Stabilization of an Inverted Pendulum via a Swinging Arm

By

Musab Jawad Salameh

Oday Fadel Idais

Project Supervisor:

Dr. Yousef M. Alsweiti

Submitted to College of Engineering

in partial fulfillment of the requirements for the

Bachelor degree in Mechatronics Engineering

Palestine Polytechnic University

Dec. 2019

Palestine Polytechnic University
Collage of Engineering
Mechanical Engineering Department
Hebron – Palestine

Stabilization of an Inverted Pendulum via a Swinging Arm

By

Musab Jawad Salameh

Oday Fadel Idais

Submitted to the Collage of Engineering
In partial fulfillment of the requirements for the
Bachelor degree in Automotive Engineering.

Supervisor Signature

....

Department Head Signature

....

Jan. 2020

Abstract

The inverted pendulum represents a challenging problem in control and it has been widely used to investigate and develop new control strategies that can effectively deal with non-linearities. Our “Inverted Pendulum” consists of a horizontal rod (swinging arm) and vertical rod ("pendulum"). The horizontal rod is connected to a motor to balance and control the angle of the vertical rod by state feedback method, that is designed and simulated by MATLAB and Simulink. [1]

The aim of this project is to stabilize the inverted Pendulum at its perpendicular state (equilibrium), such that the angle is controlled quickly and accurately so that the pendulum is always be at that angle in the presence of disturbances, simulation results show that the pendulum is stabilized and achieved the desired transient response specifications using state feedback control strategy while considering the input as a torque applied to the upper joint connecting the swinging arm with the pendulum, also we considered and achieved the stability robustness due to variation in the system parameters.

ABSTRACT IN ARABIC...

يمثل البندول المقلوب مشكلة صعبة في مجال التحكم وقد استخدم على نطاق واسع لتحقيق وتطوير استراتيجيات تحكم جديدة يمكنها التعامل بفعالية مع الأنظمة غير الخطية يتكون "البندول المقلوب" من قضيب أفقي (ذراع متأرجح) وقضيب رأسي "البندول". يتم توصيل القضيب الأفقي بمحرك لتحقيق التوازن والتحكم في زاوية العمود العمودي باستخدام طريقة state feedback control ، التي صممت باستخدام MATLAB ومحاكاة في Simulink. الهدف من هذا المشروع هو تثبيت البندول المقلوب في حالته العمودية (التوازن) ، بحيث يتم التحكم في الزاوية بسرعة وبدقة بحيث يكون البندول دائمًا في تلك الزاوية في وجود التقلبات . وتظهر نتائج المحاكاة أن البندول استقر وحقق مواصفات الاستجابة المرغوبة باستخدام استراتيجية state feedback control مع الأخذ في عزم دوران مطبق على المفصل الأعلى الذي يربط ذراع التأرجح مع البندول ، كما يتم النظر في متانة الاستقرار بسبب الاختلاف في معاملات(ثوابت) النظام.

Dedication

We dedicate this project to our great parents and our lovers, we could never have done this work without their faith, support and ceaseless encouragement.

Thank you for all the unconditional love, guidance and support that you have always given us. We will never let you down and we will be able to do the right things in our future life.

Thank you again for believing in us and for everything.

Acknowledgment

We would like to express our deep gratitude to our supervisor Dr. Yousef M. Alsweiti for his hard work, patience, guidance and for his encouragement throughout the preparation of this project, he inspired us great to work in this project and to do the measurements so accurately, and his knowledge and endless support was a great asset from which we learned plenty.
Special thanks to Eng. Mahmoud Alsharabati for his support, finally, we would like to thank the staff of mechanical engineering department for standing with us and helping us in every possible way to complete and end this project in a pretty and appropriate way.

Table of Contents

List of Figures

List of symbols

Symbol	Definition	SI-Unit
l_2	Length of vertical link.	m
l_1	Length of upper link (horizontal link).	m
φ	The angle between the perpendicular line of first link and the second link.	Degree
θ	The angle between the vertical state (90 Degree) and the link.	Degree
m_1	Mass of the upper link (horizontal link).	kg
m_2	Mass of vertical link.	kg
m_3	Mass hanged.	kg
m_c	Mass for motor case.	kg
m_a	Mass for motor armature.	kg
g	Gravity force.	$\frac{m}{s^2}$
R_a	Radius for armature.	m
J_1	Polar moment of inertias for vertical link (Pendulum)	$kg.\ m^2$
J_2	Polar moment of inertias for horizontal link (Arm link)	$kg.\ m^2$
T_1	Kinetic energy for vertical link (Pendulum)	J
T_2	Kinetic energy for horizontal link (Arm link)	J
U_1	Potential energy for vertical link (Pendulum)	J
U_2	Potential energy for horizontal link (Arm link)	J
y_2	Half-length of the arm link	m
y_c	Length of the arm link	m
I_G	Moment of inertia for the center of gravity for horizontal part	$kg.\ m^2$

Chapter 1: **Introduction**

1.1 Recognition of the need

1.2 Requirements

1.3 System overview

1.4 Literature reviews

1.5 Alternative design

1.1 Recognition of the need

Need to stabilize and control an inverted pendulum using a horizontal swinging arm that is attached to the top of the pendulum by using any dedicated control method, and conduct some experiments that can help us to understand. In addition to do the modeling, analysis and the controller design and simulate the experiment.

1.2 Requirements:

This section discusses the requirements of the system as follows:

1) The pendulum must be stable in its pedicellar state.
2) The system should return to its initial condition in time period two sec (settling time), with an overshoot less than 10%.
3) The system can handle with disturbances that effect the system that less than 30 degree (0.523 radian).
4) The system is made of Aluminum, on plant, open system, no necessary to be painted.
5) The length of pendulum arm less than one meter, the length of swinging arm less than twice of the pendulum arm length.
6) Power supply must be at most 24VDC.
7) The system must be safe for the user and the system itself.
8) The system will have switch on/off power, the disturbances that will affect the system will by hand (small external force).
9) The system must be easy to maintenance, and further development on future.

1.3 System overview:

The Figure 1-1 shows the system that is built with the requirements that are proposed and stated above. We add more components if we need to make it more stable and safer. However, the setup in Figure1-1 includes:

1. DC motor with Encoder2 to measure φ .Therefore, we can observe the angular velocity $\dot{\varphi}$.
2. Encoder1 at the end of pendulum arm to measure θ .Therefore, we can observe the angular velocity $\dot{\theta}$.
3. Two masses in the horizontal link (Hanged mass).
4. Horizontal link attached with motor and vertical link attached with encoder.

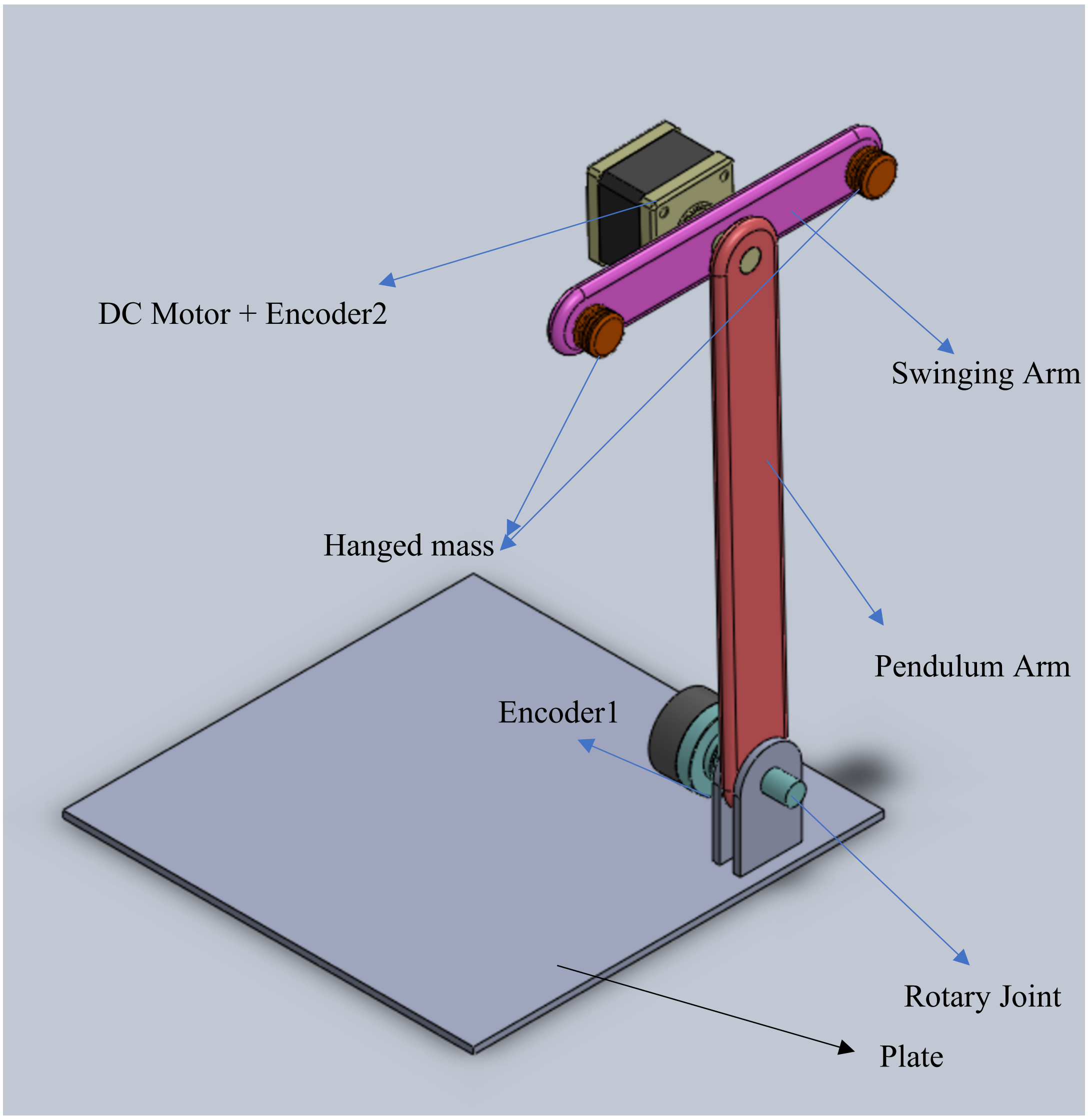

Figure 1-1 System overview

1.4 Literature reviews

Many literatures, papers and researches study the inverted pendulum, especially different ways and concepts to stabilize the inverted pendulum, changing the center of gravity of the system, moving cart on the center of the inverted pendulum, using different types of controllers to stabilize the equilibrium point and reject disturbances.
We will now show you some literature reviews that we collect as following:

1.4.1 Swing-up control of inverted pendulum using pseudo state feedback:

Yamakita, and Kobayashi [2] have studied the swing-up of a pendulum from the stable hanging state to the upright position, and he presents a new type of pendulum on a rotating arm fixed to a rotating shaft and a swing up control algorithm based on state space feedback. *Figure (1-2)* represents the proposed controller realized on a personal computer.

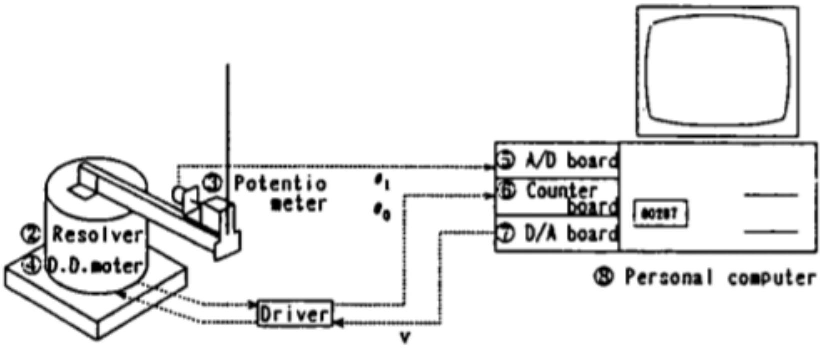

Figure 1-2 Configuration of the system [2]

They proposed a robust swing-up control using a subspace projected from the whole state space based on pseudo-state, the control determined depending on the partitioning of the state as a bang-bang type control. [2]

They applied a new kind of inverted pendulum (TIT ech pendulum), and the effectiveness and robustness of the planned control checked and examined by many experiments.

1.4.2 Control of the inverted pendulum using poles placement pole with help Ackerman's method:

Priecinsky, and Paskala [3] used modern control method by poles placement with help of Ackerman's method, *Figure (1-3)* represents their system that consist of cart, motor and vertical pendulum. Ackermann's formula is used for inverted pendulum control, with a view to determine full-state feedback gain. It is generalized in the sense of desired eigenvalues instead of characteristic polynomial coefficients.

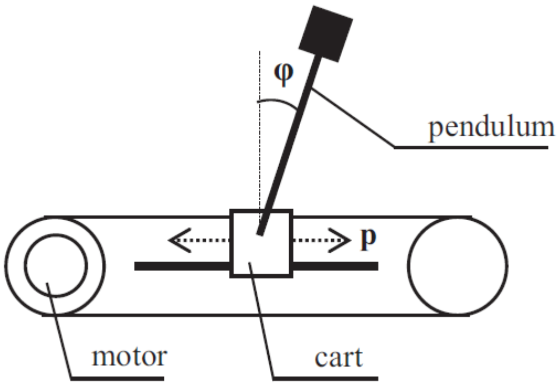

Figure 1-3 Setup of inverted pendulum [3]

They tested their system using simulation analysis and in real - time experiments which consists of hardware system PS600 and control units, see *Figure (1-4)*

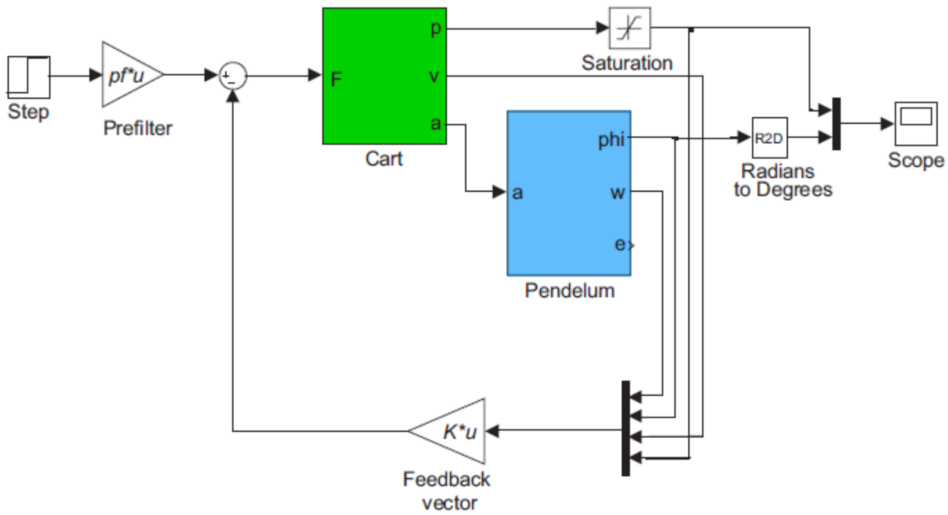

Figure 1-4 Simulink scheme of full-state feedback control

1.4.3 Stabilizing inverted pendulum using state feedback:

Hroub [4] has studied ECP model 505 inverted pendulum system consists of a horizontal sliding rod and vertical "pendulum" rod. *Figure (1-5)* represents the whole system that consist of a horizontal rod that is connected to electrical motor through rack and pinion mechanism. Thus, it steers left or right to balance and control the position of the vertical rod. He used state feedback control for stabilize their system and using MATLAB as a major software.

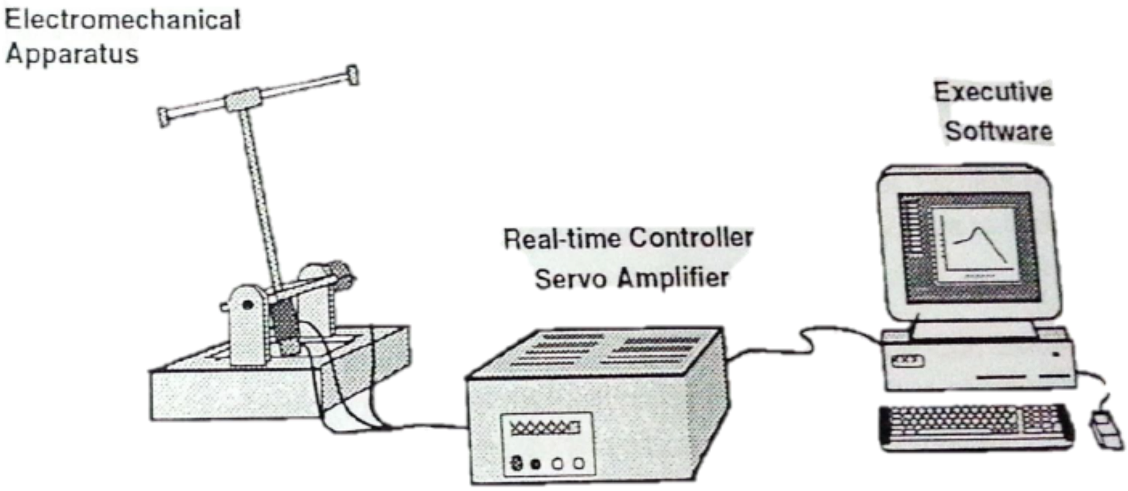

Figure 1-5 Configuration of the system

The structure of controller must be selected carefully due to the non-minimum phase characteristics.

