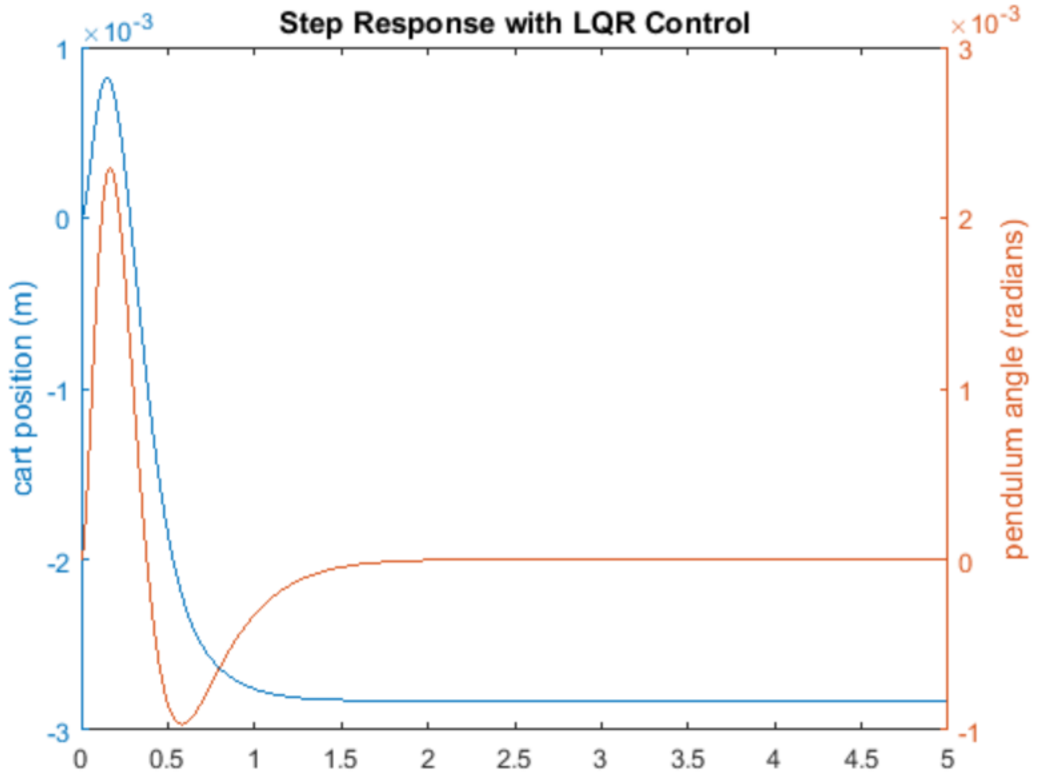
The red curve on the plot represents the pendulum's angle in radians, while the blue curve corresponds to the cart's position in meters. As evident from the plot, the current response is unsatisfactory. While the overshoot of the pendulum and cart appears acceptable, the settling times require improvement, and the cart's rise time needs to be reduced. Additionally, it is notable that the cart's final position deviates in the opposite direction from the desired location.

To address these issues, focus on modifying the *Q* matrix in your m-file to obtain a better response. Increasing the elements at positions (1,1) and (3,3) in *Q* will lead to decreased settling and rise times but may also result in reduced pendulum movement (angle). Essentially, you will be emphasizing the significance of the errors at the expense of increased control effort *u*.

Consider updating your code so that the element at (1,1) of *Q* is set to 5000 and the element at (3,3) is set to 100. This adjustment will yield a new value for *K* and produce the step response shown below:

```
Q = C'*C;

Q(1,1) = 5000;
```

```
Q(3,3) = 100

R = 1;

K = lqr(A,B,Q,R)



Ac = [(A-B*K)];

Bc = [B];

Cc = [C];

Dc = [D];



states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'r'};
outputs = {'x'; 'phi'};

sys_cl = ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',inputs,'outputname',outputs);

t = 0:0.01:5;
r =0.2*ones(size(t));
[y,t,x]=lsim(sys_cl,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(get(AX(2),'Ylabel'),'String','pendulum angle (radians)')
title('Step Response with LQR Control')
```



As you might have observed, raising the values of the elements in *Q* even further could lead to further enhancements in the system's response. However, the current choice of weighting in *Q* was specifically made to meet the transient design requirements adequately. Increasing the magnitude of *Q* beyond its current level would indeed reduce the tracking error, but it would necessitate a higher control force *u*. Such increased control effort usually corresponds to greater costs, including higher energy consumption and the need for larger actuators, among other factors. Therefore, the decision to strike a balance between maximizing system performance and managing control effort becomes crucial in the design process, considering the practical constraints and cost implications of the control system.

Chapter 4: Conclusion

In conclusion, the LQR (Linear Quadratic Regulator) approach in optimal control offers several advantages when compared to pole placement techniques. While both methods aim to design control systems for desired closed-loop pole locations, the LQR approach provides a more comprehensive and flexible framework.

Advantages of LQR over pole placement:

1. System Performance Optimization: The LQR approach optimizes a quadratic cost function, considering both control effort and state error. This optimization enables better control system performance, leading to minimized overshoot, faster settling times, and improved disturbance rejection.

2. Trade-off Flexibility: With LQR, engineers can adjust the relative importance of control effort and state error through the Q and R weighting matrices. This flexibility allows for fine-tuning the controller's behavior to meet specific design requirements effectively.

3. Handling Uncontrollable Systems: LQR can be applied to systems that are not fully controllable, while pole placement requires full controllability for precise pole placement. LQR compensates for uncontrollable modes and stabilizes the system, making it more versatile for a wider range of control scenarios.

4. Robustness: The LQR controller inherently considers uncertainties and disturbances by minimizing the cost function across various operating conditions. This robustness leads to a more reliable and stable control system in real-world applications.

5. Simplicity in Design: LQR provides a straightforward and efficient way to design optimal controllers without the need for explicit pole calculation. By focusing on the state-space representation, the controller design process is often simplified and less computationally intensive.

While pole placement can be useful in specific cases where the desired closed-loop poles are known a priori, the LQR approach offers a more versatile and robust control design methodology. Its ability to optimize system performance, flexibility in tuning control behavior, and robustness to uncertainties make it a favored choice for many practical control applications.