

# The LQR Controller

Muhammed Sezer<sup>a,b</sup> and Şevval Belkıs Dikkaya<sup>a,b</sup>

<sup>a</sup>Middle East Technical University; <sup>b</sup>Orion Robotics Ltd.

June 3, 2023

This guide provides an easy-to-follow overview of the Linear Quadratic Regulator (LQR) controller. It covers both practical and theoretical aspects, allowing you to run simulations and apply it in real-world scenarios. The guide includes explanations of underlying principles, practical applications, and a step-by-step computer simulation demonstration. Whether you're a beginner or an experienced practitioner, this comprehensive guide aims to showcase how the LQR controller can effectively tackle complex control problems.

Linear Quadratic Regulator | LQR | Control Theory | System Control | Practical Application | Simulation | Performance Analysis | Case Study

## 1. Introduction

**A. Overview of LQR.** Once upon a time, in the mid-20th century, the field of control systems was undergoing a revolution. Engineers and mathematicians were seeking new ways to manage complex, dynamic systems. Amidst this backdrop, the Linear Quadratic Regulator (LQR) was born, a brainchild of the need for optimal control strategies.

The LQR was first introduced in the 1960s, during a period of rapid advancement in aerospace technology. The space race was in full swing, and the need for precise, reliable control systems for spacecraft was paramount. The traditional methods of control system design were not sufficient for these high-stakes, complex tasks. The world needed a new approach, and the LQR provided just that.

The LQR was a game-changer. It provided a systematic, principled approach to control system design. By defining a cost function that quantified the trade-off between system performance and control effort, and then minimizing this cost function, the LQR could determine the optimal control law for a given system.

Over the years, the LQR has proven its worth in a wide range of applications, from aerospace and robotics to economics and beyond. It is particularly well-suited for systems that can be accurately modeled by linear equations, and where the cost to be minimized can be expressed as a quadratic function. However, even for non-

linear systems, LQR can often provide useful results when applied to a linear approximation of the system around a particular operating point.

In the following sections, we will delve deeper into the story of LQR, its applications, and its implementation. We will start with a specific problem, then introduce the LQR as a solution, and finally explore the theory behind it. We will also discuss how to implement an LQR solution and analyze the results of a simulation. So, buckle up and get ready for an exciting journey into the world of LQR!

**B. Importance and Applications of LQR.** The Linear Quadratic Regulator (LQR) has been a cornerstone in the field of control systems engineering due to its systematic and optimal approach to control system design. Its ability to balance system performance against control effort has made it a valuable tool across a multitude of industries.

The parameters used in LQR control design are primarily the weights in the quadratic cost function. These weights are chosen based on the specific requirements of the system being controlled. They reflect the relative importance of achieving a particular system state versus the amount of control effort expended.

Let's delve into some specific applications and the parameters typically used:

### Significance Statement

This document's significance lies in its comprehensive exploration of the Linear Quadratic Regulator (LQR) controller, a key tool in the realm of system control.

<sup>1</sup> Muhammed Sezer and Şevval Dikkaya contributed equally to this work.

<sup>2</sup> To whom correspondence should be addressed. E-mail: muhammedsezer12@gmail.com

1. **Aerospace:** Control systems for aircraft and spacecraft prioritize achieving specific system states like altitude or speed while minimizing control effort.
2. **Automotive:** Automotive applications such as cruise control and active suspension systems balance ride comfort and fuel efficiency.
3. **Robotics:** LQR control is used in robotic arms, drones, and autonomous vehicles to optimize task precision and energy efficiency.
4. **Energy Sector:** LQR optimizes power system operation by balancing stable power output and minimizing operational costs.

The weights in LQR design vary based on the system's requirements and constraints in each application. In the upcoming sections, we will explore the theory and implementation of LQR, providing insights into the selection of these parameters and their impact on the controller's performance.

**C. Prerequisites.** Knowledge in the following areas is assumed for this course:

- **Linear Algebra:** Understanding of matrices, vectors, and operations on them is essential as the state-space representation of systems, which is central to LQR, is expressed in matrix form.
- **Differential Equations:** Familiarity with ordinary differential equations is crucial as the dynamics of the systems we'll be controlling are often expressed as such equations.
- **Control Systems:** Basic understanding of control systems, particularly state-space representation of systems, is required. This includes knowledge of system dynamics, stability, and controllability.
- **Laplace Transforms:** Familiarity with Laplace transforms is helpful for understanding the analysis and design of control systems.

## 2. Example Problem

**A. Problem Statement: The Inverted Pendulum.** Our challenge is the inverted pendulum problem, a topic that has intrigued researchers from the past to the present. With its canonical and nonlinear structure, the inverted pendulum problem has been a popular subject for control theories and applications.