1.5 Alternative design:

The plant that is shown in Figure 1-6 is the ECP model 505 Inverted pendulum apparatus, which exists in the computer control lab at Palestine Polytechnic University (PPU). It consists of pendulum rod which supports the sliding balance rod. The mechanism itself is open-loop unstable and non-minimum phase, thus closed-loop feedback control is essential for equilibrium point. The balance rod is driven via a belt and pulley which in turn is driven by a drive shaft connected to a dc servo motor below the pendulum rod. The pendulum rod angle is controlled by moving the sliding rod on the presence of gravity using servo motor in the end of pendulum arm. The weights at the bottom may be adjusted to alter the inertia configurations of the pendulum rod, and as a result the dynamics of the system. A brushed dc motor and encoders are used to drive the sliding rod through measurements of the angular position of the pendulum rod and linear position of the sliding rod. Therefore, the only input on the plant is the force applied at the sliding rod. [3]

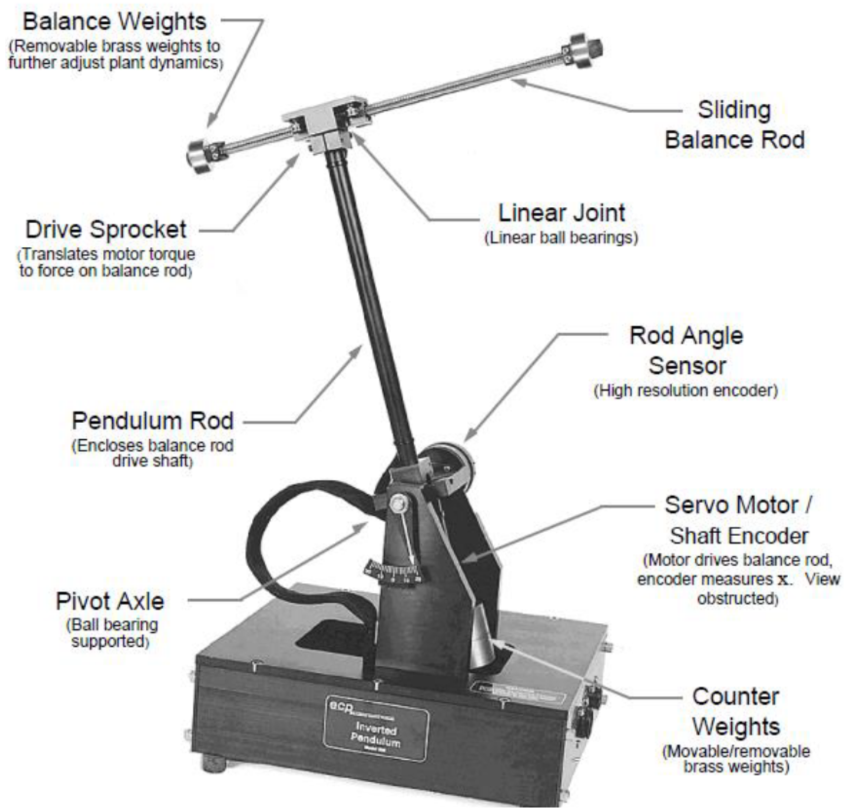

Figure 1-6 The ECP model 505 Inverted pendulum [3]

Chapter 2: **Conceptual design**

2.1 Introduction

2.2 Conceptual design schematic

2.1 Introduction

This section describes the stabilization of an inverted pendulum via a swinging arm workflow, including the system components (subsystems), parts functions and relations between elements, see Figure 1-1.

2.2 Conceptual design schematic

Conceptual design schematic for our project in the Figure 2-1:

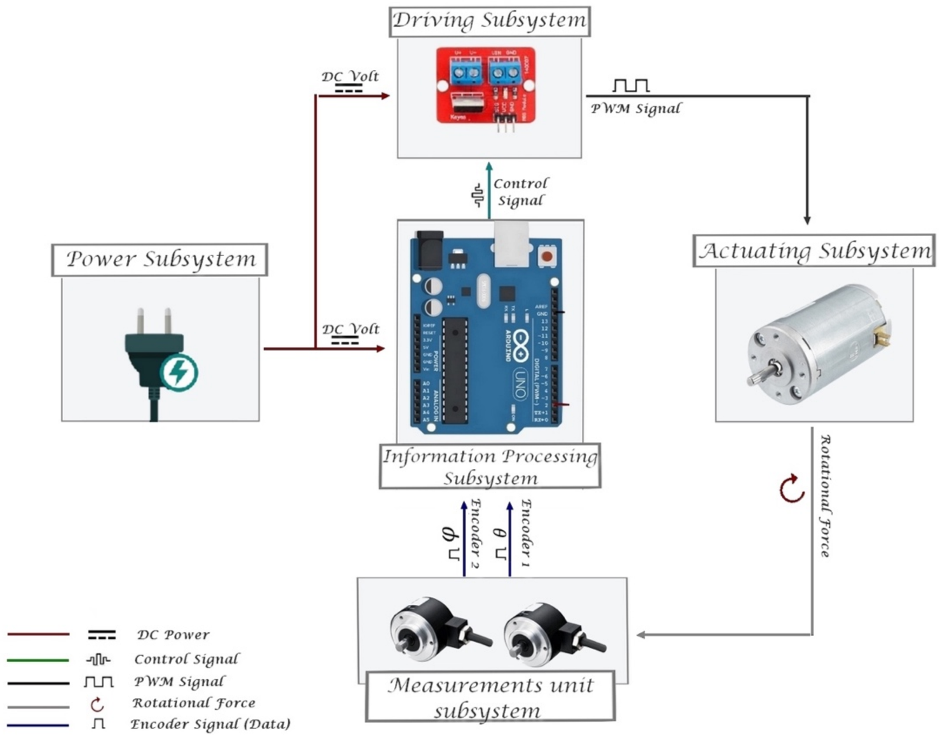

Figure 2-1 Conceptual design schematic

Figure 2-1 shows the system operative blocks, we have power supply that is connected to both controller and driver. The driver is connected to the controller that process the value of Encoder1 and Encoder2, the Encoder1 provides the state of angle θ, and the Encoder2 provides the state of angle φ, and gives actions to the driver that connected to the motor to rotate and, and the encoder1 provide us feedback for any change in θ so, the controller will generate different actions that depends on the state of θ.

Chapter 3: **Design of mechanical components**

3.1 Introduction

3.2 Design requirements

3.1 Introduction

This chapter will discuss the mechanical design of our project that shown in Figure 1-1.

3.2 Design requirements

First, we define specifications that are supposed to be in our project, the requirements of the design as follows:

- The system is made from the Aluminum, and on a plant that not declined, also the system should be open system, and there is no need to be painted.
- The system must be safe by turning off the DC motor if the vertical link (pendulum) declines more than 20 degree.
- The length of pendulum arm is between 20cm and 50cm, since the length of the vertical link is 30cm, the length of swinging arm less than twice of the pendulum arm length.

3.3 Design through solid works

In this section we design the mechanical parts using SolidWorks [5], the mechanical parts that we designed include the pendulum arm and swinging arm.

Chapter 4: **Mathematical modeling**

4.1 Introduction

In this chapter we will show the mathematical model of the system that consists of two second order nonlinear differential equations, these equations are derived using Lagrange approach, two models will be derived for the system, first the linear system for controller design and analysis purposes, second nonlinear model for testing and simulating the dynamic system response as accurately as possible.

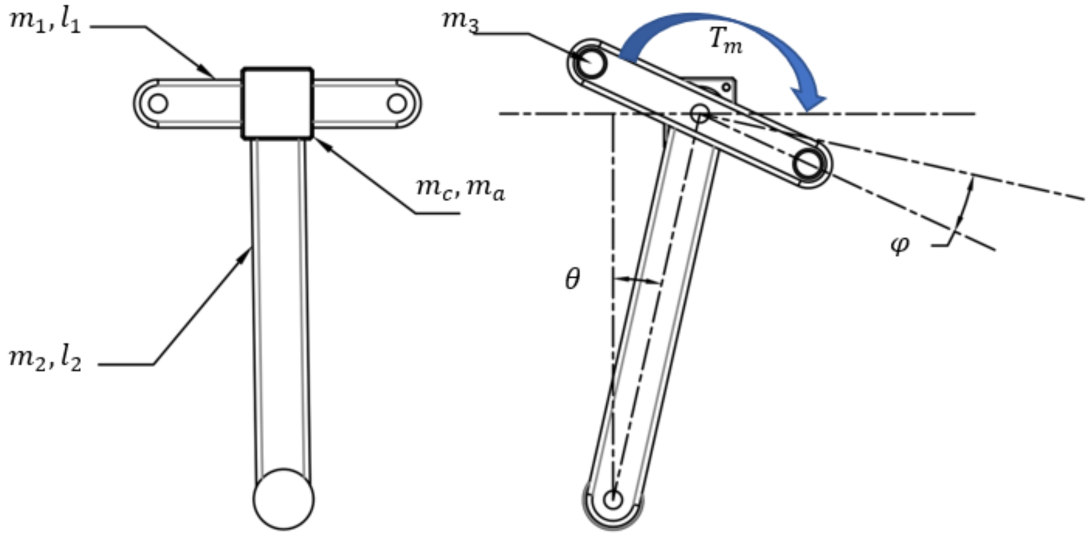

Figure 4-1 The system in 2-D

Figure 4-1 represents the location of the angles and the input of the system, and each symbol what is represented for.

4.2 Lagrange approach

In order to obtain the mathematical model for our system, Lagrange's approach [6] is used to drive the basic differential equations that govern and optimize the system dynamics, since Newton's formulation of classical mechanics is not convenient.

The Lagrange differential equation:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \left(\frac{\partial T}{\partial q_i}\right) + \left(\frac{\partial U}{\partial q_i}\right) = Q_i \qquad (4\text{-}1)$$

Where:

T: The total kinetic energy of the system.
U: The total potential energy of the system.
qi: The generalized coordinates that describes system motion.
Qi: Generalized forces and torques which acts in each generalized coordinate.

4.3 Non-linear expression

Based on Figure 4-1, the total kinetic and potential energies of the system can be expressed as:

For vertical part (pendulum) (l_2):

Total kinetic energy:

since we have rotation about fixed axis the kinetic energy will be

$$T_1 = \frac{1}{2}J_2\dot{\theta}^2 + \frac{1}{2}m_c\left(l_2\dot{\theta}\right)^2 \tag{4-2}$$

and J_2 is equal to;

$$J_2 = \frac{m_2\, {l_2}^2}{3}$$

Where J_2 is the polar moment of inertia for horizontal link , and $\dot{\theta}$ is the angular velocity.

The total kinetic energy will be;

$$T_1 = \frac{1}{2}\left[\frac{m_2 l_2^2}{3} + m_c l_2^2\right]\dot{\theta}^2 \tag{4-3}$$

Total potential energy:

$$U_1 = m_2 g y_2 + m_c g y_c \tag{4-4}$$

where:

$$y_2 = \frac{l_2}{2}cos\theta$$

and

$$y_c = l_2 cos\theta$$

Then the total potential energy will be;

$$U_1 = \left[\frac{m_2}{2} + m_c\right] g l_2 cos\theta \tag{4-5}$$

For horizontal part (l_1):

Total kinetic energy:

Since we have translation and rotation the kinetic energy will be;

$$T_2 = T_{translation} + T_{rotation} \tag{4-6}$$

where:

$$T_{translation} = \frac{1}{2}[m_a + 2m_3 + m_1](l_2\dot{\theta})^2$$

$$T_{rotation} = \frac{1}{2} I_G(\dot{\varphi} + \dot{\theta})^2$$

$$I_G = 2m_3\left(\frac{l_1}{2}\right)^2 + \frac{1}{12}m_1 {l_1}^2 + \frac{1}{2}m_a {R_a}^2$$

and I_G is moment of inertia for the center of gravity for horizontal part.

Total potential energy:

The total potential energy for this part;

$$U_2 = (m_a + m_1) g l_2 cos\theta + 2m_3 g l_2 cos\theta \tag{4-7}$$

Now for the total energy for the whole system:

The total kinetic energy is the summation between T_1 and T_2;

$$T_{total} = T_1 + T_2$$

$$T_{total} = \frac{1}{2}\left[\left(\frac{m_3 l^2{}_2}{3} + l^2{}_2 m_c\right)\dot{\theta}^2 + (m_a + 2m_3 + m_1)(l^2{}_2\dot{\theta}^2) + (2m_3\frac{{l_1}^2}{4} + \frac{1}{12}m_1 l^2{}_1 + \frac{1}{2}m_a R^2{}_a)(\dot{\theta} + \dot{\varphi})^2\right] \tag{4-8}$$

and the total potential energy:

The total potential energy is the summation between U_1 and U_2;

$$U_{total} = U_1 + U_2$$

$$\rightarrow U_{total} = gl_2 cos\theta \left[\left(\frac{m_2}{2} + m_c\right) + (m_a + m_1) + 2m_3\right] \tag{4-9}$$

Appling Lagrange's equation for each generalized coordinate θ and φ, yields:

1) In $\boldsymbol{\theta}$ direction:

The Lagrange's equation in θ direction given as follows:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) - \left(\frac{\partial T}{\partial \theta}\right) + \left(\frac{\partial U}{\partial \theta}\right) = 0 \tag{4-10}$$

where:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) = l_2^2 \ddot{\theta}\left[\frac{m_2}{3} + m_a + 2m_3 + m_1\right] + \ddot{\varphi}\left[2m_3\left(\frac{l_1}{2}\right)^2 + \frac{1}{12}m_1 l_1^2 + \frac{1}{2}m_a R_a^2\right] + \ddot{\theta}\left[2m_3\left(\frac{l_1}{2}\right)^2 + \frac{1}{12}m_1 l_1^2 + \frac{1}{2}m_a R_a^2\right]$$

$$\frac{\partial T}{\partial \theta} = 0$$

$$\frac{\partial U}{\partial \theta} = -gl_2 \sin\theta\,[[m_a + 2m_3 + m_1] + \frac{m_2}{2} + m_c]$$

Let assumed that;

$$C_1 = \left[2m_3\left(\frac{l_1}{2}\right)^2 + \frac{1}{12}m_1 l_1^2 + \frac{1}{2}m_a R_a^2\right]$$

$$C_2 = l_2^2\left[\frac{m_2}{3} + m_a + 2m_3 + m_1\right]$$

$$C_3 = g l_2[[m_a + 2m_3 + m_1] + \frac{m_2}{2} + m_c]$$

Applying Eq. (4-10) and simplifying it yields:

$$\ddot{\theta}(C_1 + C_2) + \ddot{\varphi}C_1 - \sin\theta\, C_3 = 0 \qquad (4\text{-}11)$$

2) In φ direction:

The Lagrange's equation in φ direction given as follows:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\varphi}}\right) - \left(\frac{\partial T}{\partial \varphi}\right) + \left(\frac{\partial U}{\partial \varphi}\right) = T_m \qquad (4\text{-}12)$$

Where;

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\varphi}}\right) = \ddot{\varphi}\left[2m_3\left(\frac{l_1}{2}\right)^2 + \frac{1}{12}m_1 l_1^2 + \frac{1}{2}m_a R_a^2\right] + \ddot{\theta}\left[2m_3\left(\frac{l_1}{2}\right)^2 + \frac{1}{12}m_1 l_1^2 + \frac{1}{2}m_a R_a^2\right]$$

$$\frac{\partial T}{\partial \varphi} = 0$$

$$\frac{\partial U}{\partial \varphi} = 0$$

Applying *Eq. (4-12)* and simplifying it yields:

$$\ddot{\theta}C_1 + \ddot{\varphi}C_1 = T_m \qquad (4\text{-}13)$$

4.3.1 Linearization

In this section, we want to take the linear case of the previous equations; we want to linearize our system to build the state space model;
Using *Eq. (4-11)* and *Eq. (4-13)*

Now by assuming that the both angles φ and θ are small values, that is:

$$sin\theta = \theta$$
$$sin\varphi = \varphi$$
$$cos\theta = 1$$
$$cos\varphi = 1$$

thus,

$$\dot{\theta}^2, \dot{\varphi}^2, \dot{\theta}, \dot{\varphi} = 0$$

Therefore *Eqn.4-11 and Eqn.4-13* can be written as:

$$\ddot{\theta}(C_1+C_2) + \ddot{\varphi}C_1 - \theta C_3 = 0 \tag{4-14}$$

$$\ddot{\theta}C_1 + \ddot{\varphi}C_1 = T_m \tag{4-15}$$

4.3.2 State space model

Using the linearized model of the system we have:
See *Eqn.4-14* and *Eqn.4-15*
Now by solving the equations we found the values of $\ddot{\theta}$ and $\ddot{\varphi}$

$$\ddot{\theta} = \theta \frac{C_3}{C_2} - \frac{T_m}{C_2} \tag{4-16}$$

$$\ddot{\varphi} = \theta \frac{C_3}{C_2} - \frac{T_m(1+\frac{C_2}{C_1})}{C_2} \tag{4-17}$$

Now, to yield the state space representation for the linear system, four state are needed to describe the system. These are chosen to be $\theta, \dot{\theta}$,φ and $\dot{\varphi}$. The output of the system is T_m, the state space model of the system is expressed as follows:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$$
$$\boldsymbol{y} = C\boldsymbol{x}$$

where:

A; is the plant system matrix.

B; is the input matrix.

C; is the output matrix.

Let us assume that:

$x_1 = \theta \ \rightarrow \ \frac{dx_1}{dt}$ gives → $\dot{x_1} = x_2$

$x_2 = \dot{\theta} \ \rightarrow \ \frac{dx_2}{dt}$ gives → $\dot{x_2} = \ddot{\theta} = x_1 \frac{C_3}{C_2} - \frac{u}{C_2}$

$x_3 = \varphi \ \rightarrow \ \frac{dx_3}{dt}$ gives → $\dot{x_3} = x_4$

$x_4 = \dot{\varphi} \ \rightarrow \frac{dx_4}{dt}$ gives → $\dot{x_4} = \ddot{\varphi} = x_1 \frac{C_3}{C_2} - \frac{u(1+\frac{C_2}{C_1})}{C_2}$

where:

$$u = \ T_m$$

Thus, the state space model for the system is described by the state equations:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ (C_3/C_2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ (C_3/C_2) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-1}{C_2} \\ 0 \\ \frac{-(1+\frac{C_2}{C_1})}{C_2} \end{bmatrix} u$$

and the measurement equation:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \end{bmatrix} \qquad \textit{(4-18)}$$

All states are measured using two encoders because each encoder help us to observe the other states of the angular velocity for each angle.

For output of the system we have two angles, but we have one angle to be monitored and take it value so,

$$y_o = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \end{bmatrix} \qquad \textit{(4-19)}$$

Since;

$$C_o = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

y_o: is the output to be controlled.
C_o: is the controlled output matrix.

