Degree project

Haptic Servo System

Author: Mohamad Khashab
       Mohammad Firas Moulki
Supervisor: Matz Lenells
Examiner: Pieternelia Cijvat
Date: 2015-06-14
Course code: 2ED14E, 15 hp
Topic: Electrical Engineering
Level: Bachelor of Science

Department of Physics and Electrical Engineering
Faculty of Technology
Summary

A “Haptic servo system” is here understood as a servo system where forces from a controlled system are fed back to an operator. This thesis work is a design work where the work among other things comprises the choice of suitable motors, one for operating the beam and another one for operating the steering wheel. Data for the beam and ball are assumed to be known. Data for the feed back torque to the steering wheel is assumed to be specified in advance.

The ball and beam system is modeled into state space equations using both Newtonian mechanics and Euler-Lagrange equations. Two models to represent the human response are suggested: Linear Quadratic Regulator and an ad-hoc method based on the operator’s visual response. One simulation study is done to test the linear controller. Another is carried out to show that the system works according to some specification. The ball and beam process is simulated with hardware in the loop. The hardware in the loop is a Maxon motor. The motor is used as the steering wheel and the motor will also propagate the torque feedback to the operator.

The task of the thesis work could then be formulated as: Can a human, with torque feedback, manually control the ball on the beam without looking at the ball and the beam?
Abstract

A "Haptic servo system" is here understood as a servo system where forces from a controlled system are fed back to an operator. This thesis work is a design work where the work among other things comprises the choice of suitable motors, one for operating the beam and another one for operating the steering wheel. Data for the beam and ball are assumed to be known. Data for the feedback torque to the steering wheel is assumed to be specified in advance. Two models to represent the human response are suggested. A simulation study is carried out to show that the system works according to some specification. The ball and beam process is simulated with hardware in the loop. The hardware in the loop is a Maxon motor. The motor is used as the steering wheel and the motor will also propagate the torque feedback to the operator.

The task of the thesis work could then be formulated as: Can a human, with torque feedback, manually control the ball on the beam without looking at the ball and the beam?

Keywords: ball and beam system, haptic, nonlinear control, stability, Euler-Lagrange equations, human tactile delay
Acknowledgment

First of all we would like to express our sincere gratitude to our supervisor, senior lecturer Matz Lenells, for giving us the opportunity and undertake our Bachelor project, for his much valued comments, guidance, patience and encouragement throughout our studies. He has always encouraged and motivated us to understand the concepts that are needed to complete this project. His guidance helped us in all the time during the thesis work. We would also like to extend our deepest gratitude to our program coordinator Dr. Pierronella Cijvat for her help through our studies. We greatly appreciate our teachers’ effort and their time to help us throughout the whole year. Their knowledge and expertise have truly helped us complete our studies successfully, and the time we spent here at the Department of Physics and Electrical Engineering gave us new insights and ideas.
Mohammad Firas Moulki

I would like to express my deepest gratitude to my family for their love and support throughout the years. To my father, who protected and provided. To my mother, who raised and supported. To my three angelic siblings. Without you I wouldn’t have made it this far.

To Mr. Ayman Asfari, I send my thanks and appreciation. Your generosity allowed me to receive such an education.

To all people who love me no matter how little, I love you more.

Mohamad Khashab

I would like to thank my family members. I thank my parents for their continuous support and encouragement. Special thanks to my father, you have always been my hero and I hope I made you proud even if you expected more.

To my mother, thanks for your unconditional love and prayers. I truly love you. I thank my sisters and brother for taking care of my parents while I am away. My thankful feeling needs more than thousands words to be expressed but thank you very much and without all your help I would not reach to this point. To Vicky, I would like to thank you for all your supports throughout my study life. My specially warm and lovely thanks will always be for you.
# Contents

1 Introduction  

2 Project Description  

3 Background  
   3.1 Newtonian Mechanics  
      3.1.1 Kinematic  
      3.1.2 Dynamic  
      3.1.3 Torque  
   3.2 Euler Lagrange  
   3.3 Robot Manipulator  
   3.4 Control Theory  

4 System Modeling  
   4.1 System Description  
   4.2 Newtonian Mechanics  
   4.3 Euler-Lagrange Equations  
   4.4 A Comparison of the Two Methods  

5 Linearizing The System  

6 Modeling The human response  
   6.1 Linear Quadratic Controller  
   6.2 Ad-hoc Method  

7 Simulations on The Linear System
1 Introduction

Haptic comes from a Greek language “haptesthai” which means touch [1]. Haptic technology is commonly referred to as the ability to interact with the virtual environment through physical contact such as receiving sensation associated with what is going on in the virtual world through touch [2]. This type of technology is increasing the human interaction by improving the relation between the human and their physical environment. Haptic technology is also known as kinesthetic communication and it is consider to be a tactile feedback system [3]. This is because of the recreation of the sense and feeling which has been caused by an applied forces, vibrations or motions to the users. As a result, the general method and operation of these technologies will be based on the involvement of sensation of an abject.

This technology has been found some ages ago when Thomas D. Shannon did invent the tactile telephone in 1973 and that was the first official signed invention [4]. Since then, this type of science has been developed by many scientists and organizations through the years. Nowadays, this technology is involved in everyday life and in almost every type of electrical equipment. Haptic devices are spread into many different fields such as entertainments, health and educations. For examples, smart phones, games remote controllers and some medical equipments such as robotic surgery tools.

This technology has some advantages and disadvantages as well. Haptic equipments provide immediate response, errors free, save time and also minimize the number of workers in the business. On the other hand, some of these devices can get very complicated to deal with especially for old generations and some small institutions or companies find it expensive for buying
such equipment.

2 Project Description

The purpose behind this project is designing a Haptic device that can control a ball on the beam by an external force. This beam ball system will be attached with a servo motor in the middle of the rod. The system is divided into two stages: the first stage is a rod and ball being control by the attached motor. The second stage is the user who controls a secondary motor trying to keep the position of the ball stable. This is explained in figure 1.

![Project's main blocks diagram]

Figure 1: The project’s main blocks
The goal of this paper is investigating whether the user can balance the ball on the rod using only a tactile feedback from the rod and ball system. This will be done through few steps. First, the rod and ball system should be modelled. This modeling part will be done using both Newtonian Mechanics and Euler-Lagrange Equations. Second, the system will be linearized to simplify the study. Third, suggesting a reasonable feedback signals for the user to feel. Fourth, testing if the user will be able to control the system according to the suggested feedback signals. Finally, checking whether or not the linear controller works on the nonlinear system.

The theories that are going be discussed in this paper will be helpful to understand how controlling methods are applied to such a mechanical system. Moreover, this might give a good knowledge on how these concepts and principles are applied to some real life projects such as horizontally stabilizing an airplane during landing and in turbulent airflow. Thus, this study can enable us to analyze broader prospects in the control field.
3 Background

This chapter explains the concepts that are needed in term of understanding the designed Haptic device. These concepts will be introduced generally for providing a simple guide for modeling the system first. Two principles of modeling technique will be explained which will be helpful by applying them on the given system. Finally, some control theories will be discussed.

3.1 Newtonian Mechanics

Newtonian mechanic is the study of object’s motion which can predict the future given the present. Any motion particle can be analytically studied by knowing its initial status such as its location and initial time [5, p63-78]. This analysis will lead us to know how this object might be changed in the future based on the surrounded effects such as forces and mass. Furthermore, Newtonian Mechanic is divided into two parts which are Kinematic and Dynamic [5, p3]. However, these two sub visions of Newtonian Mechanics are used by Newton’s laws of motion to analyze any movements.

3.1.1 Kinematic

Kinematic is the part of the Newtonian mechanics that describes the present of any picked object. It shows the information that is needed in term of predicting the future of the system that is under study [5, p7-18]. So, Kinematic is consider to be the stage of verification of prediction at the initial time and it does not have to deal with what are the reasons that cause the object motion or change in displacement. Kinematic explains the condition
of a system at a starting time for example, location and initial velocity where they are considered to be vector quantities.

3.1.2 Dynamic

Dynamic is the other branch of the Newtonian Mechanics which describes the reasons and principles that cause the movement of an object or system. These descriptions can be forces or torques. The basic of Dynamic analysis is the Newton’s laws of motion which govern the interaction between velocity, positions and time. The three basic rules are:

- First law: an object will either remain at rest or in motion at a constant speed unless an external force is applied to change its status[5, p107].

