Control of an inverted pendulum with a supplementary swing up controller Title

Prepared by:

Ngonidzashe Muzorori

MZRNGO002

Department of Electrical Engineering
University of Cape Town

Prepared for:

Mr De Maar

Department of Electrical Engineering
University of Cape Town

June 2021

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronics

Key Words: INSERT 4-6 KEYWORDS WHICH DESCRIBE YOUR PROJECT

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.
3. This final year project report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof

Name: **NGONIDZASHE MUZORORI**

Signature: _____ **Date:** 21 June 2021

Acknowledgements

Firstly, I would like to thank Almighty God for sustaining me and helping me during my studies. I also would like to thank my Parents who have been fully supporting me all the way even in times of failure.

I also thank my Supervisor Mr De Maar for always being supportive through the course of this project. He did support me whenever I had challenges and I am so grateful for that.

Lastly, I thank all my lectures for teaching and sharing all this invaluable knowledge.

Abstract

In this research, an inverted pendulum control system is designed and implemented on a microcontroller. The aim is to swing up a pendulum form rest and bringing it in to the balancing region. The balancing controller will takeover to balance the pendulum. A literature review is done on the nature of control systems and why inverted pendulums are good test beds. Controller methods both for swinging up and balancing of pendulum systems are analysed. In the methodology the procedure for designing a control system to control the inverted pendulum is outlined. First the mathematical modelling is carried out for first principles. System identification is done to get transfer functions. A feasibility study is done including system validation. After system validation, controller methods are chosen. Controllers are designed in Simulink as continuous time controllers. The chosen methods are state space feedback controller and Linear Quadratic Controller. Each method and parameters used is simulated in Matlab Simulink. Using specifications set, the best performing controllers designed are chosen and discretized for real time software implementation. Other non Linearities are incorporated in the design to make the solution more robust. The two methods are compared and the best controllers is implemented. A Kalman Filter is also designed to reduce noise on the sensor data and to estimate unmeasured states. The implementation is outlined and the result are obtained. Results are analysed and conclusions are made. At the end, recommendations are made on what could be done to improve

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background to the study

Control engineering is a field that deals with regulation and control of engineering systems. Without it the world would not be as we know it today with all the technological advancements. Control engineering finds its application in aerospace industry controlling commercial flights, flying rockets into space, flying of unmanned vehicles and controlling of satellites. In addition, control systems are applied in industrial automation, automobile driver and driverless vehicles and also in our homes for temperature control etc. Recently there has been increase in application of control engineering in robotics as the world is becoming more reliant on artificial intelligence. These are some of the few examples of application of control theory. Most of these control systems have some common properties. Systems are usually non-linear, multistate and unstable. In robotics there are systems that are also underactuated meaning they have more degrees of freedom than actuators. An inverted pendulum consists of all of these properties which makes it a good testbed for control systems. Many control engineers use inverted pendulum to test their controllers for performance. Apart from the pendulum being used as a testbed, there are many applications as well. These consists of balancing of humanoid robots, control of satellites orientation in space and balancing of rockets.

1.2 Objectives of this study

In this research the aim is to swing up and balance the inverted pendulum system using the control know attained. The system will be identified by running known inputs and observing the output at the end. The relationship of the input and the output are used to derive a transfer function that is a mathematical model of this system. Using text book information and other sources a suitable controller will be chosen and designed. A linear controller will be designed for balancing of the pendulum and a nonlinear will be for the swing up. The concept of gain scheduling will be applied to using these two controllers each at different positions.

1.2.1 Problems to be investigated

In this research the main problem is that of swinging up and balancing of an inverted pendulum. The other problems that are associated with control systems like nonlinearities, uncertainties, instabilities are also investigated in this research. In line with that, methods of system identification are also investigated as system cannot always be modelled from first principles. In this research an investigating on how to reduce sensor and other system noise was also investigated.

1.2.2 Purpose of the study

The purpose of this study is to show and understanding of the design process of control systems. From literature, a control system will be designed and tested. Evaluation will be done on the results obtained. The results will be compared with the literature expectations to prove correlation between the practical and the theory. The other purpose is that designing a control system for the pendulum provides a learning framework for designing of other control systems. While there are many projects done on the inverted pendulum project it remains the most fundamental close to reality control project. In this research the best methods will be chosen for design. The controllers will be chosen based on effectiveness as well as how easy they are to implement.

1.3 Scope and Limitations

This project focusses on designing a swing up and balancing of an inverted pendulum. Other aspects of controller design like disturbance rejection, noise attenuation and non-linear compensation will be

carried out. The control methods chosen are the state space feedback and linear quadratic regulators. Other classical control methods can be implemented based on performance. However, other complex methods like reinforcement learning to do both balancing and swinging up will not be done. Other methods like the sliding mode control methods will not be tried as they are complicated even though they are effective. Also other capabilities of inverted pendulum control like balancing on an inclined surface will not be done.

1.4 Plan of development

In this project an introduction is given on control systems in general and then on the topic of study which is the inverted pendulum. A literature review is done which focuses mainly on inverted pendulums. After literature review a methodology section explains the procedure of how the system was designed. It also consists of modelling using fundamentals and also using data. A feasibility study is carried out before a controller is designed. Controllers will be designed in Matlab using set out specifications for the system. The designs will be simulated in Matlab and analysed. The best controller will be chosen for implementation. A swing up controller will be designed and used with the best balancing controller. Results will be recoded, analysed and compared with expectation from literature.

2. Literature Review

2.1 The Inverted Pendulum

Inverted pendulums are generally used to test controller designs in control engineering systems. This is due to the fact that inverted pendulums have properties that emulate most of the control systems found in industry [1] [2] [3]. While they are simple, inverted pendulums are categorized as non-linear, underactuated and unstable control systems. They can also be configured as Single Input Multi Output (SIMO), high order systems [4] [5] which is what most of the control systems consist of. The inverted pendulums are associated with many fields due to those properties. They are found in both linear and non-linear control systems, in underactuated robotics [4], in mechanical systems [6] and in computer science and algorithms [7] to name a few. Apart from being used as testbeds for control systems, inverted pendulums have many applications in real life too.

Real life applications of inverted pendulum consist of:

1. Humanoid robots as underactuated systems,
2. Segway transportation systems (balanced 1 axel 2 wheeled carts)
3. Single wheeled bikes.

Other applications of inverted pendulum feedback control are, aircraft landing and stabilization against turbulent winds, stabilization of ships against tides [4], rocket launching and flight control [2]. One of the most common recent demonstration of inverted pendulum control is the landing of the Falcon 9 rocket by the company Space-X which was done in 2015. The aim of the mission was to recover used flown rockets for future use and hence reducing the cost of rocket launching and of space missions overall.

2.2 Structure and Dynamics of the Inverted Pendulum System

There are many configurations for inverted pendulum systems. The most common ones are the cart inverted pendulum and the rotational single arm system []. Other configurations consist of double [4] and triple pendulums which are of higher order. There are also, the reaction wheel balanced [8] and the dual axis reaction wheel balanced inverted pendulums [9]. These have different mechanical structure and use angular momentum of fly wheels for balancing. In this project the cart inverted pendulum system is used. It has 2 degrees of freedom which are, the movement of the cart usually denoted by x, and the angle of the pendulum by θ. The figure 1 below shows the inverted pendulum system.

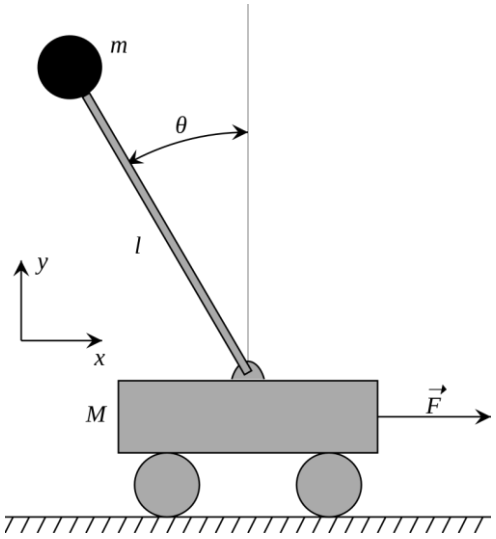

Figure 1: The Inverted Pendulum System

The system input voltage drives the cart and the cart drives the pendulum angle. This means it has one actuator and two possible outputs which are position and angle. Having 1 actuator and 2 outputs makes this system be classified as an underactuated mechanical system. Like most mechanical systems, the cart pendulum system suffers from friction [1] which affects its efficiency. The friction is on the rail of the cart and on the pivot of the pendulum. These affect the dynamics of the system and may introduce non linearities like dead bands to the drive signals and hysteresis. The friction at the pivot in particular causes damping in the pendulum swinging which causes the system to be underdamped. This dynamics information helps in deriving the model of the system mathematically and that enables the design of the control system. The following are the dynamic equations of the cart pendulum system which can be used for modelling, simulation and design of control systems for the inverted pendulum.

$$F = (M + m)\ddot{x} + \alpha\dot{x} + ml\left(\ddot{\theta}\cos\theta - \dot{\theta}^2 \sin\theta\right) \quad (1)$$

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

In (1), F is the force applied on the cart, M is the mass of the cart, m is mass of the pendulum, $\ddot{x}$ is the acceleration of the cart, $\dot{x}$ is the cart velocity, α is the coefficient of friction of the cart on the track, l is the length of the pendulum to its centre of mass, $\ddot{\theta}$ is the angular acceleration of the pendulum and $\dot{\theta}$ is its angular velocity. The equation (2) consists of some of the defined variables but the undefined are: I which is the moment of inertia of the pendulum and β which is the coefficient of friction of the swinging pendulum [1].

The 2 second order differential equations above shown from the term $\ddot{x}$, and $\ddot{\theta}$ imply that the pendulum system is 4th order. Since it has trigonometry functions and squares, it means it is also a non-linear system. These dynamic factors have to be taken into account when designing a control system for the inverted pendulum system. In the paper [3] it is noted that the pendulum system has its lowest energy when it is in the down position and highest when it is in the up position. This helps in the process of designing the swing up controller of the pendulum using energy based control [3]. Derivation of the dynamic equations is done in the methodology section.

2.3 Control systems of the inverted pendulum

As discussed above, inverted pendulum systems find their applications in control engineering design. This section discusses most of the aspects of control theory associated with inverted pendulum systems. The system dynamics gave the mathematical model which is used to design controller for the pendulum. Before the control system can be designed, the feasibility study has to be conducted. For example, the system has to be checked for stability, controllability and observability. The dynamic equations above are linearized about angle θ = 0 and θ = 180° and changed into state space of the form [10]:

$$\dot{x} = Ax + Bu \quad (3)$$

$$y = C^T x + Du \quad (4)$$

The form given is the matrix of Ordinary Differential Equations. The $\dot{x}$ is the time derivative of the states of the system. In [6] and [1], the states of the pendulum systems are given to be the angle θ and its

derivative $\dot{\theta}$, the position x and its derivative $\dot{x}$. Matrix A gives the relationship of differentials and their states and B gives the relationship of the differentials and the inputs. The input in the inverted pendulum system is the voltage which drives the motor of the cart. The C matrix relates the output y to the states and D to the inputs. Outputs of the pendulum system are usually the angle and the position. In this case these are of interest because the pendulum has to be balance at θ = 0 and because of the swing up control, the position has to be controlled as well. The equations above are used to test for controllability using the following criteria [10]:

$$\text{Let} \quad P = [B \;\; AB \;\; A^2B \ldots A^{n-1}B]$$

If the rank of P is n which is the number of states of the system, it means all the states are controllable [11]. In [1], the inverted pendulum was shown to be fully controllable. That means the angle and the position and their time derivatives angular velocity and cart velocity respectively can be controlled. Using the criteria in [10] the system can be tested for observability.

$$\text{Let} \quad Q = [c^T \;\; c^TA \;\; c^TA^2 \ldots c^TA^{n-1}]^T$$

If Q is full rank n it means the system is observable. In [1] the pendulum system was shown to be fully observable. That means all the states can be monitored as the input changes. This helps in design analysis for the system. The system also had to be tested for stability. To get the stability status of the system, the Eigen values of the matrix **A** have to be negative. It was found that the inverted pendulum is unstable since it has a positive Eigen value. However, since all the states are controllable, it means the system can be stabilized [10] [5]. This require finding new Eigen values which stabilise the system and that is the process of controller design. In [10] and [5], the following equations are used for stabilisation in state space.

$$\text{Let } u = -Kx$$

$$\therefore \; \dot{x} = (A - BK)x \tag{5}$$

Stable Eigen values for (5) can be chosen such that the system becomes stable [11]. The Eigen values will constitute the controller gains K which can make the system behave as expected. This is the process of feedback controller design. Controller values can also be chosen to achieve instability. This will help with the task of designing a swing up controller where nonlinear terms of K are necessary. This is a form of nonlinear control [6] which will be further discussed in the coming sections.

2.3.1 Control methods for Inverted Pendulums

Various control methods can be implemented on the inverted pendulum system. The choice of method comes down to what needs to be achieved and what resources are available for implementation. These are in a nutshell the most crucial decision making points. In this case 2 types of controllers are needed. The first one is to balance the angle of the pendulum at its unstable fixed point and the second one is to swing up the pendulum from it stable fixed point to the unstable point. These control tasks require different techniques of control. It is identified that the balancing task requires, linear feedback control and the swing up requires nonlinear feedback control [6] [3]. There are also other control systems which can achieve both the linear and nonlinear control [3] [7]. Some of these are called intelligent control systems which will be discussed briefly. In some cases, machine learning methods have been used for

swing up and balancing of the pendulum. The methods used in this report are linear and nonlinear feedback systems.

i. Why feedback control for Inverted Pendulum System

There several reasons why feedback control systems are used to control many engineering systems. This is because of the general properties of engineering system and also the intended behaviours required in a system. First the mathematical model of the systems is never accurate and there are uncertainties in any system [12]. States and other parameters also change in time and not repeatable. Unstable systems sometimes need to be stabilized to achieve the desired state or control [1…]. Also, there are input and output disturbances since systems operate in unpredictable environments. Lastly there is need for system efficiency to save energy while still getting quality outputs. The feedback control system is able to resolve those issues in most systems by comparing the output with the expected output and generating plant input with a suitable controller to reduce error in output. In some cases, where feedback is not applicable, different control methods can be employed. All these aspects, the inverted pendulum is susceptible to and that makes it a good testbed for control design. From the dynamic equations, it was shown that the pendulum is nonlinear and is also affected by friction. These are the sources of uncertainty in the pendulum model. To succeed in controlling or balancing the pendulum system, the model has to be linearized and put in a feedback loop system with a controller. The same applies to the fact that an inverted pendulum is an unstable system, Eigen values that can stabilize the angle must be found and used to design a controller. The pendulum must also be able to withstand changes in the external environment which comes in the form of input (system noise) and output disturbance. Examples of these are changes in voltage and small impulsive forces on the pendulum respectively. The pendulum must remain balanced in the unstable fixed point regardless. It is also beneficial to balance the inverted pendulum with the minimum energy possible to achieve efficiency.

ii. Controlling the pendulum angle (linear control)

In this section, the task of controlling the pendulum angle is discussed. There are many methods proposed for control of the inverted pendulum angle. For the inverted pendulum system, the model equations have to be linearized at the θ = 0 which is the balancing point. The linearization is shown in the Methodology section. The following are different methods for angle control and their capabilities and drawbacks:

Proportional, Integral Derivative control (PID)

PID control is a common method in controlling linear, Single Input Single Output control systems [2]. It is one of the most used control method due to its simplicity in design and implementation. Since the inverted pendulum has 2 states to be controlled, x and θ, two PIDs have to be designed. The PID controller takes the form $u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt} e(t)$, where K_p is the proportional gain, K_i is the integral gain and K_d is the derivative gain. The e(t) is the error or difference between desired output and the actual output like the angle and the position. The following is the general schematic of the inverted pendulum.

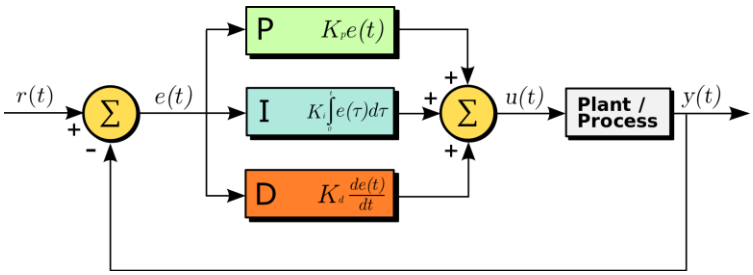

Figure 2: Structure of PID control

The *r(t)* is the set point which can be the angle or the position and *y(t)* is the output which can be the output angle or the output position. In [13] a PID to control the inverted pendulum is designed. Two PIDs are used to control pendulum angle and the cart position. Also 2 configurations are used which are the double series PID for position and angle control and the double parallel setup. It is shown that both setups can control the position of the cart and angle of the pendulum from an initial deviated angle. The double series has better robustness and the double parallel has better tracking. There some overshoots that can be noticed in the simulated result. In [4] it is shown that the PID does not produce optimum control results using the Integral Square Time Error cost function. Also in [7] the PI section of the PID gives large overshoots due to the integrator. This is not desired as the track length is short and that may render the solution infeasible in some cases. In [2], the same effect is observed.

Proportional Integral-Proportional Derivative controller (PI-PD)

The PI-PD control method is similar to the PID but the PI controller is placed in the forward loop while the PD is in the feedback loop. This configuration works well for unstable linear systems that have dead time and those that do not [4]. It is a good choice for the inverted pendulum control at linearized points for those reasons. In [4], the PI-PD controller is deigned and compared with the PID controller. The PI-PD controller uses PI to reduce steady state error and the PD in the feedback to shift the poles of the system from the origin to avoid marginal stability and instability. The following is the set-up of the inverted pendulum PI-PD controller.

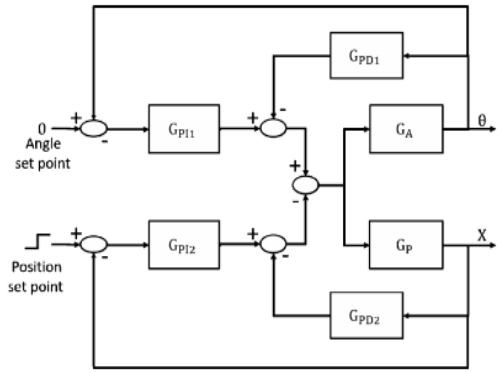

Figure 3: Structure of PI-PD control for the Inverted Pendulum

The transfer functions are G_A and G_P for the angle and cart position respectively. The G_{PI1} and G_{PD1} are the PI-PD for the angle and G_{PI2} and G_{PD2} are for the position. With the PI-PD designed with the same criteria as the PID, the two control methods are tested in [4]. To make the control much more efficient an Integral Square Error is used as a cost function to generate optimum gains of the PI-PD and the PID controllers. The following is the cost function.

$$J_i(\theta) = \int_0^\infty \left[t^i e(\theta, t)\right]^2 dt \qquad (6)$$

The t is the time component but in [4] a first order cost function is used. Controller gains take the form K_p and K_d. Under the same conditions (i.e. same gains), [4] shows that PI-PD controller performs better at pendulum balancing than PID. The report shows that the settling after disturbance is shorter and the over or undershoot angle from the set angle is smaller too implying better results at better control cost as compared to PID.