The system parameters are defined in Figure 4-2;

```
Ts=0.0628318;%% (sec) Sampling time.
l2 =0.37; %% (m)length of vertical link.
l1=0.34; %% (m)length of upper link (horizantal).
mc =0.1325; %% (kg)mass for motor case.
ma = 0.1325; %% (kg)mass for motor armuture.
m3 =0.1;%% (kg)mass hanged.
m1 =0.232;%% (kg)mass of the upper link (horizantal).
m2 =0.1915;%% (kg)mass of vertical link .
Ra = 0.006;%% (m)raduis for armuture.
g=9.81; %%(m/s^2) gravity force.
 %%we assumed the values for the lengths and masses of our system
```

Figure 4-2 System parameters

The Figure 4-2 represents the real parameters, that used for achievement this project, to get satisfied response. We know that our parameters are suitable from the simulation that we used in MATLAB.
By substituting the values in the Figure 4-2 into state space model matrices yields;

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -31.8182 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -31.8182 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ -13.1182 \\ 0 \\ -345.8276 \end{bmatrix} u \tag{4-20}$$

$$y_o = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \end{bmatrix} \tag{4-21}$$

4.4 Controller design

4.4.1 Introduction

In this chapter we want to design a controller to give the requirements we decided using a state feedback controller needed to be robust here, we will talk about the sate feedback controller for the linear system, non-linear system, comparing between the outputs states of two systems, and the robustness parameters limits of the controller, the controller is to be built by MATLAB and Simulink [9].

In control engineering, a state space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations. The state space representation (also known as the "time-domain approach") provides a convenient and compact way to model and analyze systems with multiple inputs and outputs. Unlike the frequency domain approach, the use of the state space representation is not limited to systems with linear components and zero initial conditions. However, in state feedback method you can place the eigenvalues anywhere in the S-plane to get the desired response, in Figure 4-3, we note the design of state space plant.

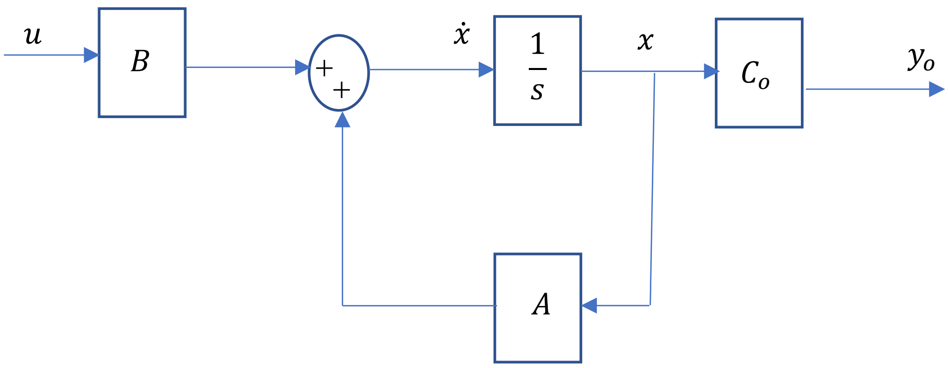

Figure 4-3 Plant

4.4.2 Linear system

According to the linearized equations of our system, see *Eq. (4-14)* and *Eq. (4-15)* and the state space representation that we build, see *Eq. (4-20)* and *Eq. (4-21)*

Now, we will apply the state feedback controller for the linear system, according to the requirements for the project we apply damping ratio = [0.8,0.9] that gives us overshoot less than 10%, for that the system must return to the stable condition in less than two seconds we apply Settling time less than two seconds, we have natural frequency = [2.22-2.5](rad/s), by adding two times factor of safety to the system we have natural frequency 5 (rad/s), to be able to design a state feedback controller, the controllability of the system must be checked. If an input to a system can be found that takes every state variable from a desired initial state to a desired final state, the system is said to be controllable; otherwise, the system is uncontrollable.
To check the possibility for the closed loop poles of the system; as to achieve stability and desired transient response, the controllability of the system is checked. The controllability of the pair (A, B) is checked by calculating the controllability matrix (C_M), such that:

$$C_M = [B\ AB\ \cdots A^{n-1}B]_{n*m}$$

If C_M has a rank n (full row rank), then the system is controllable, and it's possible to find a gain vector $[K]$. To find a gain vector pole placement concept will be used, pole placement method, in this method the gains are calculated as to place the eigenvalues of the system matrix,
which are the closed-loop poles, in the desired location. After determining the desired poles location, MATLAB function $(place)$ can be used to calculate the necessary gain values.

Regulator design:

For the regulator design, the problem in this section is to design a state feedback controller that stabilizes the inverted pendulum at its inverted position, which is the desired operating position, see Figure 4-4.

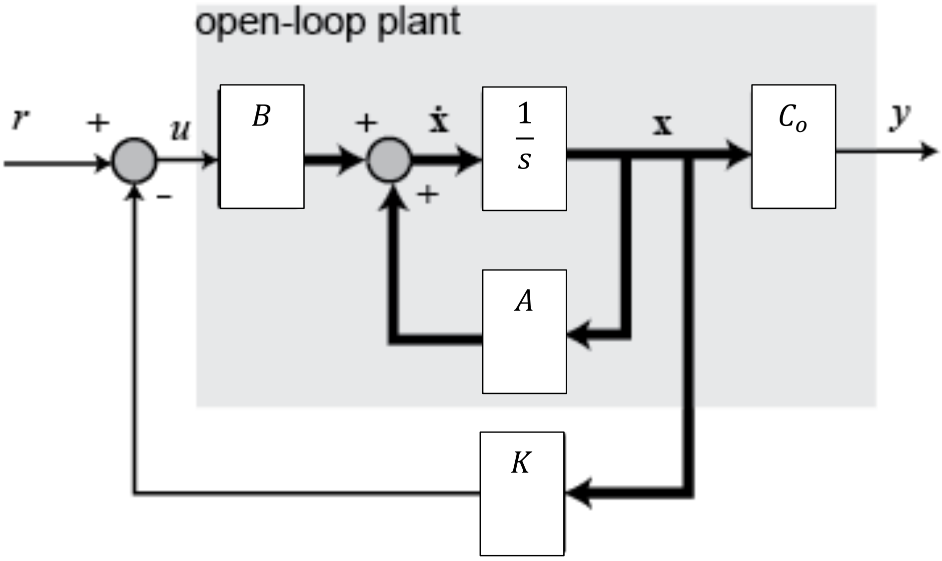

Figure 4-4 Plant design with regulator

1. **Principle:**

The inverted pendulum open loop system dynamics are given by:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

Recall that the system poles are given by the eigenvalues of A. Want to use the input $\boldsymbol{u}$ to modify the eigenvalues of A to change the system dynamics. Assume a full-state feedback of the form:

$$u = r - Kx$$

Where:

$\boldsymbol{r}$: is some reference input.

$\boldsymbol{K}$: is a gain vector.

Find the closed-loop dynamics:

$$\dot{x} = Ax + B(r - Kx)$$

$$\dot{x} = (A - BK)x + Br$$

$$\dot{x} = A_c x + Br$$

$$y = Cx$$

Where A_c: the closed loop system matrix

So, the eigenvalues of A_c (the closed loop poles) could be placed anywhere in the S-plane to get the desired response just by changing in matrix K. But to be able to do that, the open loop system must be controllable.

2. Controller:

The open loop poles and zero for the system are found to be:
Open loop poles:

0.0000 + 0.0000i
0.0000 + 0.0000i
0.0000 + 5.6408i
0.0000 - 5.6408i

Open loop zeros: two zeros on the origin [0+0i]
The open loop system is margin stable.
To check the controllability of the pair (A, B), first the controllability matrix is calculated, and then its rank is found. This is performed using MATLAB as follows:

$$C_m = ctrb(A, B)$$
$$R_c = rank(C_m)$$

Which is found to be 4, full rank, meaning that the system is fully controllable, and the gain $[K]$ can be calculated to achieve the desired response. The gain ***K*** can be found either by pole placement or optimal control methods. Assuming the natural frequency (ω_n) and the damping ratio (ζ) of the desired closed loop poles rang between (5) rad/s and (0.8 to 0.9) respectively.

By applying the requirements in equation → $-\zeta * \omega n \pm \omega n * \sqrt{(1-\zeta^2)}$ we get:

Poles = [-4.0000 -3.0000i -4.0000 +3.0000i -4.5000 -2.1794i -4.5000 + 2.1794i]

With MATLAB, the gains required to achieve the desired closed loop poles are found as follows:

$$K = place(A, B, Poles)$$

$$K = [-5.3181 \quad -0.2375 \quad -0.0590 \quad -0.0401]$$

3. Simulink model:

The regulator is built and simulated using MATLAB-Simulink, Figure 4-5 shows the Simulink model with initial condition 5 degree for pendulum arm angle.

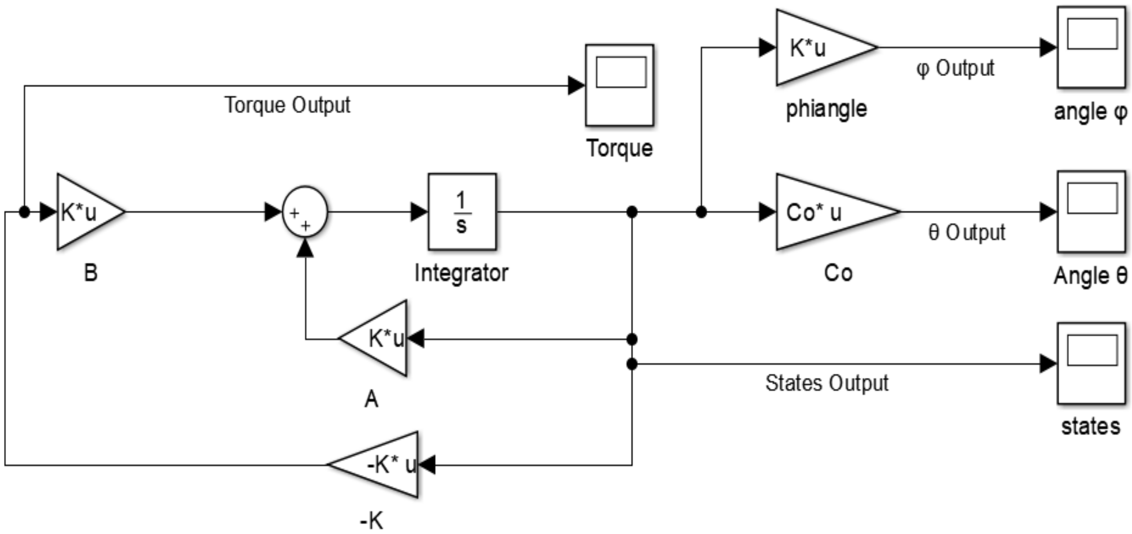

Figure 4-5 Linear system on Simulink

For response of the scopes as following:

For (angle (θ)) scope:

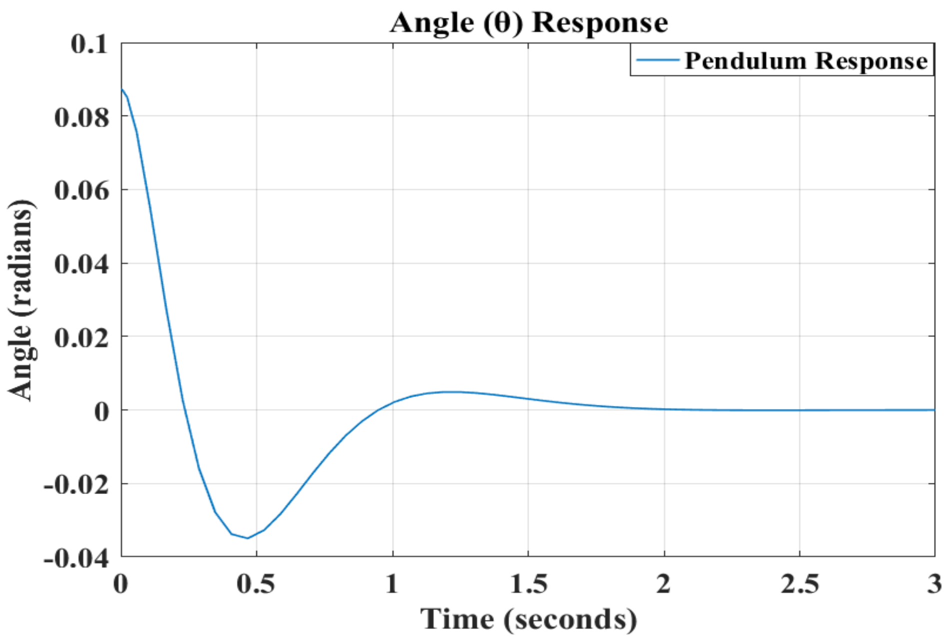

Figure 4-6 angle (θ) scope the condition of pendulum angle (θ)

In Figure 4-6, we see the state of the pendulum angle (θ), note that the response starts from the initial condition that we give, and it shows the settling time is around 2 (sec) and the maximum angle is around 0.035 (rad).

For (states) scope:

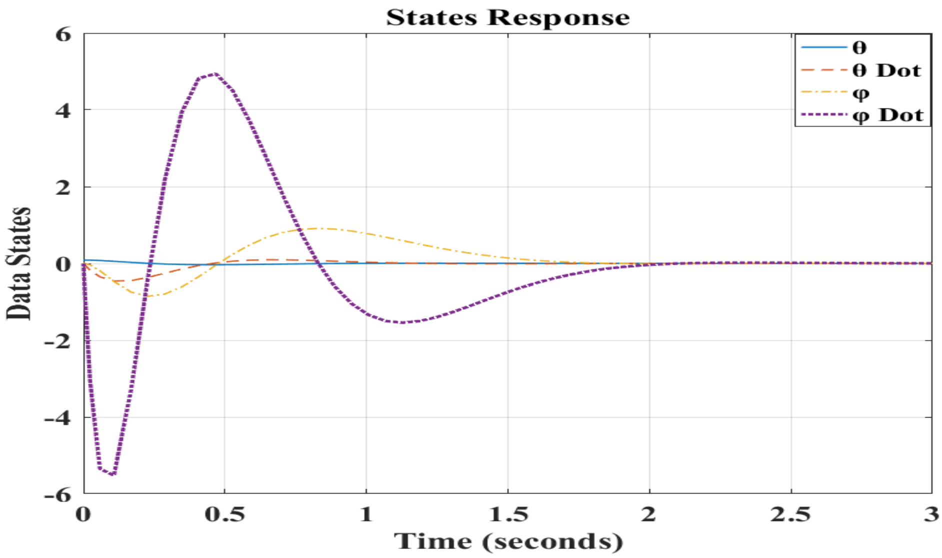

Figure 4-7 states scope output of each state

In Figure 4-7, this scope represented each state of the linear system, and the reaction of the state's corresponding to the initial condition, note that all states return on final value in less than 2 sec.

For (Torque) scope:

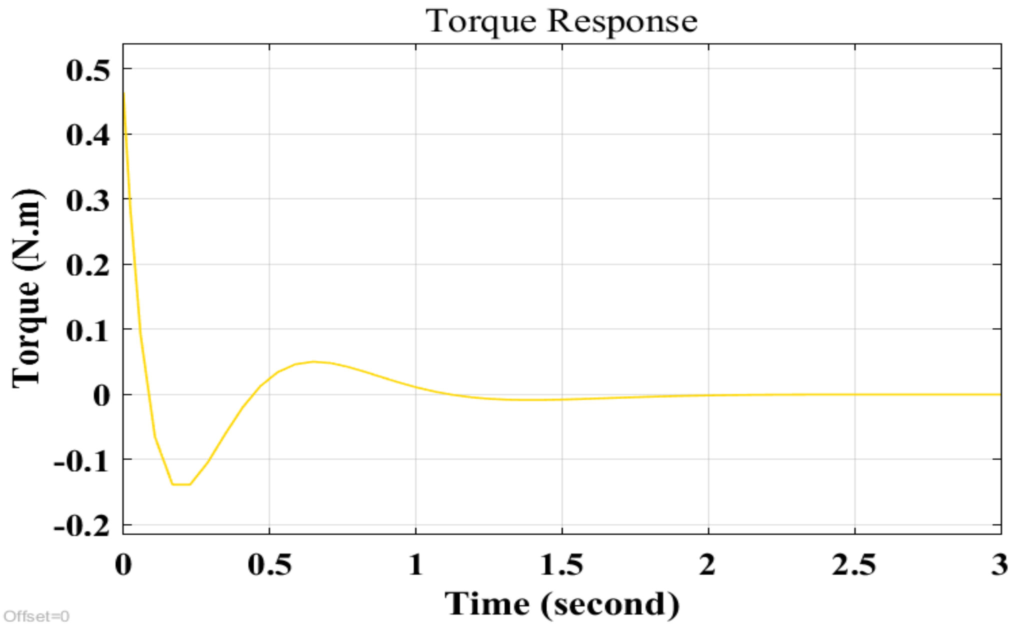

Figure 4-8 Torque scope expected torque of the motor.

The Figure 4-8, gives us approximate expected torque of the motor to be controlled to be used to build the system and it shows maximum torque at 0.138 (N.m) and overshoot 5%.

For angle (φ) scope:

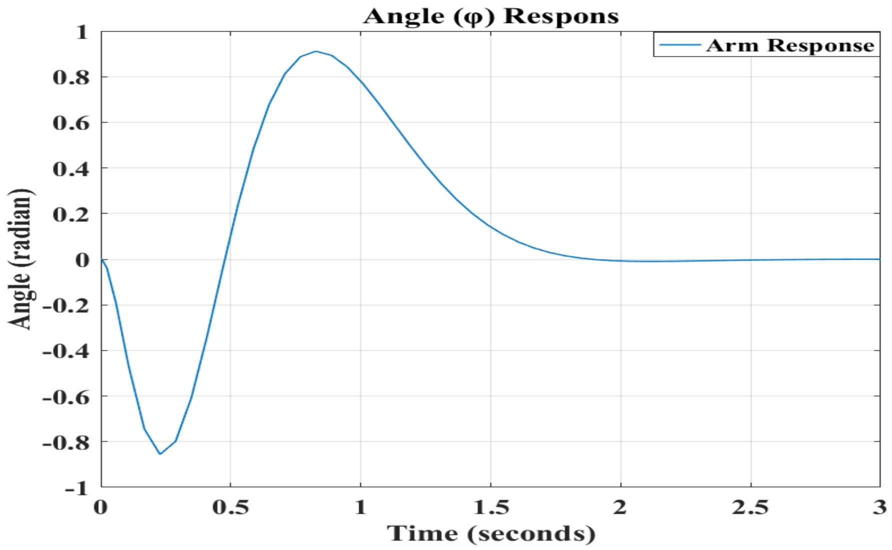

Figure 4-9 angle (φ) scope the condition of angle (φ).