- Second law: an object experiencing a force will start to be in motion with acceleration depending on the amount of this applied force [5, p107]. This proportional relation is showing by the following formulas:

\[
\begin{align*}
\vec{F} &= ma \\
\vec{a} &= \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2}
\end{align*}
\]

- Third law: for every action, there is an equal and opposite reaction [5, p107]. That means if an object A is exerting a force on object B, then object B will exert a reaction force on object A. This reacted force will
be equal in magnitude but in an opposite direction of the force that has been exerted by A.

\[ F_A = -F_B \]

### 3.1.3 Torque

Torque is a measurement of how much force is acting on an object and causes a rotational movement [6]. The point that the object is rotating around is called a pivot point where it is consider being the center of this circular path. Moreover, Torque is a vector which has a magnitude as well as direction, depending on how the viewer is analyzing the coordinate’s configuration [6]. Next figure shows a general overview of a force causing a rotational movement, followed by the torque equation:

\[ \tau = r \times F \]

\[ \tau = |r||F| \sin \theta \]

Where \( \theta \) is the angle between the two vectors \( r, F \) and \( r \) is the distance between the pivot point and the applied force \( F \).
The direction of the torque vector is determined by using the right hand rule depending on the direction of rotation.

### 3.2 Euler Lagrange

Euler-Lagrange theory was developed in 1770s by two scientists; Leonhard Euler and Joseph-Louis Lagrange. Euler-Lagrange (EL) is considered to be another theory for modeling a mechanical system where set of equations can describe the status of a system. In addition, this method leads to an equivalent result as using Newtonian Mechanics analysis. EL technique is more distinguished than using Newton’s technique. This is because, EL can be applied to electrical systems as well as any systems of generalized coordinates and forces [7, p4-8]. Generalized coordinate in analyzing a mechanical system means the parameters that describe the configuration and arrangement of a specific system related to a reference configuration.

The Euler-Lagrange general equation for a generalized coordinate $q$ is given by the following expression [8]:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \sum_{j=1}^{N} (F_j | \frac{\partial r_j}{\partial q})$$

In the above mathematical expression, $L$ represents Lagrangian that can be computed using the kinetic ($T$) and potential ($V$) energy [8]. In general, energy means the ability to do some works and this can be in many different type of classification. However, the two types of energies that are engaged with Lagrangian are energy due position ($V$) and motion ($T$). These two components are in a relation with Lagrangian as follows [8]:

$$L = T - V$$
3.3 Robot Manipulator

A Robot manipulator is considered to be a mechanism device as well as a Haptic device that can manipulate materials without a direct contact [9, p1-4]. This technology is still a growing market and it is now being applied beyond conventional areas. This brings a lot of benefits to many different fields such as industrial manufacturing, medical and surgery [9, p20-24]. In the industrial field a wheel loader is a good example of manipulator technology. Also, this technology can be found in almost every hospital operator rooms. A robot itself consists of several components that are integrated together to form the whole set. These components are manipulator, Actuators, sensors, controller, processor and software [10, p1-18]. However, this section will focus on the manipulator component as it is the most related concepts to the main topic of the paper.

The Manipulator

Manipulator is considered to be an arm-like mechanism that consists of series of segments usually connected together which can grasp and move objects [9, p6]. Some of the manipulators are also considered to be tactile feedback devices [11]. However, figure 2 shows a general design of a simple manipulator. The study of manipulators involves dealing with positions and orientations of manipulator parts. Also, it consists of practical theories for kinematical and dynamical modeling and computations. In addition, kinematic model represents the motion of the manipulator without focusing on the causes. On the other hand, dynamic modeling describes the relationship between the motion and forces that were involved as well as all the masses.
The manipulator parts are composed of an assembly of links and joints where links are the rigid section that makeup the mechanism. These solid segments are connected together by joints where most joints connect two parts with each other. Furthermore, the part that is connected to the last joint is called End-Effectors where it is directly interacting with the environment to perform a certain task [9, p6-7]. The joint will cause a specific motion for the segment that is connected to it but these movements are depending on the type of the joint. As a result, there are five different types of joint and each one of them performs uniquely. These five type of joints as the following [9, p11-19]:

- Revolute: causing a rotational movement around a fixed point, providing a one degree of freedom.
- Cylindrical: allows a rotational movement around one axis and a single
axis sliding (shifting position).

- **Prismatic**: providing a linear sliding between two segments that are connected to this joint which is also known as slider joint.

- **Spherical**: allows three degree of rotational freedom around the centre of the joint, this provides a rotational movement with an angle.

- **Planar**: allows translation on a plane and rotation about an axis that is perpendicular to the plane.

However, these joints will not be active without being attached to actuators which are controlled by the controller and these actuators are often to be servo motors [9, p7].

These devices can influence the market since there is a strong demand and this technology has many disadvantages as well as advantages. Robotic technology can increase the productivity, safety, efficiency, quality and perform multi-tasks [9, p5]. On the other hand, robot manipulators sometimes can replace human which cause an economics problem as lost salaries; Also some manipulators can be dangerous which might cause human injuries [9, p6].
3.4 Control Theory

System means a set of elements interacting among themselves and with the environment which can be controlled. Usually, a system can be expressed using mathematical models. These models are used to predict its future behaviour under certain conditions. The models can have many forms such as a transfer function or state space equations.

Dynamic systems can be controlled by different controllers and one type of controller is called a manual control where system involves human controlling a machine [12, p21]. While machines that do not require a human interaction to do the controlling are called automatic control systems [12, p21]. There are several types of controllers and these are applied differently depending on the type of the system that needs to be control for example open/closed loop control system and feed-forward control [12, p21]. “If the controller does not use a measure of the system output being controlled in computing the control action to take, the system is called open-loop control” [12, p21]. This shows that the output of the open-loop system is being independent and has no effects due to the input signal. Close-loop control is done through a feedback of the output signal after it is measured [12, p21].

Systems are usually categorized into linear and nonlinear systems, each category having its own theory. Some of the most popular linear controllers are PID (Proportional, Integral, Derivative) controllers, Linear Quadratic Regulators (LQR) and Fuzzy Logic controllers. LQR is used in section six and some basic concepts are introduced there. Linear systems are studied because of their simplicity and because they are good approximations [13, p27].
Moreover, linear controllers can work on the equivalent nonlinear systems if the system states are kept close to the equilibrium where the linearization was made [13, p371-372]. This approach is usually used by engineers as a first step in constructing a controller, and this is the approach used in this report. If it fails, one has to revert to the more complicated nonlinear control theory.
4 System Modeling

4.1 System Description

The system under study consists of a solid iron ball, a beam, and a current controlled motor. The naming of the parameters and vectors associated with each object is explained below. Some parameters are shown in figure 3.

Note that polar coordinates where will be used:

\[ \hat{r} = \hat{x}(\cos\theta) + \hat{y}(\sin\theta) \]
\[ \hat{\theta} = \hat{x}(-\dot{\theta}\sin\theta) + \hat{y}(-\dot{\theta}\cos\theta) \]

The Ball

- \( R_B \): the radius of the ball
- \( \rho_B \): the density of the ball’s material
- \( m_B \): the mass of the ball
  
  \[ m_B = \frac{4}{3} \pi R_B^3 \rho_B \]
- \( J_B \): the ball’s moment of inertia around an axis passing through the center parallel to the beam plain
  
  \[ J_B = \frac{2}{5} m_B R_B^2 \]
- \( R_B = R_B \hat{\theta} \)
- \( r_B \): the vector from the origin to the ball’s center of mass
• $\mathbf{r}_c = r_c \hat{r}$: the vector from the origin and the point where the ball touches the beam

The Beam
The beam has a mass $m$. The dimensions of the beam are $l$ (length), $w$ (width), and $t$ (thickness).

• $J_M$: the beam’s moment of inertia around an axis passing through it’s center perpendicular to the LT plain

$$J_M = \frac{1}{12} m(l^2 + t^2);$$

The Motor
The torque of the motor is directly proportional to the current:

$$\tau_M = K_I I$$

General variables
The system will be modeled using two general variables:

• $\theta$: the angle the beam makes with the $x$ axis

• $\psi$: the angle the ball rolls starting from an initial state

External Forces
In this analysis, all kinds of friction forces will be neglected for simplicity, except for the surface tension $F_t$ which is required for the ball to roll. The forces are shown in figure 3.

• $m_B g \hat{y}$: the weight of the ball
Figure 3: The beam and ball system with External Forces

- \( F_N = F_N \hat{\theta} \) and its reaction: the normal force at the point the ball touches the beam.

- \( F_t = F_t \hat{r} \) and its reaction: the surface tension at the point the ball touches the beam.

