بسم الله الرحمن الرحيم

PPU College Of Engineering and Technology

The Home of Competent Engineers and Researchers

Mechanical Engineering Department

Graduation Project

Linear and non-linear control for an Inverted Pendulum

Applied to model "ECP 505"

Project Team

Ammar Halaiqa

Ibrahim Hroub

Project Supervisor

Dr. Yousef M. Alsweiti

Hebron-Palestine

June 2014

Palestine Polytechnic University

(PPU)

Hebron – Palestine

Linear and non-linear control for an Inverted Pendulum

Applied to model “ECP 505”

Project Team

Ammar Halaiqa Ibrahim Hroub

According to the project supervisor and according to agreement of testing committee members, this project is submitted to Department of Mechanical Engineering at college of engineering and technology in partial fulfillment of the requirements of the Bachelor's degree.

Supervisor Signature

…………………………………………

Committee Member Signature

……………… …………………. ..……………

Department Head Signature

..……………………………

Dedication

" وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ"

We dedicate this project to our great parents; we could never have done this without your faith, support and ceaseless encouragement. Thank you for all the unconditional love, guidance, and support that you have always given us. Thank you for believing in us, thanks for everything.

Acknowledgment

We would like to express our deep gratitude to our supervisor Dr. Yousef M. Alsweiti for his hard work, patience and guidance and for his encouragement throughout the preparation of this project. He inspired us greatly to work in this project. His Knowledge and endless support was a great asset from which we learned plenty.

Special thanks to Eng. Hussain Amro for supporting us from the first step of the project.

Special thanks to Eng. Abdalkarim Mohtasib for his support.

Finally, we would like to thank the staff of mechanical engineering department for helping us in every possible way to complete this project.

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 nonlinearities. The “Inverted Pendulum ECP model 505” which exist in the control lab at Palestine Polytechnic University (PPU), consists of a horizontal sliding rod and vertical ("pendulum") rod. The horizontal rod is connected to electrical motor through rack and pinion mechanism so it steers left or right to balance and control the position of the vertical rod. The controllers designed and simulated using MATLAB and Simulink. The aim of this project is to stabilize the Inverted Pendulum at its inverted position or to track any other position within the physical limits of the device, such that the position is controlled quickly and accurately so that the pendulum is always be at that position during such movements. However, Simulation of dynamics of a robotic arm and model of human standing still are some applications of an inverted pendulum.

Table of Contents

List of Figures

List of tables

List of Symbols

Symbol	Definition	SI-Unit
x	Displacement of the sliding rod	m
θ	Angle of the pendulum rod	rad
m_1	Mass of the complete sliding rod including all attached elements	kg
m_2	Mass of the complete moving assembly minus m1	kg
J_1	Polar moment of inertias around the center of gravities of the sliding link	$kg.m^2$
J_2	Polar moment of inertias around the center of gravities of the pendulum rod	$kg.m^2$
α	dynamic angle between pendulum rod an center of gravity of slider	rad
l_{m1}	Position of center of gravity for m_1	m
l_c	Position of center of gravity for m_2	m
L	Position of disturbance force	m
$F(t)$	The force driving the slider	N
$P(t)$	Disturbance force	N
f_1	Viscous friction on joint (O)	$N.m.s/rad$
f_2	Viscous friction between slider and the joint A	$N.m.s/rad$
T	The total kinetic energy of the system	$Joule$
U	The total potential energy of the system	$Joule$
R	The total energy loses from the system due to viscous damping	$Joule$

Dedication

" وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ"

We dedicate this project to our great parents; we could never have done this without your faith, support and ceaseless encouragement. Thank you for all the unconditional love, guidance, and support that you have always given us. Thank you for believing in us, thanks for everything.

Acknowledgment

We would like to express our deep gratitude to our supervisor Dr. Yousef M. Alsweiti for his hard work, patience and guidance and for his encouragement throughout the preparation of this project. He inspired us greatly to work in this project. His Knowledge and endless support was a great asset from which we learned plenty.

Special thanks to Eng. Hussain Amro for supporting us from the first step of the project.

Special thanks to Eng. Abdalkarim Mohtasib for his support.

Finally, we would like to thank the staff of mechanical engineering department for helping us in every possible way to complete this project.

1 Introduction

1.1 An Overview

The inverted pendulum (IP) represents a challenging problem in control and it has been widely used to investigate and develop new control strategies that can effectively deal with nonlinearities. The ECP model 505 Inverted Pendulum system consists of a horizontal sliding rod and vertical ("pendulum") rod. The horizontal rod is connected to electrical motor through rack and pinion mechanism so it steers left or right to balance and control the position of the vertical rod. The mechanism is open loop unstable (right half plane pole) and non-minimum phase (right half plane zero), with high nonlinear dynamics, and with some state that are not directly measurable. As a result feedback control is essential for stability, and the structure of the controller must be selected carefully due the non-minimum phase characteristics.

The application of IP ranges widely like:

- Simulation of dynamics of a ROBOTIC arm. The Inverted Pendulum problem resembles the control systems that exist in robotic arms.
- Model of human standing still**.** The inverted pendulum is widely accepted as an adequate model of a human standing still (quiet standing).
- Space rocket guidance systems.

1.2 Recognition of the need

To operate and control the ECP 505 model “Inverted pendulum”, which belong to Computer Control Lab at Palestine Polytechnic University (PPU), by applying several control theories. And to create some experiments that help students to apply what they learned in control courses, such as:

- Root locus and frequency domain techniques.

- State feedback control.
- Digital control methods.
- Optimal control methods.
- Nonlinear control methods.
- Intelligent control techniques.
- Embedded systems.

Such control methods can be tested and compare via a set of performance specifications including:

- Stabilization control.
- Position tracking.
- Disturbance rejection.
- Robustness.

1.3 Literature review

There are many researches and papers studies the inverted pendulum, especially the inverted pendulum on a cart, which become a regular and simple problem in control engineering. These studies helps to understand the concept of an inverted pendulum systems in general. So it was able to start driving the mathematical model of "ECP 505 model Inverted pendulum" and designing the controllers.

1.4 System Overview

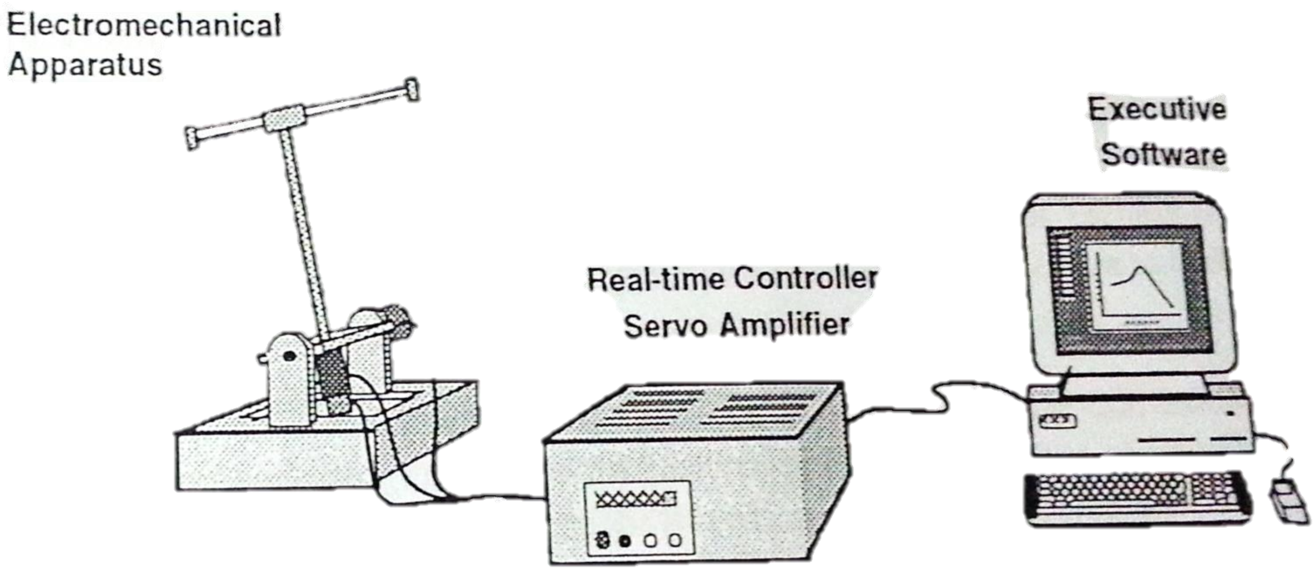

Figure 1.1: The Experimental control system

1.4.1 Mechanical and electrical part

The experimental control system is composed of the three subsystems shown in Figure 1.1. The first of these is the electromechanical plant which consists of the inverted pendulum mechanism, its actuator and sensors. The design features are:

- DC servo motor.
- High resolution encoders.
- A low friction sliding balance rod.
- Adjustable balance weight.

Next is the real-time controller unit which contains the digital signal processor (DSP) based real-time controller, servo/actuator interfaces, servo amplifier, and auxiliary power supplies. The DSP is capable of executing control laws at high sampling rates allowing the implementation to be modeled as continuous or discrete time. The controller also interprets trajectory commands and supports such functions as data acquisition, trajectory generation, and system health and safety checks. A logic gate array performs encoder pulse decoding. Two optional auxiliary digital-to-analog converters (DAC's) provide for real-time analog signal measurement. This controller is representative of modern industrial control implementation [8].

1.4.2 Computer part

With Inverted Pendulum, computers are used for two major purposes:

1. Design, analysis, and simulate the control part of the system
2. Running the real time controller

To achieve the first requirement MATLAB and Simulink is used. MATLAB provide a wide verity of functions, numerical algorithms, and toolboxes that help significantly not only to design and simulate the control system, but also to build executable real time applications. The second requirement which is controlling the Inverted Pendulum system at real time is achieved by using xPC target technique or embedded system based on microcontroller.

1.5 Control system

Inverted pendulum, in general is used in control lab for testing various control theories, including the classical linear control theories, and can be extended to the modern non-linear control fields.

As stated earlier, IP presents a number of complications and challenges in terms of their control, due to the fact that they are under-actuated mechanical systems, inherently open-loop unstable, with highly nonlinear dynamics.

Any controller applied to an IP must guarantee first the closed loop stability at the unstable inverted position. Second, disturbance rejection and robustness are also to be achieved. The possibility for a controller to satisfy these requirements varies according to the control strategy behind it. Here is a brief description of the main control technique that may be used in this project:

1. PID controller

PID is a proportional-integral-derivative controller which creates a control loop feedback mechanism. A PID controller calculates an "error" (e) value as the difference between a measured process variable and a desired set point. The desired closed loop dynamics is obtained by adjusting the three parameters K_P (proportional gain), K_I (integral gain) and K_D (derivative gain), based on the linear system transfer function [2].

2. State space control

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 [9].

3. Optimal control

The theory of optimal control is concerned with operating a dynamic system at minimum cost. Basically, a measure of the quality of a controller is formulated in terms of a performance index. This index is used to design the controller and depends on the control signal and the state vector. In this way the 'best' control signal is found that results in the minimum (or maximum) value of the index. The job of the control engineer in Linear Quadratic Regulator (LQR) design is therefore not to determine control parameters directly, but to define the appropriate measure for controller quality, the performance index, and to minimize or maximize it [1].

4. Adaptive control

That mean the controller must adapt with parameters. For example, as an IP, if one of the weights removed; a control law is needed that adapts itself to such changing conditions. Adaptive control is different from robust control in that it does not need a priori information about the bounds on these uncertain or time-varying parameters.

5. Non-linear control

Processes in reality like IP, robots and space craft typically have strong nonlinear dynamics. In control theory it sometimes possible to linearize such classes of systems and apply linear technique, but in many cases is desirable to expand the sight beyond linear theories, permitting the control of nonlinear system. These normally take advantage of results based on Lyapunove's theory.

6. Intelligent control

Intelligent control is a class of control techniques that use various Artificial Intelligence computing approaches like neural networks, fuzzy logic and machine learning.

1.6 Scope of Work

1. Drive the mathematical model for the inverted pendulum (ECP model 505).
2. Design different controllers using conventional technique, and in next semester we will use artificial intelligence technique (neural networks, fuzzy logic).
3. Simulate the controllers using MATLAB and shows the simulation results.
4. Apply these controllers on the device and shows the experimental results.
5. Compare between simulated and experimental results.

1.7 Chapters overview

This report consists of seven chapters including this chapter. The scope of each chapter is explained as stated below:

Chapter 1:
This chapter gives the introduction to the project report, recognition of the need, literature review, system overview, control system and scope of work.

Chapter 2:
This chapter discusses modeling of an inverted pendulum. It is contained the derivation in mathematical modeling for the dynamic of the inverted pendulum system, including the nonlinear and linearized equations.

Chapter 3:
This chapter discusses the linear controller's used in this project, including the theory of these controllers, controller design, Simulink model of the controller and the simulated results.

Chapter 4:
This chapter discusses the x-pc target technique and shows the experimental results.

Chapter 5:
In this chapter, the embedded system is discussed and shown the designed circuit and the experimental results.

Chapter 6:
This chapter discusses one branch of nonlinear control which is feedback linearization.

Chapter 7:
Conclusion and suggestion for future work.

2 Mathematical modeling

2.1 Introduction

The plant shown in Fig 2.1 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 stability. 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. 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.

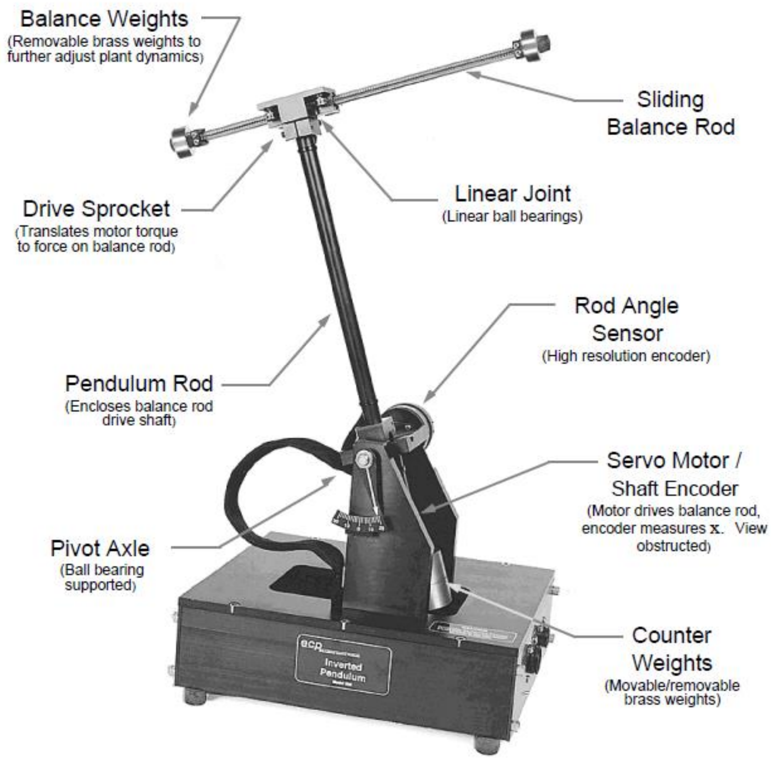

Figure 2.1: ECP model 505 Inverted pendulum

Mathematical modeling of the ECP model 505 Inverted Pendulum represents all important features of the system and describes its behavior in terms of differential equations. However the purposes for the modeling is to predict the dynamic behavior of the system as accurately as possible, and to have a knowledge of stability margins, controllability, observability, and the sensitivity of response to parameter changes.

Therefore two models will be derived for the system, linear one for controller design and analysis purposes, and nonlinear model for testing and simulating the dynamic system response as accurately as possible. In order to obtain the mathematical model, Lagrange's approach is used to drive the basic differential equations that govern system dynamics.

Lagrange differential equation:

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = Q_i \qquad (2-1)$$

Where:

- T: The total kinetic energy of the system.
- U: The total potential energy of the system.
- R: The total energy loses from the system due to viscous damping.
- q_i: The generalized coordinates that describes system motion.
- Q_i: Generalized forces and torques which acts in each generalized coordinate.

2.2 Dynamics Equations for the Inverted Pendulum

In this section a nonlinear model for the inverted pendulum is derived. This model includes the viscous friction of ball bearing pivot and viscous friction between the sliding rod and pendulum rod. Also the effect of disturbances acting on the system is included too.