In Figure 4-9, we note the reaction of the upper angle (φ) that moves to return to the stable state, around 0.91 (rad) and the settling time is around 1.91 (sec)

4.4.3 Non-linear system

According to the non-linear equations of the system;

$$\ddot{\theta}(C_1+C_2) + \ddot{\varphi}C_1 - \sin\theta\, C_3 = 0$$

$$\ddot{\theta}C_1 + \ddot{\varphi}C_1 = T_m$$

In this section we want to build the system according to the non-linear equations of the system.

Let;

$x_1 = \theta \rightarrow \frac{dx_1}{dt}$ gives → $\dot{x_1} = x_2$

$x_2 = \dot{\theta} \rightarrow \frac{dx_2}{dt}$ gives → $\dot{x_2} = \ddot{\theta} = sinx_1\frac{C_3}{C_2} - \frac{u}{C_2}$

$x_3 = \varphi \rightarrow \frac{dx_3}{dt}$ gives → $\dot{x_3} = x_4$

$$x_4 = \dot{\varphi} \;\rightarrow\; \frac{dx_4}{dt} \quad \text{gives} \rightarrow \quad \dot{x_4} = \ddot{\varphi} = sinx_1\frac{C_3}{C_2} - \frac{u(1+\frac{C_2}{C_1})}{C_2}$$

Since;

$$u = T_m$$

Now, the system will be built by the Simulink on the MATLAB, since we have four states each one of the states will be defined on functional blocks

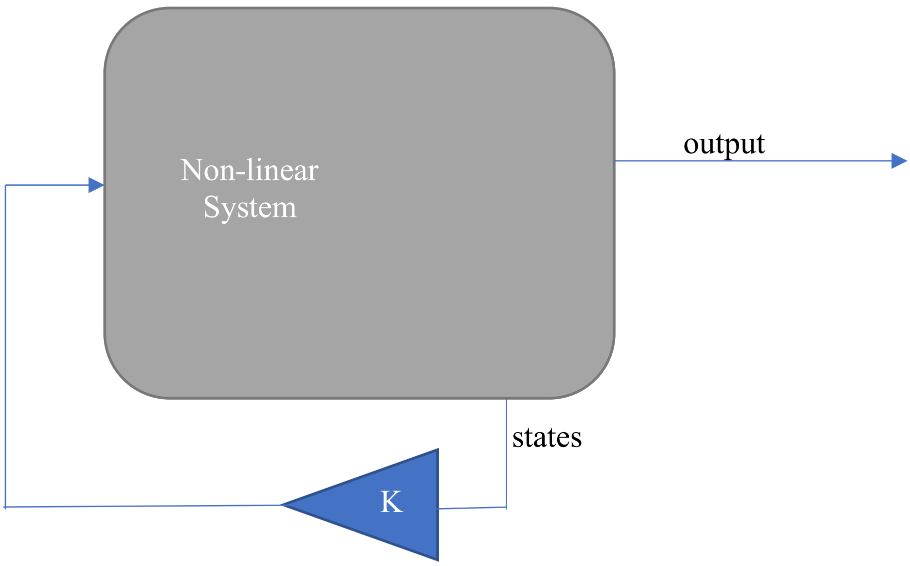

Figure 4-10 Non-linear system with linear gain

In Figure 4-10, non-linear system that have the gain for linear system as feedback.

1. Simulink build:

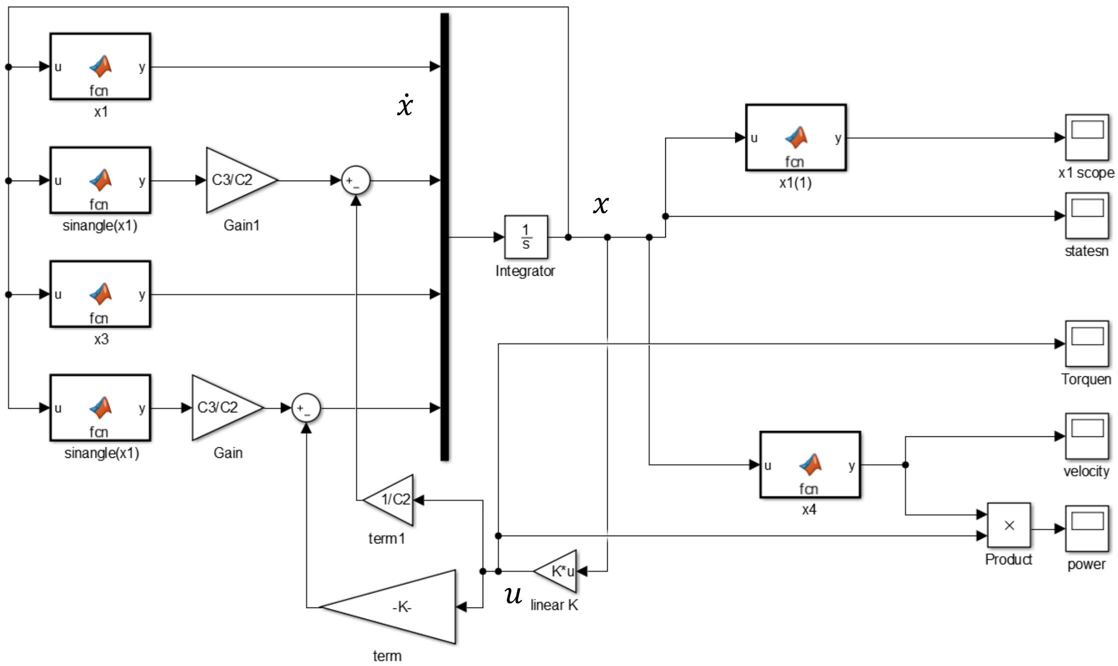

Figure 4-11 Non-linear system on Simulink

In Figure 4-11, the four states are built by the functional blocks using the same gain vector [linear K] for the linear system.

2. Output response:

In this part, after the non-linear system was built, we will take now over the scopes responses to compare them with linear system output, for the same initial condition.

For x1 scope:

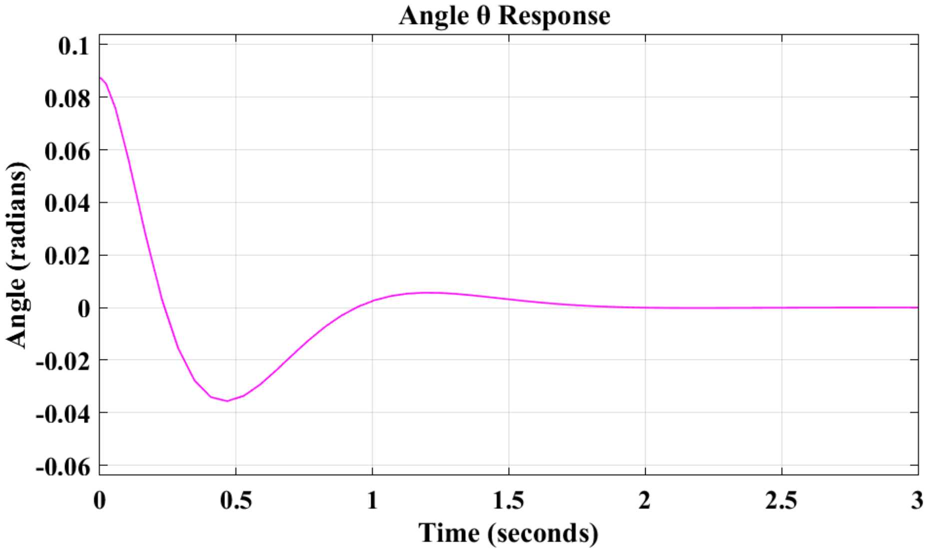

Figure 4-12 x1 Scope is the pendulum angle (θ)

The Figure 4-12, tells us the condition of the pendulum angle (θ), note that the response starts from the initial condition that we give, and it shows the settling time is around 2 (sec) and the maximum angle is around 0.035 (rad).

For states scope:

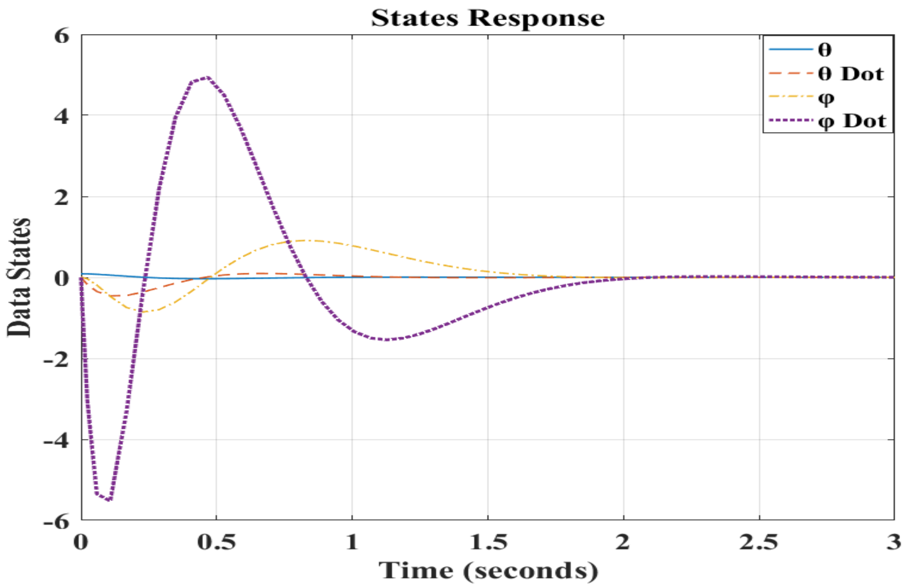

Figure 4-13 States scope is non-linear states conditions.

For Figure 4-13, this scope represented the states of the non-linear system, we see that the states of non-linear system are symmetry with the states of the linear system.

For Torque scope the output is as follows:

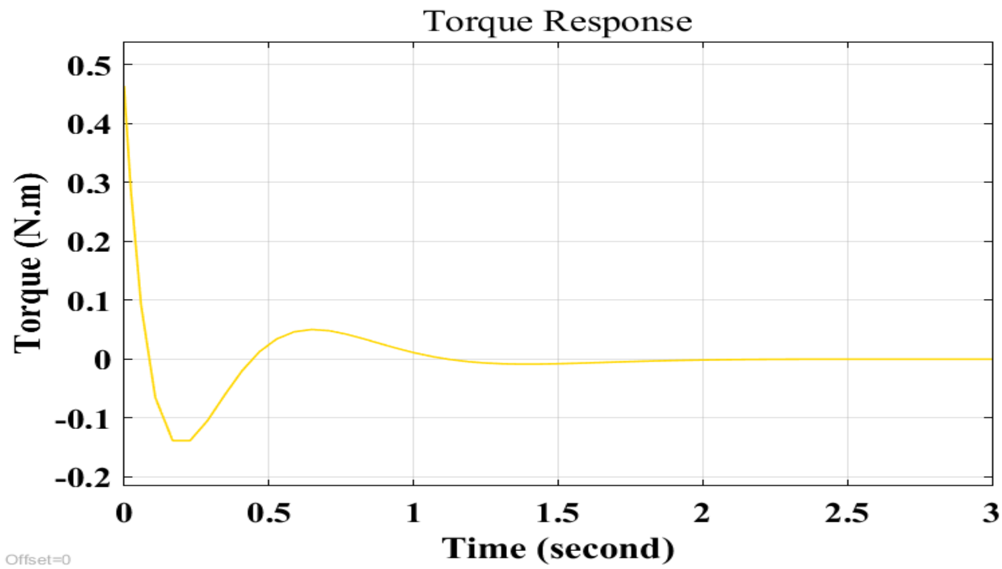

Figure 4-14 Torque scope is expected torque needed for non-linear system

The Figure 4-14, gives us approximate expected torque of the motor to be controlled to be used to build the system and it shows maximum torque at 0.138 (N.m) and overshoot 5%.

For power scope:

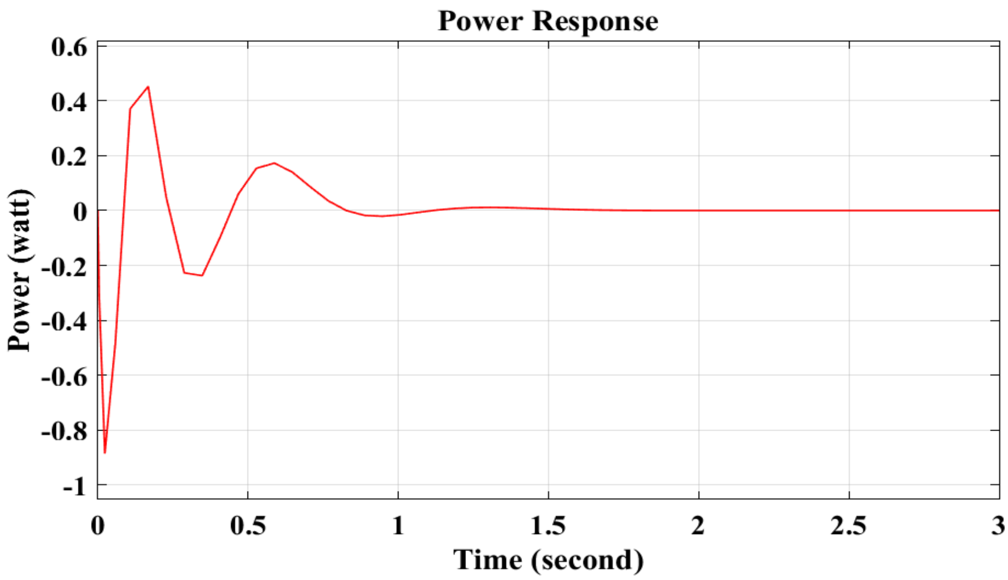

Figure 4-15 power scope is the expected power needed of the motor

The Figure 4-15, represents the product between $\dot{\varphi}$ and the needed expected torque of the motor, which gives us the power of motor needed, hence the maximum power is needed around 0.88 (W).

For velocity scope**:**

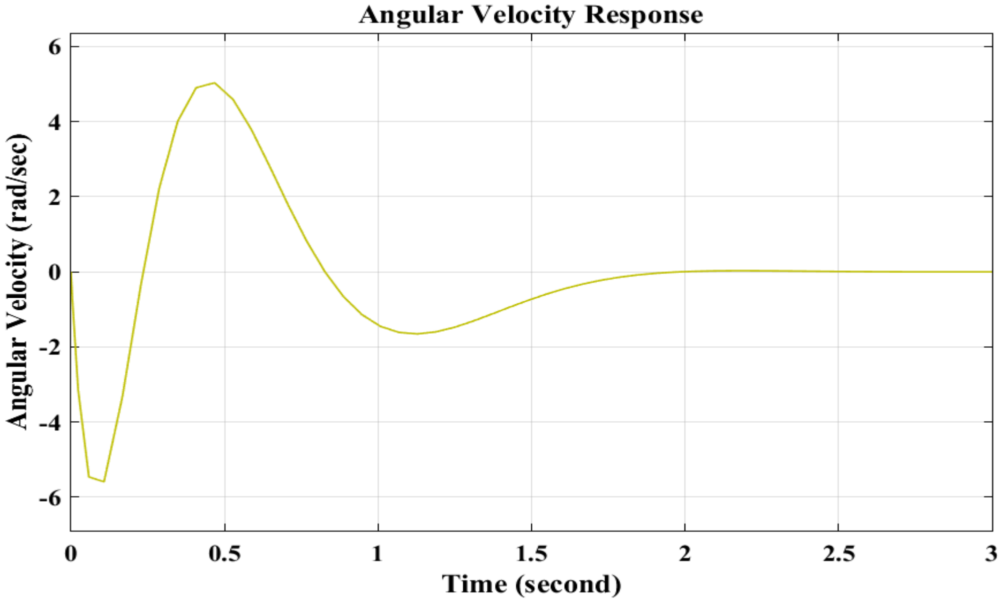

Figure 4-16 Velocity scope the expected velocity needed of the motor

The Figure 4-16, represents the velocity of our inverted pendulum and as we see here the maximum velocity that our system could reach is 5.6 (m/sec) and the settling time is around 1.88 (sec).

Conclusion:

The non-linearity of the system is weak since the response of the non-linear system and the linear system are identical, that's mean if we use the linear system or the non-linear system to build the system we will have same results.

4.4.4 Stability Robustness

In this section, we will discuss the effect of changing the parameters of the system, using the same controller that we used for building the system, and study the stability when changing in location of poles in each of the possible cases.