- The forces causing the torque \( \tau_M \)

4.2 Newtonian Mechanics

The rotational form of Newton’s second law is used on the ball and the beam.

\[
\sum \tau_{\text{external}} = J \ddot{\theta}
\]
The following equations are obtained:

\[ J_B \frac{d^2(\psi + \theta)}{dt^2} \dot{z} = F_t \times R_B \]  
(4.1)

\[ J_M \frac{d^2\theta}{dt^2} \dot{z} = \tau_M + r_c \times (-F_N) \]  
(4.2)

Then, Newton’s second law (normal from) is applied on the ball alone:

\[ m_B \frac{d^2r_B}{dt^2} = F_t + F_N + m_B g(\hat{y}) \]  
(4.3)

Note that:

\[ \frac{d^2r_B}{dt^2} = \frac{d^2r_c}{dt^2} + \frac{d^2R_B}{dt^2} \]  
(4.4)

Where \( r_c = r_c \cdot \hat{r} \) and \( R_B = R_B \cdot \hat{\theta} \)

It can also be shown that:

\[ \frac{d\hat{r}}{dt} = \hat{\theta} \quad \frac{d\hat{\theta}}{dt} = -\dot{\theta} \hat{r} \quad \frac{d^2\hat{r}}{dt^2} = \ddot{\theta} - \dot{\theta}^2 \hat{r} \]

Thus,

\[ \frac{d\dot{r}_c}{dt} = \dot{r}_c \cdot \hat{r} + r_c \cdot \dot{\hat{\theta}} \]

\[ \frac{d^2\dot{r}_c}{dt^2} = \dot{r}_c \cdot \hat{r} + r_c \cdot \dot{\hat{\theta}} + (\dot{r}_c \cdot \dot{\hat{\theta}} + \ddot{r}_c) \cdot \hat{\theta} - r_c \cdot \ddot{\theta} \]

\[ \frac{d^2r_c}{dt^2} = (\dot{r}_c - r_c \ddot{\theta}^2) \cdot \hat{r} + (2\dot{r}_c \cdot \dot{\hat{\theta}} + \ddot{r}_c) \cdot \hat{\theta} \]

\[ \frac{d^2R_B}{dt^2} = \frac{d}{dt}(-R_B \dot{\theta} \cdot \hat{r}) = (-R_B \ddot{\theta} \cdot \hat{r}) + (-R_B \ddot{\theta}^2) \cdot \hat{\theta} \]

Applying this to (4.4):

\[ \frac{d^2r_B}{dt^2} = (\dot{r}_c - r_c \ddot{\theta}^2 - R_B \ddot{\theta}^2) \cdot \hat{r} + (2\dot{r}_c \cdot \dot{\hat{\theta}} + \ddot{r}_c - R_B \ddot{\theta} \cdot \hat{\theta}) \]

Substituting in (4.3):

\[ m_B[(\dot{r}_c - r_c \ddot{\theta}^2 - R_B \ddot{\theta} \cdot \hat{r}) + (2\dot{r}_c \cdot \dot{\hat{\theta}} + \ddot{r}_c - R_B \ddot{\theta} \cdot \hat{\theta}) \cdot \hat{\theta}] = F_t + F_N + m_B g(-\hat{y}) \]
The last equation can be simplified into two equations by projecting it on the \( \hat{r}, \hat{\theta} \) axis respectively,

\[
m_B(\ddot{r}_c - r_c \dot{\theta}^2 - R_B \ddot{\theta}) = F_t - m_B g \sin \theta \quad (4.5)
\]

\[
m_B(2 \ddot{r}_c \dot{\theta} + \ddot{\theta} r_c - R_B \dot{\theta}^2) = F_N - m_B g \cos \theta \quad (4.6)
\]

One more equation is needed in order to solve for \( \psi \) and \( \theta \). The equation comes from the rolling ball:

\[
r_c = -R_B \psi + r_\alpha \quad (4.7)
\]

The process of solving the system of equation is by eliminating of the contact forces \( F_t, F_N \). From (4.6),

\[
F_N = m_B(2 \ddot{r}_c \dot{\theta} + \ddot{\theta} r_c - R_B \dot{\theta}^2 + g \cos \theta)
\]

Substituting in (4.2):

\[
J_M \ddot{\theta} = \tau_M - r_c m_B(2 \ddot{r}_c \dot{\theta} + \ddot{\theta} r_c - R_B \dot{\theta}^2 + g \cos \theta)
\]

Using (4.7):

\[
J_M \ddot{\theta} = \tau_M + m_B(R_B \psi - r_\alpha)(g \cos \theta - 2 R_B \dot{\psi} \dot{\theta} + \ddot{\theta}(-R_B \psi + r_\alpha) - R_B \dot{\theta}^2) \quad (4.8)
\]

From (4.5),

\[
F_t = m_B(\ddot{r}_c - r_c \dot{\theta}^2 - R_B \ddot{\theta} + g \sin \theta)
\]

Substituting in (4.1):

\[
J_B(\ddot{\psi} + \ddot{\theta}) = R_B m_B(\ddot{r}_c - r_c \dot{\theta}^2 - R_B \ddot{\theta} + g \sin \theta)
\]

Using (4.7):

\[
J_B(\ddot{\psi} + \ddot{\theta}) = R_B m_B(-R_B \ddot{\psi} + (R_B \psi - r_\alpha) \dot{\theta}^2 - R_B \ddot{\theta} + g \sin \theta)
\]
Rearranging,

\[ J_B(\dot{\psi} + \dot{\theta}) = R_B^2 m_B(\frac{g}{R_B} \sin\theta - \dot{\psi} + \psi \dot{\theta}^2 - \dot{\theta}) - R_B m_B r_c \dot{\theta}^2 \] (4.9)

state space equations are going to be used in order to find a suitable control for this system. The state choice is going to be as follows:

\[ x_1 = \theta \quad x_2 = \dot{\theta} \]
\[ x_3 = \psi \quad x_4 = x_4 \]

Substituting in (4.8) and (4.9) and Rearranging:

\[ (J_M + m_B(-R_B x_3 + r_c)^2) \dot{x}_2 = \tau_M + m_B(R_B x_3 - r_c)(g \cos x_1 - 2 R_B x_4 x_2 - R_B x_2^2) \]
\[ \frac{J_B}{R_B^2 m_B}(\dot{x}_4 + \dot{x}_2) = (\frac{g}{R_B} \sin x_1 - \dot{x}_4 + x_3 x_2^2 - \dot{x}_2) - \frac{r_c}{R_B} x_2^2 \]

Finally,

\[ \dot{x}_2 = \frac{\tau_M + R_B m_B(R_B x_3 - r_c)(\frac{g}{R_B} \cos x_1 - 2 x_4 x_2 - x_2^2)}{J_M + m_B(-R_B x_3 + r_c)^2} \] (4.10)
\[ \dot{x}_4 = \frac{\frac{g}{R_B} \sin x_1 + x_3 x_2^2 - \frac{r_c}{R_B} x_2^2}{J_B \frac{R_B}{R_B^2 m_B} + 1} - \dot{x}_2 \] (4.11)

### 4.3 Euler-Lagrange Equations

The Euler-Lagrange equations for the generalized coordinates \( q \) are given by:

\[ \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \sum_{j=1}^{N} (F_j | \frac{\partial r_j}{\partial q}) \] (4.12)

Where \( L = T - V \) is the Lagrangian. \( T \) is the kinetic energy and \( V \) is the potential energy of the system.

First, \( T \) and \( V \) for the system will be calculated:

\[ T = 0.5[J_M \dot{\theta}^2 + J_B(\dot{\psi} + \dot{\theta})^2 + m_B(\frac{d r_B}{dt} | \frac{d r_B}{dt})] \]
\[ \frac{d \mathbf{r}_B}{dt} \text{ was calculated in the previous section:} \]

\[
\begin{align*}
\frac{d \mathbf{r}_B}{dt} &= \frac{d \mathbf{r}_c}{dt} + \frac{d \mathbf{R}_B}{dt} = \dot{r}_c \hat{\mathbf{r}} + r_c \dot{\theta} \hat{\mathbf{\theta}} - R_B \dot{\mathbf{\theta}} \\
\left( \frac{d \mathbf{r}_B}{dt} \right) \left( \frac{d \mathbf{r}_B}{dt} \right)^T &= (\dot{r}_c - R_B \dot{\theta})^2 \hat{\mathbf{r}} + (r_c \dot{\theta})^2 \hat{\mathbf{\theta}} \\
&= \dot{r}_c^2 + (R_B^2 + r_c^2) \dot{\theta}^2 - 2 R_B \dot{r}_c \dot{\theta} \\
&= R_B^2 \dot{\psi}^2 + (R_B^2 + (R_B \psi - r_c))^2 \dot{\theta}^2 + 2 R_B^2 \dot{\psi} \dot{\theta}
\end{align*}
\]

Thus, the total kinetic energy of the system is:

\[ T = 0.5 [J_M \dot{\theta}^2 + J_B (\dot{\psi} + \dot{\theta})^2 + m_B (R_B^2 \dot{\psi}^2 + (R_B^2 + (R_B \psi - r_c))^2 \dot{\theta}^2 + 2 R_B^2 \dot{\psi} \dot{\theta})] \]  

\[ (4.13) \]

\[ y = 0 \text{ is taken as the reference for altitude: } V(h = 0) = 0. \text{ Notice also that } V_{\text{beam}} = 0 \text{ since the beam is symmetric and is centered at the origin.} \]