In this study, the nonlinear mathematical model of the pendulum is linearized, and control is performed with full state feedback.

When designing the controller, the goal is to keep the pendulum balanced vertically and bring the cart back to the starting point. You can see the pendulum system at figure 1

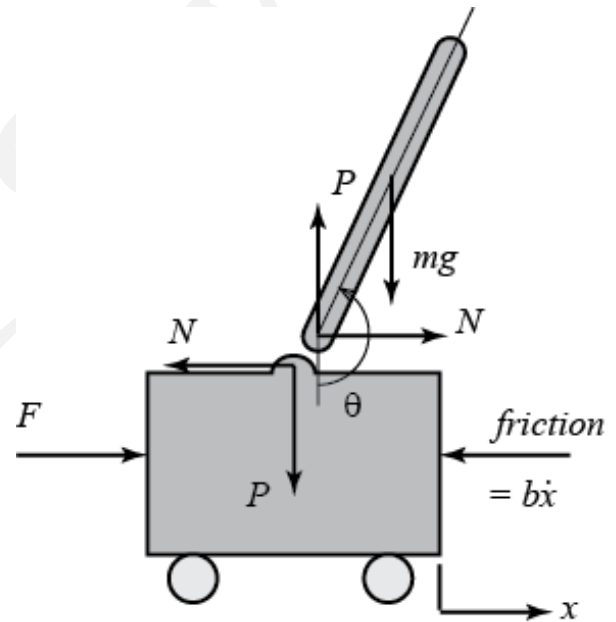


Fig. 1. Inverted Pendulum System

**B. Problem Analysis.** The system consists of a cart of mass  $M$  that can move horizontally and a pendulum of length  $l$  and mass  $m$  attached to the cart. The pendulum is free to rotate and its angle with the vertical, denoted by  $\theta$ , is a critical parameter. The pendulum is in its unstable equilibrium position when  $\theta$  is 180 degrees, i.e., when it's perfectly vertical.

The control input to the system is a force  $F$  applied horizontally to the cart. The goal is to adjust this force to keep the pendulum upright, i.e., to keep  $\theta$  as close to 180 degrees as possible.

131	However, the system is nonlinear and under-	177	tual final state that the system reaches after
132	actuated, meaning it has more degrees of free-	178	a long time. For our system, we want the
133	dom (two, in this case: the position of the cart	179	steady-state error to be less than 2% for both
134	and the angle of the pendulum) than it has in-	180	$x$ and $\theta$ .
135	dependent control inputs (one, in this case: the		
136	force applied to the cart). This makes the control		
137	problem challenging.		
138	Moreover, the system is highly sensitive to dis-	181	In summary, our goal is to design a controller
139	turbances and changes in the initial conditions.	182	that can meet these performance requirements,
140	A small push to the pendulum or a small change	183	effectively balancing the pendulum in its upright
141	in the initial angle can cause the pendulum to	184	position while controlling the position of the
142	fall over.	185	cart.
143	In the next section, we will define the specific		
144	goals and objectives for the control problem.		
145	<b>C. Goals and Objectives.</b> In this section, we will	186	<b>3. Solution! LQR</b>
146	define the specific goals and objectives for the		
147	control of the inverted pendulum system. Our	187	<b>A. Introduction to LQR as a Solution.</b> We will
148	aim is to control both the pendulum's angle $\theta$	188	design a controller for our inverted pendulum
149	and the cart's position $x$ .	189	system to meet the performance requirements.
150	To ensure the performance of our system,	190	The chosen method is the Linear Quadratic
151	we need to set some quantitative requirements.	191	Regulator (LQR), which optimally designs con-
152	These requirements will serve as our definition	192	trollers for linear systems. The LQR minimizes a
153	of a "good" system. In control theory, we of-	193	quadratic cost function by adjusting the control
154	ten use specific performance measures to define	194	input based on the system state. This allows us
155	these requirements. For our inverted pendulum	195	to keep the pendulum upright, control the cart's
156	system, we will focus on the following perfor-	196	position, and minimize control effort. In the up-
157	mance measures:	197	coming sections, we will explore the theory and
158		198	implementation of LQR for our problem.
159		199	<b>B. Benefits of Using LQR.</b> The Linear Quadratic
160		200	Regulator (LQR) offers several benefits that make
161		201	it a powerful and popular method for control
162		202	system design:
163		203	
164		204	
165		205	
166		206	
167		207	
168		208	
169		209	
170		210	
171		211	
172		212	
173		213	
174		214	
175		215	
176		216	
		217	
		218	
		219	