The Root Locus Method

The root locus method makes use of the information of the position of the poles and zeros of a system in order to synthesize a controller that satisfies the design requirements. The pendulum system has some poles which are in the right hand plane of the root locus which cause instability [10]. A designed compensator will have to move poles into the stable region with considerable stability. A simplified version of the transfer function of the pendulum system in [14] is used together with a cart motor transfer function. The following is the transfer functions from the equations of motion.

$$G(s) = \frac{\theta}{X} = \frac{-s^2/g}{(\tau_L+1)(\tau_L-1)} \text{ and } M(s) = \frac{X}{V} = \frac{k_m}{s(\tau_m s+1)}$$

With the closed loop denoted by *L(s) = G(s)M(s)* [14], the closed loop root locus poles are identified and tuned to design a compensator. In designing the compensator, adding a pole at origin helps cancelling one zero there and hence attracting the unstable pole at the right hand plane. Another reason is that a pole helps to achieve 0 steady state error for the angle. A zero is then added [14] to make sure the equation *p-z=2*. This makes sure that the asymptote of the root locus is $\pm 90°$ to achieve robustness and not marginal stability [5], [10], [14] . The controller takes the form,

$$K(s) = \frac{\tau_k s + 1}{\tau_k s}$$

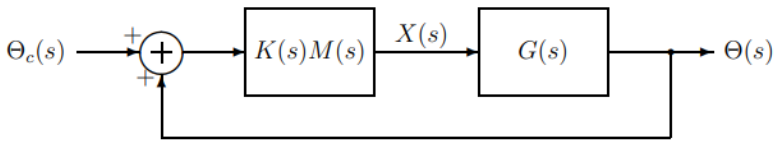

Figure 4: Lead/Lag Controller for the Inverted Pendulum System

This type of controller suffers from the wind up problem of an integrator which causes the cart to slide with an overshoot too large for the limited track length. That makes the system uncontrollable. However, adding a positive feedback around the motor and the compensator moves the poles to the left hence avoiding the integrator problem [4], [14].

The State Space Pole Placement Method

A system represented in state space allows for control of specific states of the system. States space control is very good for systems with multiple inputs and multiple outputs. The inverted pendulum has 4 states in total and in this case, the position and the angle can be controlled in one goal by choosing the gain vector *K* which stabilize the corresponding cart position and pendulum angle [11]. In [6] a state space controller is designed and simulated. The gain vector *K* is designed based on the Lyapunov's first method. This issues a control law that stabilizes the pendulum about its unstable equilibrium without accurate knowledge of parameters of the system. The controller archives stability very quickly from angle disturbance or from a pendulum swing up with very small amplitudes of oscillations and overshoots. The state space however, does not give an insight into the control law [2]. It does not allow

for optimisation of the controller. Despite that, it allows for state estimation which is beneficial to a lot of Multi Input Multi Output. This helps to reduce costs as measuring all system states can be expensive [11]. It also allows for input and output disturbance rejection as well as noise suppression in sensor measurements. The pendulum has angle and position measurements which are all prone to noise. A state estimator or a Kalman Filter can be designed in state space method to reduce these. In [2], a state space controller is designed and compared to the PID and LQR methods. The State space method was better than the PID in steady state error, settling time and overshoots. The LQR was however able to outperform the other 2 methods because it produced an optimum control solution at a lower cost.

Linear Quadratic Regulator

The LQR is similar to the State Space method in that it makes use of the state space model to control system states. It however offers a method for optimising the control depending on the desired outcome [11]. The ability to optimize controller output makes it cheap to control states and outputs whilst achieving the good results. The following is the structure of the LQR controller.

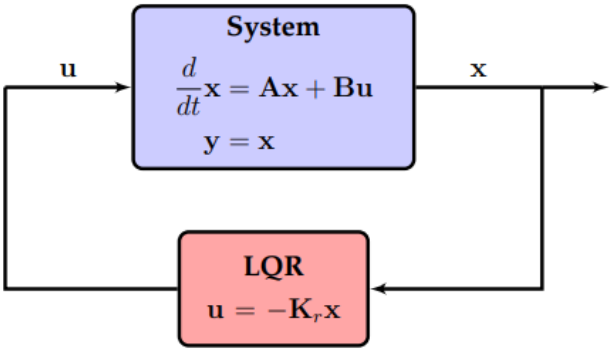

Figure 5: Structure of a Linear Quadratic Controller

The aim is to design K_r that stabilizes the system in this case the angle and position. However unlike in the state space method where Eigen values are chosen without an insight into how much control is required for a certain output, the LQR uses a cost function. The cost function is given by

$$J(t) = \int_0^t x(\tau)^* Q x(\tau) + u(\tau)^* R u(\tau)\, d\tau \tag{7}$$

The Q matrix is used to optimise the state output of a controller by giving the level of penalty to the state. In this case the angle has to be stable at 0rads (unstable equilibrium of the pendulum) with good disturbance rejection hence Q corresponding to θ must have higher penalty. Other states like position, velocity and angular velocity maybe tuned in Q accordingly as well. The R matrix minimizes the control output *u* by setting a penalty to it. This helps to archive optimal control with minimum controller output. It also helps to maintain the control signal within the operating range to avoid controller saturation. The chosen Q and R matrices are used in the Riccati's equation to obtain the controller gains. The Riccati equation is shown below

$$A^*X + XA - XBR^{-1}B^*X + Q = 0 \tag{8}$$

This equation is solved to give the solution that minimises the cost function $J = \lim_{t\to\infty} J(t)$ and give the controller gains $K_r = R^{-1}B^*X$ for control. Apart from the ability to optimise a controller using the Q and

R matrices, the LQR can also be used with a state estimator or a Kalman filter [11]. This is of great advantage because usually the angle and position sensor data is noisy. The system also has model uncertainties and disturbances. A Kalman filter takes in the noisy data and produce a better estimate of the system state. This yields better results in the controlling of the Inverted pendulum system with uncertainties reduced. The following is the implementation of the Kalman filter in conjunction with the LQR

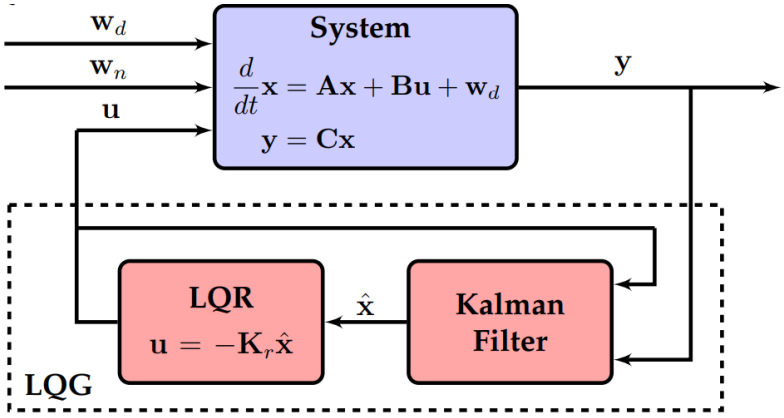

Figure 6: Linear Quadratic Gaussian that consists of LQR and Kalman Filter

The set up above is normally referred to as Linear Quadratic Gaussian controller (LQG). In [11], the LQG is implemented for an inverted pendulum and produced better results due to good estimates of angle. In [1], an LQR is used to control the inverted pendulum while testing the effects of friction of the system. The cart friction and the pendulum pivot friction are varied and their effects are recorded. It is shown that increase in friction causes longer settling time to the steady state and higher over and undershoots. This further proves that friction as an uncertainty has to be considered in controller design. The LQG can be used to counter the effects friction as an input disturbance. This can be done without explicitly measuring the value of friction coefficient on the cart and the pendulum.

Other control methods

There are several other methods that can be used to control the inverted pendulum. Each control method has its advantages and disadvantages. In [7], a fuzzy controller is designed and simulated to control angle and position of the inverted pendulum. A fuzzy controller is a non-linear intelligent control method that works as a digital controller. It implements fuzzy logic derived from fuzzy set theory. In a nutshell, a fuzzy controller has a fuzzification knowledge base (set) which is used for fuzzy reasoning. These are used to generate the desired control law. In its fuzzy set, the angle error and the angle rate error are used to generate the controller output. The fuzzy controller was compared to the PID and outperformed it significantly [7]. The fuzzy controller however requires more computational power to do control which can be a limitation. Another method that can be used is the reinforcement learning method. Reinforcement learning is a machine learning method that uses the Markov Decision process to learn. It consists of a set of environment, states, actions, reward and policy. The idea for the reinforcement learning as it applies to the inverted pendulum system is that it must learn in order to get an optimum policy that maximises the reward. The policy of the system seeks to achieve stability about the unstable equilibrium of the pendulum. This is achieved by optimizing actions and updating reward to further improve on actions taken until optimum policy is reached. In [15] a swing up and balancing policy from reinforcement learning was adopted and simulated. An offline BP neural network was used to train the model and getting the policy. The design allows for control policy generation without full knowledge of the system or model. The controller was able to swing up the pendulum from its rest

position to its up position. Other methods used are presented in [4] which are: Sliding Mode control, back stepping which are part of non-linear control and Neural network, genetic algorithm which are intelligent control methods. These types of controllers are used for different aspects of pendulum control like angle control on inclined surfaces. Some of these capabilities are beyond the scope of this research.

iii. Swing up controller for the Inverted pendulum (Non-linear)

In this research a swing up controller is also designed to bring the pendulum to the balancing region. A non-linear controller is capable of achieving this result. Ideally the cart has to be moved back and forth on the track to make the pendulum swing. There are several methods that can be used to achieve this swing up control of the pendulum. Other approaches utilise the non-linear nature of the system with positive feedback to do swing up and negative feedback for balancing all in one go [6] [16]. In this method there is no need for switching controllers when it comes to balancing. Another common method is the energy based method. This utilizes the fact that a pendulum has its lowest energy on the down position and highest when on the up. The controller pumps the energy until the pendulum reaches the desired angle. In this method there is need for switching to the balancing controller. There are other nonlinear methods like feedforward bang bang controller [17] which is susceptible to model uncertainties and disturbances. The bang bang controller can be used in conjunction with state space feedback controller. This method is good for rotating arm pendulum system with no track limitations. Sliding mode controller can also be used for swing up control. Below some of the swing up methods are further explained.

Non-Linear State Space Feedback

In this method the equations (3), (4) and (5) are used to design a swing up and balancing controller. In [6], this method is demonstrated and will be outlined here. In the paper [6], the aim is to design controller gains K that give instability at the stable point of the pendulum so that it swings up. The same gains should also give stability at the unstable point of the inverted pendulum. Analysis is done to figure out the range of the gain values that achieve this behaviour. The K values are implemented as in equation (5) and hence they take the form $u = k_1x + k_2\theta + k_3\dot{x} + k_4\dot{\theta}$. Equilibrium points of the system were found when $\left[x\ \theta\ \dot{x}\ \dot{\theta}\right] = \left[-\frac{k_2}{k_1}p\pi\ p\pi\ 0\ 0\right]$, where p is an integer. When p is even the equilibrium point is stable and when odd the point is unstable. It is noticed that when p is nonzero, the cart position is also non zero always and that causes drifting. To resolve that the gain that corresponds to the angle is changed to $k_2 sin\theta$. The cart position also needed to be limited during start up and the solution was to incorporate a squared angular velocity as follows, $(k_3 - c\dot{\theta}^2)\dot{x}$, where c is a tuneable positive constant. The overall control law becomes $u = k_1x + k_2 sin\theta + (k_3 - c\dot{\theta}^2)\dot{x} + k_4\dot{\theta}$. With this control law the desired conditions for equilibrium points was achieved and becomes $\left[x\ \theta\ \dot{x}\ \dot{\theta}\right] = [0\ p\pi\ 0\ 0]$. The gains corresponding to angle and angular velocity are aggressive so that there is fast swing up and gains corresponding to cart position and velocity are such that it stays at the desired point after balancing. This methods was successful as demonstrated in [6]. The swing ups were achieved under 2s of swinging. It was also noted that the angular velocity must be close to zero when the pendulum is about to reach the balancing point so that balancing is easy. The controller requires no switching because when the pendulum is down, the controller by design, achieves positive feedback and when up it achieves negative feedback.

Energy Based Swing Up Method

This method uses the mechanical method of the system to swing up the pendulum. In [17], the pendulum energy equation is defined as

$$V = \frac{1}{2}ml^2\dot{\theta}^2 + mgl(1 - cos\theta) \tag{9}$$

Taking the derivative of energy with respect to time gives

$$\dot{V} = ml\dot{\theta}cos\theta\ddot{r} \tag{10}$$

These equations are also used in [3] in the same manner to achieve swing up. Energy is controlled depending on the sign of the cart acceleration. When the product of $\dot{\theta}cos\theta\ddot{r}$ is positive it means $\dot{V}$ is greater than 0 and when its negative $\dot{V}$ is less than 0. This is used to generate a sinusoidal reference input so that energy can be given at the right time to amplify swinging up at any instance. The reference takes the form $r = -ka_c\omega_p cos\theta$ where a_c is the acceleration of the cart, ω_p is angular velocity of the pendulum and k is design parameter. The reference is used with a position controller. This is the control law is used to pump energy into the system taking into account the track limits. When the pendulum reaches the balancing point, the balancing controller takes over.

3. Methodology

In this section the design of the controller to balance the pendulum at the unstable upright position will be discussed and carried out. The swing up controller which brings the pendulum to the balancing position is also designed. The procedure to this design process will be outlined. First the system structure is described. The system is then analysed and modelled mathematically taking into account the system structure. After system identification and modelling, the system is validated using Matlab. The controller design will be done and a matric system will be set in order to validate the controller. Simulations of the controllers will be carried out and validated using the metric system or system specifications. The after simulation the results are obtained and analysed in the next section. The simulated controllers will be implemented on the actual system. The results of simulation and implementation are compared.

3.1 The Cart Inverted Pendulum system description

The cart inverted pendulum consists of a cart that is free to move on rail. A pendulum is attached to the cart and is free to swing about its pivot point. The pivot has a potentiometer which is used to measure the swinging angle. The cart is attached to a belt that is coupled with the motor. To drive the motor, a voltage supply -12V to 12V is connected. However, the larger voltage is controlled by a microcontroller that sends signals from 0 to 3V in the form of a PWM. The microcontroller controls the larger voltage through an H-Bridge using a duty cycle, and directions pins. That way the motor voltage may be varied from -12V to 12V. The angular position of the motor is measured using an encoder attached to its shaft. This is used to deduce the distance and hence the position of the cart at any instance.

3.2 System Modelling

To be able to design an angle balancing and swing up controller for this system, the system is modelled mathematically using system dynamics. There are many methods for system modelling which usually fall in the categories of black box modelling, grey box modelling and white box modelling. In this project black box is used to derive transfer functions. However, the white box modelling is used to give the system structure and validation of what is expected. The two methods are different in that white box modelling uses the dynamics or the physics of the system where as the black box uses data collected on the system to find the relationship between input and output.

3.2.1 Dynamic Modelling of the Inverted Pendulum System

i. The Cart and Pendulum

The model of the system is to be determined using the system parameters. The following is the setup of the inverted pendulum system

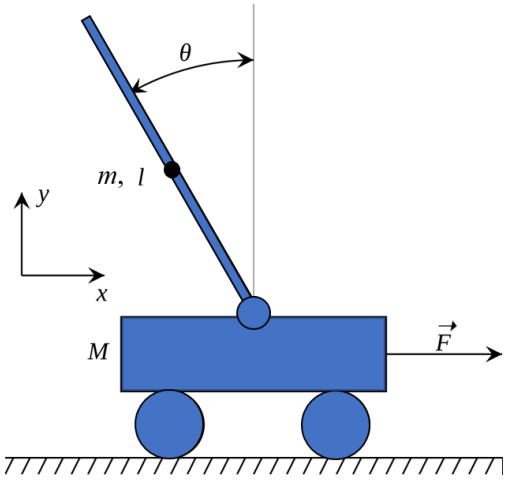

Figure 7: Free Body Diagram of the Inverted Pendulum System

The figure above shows the cart pendulum set up with its parameters. These parameters are used to derive the equations of motion. The Lagrangian $L(\dot{q}, q) = T(\dot{q}) - V(q)$ is used to derive the equations of motion. T is the total kinetic energy of the system, V is the total potential energy of the system and q are the generalised coordinates. There are 2 bodies in the system to be considered which are the cart of mass M and the pendulum of mass m. The pendulum length l is from the pivot to the centre of mass of the pendulum and it also has moment of inertia which is denoted by I. The generalises coordinates of the system for each body are given as follows in terms of $r = \begin{bmatrix} x \\ y \end{bmatrix}$:

Position for the cart

$$r_1 = \begin{bmatrix} x \\ 0 \end{bmatrix}$$

Position for the pendulum

$$r_2 = \begin{bmatrix} x + lsin\theta \\ lcos\theta \end{bmatrix}$$

The velocities of cart and pendulum using the generalised coordinates are given by:

$$\frac{d}{dt} r_1 = \dot{r_1} = \begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} \text{ and } \frac{d}{dt} r_2 = \dot{r}_2 = \begin{bmatrix} \dot{x} + l\dot{\theta}cos\theta \\ -l\dot{\theta}sin\theta \end{bmatrix}$$

The pendulum system has some movement constrains. One constraint is that the cart cannot move in the vertical direction, $y_1 = 0$. The length of the pendulum is always constant hence $x_2^2 + y_2^2 = 4l^2$, where x_2 and y_2 are x and y coordinates of the pendulum respectively.