Thus, the potential of the system is only that of the ball:

\[ V = m_B g (r_c \sin \theta + R_B \cos \theta) \]

\[ V = m_B g R_B (-\psi \sin \theta + \cos \theta) + mgr_c \sin \theta \]  

\[ (4.14) \]

And the Lagrangean L is:

\[ L = 0.5 [J_M \dot{\theta}^2 + J_B (\dot{\psi} + \dot{\theta})^2 + m_B (R_B^2 \dot{\psi}^2 + (R_B^2 + (R_B \psi - r_c))^2 \dot{\theta}^2 \dot{\theta}^2 + 2 R_B^2 \dot{\psi} \dot{\theta}) + 2 R_B^2 \dot{\psi} \dot{\theta})] + m_B g R_B (\psi \sin \theta - \cos \theta) - mgr_c \sin \theta \]  

\[ (4.15) \]
To start off, the left hand side of (4.12) will be calculated:

\[
\frac{\partial L}{\partial \dot{\theta}} = J_M \dot{\theta} + J_B (\dot{\theta} + \dot{\psi}) + 0.5m_B (2(R_B^2 + (R_B \psi - r_{c0})^2) \dot{\theta}) + 2R_B^2 \dot{\psi}
\]

\[
= \dot{\theta} [J_M + J_B + m_B(R_B^2 + (R_B \psi - r_{c0})^2)] + \dot{\psi} [J_B + m_B R_B^2]
\]

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} = \ddot{\theta} [J_M + J_B + m_B(R_B^2 + (R_B \psi - r_{c0})^2)] + \ddot{\psi} [J_B + m_B R_B^2]
\]

\[
= \ddot{\theta} [2m_B R_B \dot{\psi}(R_B \psi - r_{c0})]
\]

\[
\frac{\partial L}{\partial \theta} = m_B g R_B (\psi \cos \theta + \sin \theta) - m r_{c0} \cos \theta
\]

\[
\frac{\partial L}{\partial \psi} = J_B (\dot{\theta} + \dot{\psi}) + 0.5m_B (2R_B^2 \dot{\psi} + 2R_B^2 \dot{\theta})
\]

\[
= (\dot{\theta} + \dot{\psi}) (J_B + m_B R_B^2)
\]

\[
\frac{d}{dt} \frac{\partial L}{\partial \psi} = (\ddot{\theta} + \ddot{\psi}) (J_B + m_B R_B^2)
\]

\[
\frac{\partial L}{\partial \dot{\psi}} = m_B g R_B \sin \theta + m_B R_B (R_B \psi - r_{c0}) \dot{\theta}^2
\]
Regarding the right hand side of (4.12), from figure 3, the following forces are obtained:

$F_t$ and $F_N$ are contact forces. $m \cdot g$ is a conservative force. Thus,

$$F_t \left| \frac{\partial r_c}{\partial q} \right| = 0 \quad F_N \left| \frac{\partial r_c}{\partial q} \right| = 0 \quad m_B g \left| \frac{\partial r}{\partial q} \right| = 0$$

What is left is the forces producing the torque $\tau_M$. As shown in figure 5, these forces are going to be modeled as two opposing forces: $F_1 = b \dot{\theta}$, $F_2 = -b \dot{\theta}$. The forces are acting on the two ends of the beam: $r_1 = a \hat{r}$ and $r_2 = -a \hat{r}$ [8].

$$\sum_{j=1}^{2} (F_j) \left| \frac{\partial r_j}{\partial \theta} \right| = F_1 \left| \frac{\partial r_1}{\partial \theta} \right| + F_2 \left| \frac{\partial r_2}{\partial \theta} \right| = b \dot{\theta} \left| \frac{\partial (a \hat{r})}{\partial \theta} \right| - b \dot{\theta} \left| \frac{\partial (-a \hat{r})}{\partial \theta} \right|$$

$$= 2ab \dot{\theta} \left| \frac{\partial (\hat{r})}{\partial \theta} \right| = 2ab (\dot{\theta} | \dot{\theta}) = 2ab = \tau_M$$

(4.20)

On the other hand, $\frac{\partial r_j}{\partial \psi} = 0$ leads to:

$$\sum_{j=1}^{N} (F_j) \left| \frac{\partial r_j}{\partial \psi} \right| = 0$$

(4.21)

Substituting (4.16), (4.17), and (4.20) in (4.12):

$$\ddot{\theta} [J_M + J_B + m_B (R_B^2 + (R_B \psi - r_c))^2] + \ddot{\psi} [J_B + m_B R_B^2]$$

$$+ \dot{\theta} [2m_B R_B \dot{\psi} (R_B \psi - r_c)] - m_B g R_B (\psi \cos \theta + \sin \theta) + m g r_c \cos \theta = \tau_M$$

(4.22)

Substituting (4.18), (4.19), and (4.21) in (4.12):

$$(\ddot{\theta} + \ddot{\psi}) (J_B + m_B R_B^2) - m_B R_B (R_B \psi - r_c) \dot{\theta}^2 - m_B g R_B \sin \theta = 0$$

(4.23)
The same choice of states as in the previous section will be used:

\[
\begin{align*}
  x_1 &= \theta \\
  x_2 &= \dot{\theta} \\
  x_3 &= \psi \\
  x_4 &= \dot{\psi}
\end{align*}
\]

(4.22) and (4.23) become:

\[
\begin{align*}
  \dot{x}_2[J_M + J_B + m_B(R_B^2 + (R_Bx_3 - r_0)^2)] + \dot{x}_4[J_B + m_B R_B^2] \\
  + x_2[2m_B R_B x_4 (R_B x_3 - r_0)] - m_B g R_B (x_3 \cos x_1 + \sin x_1) + m g r_0 \cos x_1 &= \tau_M \\
  (\dot{x}_2 + \dot{x}_4)(J_B + m_B R_B^2) - m_B R_B (R_B x_3 - r_0) x_2^2 - m_B g R_B \sin x_1 &= 0
\end{align*}
\]
Or in Matrix representation,

\[
\begin{pmatrix}
J_M + J_B + m_B (R_B^2 + (R_B x_3 - r_o)^2) & J_B + m_B R_B^2 \\
J_B + m_B R_B^2 & J_B + m_B R_B^2
\end{pmatrix}
\begin{pmatrix}
\dot{x}_2 \\
\dot{x}_4
\end{pmatrix}
= \begin{pmatrix}
\tau_M - 2m_B R_B (R_B x_3 - r_o) x_2 x_4 + m_B g R_B (x_3 \cos x_1 + \sin x_1) - m g r_o \cos x_1 \\
m_B R_B (R_B x_3 - r_o) x_2^2 + m_B g R_B \sin x_1
\end{pmatrix}
\] (4.24)

Equation (4.24) is of the form \( A \mathbf{x} = B \). To solve the equation for \( \mathbf{x} \), \( A^{-1} \) needs to be calculated.

\[
det(A) = (J_M + J_B + m_B (R_B^2 + (R_B x_3 - r_o)^2))(J_B + m_B R_B^2) - (J_B + m_B R_B^2)^2
\]
\[
= (J_M + m_B (R_B x_3 - r_o)^2) + J_B + m_B R_B^2 (J_B + m_B R_B^2) - (J_B + m_B R_B^2)^2
\]
\[
= (J_B + m_B R_B^2) (J_M + m_B (R_B x_3 - r_o)^2)
\]

\[
A^{-1} = \frac{1}{detA} \begin{pmatrix}
J_B + m_B R_B^2 & -J_B + m_B R_B^2 \\
-J_B + m_B R_B^2 & J_M + J_B + m_B (R_B^2 + (R_B x_3 - r_o)^2)
\end{pmatrix}
\]
\[
= \frac{J_B + m_B R_B^2}{detA} \begin{pmatrix}
1 & -1 \\
-1 & 1 + \frac{J_M + (R_B x_3 - r_o)^2}{J_B + m_B R_B^2}
\end{pmatrix}
\]

Finally, \( \mathbf{x} = A^{-1} B \):

\[
\begin{pmatrix}
\dot{x}_2 \\
\dot{x}_4
\end{pmatrix} = \begin{pmatrix}
\frac{\tau_M + m_B R_B (R_B x_3 - r_o) (\frac{g}{R_B^2} \cos x_1 - 2x_4 x_2 - x_2^2)}{J_M + m_B (-R_B x_3 + r_o)^2} \\
\frac{g}{R_B \sin x_1 + x_3 x_2^2 - r_o x_2^2} - \frac{\dot{x}_2}{J_B + m_B R_B^2}
\end{pmatrix}
\] (4.25)
4.4 A Comparison of the Two Methods

The problem of using only one method to model a system is that it’s hard to see whether or not there are errors. That’s the main reason for using two methods of modeling. Note that the final results of the previous sections agreed perfectly. Using two methods can also give the reader more insight on the nature of the problem.