**C. Limitations of LQR.** While the Linear Quadratic Regulator (LQR) is a powerful method for control system design, it's important to be aware of its limitations:

- **Linearity Assumption:** LQR is designed for linear systems. If the system is nonlinear, as is the case with our inverted pendulum problem, it needs to be linearized around an operating point. This means that the performance of the LQR controller may degrade if the system deviates significantly from this operating point.
- **Quadratic Cost Function:** LQR minimizes a quadratic cost function. If the actual cost associated with the system's behavior is not well-represented by a quadratic function, the LQR controller may not provide the best performance.
- **Full State Accessibility:** LQR assumes that all states of the system are measurable and available for feedback. If some states are not measurable, additional techniques such as state estimation may be needed.
- **Deterministic Systems:** LQR is designed for deterministic systems. If there are random disturbances or uncertainties in the system, a stochastic control approach may be more appropriate.

Despite these limitations, LQR can still provide a good solution for many control problems, including our inverted pendulum problem, especially when used in combination with other techniques to address these limitations.

## 4. Theory

**A. Mathematical Foundations of LQR.** The state-space representation of a system is a mathematical model of a physical system as a set of input, output, and state variables related by first-order differential equations. It is written in matrix form and allows us to analyze and control systems. The matrices A, B, C, and D are part of the state-space representation of the system:

- **The A matrix**, also known as the system matrix, relates the state vector to its derivative.

It represents the dynamics of the system — how the system evolves over time in the absence of input. The main logic is that we can express the derivative of the state vector by its current values. For example, the element at the second row and third column of the A matrix represents how the angle of the pendulum affects the velocity of the cart.

- **The B matrix**, also known as the input matrix, relates the control input to the state vector. It represents how the input affects the evolution of the system. For example, the element at the second row of the B matrix represents how the force applied to the cart affects the velocity of the cart.
- **The C matrix**, also known as the output matrix, relates the state vector to the output. It represents how the states of the system affect the output. In our case, the C matrix is a diagonal matrix, meaning that each output is directly affected by only one state.
- **The D matrix**, also known as the feedforward matrix, directly relates the control input to the output, bypassing the state. In many physical systems, including ours, this matrix is often the zero matrix, meaning that the input does not directly affect the output.

In the next sections, we will delve deeper into the theory behind LQR and how to implement it for our problem.

But first let's understand what these matrices are and how to use them.

**B. State-Space Representation.** The state-space representation of our system is given by the following matrices:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \quad [1]$$

$$B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \quad [2]$$

The state-space representation of the system is given by the equation:

$$\dot{x} = Ax + Bu \quad [3]$$

This can be expanded as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} u \quad [4]$$

So, for example, the derivative of the first state variable  $x_1$  (the position of the cart) is given by:

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + A_{14}x_4 + B_1u \quad [5]$$

This equation shows how the current state of the system and the control input  $u$  (the force applied to the cart) determine the rate of change of the position of the cart.

The elements of the  $A$  and  $B$  matrices in the state-space representation, such as  $A_{11}$ ,  $A_{12}$ ,  $B_1$ , and so on, are real numbers. These numbers are determined by the physical parameters of the system, such as the masses of the cart and the pendulum, the length of the pendulum, and the gravitational acceleration. The fact that we can directly find the derivative of the state vector by multiplying these numbers with the current state and input vectors is a characteristic of linear systems.

**C. Cost Function.** The Linear Quadratic Regulator (LQR) method is based on the principle of minimizing a quadratic cost function. This cost function is typically a weighted sum of the system states and control inputs, reflecting the trade-off between system performance (which we want to maximize) and control effort (which we want to minimize).

The cost function is usually defined as:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad [6]$$

where:

- $x$  is the state vector,
- $u$  is the control input vector,

•  $Q$  is a positive semi-definite matrix that weights the states,

•  $R$  is a positive definite matrix that weights the control inputs,

The matrices  $Q$  and  $R$  are chosen by the designer and reflect the relative importance of the states and control inputs in the cost function. They can be thought of as tuning knobs or a GUI for the LQR controller. For example, if a particular state is more important to control accurately, the corresponding element in the  $Q$  matrix would be set to a higher value. Similarly, if a particular control input needs to be used sparingly, the corresponding element in the  $R$  matrix would be set to a higher value.