2.2.1 Nonlinear expression

The mathematical model of the system consists of two second order nonlinear differential equation; these equations are derived using langrage approach. Based on Figure 2.2, the total kinetic and potential energies of the system can be expressed as:

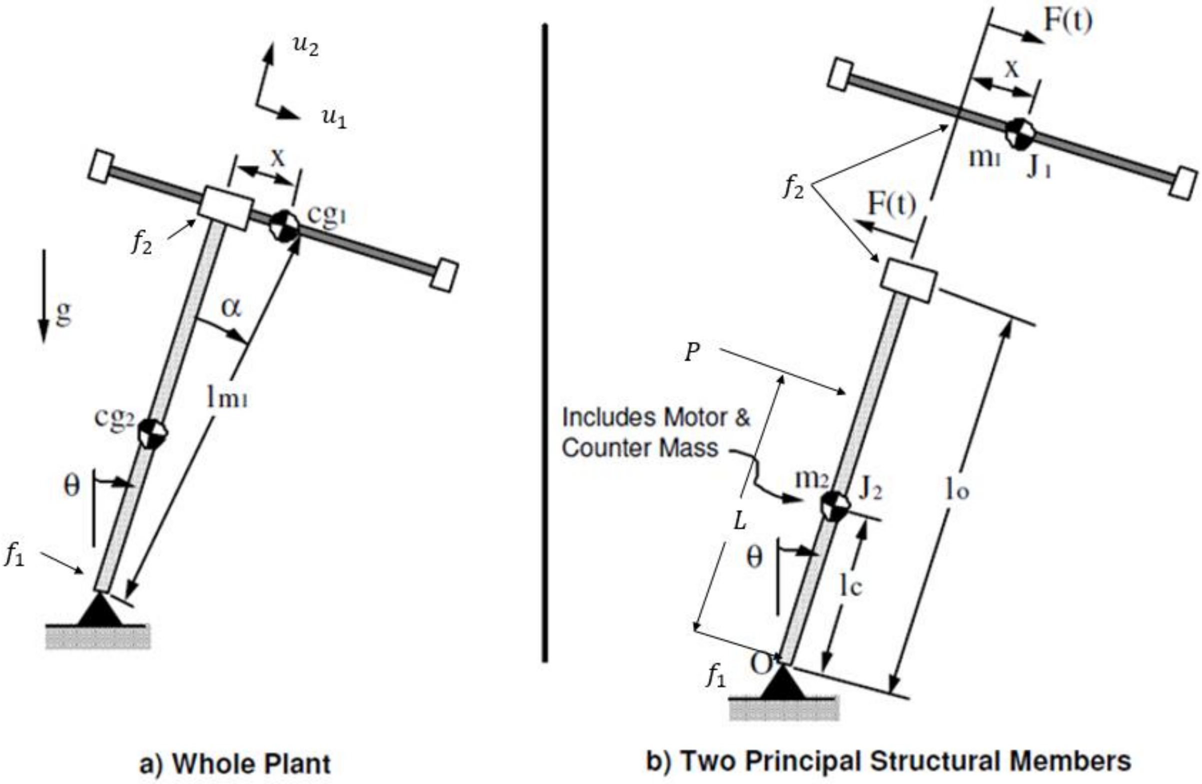

Figure 2.2: Plant model descriptions

Where:

- $F(t)$: the force driving the slider is equivalent to the motor torque divided by the drive pulley/belt contact radius.
- $P(t)$: the distarbance force.
- f_1*:* viscous friction on joint (O).
- f_2*:* viscous friction between slider and the joint A.
- α: the dynamic angle between pendulum rod an center of gravity of slider (cg_1).
- l_{m1}: Position of center of gravity for m_1.
- l_c: position of center of gravity for m_2.
- L: position of disturbance force.

Total kinetic energy:

$$\mathrm{T} = \frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2 + \frac{1}{2}J_1\dot{\Theta}^2 + \frac{1}{2}J_2\dot{\Theta}^2 \qquad (2-2)$$

Where v_1 is the magnitude velocity of sliding rod at central of gravity and v_2 is the magnitude velocity of pendulum rod at central gravity point. J_1 And J_2 are the polar moment of inertias around the center of gravities of links slider and pendulum rod respectively.

v_1 can be written as $v_1 = v_{1\mathrm{TRANS}} + v_{1\mathrm{ROT}}$. Therefore if follows

$$v_1 = (l_{m1}\dot{\theta}\cos\alpha\, u_1) + \dot{x}u_1 + (-l_{m1}\dot{\theta}\sin\alpha\, u_2)$$

Where u_1 and u_2 are unit vectors in the direction of the sliding rod and pendulum rod respectively. Therefore, we can write

$$v_1 = \left(l_{m1}\dot{\theta}\cos\alpha + \dot{x}\right)u_1 + (-l_{m1}\dot{\theta}\sin\alpha)u_2$$

And v_2 can be written as

$$v_2 = l_c\dot{\Theta}u_1$$

Combining the relationship for u_1 and u_2 into kinetic energy yields

$$\mathrm{T} = \frac{1}{2} m_1 \left\{ \left(l_{m1}\dot{\theta} \cos\alpha + \dot{x} \right)^2 + (-l_{m1}\dot{\theta}\sin\alpha)^2 \right\} + \frac{1}{2} m_2 \left(l_c \dot{\theta} \right)^2 + \frac{1}{2} J_1 \dot{\Theta}^2 + \frac{1}{2} J_2 \dot{\Theta}^2$$

Rearranging the terms into the above relationship yields

$$\mathrm{T} = \frac{1}{2} m_1 \left\{ \dot{x}^2 + \left(l_{m1}\dot{\Theta} \right)^2 + 2 l_{m1} \dot{x} \dot{\Theta} \cos\alpha \right\} + \frac{1}{2} m_2 \left(l_c \dot{\theta} \right)^2 + \frac{1}{2} J_1 \dot{\Theta}^2 + \frac{1}{2} J_2 \dot{\Theta}^2$$

Since the system is constrained by

$$l_0 = l_{m1} \cos \propto$$

$$l_{m1}^2 = x^2 + l_0^2$$

We can write the total kinetic energy of the system as:

$$T = \tfrac{1}{2} \dot{\theta}^2 [J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2] + \tfrac{1}{2} m_1 \dot{x}^2 + m_1 l_0 \dot{x}\, \dot{\theta} \qquad (2\text{-}3)$$

Now consider the potential energy. Taking the reference point as $\theta = 0^\circ$ and $x = 0$, we have

$$U = m_1 g l_{m1} \cos(\theta + \alpha) + m_2 l_c g \cos\theta$$

Or the above equation can be extended to become:

$$U = m_1 g l_{m1} \cos\alpha \cos\theta - m_1 g l_{m1} \sin\theta \sin\alpha + m_2 l_c g \cos\theta$$

Since $l_0 = l_{m1} \cos\alpha$ *and* $x = l_{m1} \sin\alpha$ we have the potential energy relationship

$$U = m_1 g l_0 \cos\theta - m_1 g x \sin\theta + m_2 l_c g \cos\theta \qquad (2\text{-}4)$$

From the Fig 2.2 we have that the system loss energy, R (t) is:

$$R = \tfrac{1}{2} f_1 \dot{\theta}^2 + \tfrac{1}{2} f_2 \dot{x}^2 \qquad (2\text{-}5)$$

From figure 2.2 the total work equals to:

$$\partial w = PL\partial\theta - Fl_O\partial\theta + F(l_O\partial\theta + \partial x)$$

Or the above equation can be simplified to be:

$$\partial w = PL\partial\theta + F\partial x \qquad (2\text{-}6)$$

Appling Lagrange's equation for each generalized coordinate $x\ and\ \theta$, yields:

1) In x direction:

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

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}}\right) - \frac{\partial T}{\partial x} + \frac{\partial U}{\partial x} + \frac{\partial R}{\partial \dot{x}} = Q_1 \qquad (2\text{-}7)$$

Where:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}}\right) = m_1\ddot{x} + m_1 l_o\ddot{\theta}$$

$$\frac{\partial T}{\partial x} = m_1 x\dot{\theta}^2$$

$$\frac{\partial U}{\partial x} = -m_1 g\sin\theta$$

$$\frac{\partial R}{\partial \dot{x}} = f_2\dot{x}$$

$$Q_1 = \frac{\partial w}{\partial x} = \frac{PL\partial\theta + F\partial x}{\partial x} = F$$

Applying Eq.(2-7) and simplifying it yields:

$$m_1\ddot{x} + m_1 l_0\ddot{\theta} - m_1 x\dot{\theta}^2 - m_1 g\sin\theta + f_2\dot{x} = F \qquad (2\text{-}8)$$

2) In θ direction:

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

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) - \frac{\partial T}{\partial \theta} + \frac{\partial U}{\partial \theta} + \frac{\partial R}{\partial \dot{\theta}} = Q_2 \tag{2-9}$$

Where:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) = [J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2]\ddot{\theta} + 2m_1 x\dot{x}\dot{\theta} + m_1 l_0 \ddot{x}$$

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

$$\frac{\partial U}{\partial \theta} = -m_1 g l_0 \sin\theta - m_1 g x \cos\theta - m_2 l_c g \sin\theta$$

$$\frac{\partial R}{\partial \dot{\theta}} = f_1 \dot{\theta}$$

$$Q_2 = \frac{\partial w}{\partial \theta} = \frac{PL\partial\theta + F\partial x}{\partial \theta} = PL$$

Applying Eq.(2-9) and simplifying it yields:

$$[J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2]\ddot{\theta} + 2m_1 x\dot{x}\dot{\theta} + m_1 l_0 \ddot{x} - (m_1 l_0 + m_2 l_c) g \sin\theta - m_1 g x \cos\theta + f_1 \dot{\theta} \;=\; PL \tag{2-10}$$

2.2.2 Linearization about Equilibrium point:

From equations 2-8 and 2-10 we have equilibrium points for a motionless system, that $(\dot{x} = \ddot{x} = \dot{\theta} = \ddot{\theta} = 0)$ and $F(t) = 0$. Linearizing the equations with respect to equilibrium points $x = x_e, \theta = \theta_e$, we have:

$$\theta_e = 0, x_e = 0.$$

Using Taylor series expansions, about a small angle, we can neglect second order terms and write $\sin\theta = \theta$, $\cos\theta = 1$, $\cos\alpha = 1$ and $l_0 = l_m \cos\alpha \cong l_m$, so the linearized equations of motion can be written as:

$$m_1\ddot{x} + m_1 l_0 \ddot{\theta} - m_1 g\theta + f_2\dot{x} = F \qquad (2\text{-}11)$$

$$[J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2]\ \ddot{\theta} + m_1 l_0 \ddot{x} - (m_1 l_0 + m_2 l_c) g\theta - m_1 g x\ + f_2\dot{\theta} = PL \qquad (2\text{-}12)$$

Simplifying equation 2-11 in terms of $\ddot{x}$ yields:

$$\ddot{x} = \frac{F}{m_1} - l_0\ddot{\theta} + g\theta - \frac{f_2\dot{x}}{m_1} \qquad (2\text{-}13)$$

Simplifying equation 2-12 in terms of $\ddot{\theta}$ yields:

$$\ddot{\theta} = \frac{1}{J_0}PL - \frac{m_1}{J_0} l_0\ddot{x} + (m_1 l_0 + m_2 l_c) g\theta - \frac{m_1}{J_0} g x - \frac{f_2\dot{\theta}}{J_0} \qquad (2\text{-}14)$$

Where $J_o = [J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2]$

Substitute (2-13) in (2-11) and (2-14) in (2-12)

$$J^*\ddot{\theta} - m_2 l_c g\theta - m_1 g x - m_1 l_0 f_2 \dot{x} = PL - l_0 F \qquad (2\text{-}15)$$

$$J^*\ddot{x} - (J^* - m_2 l_c l_0) g\theta + m_1 l_0 g x - J_1 l_0 f_1 \dot{\theta} = \frac{J_0}{m_1} F - l_0 PL \qquad (2\text{-}16)$$

Where $J^* = J_0 - m_1 l^2{}_0$

Equations (2-15) and (2-16) can be described in matrix form:

$$\begin{bmatrix} J^* & 0 \\ 0 & J^* \end{bmatrix}\begin{bmatrix} \ddot{\theta} \\ x \end{bmatrix} + \begin{bmatrix} 0 & -m_1 l_0 f_2 \\ -J_0 l_0 f_1 & 0 \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} -m_2 l_0 g & -m_1 g \\ -(J^* - m_2 l_c l_0) g & m_1 l_0 g \end{bmatrix}\begin{bmatrix} \theta \\ x \end{bmatrix} = \begin{bmatrix} -l_0 \\ \frac{J_0}{m_1} \end{bmatrix} F + \begin{bmatrix} 1 \\ -l_0 \end{bmatrix} PL \qquad (2\text{-}17)$$

2.3 State Space Model:

Using the linearized model and neglecting viscous friction, equations (2-15) and (2-16) become:

$$J^*\ddot{\theta} - m_2 l_c g\theta - m_1 gx = PL - l_0 F \;\ldots\ldots\ldots (2-18)$$

$$J^*\ddot{x} - (J^* - m_2 l_c l_0) g\theta + m_1 l_0 gx = \frac{J_0}{m_1} F - l_0 PL \;\ldots\ldots (2-19)$$

$$J^* = J_0 - m_1 {l^2}_0$$

Furthermore to yield the state space representation for the linear system, four state are needed to discribe the system. These are chosen to be $x, \dot{x}, \theta \; and \; \dot{\theta}$. The input to the system is F. the state space model of the system is expressed as follows:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + E\boldsymbol{P}$$

$$\boldsymbol{y} = C\boldsymbol{x} + D\boldsymbol{u}$$

Let

$x_1 = \theta \quad \rightarrow \quad \dot{x}_1 = x_2$

$x_2 = \dot{\theta} \quad \rightarrow \quad \dot{x}_2 = \ddot{\theta} = \frac{1}{J^*} PL - \frac{l_0}{J^*} F + \frac{m_2}{J^*} l_c g x_1 + \frac{m_1}{J^*} g x_3$

$x_3 = x \quad \rightarrow \quad \dot{x}_3 = x_4$

$x_4 = \dot{x} \quad \rightarrow \quad \dot{x}_4 = \ddot{x} = \dfrac{J_0}{J^* m_1} F - \dfrac{l_0}{J^*} PL - \dfrac{m_1 l_0}{J^*} g x_3 + \dfrac{(J^* - m_2 l_c l_0) g}{J^*} x_1$

Thus the state space model for IP:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ m_2 l_c g / J^* & 0 & m_1 g / J^* & 0 \\ 0 & 0 & 0 & 1 \\ (J^* - m_2 l_o l_c) g / J^* & 0 & -m_1 l_o g / J^* & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -l_o / J^* \\ 0 \\ J_o / m_1 J^* \end{bmatrix} F + \begin{bmatrix} 0 \\ 1 / J^* \\ 0 \\ -l_o / J^* \end{bmatrix} PL$$

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

(2-20)

System parameters are defined in Table 2.1

Table 2.1 System parameters

Symbol	**Description**	**Value**
l_o	Length of pendulum rod from pivot to the sliding rod T section	0.330(m)
m_1	Mass of the complete sliding rod including all attached elements.	0.213(kg)
m_2	Mass of the complete moving assembly minus m_1	1.785(kg)
J_o	Inertia evaluated at equilibrium point	0.0594 (kg.m^2)
J^*	$J^* = J_o - m_1 l_o^2$	0.0362 (kg.m^2)
l_C	position of center of gravity for m_2.	0.0281(m)

Substituting the values of Table 2.1 into state space model matrices yields

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -14.23 & 0 & 57.562 & 0 \\ 0 & 0 & 0 & 1 \\ 14.5 & 0 & -19 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -9.116 \\ 0 \\ 7.706 \end{bmatrix} F + \begin{bmatrix} 0 \\ 27.8 \\ 0 \\ -9.2 \end{bmatrix} PL$$

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

(2-21)

3 Control system design

3.1 Introduction

ECP model 505 Inverted Pendulum is used in control lab for testing various control theories, including the classical linear control theories, and can be extended to the modern non-linear control fields. The main challenge is to build a controller that stabilizes the inverted pendulum at its inverted position, or to track any other positions. Further-more such a controller should reject disturbances acting on the slider or on the vertical rod, but within physical limits of the device. However IP presents a number of challenges in term of its control, due to the following facts:

- Its open-loop unstable at the desired operating point.
- External disturbances that act on the system are not directly measurable.
- All of the states are dynamically coupled, that means the change in any state will affect all other states.
- Some states are not measured like velocities which should be accurately estimated.

Moreover to make the challenge more interesting, the designed controller should not exceed the limits of the device, i.e. displacement, speed and torque, which comes from the length of the sliding rod and the limited torque generated by the motor.