One can see, however, that using Euler Lagrange method alone is easier for the following reasons:

- Contact forces (here $F_N$, $F_t$) are neglected in the analyses since
  \[ F_t \frac{\partial r_c}{\partial q} = 0 \quad F_N \frac{\partial r_c}{\partial q} = 0 \]

- The two final equations are formed from the beginning. No need to solve a system of six equations as done in the classical case.

- One only deals with scalar quantities, while in the classical method, one has to use vector calculus.

The only advantage found of using the classical (Newtonian) method is that it’s well known and theoretically simple to understand.
5  Linearizing The System

In the previous section, the following state space equations were found:

\[
\dot{x}_2 = \frac{\tau_M + R_B m_B (R_B x_3 - r_{c0}) (\frac{g}{R_B} \cos x_1 - 2x_4 x_2 - x_2^2)}{J_M + m_B (-R_B x_3 + r_{c0})^2} \tag{5.1}
\]

\[
\dot{x}_4 = \frac{g}{R_B} \sin x_1 + x_3 x_2^2 - \frac{r_{c0}}{R_B} x_2^2 \left( J_M + \frac{m_B}{R_B^2 m_B} \right) - \dot{x}_2 \tag{5.2}
\]

The system will be linearized around the origin (0,0,0,0). The following approximations are applied used on equations (5.1) and (5.2):

- \(2x_4 x_2 \approx 0\)
- \(x_2^2 \approx 0\)
- \(\cos x_1 \approx 1\)
- \(\sin x_1 \approx x_1\)

The following equations are obtained:

\[
\dot{x}_2 \approx \frac{1}{J_M + m_B (-R_B x_3 + r_{c0})^2} (\tau_M + m_B (R_B x_3 - r_{c0}) g) \tag{5.3}
\]

\[
\dot{x}_4 \approx \frac{\frac{g}{R_B} x_1 - \dot{x}_2}{\frac{J_B}{R_B^2 m_B} + 1} \tag{5.4}
\]

The remaining nonlinear term in (5.3) and (5.4) is only \(\dot{x}_2\). The numerator of \(\dot{x}_2\) is linear, so the denominator is left,

\[
\frac{1}{J_M + m_B (-R_B x_3 + r_{c0})^2} \approx \frac{1}{J_M + m_B r_{c0}^2} \left[ 1 - \frac{2m_B R_B r_{c0}}{J_M + m_B r_{c0}^2} x_3 \right] \frac{1}{J_M + m_B r_{c0}^2} \left( 1 + \frac{R_B r_{c0}}{(J_M + m_B r_{c0}^2)^2} x_3 \right) \tag{5.5}
\]
Here, the following approximation was used:

\[
\frac{1}{1 - x} = 1 + x + x^2 + \ldots \approx 1 + x
\]

Substitute (5.5) in (5.3),

\[
\dot{x}_2 = \frac{1}{J_M + m_B r_c^2} \left( 1 + \frac{2 m_B R_B r_c}{(J_M + m_B r_c^2)^2} x_3 \right) (\tau_M + m_B (R_B x_3 - r_c) g)
\]

Nonlinear terms such as \( \tau_M x_3 \) and \( x_3^2 \) are neglected,

\[
\dot{x}_2 \approx \frac{\tau_M}{J_M + m_B r_c^2} + \frac{m_B g R_B}{J_M + m_B r_c^2} x_3 - \frac{2 g m_B^2 R_B r_c^2}{(J_M + m_B r_c^2)^2} x_3 - \frac{m_B g r_c}{J_M + m_B r_c^2}
\]

Finally,

\[
\dot{x}_2 = \left( 1 - \frac{2 m_B r_c^2}{J_M + m_B r_c^2} \right) \frac{m_B g R_B}{J_M + m_B r_c^2} x_3 + \frac{\tau_M - m_B g r_c}{J_M + m_B r_c^2}
\]

\[
\dot{x}_4 = \frac{g}{R_B m_B} + 1 \left( \frac{2 m_B r_c^2}{J_M + m_B r_c^2} - 1 \right) \frac{m_B g R_B}{J_M + m_B r_c^2} x_3 - \frac{\tau_M - m_B g r_c}{J_M + m_B r_c^2}
\]

These equations show that for small deviation from the origin, \( \dot{x}_2 \) is linearly dependent on the position of the ball and the torque of the motor, which is reasonable.

On the other hand, the ball acceleration is a function of the beam angle and its acceleration.

To write the (5.6) and (5.7) in matrix form, the following equations are used:

- \( \Delta \tau_M = \tau_M - m_B g r_c \)
- \( \Delta \tau_M = K_I I \)
And the final result is:

\[
\dot{x} = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & (1 - \frac{2m_Br_{c0}^2}{J_M+m_Br_{c0}^2}) & \frac{m_BR_{gg}}{J_M+m_Br_{c0}^2} & 0 \\
0 & 0 & 0 & 1 \\
\left(\frac{g}{DR_{b}^{m_B}}\right)^2 & 0 & \left(\frac{2m_Br_{c0}^2}{J_M+m_Br_{c0}^2} - 1\right) & \frac{m_BR_{gg}}{J_M+m_Br_{c0}^2} & 0 \\
\end{pmatrix} x + K_t \begin{pmatrix}
0 \\
0 \\
\frac{1}{J_M+m_Br_{c0}^2} \\
0 \\
\frac{-1}{(J_M+m_Br_{c0}^2)} \\
\end{pmatrix} u
\]  

(5.8)
6 Modeling The human response

In the following sections, some suggestions are presented for the feedback a human should feel for him to be able to control the system. The main issue here is that humans have a tactile response delay of around 0.4 seconds [14]. However, the studied control theories assume that the response is instant, i.e. they are automatic controllers. This tactile response delay is going to decrease the efficiency of the controller and it might render it useless. Another issue is that the human will not reply to the tactile feedback with an equal force/torque. This is because the human cannot accurately measure and apply a specific amount of force. However, in this modeling, it is assumed that the human will give the required force/input precisely.

The first controller suggested is the optimal controller based on the Linear Quadratic theory. The second controllers is created by us trying to mimic what a human visual response might be.

6.1 Linear Quadratic Controller

In this section, it is assumed that the human can react to the tactile feedback as a Linear Quadratic Controller. The input to the ball-beam system is [13, p242],

\[ u = -Lx \]  \hspace{1cm} (6.1)

Where,

- \( x \) is the vector representing the states of the system in equation (5.8).
  This system is of the form \( \dot{x} = Ax + Bu \)
• L is given by [13, p242],

\[
L = Q_2^{-1}B^T S
\]

Where S is the solution to the matrix equation [13, p242]:

\[
A^T S + SA + Q_1 - SBQ_2^{-1}B^T S = 0
\]

L can be calculated in Matlab using the command:

\[
L = \text{lqr}(A, B, Q_1, Q_2)
\]

The matrices \(Q_2\) and \(Q_1\) represent the sensitivity of the controller to the control input signal (u) and the system states error (e) respectively, as can be seen in the following equation [13, p239],

\[
\min(||e||^2 Q_1 + ||u||^2 Q_2) = \min \int e^T(t)Q_1 e(t) + u^T(t)Q_2 u(t)dt
\]

To calculate the generated tactile feedback \(\tau_U\), the following is introduced:

\[
\tau_U = -\frac{\tau_M}{\tau_U}
\]

(6.2)

It’s known that,

\[
\tau_M = K_t u = -K_t L x
\]

Finally, the tactile feedback is:

\[
\tau_U = r_s K_t L x
\]

### 6.2 Ad-hoc Method

In this section, a tactile feedback is created that is close in nature for the visual feedback a human would get if he was actually watching the ball. Table
<table>
<thead>
<tr>
<th>Direction of rolling</th>
<th>acceleration</th>
<th>human response</th>
</tr>
</thead>
<tbody>
<tr>
<td>counter-clockwise</td>
<td>accelerating</td>
<td>turn the knob clockwise</td>
</tr>
<tr>
<td>counter-clockwise</td>
<td>decelerating</td>
<td>turn the knob counter-clockwise</td>
</tr>
<tr>
<td>clockwise</td>
<td>accelerating</td>
<td>turn the knob counter-clockwise</td>
</tr>
<tr>
<td>clockwise</td>
<td>decelerating</td>
<td>turn the knob clockwise</td>
</tr>
</tbody>
</table>

Table 1: Suggested human response when watching the ball

1 shows how a human would basically think. One way to relate table 1 to the system states is shown in table 2. This suggests an input signal given by:

\[ \tau_M = -K \dot{\psi} \ddot{\psi} \]

K is a constant chosen experimentally.