The term  $x^T Q x$  in the cost function represents the cost associated with the states of the system. For example, if we have a 2-dimensional state vector  $x = [x_1, x_2]$ , and a 2x2 matrix  $Q = [Q_{11}, Q_{12}; Q_{21}, Q_{22}]$ , then:

$$x^T Q x = x_1^2 Q_{11} + x_1 x_2 (Q_{12} + Q_{21}) + x_2^2 Q_{22} \quad [7]$$

This shows that the cost is a weighted sum of the squares of the states and their cross product, where the weights are the elements of the  $Q$  matrix.

Similarly, the term  $u^T R u$  in the cost function represents the cost associated with the control inputs.

The goal of the LQR method is to find the control input  $u$  that minimizes this cost function. In the next section, we will discuss how this is achieved by solving the Riccati equation.

**D. Riccati Equation.** The solution to the LQR problem is given by the solution to the Algebraic Riccati Equation (ARE). The ARE is a type of nonlinear equation that arises in the context of infinite-horizon optimal control problems like the LQR problem.

For the LQR problem, the ARE takes the following form:

$$0 = A^T P + P A - P B R^{-1} B^T P + Q \quad [8]$$

where:

•  $A$ ,  $B$ ,  $Q$ , and  $R$  are the matrices defined in the state-space representation and the cost function,

•  $P$  is the solution to the ARE, a symmetric positive definite matrix that determines the optimal state feedback gain.

The solution to the ARE can be found using various numerical methods. Once the matrix  $P$  is found, the optimal state feedback gain  $K$  can be calculated as:

$$K = R^{-1}B^TP \quad [9]$$

This gain matrix  $K$  is then used in the state feedback controller to determine the control input  $u$  that minimizes the cost function:

$$u = -Kx \quad [10]$$

This gain matrix  $K$  is then used in the state feedback controller to determine the control input  $u$  that minimizes the cost function. For example, if we have a 2-dimensional state vector  $x = [x_1, x_2]$  and a 2-dimensional gain vector  $K = [k_1, k_2]$ , then the control input is given by:

$$u = -Kx = -k_1x_1 - k_2x_2 \quad [11]$$

This equation shows that the control input is a weighted sum of the states, where the weights are the elements of the gain vector  $K$ . So  $u$  is just a number.

**E. What is  $K$  and its effect.** The gain matrix  $K$  in the LQR controller plays a crucial role in determining the behavior of the controlled system. It is calculated as  $K = R^{-1}B^TP$ , where  $P$  is the solution to the Riccati equation, and  $R$  and  $B$  are the matrices defined in the state-space representation and the cost function.

The control input  $u$  is given by  $u = -Kx$ , which means that the control input is a linear function of the state. The gain matrix  $K$  determines how much each state variable contributes to the control input.

When the LQR controller is applied to the system, the state-space equation  $\dot{x} = Ax + Bu$  becomes  $\dot{x} = (A - BK)x$ . This means that the LQR controller effectively changes the system matrix  $A$  into a new matrix  $A - BK$ .

The stability of the controlled system depends on the eigenvalues of this new matrix. If all the eigenvalues have negative real parts, then the system is stable. The LQR method is designed to find the gain matrix  $K$  that makes the system stable and minimizes the cost function.

Let's consider a simple example where the system matrix  $A$  and the gain matrix  $K$  are 2x2 matrices:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad K = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \quad [12]$$

And  $B$  is a 2x1 matrix (column vector):

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \quad [13]$$

The new system matrix after applying the LQR controller is:

$$A - BK = \begin{bmatrix} A_{11} - B_1k_1 & A_{12} - B_1k_2 \\ A_{21} - B_2k_1 & A_{22} - B_2k_2 \end{bmatrix} \quad [14]$$

## 5. Solution Implementation

**A. System Modeling.** In this section, we will model our system using the state-space representation. The state-space representation of a system is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations. To represent our system, we need to define the matrices  $A$ ,  $B$ ,  $C$ , and  $D$ , which represent the system dynamics, the control input, the output, and the direct transmission term, respectively.

This system is challenging to model because of the physical constraint (the pin joint) between the cart and pendulum which reduces the degrees of freedom in the system. Both the cart and the pendulum have one degree of freedom ( $x$  and  $\theta$ , respectively). We will generate the differential equations for these degrees of freedom from first principles employing Newton's second law ( $F = ma$ ).

The equations of motion for the system are:

$$\ddot{x} = \frac{1}{M}(F - N - b\dot{x}) \quad [15]$$

$$\ddot{\theta} = \frac{1}{I}(-Nl \cos \theta - Pl \sin \theta) \quad [16]$$

It is necessary to include the interaction forces  $N$  and  $P$  between the cart and the pendulum in order to fully model the system's dynamics. The inclusion of these forces requires modeling the  $x$ - and  $y$ -components of the translation of the pendulum's center of mass in addition to its rotational dynamics.