The kinetic energy is as follows,

$$T = \frac{1}{2} \sum_{i=1}^{N} m_i \dot{r}_i^2 \tag{11}$$

Substituting the variables from the pendulum system in (11), the following Kinetic Energy equation for the system is obtained:

$$T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\left(\dot{x}^2 + 2\dot{x}\dot{\theta}lcos\theta + \left[l^2\dot{\theta}^2\,sin^2\,\theta + l^2\dot{\theta}^2cos^2\theta\right]\right) + \frac{1}{2}I\dot{\theta}^2$$

The potential energy of the system is given by

$$V = \sum_{i=1}^{N} m_i g h_i \tag{12}$$

The term h in the equations refers to the y coordinate of a body considered or its height. Also g is the gravitational acceleration.
Therefore, substituting the variables from the system the following equation is obtained;

$$V = mglcos\theta$$

The Lagrangian of the system after simplifying using trigonometric identities, grouping like terms and substituting $I = \frac{ml^2}{3}$. The Lagrangian becomes

$$L = \frac{1}{2}(M + m)\dot{x}^2 + \frac{2}{3}ml^2\dot{\theta}^2 + m\dot{x}\dot{\theta}lcos\theta - mglcos\theta \tag{13}$$

To find the equations of motion the following formula is used for each generalised coordinate.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q} = Q_i$$

The Q term denotes the non-conservative forces on a body in consideration. Therefore, the equations of motion are as follows

Variable x:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F - \alpha\dot{x}$$

Result:

$$F = (M + m)\ddot{x} + \alpha\dot{x} + ml(\ddot{\theta}cos\theta - \dot{\theta}^2 sin\theta) \tag{14}$$

Where F is the force applied by the motor to the cart and α is the coefficient of friction.

Variable θ:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = -\beta\dot{\theta}$$

This yields:

$$\frac{4}{3}ml^2\ddot{\theta} + m\ddot{x}lcos\theta - mglsin\theta + \beta\dot{\theta} = 0 \tag{15}$$

Equation (14) and (15) are non-linear equations of the cart-inverted pendulum. Since the pendulum will be balanced at 0°, which is a fixed point, the following assumptions can be made to linearize the model.

$$cos\theta \approx 1,\ sin\theta \approx \theta \text{ and } \theta^2 \approx 0$$

The equations (14) and (15) become

$$\ddot{x} = -\frac{\alpha}{M+m}\dot{x} - \frac{ml}{M+m}\ddot{\theta} + \frac{F}{M+m} \tag{16}$$

$$\ddot{\theta} = -\frac{3}{4l}\ddot{x} - \frac{3\beta}{4ml^2}\dot{\theta} + \frac{3g}{4ml^2}\theta \tag{17}$$

The equations above are the dynamic equations of the pendulum system from the first principles. The Lagrangian is used because it simplifies the derivation. The equations can be simplified but the form given will suffice for what they will be used for in this research. The issue with this way of modelling is that system parameters have to be measured and substituted into the model for controller design. However, the mathematical model helps to give structure of what the transfer function must look like. Further simplifications of the model can then be done with right assumptions.

ii. ***The DC Motor***

A DC motor is an electro-mechanical system. it consists of an armature circuit with resistance R, inductance L and the electromotive force E. Input to the motor is the voltage V which drives the current I that makes the rotor spin. On the output which is the shaft, torque and speed are produced. In the inverted pendulum system, the motor drives the belt which in turn moves the cart linearly. The following is an electro-mechanical model of the motor.

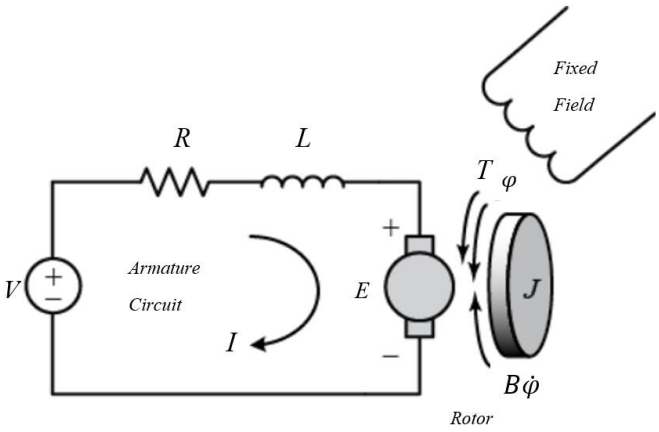

Figure 8: Electromechanical Structure of The Electric Motor

Using the Kirchhoff's voltage law and Newton's laws, the following equations can be derived.

$$V = IR + L\frac{dI}{dt} + E \tag{18}$$

The electromotive force produced by the motor of the back EMF is given by

$$E = K_e\dot{\varphi}$$

Substituting for E in (18) and assuming that L is negligible yields

$$V = IR + K_e\dot{\varphi} \tag{19}$$

The torque T_m produced by the motor is given by

$$T_m = K_m I$$

Using this equation, the Torque can be related to the input voltage as follows by substituting for I in (18)

$$V = \frac{T_m}{K_m}R + K_e\dot{\varphi} \tag{20}$$

In mechanical terms, torque is also given by

$$T_m = J_m\ddot{\varphi} + B_m\dot{\varphi} + Fr$$

J_m is the moment of inertia of the motor, B_m is the viscous damping on the motor shaft, $\dot{\varphi}$ is the angular velocity of the motor, $\ddot{\varphi}$ is the angular acceleration, Fr is the torque produced by the motor to drive the cart and hence F is the force and r is the radius of the pulley.

From before the force equation is (17). However, $m \ll M$ and hence m terms are neglected. This implies that $F = M\ddot{x} + \alpha\dot{x}$ which can be substituted in the Torque equation. Also it is realised that the angular displacement of the motor is related to the position of the cart through the coupling belt. Therefore,

$$x = r\varphi$$
$$\dot{x} = r\dot{\varphi}$$
$$\ddot{x} = r\ddot{\varphi}$$

These are also substituted in the torque equation to give

$$T_m = J_m\frac{\ddot{x}}{r} + B_m\frac{\dot{x}}{r} + Mr\ddot{x} + \alpha r\dot{x} \tag{21}$$

Substituting (21) into the (20) gives

$$V = \frac{1}{K_m}(J_m\frac{\ddot{x}}{r} + B_m\frac{\dot{x}}{r} + Mr\ddot{x} + \alpha r\dot{x})R + K_e\frac{\dot{x}}{r}$$

Rearranging and assuming $K_e = K_m$ for a constant magnetic field motor, the equation becomes

$$V = \frac{R}{K_m r}\left[(J_m + Mr^2)\ddot{x} + \left(\frac{K_m^2}{R} + B_m + \alpha r^2\right)\dot{x}\right] \tag{22}$$

The following constants can be simplified as follows

$$\text{Let} \quad J = J_m + Mr^2$$

$$B = \frac{1}{J}\left(\frac{K_m^2}{R} + B_m + \alpha r^2\right)$$

$$\frac{1}{A_m} = \frac{R\,J}{K_m r}$$

This yields the final equation that relates input Voltage to cart position

$$V = \frac{1}{A_m}(\ddot{x} + B\dot{x}) \tag{23}$$

The relationship of the motor voltage and the cart position allows for control of the position. The parameters used to derive the model can be measured on the system but it is tedious to do so and sometimes not practical. A much more efficient method is to run the system and collect data to deduce the relationship of the inputs with outputs. The dynamic model still helps to give the format of the transfer function.

3.2.2 Data Modelling of the Inverted Pendulum

In this section the process of system identification is carried out by first applying an impulse on a cart to get the impulse response of pendulum. This gives the relationship between the cart position x and the angle θ. The relationship of the cart position and the input voltage is also derived. This is because the input voltage directly controls the cart position. By changing the cart positions, the angle can also be controlled hence the need for those 2 transfer functions. The objectives are to find the transfer functions of the angle vs the position at the down position and the up position and the pendulum. The two transfer functions are used for swing up, and balancing of the pendulum respectively. The voltage and position transfer function is also derived so that swing up and balancing of the pendulum can be done.

i. Transfer Function of Cart position

The transfer function of the is derived as follows:

1. A step Voltage is applied to the motor making it run from the start of the track to just before the end of it (to avoid the cart hitting the end)
2. The value of the Step voltage is recorded
3. The step response of the cart position is obtained via the encoder which measures the cart position
4. The derivative of the cart position or the velocity is calculated using position and the sampling time
5. The response velocity at a given voltage is obtained
6. The Duty Cycle is varied to get response at different input Voltages.
7. In each case the amplitude and the response time are recorded
8. These values are averaged to get the transfer function

The velocity vs time is much easier to use to derive the transfer function of the system because it is a first order system.

The following is the response of the velocity vs time with a known input voltage.

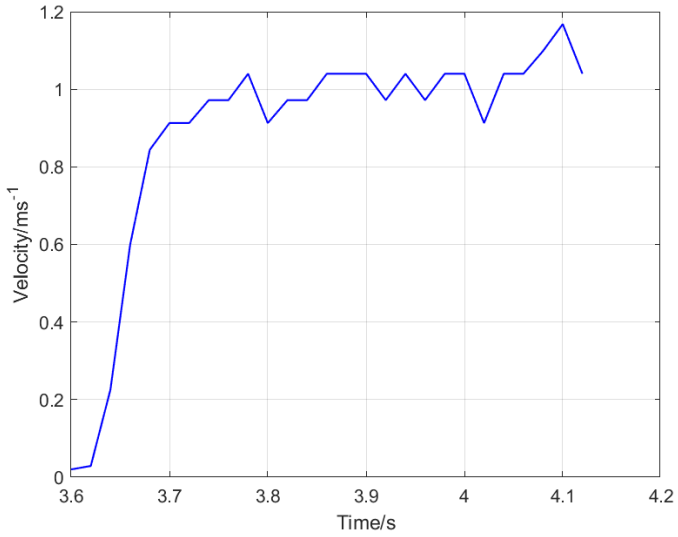

Figure 9: Plot of Cart Velocity vs Time at 70% Duty Cycle of Supply Voltage

The response above resembles a first order system according to [16]. This graph is used to find the transfer function of velocity and voltage. However, velocity is a time derivative of position and hence an integrator is multiplied to the velocity transfer function to get the position transfer function.
The transfer function of the first order system takes the following format in Laplace.

$$g(s) = \frac{A}{\tau_m s + 1}$$

Applying a step of amplitude V yields a step response of

$$\frac{g(s)}{s} = \frac{A_r}{Vs(\tau_m s + 1)}$$

Several responses are obtained in order to calculate A which is the normalised amplitude and T which is the time constant.

$$A = \frac{A_r}{V}$$

Where A_r is amplitude of the response to a given Voltage V, T is time at 63% of the maximum amplitude A [16]. According to [16], the amplitude A_r is found by taking the value of the response at steady state t_s

Many responses were taken and are shown in the following table.

Table 1: Table of Motor Input Voltage, Cart Velocity/Voltage and Time Constant for System Identification

Input Voltage Duty Cycle (%)	Voltage Step size (V)	Maximum Amplitude A (m/s/V)	Time constant T (s)
40	1.2	0.4555	0.0200
50	1.5	0.4773	0.0300
60	1.8	0.5055	0.0500
70	2.1	0.4849	0.0200
80	2.4	0.4765	0.0300
90	2.7	0.4994	0.0400

Average of A is calculated to be 0.4832, standard deviation… and T to be 0.032, standard deviation. The velocity transfer function

$$g(s) = \frac{0.4832}{0.032s + 1}$$

To get the transfer function of the position vs voltage, the velocity transfer function is integrated as follows in Laplace

$$M(s) = \frac{0.4832}{0.032s^2 + s} \tag{24}$$

Previously, the relationship of voltage and position is given in (23). Taking the Laplace transform of (23) and some algebraic manipulation, the result gives

$$\frac{X(s)}{V(s)} = \frac{A_m}{s(s + B)} \tag{25}$$

The equation (24) and (25) take the same form which validates the model obtained using data. Therefore,

$$M(s) = \frac{16}{s(s + 33)} \tag{26}$$

ii. ***Transfer function of the pendulum***

In this section, the transfer function of angle vs position is derived. To get the system model of the inverted pendulum, the following processes were followed

1. A voltage impulse was given to the system, which makes the cart move a short distance, swinging the pendulum
2. The pendulum is left to swing until it comes to rest while the pendulum angle is being recorded by the potentiometer
3. The angle vs time graph is plotted.
4. From the graph, terms like the damping coefficient ζ, the natural frequency ω_n and the damped frequency ω_d can be obtained
5. These values are used to derive the transfer function

The following is the graph of angle vs time which is used to derive the parameters of the transfer function.

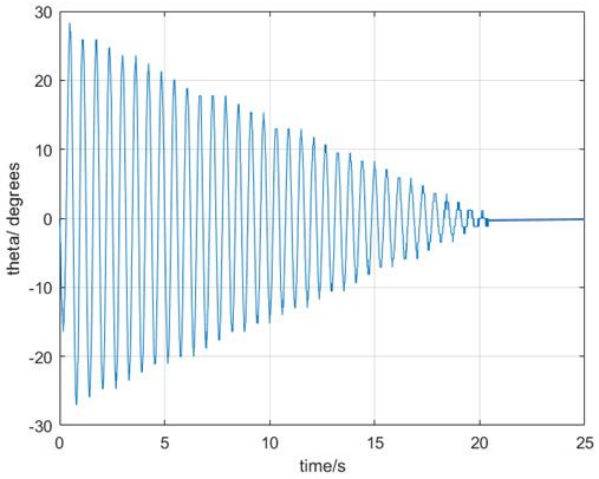

Figure 10: Plot of Pendulum Angle vs Time

The graph above shows the impulse response of the pendulum system. This response is when a pendulum is down because the up position is unstable and cannot be used to derive the transfer function using this method. From the response it is shown that the pendulum system can be modelled as a second order underdamped system [16]. In [16] the underdamped second order systems are of the following format

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} \tag{27}$$

The parameters are given by the following equations where ω_n is the natural frequency and ζ is the damping ratio.

$$\omega_n = \frac{2\pi}{T}$$

$$\zeta = \frac{Exponential\ Decay\ Frequaency}{Natural\ Frequency} = \frac{1}{2\pi}\frac{Natural\ Period}{Exponential\ time\ constant}$$

The graph Figure 10 above in time domain can be described mathematically as follows

$$\Theta = Me^{-\zeta\omega_n t}\sin(\omega_d t)$$

The aim is to find $\zeta\omega_n$ which is usually denoted by σ [16] and is the time constant of the envelop over the oscillating graph. The envelop is an exponential decay function and hence $\zeta\omega_n$ is its time constant. To find the time constant of the exponential decay graph, the time when the envelope of Θ reaches 37% of its initial value M is determined. To improve on accuracy, the time constant is determined on several sets of data each with a different impulse value as its input. The average of the time constant is used to calculate the damping ratio. The table below shows the different values used to determine the time constant $\zeta\omega_n$

Table 2: Showing Response Time constant, Natural Frequency and Damping ratio of Pendulum

Time constant $\zeta\omega_n$	Natural frequency ω_n	Damping ratio ζ

13.14	15	
14.65	15	
14.99	15	
12.67	15	0.01

The natural frequency of a swinging pendulum is given by $\omega_n = \frac{2\pi}{T}$, where T is the period of the pendulum. The swinging pendulum second order system is therefore given by

$$\Theta = \frac{225}{s^2 + 0.3s - 225} \tag{28}$$

The transfer function to control the angle is with respect to the value of x and hence the obtained Θ has to be normalised. This is because when a voltage impulse is given, the cart and pendulum accelerates to a certain distance. From the dynamic modelling section, the relationship of the angle θ and the position x is given by (17).

Taking the Laplace transform of (17) and rearranging gives

$$\frac{\theta(s)}{X(s)} = \frac{-\frac{3}{4l}s^2}{s^2 + \frac{3\beta}{4ml^2}s - \frac{3g}{4ml^2}} \tag{29}$$

Equations (28) and (29) are of a similar format except that (29) has 2 zeros introduced by incorporating the positon of the cart. Matching parameters and introducing a normalising constant for position gives the following

$$\frac{3\beta}{4ml^2} = 2\zeta\omega_n$$

$$\frac{3g}{4l} = \omega_n^2$$

$$A_p = \frac{3}{4l}$$

Using the impulse displacement A_p was found to be -4.125. Finally, the transfer function of the pendulum is given by

$$P(s) = \frac{-4.125s^2}{s^2 + 0.3s - 225} \tag{30}$$

This transfer together with the voltage-position function will be used to build a controller. Before building a controller both for balancing and for swing up, the system models above must be validated. This is part of feasibility study where it will be determined if transfer functions obtained can be used to represent the model.

3.3 System Validation

In this section the system models are validated. The way the validation is done is, the plant is simulated in Matlab using the same parameters as in the real experiment. The output of simulation is compared with the output of the actual plant. If the model is within the acceptable range of accuracy, it can be used to design a controller.

3.3.1 Position-Voltage transfer function validation

The following is the setup to validate the voltage-position function in Simulink.

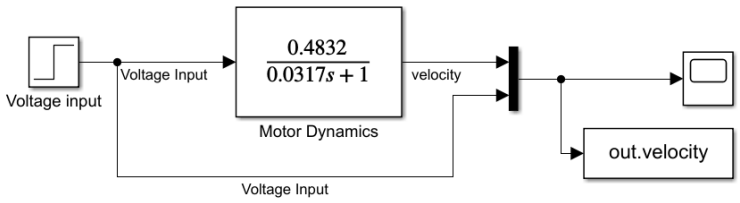

Figure 11: Simulink Block Diagram to Simulate Cart Position

Data is collected from a Simulink model when an input same as that applied to the real system is given. The output of the Simulink model and that of the real system are plotted on the same axis for comparison. The following is plot of the two outputs

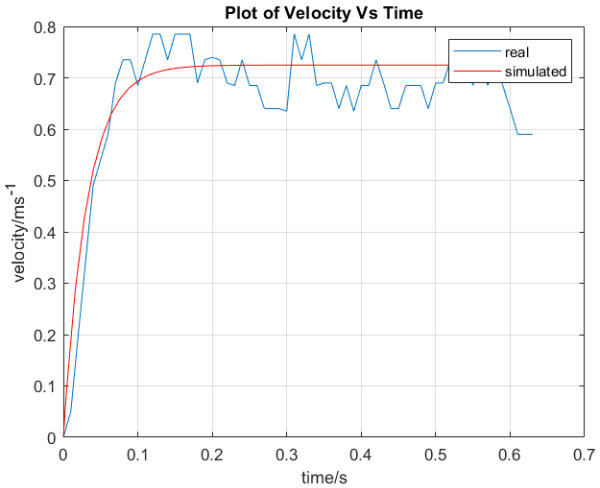

Figure 12: Plot to Compare Real vs Simulated Velocity vs Time of the Cart

The outputs above show that the real system and the simulated system follow the same profile. However, are some significant mismatches due to the noisy sensor (encoder) used to measure position data. The noise is introduced when the encoder successive positions' derivatives are calculated. The derivative of a signal does amplify noise and hence the velocity signal is noisier. A state estimator is best for tackling this problem which will be implemented in the controller design section. Position transfer

function which is derived from the velocity transfer function is also validated. The following are the graphs of the simulated system vs the real system

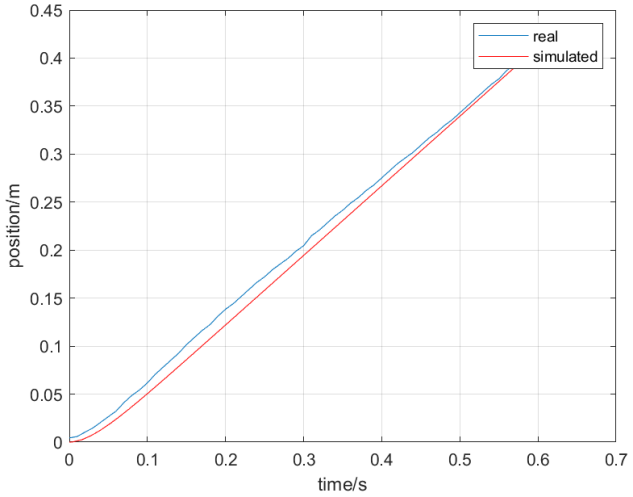

Figure 13: Plot to Compare Measured Position of the Cart Vs Simulated

The figure above shows a strong correlation between the real system and the simulated system with considerable amount of error. Since a feedback system will be designed, the mismatch will be resolved. It is also noticed that the mismatch is due to sensor bias which must be readjusted on implementation. The encoder position data is much smoother than the velocity's because, the derivative amplified noise in the sensor data for velocity. Data shown is satisfactory such that the obtained transfer function can be used as a model.