<table>
<thead>
<tr>
<th>$\dot{\psi}$</th>
<th>$\ddot{\psi}$</th>
<th>$\tau_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: Translation of table 1 to system variables

The equivalent input signal would be:

\[ u = \frac{\tau_M}{K_t} = -\frac{K}{K_t} \dot{\psi} \ddot{\psi} \]  \hspace{1cm} (6.3)

And the tactile feedback will be,

\[ \tau_U = -r_s \tau_M = r_s K \dot{\psi} \ddot{\psi} \]
7 Simulations on The Linear System

The following two sections show the simulation results of all the previous theoretical work. All simulations are run on Matlab and Simulink. The Matlab codes are provided in the Appendix A. Table 3 shows the physical data for the ball and beam system:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value and Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kt</td>
<td>Torque constant</td>
<td>100 mNm/A</td>
</tr>
<tr>
<td>$R_B$</td>
<td>Ball radius</td>
<td>2 cm</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Iron density</td>
<td>7.8 g/cm$^3$</td>
</tr>
<tr>
<td>g</td>
<td>Free fall acceleration</td>
<td>10 m/s$^2$</td>
</tr>
<tr>
<td>$l_B$</td>
<td>Beam Length</td>
<td>1 m</td>
</tr>
<tr>
<td>$w_B$</td>
<td>Beam width</td>
<td>6 cm</td>
</tr>
<tr>
<td>$t_B$</td>
<td>Beam thickness</td>
<td>1 cm</td>
</tr>
</tbody>
</table>

Table 3: A list of physical data needed in the simulation

Note: The initial beam angle for all the simulations is 0.1 radians. It’s necessary to use such a small angle for the linearized model and the linear controller to work.

The Linear Quadratic Controller

The first step is to test the Linear Quadratic controller on the linear system. Figure 6 shows the block diagram of the linear system on Simulink. The block “State-Space” is used to present the state space equations found in section 4. The matrix C is set to the unity matrix I so that the output of the
block is a vector of the system states. The Gain in the feedback is set to the vector \( L \) explained in the previous section. Note that the choice of \( Q_1 = I \) and \( Q_2 = 20 \) is used to minimize the current requirement in exchange for an increase in the time needed to balance the ball. Thus, the desired feedback \( u = -Lx \) is obtained. For the first simulation, the transportation delay is set to zero and change \( r_{c0} \) along the beam. The simulation results is shown in figures 7 and 8. Note that the linear system appears to be symmetric around the origin, so only three graphs in each figure are obtained. One notices that the further the ball is from the origin, the easier it is to control it. The user is able to balance the ball in three seconds. Note that this time doesn’t change from one graph to another. The reason for this is that the optimum controller is a function of \( r_{c0} \).

Next, the initial ball position \( r_{c0} \) is kept constant and the human tactile delay is introduced as \( t_{delay} \). This is shown in figures 9 and 10.

Figure 6: A block diagram of the linear system with LQ controller
Figure 7: $t_{delay}=0$ and $r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5$ m

Figure 8: $t_{delay}=0$ and $r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5$ m
Figure 9: $r_c0=0.3$ m and $t_{delay} = 0.01, 0.03, 0.05, 0.07, 0.09, 0.11$ s

Figure 10: $r_c0=0.3$ m and $t_{delay} = 0.01, 0.03, 0.05, 0.07, 0.09, 0.11$ s
The simulation shows that it requires more current, and equivalently torque, to control the ball as $t_{\text{delay}}$ increases as expected. The user is able to balance the ball in less than three seconds except for the last graph even though $t_{\text{delay}}$ is changing. This shows how powerful the Linear Quadratic controller is. When $t_{\text{delay}} = 0.11\, \text{s}$, the system begins to oscillate, but it still converges in short time. The maximum current required is around 8 Amperes.

When $t_{\text{delay}} = 0.13\, \text{s}$, however, the system becomes unstable as shown in figures 11 and 12.

Figure 11: Current values for the unstable system when $t_{\text{delay}} = 0.13\, \text{s}$
Figure 12: LQ Controller Failure when $t_{\text{delay}} = 0.13s$
8 Simulations on the Nonlinear System

8.1 Checking the Validity of the Model

In the beginning of this section, a simulation is run to test the validity of the non-linear model. As can be seen in figure 16, the system has no controller, and the ball is left to roll from an initial position \( r_{c0} \). As seen in figures 17 and 18, the ball rolls indefinitely and the beam rotation is limited to \( \pi/2 \) which agrees with the model proposed in section 4.

Figure 13: A Block Diagram of the NonLinear System without a controller
Figure 14: $x_1$ for the nonlinear system without a controller

Figure 15: $x_3$ for the nonlinear system without a controller
8.2 Testing the Ad-hoc Method

This subsection shows the result of testing the ad-hoc method explained in section 6. The block diagram is shown in figure 13 with the feedback signal in equation (6.3). The method is tested directly on the nonlinear system with a delay of 0.4 seconds. The value of the feedback gain is changed trying to make the system stable, but to no avail. Note that we were not able to investigate the method thoroughly due to the lack of time. Some simulation results are shown in figures 14 and 15.

Figure 16: The nonlinear system with the Ad-hoc controller
Figure 17: $x_1$ for the nonlinear system with $K=0.002$

Figure 18: $x_3$ for the nonlinear system with $K=0.002$
8.3 Testing the Linear Quadratic Controller

In this subsection, simulations are run on the nonlinear system with the LQ controller. The block diagram is shown in figure 19. The feedback in this section is different from the previous one. To calculate L, it is chosen that $Q_1 = 20$ and $Q_2 = I$ because it took the user more time to control the nonlinear system. Thus, to decrease the time, an increase in the maximum current is noticed in all the following figures compared to the linear case.

As before, the transportation delay is set to zero and change $r_{c0}$ along the beam. The simulation results is shown in figures 20 and 21. The nonlinear system differs from the linear system for it is not symmetric around the origin (which is more realistic). Thus, six graphs can be seen in both figures.

Figure 19: A block diagram of the nonlinear system with the LQ controller
Figure 20: $t_{delay} = 0$ and $r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5$ m

Figure 21: $t_{delay} = 0$ and $r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5$ m
On another hand, as in the linear case, the further the ball is from the origin, the easier it is to controller it. The user takes more time to balance the system in this case, around six seconds. Note that this time doesn’t change from one graph to another. The reason for this is that the optimum controller is a function of $r_{c0}$.

The nonlinear system has less tolerance for the increase in $t_{delay}$. The maximum $t_{delay}$ before the system becomes unstable is 0.067 seconds, which is around half that of the linear system. The simulation results are shown in figures 22 and 23. When $t_{delay}$ is increased beyond 0.067 seconds, the system becomes unstable directly. This is shown in figures 24 and 25.

![Graph showing Input Current vs. Time](image)

Figure 22: $r_{c0}=0.3$ m and $t_{delay} = 0.01, 0.03, 0.05$ s
Figure 23: $r_{co} = 0.3$ m and $t_{delay} = 0.01, 0.03, 0.05$ s

Figure 24: Current values for the unstable system when $t_{delay} = 0.07$s
Figure 25: LQ Controller Failure when $t_{\text{delay}} = 0.07s$

Note that the maximum $t_{\text{delay}}$ allowed is related to the choice of $r_{c0}$ and the initial beam angle, 0.1 radians in this case. If the initial beam angle is chosen to be much smaller and the initial ball position is very close to the middle of the beam, it might be possible to raise the maximum $t_{\text{delay}}$ allowed before the system becomes unstable.
9 Conclusion

Heptic devices are considered to be a mechanism system and understanding how these devices work help the scientists to develop this technology. Some Haptic devices are controlled by computer and some require a direct control by human. Building such a device requires a high knowledge of mechanical concepts. This can be done by analyzing the system using either Newtonian mechanics or Euler-Lagrange equations. Some developers prefer to use Euler-Lagrange due to its simplicity. In this paper both methods have been used to make sure that the results are correct.

It has been seen that the Linear Quadratic Regulator was applied to balance the ball on the beam. The controller even works with delay but, human might not be able to imitate it perfectly. This is because; human response is hard to be modeled since different people will respond differently, especially when it comes to how fast the person is reacting to the tactile feedback from the system. Moreover, the Linear Quadratic Regulator was successful for up to 0.067 seconds delay with the nonlinear system, falling behind the goal of 0.4 seconds. The introduced Ad-hoc method was not able to stabilize the nonlinear system with a 0.4 seconds delay either. Thus, a human who functions like either of these methods cannot control the system.

To gain more insight on the system, one might change both the time delay and the initial ball position using a Montecarlo simulation. One can also linearize the system at other equilibria. Modifying the LQR to suit more the nonlinear system might produce better results. However, it is not expected to reach 0.4 seconds goal.