The additional  $x$ - and  $y$ -component equations for the pendulum are:

$$N = m\ddot{x}_p \quad [17]$$

$$P = m(\ddot{y}_p + g) \quad [18]$$

However, the position coordinates  $x_p$  and  $y_p$  are exact functions of  $\theta$ . Therefore, we can represent their derivatives in terms of the derivatives of  $\theta$ . The equations for the derivatives of  $x_p$  and  $y_p$  are:

$$\dot{x}_p = \dot{x} - l\dot{\theta}^2 \sin \theta + l\ddot{\theta} \cos \theta \quad [19]$$

$$\dot{y}_p = l\dot{\theta}^2 \cos \theta + l\ddot{\theta} \sin \theta \quad [20]$$

These expressions can then be substituted into the expressions for  $N$  and  $P$  to give:

$$N = m(\ddot{x} - l\dot{\theta}^2 \sin \theta + l\ddot{\theta} \cos \theta) \quad [21]$$

$$P = m(l\dot{\theta}^2 \cos \theta + l\ddot{\theta} \sin \theta + g) \quad [22]$$

We can now represent these equations within a simulation environment. The simulation environment can work directly with nonlinear equations, so it is unnecessary to linearize these equations.

**B. State Space Deriving.** The linearized equations of motion can also be expressed in a state-space form by rearranging them into a set of first-order differential equations. Since the equations are linear, they can be represented in the standard matrix form as follows:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u \quad [23]$$

The output vector,  $y$ , is defined as:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad [24]$$

In the  $C$  matrix, the first row represents the cart's position while the second row represents the pendulum's deviation from its equilibrium position. The cart's position is the first element of the output vector  $y$ , and the pendulum's deviation from equilibrium is the second element of  $y$ .

**C. State-Space Equations.** From the main problem, the dynamic equations of the inverted pendulum system in state-space form are the following:

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

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad [2]$$



## 6. System Simulations

We employ Python, a powerful high-level programming language, to simulate our system and gain insights into its behavior under different initial conditions.

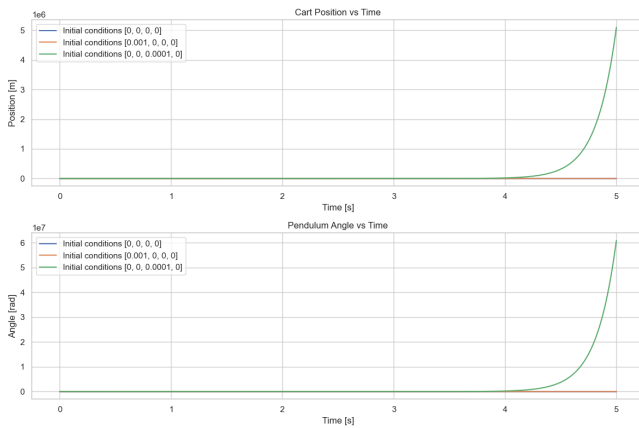
In the first scenario, where all parameters are set to zero, the system remains in a steady-state equilibrium. This outcome is expected since no perturbations or stimuli are introduced, indicating system stability.

In the second scenario, a small disturbance is introduced in the initial angle. This slight change leads to a significant variation in the system's behavior. Over time, the effect of the initial angle disturbance amplifies exponentially, characteristic of unstable systems.

It is crucial to note that the simulation predicts unbounded growth, but this is limited to the linearized approximation used in our model. As the system deviates further from the equilibrium, the linear approximation becomes less accurate. In reality, the system would exhibit oscillations rather than unbounded growth.

This highlights the limitations of linear models and emphasizes the importance of understanding the true non-linear dynamics of the system. Nonetheless, linear models provide valuable insights and are effective for control design, especially when the system operates near a specific operating point.

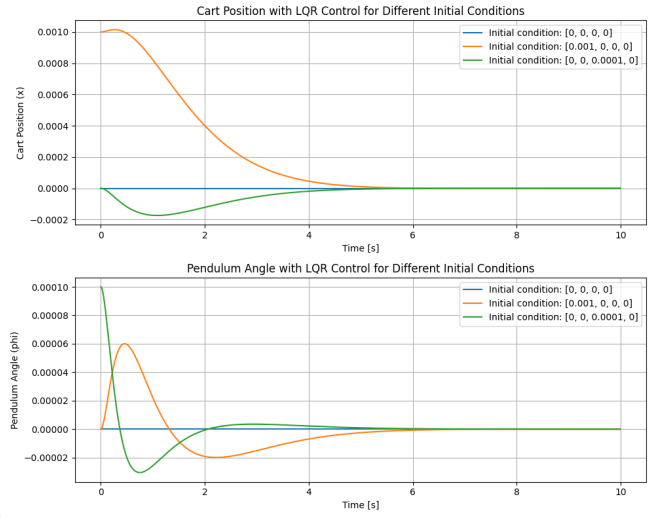
As shown in Figure 2, the system simulation results illustrate these points clearly.