3.3.2 Angle-Position transfer function validation

The model obtained for the pendulum using the real system is tested against simulated. The simulation set up is shown below using MATLAB Simulink. Transfer function for the inverted pendulum linearized at θ = 180° is used in simulation.

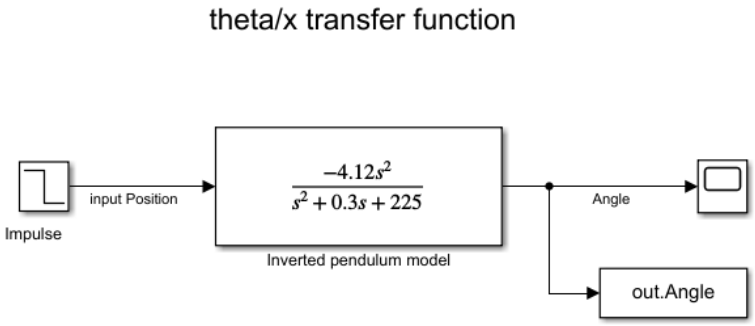

Figure 14: Simulink Block to Simulate The Angle Transfer Function of the Pendulum

An impulse was supplied in the form a short pulse or step input. The value of the step matches the step that was used in the real system for comparison. Figure below shows the angle plot of the real system versus the simulated model. The plots show the behaviour of the pendulum when it is swing about its bottom equilibrium.

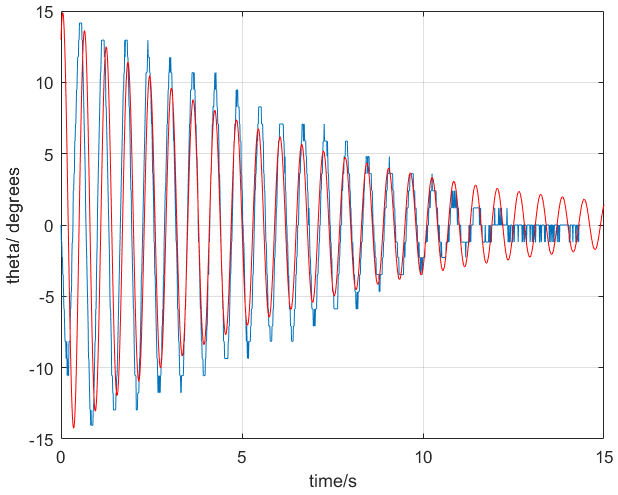

Figure 15: Plot to Compared Real pendulum Angle vs Simulated

The two plots show some correlation in frequency of oscillation and in amplitude. Due to sensor noise some major differences can be observed in amplitude of the outputs. The amplitude of the real system is not consistent with the model and does not decaying exponentially the same as the simulated. This is due to non-linearities in the real system caused by friction. The model can still be used to represent the system but the non-linearities will be considered when deriving a controller.

3.3.3 Combined Cart Pendulum System

The transfer functions of the separate sections obtained above are combined to give the overall transfer function of the cart pendulum system. Depending on the control objective which is to balance the pendulum on its unstable fixed point (i.e. the upright position) a series system configuration is chosen. The following Simulink series model is the chosen as the configuration of the open loop control system.

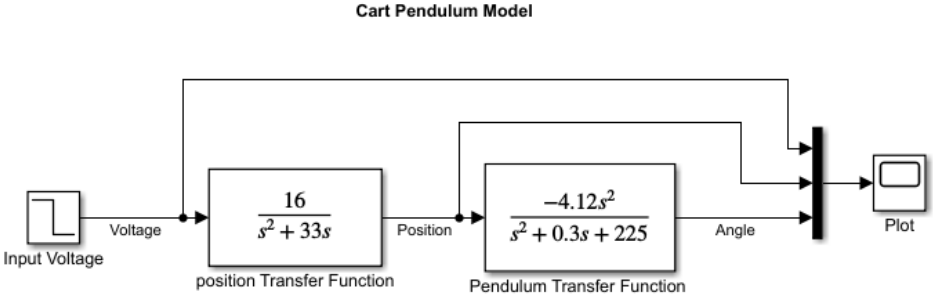

Figure 16: Combined Transfer Function of the Inverted Pendulum in Simulink

The series configuration is such that the cart position is the input of the pendulum transfer function. The pendulum function can be changed to suit for inverted pendulum control in the up position. In this set up, the overall system is simulated when the pendulum is at the bottom stable point. The following plot is the impulse response of the system.

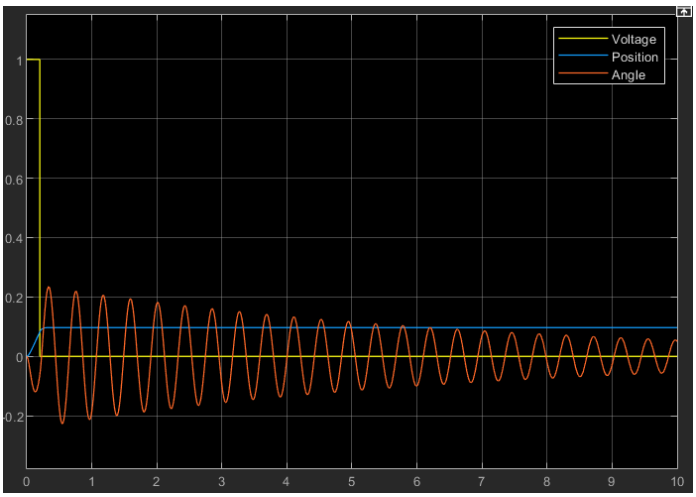

Figure 17: Simulation of the Inverted Pendulum Transfer Function in Simulink

In the simulation, an input voltage is given as a pulse. The input voltage causes the cart to accelerate from rest. Since the input is a pulse, the cart comes to rest at a certain position. The moving of the cart causes the pendulum to swing from rest and when the cart stops the pendulum continues to oscillate until it comes to rest. This plot validates the model for the pendulum system used. On the real system the same type of output is obtained. The simulation above is for the idealized system but as it has been established, the system has uncertainties and non-linearities causing a model mismatch. The real system model is not constant throughout and hence the mathematical model obtained is not its full representation. To get quality results, non-linearities have to be modelled and incorporated into the design. The next section is modelling of these non-linearities.

3.3.4 Non-Linearities Modelling

The system validation process reveals some mismatches of the real pendulum system and the simulated system. As discussed before these non-linear entities are due to the friction on the pendulum itself, the friction of the cart on the track and the slack of the belt that transmits torque of the motor to the cart pendulum system. There are many other sources of non-linearities like the state uncertainties. These will not be modelled in this section but will be solved during controller design and simulation using Filters. There are three non-linearities that are modelled in this section. These are the voltage input dead band, the input voltage saturation and the backlash or hysteresis.

i. Input Voltage Dead Band

The static friction of the cart causes the input voltage to have a dead band. This is the voltage which when applied to the motor, the cart does not move or that it does not move linearly. In the system used, the dead band was obtained using the iterative method. Different input voltages from zero were supplied to the system while the output is being observed. This was done to both direction of the cart by switching the polarity of the input voltage. It was observed that the system did not respond to voltages below 20% ($\pm$0.6V) of the maximum supply voltage which is 3V. This dead band can be represented graphically as follows

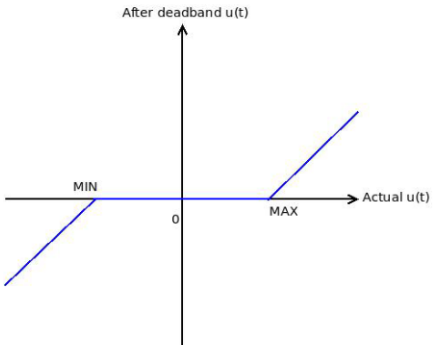

Figure 18: Plot of System Input Dead Band

Mathematically,

$$U_{MIN} < U_{actual} < U_{MAX}, U = 0$$

When generating a control signal *U*, this dead band will be compensated to give the required actuation signal to the plant (i.e. the cart system).

ii. ***Input Voltage Saturation***

After the input dead band, the input voltage is linearly related to the velocity of the cart. However, beyond 90% of the maximum supply voltage, the response is the same (i.e. the velocity of the cart does not change with further increase in input). To also protect electronic components, the saturation has to be modelled so that the input does not exceed limits. The following is the representation of input saturation.

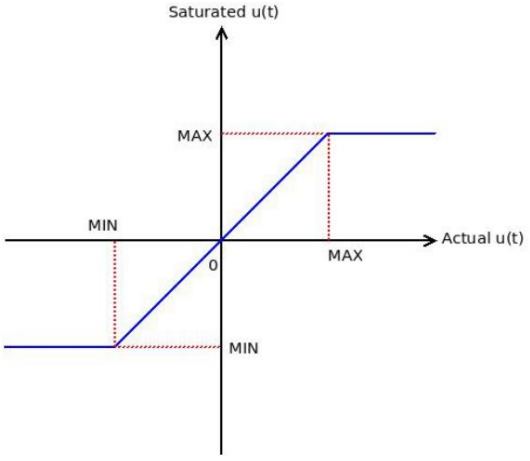

Figure 19: Plot of Input Saturation of the system

The plot shows that the input signal is clipped when it reaches the saturation value. In mathematical terms this can be modelled as

$$U_{MIN} < U < U_{MAX}$$

This is the linear region of the system. In the controller, clipping of actuation value will be implemented to make the signal to not go into the unwanted region.

iii. ***Backlash***

Because the cart is coupled to a belt for the motor to drive the cart, there is backlash caused by the slack of the belt. This causes the cart to travel different distances in different directions for the same amount of input. Also the response is delayed while the belt is catching up with the cart. To remedy this, the belt

could be tightened but that causes more friction to the motor shaft which is another form of non-linearity. That can also make the motor be difficult to drive, hence the backlash has to be modelled. The plot below shows the backlash.

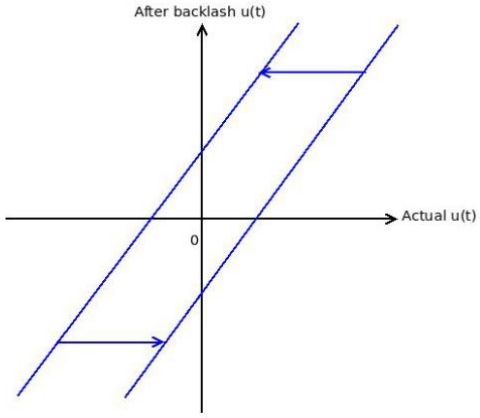

Figure 20: Plot of Input Backlash

This backlash will be quantified and incorporated in the controller design in order to produce good control results. The linear model and the non-linear model of the system have now been obtained and the controller design can be carried out. In the following section, the controller design is discussed.

3.4 Controller Design

In this section a swing up and balancing controller will be designed for the pendulum. The sing up controller is a non-linear and so a non-linear controller will be designed. The balancing of the inverted pendulum is a linear controller hence the system will be linearized at a fixed point θ = 0°. Before the controllers can be designed, feasibility study has to be carried out. However, part of feasibility has been done in system validation section. In this section, the system stability, controllability, and observability will be analysed. The transfer function of the system is changed to state space for easier analysis. The following is the state space model of the inverted pendulum linearized at the top fixed point.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -33 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 136.125 & 225 & -0.3 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 16 \\ 0 \\ -66 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

The above linearized equation takes the form $\dot{x} = Ax + Bu, y = C^T x + Du$, where A and B are linear combination of states and of input respectively. The C matrix give relationship of states and the output while D gives the relationship of output to the inputs. In this system $Du=0$ which means there is no feedforward of inputs on the output. Controllers will first be designed in continuous time for easier analysis and for quick simulations. The chosen control solution will be discretized so that it can be implemented on a microprocessor. An STM32F4 microcontroller will be used for implementation.

3.4.1 Feasibility Study for control of the system

i. Stability of the inverted pendulum.

Since 2 controllers will be designed for swing up and balancing, the system will be tested for stability at 2 fixed points. To test for stability, the polarity of the Eigen values of the system are analysed. Negative Eigen values are stable and positive are not stable. Also Eigen values that are imaginary mean that the system is oscillatory. The Eigen values of the system when the pendulum is at the top are given by

$$\det(\lambda I - A) = 0$$

Where I is the 4x4 identity matrix and λ are the Eigen values. Substituting A in the equation above gives

$$\lambda = \begin{bmatrix} -33 \\ -15.1507 \\ 14.8507 \\ 0 \end{bmatrix}$$

The system has one positive Eigen value implying that the system is unstable. Intuitively this makes sense as the pendulum does not stay upright without any control action (the pendulum angle diverges indefinitely from the fixed point). A controller that can stabilise the system has to be designed by choosing Eigen values that stabilize the pendulum system. The system was also linearized at the bottom $\theta = \pi$. This transfer function at the bottom is used for swing up control and the Eigen values are given by

$$\lambda = \begin{bmatrix} -33 \\ -0.15 + 15i \\ -0.15 - 15i \\ 0 \end{bmatrix}$$

The Eigen values show that the system is stable in the sense that it decays to zero or converges to the fixed point angle. However, some Eigen values are imaginary implying that the system is oscillatory. The damping ratio of the pendulum was found to be 0.01 and since its less than 1, the system is underdamped. The swing up controller does not require for the system to be stable but the frequency of oscillation can be utilised to achieve optimum swing up by matching that frequency to achieve resonance.

ii. Controllability and Observability of the inverted pendulum

When designing a controller, it is important to know beforehand that the system is controllable. The following is the matrix used to test for controllability.

$$P = [B \;\; AB \;\; A^2B \ldots A^{n-1}B]$$

The A and B matrices are substituted for the pendulum system. For full controllability, all the columns of matrix P must be linearly independent. That means P must be a full rank matrix. It is found that pendulum is fully controllable using the MATLAB command *rank(ctrb(A, B))* which finds the P matrix and its rank. That means all the 4 states of the inverted pendulum system can be controlled by finding a controller K. In this research, focus will be on controlling the pendulum angle for balancing in the upright position and the cart position to avoid hitting the end of the track.

The system is also tested for observability to see if states can be predicted by knowing the time history of the system output. The following is the matrix used to test for observability.

$$Q = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

If Q is full rank, the system is observable. The matrices A and C for the system are substituted into the matrix and the rank is found in MATLAB using the *rank(obsv(A,C))* command. Q is full rank for the inverted pendulum and hence the system is fully observable. That means the states can be estimated using the state estimator if they cannot be measured directly or need to be improved. A Kalman filter can also be used to get better sensor measurements and hence improving the results.

3.4.2 System specifications

In order to have a sound design of a system, specifications must be taken into account and be used as guidance throughout the design process. The pendulum system is required to swing up from its bottom stable fixed point to the top unstable fixed point. The following list are the quantitative specifications for the system. These will be checked and used to evaluate the performance of the controller designs

Swing up controller specifications

- Swing up must happen within 10s of swinging from rest
- The swing up distance must be with 0.3m since the track is only 0.45m long
- The swing up must reach the balancing angle of the pendulum with angular velocity $\dot{\theta} \approx 0$

Balancing controller specifications

- The pendulum must settle within 3s after swing up
- Angle overshoots should be within 15% of the initial value at balancing actuation
- The pendulum must have zero steady state error after settling so that it doesn't oscillate
- The system must reject small forces of disturbance around 0.5N and maintain stability
- Noise must be reduced by over 80% so that the control is smooth
- The cart must stay within the track length of 0.45m

The specifications outlined above are used when designing, simulating and implementing of the controllers. Apart from these specifications, the cost of control will be analysed using Integral Square Error.

3.4.3 Swing controller design

In this section as swing up controller that swings a pendulum from rest at 180 to upright position of 0 is designed. Several control methods are considered in this design section and the best controller in terms of meeting specifications will be implemented on the real system. The following are the methods used.

i. Bang Bang controller

This controller works by setting positions of the cart in a square waveform such that the cart accelerates back and forth swinging the pendulum. The square wave will be matched to the resonance or natural frequency of the swinging pendulum. First the cart position of the controller is designed.

Position controller

A compensator is tuned in Matlab using the transfer function of the motor (26). Using the poles of the motor transfer function, a position controller is designed. When tuning, an integrator was introduced to achieve zero steady state error of set position. A gain of 21.8 was to make the position reach the wanted level in terms of position. A zero of 2.2 is added for compensation of the system. The controller that is obtained in continuous time is

$$K(s) = \frac{21.8s + 47}{s}$$

The controller was tested in Simulink and the following block diagram is used for simulation

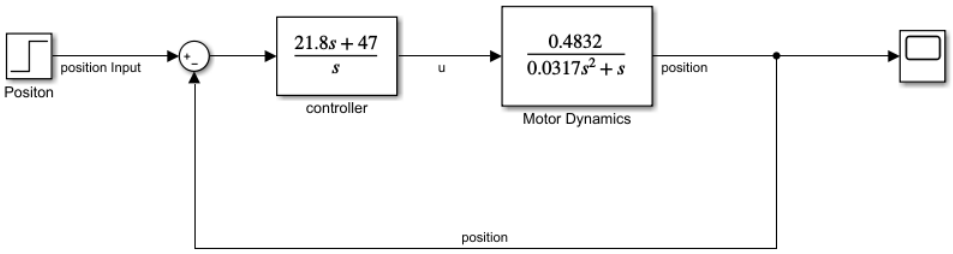

Figure 21:Simulation of a Cart Position Controller

The controller was simulated in Matlab to test for performance. The position control specification was not strict as long as it would achieve swing up control. The following is the controller response after setting reference or input position to 0.3m.