Future work can include studying the effect of changing the system struc-
ture on stabilizing and controlling system, for examples, having a smaller ball or longer beam or decreasing the initial beam angle. Another idea is to change the geometry of the system by hanging the beam a few centimeters away from the torque axis. This is what the model would be in reality, but the design studied in this report has been chosen for simplicity. If all of the above fails, it is better to chose a different controller. A more advanced approach can also use nonlinear control theory.
References


# List of Figures

1. The project’s main blocks ................................................. 8
2. Robotic Manipulator .......................................................... 15
3. The beam and ball system with External Forces ..................... 21
4. Calculating \( V \), The system potential .............................. 26
5. Modeling \( \tau_M \) ................................................................. 28
6. A block diagram of the linear system with LQ controller ......... 38
7. \( t_{delay} = 0 \) and \( r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5 \) m ........ 39
8. \( t_{delay} = 0 \) and \( r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5 \) m ........ 39
9. \( r_{c0} = 0.3 \) m and \( t_{delay} = 0.01, 0.03, 0.05, 0.07, 0.09, 0.11 \) s .... 40
10. \( r_{c0} = 0.3 \) m and \( t_{delay} = 0.01, 0.03, 0.05, 0.07, 0.09, 0.11 \) s .... 40
11. Current values for the unstable system when \( t_{delay} = 0.13 \)s ... 41
12. LQ Controller Failure when \( t_{delay} = 0.13 \)s ....................... 42
13. A Block Diagram of the NonLinear System without a controller 43
14. \( x_1 \) for the nonlinear system without a controller ............. 44
15. \( x_3 \) for the nonlinear system without a controller ............. 44
16. The nonlinear system with the Ad-hoc controller ................. 45
17. \( x_1 \) for the nonlinear system with \( K=0.002 \) ..................... 46
18. \( x_3 \) for the nonlinear system with \( K=0.002 \) ..................... 46
19. A block diagram of the nonlinear system with the LQ controller 47
20. \( t_{delay} = 0 \) and \( r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5 \) m ........ 48
21. \( t_{delay} = 0 \) and \( r_{c0} = 0.5, 0.3, 0.1, -0.1, -0.3, -0.5 \) m ........ 48
22. \( r_{c0} = 0.3 \) m and \( t_{delay} = 0.01, 0.03, 0.05 \) s ................. 49
23. \( r_{c0} = 0.3 \) m and \( t_{delay} = 0.01, 0.03, 0.05 \) s ................. 50
24. Current values for the unstable system when \( t_{delay} = 0.07 \)s ... 50

57
LQ Controller Failure when $t_{delay} = 0.07s$ . . . . . . . . . . . . 51
List of Tables

1  Suggested human response when watching the ball  .  .  .  .  .  36
2  Translation of table 1 to system variables  .  .  .  .  .  .  .  .  .  .  .  36
3  A list of physical data needed in the simulation  .  .  .  .  .  .  .  .  .  37
A Matlab Simulation Codes

%Linear1.m

clear
clf
Sample_time=0;

for i=0:5,

Kt=0.1;
R_B=0.02;
rau= 7.8e3;
m_B=(4/3)*pi*R_B^3*rau;
r_c0=0.5-0.2*i;
TD=0.01;
J_M=(1/12)*1*(1+0.0001);
J_B=(2/5)*m_B*R_B^2;
g=10;

A=[0 1 0 0
 0 0 (1-(2*m_B*(r_c0)^2)/(J_M + m_B*(r_c0)^2))*(m_B*R_B*g)/(J_M + m_B*(r_c0)^2) 0
 0 0 0 1
 0 0 0 1]
\[
\frac{g}{R_B} \frac{1}{(J_B/(R_B^2m_B) + 1) \left( (2m_B(r_c0)^2)/(J_M + m_B(r_c0)^2) - 1 \right) (m_BR_Bg)/(J_M + m_B(r_c0)^2) 0}
\]

\[
B = \left[ 0 \quad \frac{Kt/(J_M + m_Br_c0^2)}{0 \quad -Kt/(J_M + m_Br_c0^2)} \right]
\]

\[
C = \left[ 1 \quad 0 \quad 0 \quad 0 \\
0 \quad 1 \quad 0 \quad 0 \\
0 \quad 0 \quad 1 \quad 0 \\
0 \quad 0 \quad 0 \quad 1 \right]
\]

\[
D = [0; 0; 0; 0]
\]

\[
Q1 = \text{eye}(4);
\]

\[
Q2 = 20;
\]

\[
L = \text{lqr}(A, B, Q1, Q2);
\]

parNameValStruct.AbsTol = '1e-6';
parNameValStruct.RelTol = '1e-6';
parNameValStruct.MaxStep = '0.01';
parNameValStruct.StartTime = '0';
parNameValStruct.LimitDataPoints = 'off'; %Default is '1000'.
parNameValStruct.TimeSaveName = 't'; %Default is the name 'tout'.
parNameValStruct.StateSaveName = 'myx'; % SaveState = 'on'; %
%parNameValStruct.OutputSaveName = 'y1'; %Is not used.
% This command is for the case when one uses outports. Outport, see
% Simulink->Sinks->Out1.
parNameValStruct.SaveOutput = 'on'; %Is not used.
parNameValStruct.StopTime = '6';
x0 = [.1 ; 0 ; 0 ; 0];

x0String = strcat(' [',num2str(x0(1)),' ; ', num2str(x0(2)),' ; ', num2str(x0(3)),' ; ', num2str(x0(4)),' ] ');
set_param('LinearDelay1', 'LoadInitialState', 'on');
parNameValStructInitialState =x0String;
simout = sim('LinearDelay1', parNameValStruct);
t=simout.get('t');
figure(1)
plot(simout.get('u'));
xlabel('Time (seconds)');
ylabel('Input Current (Amperes)')
hold on
figure(2)
plot(t,simout.get('y'));
xlabel('Time (seconds)');
ylabel('State Variables')
hold on
end
clear
clf
Sample_time=0;

for i=0:5,

Kt=0.1;
R_B=0.02;
rau= 7.8e3;
r_c0=0.5;
TD=0.01+0.02*i;

m_B=(4/3)*pi*R_B^3*rau;
J_M=(1/12)*1*(1+0.0001);
J_B=(2/5)*m*B*R_B^2;
g=10;

A=[0 1 0 0
 0 0 (1-(2*m_B*(r_c0)^2))/(J_M + m_B*(r_c0)^2))*(m_B*R_B*g)/(J_M + m_B*(r_c0)^2) 0
 0 0 0 1
(g/R_B)*1/(J_B/(R_B^2*m_B) +1) 0 ((2*m_B*(r_c0)^2)/(J_M + m_B*(r_c0)^2) -1)*(m_B*R_B*g)/(J_M + m_B*(r_c0)^2) 0];
B=[0

63
\begin{align*}
\frac{K_t}{(J_M + m_B r_c 0^2)} \\
0 \\
-\frac{K_t}{(J_M + m_B r_c 0^2)}\end{align*}

\begin{align*}
C &= \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}; \\
D &= \begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix}; \\
Q_1 &= \text{eye}(4); \\
Q_2 &= 20; \\
L &= \text{lqr}(A, B, Q_1, Q_2); \\
\text{parNameValStruct.AbsTol} &= '1e-6'; \\
\text{parNameValStruct.RelTol} &= '1e-6'; \\
\text{parNameValStruct.MaxStep} &= '0.01'; \\
\text{parNameValStruct.StartTime} &= '0'; \\
\text{parNameValStruct.LimitDataPoints} &= 'off'; %Default is '1000'. \\
\text{parNameValStruct.TimeSaveName} &= 't'; %Default is the name 'tout'. \\
\text{parNameValStruct.StateSaveName} &= 'myx'; % \\
\text{parNameValStruct.SaveState} &= 'on'; % \\
%\text{parNameValStruct.OutputSaveName} &= 'y1'; %Is not used. \\
% This command is for the case when one uses outports. Outport, see \\
% Simulink->Sinks->Out1. \\
\text{parNameValStruct.SaveOutput} &= 'on'; %Is not used. \\
\text{parNameValStruct.StopTime} &= '6'; \\
x_0 &= \begin{bmatrix}
0.1 \\
0 \\
0 \\
0
\end{bmatrix};
x0String = strcat( ' [ ' , num2str(x0(1)) , ' ; ' , num2str(x0(2)) , ' ; ' , num2str(x0(3)) , ' ; ' , num2str(x0(4)) , ' ] ' ) ;
set_param('LinearDelay1' , 'LoadInitialState' , 'on' ) ;
parNameValStruct . InitialState =x0String ;
simout = sim('LinearDelay1' , parNameValStruct ) ;
t=simout . get (' t ' ) ;
figure (1)
plot (simout . get (' u ' ) ) ;
xlabel ('Time (seconds)' ) ;
ylabel ('Input Current (Amperes)' )
hold on
figure (2)
plot (t , simout . get (' y ' ) ) ;
xlabel ('Time (seconds)' ) ;
ylabel ('State Variables' )
hold on
end
Kt=0.1;
R_B=0.02;
rau= 7.8*e3;
m_B=(4/3)*pi*R_B^3*rau;
r_c0 =0.3;
J_M=(1/12)*1*(1+0.0001);
J_B=(2/5)*m_B*R_B^2;
g=10;