**Fig. 2.** Simulation of the system under different initial conditions. The blue line represents the system starting at a steady state (0,0,0,0). The orange line represents the system with a slight disturbance in velocity (0.001,0,0,0). The green line depicts the system with a disturbance in angle (0,0,0.0001,0).

## 7. Implementation of LQR Controller

The LQR gain matrix,  $K$ , is computed using the control.lqr function from the Python Control Systems Library. This function solves the continuous or discrete Algebraic Riccati equation based on the system type. With  $K$  obtained, the control law  $u = -Kx$  is implemented, updating the control inputs at each time step. The LQR controller successfully stabilizes the pendulum in the upright position, as shown in Figure 3.



**Fig. 3.** Response of the system under the control of the LQR controller.

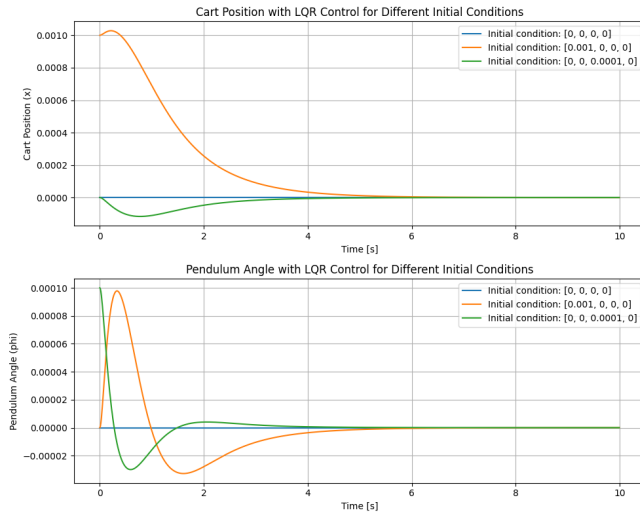
The implementation of the LQR controller exhibits its ability to handle multi-input, multi-output (MIMO) systems, while ensuring stability and robustness. It does this by finding the optimal control law that minimizes the cost function, balancing between the control effort and the state deviation. This makes LQR a versatile and powerful controller for a variety of systems.

## 8. Exploring Different LQR Parameters

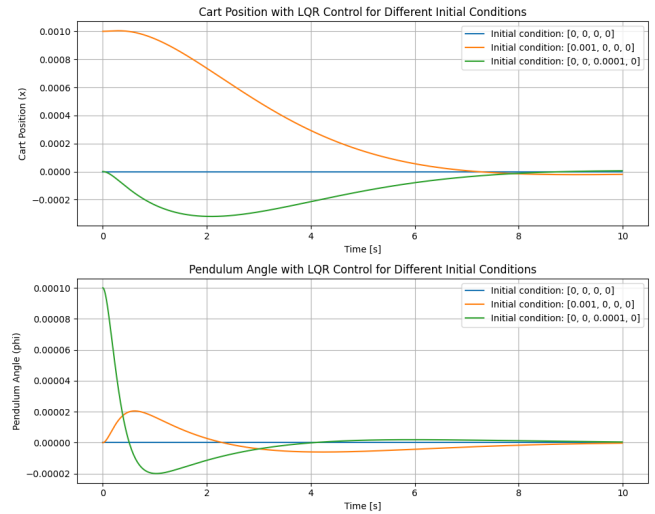
In LQR controller design, we can adjust the weighting matrices  $Q$  and  $R$  to emphasize certain control objectives. The matrix  $Q$  is used to weigh the states, whereas  $R$  is used to weigh the control efforts.

As a result, changing these matrices affects the control behavior of the system. We performed simulations using high values for elements of the  $Q$  matrix (high  $Q$ ) and high values for elements of the  $R$  matrix (high  $R$ ) to illustrate their impact on system behavior.





**Fig. 4.** System response with high  $Q$ . In this case, the states are heavily penalized, and the controller works hard to keep the states near the desired values. This may lead to aggressive control action.