In the upcoming sections, linear control theories and strategies will be tested and discussed, including:

- PID controller for the slider.
- State feedback controllers (regulator, tracker, observer and discrete controller).

3.2 Controller design

In this section every linear control strategy used is discussed in details, after that a Simulink model built for the controller, and then shown the simulated results.

3.2.1 PID controller for the slider

In this section the PID controller is design only for the slider, to check if the encoders, motor and the driver working properly. So the PID is a proportional-integral-derivative controller which creates a control loop feedback mechanism. A PID controller calculates an "error" (e) value as the difference between a measured process variable and a desired set point. The desired closed loop dynamics is obtained by adjusting the three parameters K_P (proportional gain), K_I (integral gain) and K_D (derivative gain), based on the linear system transfer function.

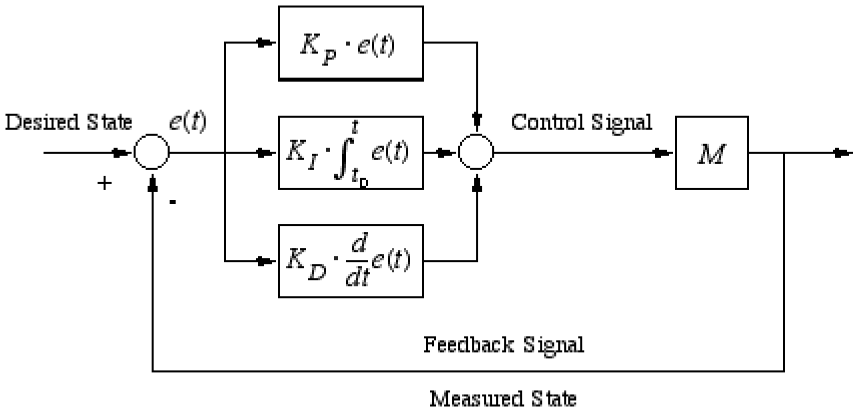

Figure 3.1: PID controller

1. Principle

The pendulum rod is fixed at its inverted position, so the sliding rod can be simply modeled as a mass, and the differential equation is:

$$m_1\ddot{x} = F \tag{3-1}$$

The transfers function of the mass:

$$G(s) = \frac{1}{m_1 s^2} \tag{3-2}$$

Where

m_1: Mass of the complete sliding rod including all attached elements ($0.213\ kg$).

2. Controller

As design requirement, let the settling time of the system equals to $0.15\ sec$ and the percent over shoot ($\%OS$) to be 10%. For designing the controller it's preferred to use MATLAB SISOTOOL, because it's simple and easy function. So the controller gains are:

- $K_P = 422.1$
- $K_I = 86.17$
- $K_D = 11.25$

3. Simulink model

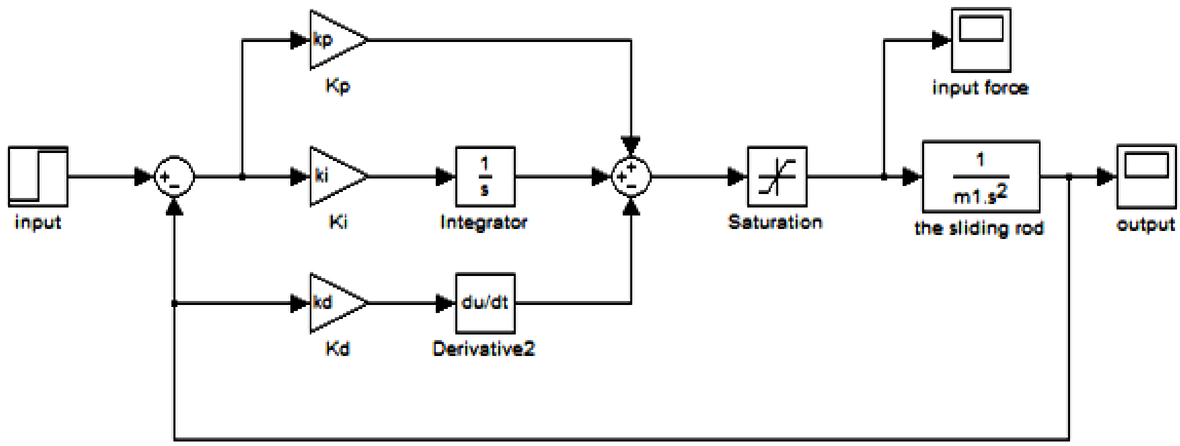

Figure 3.2: PID controller with D-element in reveres path

The Simulink model is shown in the figure 3.2, as noticed it was preferred to use the D-element in reveres path, because the zero does not appear when obtaining the closed loop transfer function[1].

[1] The transfer function of the inner loop $G_1(s) = 1/(m_1 s^2 + K_D s)$, and the closed loop transfer function of the outer loop $T(s) = (K_P s + K_I)/(m_1 s^3 + K_D s^2 + K_P s + K_I)$. This discussed in details in [2] chapter 9, page 495.

4. Simulation results

The slider commanded to move 5 cm, so the input force and the response as follow:

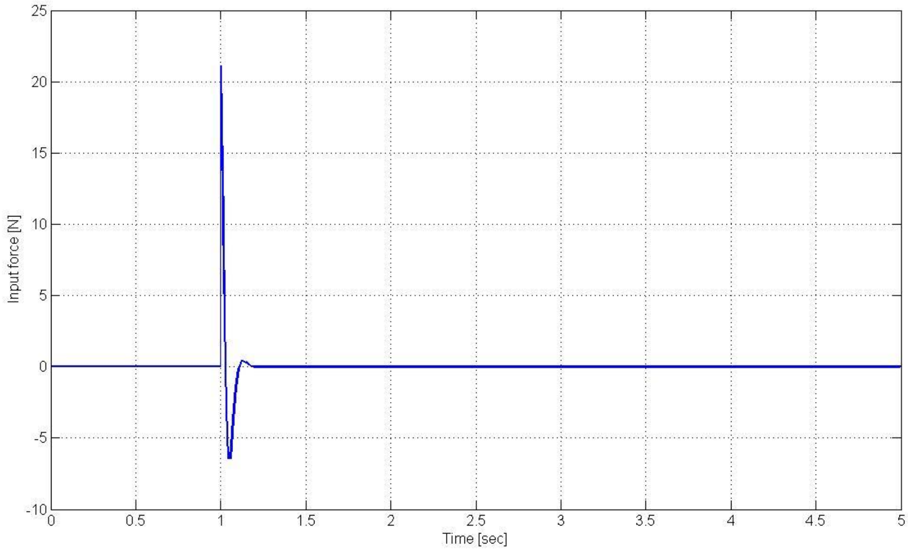

Figure: 3.3: Input force (PID)

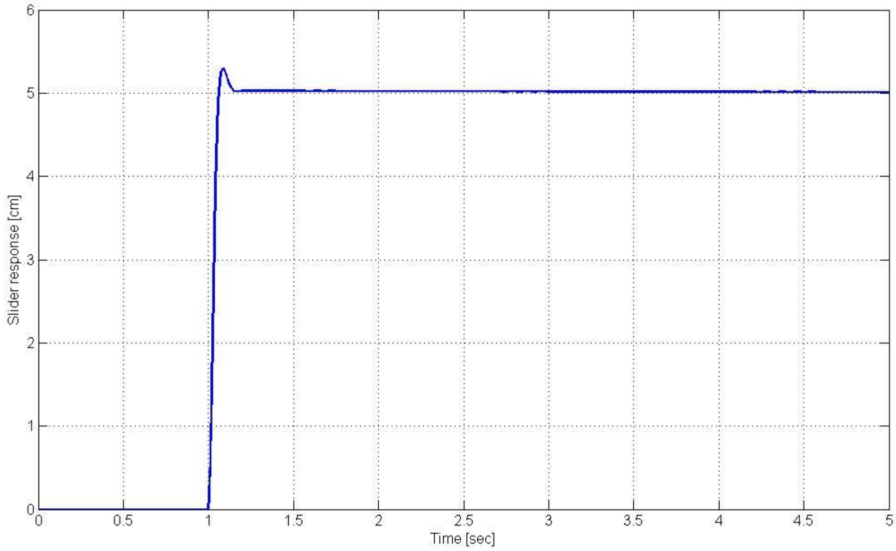

Figure 3.4: Response (PID)

Figure 3.3 shows the input force needed to move the mass 5 cm, and Figure 3.4 shows response of the mass with 10.22% over shoot and settling equal to 0.138 sec which approximately meets the design requirement.

3.2.2 State feedback controller

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.

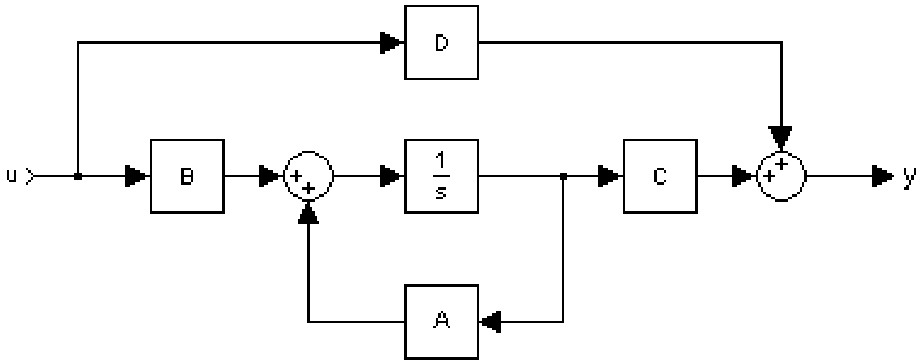

Figure 3.5: Block diagram representation of the state space equations

In order to obtain the state-space representation for any system, you need to know system inputs, outputs, in addition to the states. The general linear time invariant state space model that is used throughout this chapter:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + E_d\boldsymbol{p}$$
$$\boldsymbol{y} = C\boldsymbol{x} + D\boldsymbol{u} \qquad (3\text{-}3)$$

Where:

- $\boldsymbol{x} \in R^n$: The state vector.
- $\boldsymbol{u} \in R^m$: The input vector. Where m equals 1 in the IP.
- $\boldsymbol{y} \in R^r$: The output vector.
- $\boldsymbol{p} \in R^m$: The disturbance vector.

- $\boldsymbol{A} \in R^{n\times n}$: The system matrix.
- $\boldsymbol{B} \in R^{n\times m}$: The input matrix.
- $\boldsymbol{E_d} \in R^{n\times m}$: The disturbance matrix.
- $\boldsymbol{C} \in R^{r\times n}$: The output matrix.
- $\boldsymbol{D} \in R^{r\times m}$: The feed-forward matrix. In this case D matrix equals zero, since the transfer functions are strictly proper.

Previously from chapter two, the state space representation for the IP (ECP 505 model) is:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -14.23 & 0 & 57.562 & 0 \\ 0 & 0 & 0 & 1 \\ 14.5 & 0 & -19 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -9.116 \\ 0 \\ 7.706 \end{bmatrix} F + \begin{bmatrix} 0 \\ 27.8 \\ 0 \\ -9.2 \end{bmatrix} PL$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} \qquad \text{(2-21)}$$

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, as explained in [2].

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. This can be done by Hautus method by finding rank $[\lambda I - A \quad B]$ for every eigenvalues (λ_i) of the system. Alternatively, the controllability of the pair (A, B) is checked by calculating the controllability matrix (C_M), such that:

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

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, two methods are shown in this chapter, which are pole placement and optimal control methods; however pole placement method is used in this chapter.

a) 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.

b) Optimal control method:

The theory of optimal control is concerned with operating a dynamic system at minimum cost. That means the gains $[K]$ are determined to minimize the quadratic performance index [1].

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

Where:

- Q: It's a positive semi-definite matrix that represents the importance of the states relative to each other.
- R: It's a positive definite matrix that represents the relative importance of control inputs. Since there is only one control input this matrix is (1×1) matrix.

After determining the Q and R matrices, MATLAB function (lqr) is used to obtain the optimal gains value and the corresponding eigenvalues of the system.

3.2.2.1 Regulator design

The problem in this section is to design a state feedback controller that stabilized the inverted pendulum at its inverted position, which is the desired operating position, were the system is linearized. This requires overcoming the effects of disturbances and non-zero initial conditions.

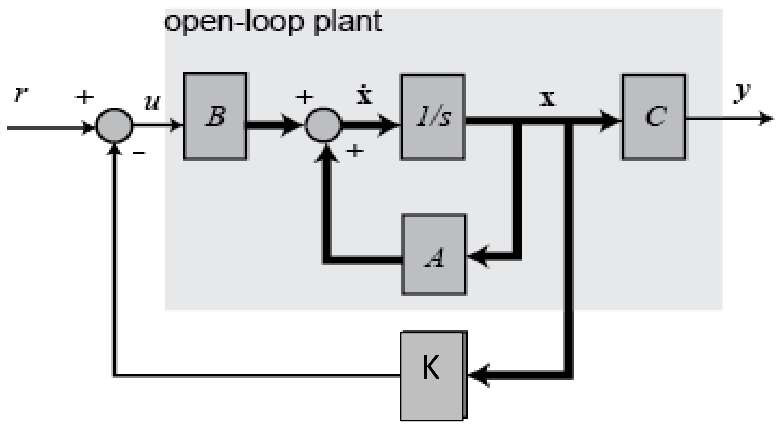

Figure 3.6: Regulator schematic

1. Principle:

The inverted pendulum open loop system dynamics are given by:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \tag{3-4a}$$

$$\boldsymbol{y} = C\boldsymbol{x} \tag{3-4b}$$

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:

$$\boldsymbol{u} = \boldsymbol{r} - K\boldsymbol{x} \tag{3-5}$$

Where:

- $\boldsymbol{r}$: is some reference input, in the case of regulator $\boldsymbol{r} = 0$.
- $\boldsymbol{K}$: is a gain where $\boldsymbol{K} \in R^{m\times \mathrm{n}}$

Find the closed-loop dynamics:

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

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

$$\dot{\boldsymbol{x}} = \boldsymbol{A_k x} + \boldsymbol{Br}$$

$$\boldsymbol{y} = \boldsymbol{Cx} \qquad (3\text{-}6)$$

Where $\boldsymbol{A_k}$: the closed loop system matrix.

So the eigenvalues of A_k (the closed loop poles) could be placed anywhere in the S-plane to get the desired response just by playing with matrix $\boldsymbol{K}$. 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: $3.5360 \quad -3.5360 \quad 6.7530i \quad -6.7530\mathrm{i}$

Open loop zeros: $5.4643 \quad -5.4643$

The open loop system is unstable; because there is one pole at right half plan, as appeared in the root locus figure 3.7.

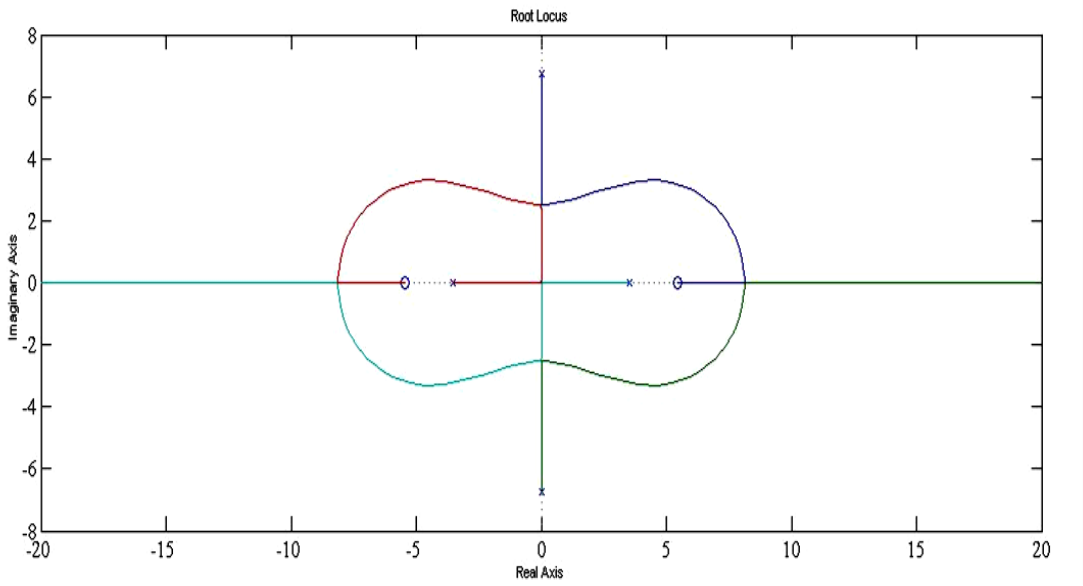

Figure 3.7: Root locus for the plant

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:

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

$$Rc = rank(C_m)$$

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 $\boldsymbol{K}$ 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 $(10\ to\ 12)$ rad/s and (0.8 to 0.9) respectively.