Step_time=1;
Initial_value=0;
Final_value=0;
Sample_time=0;

parNameValStruct.AbsTol = '1e-6';
parNameValStruct.RelTol = '1e-6';
parNameValStruct.MaxStep = '0.01';
parNameValStruct.StartTime = '0';
parNameValStruct.LimitDataPoints = 'off'; %Default is '1000'.
parNameValStruct.TimeSaveName = 't'; %Default is the name 'tout'.
parNameValStruct.StateSaveName = 'myx'; %
parNameValStruct.SaveState = 'on'; %
%parNameValStruct.OutputSaveName = 'y1'; %Is not used.
% This command is for the case when one uses outports. Outport, see
% Simulink→Sinks→Out1.
parNameValStruct.SaveOutput = 'on'; %Is not used.
parNameValStruct.StopTime = '3';
x0String = '[0.1 ; 0 ; 0 ; 0]';
set_param('NonLinearNoControl', 'LoadInitialState', 'on');
parNameValStruct.InitialState = x0String;
simout = sim('NonLinearNoControl', parNameValStruct);
t = simout.get('t');
figure(1)
plot(t, simout.get('x1'));
xlabel('Time (seconds)');
ylabel('Beam angle (radians)');
figure(2)
plot(simout.get('x3'));
xlabel('Time (seconds)');
ylabel('Ball rotation (radians)');
Kt=0.1;
R_B=0.02;
rau= 7.8 e3;
m_B=(4/3)*pi*R_B^3*rau;
rc0 =0.3;
J_M=(1/12)*1*(1+0.0001);
J_B=(2/5)*m_B*R_B^2;
g=10;

Step_time=1;
Initial_value=0;
Final_value=0;
Sample_time=0;

parNameValStruct.AbsTol = '1e−6';
parNameValStruct.RelTol = '1e−6';
parNameValStruct.MaxStep = '0.01';
parNameValStruct.StartTime = '0';
parNameValStruct.LimitDataPoints = 'off'; %Default is '1000'.
parNameValStruct.TimeSaveName = 't'; %Default is the name 'tout'.
parNameValStruct.StateSaveName = 'myx'; %
parNameValStruct.SaveState = 'on';
%parNameValStruct.OutputSaveName = 'y1'; %Is not used.
% This command is for the case when one uses outports. Outport, see
% Simulink->Sinks->Out1.
parNameValStruct.SaveOutput = 'on'; %Is not used.
parNameValStruct.StopTime = '3';
x0String = '[0.1 ; 0 ; 0 ; 0]';
set_param('NonLinearControl', 'LoadInitialState', 'on');
parNameValStruct.InitialState = x0String;

for i=1:1,
K=-0.002;

simout = sim('NonLinearControl', parNameValStruct);
t=simout.get('t');
figure(1)
plot(t, simout.get('x1'));
xlabel('Time (seconds)');
ylabel('Beam angle (radians)');
hold on
figure(2)
plot(simout.get('x3'));
xlabel('Time (seconds)');
ylabel('Ball rotation (radians)');
hold on
end
clear
clf

for i = 0:5;

Kt = 0.1;
R_B = 0.02;
rau = 7.8e3;
m_B = (4/3) * pi * R_B^3 * rau;
r_c0 = 0.5 - 0.2 * i;
TD = 0;
J_M = (1/12) * 1 * (1 + 0.0001);
J_B = (2/5) * m_B * R_B^2;
g = 10;

Sample_time = 0;

A = [0 1 0 0
     0 0 (1 - (2 * m_B * (r_c0)^2)) / (J_M + m_B * (r_c0)^2)) * (m_B * R_B * g) /
     (J_M + m_B * (r_c0)^2) 0
     0 0 0 1
     (g / R_B) * 1 / (J_B / (R_B^2 * m_B) + 1) 0 ((2 * m_B * (r_c0)^2) /)
     (J_M + m_B * (r_c0)^2) - 1) * (m_B * R_B * g) / (J_M + m_B * (r_c0)^2) 0];
B = [0
...
\[ \frac{K_t}{(J_M + m_B r_c 0^2)} \]

\[ 0 \]

\[ -\frac{K_t}{(J_M + m_B r_c 0^2)} \]

\[ C = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1] \]

\[ D = [0, 0, 0, 0] \]

\[ Q_1 = \text{eye}(4) \]

\[ Q_2 = 1 \]

\[ L = \text{lqr}(A, B, Q_1, Q_2) \]

\[
\text{parNameValStruct.AbsTol} = '1e-6'; \\
\text{parNameValStruct.RelTol} = '1e-6'; \\
\text{parNameValStruct.MaxStep} = '0.01'; \\
\text{parNameValStruct.StartTime} = '0'; \\
\text{parNameValStruct.LimitDataPoints} = 'off'; \text{Default is '1000'.} \\
\text{parNameValStruct.TimeSaveName} = 't'; \text{Default is the name 'tout'.} \\
\text{parNameValStruct.StateSaveName} = 'myx'; \% \\
\text{parNameValStruct.SaveState} = 'on'; \% \\
\text{parNameValStruct.SaveOutput} = 'on'; \% \text{Is not used.} \\
\%
\text{This command is for the case when one uses outports. Outport, see Simulink->Sinks->Out1.} \\
\text{parNameValStruct.StopTime} = '6'; \% \\
\]
x0String = '[0.1; 0; 0; 0]';
set_param('NonLinearDelay', 'LoadInitialState', 'on');
parNameValStruct.InitialState = x0String;
simout = sim('NonLinearDelay', parNameValStruct);
t = simout.get('t');
figure(1)
plot(simout.get('u'));
xlabel('Time (seconds)');
ylabel('Input Current (Amperes)');
hold on
figure(2)
plot(t, simout.get('y'));
xlabel('Time (seconds)');
ylabel('State Variables');
hold on
end
```matlab
% NonLinear2.m

clear
clf

for i = 0:2,

Kt = 0.1;
RB = 0.02;
rau = 7.8e3;
m_B = (4/3) * pi * RB^3 * rau;
r_c0 = 0.3;
TD = 0.01 + 0.02 * i;
J_M = (1/12) * 1 * (1 + 0.0001);
J_B = (2/5) * m_B * RB^2;
g = 10;

Sample_time = 0;

A = [0 1 0 0

0 0 (1 - (2 * m_B * (r_c0)^2)) / (J_M + m_B * (r_c0)^2)) * (m_B * RB * g) / (J_M + m_B * (r_c0)^2)

0 0 0 1

(g / RB) / (J_B / (RB^2 * m_B) + 1) 0 ((2 * m_B * (r_c0)^2)) / (J_M + m_B * (r_c0)^2)

(J_M + m_B * (r_c0)^2) - 1) * (m_B * RB * g) / (J_M + m_B * (r_c0)^2) 0];

B = [0

74
```
\[
\frac{K_t}{(J_M + m_B r_c 0^2)} - K_t\left(\frac{1}{J_M + m_B r_c 0^2}\right) \]

\[
C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};
\]

\[
D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix};
\]

\[
Q_1 = \text{eye}(4);
\]

\[
Q_2 = 1;
\]

\[
L = \text{lqr}(A, B, Q_1, Q_2);
\]

parNameValStruct.AbsTol = '1e-6';
parNameValStruct.RelTol = '1e-6';
parNameValStruct.MaxStep = '0.01';
parNameValStruct.StartTime = '0';
parNameValStruct.LimitDataPoints = 'off'; % Default is '1000'.
parNameValStruct.TimeSaveName = 't'; % Default is the name 'tout'.
parNameValStruct.StateSaveName = 'myx'; %
parNameValStruct.SaveState = 'on'; %
%parNameValStruct.OutputSaveName = 'y1'; % Is not used.
% This command is for the case when one uses outports. Outport, see
% Simulink->Sinks->Out1.
parNameValStruct.SaveOutput = 'on'; % Is not used.
parNameValStruct.StopTime = '6';

75
x0String = '[0.1 ; 0 ; 0 ; 0]';
set_param('NonLinearDelay', 'LoadInitialState', 'on');
parNameValStruct.InitialState = x0String;
simout = sim('NonLinearDelay', parNameValStruct);
t = simout.get('t');
figure(1);
plot(simout.get('u'));
xlabel('Time (seconds)');
ylabel('Input Current (Amperes)');
hold on
figure(2);
plot(t, simout.get('y'));
xlabel('Time (seconds)');
ylabel('State Variables');
hold on
end