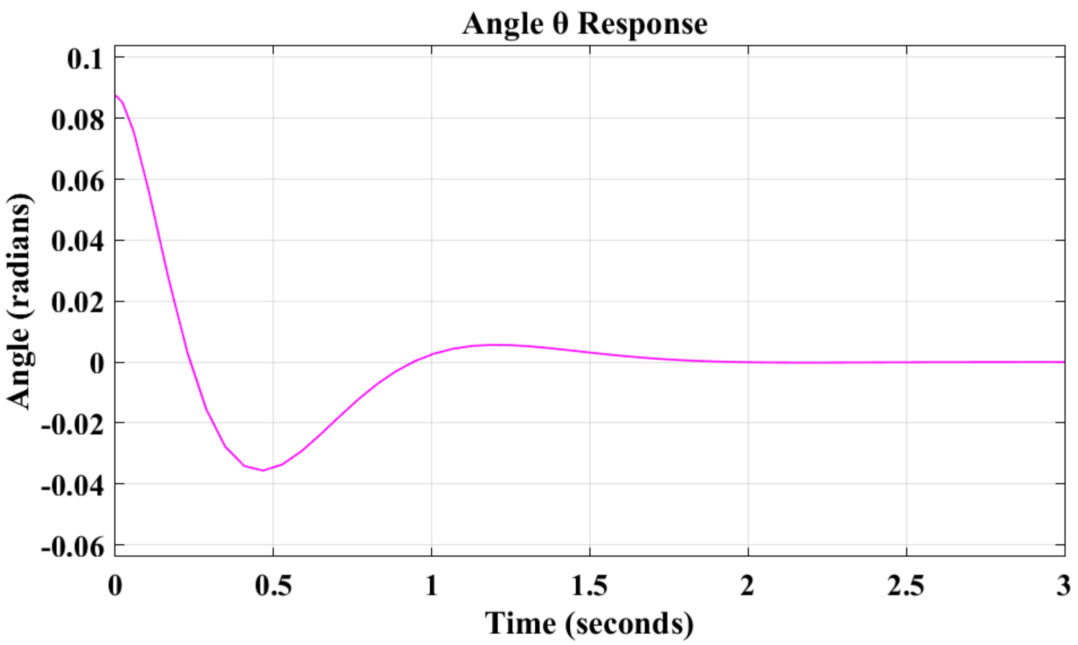

Figure 4-17 The response of angle with the system parameters (normal system)

In Figure 4-17, the response of the system in our chosen parameters (Figure 4-2), note that our system gives us the requirement's that included in building the system, using "$damp$" code on MATLAB we have;

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-4.00e+00 + 3.00e+00i	8.00e-01	5.00e+00	2.50e-01
-4.00e+00 - 3.00e+00i	8.00e-01	5.00e+00	2.50e-01
-4.50e+00 + 2.18e+00i	9.00e-01	5.00e+00	2.22e-01
-4.50e+00 - 2.18e+00i	9.00e-01	5.00e+00	2.22e-01

And for the location on S-plane we note in Figure 4-18 that the location of poles satisfies the required location of them for the controller that we build that have gain;

$$K = [-5.3181 \quad -0.2375 \quad -0.0590 \quad -0.0401]$$

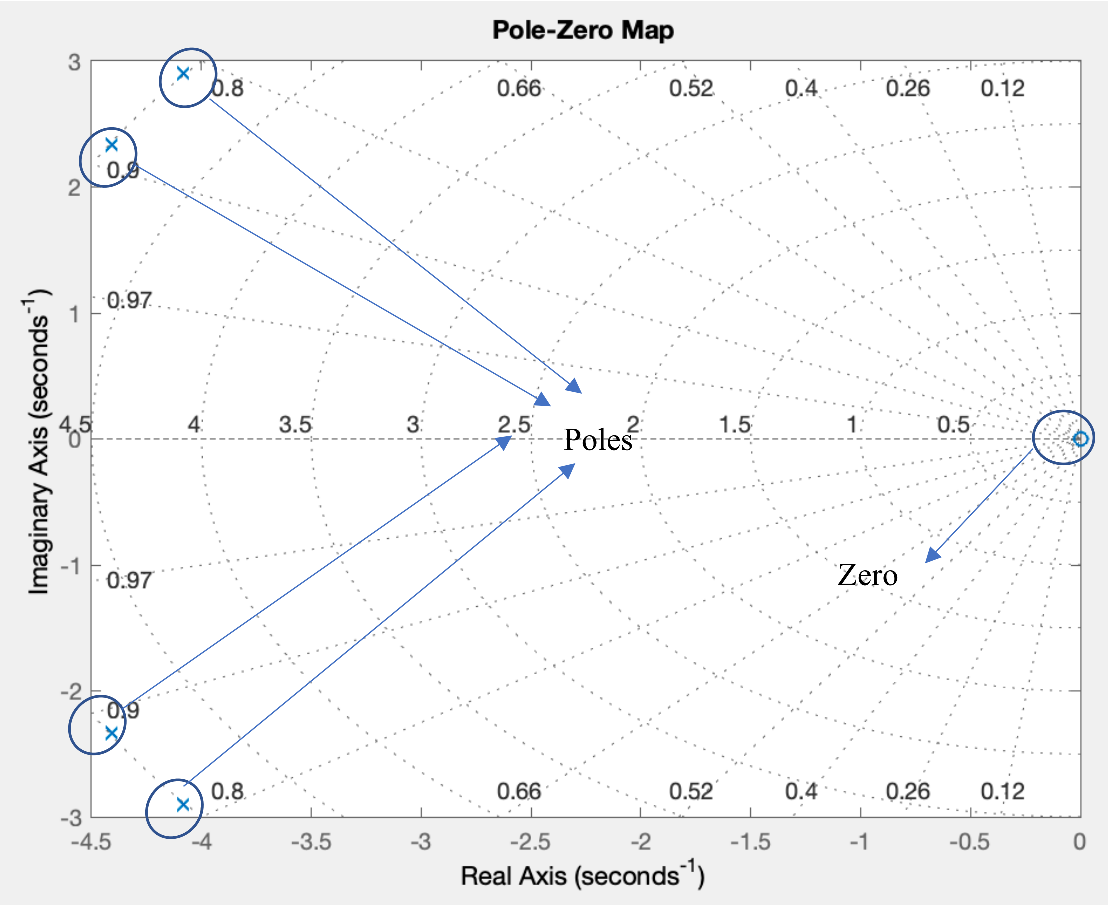

Figure 4-18 Poles and zero location for closed loop system

now we will study the effect of changing parameters in such a different of cases as following:

1. Swinging arm length l_1 →between (0.1:0.4) (m)
2. Pendulum arm length l_2 → between (0.1:0.7) (m)
3. Swinging arm weight m_1 → between (0.05:1) (kg)
4. Pendulum arm weight m_2 → between (0.15:0.6) (kg)
5. Hanged mass weight m_3 → between (0:0.5) (kg)

Case 1: l_1 →between (0.1:0.4) (m);

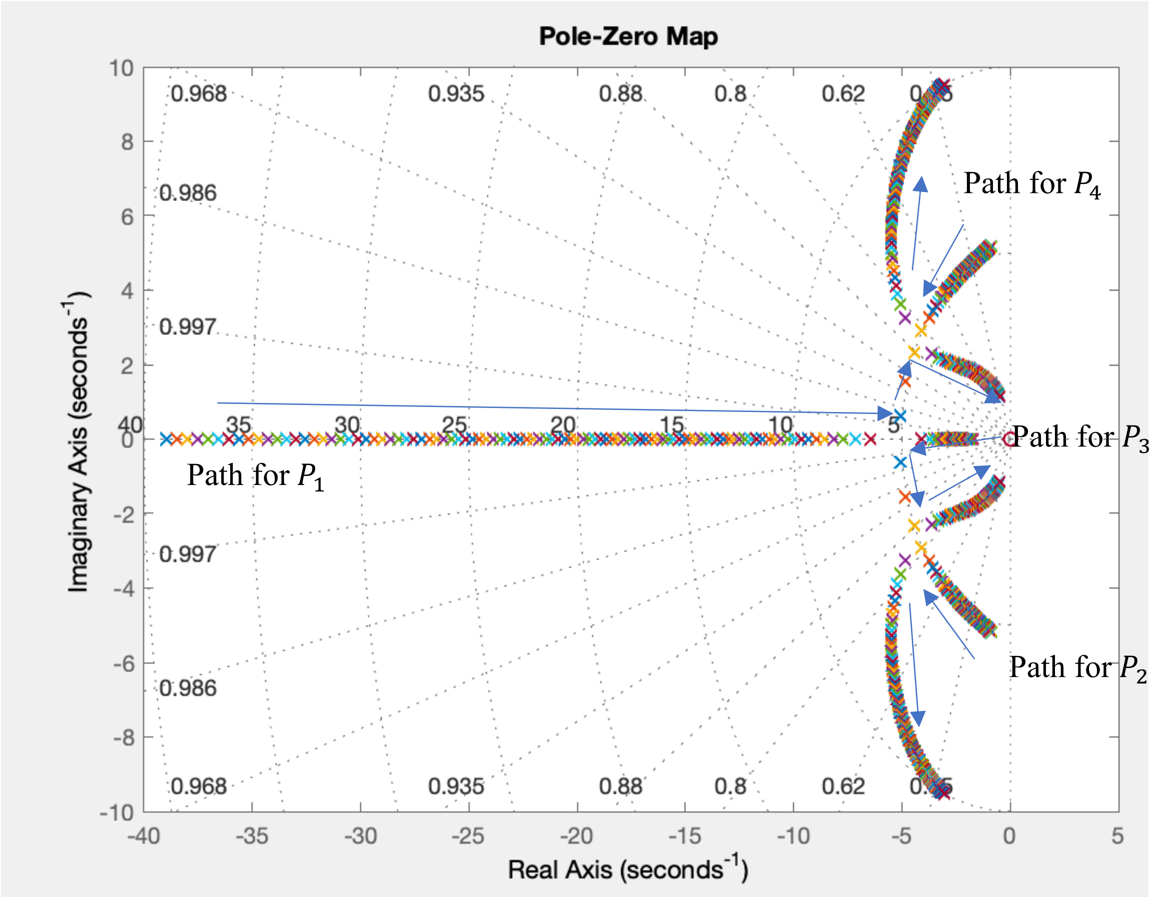

Figure 4-19 Poles paths as l_1 changing

In Figure 4-19, for P_1 path we note that $P_1$was on the very stable location on s-plane as we increased on the length P_1 more closer to the unstable location, for P_2we see that P_2 was close to the unstable location as we increased in length until 0.2 m P_2 is going far but if we increased more than that P_2 return to be closer to the unstable location, for P_3 the pole is close to the origin as we increased on length until 0.2 m P_3 is return back to be closer to the unstable region, for P_4 is the same for P_2, from tha we note that the dominant poles of the system are close to the imaginary axis the system, the effect of changing the length is critical that's mean any small change in the length of swinging arm causes big change in the system, note that we can't increase the length more than $\frac{l_1}{2} = l_2$, length 0.2 as we can note perfect for this system it satisfies the requirements of the system, in case the swinging arm was shorter, the system needs more time to reach the final value because of the increasing in natural frequency, and more overshoot because the decreasing on the damping ratio of the dominant poles of the system, and the system is in unstable region in the S-plane, for shorter l_1 the system marginally stable, can't handle with 30-degree disturbances.

Case 2: $l_2 \rightarrow$ between (0.1:0.7) (m):

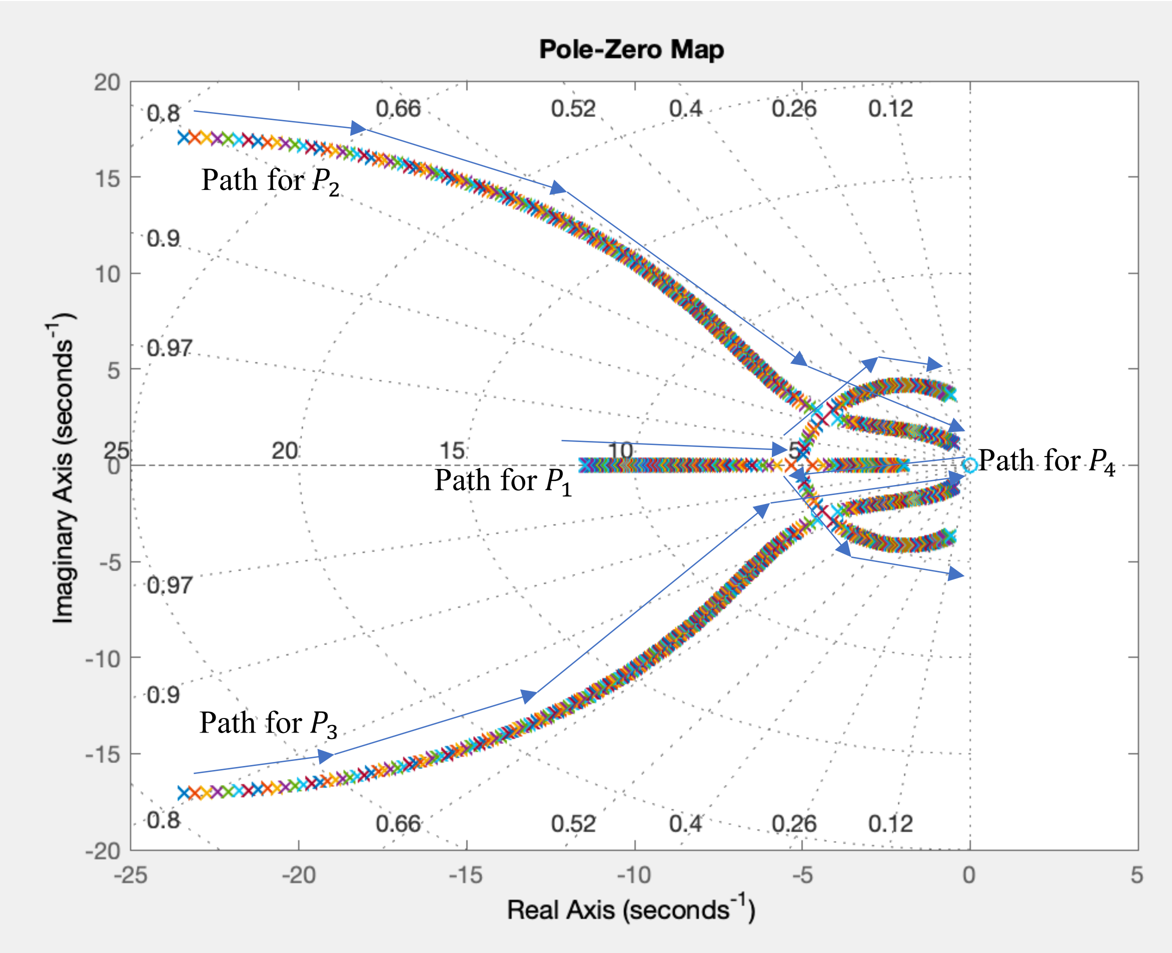

Figure 4-20 Poles paths as l_2 changing

In Figure 4-20, for P_2 path we note that the pole was on the very stable region as we increased on the length it's going to be closer from unstable region but if we increased more than 0.4 m the pole is return back to the stable region, for P_1 path if the length was small the pole will be far from unstable region but as we increased in length the pole is going to be closer to the unstable region, for P_3 and P_4 paths we note that the poles for small length is close to the unstable region, as we increased on the length the poles is going to be more stable but if we increased more than 0.4 m the poles (P_3&P_4) is return back to be closer to the unstable region, note we have that all location poles are in the stable side of S-plane, as we increased the length the dominant poles are going away from the imaginary axis until we reach 0.4, it seems that the system return to the imaginary axis as we increased more length is more closer to the imaginary axis, that's mean increasing more length causes something like critically stable that's maybe can't handle with big range with disturbances and need more energy to make the system stable, since the damping ratio is increased.

Case 3: $m_1 \rightarrow$ between (0.05:0.3) (kg):

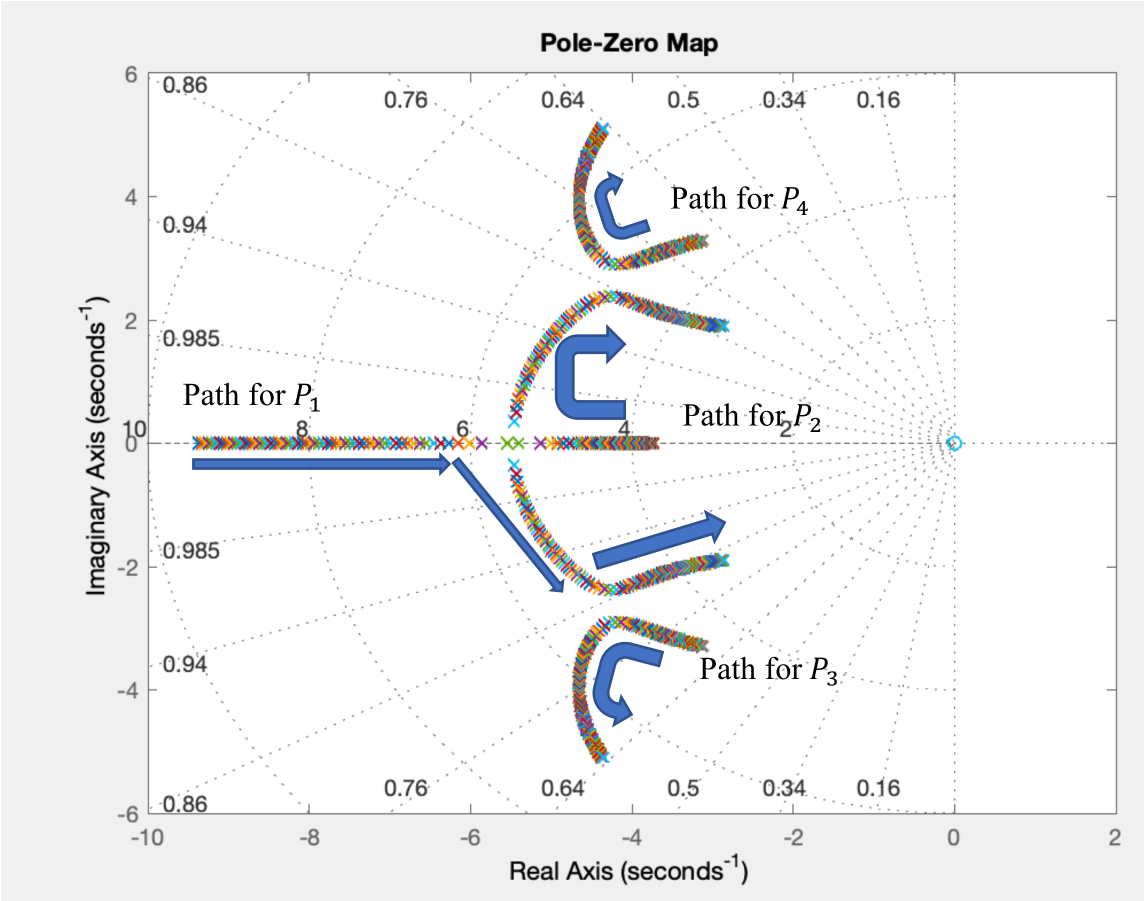

Figure 4-21 Poles paths as m_1 changing

From Figure 4-21, for P_1 path we see that if we have small mass P_1 will be at the stable region as we increased the mass P_1 will be more closer to the unstable region, for P_2 and P_3 paths we see that for the small mass the poles are in stable region as we increased the poles still in the safe region but as we increased the mass the damping ratio is decreased, for P_4 path as we increased the mass the pole going away to the stable region but if we increased more than 0.15 kg the pole start to increase the overshoot and decreasing in damping ratio, we note that the weight of the swinging arm makes the system faster to stabilize, but as we increased the weight the system is closer to the imaginary axis it means it's harder to be controlled for the same motor and torque.