The figure below shows the design region in S-plan.

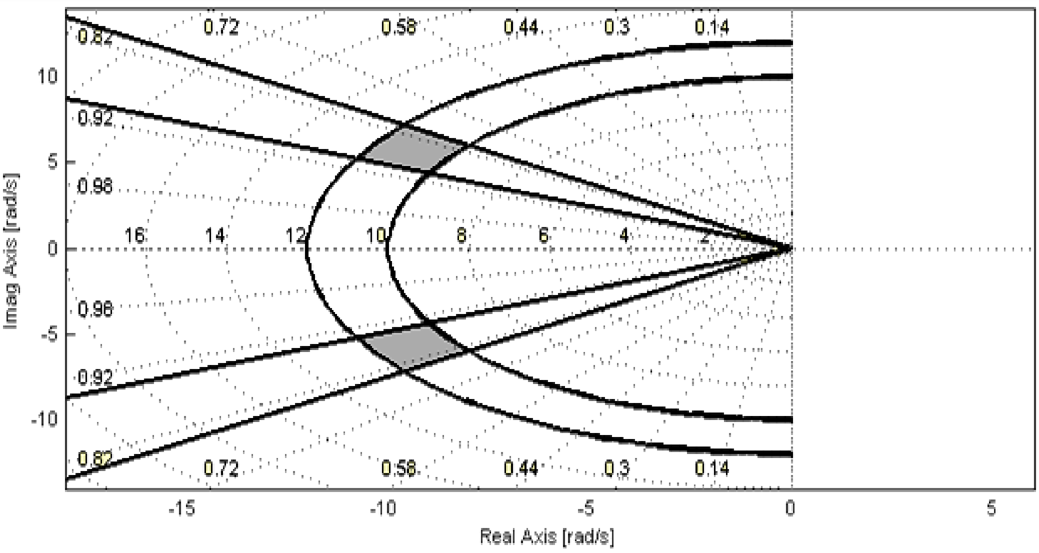

Figure 3.8: Design regions for the

So let the closed loop poles be at:

Poles = $[-9+4.36i \quad -9-4.36i \quad -13.33+5.23i \quad -13.33-5.23i\ ]$

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

$K_{Reg} = place(A, B, Poles)$

$K_{Reg} = [\ 95.6892 \quad 26.6820 \quad 210.9835 \quad 37.3700]$

3. Simulink model

The regulator is built and simulated using MATLAB SIMULINC, figure 3.9 shows the Simulink model.

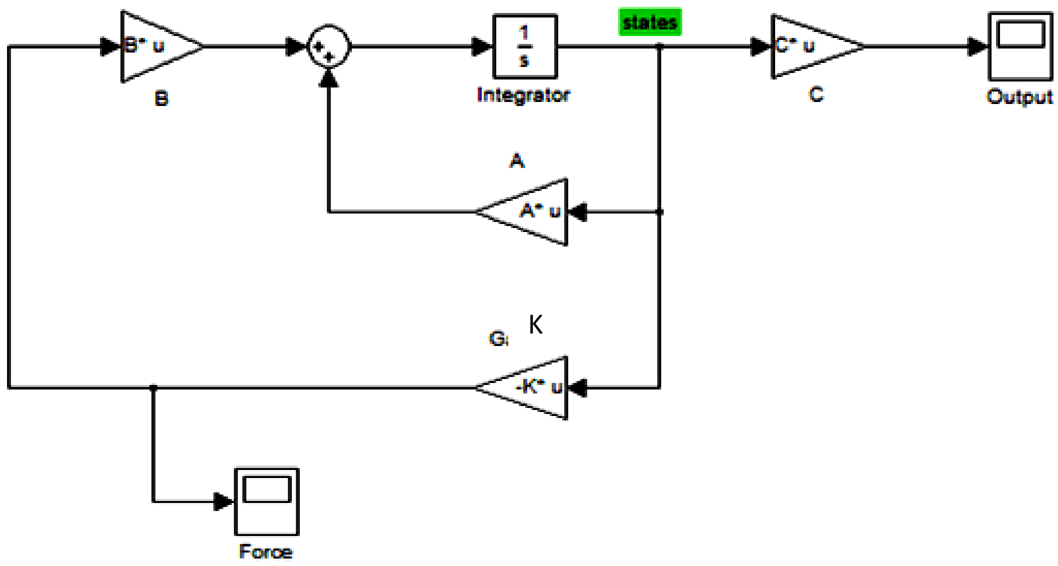

Figure 3.9: Regulator Simulink model

4. Simulation results

The simulation results used to check if the system response meets the requirement or not. The initial conditions supposed to be 0.1 rad for pendulum rod angle and 0.0 m for the slider displacement. These results show the controller response with initial condition and the input force to the plant.

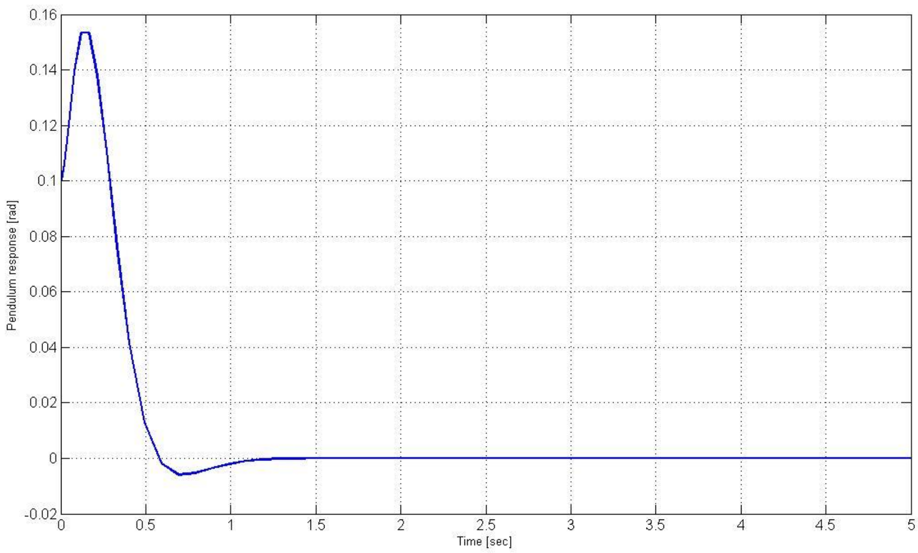

Figure 3.10: Pendulum rod response (Regulator)

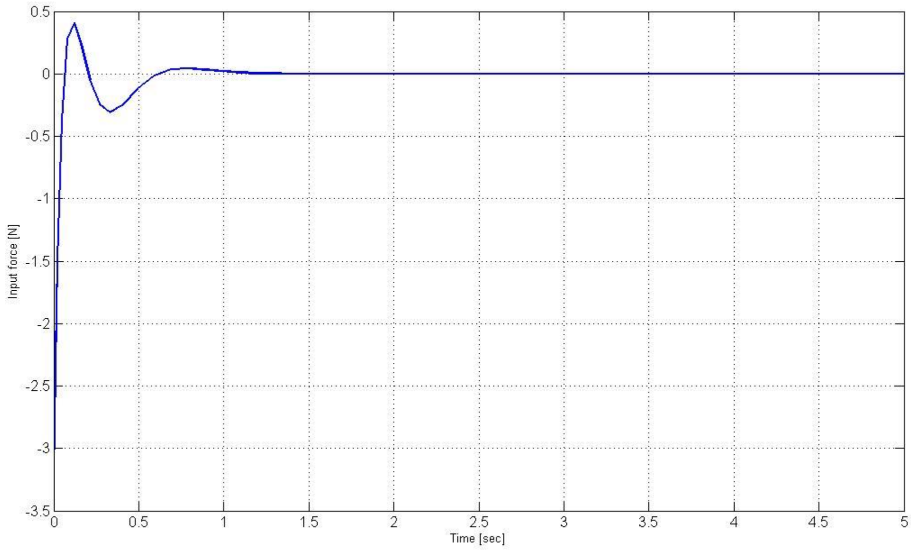

Figure 3.11: Input force (Regulator)

3.2.2.2 Tracker design

In this section it is desired to design a state feedback controller that able to track a desired reference input of the pendulum rod, that means to stabilize the inverted pendulum at any angle within the physical limits of the device ($\pm 20^{\circ}$) and to overcoming the effects of disturbances.

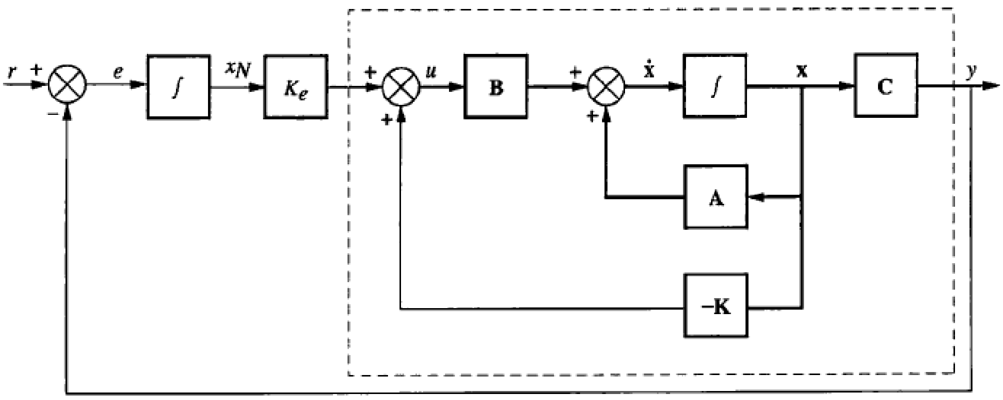

Figure 3.12: Integrator control for steady state error design

1. Principle:

From figure 3.12, the controller shown inside the dashed box is a Regulator which discussed in Section I. In order to to make the output follows the input; a feedback path from the output has been added to form the error, *e*, which is fed forward to the controlled plant via an integrator.

Since the integrator increases the system type; we were able to make the output track a step input with zero steady-state error and with desired transient response specifications.

To drive the state equations for the Tracker consider Figure 3.12:

$$\dot{x_n} = r - \boldsymbol{Cx} \tag{3-7a}$$

Where $\dot{x_n}$: error signal

Writing the state equations from Figure 3.12, gives:

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{B}u$$

$$y = \boldsymbol{Cx} \tag{3-7b}$$

Equations 3.7 a, b can be written in compact form as:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ -\boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_n \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix} u + \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} r$$

$$y = [\boldsymbol{C} \quad 0] \begin{bmatrix} \boldsymbol{x} \\ x_n \end{bmatrix} \tag{3-8}$$

But,

$$u = -[\boldsymbol{K} \quad -K_e] \begin{bmatrix} \boldsymbol{x} \\ x_n \end{bmatrix} \tag{3-9}$$

Substituting Eq. (3-9) into (3-8) and simplifying, we obtain

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} - \boldsymbol{BK} & \boldsymbol{B}K_e \\ -\boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_n \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} r$$

$$y = [\boldsymbol{C} \quad 0] \begin{bmatrix} \boldsymbol{x} \\ x_n \end{bmatrix}$$

(3-10)

2. Controller

The extended matrices of the system are given as follows:

$$\boldsymbol{A_e} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ -\boldsymbol{C} & 0 \end{bmatrix} \quad (3\text{-}11a)$$

$$\boldsymbol{B_e} = \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix} \quad (3\text{-}11b)$$

$$\boldsymbol{C_e} = [\boldsymbol{C} \quad 0] \quad (3\text{-}11c)$$

Now to check the controllability of the pair (A_e, B_e), it is performed using MATLAB as follows:

$C_{me} = ctrb(A_e, B_e)$

$Rce = rank(C_{me})$

Which is found to be 5, full rank, meaning that the system is fully controllable, and the gains $[K \; -K_e]$ vector can be calculated to achieve the desired response. As in Section I the gains found by pole placement method. Then let the natural frequency (ω_n) rang between $(10 \; and \; 12)$ rad/s and the damping ratio (ζ) rang between (0.8 and 1). Figure 3.13 shows the design region in S-plane.

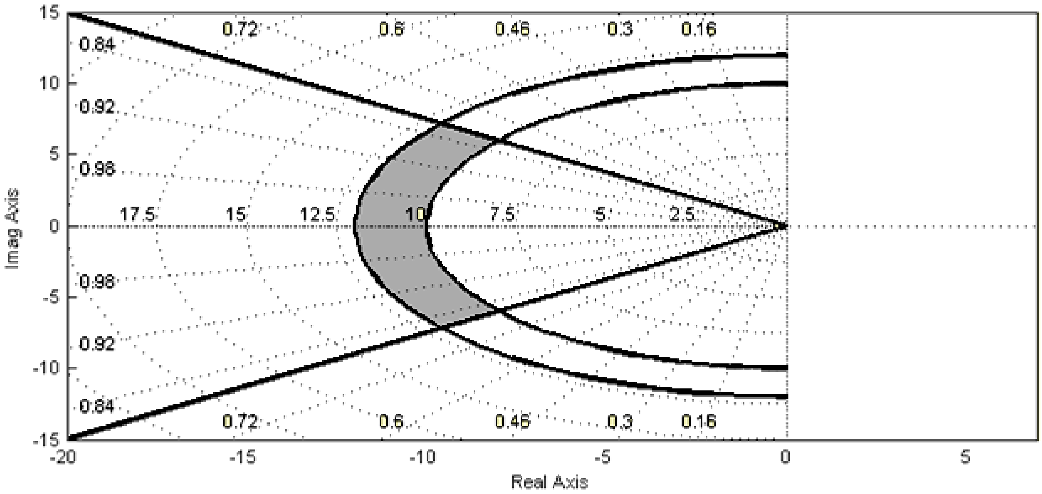

Figure 3.13: Design region for the tracker in S-plane

Then a possible selection of the closed loop poles, taking into account the desired transient response, and motor saturation limits, is:

Poles = $[-9+4.36i \quad -9-4.36i \quad -13.33+5.23i \quad -13.33-5.23i \quad -10\ ]$

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

$K_{track} = place(A_e, B_e, Poles)$

$K_{track} = [362.5092 \quad 87.5148 \quad 584.6835 \quad 110.6239 \quad -760.3618]$

$K = K_{track}(1{:}4)$

$K_e = -K_{track}(5)$

3. Simulink model

The model used in simulation process is shown in figure 3.14, this model includes a robust tracking state feedback.