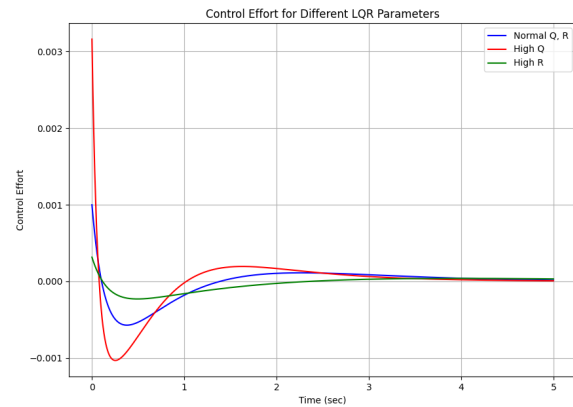


**Fig. 5.** System response with high  $R$ . In this scenario, the control effort is heavily penalized, and the controller avoids large control actions. This might lead to slower convergence to the desired state, but it also means less energy consumption and less aggressive behavior.

As mentioned earlier, changing these matrices affects the control behavior of the system. Here, we explore different LQR parameters and their impact on control effort. Control effort refers to the magnitude of the control signal required to stabilize the system.

Figure 6 demonstrates how different LQR parameter settings affect the control effort. Smaller control efforts indicate less aggressive control actions, prioritizing energy efficiency and minimizing mechanical stress. Adjusting the weighting matrices  $Q$  and  $R$  allows us to balance control effort and system performance. Higher  $R$  values result in smoother control actions but potentially slower convergence, while higher  $Q$  values prioritize reducing state error and may lead to more aggressive control actions. Choosing the appropriate LQR parameters depends on system requirements, constraints, and desired trade-offs between performance and control effort. Tuning these parameters enables us to achieve the desired control behavior while optimizing system performance and considering limitations.

**A. LQR Controller with Varying Cost Matrices.** In this experiment, we implement a Linear Quadratic Regulator (LQR) controller with different weightings for the state and control cost matrices,  $Q$  and  $R$ , respectively. Three scenarios are considered: normal weightings (identity matrices for  $Q$  and  $R$ ), increased state cost (ten

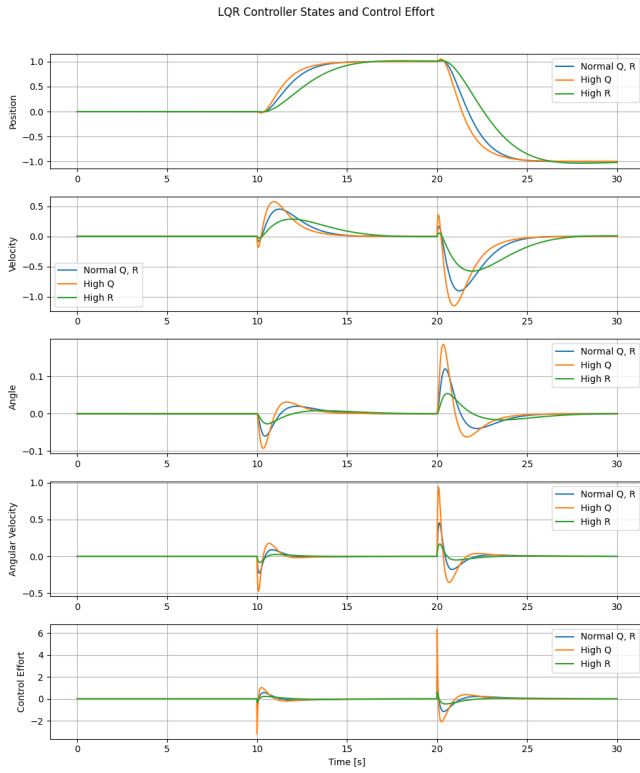


**Fig. 6.** Control effort for different LQR parameters. The blue line represents the control effort with normal  $Q$  and  $R$  matrices. The red line corresponds to a high  $Q$  value, and the green line corresponds to a high  $R$  value.

times the identity for  $Q$  and identity for  $R$ ), and increased control cost (identity for  $Q$  and ten times the identity for  $R$ ).

The state and control effort of the system are recorded and plotted over a duration of 30 seconds, with setpoints changing at 0s, 10s, and 20s.