Case 4: $m_2 \rightarrow$ between (0.15:0.6) (kg):

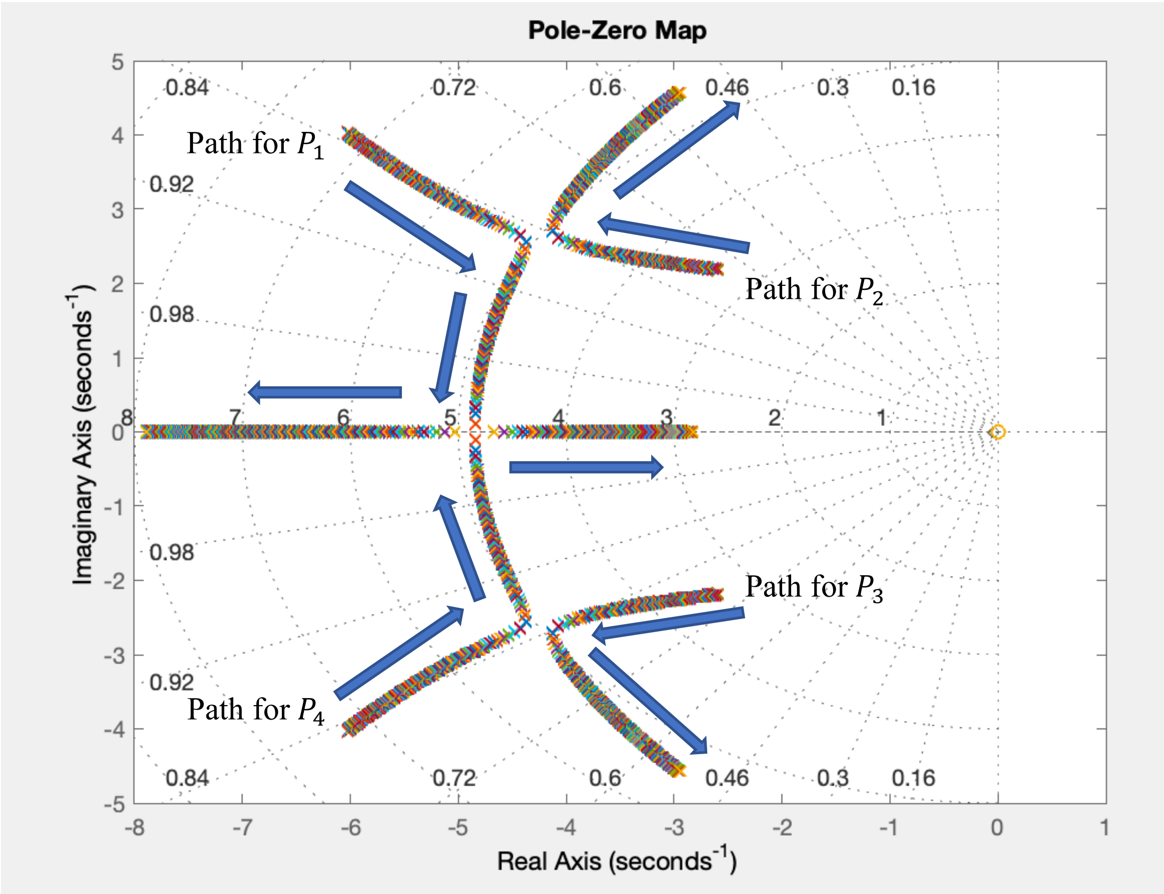

Figure 4-22 Poles paths as m_2 changing

In Figure 4-22,for P_2 path we see that for small weight the pole is in stable region but as we increased in weight the damping ratio increased and its more closer to the unstable region, for P_1 path we see that for small weight the pole is in stable region but as we increased in weight the damping ratio increased and the pole is going to away from unstable region, for P_3 and P_4 paths as we increasing the mass the poles is going away from the unstable region but the damping ratio is decreasing, in case the pendulum arm weight was small the system will be faster as we increased the weight, the system speed is decreased and the overshoot is decreased, but if we increased the mass more than 0.4 kg the system speed is almost the same but the system overshoot is increased, the weight of the arm will affect the controller and the system maybe can’t handle with this case for chosen torque of motor.

Case 5: $m_3 \rightarrow$ between (0:0.3) (kg):

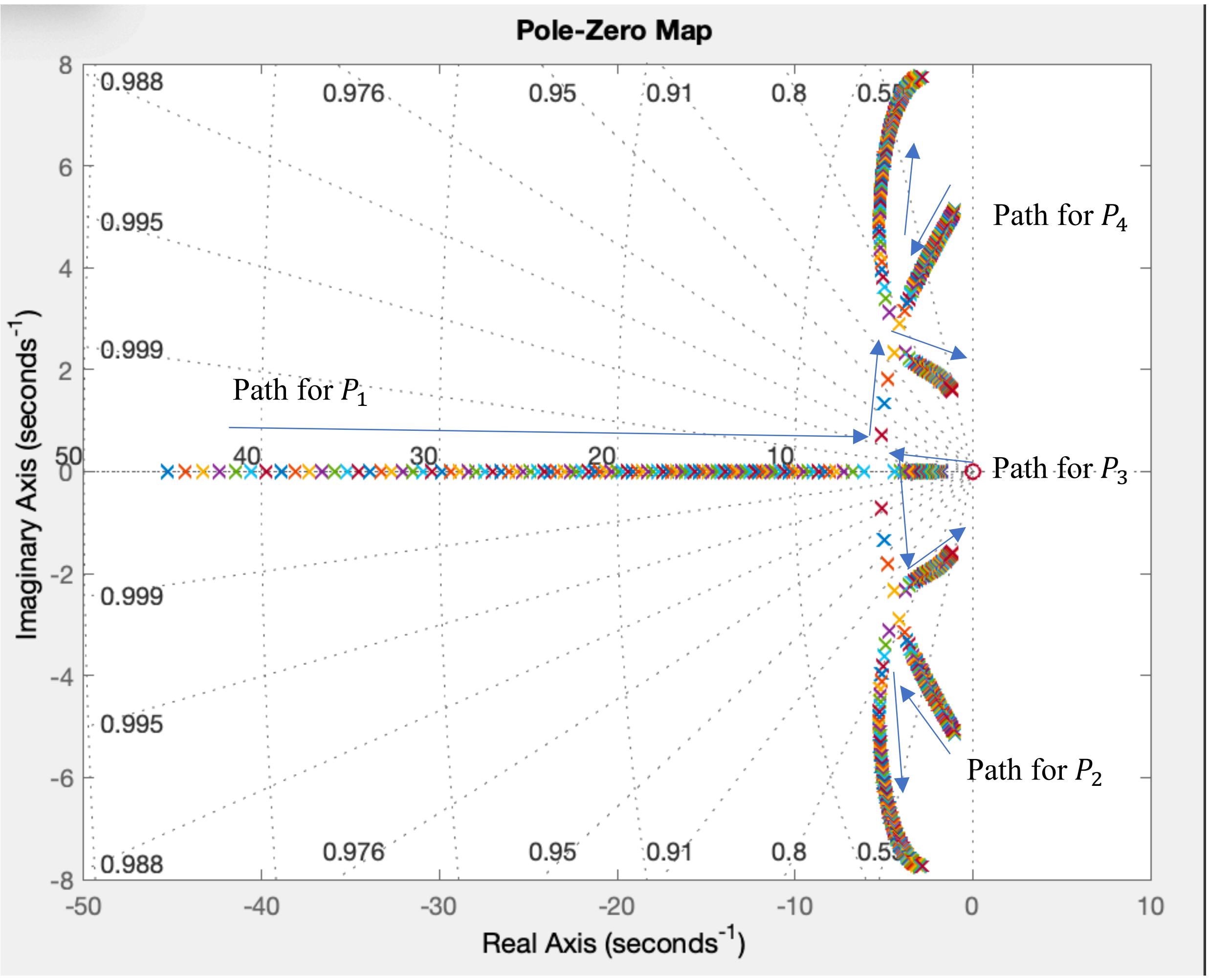

Figure 4-23 Poles paths as m_3 changing

In Figure 4-23, for P_1 path we see that P_1 was in very stable region with high damping but as we increased the mass the pole is going to be closer to unstable region and the damping ratio is decreased but if we increased more than 0.1 kg the pole damping ratio will be increased again, for P_4 path the pole is close to the unstable region and have high damping ratio but if we increased the mass P_4 is going to be stable but increasing more than 0.1 kg will cause more decreasing on damping ratio and closer to the unstable region, for P_2&P_3 paths the poles closer to the unstable region and have very small damping increasing until 0.1 kg will decrease the damping ratio be more stable, more than 0.1 kg will cause decreasing in damping and return to be closer to unstable region, if we don't have any hanged mass on the swinging arm the dominant poles are close to imaginary axis, but as we increased the mass the system is almost go away from the imaginary axis but if we increased the mass more than 0.2 kg its start again to close to the imaginary axis cause increasing in the overshoot of the system.

Chapter 5: Selection for electrical components

5.1 Introduction

5.2 Electrical parts

5.3 Arduino

5.1 Introduction

This chapter introduces the design of electrical components in the subsystems, explaining the motors design, electric driver, and the encoders.

5.2 Electrical parts

The inverted pendulum electrical parts include motor and interfacing circuits that connect these encoders with the system controller. The inverted pendulum electrical parts are listed as follow:

5.2.1 Dc Motor

For this project we need one motor for that swinging arm in the tip of the pendulum arm, from the simulation result we conclude the specifications of the motor like the speed (rad/sec) see Figure 4-18 and the torque (N.m) see Figure 4-16.
For the selection we need to take factor of safety for the specifications we assume the initial condition for the pendulum angle (pi/6), for this we have the maximum torque and speed that will needed to achieve the stabilization.

For the pervious specification was found a motor for that specifications and this motor was chosen for achieved the goal and it shows as follows:

voltage		no load		At maximum efficiency				stall		reducer	
Workable	Rated	Speed	Current	Speed	Current	Torque	Output	Torque	Current	Ratio	Size
6-12V	12V	250	680	200	3.2	16	42	50	15.5	56	24

Figure 5-1 Motor workspace

1-1 Rated Voltage: **12v.**
1-2 Direction of Rotation: **CW** when viewed from output shaft side
1-3Operating Temperature and Humidity:Temperature range of **-10C~+50C**,Humidity range of **30%~80%**
1-4Storage Temperature:Temperature range of **-20C~+60C**
2.Measuring Conditions:
2-1Motor Position:To be place t horizontally when measuring
2-2Power Supply: Regulated DC power supply
--No load speed: **17-2240rpm**.
--Robotics, Small Appliance, Fanner, Electric Curtain;
--Medical Pump, Surgery Tools, Medical Stirrer, centrifugal Machine; Electric Valve, Actuator, medical device
--Electric motor,low noise,low current and no spark.

Figure 5-2 DC motor specifications

It shows No-load (speed and current) and the maximum efficiency (speed, current, torque and the output power).
Since we are using DC motor with a gear, the gear ratio is 56.
We made a system identification to get the parameters that does not exist in the website that we purchased so, we used oscilloscope that exist in the lab as a way to identify our DC motor parameters, see the figure below:

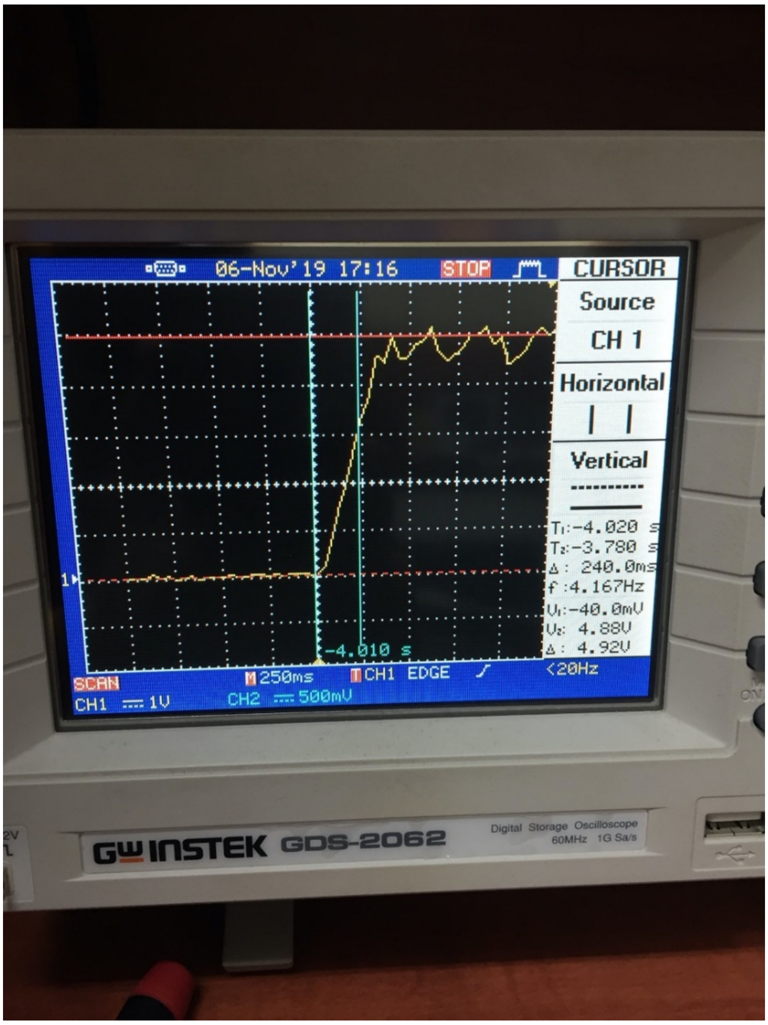

Figure 5-3 DC motor response

Using the knowledge that we learned from the control book [10], using section 4.3 for first-order systems, we built the transfer function for the response of our DC motor using the figure below:

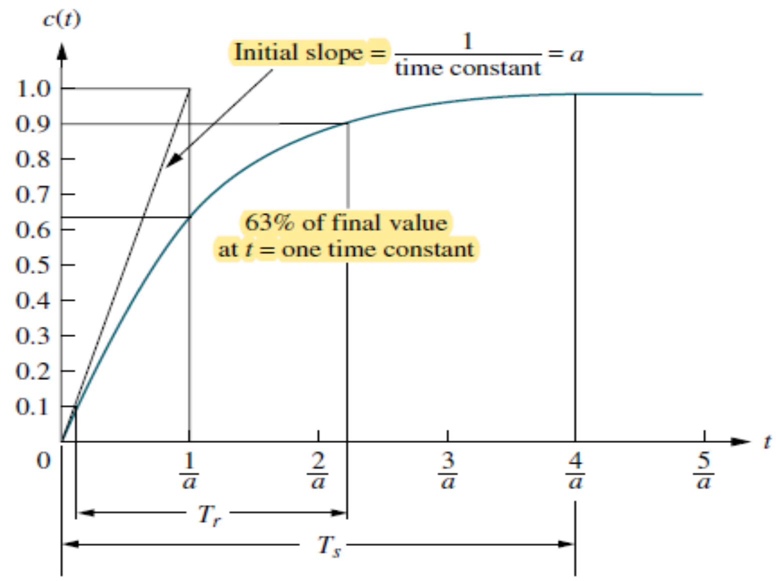

Figure 5-4 First order response

So, from the oscilloscope we got the final value, it is equal 4.92 Volt, we will calculate τ at 0.63 as follow:

$\tau = 0.63$ * the final value

$\tau = 3.0996$

$\rightarrow$ The final value $= \frac{K}{a} = 4.92$

$\tau = \frac{1}{a} = 3.0996 \quad \rightarrow a = 0.323$

$\frac{K}{a} = 4.92 \quad \rightarrow K = 4.92*(0.323) = 1.587$

So, we can now write the transfer function for our DC motor without including the gear in our calculations, and this will help us to get the parameters for our DC motor.

$$G(s) = \frac{1.587}{s + 0.323} \tag{5-1}$$

We built the actual response using MATLAB Simulink as follows:

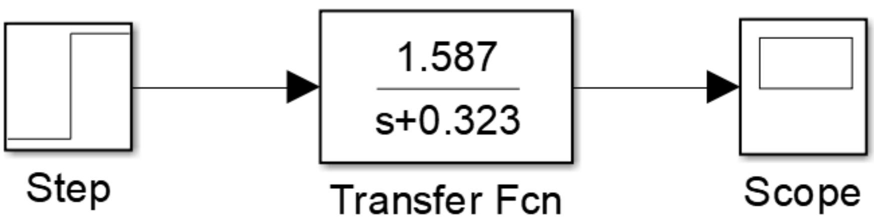

Figure 5-5 Motor Simulink model

The output of the scope with step input:

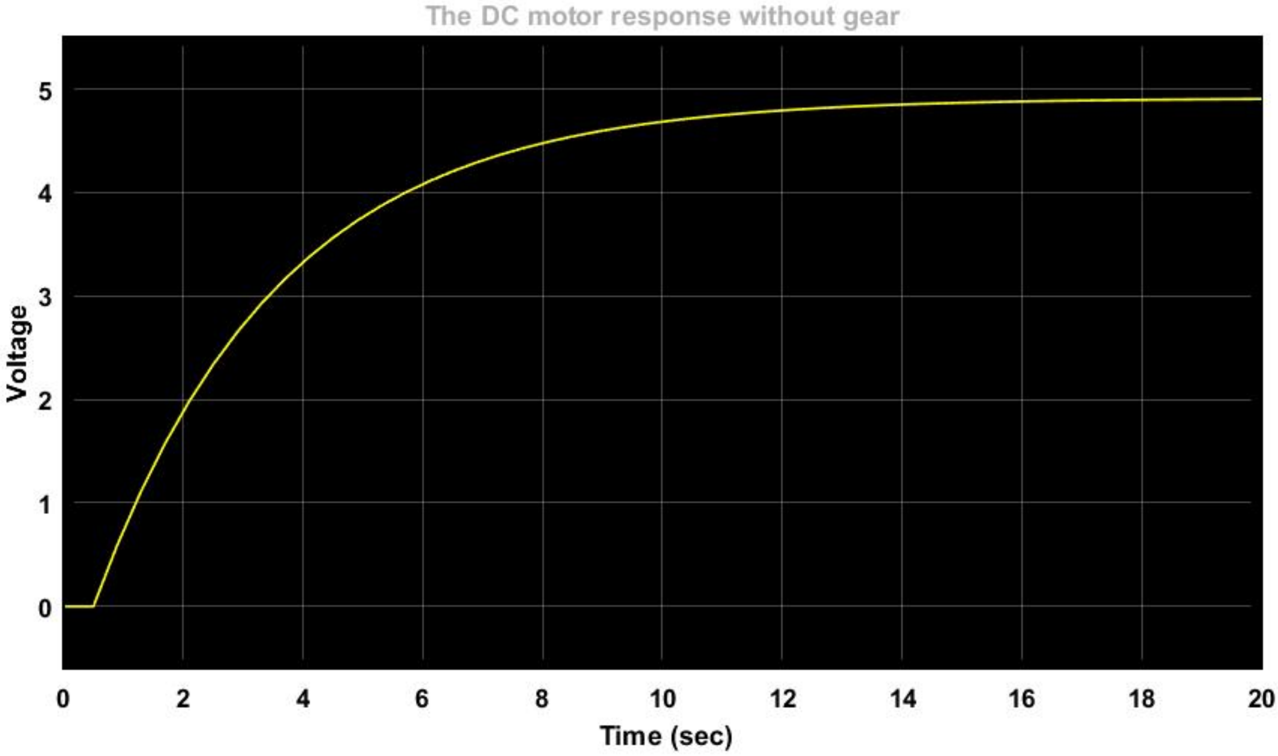

Figure 5-6 Motor response for step input

Using the previous method that described above, we will now produce the transfer function for our DC motor including the gear, this transfer function represents the whole DC motor.

$$G(s) = \frac{1.5867}{s + 11.2} \tag{5-2}$$

Using the transfer function (5-2), we designed PI controller for our DC motor to control the horizontal arm position, that will help us to stabilize the inverted pendulum (vertical rod).