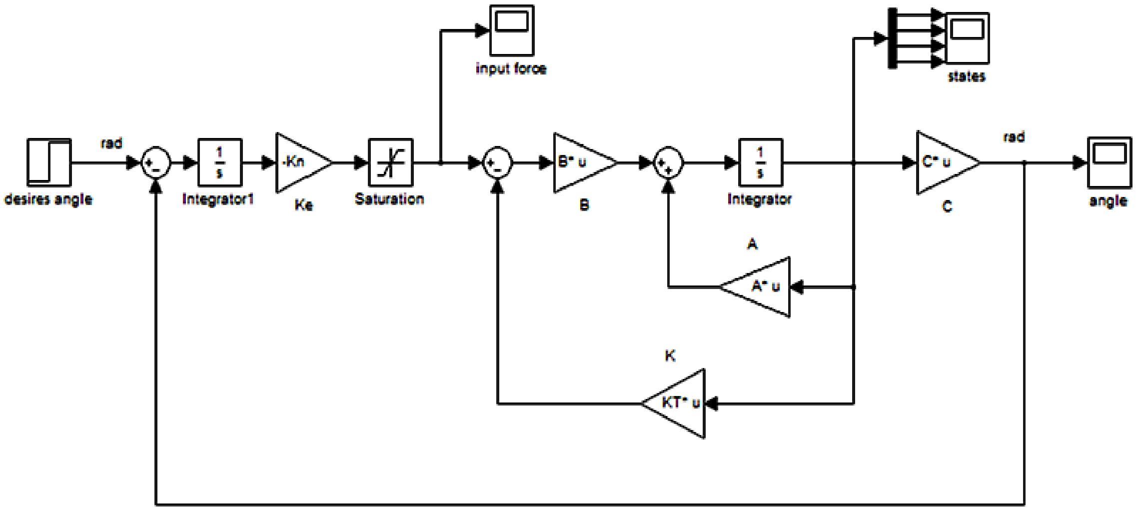

Figure 3.14: Simulink model for tracker

4. Simulation results

The initial pendulum rod angle = 0.0 rad and the desired angle of the pendulum rod = 0.2 rad; so the simulation results as follows:

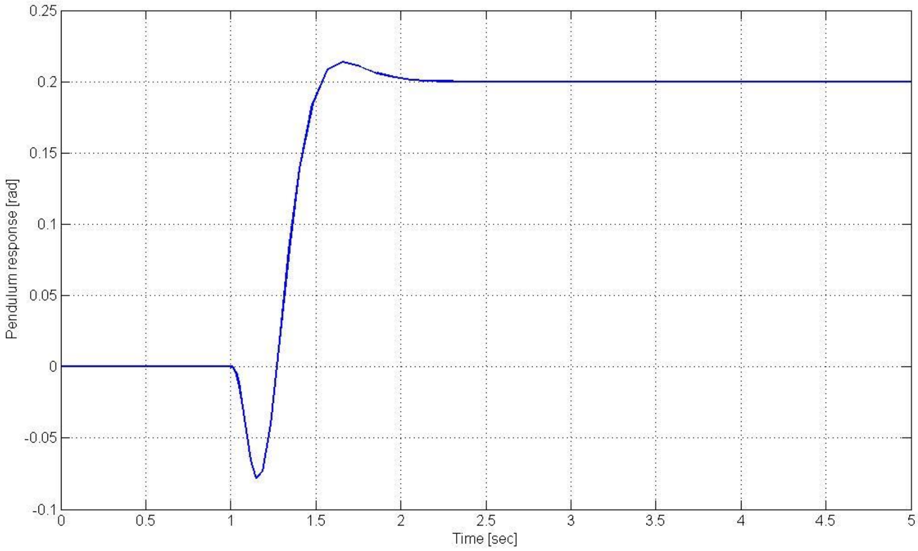

Figure 3.15: Pendulum response (Tracker)

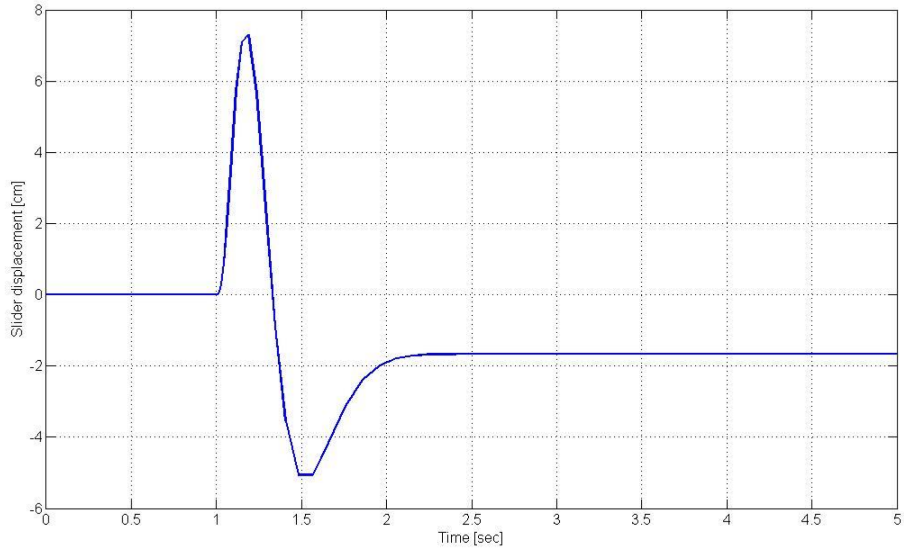

Figure 3.16: Slider displacement (Tracker)

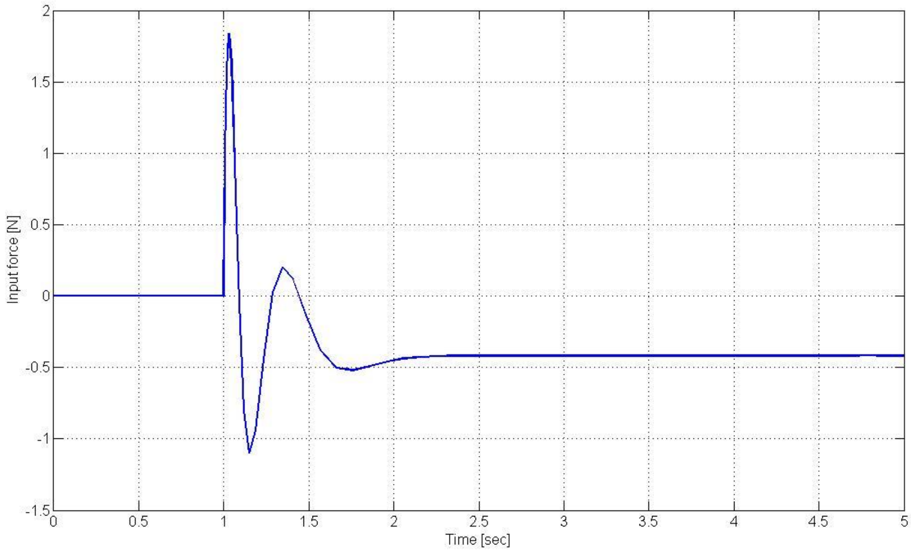

Figure 3.17: Input force (Tracker)

From figure 3.15 the percent overshoot equals to 6.9% and the settling time equals to 0.9 seconds, figure 3.16 shows the displacement needed by the slider to make pendulum angle 0.2 rad, however the inverted pendulum at this position is stable, and figure 3.17 shows the force needed to hold the slider from slipping.

3.2.2.3 Observer design

The observer (model of the plant), is used to calculate state variables that are not directly measured. Figure 3.18 shows the basic concept of observer design, where the measured outputs from the system are compared to those estimated, and then error signal is fed back to the observer. However, the dynamics of the observer should be made much faster than the dynamics of control system; to pick x out from the hat.

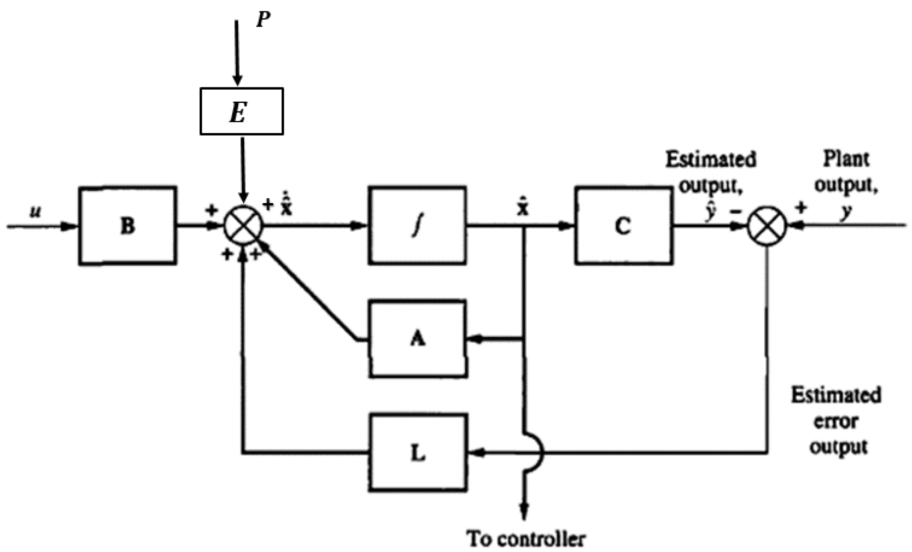

Figure 3.18: Observer design process

1. Principle

From the above figure, the state equation of the observer is found as follows:

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \quad \text{(3-12a)}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C}\hat{\boldsymbol{x}} \quad \text{(3-12b)}$$

That it is assumed no disturbances acting on the rod ($p = 0$). The error signal between the measured output and observer output is:

$$\hat{\boldsymbol{e}} = \boldsymbol{x} - \hat{\boldsymbol{x}} \quad \text{(3-13)}$$

Subtract Eq.3-12 from Eq.3-4 and Substituting the output equation into the state equation

$$(\dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}}) = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})(\boldsymbol{x} - \hat{\boldsymbol{x}}) \quad \text{(3-14)}$$

Now the error dynamic equation is:

$$\dot{\hat{\boldsymbol{e}}} = (\boldsymbol{A} - \boldsymbol{LC})\hat{\boldsymbol{e}} \tag{3-15}$$

So to achieve the desired speed of the observer, the poles of the error characteristic equation must be place faraway from poles of the controlled system; this can be achieved by choosing appropriate gain vector (**L**). To obtain the gain vector (**L**); pole placement or optimal control methods can be used in a similar way to the feedback gain, where observer poles are selected to be 5 to 10 times faster than the system.

To be able to design an observer, the system must be checked if it observable or not. If the knowledge of the output $y(t)$ and the input $u(t)$ over a finite interval of time suffices to determine the initial states the system is said to be observable, otherwise the system is said to be unobservable. This can be done by check the observability of the pair (A, C) through calculating the observability matrix (O_M), such that:

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

If O_M has a full column rank, then the system is observable, and it’s possible to find a gain vector $[L]$. This matrix can be calculated with $(obsv)$ MATLAB function.

2. Controller

Checking the observability of the system based on the two direct measurements $x\ and\ \theta,$ the rank of the observability matrix is found as follows:

$$O_m = obsv(A, Cm)$$

$$R_o = rank(O_m)$$

Where C_m is the measurement matrix, $C_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Which is found to be 4, full rank, meaning that the system is observable, and the gain vector $[L]$ can be calculated.

Then let observer poles 10 times faster than the poles of the controller, so

$$Obs\_Poles = [-19 + 4.36i \;\; -19 - 4.36i \;\; -23.33 + 5.23i \;\; -23.33 - 5.23i \;]$$

With MATLAB the gains required to achieve this are found as follows:

$$L = place(A', Ce', Obs\_Poles)'$$

$$L = \begin{bmatrix} 42.64 & 1.0447 \\ 459.17 & -58.83 \\ -0.68 & 42.02 \\ -20.66 & 439.88 \end{bmatrix}$$

3. Simulink model

The model used in simulation process is shown in figure 3.18, this model includes a robust tracking state feedback and an extended observer for states estimation.

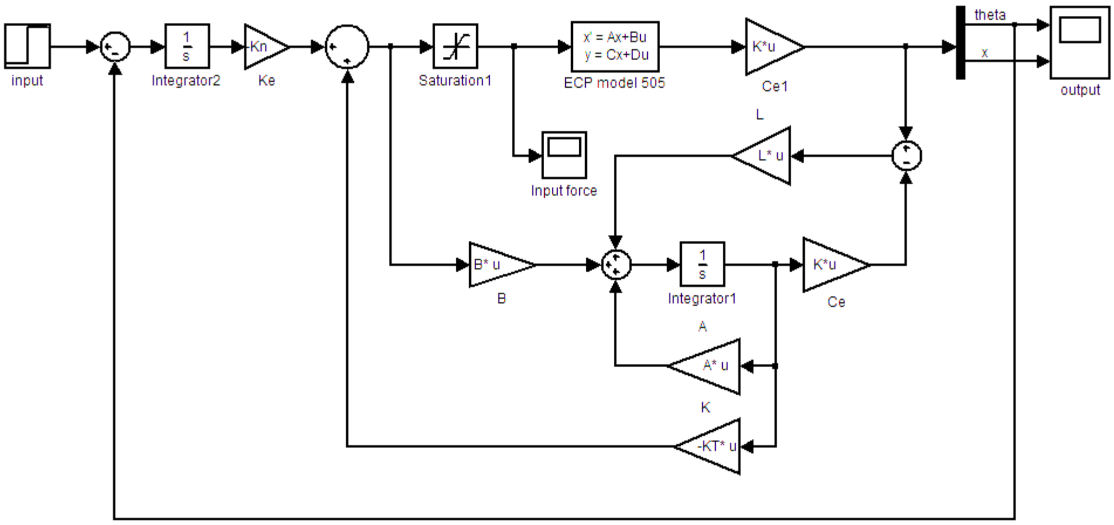

Figure 3.19: Simulink model Tracker with an observer

4. Simulation results

The initial pendulum rod angle = 0.0 rad and the desired angle of the pendulum rod equals to 0.2 rad; so the simulation results as follows:

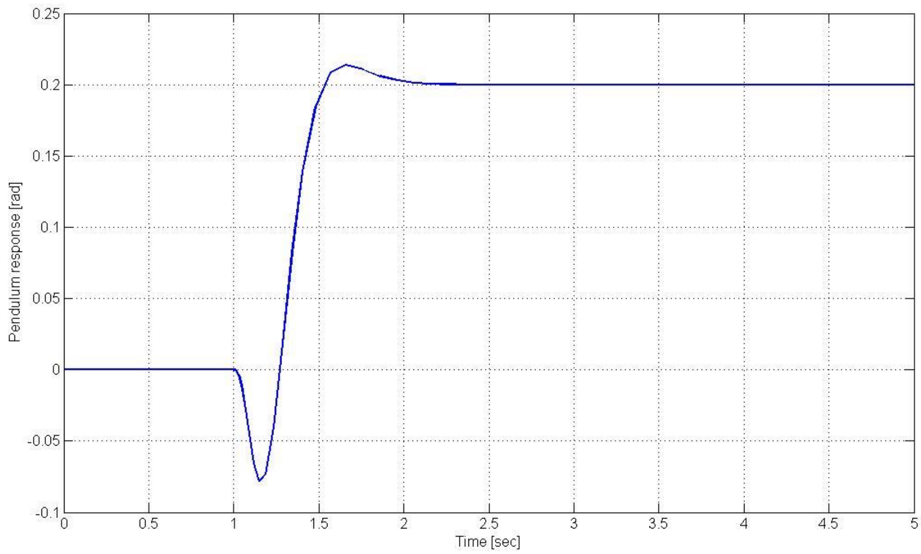

Figure 3.20: Pendulum response (Observer)

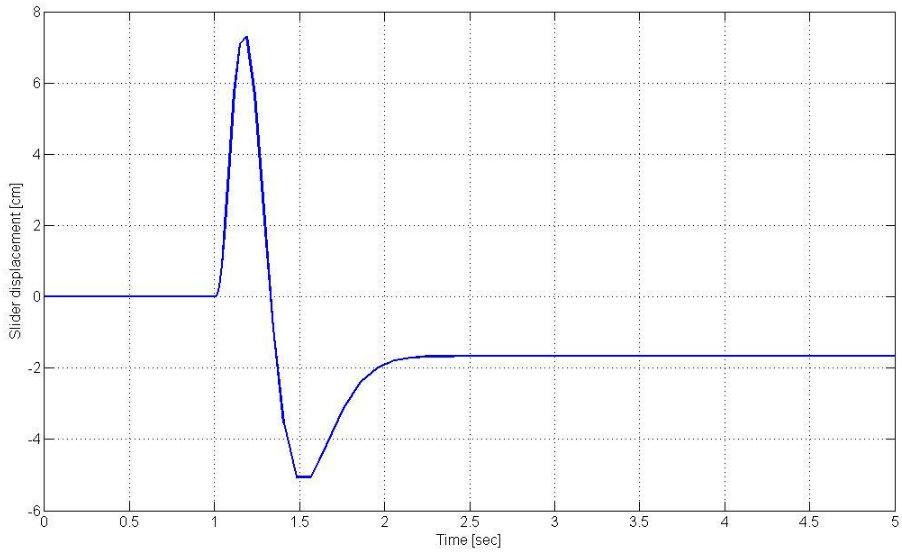

Figure 3.21: Slider displacement (Observer)

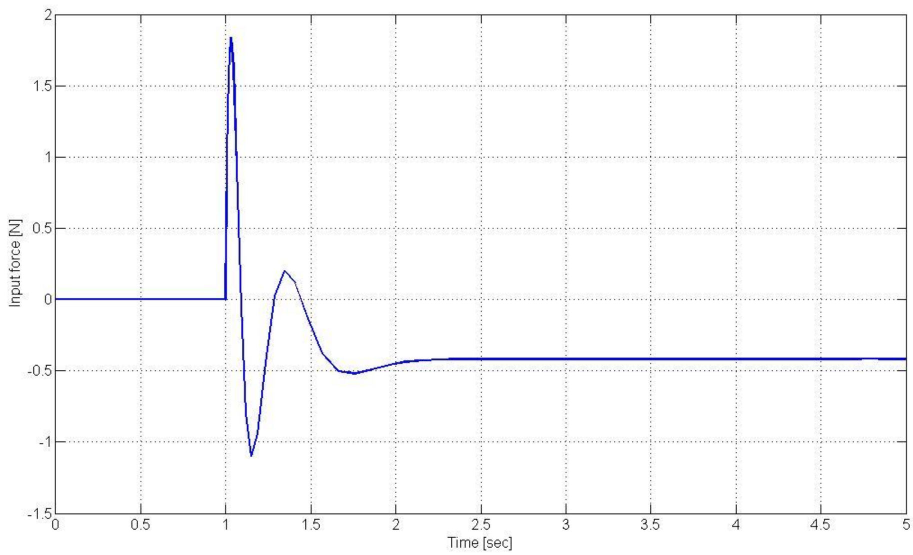

Figure 3.22: Input force (Observer)

As noticed from the results the observer works correctly, because the results are the same as tracker controller.