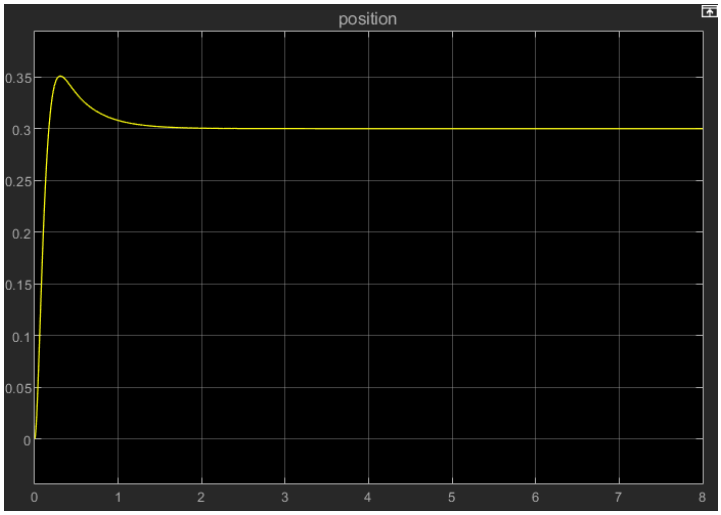

Figure 22: Response of a Simulated Cart position controller

The controller gives good settling time that is within 2s with zero steady state error. The overshoot is around 17% which is tolerable for the swing up operation. To achieve swing up a position input reference was designed as a square wave. The period of the swinging pendulum from figure 15 was found to be 0.6s. In the attempt to achieve resonance, the square wave would have a period of 0.6s. The square period is tuned still until better results for swinging up are achieved. A position amplitude is chosen arbitrarily to be 0.15m to have a peak to peak of 0.3m. This is for the cart to avoid hitting the ends of a 0.45m track. The Simulink block diagram below shows the simulation of the swing up controller

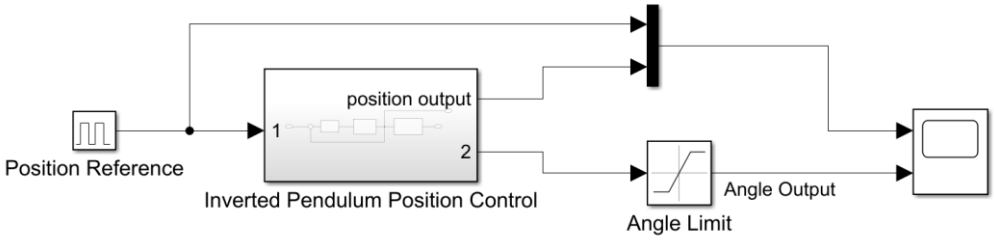

Figure 23: Simulation of a Swing up Controller of the inverted Pendulum

Figure 23 is a combination of the position controller, the motor transfer function and the pendulum angle transfer function. The angle is limited by a saturation block so that it varies from –π to π. The

position reference is a square wave block in Simulink. The following are the results obtained when the simulation is run where the peak to peak position is 0.3m at period of 0.6s.

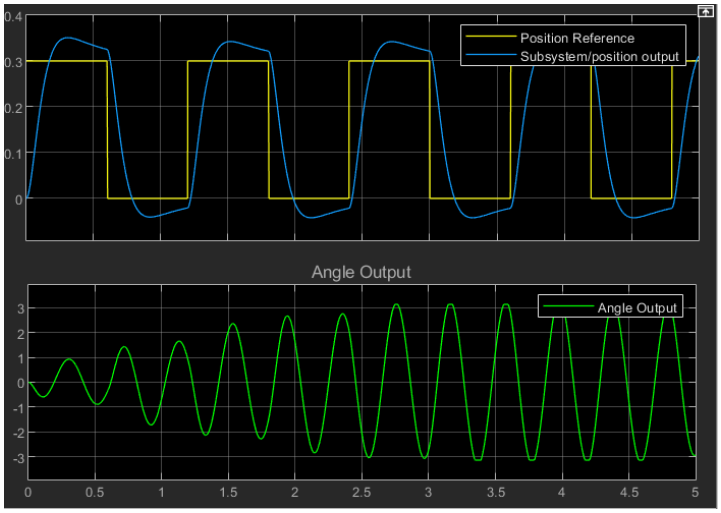

Figure 24: Simulation of Cart position (up) as compared to Reference. Simulation of Pendulum Angle Response (Bottom)

The simulation above shows position reference (Yellow) and the output position (Blue). The output tracks the input simulating the movement of the cart back and forth. A simulation of the angle of the pendulum is show at the bottom (Green). The plot shows the swinging up of a pendulum. The resting angle of the pendulum in this case is assumed to be 0 even though figure 7 shows angle 0 to be when pendulum is up. This is to simplify simulations. The angle swings up until it reaches π rads which is the assumed to be the upright position of the pendulum. This simulation shows that the pendulum is brought to the upright position in less than 4s. Introducing angle feedback would have been much more efficient. In this simulation, the pendulum continues to swing even after reaching the desired angle. However, during implementation the balancing controller will take over when the desired angle is reached. This controller is implemented on the real system and the results are shown in the results section.

3.4.4 Balancing Controller

In this section, a controller to balance the pendulum is designed. There are several methods that can be used for this design but since the inverted pendulum is a Single Input Multi Output system, the State Space controller methods are chosen. Other methods like the PID and lead/ lag compensators controllers require 2 solutions to control the cart position and the angle which can be tedious. As mentioned in the literature review section, PIDs and compensators do not produce the best results and are not optimisable. Other complex methods like the reinforcement learning, Neural networks and Fuzzy logic controller require high computational power. Two methods are chosen here to control the inverted pendulum which are, the pole placement method in state space and the Linear Quadratic Regulator

i. Pole Placement Method

This type of controller stabilizes a system by choosing K, which is a vector of gains used to stabilize the system. The following is the general set up of the state space feedback control method.

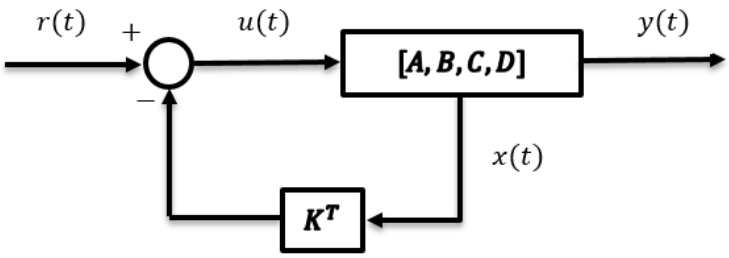

Figure 25: Block Diagram of State Space Feedback Control

The $r(t)$ is the set point or the desired output of the system. In this pendulum system, the possible set points are the output angle, position, angular velocity and cart velocity which can be given by

$$r(t) = \begin{bmatrix} x_r \\ \dot{x}_r \\ \theta_r \\ \dot{\theta}_r \end{bmatrix}$$

Output of the system is denoted by *y(t)* which represent the output angle, position, velocity and angular velocity. These are also the states of the system which are denoted by *x(t).* The signal *u(t)* is the input voltage of the system which is tuned for control. In general,

$$u(t) = ke(t) = k[r(t) - y(t)]$$

Where e*(t)* is the error and k is the controller. Using $\dot{x} = Ax + Bu \; and \; y = C^T x$, *u(t) and y(t)* are substituted to give

$$\dot{x} = Ax - BkC^T x + Bkr,$$

$$kC^T = K^T$$

The system in Laplace transform is now given by

$$\frac{y(s)}{r(s)} = \frac{C^T Adj[sI - A + BK^T]B}{|sI - A + BK^T|}$$

The above equation resembles a closed loop transfer function of the system. There exist poles s or Eigen values λ that can stabilize the cart pendulum system since it is controllable. The design process will be to equate the desired Eigen value equation which is denoted by $D_{desired}(\lambda)$ and the closed loop characteristic equation $D_c(\lambda)$ to get K values by comparing coefficients.

$$D_c(\lambda) = |\lambda I - A + BK^T| = 0$$

$$D_{desired}(\lambda) = (\lambda + p_1)(\lambda + p_2)(\lambda + p_3)(\lambda + p_4) = 0$$

The desired poles or Eigen values P = [p_1, p_2, p_3, p_4] correspond to $\chi = [x, \dot{x}, \theta, \dot{\theta}]$. The P vector has to be chosen as stable values that satisfy the system specifications. With the chosen P, the equations above are compared so that values of K which satisfies the stability given by Eigen values P can be calculated. K values give the controller *u = -Kx.* The criteria for choosing P is as follows

$$\tau = -\frac{1}{P} \leq settling \; time$$

$$\%Overshoot = 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

The symbol τ is the time constant due to the dominant pole of the system and ζ the damping ratio of the poles. In this case the damping ratio is assumed to be too small that it can be neglected. Hence real poles will be chosen and their polarity is negative to have stability. The following is the table of controller designs using this method. The K values are found in MATLAB using *K = place(A,B, P)* command.

Table 3: Eigen Values and corresponding Controller Gains K for Pole Placement Method

Controller number	Eigen values $P = [p_1, p_2, p_3, p_4]$	Controller gains $K = [k_1, k_2, k_3, k_4]$
1	[-6, -5, -7, -4]	[-0.2333, -2.240, -6.1907, -0.3718]
2	[-8, -7, -9, -6]	[-0.8400, -2.522, -8.7219, -0.5614]
3	[-20, -11, -10, -9]	[-5.50, -4.0059, -18,5050, -1,2242]

The |P| vector was chosen to be greater than 1/3 or poles faster than 3s, which is the settling time set in the specifications. This will make the controllers faster and hence archive the settling time required. These value were chosen arbitrarily close to each other with particular attention to p_3 which corresponds to the angle output. The controllers are simulated in a Simulink model to see their response. The following is the Simulink Block diagram used for simulation of the system.

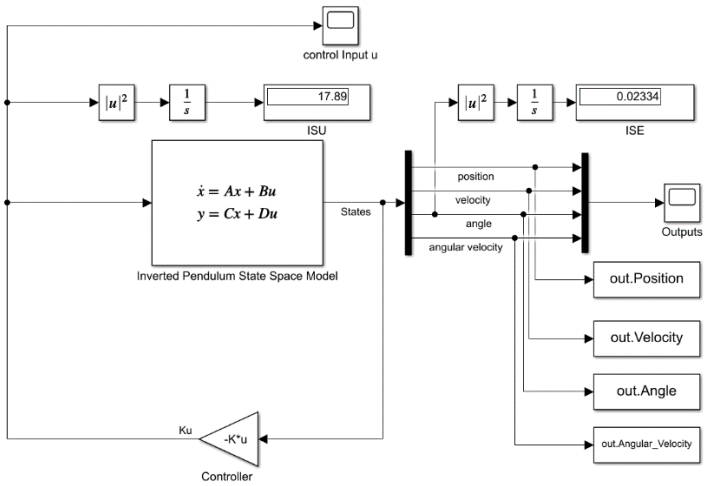

Figure 26: Block Diagram of State Space Feedback Controller in Simulink

Controllers 1, 2 and 3 are simulated in the Simulink block above. The controllers are compared to see which controller gives the best output. The poles corresponding to the angle were increased to see the effect of reducing settling time. The following is the table comparing the three controllers.

Table 4: Comparison of Controllers Designed using Pole Placement

Controller Number	Settling Time/s	Overshoot Percentage	ISE	ISU
1	2	5%	0.001407	0.8748
2	1.7	8%	0.001280	0.6849
3	1.1	10%	0.001585	0.8883

The table above shows that controller 1 has the worst settling time and best over shoot. Its Integral Square Error is relatively small as well as its cost of control ISU. This is because the placed poles are slower producing a soft controller. Controller 2 has better settling time and better overshoot. The ISE and ISU for controller 2 are also the lowest. This makes controller 2 the ideal solution since it satisfies

the control specifications at low cost of control. However, controller 3 has the highest cost of control, ISE and overshoot but has the lowest settling time. Since controller 3 satisfies specifications and cost of control is not much of an issue, it is chosen because of the settling time. Low settling time means the controller will be able to catch the pendulum quickly before the cart hits the end of track hence it is necessary to have a slightly aggressive controller. The following plot is the simulation of controller 3.

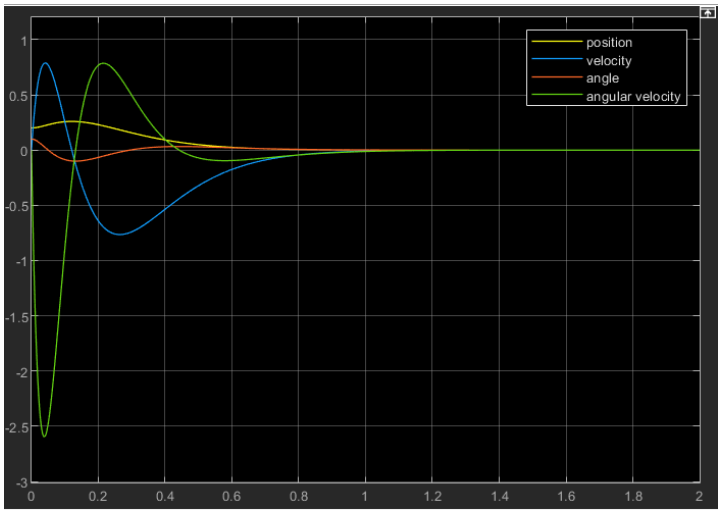

Figure 27: Simulation of Controller 3 Designed using Pole Placement

The State Space Feedback controller with the best output will be implemented in the real plant and compared with the Linear Quadratic Regulator which is designed below. The controllers above were designed in continuous time for easier analysis. To implement the controller on a microprocessor, a digital controller must be designed. The following is the redesign of controller 3.

Digital Controller using pole placement Method

The inverted pendulum is changed into discrete state space for the controller to be designed and implemented in a discrete system. The microcontroller used will sample sensors and implement control at 5ms. To discretise a system, a Matlab command *c2d(A,B,C,D, Ts)* is used where A, B, C, D are continuous state space Matrices of the pendulum system. The variable Ts is the sampling time which has been chosen to be 5ms because it is close to analogue in terms of the system response. The following matrices are the discretised inverted pendulum system

$$\begin{bmatrix} \dot{x}(k+1) \\ \ddot{x}(k+1) \\ \dot{\theta}(k+1) \\ \ddot{\theta}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.0046 & 0 & 0 \\ 0 & 0.8479 & 0 & 0 \\ 0 & 0.0016 & 1.003 & 0.005 \\ 0 & 0.6276 & 1.125 & 1.001 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ \theta(k) \\ \dot{\theta}(k) \end{bmatrix} + \begin{bmatrix} 0.00019 \\ 0.07375 \\ -0.0008 \\ -0.3043 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ \theta(k) \\ \dot{\theta}(k) \end{bmatrix}$$

Since controller 3 had the desired response, it was redesigned in digital form. To achieve similar response, the poles were changed from s domain in Laplace to z domain using the formula $z_p = e^{sT_s}$ where s is a pole or Eigen value in the continuous frequency domain. For stability the poles of a digital system lie within the unit circle of the z-plane. The poles of the system were found to be

$$P_z = [0.9048, 0.9464, 0.9512, 0.9560]$$

Obtained using the sampling time T_s and the poles of controller 3. The same process of designing a controller in continuous time was followed to find K_d gains. The gains were found to be

$$K_d = [-5.2705, -3.9382, -18.4104, -1.2177]$$

Digital Observer Design

To meet other specifications like noise cancelling, a state observer is designed so that the noise from the sensor is limited. Also other states like cart velocity and angular velocity are noisy due to taking derivatives of cart position and pendulum angle respectively. The solution is to have them derived from the observer instead of from derivative of sensor outputs. As a rule of thumb, the poles of the observer gain should be 4-10 times faster than the actual system. The poles of the observer in this case are found to be

$$P_{zo} = [0.37, 0.58, 0.61, 0.64]$$

These are the values used in the following observer equation and they are 10 times faster than those of the controller.

$$\hat{x}(k+1) = A_d\hat{x}(k) + B_d u(k) + L(y(k) - \hat{y}(k))$$
$$\hat{y}(k) = C_d\hat{x}(k) + D_d u(k)$$

The above equation is the digital state space observer representation of the pendulum system where, $\hat{x}(k+1)$ is an estimate of future states, $\hat{x}(k)$ are the estimated current states, $u(k)$ is the current input, $\hat{y}(k)$ is the estimated output and $y(k)$ is the measure actual output of the system. The L constant is the observer gain which is found by setting desired Eigen of

$$D_c(\lambda) = |\lambda I - A_d + LC_d| = 0$$

Which is compared to the desires poles P_{zo}. The L matrix was found to be

$$L = \begin{bmatrix} 0.6321 & 0.0046 & 0 & 0 \\ 0 & 0.2709 & 0 & 0 \\ 0 & 0.0016 & 0.3963 & 0.005 \\ 0 & 0.6276 & 1.1252 & 0.364 \end{bmatrix}$$

Non Linear Compensation

The system has non Linearities that were modelled the methodology section. In this section a non-linear controller is designed as an adaptive compensation algorithm. To begin with, the system has a dead band/ zone which was found to be less than 20% of the maximum input of 3V. Since the input is the small voltage which controls the larger motor voltage, the bias is applied as 0.6V to the controller output. The following is the implementation of the dead band compensation in Simulink

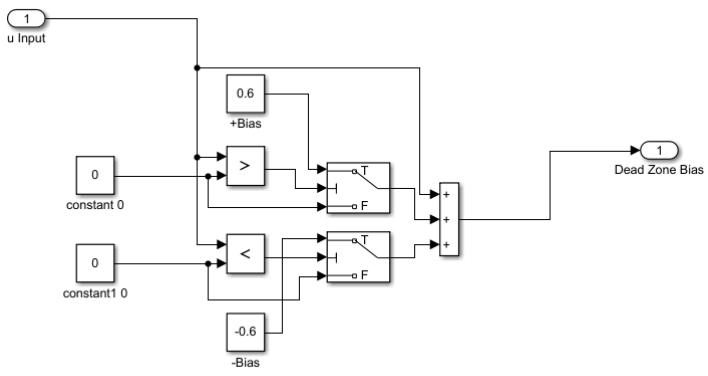

Figure 28: Implementation of Dead zone compensation in Simulink

The block above takes the output of the control voltage and checks if it is greater than or less than 0. If the voltage is greater than 0, a bias of 0.6V is added to the input and if less, a bias of -0.6V is applied. The comparators are strictly greater than or strictly less than blocks to allow the controller to have a zero value when there is no need for actuation or the actuation is 0. It is noted in the methodology section that the system also saturates at above 90% of the maximum control voltage. This was resolved by implementing a saturation of +2.7V and -2.7V. The backlash was not quantified and will be implemented using trial and error but the idea is to add input to the controller to catch up with belt slack. The polarity of the backlash depends on the cart's direction of travel.