The Figure below represents PI controller for our DC motor with gear as follow:

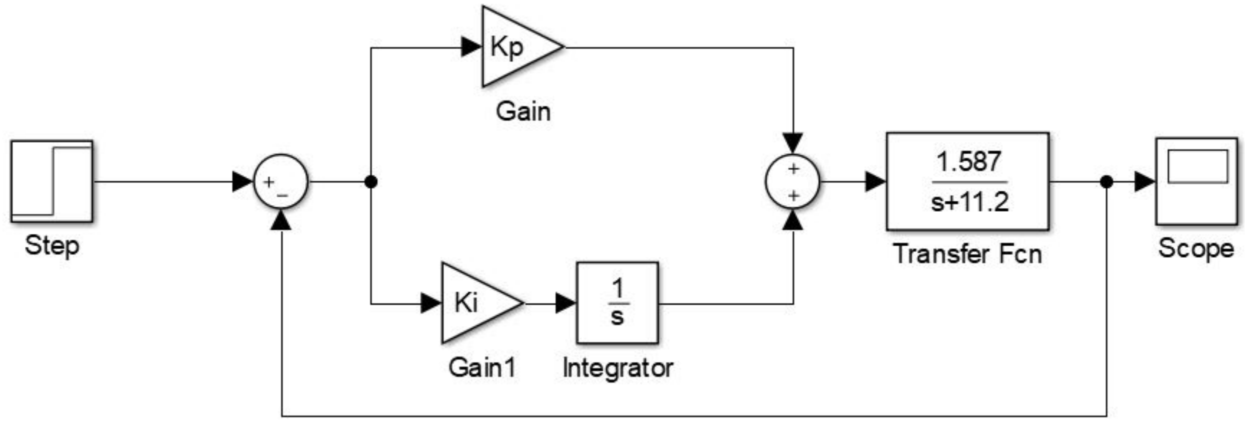

Figure 5-7 PI-controller Simulink

The output for the scope in the Figure 5-7 above as follows:

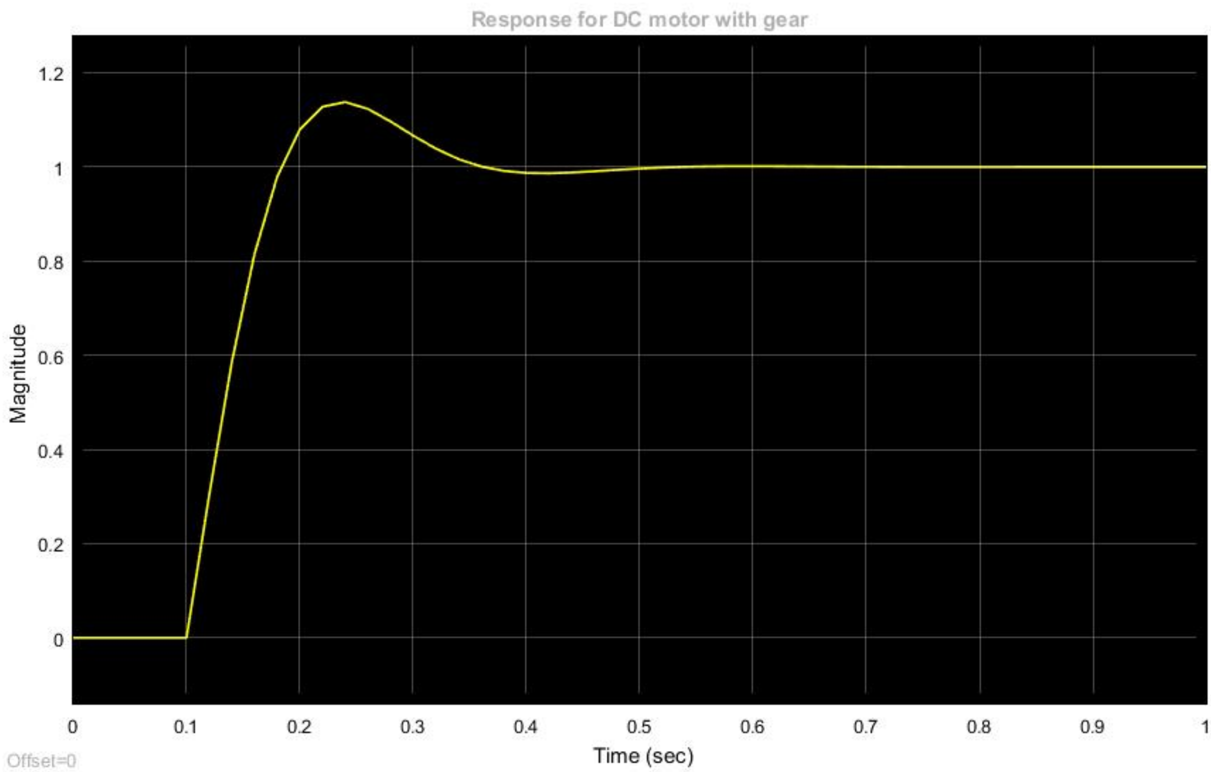

Figure 5-8 DC motor response with PI-controller

In the *Figure 5-8* the specifications for this response are that overshoot is around 10% with 0.3 settling time so, the gains are $K_p = 9.28$ and $K_i = 302$

5.2.2 Encoders:

For the project to have full states that needed to control the system two encoders will be needed first encoder will be connected with the swinging arm or with the shaft of the motor, for the previous shown Dc motor it has hall sensor (encoder), the other encoder will be coupling with the pendulum arm.

5.2.2.1 Hall Encoder:

For this encoder it will be attached with the motor as shown in Figure 5-9, this encoder as measuring the pulses using the Arduino gives (1920P/R), with frequency (0-1.37 KHz) this encoder will give the states of φ .

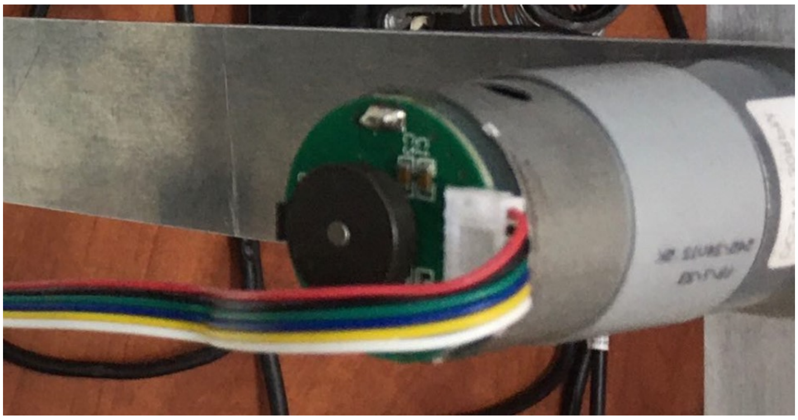

Figure 5-9 The hall sensor

5.2.2.2 **Pendulum Encoder:**

This encoder is incremental rotary encoder two phases gives (600P/R) for that total pulses (2400P/R), with frequency (0-20 KHz), NPN output collector for the wiring diagram (phase A: Green, phase B: White, V_{cc}: Red, 0V: Black). See Figure 5-10.

Figure 5-10 Incremental Encoder

5.3 Arduino:

This subsection will discuss the Arduino for the system, as this system have two encoders to read values, two interrupts will be needed to achieve the response and action that needed, for that Arduino Mega is best option to achieve this, It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with AC-to-DC adapter or battery to get started, and it suppled 5 Volt, see Figure 6-7.

5.4 Power supply:

Figure 5-11 Power supply

In Figure 5-11, we used this power supply to supply our electrical components, as follows the red +5Volt, yellow +12Volt, Black is the common.

Chapter 6: **Integration of the system**

6.1 Introduction

6.2 Hardware connections

6.3 Electrical connections

6.4 Arduino programming

6.1 Introduction

This chapter will discuss the integration of the hardware and software parts of the system, this chapter will contain the hardware parts, how its connected to each other, and the software, the code and any other needed programing.

6.2 Hardware connections:

This section discusses the integration of hardware of the system, and explain how its connected to each other.

6.2.1 Plate:

This subsection discusses the plate dimensions and specifications, the plate is made from wood, with dimension (57*27*2) cm.

6.2.2 Pendulum arm:

This subsection discusses the coupling of the pendulum arm with the incremental encoder as shown in the Figure 6-1.

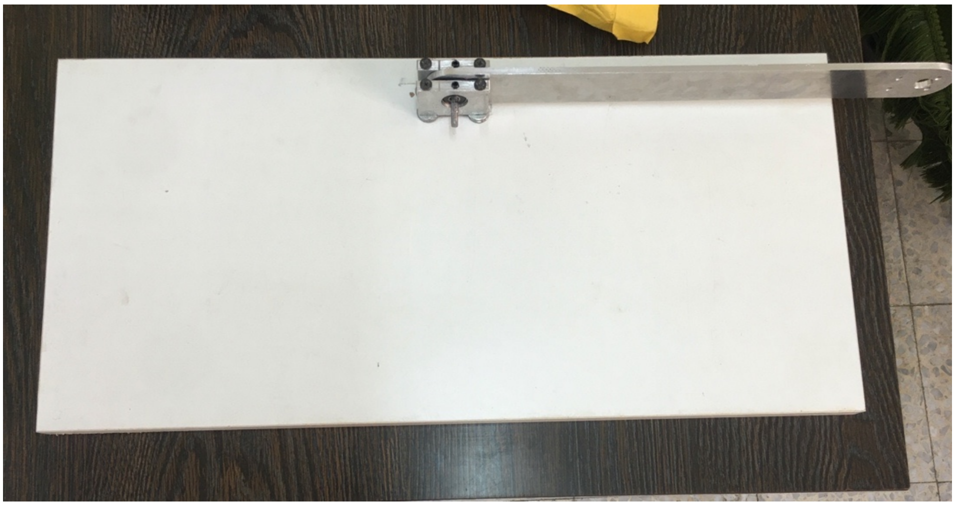

Figure 6-1 Pendulum Arm & Plate

The pendulum arm was coupling with bearing in both sides, and with shaft was holding as shown in the Figure 6-1, small shaft was been added to do coupling with the incremental encoder to decrease the pressure on encoder shaft.

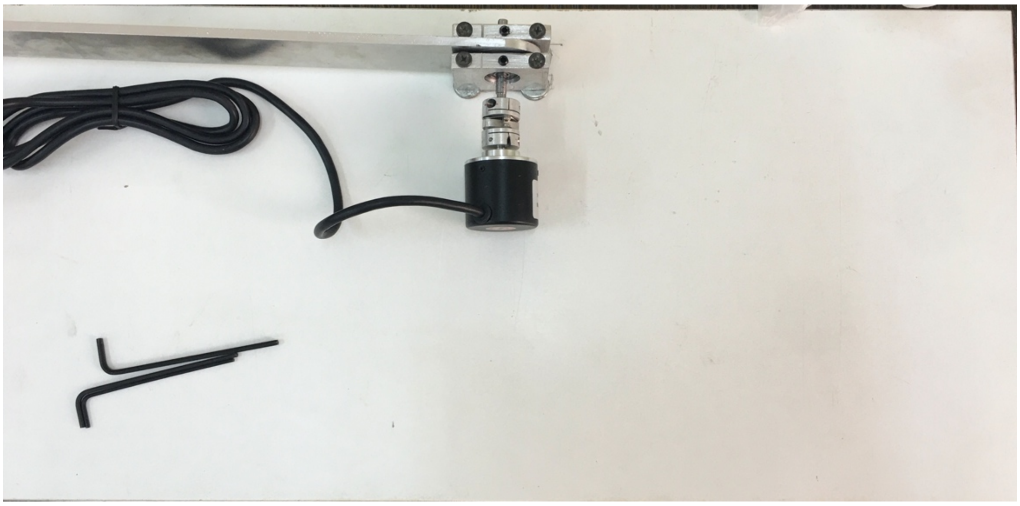

Figure 6-2 Encoder coupling

The Figure 6-2 is shown the encoder after been connected with coupling.

6.2.3 Swinging arm:

This subsection discusses the connection of the swinging arm with the shaft of the motor, and the hanged masses, see the Figure 6-3

Figure 6-3 Motor

The Figure 6-3 is shown the motor after been connected to the top of the pendulum arm.

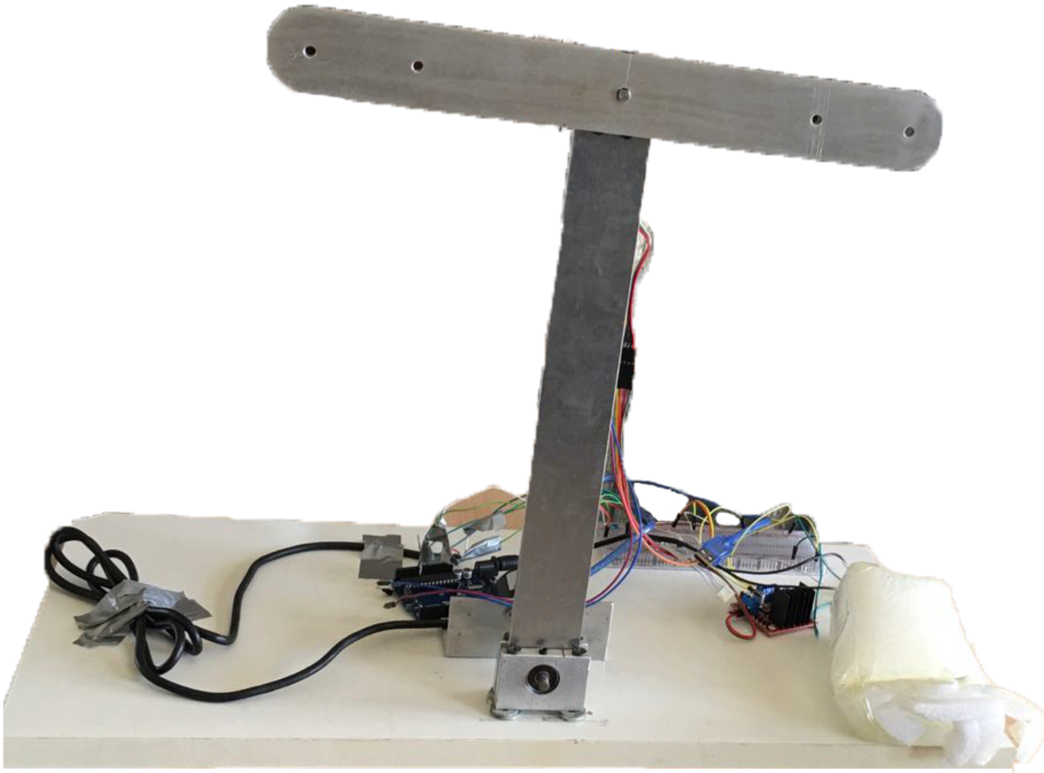

Figure 6-4 The system

The Figure 6-4 is shown the system after integration of the hardware, note that the swinging arm was chosen to be in different lengths for calibration the system in experiment purpose.

6.3 Electrical connections:

This section will discuss the wire connection for the electrical parts with each other, the Figure 6-5 is shown how the rotary encoder (that measure pendulum angle) must connected to take the correct values.

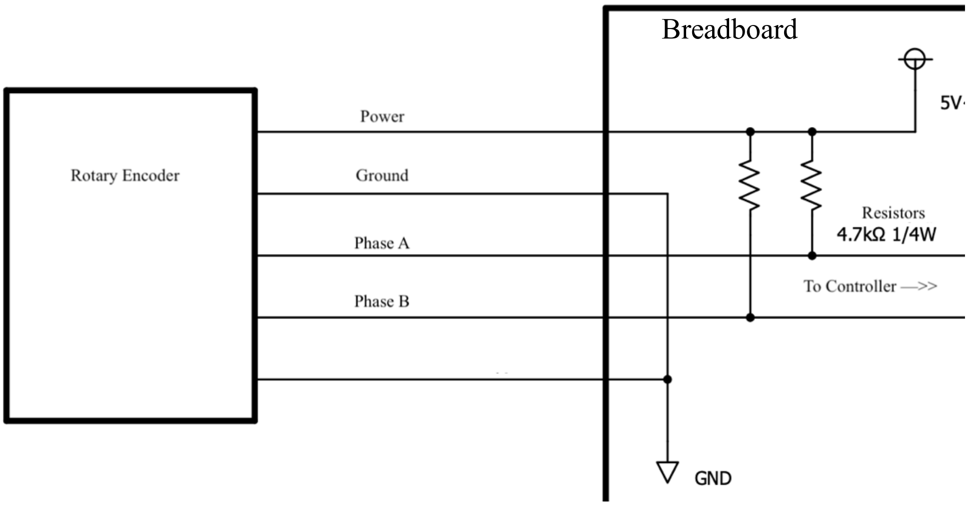

Figure 6-5 Incremental encoder wiring

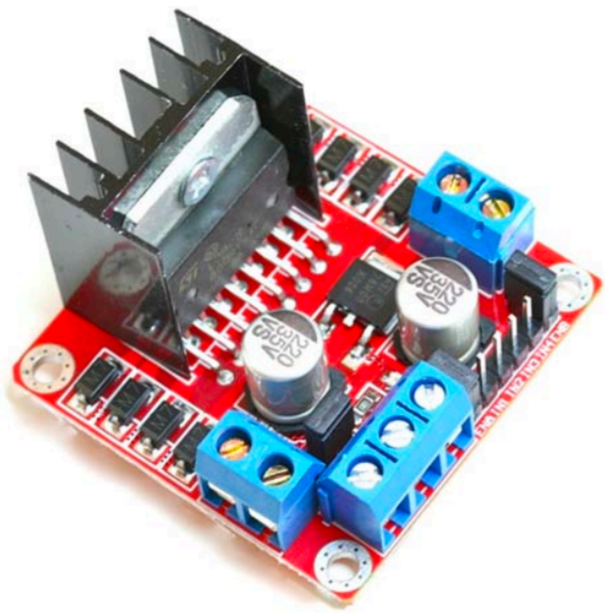

Figure 6-6 Driver L298N

The Figure 6-6 is shown the used driver (L298N) for controlling the motor, since the driver need three inputs from the controller (two direction & PWM control) and two outputs will go to the motor.