3.2.2.4 Discrete controller

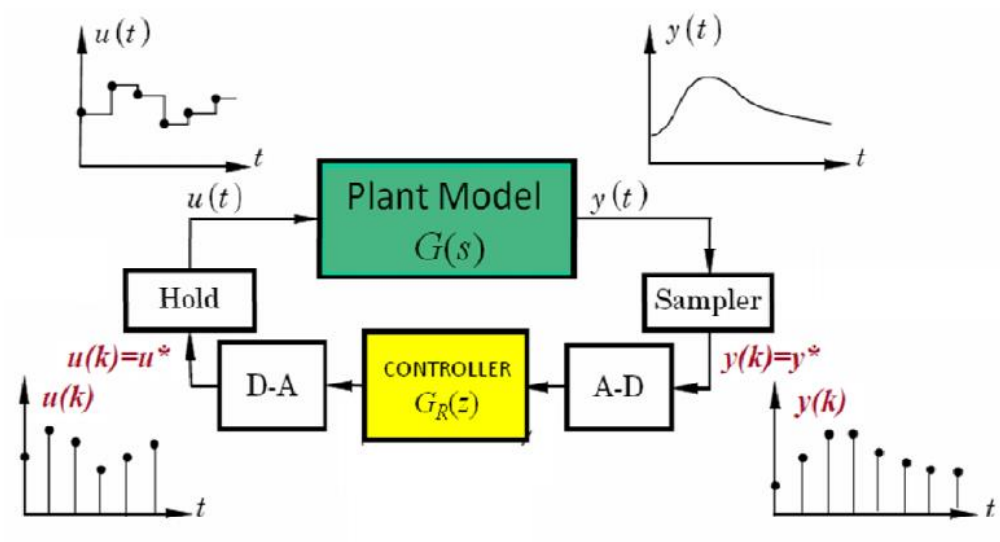

Figure 3.23: Discrete controller

The challenge here is to build a discrete controller that stabilizes the Inverted Pendulum, so the Digital (Discrete) control is a branch of control theory that uses computers to act as the controllers. Since the computer is a discrete system, the Laplace transform is replaced with the Z-transform. However, the main advantages of digital control are there flexibility, adaptability (parameters of the program can change with time).

1. Principle:

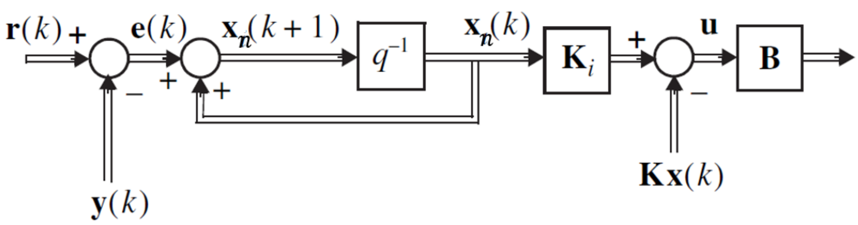

Figure 3.24: Discrete time state feedback with integration

To drive the state equations for the discrete integral controller consider Figure (3.24), thus the set of equations for the overall system as follows:

$$\boldsymbol{x}(k+1) = \boldsymbol{Fx}(k) + \boldsymbol{Gu}(k) \qquad (3\text{-}16a)$$

$$y(k) = \boldsymbol{Cx}(k) \qquad (3\text{-}16b)$$

Where: $F = I + TA$

$G = TB$

$T: sampling\ time\ (0.004\ s)$

Control low for integral controller

$$\boldsymbol{u}(t) = -\boldsymbol{kx}(t) + k_I x_n(t) \qquad (3\text{-}17)$$

Rewrite equation 3-17 discrete form, becomes

$$\boldsymbol{u}(k) = -\boldsymbol{kx}(k) + k_I x_n(k) \qquad (3\text{-}18)$$

$$\dot{x_n} = [0]x_n + r(t) - y(t) \qquad (3\text{-}19a)$$

Now convert the equation 3-19a into discrete form

$$x_n(k+1) = x_n(k) + r(k) - y(k) \quad \text{(3-19b)}$$

Where $y(k) = \boldsymbol{Cx}(k)$, so equation 3-19b become

$$x_n(k+1) = x_n(k) + r(k) - \boldsymbol{Cx}(k) \quad \text{(3-19c)}$$

Equations 3.16a and 3-19c can be written in compact form as:

$$\begin{bmatrix} \boldsymbol{x}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u}(k) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{r}(k) \quad \text{(3-20a)}$$

Or

$$\boldsymbol{x_e}(k+1) = \boldsymbol{F_e x_e}(k) + \boldsymbol{G_e u}(k) + \boldsymbol{Er}(k) \quad \text{(3-20b)}$$

Where: $\boldsymbol{F_e} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{I} \end{bmatrix}$, $\boldsymbol{G_e} = \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix}$, $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}$

Inserting the feedback law

$$\boldsymbol{u}(k) = -[\boldsymbol{k} - k_i] \begin{bmatrix} \boldsymbol{x}(k) \\ x_n(k) \end{bmatrix} = -\boldsymbol{K_e x_e}(k)$$

Gives finally,

$$\boldsymbol{x_e}(k+1) = (\boldsymbol{F_e} - \boldsymbol{G_e k_e})\boldsymbol{x_e}(k) + \boldsymbol{E}r(k) \quad \text{(3-21)}$$

Where $(\boldsymbol{F_e} - \boldsymbol{G_e k_e})$: system matrix of the closed loop system.

2. Controller

In analog controllers, the poles are put in the left half of S-plane to stabilize the system, but when dealing with discrete controllers the poles are placed inside a circle at the origin of Z-plane with radius equals to one.

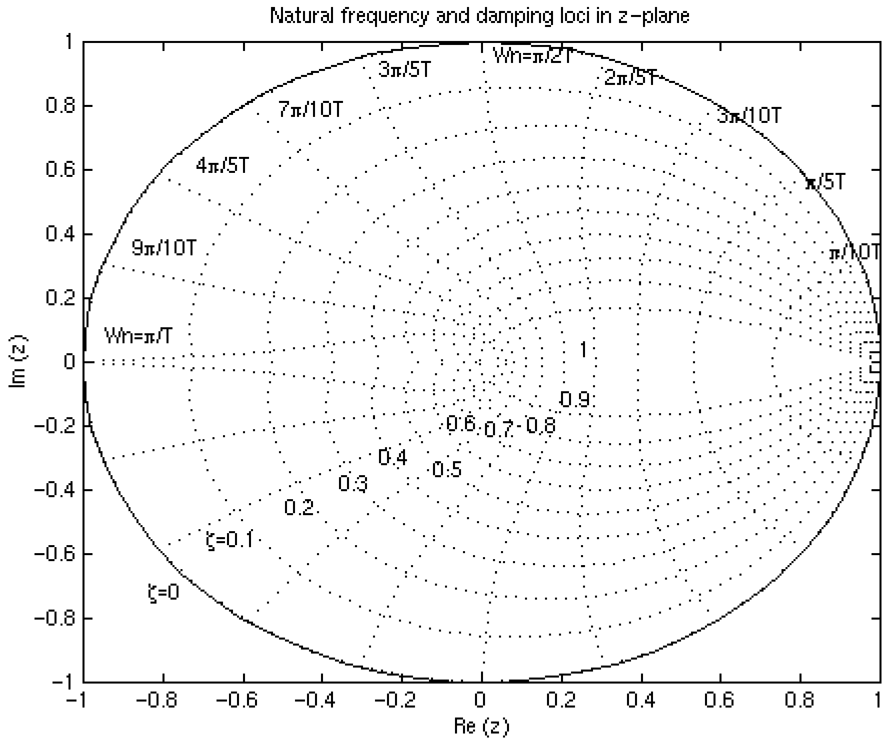

Figure 3.25: Natural frequency and damping loci in z-plane

However to be able to design the controller the controllability of the pair (F_e, G_e) must checked, this could be performed using MATLAB as follows:

$C_{med} = ctrb(F_e, G_e)$

$R_{ced} = rank(C_{med})$

Which is found to be 5, full rank, meaning that the system is fully controllable, the gains $[K\ -K_I]$ vector is found by using place command as follows:

- Select the location of the poles in the z-plane

$$des_{poles} = [0.62 \quad 0.958 \quad 0.998 \quad 0.984 + 0.0047i \quad 0.984 - 0.0047i]$$

- Thus $k_e = place(F_e, G_e, des_{poles}) = [110 \quad 45 \quad 371 \quad 68 \quad -0.099]$

$k = k_e(1{:}4)$

$k_i = -k_e(5)$

3. Simulink model

The model used in simulation process is shown in figure 3.26, this model includes a robust tracking state feedback.

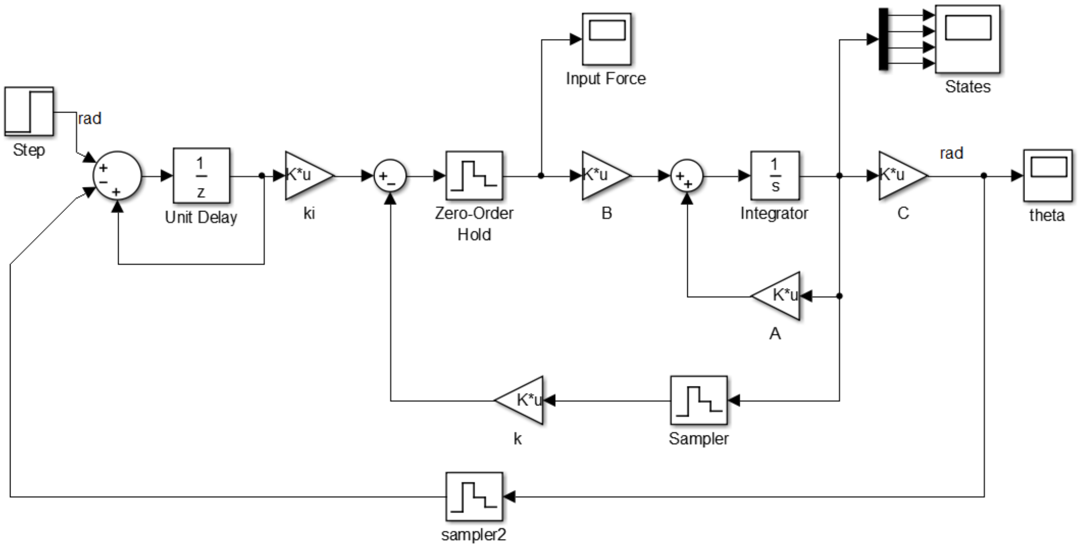

Figure 3.26: Simulink model for discreet tracker

4. Simulation results

The initial pendulum rod angle = 0.0 rad and the desired angle of the pendulum rod = 0.4 rad; so the simulation results as follows:

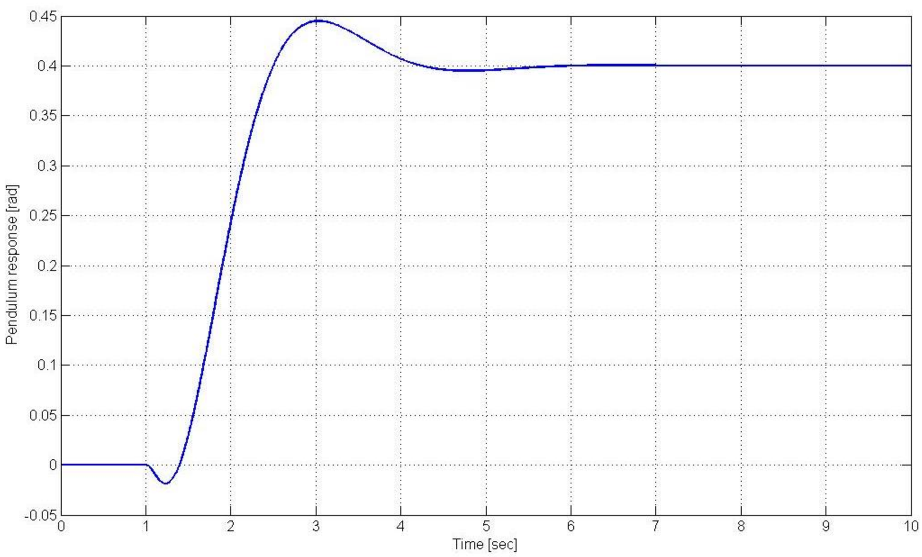

Figure 3.27: Pendulum response (Discreet Tracker)

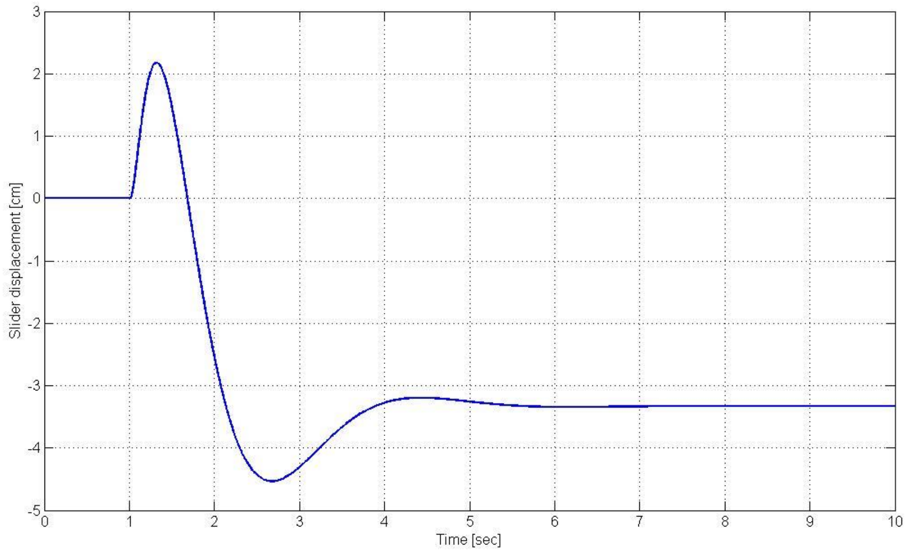

Figure 3.28: Slider displacement (Discreet Tracker)

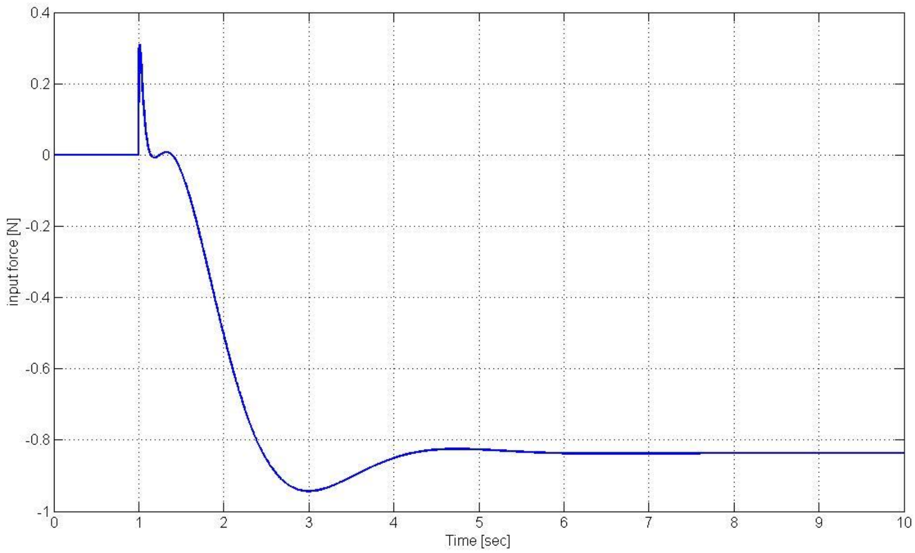

Figure 3.29: Input force (Discreet Tracker)

Figure 3.27 shows pendulum response and the percent overshoot and settling time equals to 12.3% and 3.9 seconds respectively. From the same figure it noticed that as the response slow down the negative overshoot decreases.