Digital Pole Placement Controller Design simulation

The controller designed is simulated in Simulink for performance check. The following is the Simulink block used for simulation.

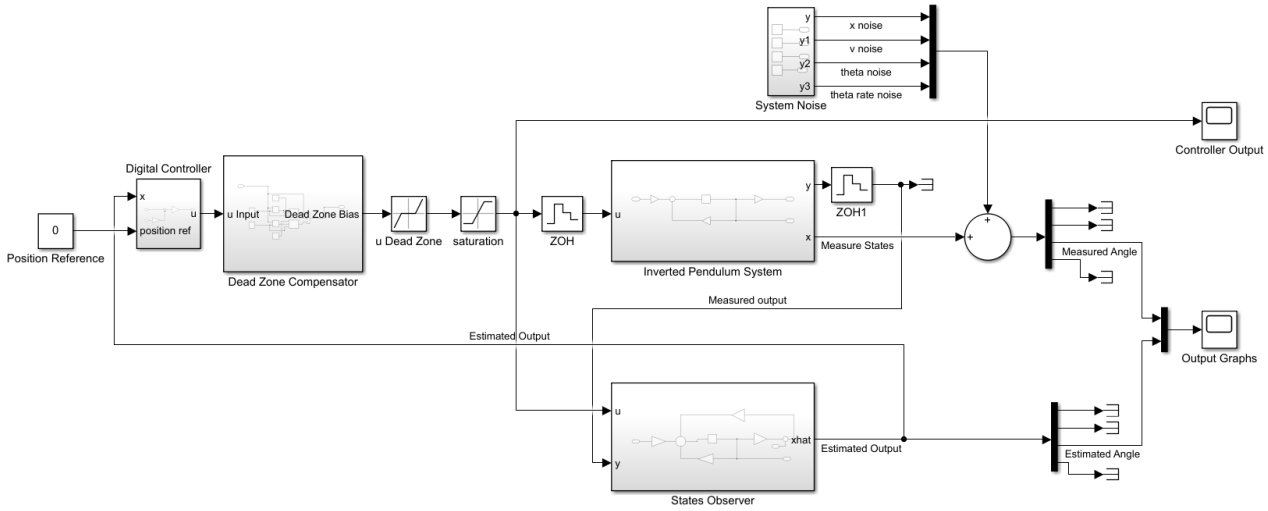

Figure 29: Simulink block Diagram for implementing a Controller for the Inverted pendulum

The system above consists of the continuous time block diagram of the inverted pendulum system. An observer in in the digital form takes the output of the real system by sampling it at 5ms using the zero order hold. The system observer also takes in the controller output (actuation signal u) so that using the observer gain L, the output can be estimated. The estimated output is used as feedback of the real system through a controller. Usually the system output or feedback is compared to the reference signal but the reference for all states are 0 for the pendulum system. The position reference can be changed but it is set to 0m so that the pendulum is balanced while the cart is at the middle of the rail. This makes sure that the cart does not hit the end of the track. The controller consists of the digital control law K_d, the dead zone compensator (also shown in Figure 28) and the saturation block. Non Linearities of the model are also incorporated which are the input u dead band, the system sensor noise and other forms of noises. A white band limited white noise block is used as the system noise is assumed to be Gaussian. This system is simulated and the output of the angle is shown below. Angle was chosen because it is the main output of the system.

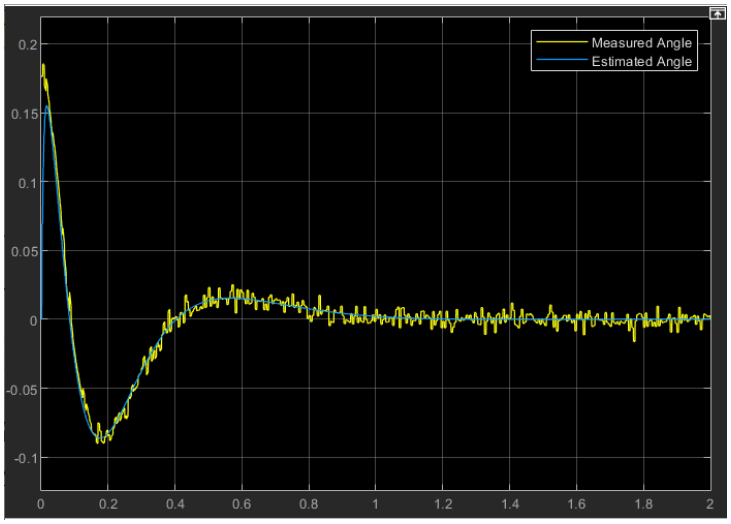

Figure 30: Simulated pendulum system showing the measured angle and the estimated angle

Figure 30 above shows that the observer quickly narrows the error between the measured angle and the estimated angle. This shows that the estimated angle is good enough to use to generate the control law. It is also not noisy and hence improves the controller result. This controller is to be compared with the LQR version that is designed below.

ii. ***Linear Quadratic Regulator Method***

The Linear Quadratic Regulator Controller is designed using the cost function in order to achieve optimum control. This is good for meeting specifications given for the system design using cheap control. The structure of an LQR controller is very similar to the pole placement method because they are all forms of linear State Space Feedback control. The objective is to design the gains vector K which stabilize the Inverted Pendulum System. To design the system, the following cost function is used

$$J(t) = \int_0^t x(\tau)^* Q x(\tau) + u(\tau)^* R u(\tau)\, d\tau$$

J denotes the cost of control overall while x and u are the controlled states and the control inputs respectively. Q is a diagonal matrix that weighs deviations from state 0 of each corresponding state. It gives penalty to undesired state deviations in order to the correct state in minimum time. The R matrix weighs the cost of actuation u. Similar to Pole Placement method, the controller takes the form u = -Kx, where ideally K is found by choosing the stabilising Eigen Values of (A-BK). From the cost function, the controller cost is optimised using the Riccati's equation given below.

$$A^*X + XA - XBR^{-1}B^*X + Q = 0$$

The Matrices of the inverted pendulum A and B are substituted in the equation. Also, the chosen states costs Q and input cost R are substituted and the equation is solved for X. The X values are used in

$$K = R^{-1}BX$$

To find K. The Riccati's equations become tedious to solve for a 4th order system like the inverted pendulum and hence a MATLAB command *K = lqr (A, B, Q, R)* is used. The following table shows the values of K for different controllers designed.

Table 5: Q Matrix vs R Matrix and the corresponding Controller Gains K for LQR

Controller number	Q Matrix $Q = [Q_{11}, Q_{22}, Q_{33}, Q_{44}]$	R Matrix	Controller gains $K = [k_1, k_2, k_3, k_4]$

4	[100, 1, 500, 1]	1	[-10.0000, -6.5427,-43.8238, -2.6795]
5	[100, 1, 200, 1]	1	[-10.0000, -6.3267, -38.6706, -2.5777]
6	[10, 1, 50, 1]	1	[-3.1623, -4.9522, -30.1148, -2.1765]

The table 5 above shows the chosen values of the Q matrix and the R matrix and their resulting controller values K. The R matrix was kept at value 1 because the cost of control was not of importance. The Q matrix was varied mostly for the angle because that is the main output of the system. Position value of the Q matrix was also varied to see the effect of changing position weight. The velocity and angular velocity states were not changed because they were not of importance in this research. The different controllers were simulated using the Simulink block diagram in figure 26 and the following are the results from different controllers

Table 6: Comparison of Controllers Designed using LQR

Controller Number	Settling Time/s	Overshoot Percentage	ISE	ISU
4	1.4	8%	0.007059	0.6993
5	1.2	10%	0.0008482	0.7224
6	1.6	5%	0.0004931	0.3591

Controller 4 is relatively fast with settling time of 1.4 and 8% overshoot. The control cost and the ISE are also good. Decreasing Q_{33} from 500 to 200 yields controller 5 as shown in the table. This had an effect of improving settling time at the cost of overshoot and cost of control. The integral square error is reduced significantly. Regardless of increase in overshoot and cost of control, controller 5 satisfies the given specifications better. Controller 6 was as a result of decreasing the penalties of position and angle to 10 and 50 respectively. This gave controller with low overshoot, ISE and ISU but slow settling time. It is observed that a high Q_{33} of 500 yields a slow controller and also a low Q_{33} of 50 produced a slow controller. A moderate value of Q_{33} produced faster controller and hence controller 5 is chosen for implementation. The following is a simulation of controller 5 in Simulink.

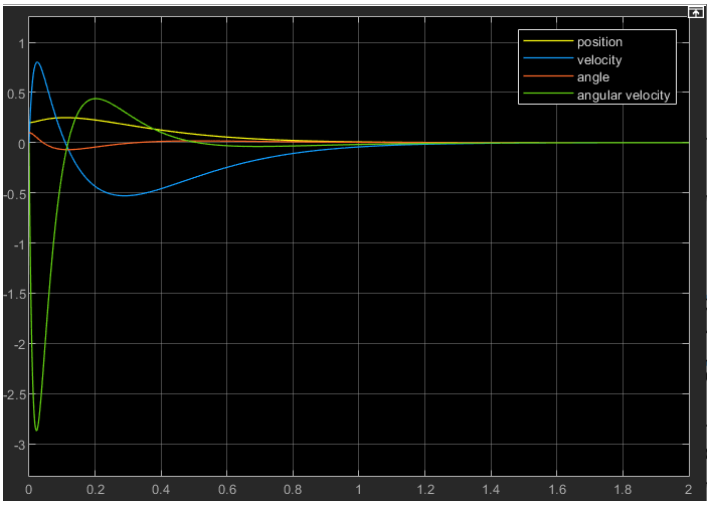

Figure 31: Simulation of Controller 5 Designed using LQR

The controllers above were tested on the real system but the values for angular velocity and cart velocity were noisy causing some oscillations. Also the nonlinearities like dead band, saturation and backlash were not taken into account when the controllers were designed and simulated. The solution to these problems is designing of an LQR controller in conjunction with a Kalman Filter while incorporating

nonlinear compensation. This is simply known as the Linear Quadratic Gaussian controller (LQG). The LQG solves the noise issue by doing state estimation and also noise suppression (filtering) and disturbance rejection. The other non-linear issues are solved by using compensation. In the next section and LQG and Nonlinear compensation controllers are designed.

iii. Linear Quadratic Gaussian and Non-Linearity compensation controller

In this section, an LQR is combined with a Kalman Filter to improve the performance of the controller. Nonlinear models will also be used to build non-linear compensation. The designed controller will be tested and the improvements will be assessed.

Kalman Filter Design

The LQR control method relies on full state measurements for optimum control. In in the inverted pendulum system, angular position of the pendulum and cart position can be measured directly. The angular velocity and the cart velocity are derived from the angle and position respectively. This introduced noise to the signal hence compromising the quality of control. Despite that, the angle sensor (potentiometer) and the quadrature encoder are noisy as well. These issues defeat the purpose of using a Linear Quadratic regulator which is supposed to give optimum control. Apart from that, there is need for disturbance rejection so that the controller is robust. These issues are solved using a Kalman filter. The main objective of designing a Kalman Filter is to optimise measurement noise attenuation, disturbance rejection and model uncertainties. Therefore, the system can be modelled as follows in digital state space,

$$\hat{x}(k+1) = A_d\hat{x}(k) + B_d u(k) + d$$
$$\hat{y}(k) = C_d x + Du + n$$

Where d is disturbance and n is noise of the sensor data. It is assumed that the sensor noise is a Gaussian signal with zero mean as previously stated. That implies that the mean of the sensor data does not cause any bias to the actual state value. The noise and disturbance parameters are modelled statistically using the following equation.

$$\mathbb{E}(d(k).d(k)^*) = V_d\delta(k-\tau)$$
$$\mathbb{E}(n(k).n(k)^*) = V_n\delta(k-\tau)$$

Where $\mathbb{E}$ is the expectation function, V gives the covariance of each corresponding parameter and δ is the Dirac Delta function. With the noise model, the estimate $\hat{x}$ for states x can be obtained from the measurements of the input u and the output y. Therefore, the estimated model of the inverted pendulum is now given by,

$$\hat{x}(k+1) = A_d\hat{x}(k) + B_d u(k) + K_f(y(k) - \hat{y}(k))$$
$$\hat{y}(k) = C_d\hat{x}(k) + D_d u(k)$$

This is a full state space observer feedback control where K_f is the Kalman filter gain. The gain

$$K_f = YC_d{}^*V_n$$

Similar to the LQR, Y is the solution of the following Riccati equation

$$YA_d{}^* + A_dY - YC_d{}^*V_n^{-1}C_dY + V_d = 0$$

The goal is to minimise the cost function

$$J = \lim_{k\to\infty}\mathbb{E}[\big(x(k) - \hat{x}(k)\big)^*(x(k) - \hat{x}(k))]$$

The data obtained in the system identification is used to find the covariance's of the angle, angular velocity, cart position and velocity. To find the angle variance, data was logged while the pendulum was kept at a constants angle. The standard deviation of the angle was calculated in excel using the standard deviation function. Variance found by squaring the standard deviation. The position variance is calculated the same way. For the velocity the cart accelerates until constant speed is reached. Variance is calculated the same way in the range where the speed is constant. It was difficult to measure the

variance of angular velocity and hence the value was estimated and tuned until it matches what would be the real value. For these signals, their noise variance was assumed to be uncorrelated and hence the covariance matrix is diagonal. The following is the table of variances of the states sampled at 5ms. They are shown in terms of noise power which is used in Simulink white noise block. It is noted that the variances for velocity and angular velocity are high because their noise is amplified when getting derivatives.

Table 7: Showing the Noise and Disturbance variance of the states

	Position x	Velocity v	Angle θ	Angular velocity
Noise variance	0.00001505	0.00508055	0.0000161	0.001
Disturbance	0.01	0.01	0.01	0.01

Therefore, the variance matrices are as follows

$$V_n = \begin{bmatrix} 0.000015 & 0 & 0 & 0 \\ 0 & 0.00508055 & 0 & 0 \\ 0 & 0 & 0.0000161 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$$

And

$$V_d = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

The V_d was chosen arbitrarily so that the system can handle some disturbances. Having that allows for disturbance rejection which is an improvement on controller robustness. These values are substituted in the Riccati's equation in order to calculate the Kalman Filter gain. The following is the Kalman Filter gain matrix obtained from the system.

$$K_f = \begin{bmatrix} 0.999 & 0 & 0 & 0 \\ 0 & 0.9532 & 0 & 0.0012 \\ 0 & 0 & 0.998 & 0 \\ 0 & 0.0002 & 0 & 0.99 \end{bmatrix}$$

The Kalman Filter gain give stability in system estimation if the Eigen values of $(A_d - K_f C_d)$ are within a unit circle for digital systems. In this case Eigen values are found to be $E = [-0.0857, -0.0857, 0.08, 0]$. These values yield a stable estimator since they satisfy the Eigen value requirements. This Kalman Filter is combined with a digital LQR controller to make an LQG. Also the nonlinearities that were incorporated in digital pole placement design were included in this LQG design. The controller is simulated in Simulink using the block in figure 29 except the observer block is replaced by a Kalman filter gain. Similar to the pole placement method, the LQR controller that was designed in continuous time was changed to discrete time so that it can be implemented on a microcontroller or microprocessor. The Q and R matrix remain the same but they are applied to the digital state space obtained in digital pole placement controller design section. The digital LQR gain Kd is now as follows

$$K_d = [-8.4254, -5.6779, -33.9159, -2.2590]$$

With these parameters the simulation of the inverted pendulum yields the following result.

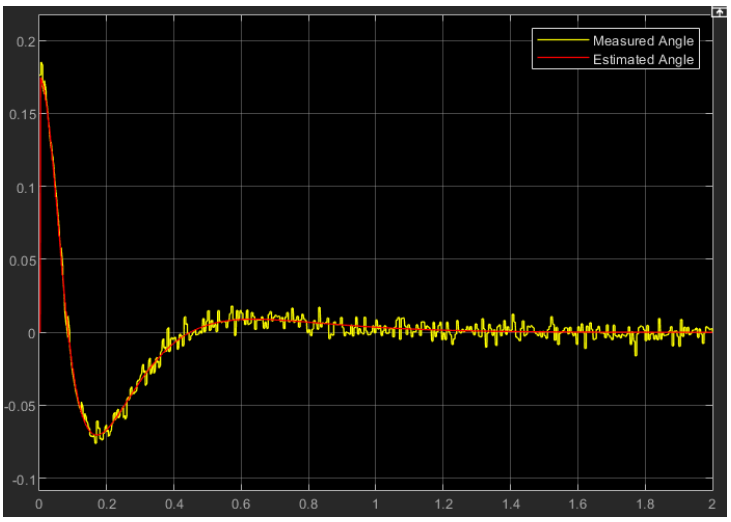

Figure 32: Simulation of the LQG controller for the Inverted pendulum System

The plot shows that the digital controller behaves similar to the continuous version, satisfying the requirements. The estimator done by the Kalman filter shows that it quickly matches the measured value but with good noise reduction. Hence this controller will be further simulated and compared with the digital pole placement controller with an observer. The controllers will be tested for settling time, overshoot, disturbance rejection, noise reduction, control cost and steady state error.

4. Results

In this section the two controllers that were designed in the previous section will be simulated and compared. The best controller in terms of meeting the control requirements will be implemented. On implementation, the swing up controller will be used to bring the pendulum to the balancing angle where the simple switching logic will be applied in software. In order to test the controllers, the simulating block diagram is modified in Simulink as follows.