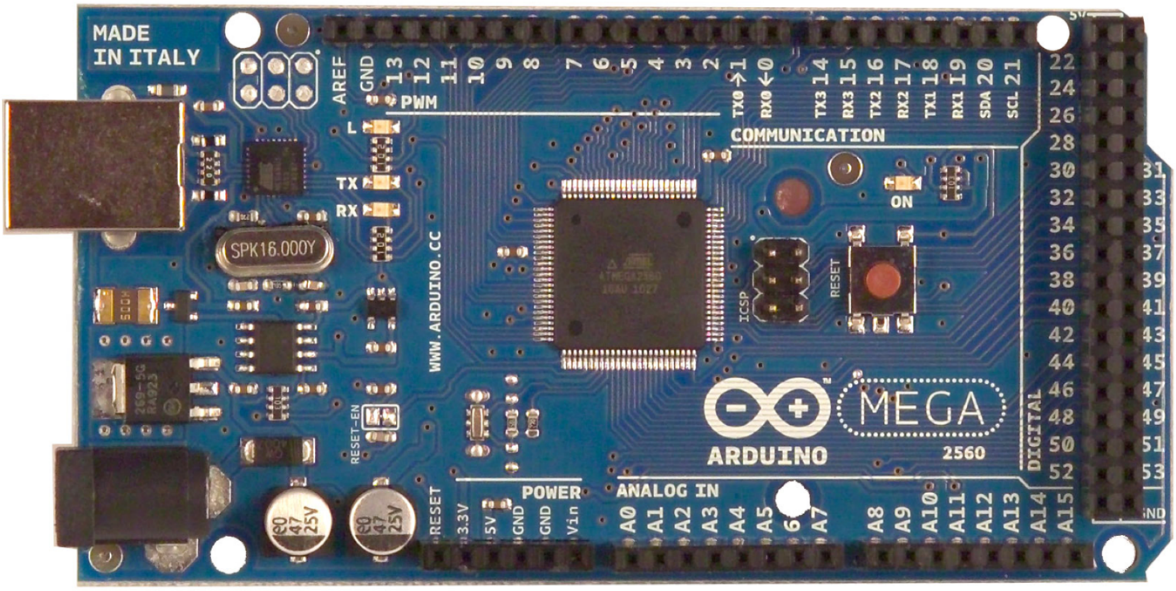

Figure 6-7 Arduino Mega Connection

For Figure 6-7 the connection with Arduino mega will be as follows:

Pin 10 connected to → PWM output to the driver.
Pins 8 and 9 connected to → Direction output to the driver.
Pin 20 and 21 connected to → Incremental encoder phases.
Pin 2 and 3 connected to → Hall sensor phases.

6.4 Arduino programming

This subsection will discuss the parts of the main code that used to achieved this project as blocks, the figure shown the basic main parts of the code, for this system the processing unit will check the state of pendulum angle, as the system starts at the perpendicular state it considered the angle is zero, then the system check any change in the angle and do the calculations to the output of the motor, then check the state of the pendulum angle in loop, and give different output each time depending on the changing on pendulum angle, see Figure 6-8.

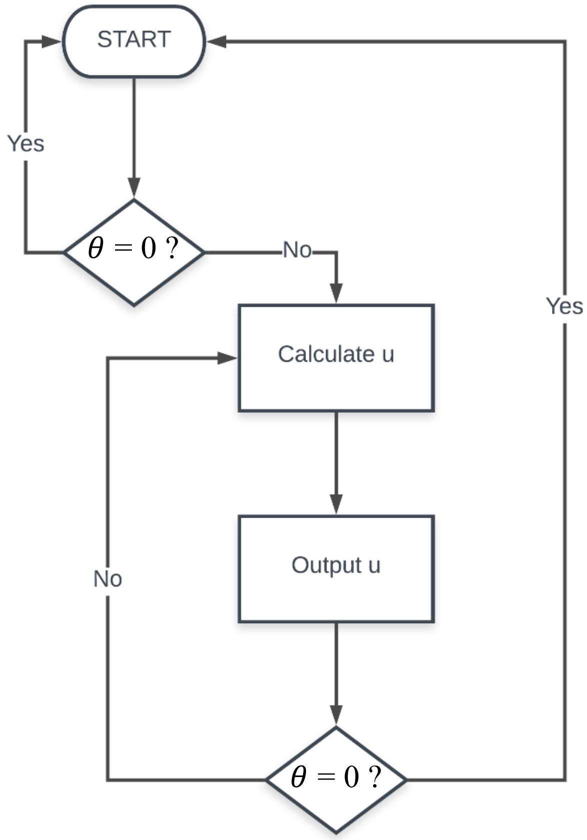

Figure 6-8 Processing flow chart

The following Figure represents the concept in how we programmed our system using **Arduino mega:**

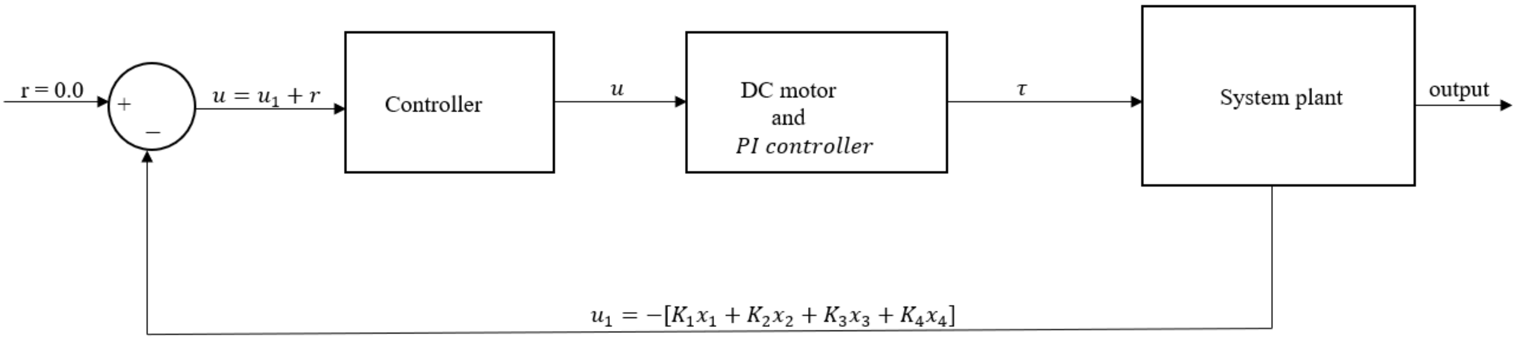

Figure 6-9 The concept that we programmed our system

Thus,
r: Is the reference and it is equal zero.
u_1: The value of the state feedback.
u: The required value of voltage that need to enter the DC motor to give a suitable torque.
τ: The suitable torque that required to drive the system to be stabilized.
e: Is the error between the value of the state feedback and the value of the DC motor (PI-Controller).
K_n: Represents the gains that needs to be added for the system.
x_n: Represents the states and it is measured by two encoders.

Chapter 7: **Experiments and results**

7.1 Introduction

7.2 Experiment # 1

7.3 Experiment # 2

7.4 Experiment # 3

7.1 Introduction

This chapter will discuss the experiment and results for the project, using Arduino as controller and for different lengths of swinging arm, the Figure 7-1 shown the different length that used in this project.

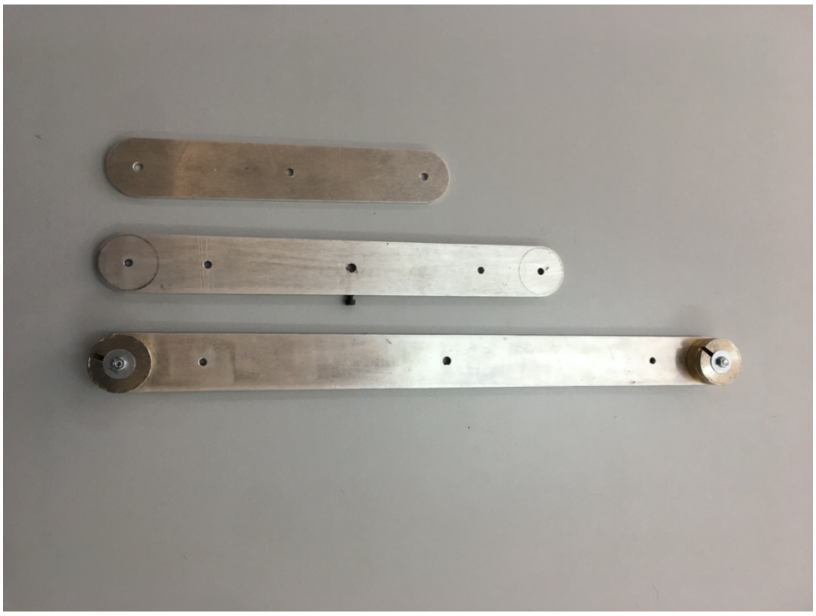

Figure 7-1 Swinging arm different lengths

7.2 Experiment # 1

For this section achievement the length of the swinging arm is 0.2 m as shown in Figure 7-2 for this length, we did three trials in different hanged mass for each side 100gr, 200gr and 250gr, for all the trials the system was failing because the length was not enough to provide a required torque to stabilize the pendulum arm, and the equilibrium point was hard to find for this length.

Figure 7-2 First length 0.2m

Results for trials in the first experiment:

Mass (gr)	Result	Comments
100	Fail	The torque not enough so, it needs more torque
200	Fail	The torque not enough so, it needs more torque
250	Fail	The torque not enough so, it needs more torque

7.3 Experiment # 2

For this section the length of swinging arm is 0.34 m as shown in Figure 7-3 for this length, we did five trials in different hanged mass for each side 50gr, 100gr, 150gr, 200gr, and 250gr, for the last mass the system succeed to be stabilized the pendulum at small disturbances (less than 5 degree) but, for rejection disturbances more than (5 degree) and less than (10 degree) needs more torque to stabilize the pendulum, and to avoid the damage for the motor, experiment three was achieved using new length taller than 0.34m .

Figure 7-3 Second length 0.34m

Results for trials in the second experiment:

Mass (gr)	Result	Comment
50	Fail	The torque not enough so, it needs more torque
100	Fail	The torque still not enough so, it needs more torque
150	Fail	The torque still not enough so, it needs more torque
200	Almost success	The torque cannot stabilize the pendulum very well so, it needs more torque
250	Success	The torque can stabilize the pendulum very well at small disturbances (less than 5 degree)

7.4 Experiment # 3

For this section the length of swinging arm is 0.5 m as shown in Figure 7-4 for this length, we did five trials in different hanged mass for each side 50gr, 100gr, 150gr, 200gr, and 250gr, for the mass 150gr the system stabilized itself at small disturbances and for adding more masses like 200gr and 250gr respectively, the system become more stable and reject less than (13 degree), and the equilibrium point was easy to find for this length.

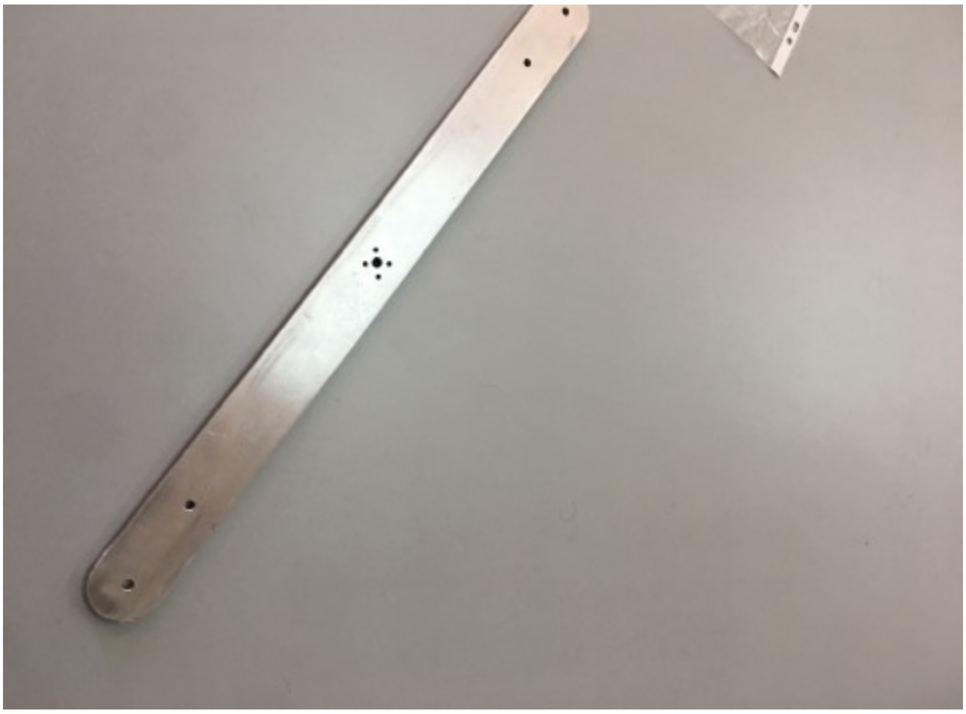

Figure 7-4 Third length 0.5m

Results for trials in the third experiment:

Mass (gr)	Result	Comment
50	Fail	The torque not enough so, it needs more torque
100	Almost success	The torque cannot stabilize the pendulum very well so, it needs more torque
150	Success	The torque can stabilize the pendulum very well at small disturbances (less than 5 degree)
200	Success	The torque can stabilize the pendulum very well at medium disturbances (less than 10 degree)
250	Success	The torque can stabilize the pendulum very well at large disturbances (less than 13 degree)

7.5 Summery

To achieve the goal of this project three experiments for different lengths and different masses were done to study the effect of the swinging arm length and weight on the controller and other parameters, for the short arm the system was unstable and cannot stabilize itself, and the equilibrium point was hard to find for this short length.
For the medium length the system at the fourth and the fifth mass almost succeed and the system stabilized itself and as the result we expect that if we add more masses at the same length it will hold the pendulum better but it will require more torque.
For the larger length the system at the first mass fail but in the other masses the system stabilized itself and as the result we expect that if we add more masses at the same length it will hold the pendulum better and it will not require high torque as the previous length (medium length), and the equilibrium point was easy to find for this length, since the damping ratio in the real system is too small the pendulum arm stabilize in settling time 2 seconds and stay for 3 seconds in the stable condition and failing if the encoder didn't detect the changing in the pendulum angle

Chapter 8: **Conclusions and recommendations**

8.1 Introduction

8.2 Conclusions

8.3 Recommendations

8.1 Introduction

This chapter will discuss the conclusions that been concluded in implementation for this project and recommendations for future work on this project.

8.2 Conclusions

After implementation this project, project team conclude some conclusions as follows:

- The length and the weight of swinging arm have main effect on the system.
- We can control our system smoothly and easily whenever the length is large, the swing arm gives more stability to our pendulum.
- The motor selection depends on the swinging arm weight and hanged masses that effects the total torque applying on the shaft motor.

8.3 Recommendations

After implementation this project, project team was outputting some recommendations for future work as follows:

- Firstly, for mathematical modeling for the system derive the equations of the system as function of the voltage the input of the system must be voltage.
- Secondly, for the pendulum arm, friction must add to the coupling with encoder to decrease the speed of the system and the speed for the selection motor.
- Thirdly, add more stability to the system by adding two springs in both sides for the pendulum therefore the system trying to stabilize itself without any failing.
- Lastly, trying to implement this project using DAQ as a processing unit.

References

[1] Mathworks, “MATLAB - Mathworks - MATLAB & Simulink,” *Www.Mathworks.Com*. 2016.

[2] K. Furuta, M. Yamakita, and S. Kobayashi, “Swing-up Control of Inverted Pendulum Using Pseudo-State Feedback,” *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, 2007.

[3] A. Rybovic, M. Priecinsky, and M. Paskala, “Control of the inverted pendulum using state feedback control,” *Proc. 9th Int. Conf. ELEKTRO 2012*, pp. 145–148, 2012.

[4] I. Hroub, “inverted pendulum,” no. June, 2014.

[5] SolidWorks, “3D CAD Design Software,” *Dassault Systèmes*, 2019. .

[6] S. Widnall, “Lecture L20 - Energy Methods : Lagrange ’ s Equations Derivation of Lagrange ’ s Equations in Cartesian Coordinates,” *Energy*, 2009.

[7] M. Huebla, “Non-holonomic Differential Drive Mobile Robot Control & Design : Critical Dynamics and Coupling Constraints,” *intoxicacion as frecuentes y sus Princ. factores influyentes en niños atendidos en el Serv. Pediatr. del Hosp. Prov. Gen. docente riobamba periodo enero-agosto del 2013*, 2013.

[8] B. Joshi, R. Shrestha, and R. Chaudhary, “Modeling , Simulation and Implementation of Brushed DC Motor Speed Control Using Optical Incremental Encoder Feedback DC Motor Model Simulink Modeling , Simulation and Parameter estimation,” *Proceeding IOE Grad. Conf.*, 2014.

[9] D. Baez-Lopez, “Simulink,” in *MATLAB with Applications to Engineering, Physics and Finance*, 2009.

[10] N. S. Nise, *Control Systems Engineering, 6th Edition*. 2012.