4 Experimental results

4.1 Introduction

PID and state feedback controllers which discussed in the previous chapter, are applied and tested on ECP model 505 Inverted Pendulum. The PID controller applied only on the slider when the pendulum rod fixed at its inverted position, just to check the devise encoders and the dc servo motor, it was impossible to apply this controller at the whole system; because of its nonlinearities. On the other hand state feedback controllers have the ability to stabilize such a system on its inverted position, even more to track any position within the limits. The effects of poles locations, gain values, input saturation and disturbances are studied and experimented on the device.

In this chapter, xPC target technology is demonstrated, and then PID controller, robust tracking and disturbance rejection state feedback controllers are applied to the IP, and the experimental results are demonstrated and discussed for each case.

4.2 xPC Target Technology

The inverted pendulum is usually used for educational and research purposes for testing the controllers and different control techniques since the system is unstable and has a fast dynamic behavior it is difficult to achieve real time control by using ordinary PCs, so the x-PC target technology is used to accomplish the real time control. The xPC target is a solution for prototyping, testing, and deploying real-time systems using standard PC hardware. In this technique two PCs are used, host and target, with the host PC, one can design the controller, simulate it, and download it to the target computer for hardware-in-the-loop (HIL) simulation. The target PC, which is connected to the controlled plant, is just used to run control functions in real-time and monitor the controlled application [7].

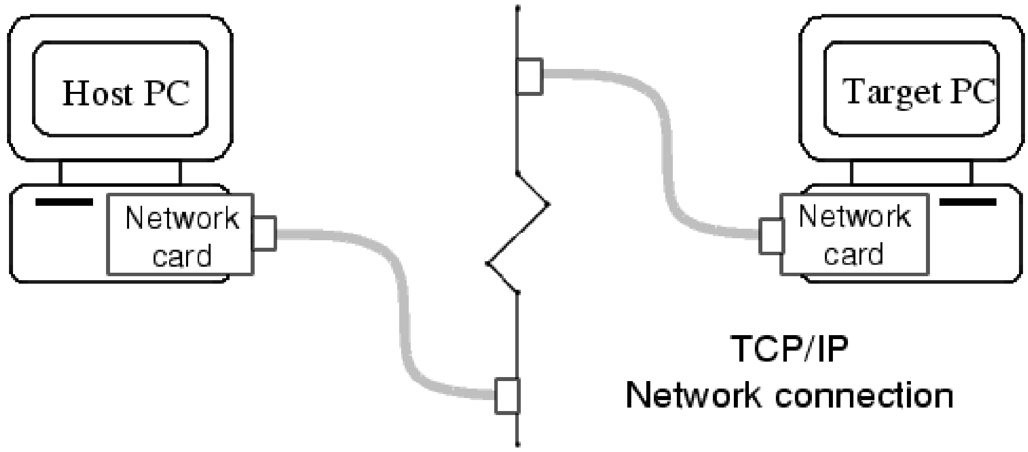

Figure 4.1: xPC target technology

With the help of MATLAB, Simulink and xPC target technique, one can design, simulate and easily modify the controller for target application, and run the controller in real time. The two PCs (host and target) and there peripheral such as DAQ cards (PCI-6024E DA and PCI-6602 ENC) are already exists in control lab, so we used them to apply and test the controllers on the plant "ECP model 505".

4.3 Controller design methodology

The design steps of the controller could be summarized as:

1. Design the controller using MATLAB and Simulink.
2. Simulate the model to check its response to apply any necessary improvements before applying the controller to the plant.
3. Create the target application by combining the real time workshop, xPC target and C-compiler.
4. Execute the target application in real time.

5. Monitor the target application using the xPC target scope and save the signals to a file for later use.
6. Tune parameters, after the controller has been built and re-download to the target PC, xPC target technique permits an online modification for some controller's parameters, such as gain values and sampling time.

4.4 Experimental results

As state before, one of the challenges that face the control problem of inverted pendulum is that the controller effort must not exceed the limits of the device, moreover some of system states are not directly measured, like velocities of the slider and pendulum rod. However in order to be able to implement the state feedback controller practically, the unmeasured states should be estimated. This can be achieved by using either state observers or by numerical differentiation methods.

4.4.1 Hardware Gain

Since the controller output is a force (Newton) and the motor which drives the sliding rod needs voltage, the IP will not work as desired. To achieve the desired response; the output force must convert to voltage. So the Hardwar gain is the voltage potential generated from effecting of 1 Newton force.

There are a lot of methods to calculate the hardware gain, one of them is to make the sliding rod perpendicular to the ground, Give the system electrical potential to move the slider, when the slider begins to move we take the voltage and calculate the Hardware Gain. The voltage in this case is the critical voltage, which equals to $0.6\ v$ and the total mass of the slider is $0.213\ kg$; so the hardware gain is equals to 3.4825 N/v. This result we found is near the actual value (actual value equal 4.2 N/v). Thus we use the actual value.

4.4.2 PID controller (Experimentally)

In this experiment, it is desired to design PID controller for the sliding rod only using theories in chapter three.

Control system specifications

- Uncompensated system poles [0 0]
- System zeros None
- Desired response $\%OS = 10\%, settling\ time\ 0.15\ s$
- Gain values $K_d = 11.25 \quad K_p = 422 \quad K_i = 86$

Here, the closed-loop is selected to obtain under damped transient response, with performance specification as shown in table 4.1. The m-file used for this experiment is available at appendix A.1.

Table 4.1 Performance specifications

	Design	**Simulated**	**Actual**
%Over Shoot	10	10.22	0
Settling time (sec)	0.15	0.138	0.148
Steady stat error	0.0	0.0	0.0

Experimental results are shown in the following figures, including the Simulink model, the input force command and final response.

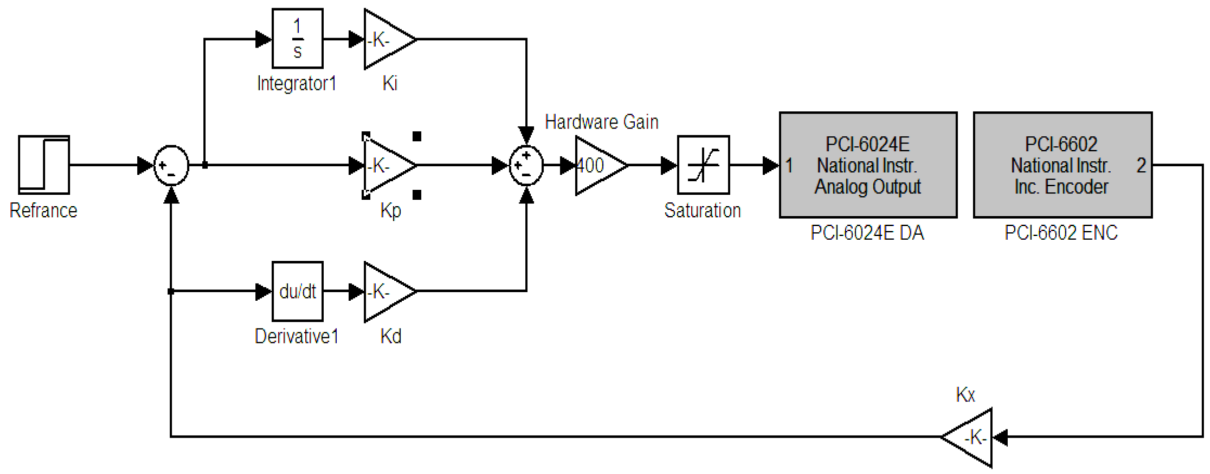

Figure 4.2: Slider controller Simulink model

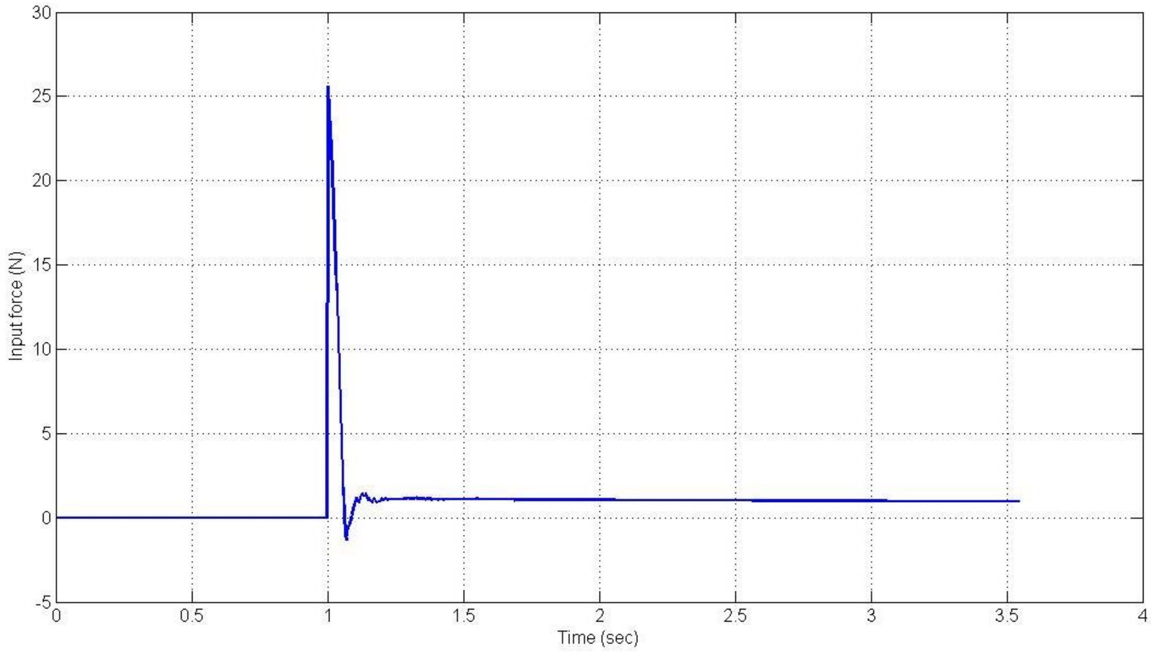

Figure 4.3: Input force (PID experimentally)

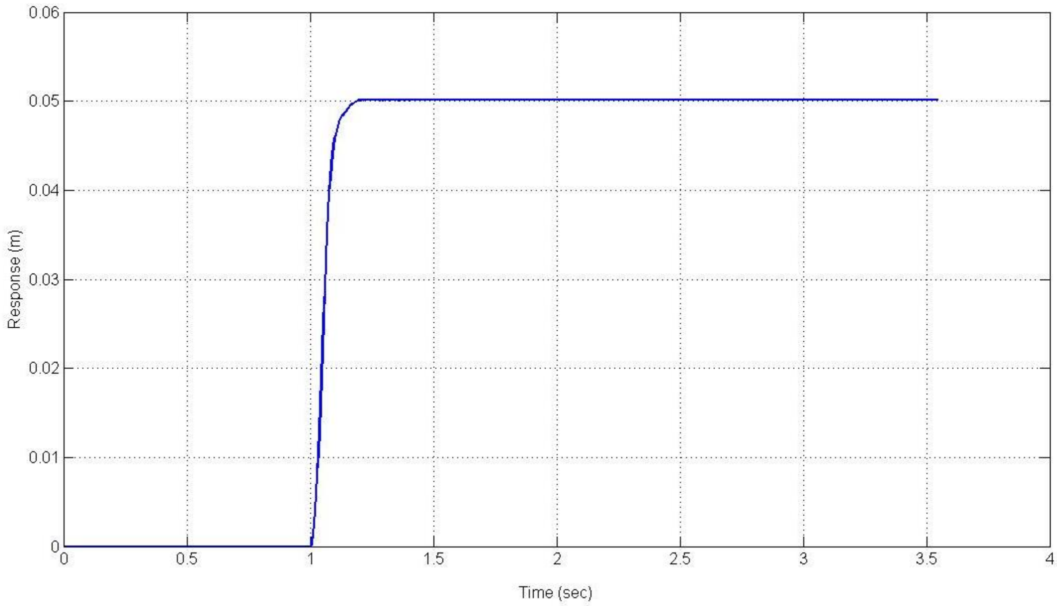

Figure 4.4: Slider response (experimentally)

From the previous results it could be noted that:

- The deviation between the actual and desired results comes from the dry friction force that acts on the slider. Anyway this damping type is nonlinear, and it was ignored when the system was linearized.
- The oscillation that occur at steady state (clearly appeared in position signals) are a result of the steady state error that near to zero; due to the encoder's resolution.

4.4.3 Regulator controller (Experimentally)

In this experimental, it is desired to stabilize the inverted pendulum at its inverted position ($\theta = 0$) using state feedback control theories discussed in chapter three.

$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{BK})\boldsymbol{x} + \boldsymbol{Br}$ $\quad$ $\boldsymbol{y} = \boldsymbol{Cx}$ $\quad$ $\boldsymbol{x} = [\,\theta \quad \dot{\theta} \quad x \quad \dot{x}\,]^T$

Control system specifications:

- Uncompensated system poles $3.5360 \quad -3.5360 \quad 6.7530i \quad -6.7530i$
- System zeros $5.4643 \quad -5.4643$
- Controlled system poles $[-9 \pm 4.36i \quad -13.33 \pm 5.23i\ ]$
- Feedback gains $\boldsymbol{K} = [158.3 \quad 28.6 \quad 68.1 \quad 19.8]$

Here, the closed-loop is selected to obtain under damped transient response. The m-file used for this experiment is available at appendix A.2. Experimental results are shown the response of IP to disturbances in the following figures, including the Simulink model, pendulum response, slider response and the input force command.