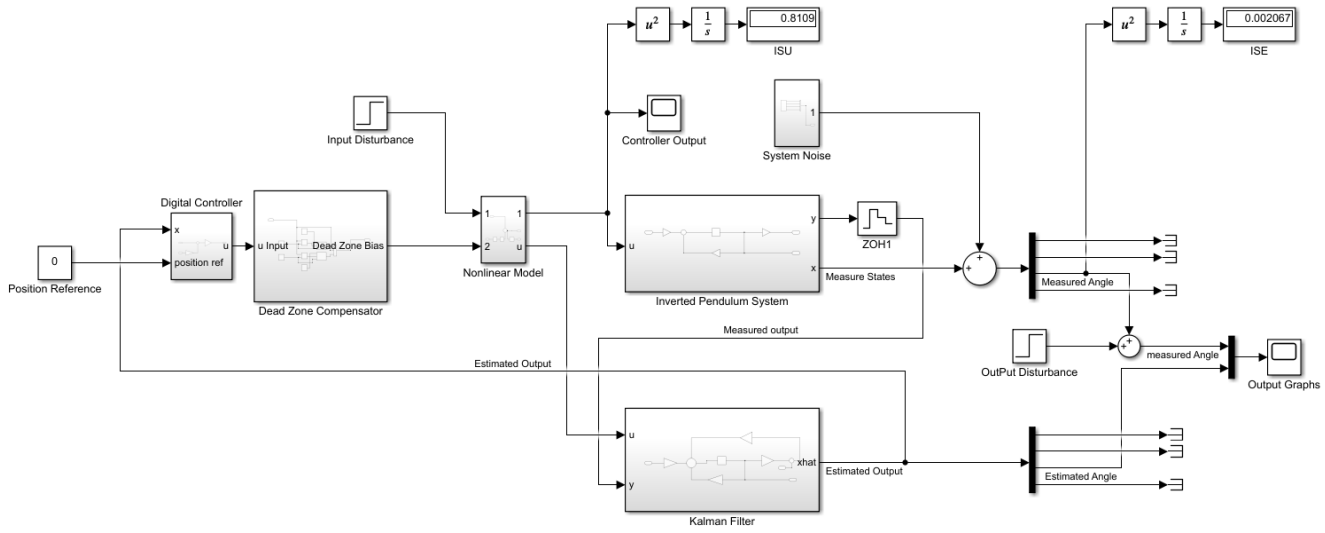

Figure 33: Simulink Block to test controllers for comparison

The block diagram in figure 29 is modified so that controllers can be tested and evaluated. In addition to its previous blocks, the input and output disturbances are added. These are used to test for disturbance rejection of each controller. The disturbance rejection test will be evaluated by comparing the overshoot and the settling time of the angle of the inverted pendulum. The ISU and ISE are added in the model so that the amount of control and steady state error can be measured respectively. These values will be used to evaluate controllers designed.

4.1 Disturbance rejection Test

In this section, the two chosen controller which are the digital pole placement controller with an observer and a digital LGQ are tested to see how well they handle input and output disturbance. The disturbances are introduced at 2s into the simulation of each controller for uniform comparison. The plots below show the input disturbance and two different responses to input disturbance.

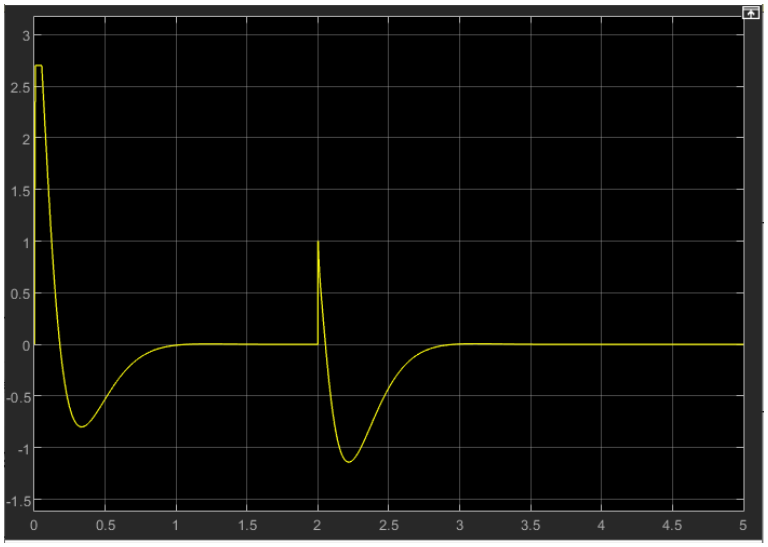

Figure 34: The Input Disturbance Introduced into the system

The plot above shows an external voltage to the input of the system to introduce input disturbance. This disturbance is applied to both controllers which are the digital state space feedback controller and the LQG controller. The results of each are shown in the following figures.

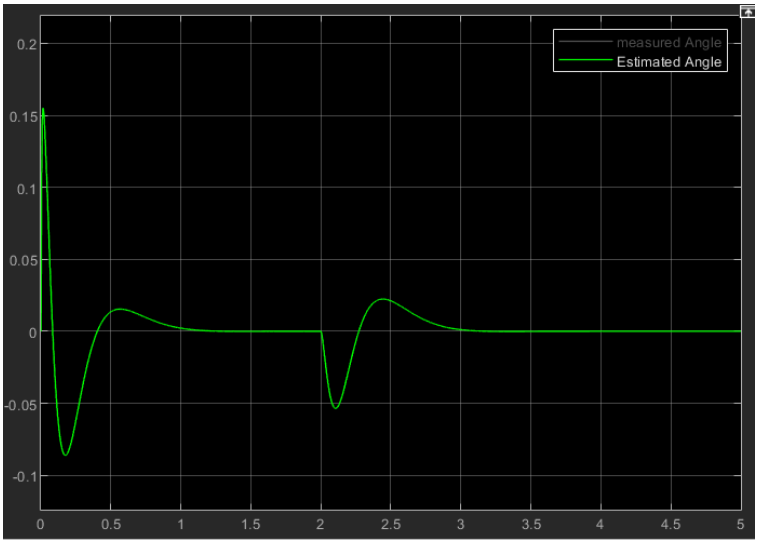

Figure 35: Simulation of Pole Placement Digital Controller to Test for Input Disturbance Rejection

This plot shows the response of the pendulum system to the input of 1V. And undershoot of about -0.058rads and an overshoot of 0.025rads were obtained. The settling time obtained is 1s.

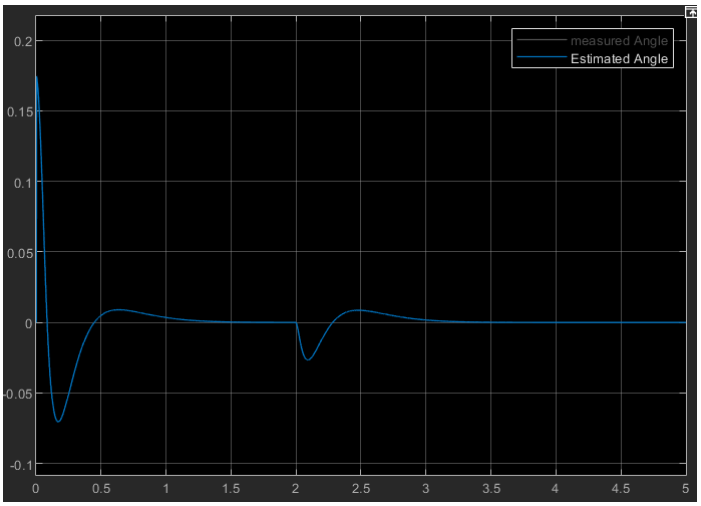

Figure 36: Simulation of LQG Controller to Test for Input Disturbance Rejection

This plot shows the simulation of the LQR controller. The pendulum undergoes an undershoot of -0.025rads and an overshoot of 0.01rads. The settling time after disturbance is 1s. The plots above show that the LQR controller is much more resistant to the input disturbance than a state space feedback controller. The controllers are further tested for noise rejection. The following plots show the noise rejection capabilities of the filter or observer part of the controller.

4.2 Noise Reduction Test

For the controllers to be efficient, all the states have got to be measured and used in control. In this system only position and angle can be measured directly. In order to fulfil full state measurement and control of a system, all states must be measured or estimated. The two different estimators and filters will be compared here. The following are the plots of the state space feedback and the Kalman Filter estimation of the pendulum angle.

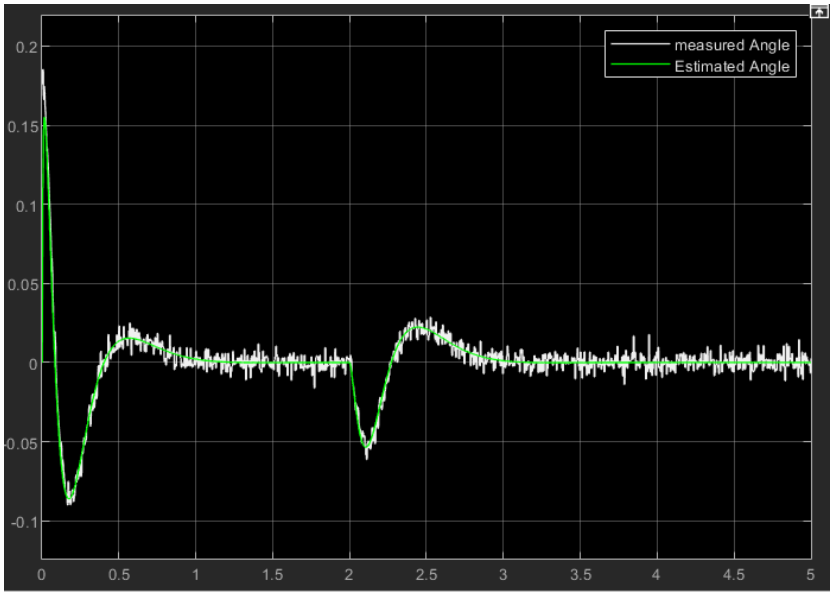

Figure 37: Simulation of measured angle and estimated angle of State Space controller and Observer

The state space observer used to estimate the measurements of the angle was made to have poles 10 times faster than the controller. This enables it to estimate the angle precisely which can be used as feedback. This improves provided the estimate is close enough to the real value. Since angle can be measured the above signals show that the measured angle is closely matched with the estimated.

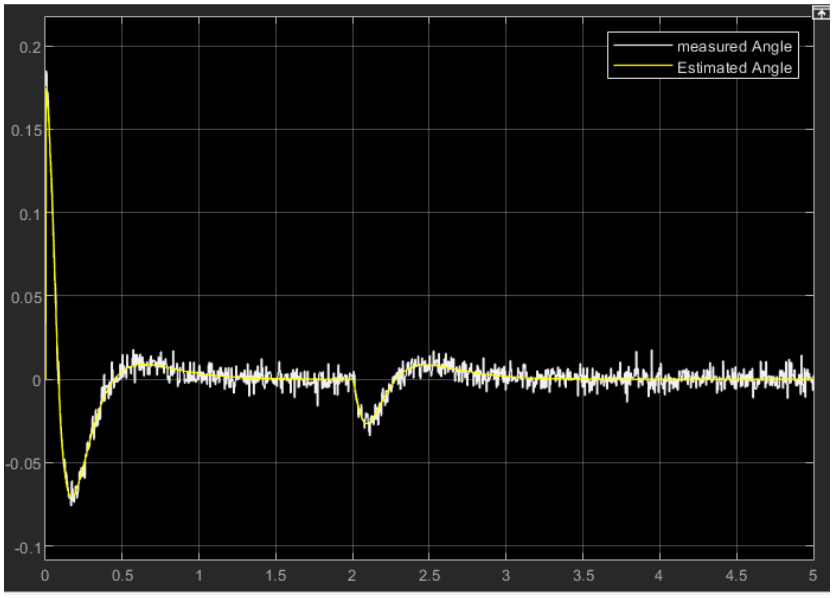

Figure 38: Simulation of LGQ controller

The plot above also shows the estimated pendulum angle and the measured angle. The estimate is done by a Kalman filter in an LQG controller. It shows that the Kalman filter also estimates the measured angle very closely like the observer. In comparison however, the graphs show that the Kalman filter converges with the measured angle slightly faster than the observer. Hence the Kalman filter is better filtering and estimating of the unknown signals like the angular velocity.

4.3 Results of controllers and meeting of specifications

The controllers are further tested to see which controller best meets the specifications of the system. In the table below, the controller overshoots are compared. The controllers are also assessed for settling time to see which controller brings the pendulum back quickly enough before it falls. In addition, the cost of control is also recorded to see which controller does more actuation for the same task. This is represented by ISU which is the cumulative square of the control value. The steady state error of the pendulum is also recorded as the integral squared error.

Table 8: Results of the two Types of Controllers Designed (LQG and State Space Feedback)

Controller Type	Overshoot %	Settling Time /s	ISU	ISE
State Space feedback	12%	1s	1.136	0.002911
LQR	10%	1s	0.7933	0.002146

In the table it is show shown that the state space feedback controller has overshoot of 12% as compared to the LQG with 10%. This is because the LQG penalises high deviation of states and outputs. The 2 controllers achieve the same settling time which is favourable. The measured control cost shows that the state space feedback controller uses more actuation to keep the pendulum balancing. Although the penalty for cost of control in the LQG controller was R = 1, it managed to still balance the pendulum at a lower cost than the state space feedback controller. Also, the ISE of the LQG was lower than that of the state space feedback controller. This is still because of the optimization ability using the cost function of the LQG. Both controllers meet the required specifications which are the following:

- The controllers settle within 3s

- The over and undershoots are within 15% of the final value
- The pendulum angle is maintained steady to achieve zero steady state error.
- Noise was reduced so that there is close to zero variance in the estimated signal and hence noise was reduced.
- Other Non Linearities of the system like the dead band were mitigated

Although they all satisfy the system requirements, the LQG was chosen for implementation in the real system for its overall better performance.

It was also required that the pendulum cart stays within the rail length of 0.45m. The following simulation shows that when the pendulum is balanced, the cart position stays the same. Also when disturbance was applied, the cart moves to balance the angle but stays at the new position hence as long as the disturbance is reasonable, the cart will a ways be in the wanted region.

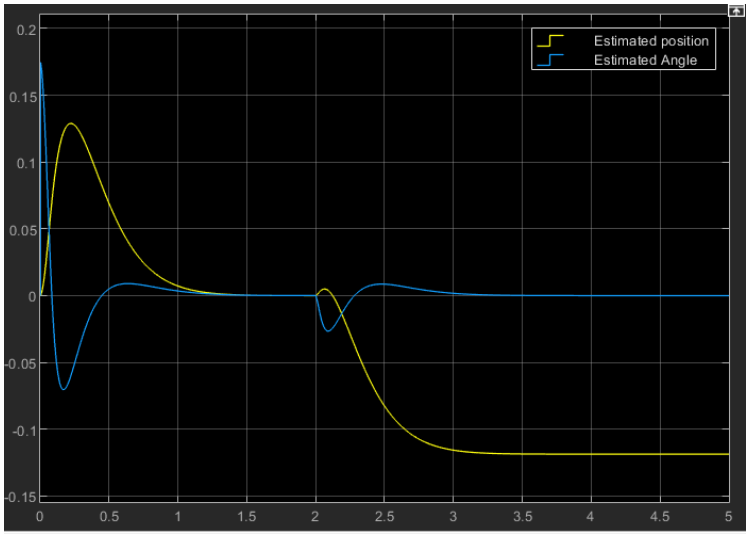

Figure 39: Simulation of Position and Angle of the Pendulum

The swing up controller was shown in the swing up controller design section and it satisfies its requirements. In the following section, the swing up controller is combined with the LGQ balancing controller and implemented on the actual system.

4.4 Controller Implementation

The chosen controller which is the LQG is implemented in the STM32F4 discovery board. It is working in conjunction with the swing up controller. The swing up controller begins swinging the pendulum from rest until it is about the balancing region which is within 0.1 rads. From there, the balancing controller takes over to balance the pendulum. The following is the code of the swing up controller in the timer handler.

```
void TIM4_IRQHandler(void){
  TIM4->SR &= ~(0x1);                                    // clear interrupt


// Implemetation of the swing up controller
  if(swingup == 1){
    if(count < 120){      // when 120 it means it stays at position until 120*5ms and then changes position
      xRef = -0.06;
      count = count+1;
    }
    else{
      xRef = 0.08;        // Positive setpoint of the cart position
      count = count + 1;
      if(count>240){
        count = 0;
      }
    }
  }
```

Figure 40: Implementation of the Swing up Controller

This is archived as shown by setting the variable xRef which was made to change position after 120x5ms. This was implemented in the timer handler of the controller. In order to control the reference position, a position controller was designed. The following controller is used in this design

$$K(s) = \frac{21.8s + 47}{s}$$

This controller was changed to discrete time for implementation in the microcontroller. The controller in z-transform at 5ms sampling time is

$$K(z) = \frac{21.8z - 21.57}{z - 1}$$

To implement, K(z) is changed to a difference equation and implemented in a software as follows

```
    // Position controller
    uCurrent = uCurrent + 21.8*xError - 21.793*xEprevious;

    if(uCurrent > 2.7){

      uCurrent = 2.7;
    }
    if(uCurrent < -2.7){
      uCurrent = -2.7;
    }

    pwmValue = (int)(100*uCurrent/3);
    if(uCurrent > 0){
      GPIOD->BSRR |= GPIO_BSRR_BR3;
      GPIOD->BSRR |= GPIO_BSRR_BS2;
      TIM3->CCR2 = pwmValue;
    }
    else if(uCurrent < 0){
      GPIOD->BSRR |= GPIO_BSRR_BS3;
      GPIOD->BSRR |= GPIO_BSRR_BR2;
      TIM3->CCR2 = pwmValue*-1;
    }
    else{
      GPIOD->BSRR |= GPIO_BSRR_BR3;
      GPIOD->BSRR |= GPIO_BSRR_BR2;
      TIM3->CCR2 = 0;
    }

    uPrevious = uCurrent;
    xEprevious = xError;

    //uCurrent = 0;
```

Figure 41: Implementation of a Position controller for Swing up

This controller is run in software and the following results are obtained.