Increasing the weighting on the state cost matrix,  $Q$ , leads to a more aggressive response with higher control effort. However, it may not significantly improve the system's convergence to the setpoint. Conversely, increasing the weighting on the control cost matrix,  $R$ , results in a more conservative controller that minimizes control effort at the potential expense of larger devia-



**Fig. 7.** State and control effort of the LQR controller for normal, high  $Q$ , and high  $R$  cases.

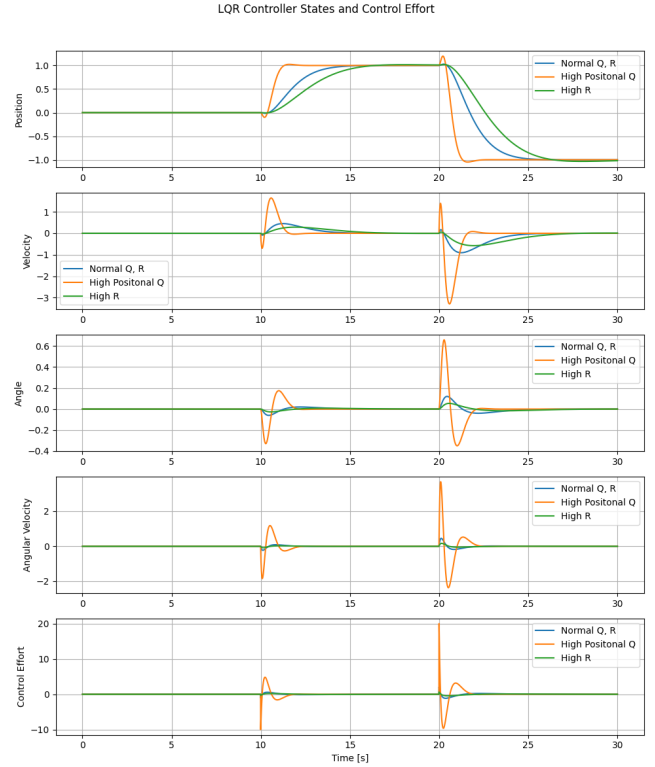
tions from the setpoint. This can be advantageous when considering actuator limitations or energy consumption.

Further exploration could involve adjusting the weightings in the  $Q$  matrix for specific state variables. For example, heavily weighting position in the  $Q$  matrix while setting other weights to zero would reveal the effect of prioritizing position control over other variables.

**B. LQR Controller with Varying Cost Matrices and Emphasis on Position.** In the previous experiment, we implemented a Linear Quadratic Regulator (LQR) controller with different weightings for the state and control cost matrices,  $Q$  and  $R$ , respectively. Extending this investigation, we now examine a scenario where the  $Q$  matrix is manipulated to assign a significantly higher weight to the positional state, while the weights for other states (velocity, angle, angular velocity) are set to zero.

This adjustment causes the controller to focus primarily on minimizing positional error, often at the expense of other states. The outcome of

this experiment, in terms of state and control effort, is illustrated below.



**Fig. 8.** State and control effort of the LQR controller with high positional weight in  $Q$ .

By assigning a higher weight to the positional state in the  $Q$  matrix while setting other state weights to zero, the LQR controller achieves faster convergence to the positional setpoint. However, this improvement comes at the expense of increased deviations in other state variables. These findings highlight the trade-off involved in tuning the LQR controller and showcase its flexibility in adapting to specific requirements. By adjusting the weights in the  $Q$  and  $R$  matrices, the controller's behavior can be tailored to prioritize positional accuracy or other state variables, as needed.

## 9. Conclusion

This paper provides an in-depth analysis and investigation of the implementation of Linear Quadratic Regulator (LQR) control for a cart-pole system. Various scenarios were examined to better understand the implications of the cost function matrices  $Q$  and  $R$  on system performance. Our results demonstrated the significant impact

662 of these matrices on both settling time and con-  
663 trol effort.

664 A crucial observation from our experiments  
665 was the delicate trade-off between system perfor-  
666 mance and control effort. It was seen that simply  
667 increasing the weights in the  $Q$  matrix doesn't  
668 necessarily lead to better system response, as it  
669 may result in excessive control effort. Therefore,  
670 the selection of these weights is an important as-  
671 pect of designing an LQR controller, and it may  
672 require a certain degree of tuning and system  
673 knowledge.

674 Detailed discussion, code, and animations for  
675 the results presented in this document [are avail-](#)  
676 [able in our GitHub repository](#).

## 677 10. Acknowledgments

678 The authors would like to express their gratitude  
679 to the [University of Michigan's Control Tutori-](#)  
680 [als for MATLAB and Simulink \(CTMS\)](#) website  
681 for the system model and equations used in this  
682 paper. The website served as an invaluable re-  
683 source, providing a detailed breakdown of the  
684 cart-pole system and contributing significantly  
685 to our understanding of the Linear Quadratic  
686 Regulator control technique implemented in this  
687 research.

688 Special thanks to the contributors of the CTMS  
689 website for making the information publicly ac-  
690 cessible and promoting a culture of shared learn-  
691 ing within the field of control systems.