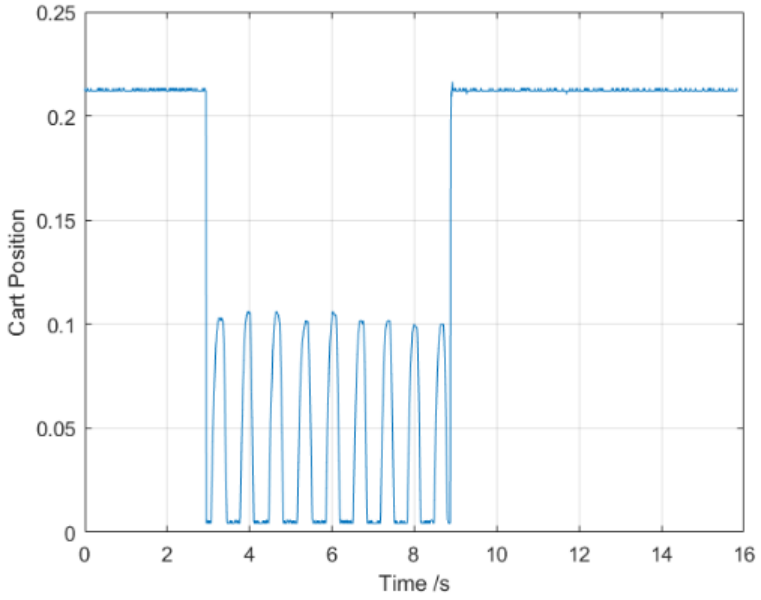

Figure 42: Position control moving the cart back and forth

The graph above shows that the cart moves back and forth. However, the plot clipped the values of position when it's at position zero but nonetheless the controller brought the pendulum up. The plot shows that the cart moves back and forth and then moves to around 0.2m this the point at which the pendulum would be balanced. The switching logic was applied so that the swing up controller hands over the control to the balancing controller and is as follows.

```
// Setting swing up to 0 if the angle reaches a point the pndulum can balance
    theta = angleFn(ADCvalue);                          // the pendulum angle*/
    if((theta<0.1) && (theta > -0.1)){
      swingup = 0;
    }
    else{
      swingup = 1;
    }
```

Figure 43: Implementation of a switching controller

The code above checks if the angle is within 0.1 rads which is the region the balancing controller can take over. The balancing controller is also implemented below together with the Kalman Filter. First the Kalman filter is initialised as follows in the code.

```
// filter initialisations
// estimated values
double x_est;
double v_est;
double theta_est;
double omega_est;
// previous estimates
double xp_est = 0;
double vp_est = 0;
double thetap_est = 0;
double omegap_est = 0;

// A matrix
double a11 = 1;      double a12 = 0.0046;   double a13 = 0;      double a14 = 0;
double a21 = 0;      double a22 = 0.8479;   double a23 = 0;      double a24 = 0;
double a31 = 0;      double a32 = 0.0016;   double a33 = 1.003;  double a34 = 0.005;
double a41 = 0;      double a42 = 0.6276;   double a43 = 1.125;  double a44 = 1.001;

// B Matrix
double b1 = 0.0002;
double b2 = 0.07375;
double b3 = -0.00078;
double b4 = -0.3043;

// Kalman Filter Gains
double k11 = 0.9095;     double k12 = 0.0046;      double k13 = 0;      double k14 = 0;
double k21 = 0;          double k22 = 0.7532;      double k23 = 0;      double k24 = 0;
double k31 = 0;          double k32 = 0.0016;      double k33 = 0.9077; double k34 = 0.005;
double k41 = 0;          double k42 = 0.6276;      double k43 = 1.1252; double k44 = 0.057;
```

Figure 44: Initialisation of the Kalman Filter

The values of the estimated states will be derived from the Kalman filter initialised. The following is the code of the balancing controller.

```
    // Controller u
    u = 1*(k1*(x_est) + k2*v_est + k3*(theta_est+1.78829) + k4*omega_est);

// implementing Saturation
    if(u > 2.8){
      u = 2.8;
    }
    else if (u < -2.8){
      u = -2.8;
    }
    // setting PWM control value
    pwmValue = (int)(fabs(u)*100/3);

    if(u > 0){
      GPIOD->BSRR |= GPIO_BSRR_BR3;
      GPIOD->BSRR |= GPIO_BSRR_BS2;
      TIM3->CCR2 = 5 + pwmValue;        // add 5 to implement dead band
    }
    else if(u < 0){
      GPIOD->BSRR |= GPIO_BSRR_BS3;
      GPIOD->BSRR |= GPIO_BSRR_BR2;
      TIM3->CCR2 = 5 + pwmValue;        // will add 5 to compensated dead band
    }
    else{
      GPIOD->BSRR |= GPIO_BSRR_BR3;
      GPIOD->BSRR |= GPIO_BSRR_BR2;
      TIM3->CCR2 = 0;
    }

    // Implement Filter here
    x_est     = (a11-k11)*xp_est + (a12-k12)*vp_est + (a13-k13)*thetap_est + (a14-k14)*omegap_est + k11*x + k12*v + k13*theta + k14*angv + b1*u;
    v_est     = (a21-k21)*xp_est + (a22-k22)*vp_est + (a23-k23)*thetap_est + (a24-k24)*omegap_est + k21*x + k22*v + k23*theta + k24*angv + b2*u;
    theta_est = (a31-k31)*xp_est + (a32-k32)*vp_est + (a33-k33)*thetap_est + (a34-k34)*omegap_est + k31*x + k32*v + k33*theta + k34*angv + b3*u;
    omega_est = (a41-k41)*xp_est + (a42-k42)*vp_est + (a43-k43)*thetap_est + (a44-k44)*omegap_est + k41*x + k42*v + k43*theta + k44*angv + b4*u;
```

Figure 45: Controller code for LQG balancing controller

The Kalman filter gains are denoted by k_{ij} while the LQG gain is denoted by k_x. The values obtained in the design are substituted in the controller software. It was noticed in the implementation that the Kalman filter worked well within the linear region of the controller. Deviating from the region slightly caused problems with balancing especially due to the uncertainty of angular velocity. It was decided that the Kalman filter works only within small angles about the balancing point of 0 rads. The control law u was saturated so that it does not go beyond 2.8V either way. Also, the dead band was mitigated by adding 5% duty cycle to the control voltage. This was through observation that the earlier stated dead band of 20% duty cycle overcompensated the system and hence cause destabilisation. Using this code, the software was run to swing up and balance the pendulum. The following is the result of the swinging and balancing pendulum.

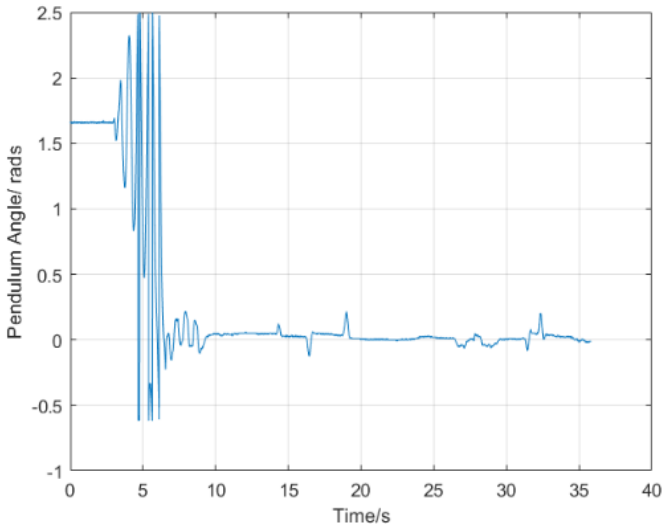

Figure 46: The Angle of the pendulum during swing up and balancing

The plot shows the real time swinging and balancing of the inverted pendulum. The pendulum swung until the pendulum reached around zero radians and the LGQ controller took over. The controller result was satisfactory and more graphs are shown in the appendix which were obtained from the actual system

5. Discussion

The results of the LQG controller above together with the swing up controller worked as expected. The literature shows that an LQR is good for Multi Input Multi Output system. It also is good at minimising controller output value and that is proven form results when the LQG was compared to the State Space Feedback controller. However, from time to time the position of the cart changed which was undesirable as it would still hit the end of the track. This problem was later resolved by having the set point of the angle to always be learning towards the centre of the track.

6. Conclusions

In conclusion the project worked well except that not all the results were obtained. The controllers were not tested for robustness as per design. However, the main objective which is to swing up and balance an inverted was archived. The Kalman filter set out to improve the sensor measurements was not implemented efficiently in the code and hence was partially working. Overall the project could be improved as shall be mentioned in the recommendations section.

7. Recommendations

Although the project worked and satisfied the initial requirements, it can be improved in a lot of areas. The project can be improved in system identification to achieve more accurate models. A system identification tool in Matlab can be used to model the system. For better control of the inverted pendulum, methods like fuzzy control, reinforcement learning can be considered since they are much more robust. That also means higher processing power to support these maybe required. Unscented and extended Kalman Filters can be applied so that estimations can happen even in non-linearized regions of the system

8. References

[1] B. X. Y. Xin and H. Xin, "The inverted-pendulum model with considerationof pendulum resistanceand its LQR controller," 2011.

[2] M. E. E.-D. Anwar, "Inverted Pendulum".

[3] S. Brock, "SWING-UP METHODS FOR INVERTED PENDULUM," in *INTERNATIONAL CONFERENCE ON ELECTRICAL DRIVES AND POWER ELECTRONICS*, Slovakia, 2003.

[4] F. PEKER and I. KAYA, "Identification and real time control of an inverted pendulum using PI-PD controller," in *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*, 2017.

[5] N. S. Nise, "Time Response," in *Control Systems Engineering*, 2015, pp. 157-209.

[6] A.-A. D. Papadopoulos and A. T. Alexandidis, "Unified Swing Up and Upright Position Stabilizing Controllers for Inverted-pendulum on a Cart," in *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, San Francisco, 2016.

[7] L. Hong-yu and F. Jian, "An inverted pendulum fuzzy controller design and simulation," in *International Symposium on Computer, Consumer and Control*, 2014.

[8] G. Belascuen and N. Aguilar, "Design, Modeling and Control of a Reaction Wheel Balanced Inverted Pendulum," in *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, de Buenos Aires, 2018.

[9] A. Türkmen, Y. Korkut, . M. Erdem, Ö. Gönül and . V. Sezer, "Design, Implementation and Control of Dual Axis Self Balancing Inverted Pendulum using Reaction Wheels," Istanbul.

[10] M. S. Tsoeu, EEE3094S - Control Systems Engineering, Cape Town, 2018.

[11] "Linear Control Theory," in *Data Driven Science & Engineering: Machine Learning, Dynamical Systems, and Control*, Seatle, 2017, pp. 323-361.

[12] G. V. Troshina and A. A. Voevoda, "Active Identification of the Inverted Pendulum Control System," in *2015 XVIII International Conference on Soft Computing and Measurements (SCM)*, 2015.

[13] . J.-J. WANG, "Position and speed tracking control of inverted pendulum based on double PID controllers," in *Proceedings of the 34th Chinese Control Conference*, Hangzhou, 2015.

[14] K. Lundberg, "The Inverted Pendulum System," 1994.

[15] Y. Chao, L. Yongxin and W. Linglin, "Design of Reinforcement Learning Algorithm for Single Inverted Pendulum Swing Control".

[16] G. Zhai, S. Okuno, J. Imae and T. Kobayashi, "Swinging up and Stabilizing an Inverted Pendulum by Switching Control," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation*, Harbin, 2007.

[17] K. Yoshida, "Swing-up Control of an Inverted Pendulum by Energy-Based Methods," in *Proceedings of the American Control Conference*, 1999.

[18] J. Schmidt, "Pendulum Control Project," 2014.

[19] K. Ioannis and L. Moysis, "Inverted Pendulum: A system with innumerable applications," in *9th International Week Dedicated to Maths*, Thessaloniki, 2017.

[20] I. Kafetzis and L. Moysis, "Inverted Pendulum: A system with innumerable applications," in *9th International Week Dedicated to Maths*, Thessaloniki, 2017.

[21] K. Y. Chou and Y. P. Chen, “Energy Based Swing-up Controller Design using Phase Plane Method For Rotary Inverted Pendulum,” in *2014 13th International Conference on Control, Automation, Robotics & Vision*, Marina Bay Sands, 2014.

[22] L. Shuang and F. Jian, “Linear Quadratic Optimal Controller Design an Inverted Pendulum,” in *2014 International Symposium on Computer, Consumer and Control*, 2014.

[23] N. Muskinja and . B. Tovornik, “Swinging Up and Stabilization of a Real Inverted Pendulum,” *TRANSACTIONS ON INDUSTRIAL ELECTRONICS,* vol. 23, no. 2, 2006.

[24] M. S. Tsoeu, “EEE4118F - Model Predictive Control (MPC),” Cape Town, 2010, pp. 101-110.

9. Appendices

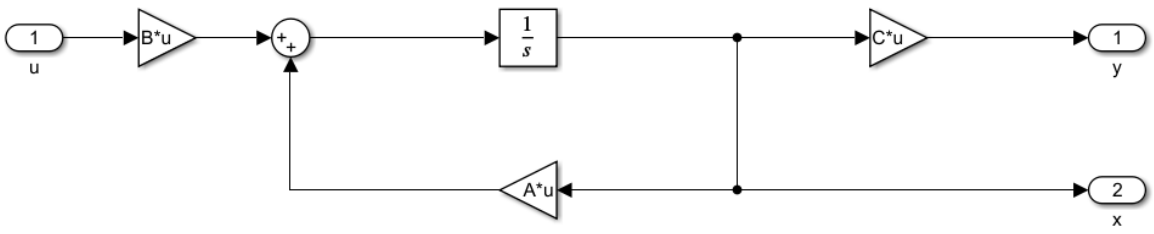

Figure 47: Continuous Time State Space implementation in Simulink

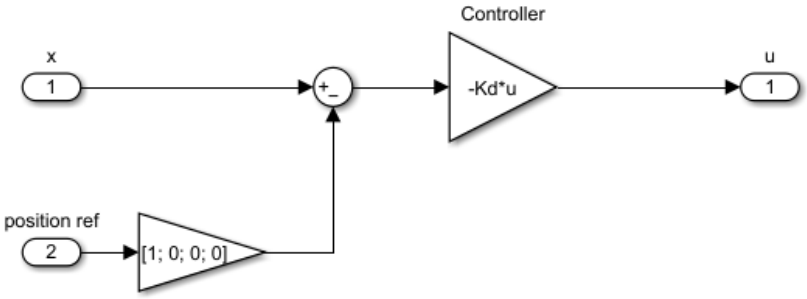

Figure 48: Implementation of a digital LQR controller in Simulink

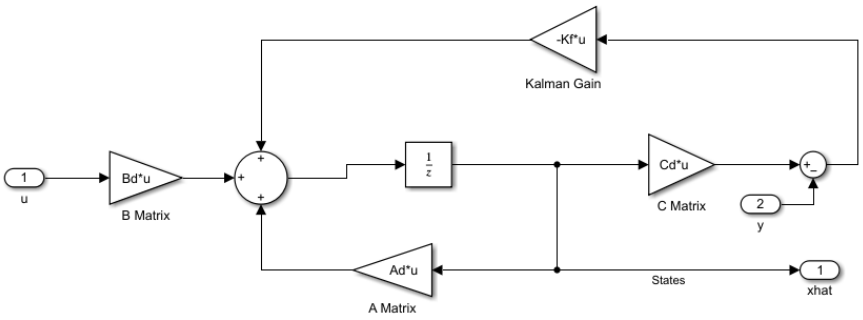

Figure 49: Implementation of Kalman Filter in Simulink

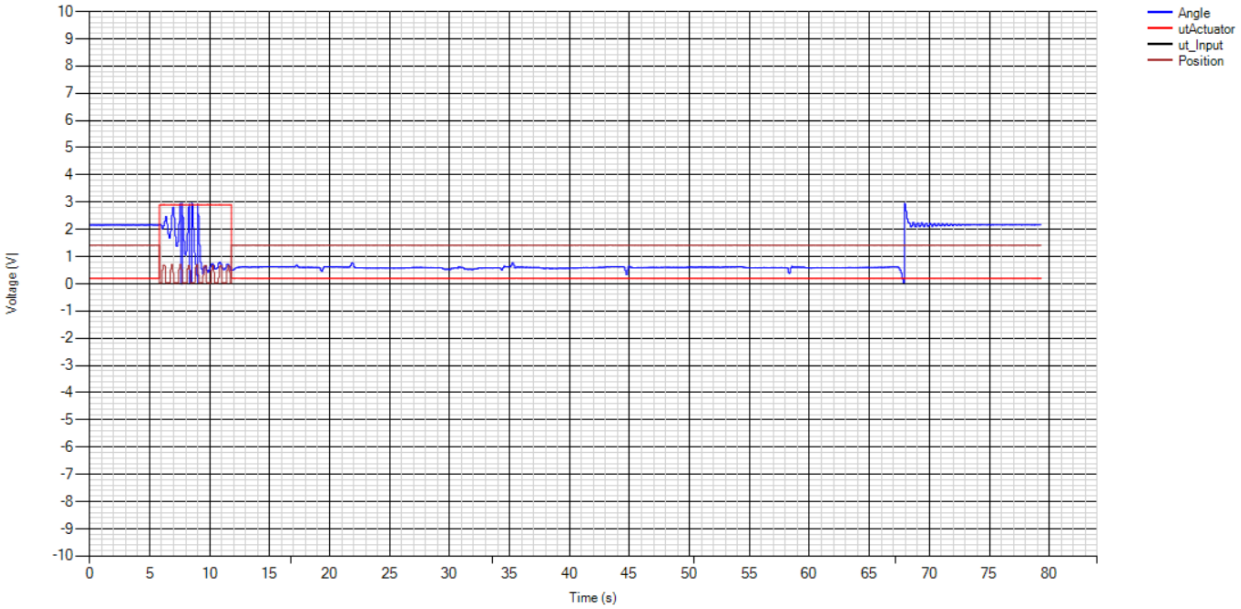

Figure 50: Swinging and balancing of the Inverted Pendulum in Real time

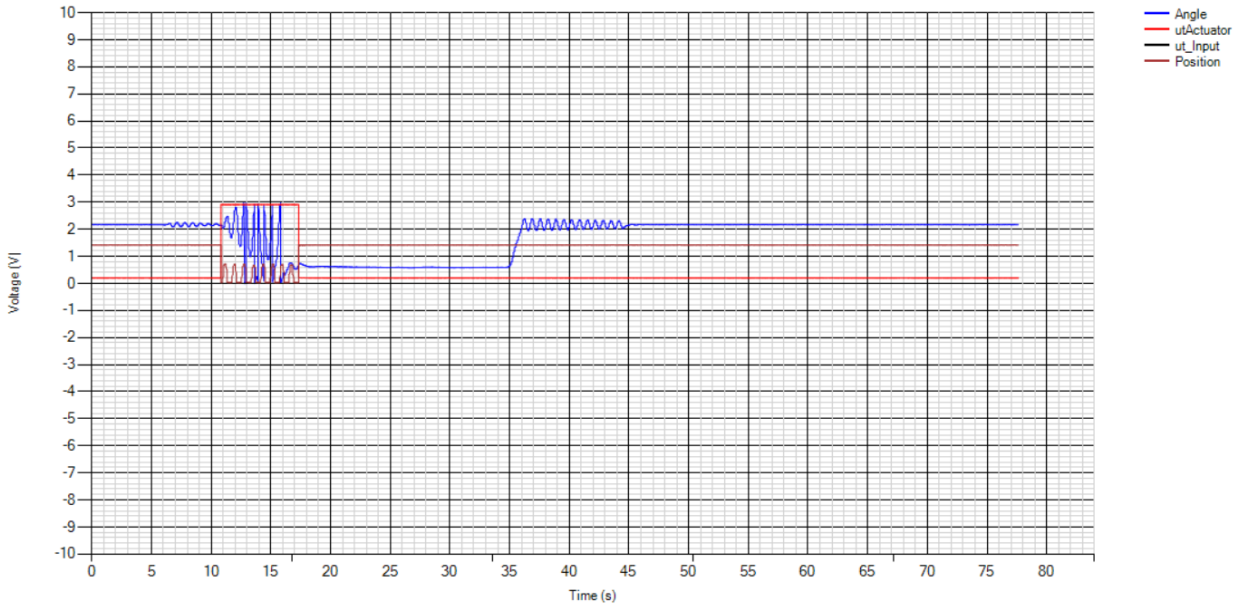

Figure 51: Real time Swing up and Balancing of the Pendulum

10. EBE Faculty: Assessment of Ethics in Research Projects