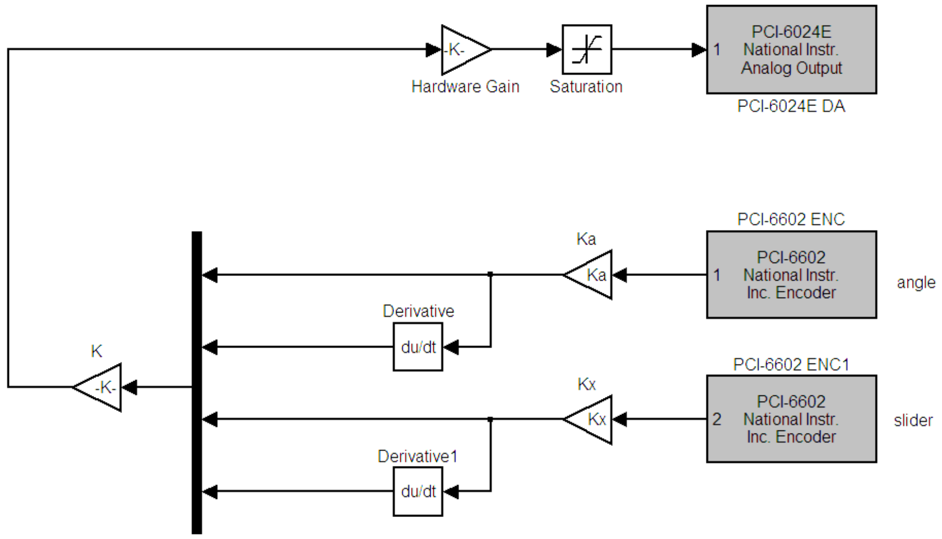

Figure 4.5: Regulator controller Simulink model

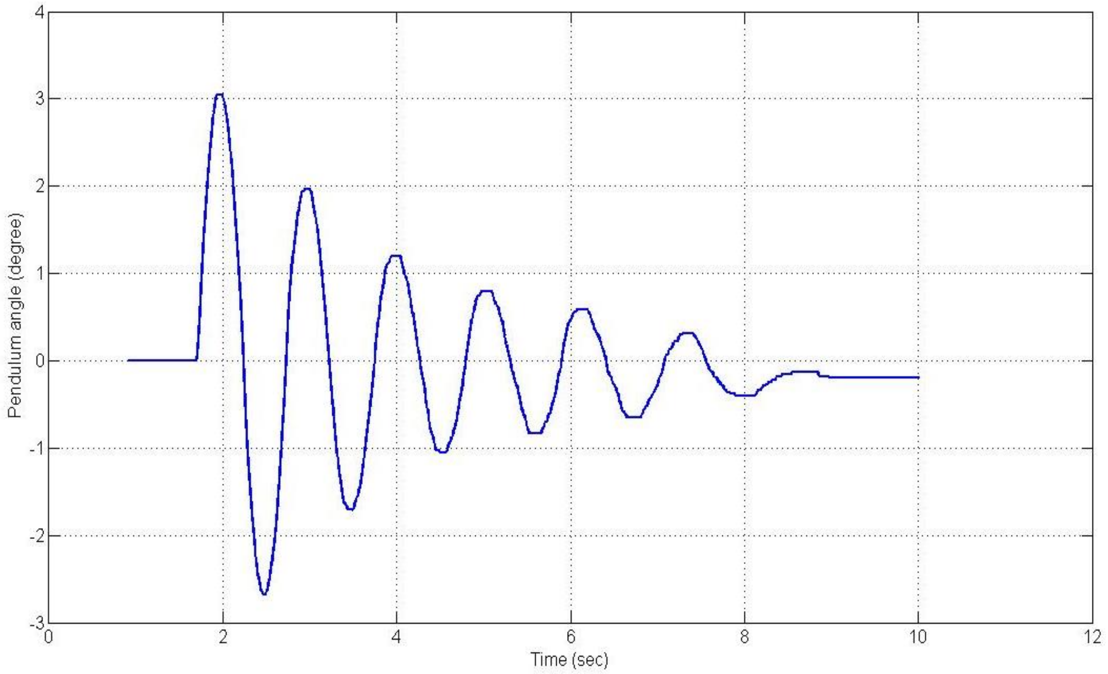

Figure 4.6: Actual pendulum response to disturbances

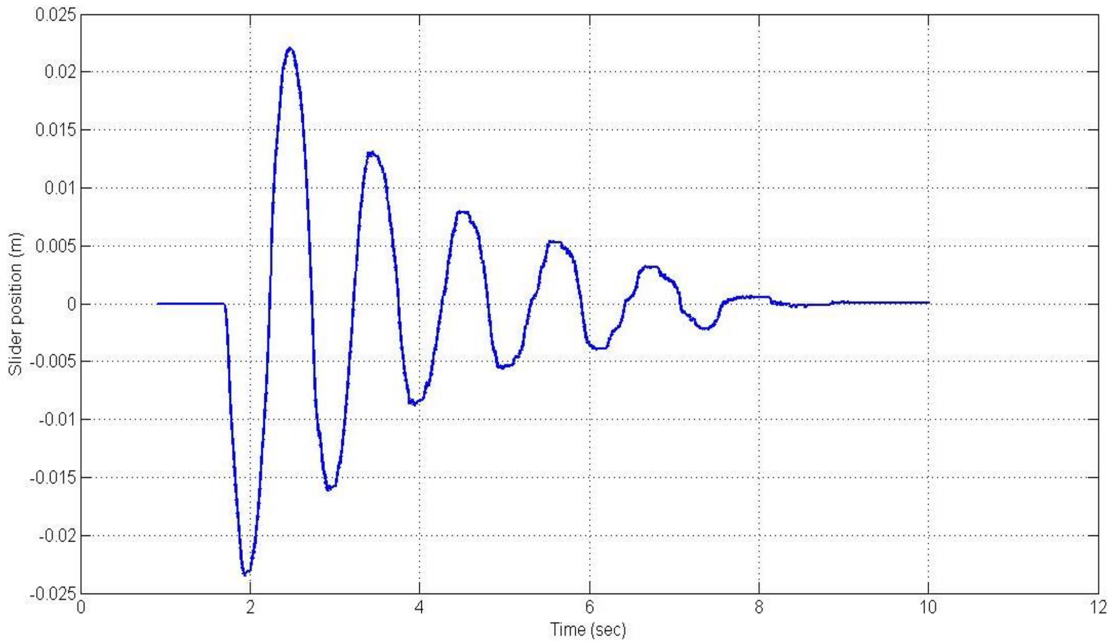

Figure 4.7: Slider position output

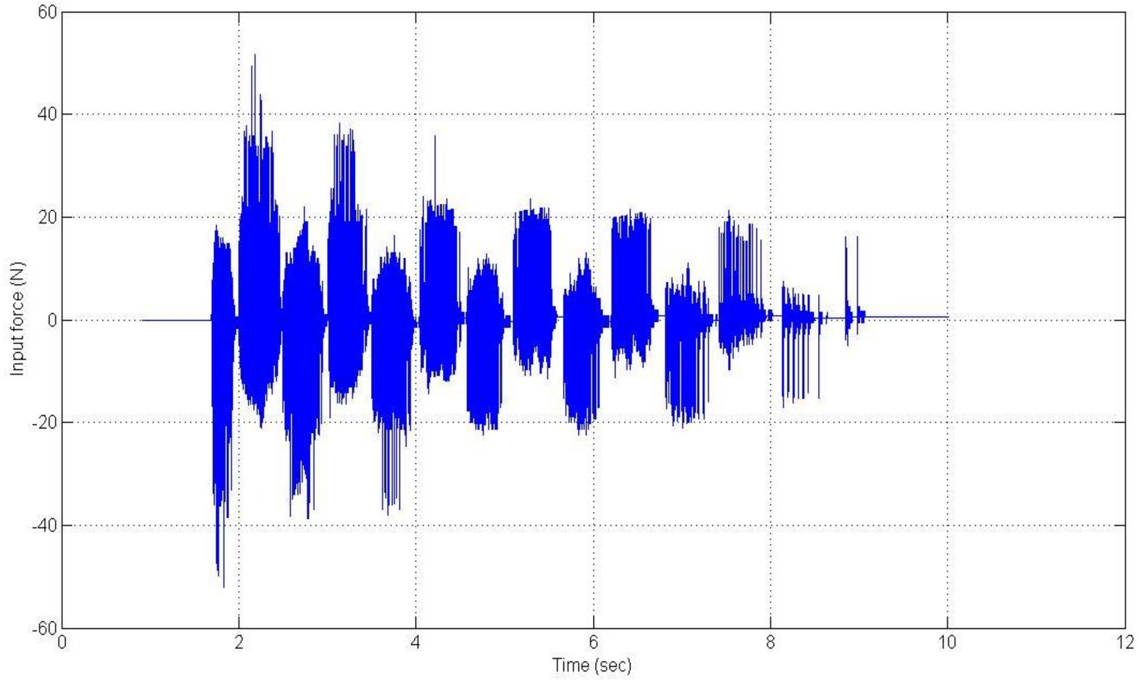

Figure 4.8: Experiment output force to the motor.

From the results, it could be noted that:

- The maximum input force to the inverted pendulum about 50N experimental.
- The controller rejects disturbances and manage to stabilize the IP at its inverted position.
- By increasing the gain value it's possible to obtain faster response but this will require more force to use and may be exceed the limits of the motor and the system become unstable!
- As noticed from figure 4.6 there is a steady state error approximately equal 0.2 degree due to frictions which were neglect able when driving the model.
- The oscillations in the response because the system is under damped.
- Figure 4.7 shows the slider movement to stabilize the pendulum.

4.4.4 Robust Tracing controller (Experimentally)

In this experiment a robust controller is design to track a step position command for the pendulum rod and reject disturbances acting on the system.

From chapter three:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}-\boldsymbol{BK} & \boldsymbol{B}K_e \\ -\boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ x_n \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} r$$

$$y = [\boldsymbol{C} \quad 0] \begin{bmatrix} \boldsymbol{x} \\ x_n \end{bmatrix}$$

Control system specifications

- Controlled system poles $[-9 \pm 4.36i \quad -13.33 \pm 5.23i \quad -12]$
- Feedback gains $K_{track} = [501 \quad 94.7 \quad 305 \quad 74 \quad -629]$

$K = K_{track}(1:4)$ $K_e = K_{track}(5)$

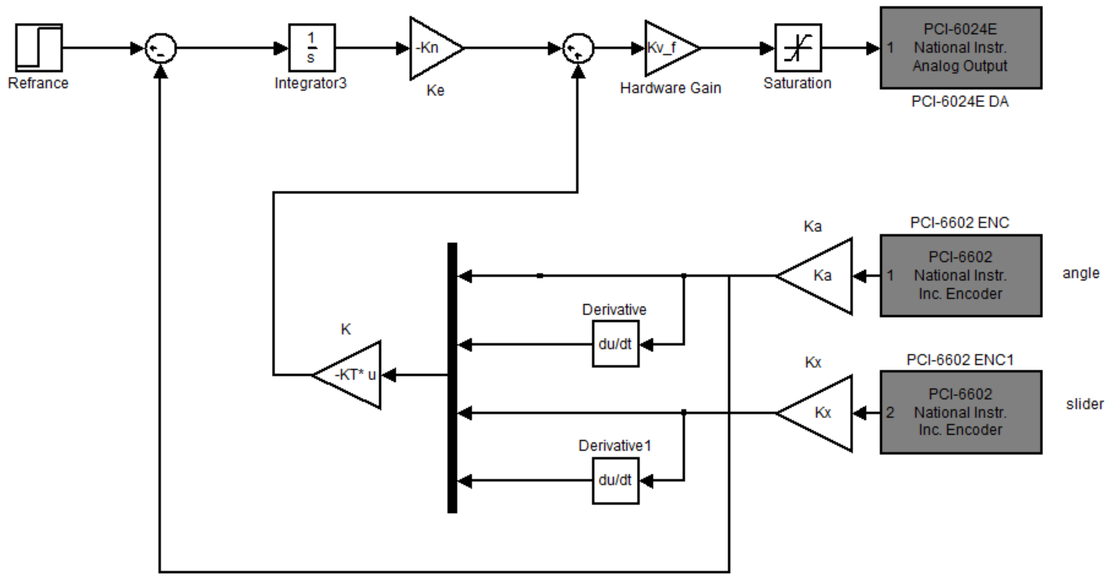

Figure 4.9: Experiment Simulink model for the tracker

Experimental results are shown in following figures:

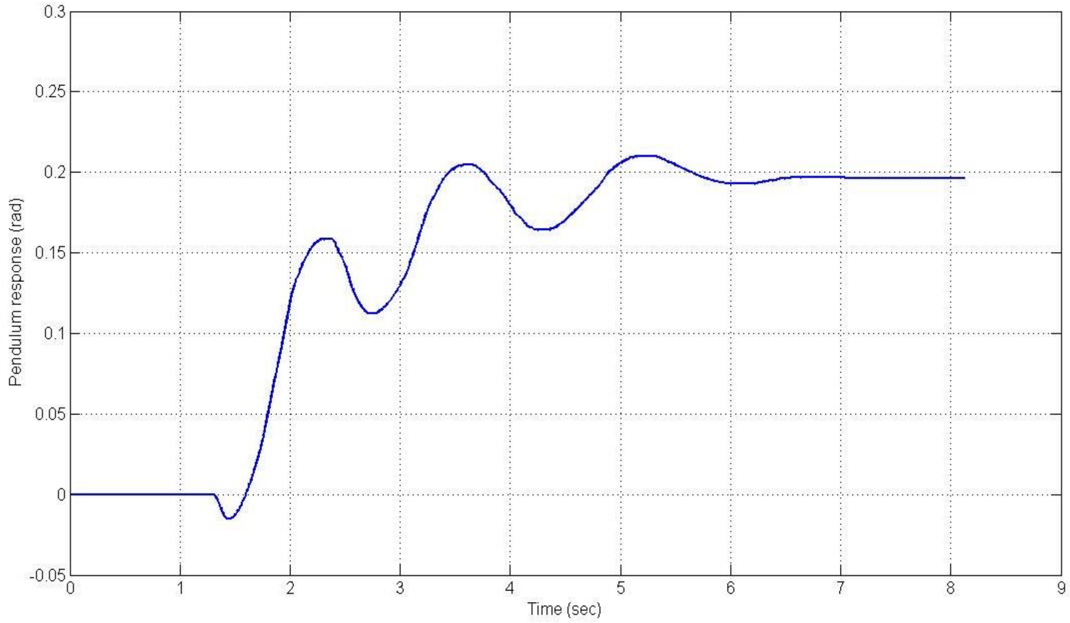

Figure 4.10: Pendulum response for the tracker (experimentally)

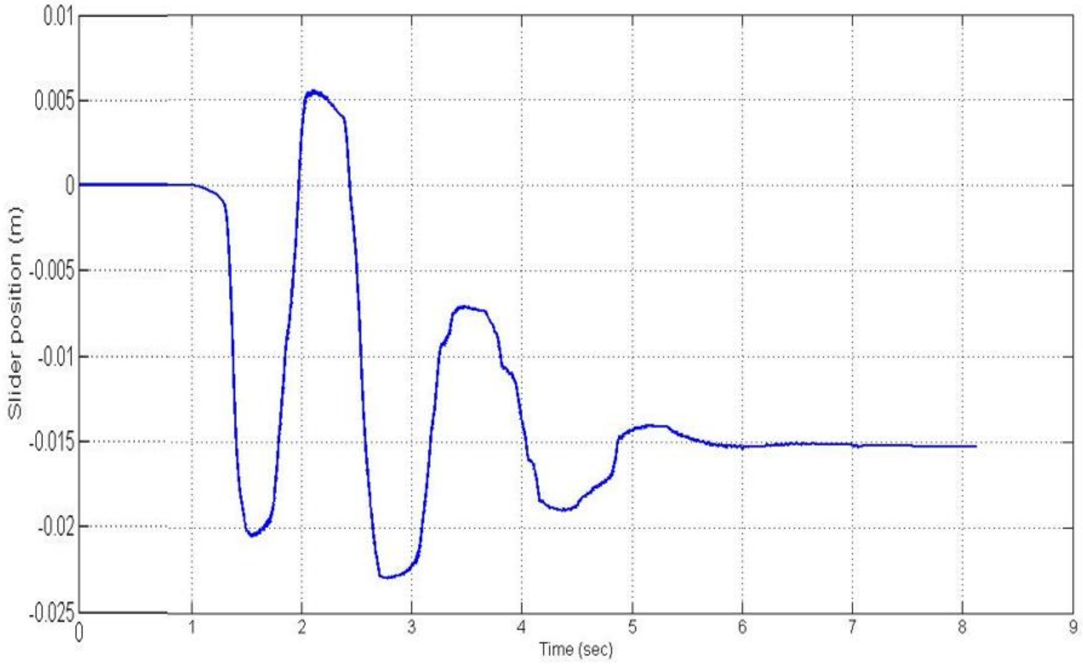

Figure 4.11: Slider response for the tracker (experimentally)

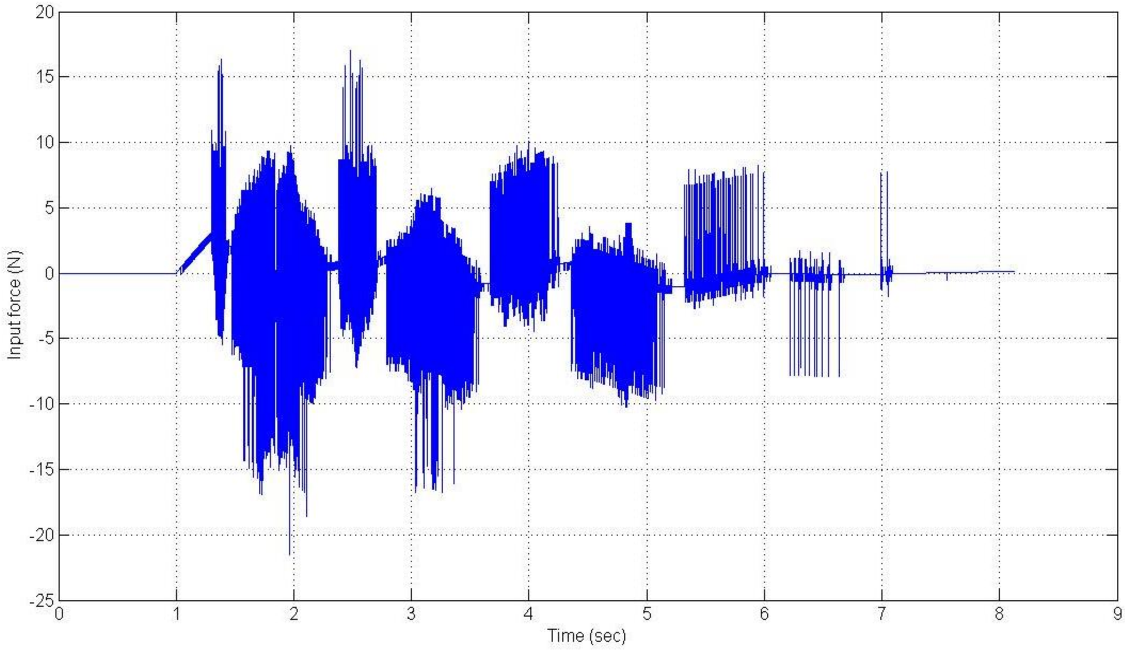

Figure 4.12: Input force (Tracker experimentally)

From the results, it could be noted that:

- The controller was able to track our desired position, but as noticed from figure 4.10 the response is slow (slower than the simulation); because of tuning the value of K_e to 53, since the old value was too large and tend to make the system unstable.
- As notices from figure 4.10 there is a negative over shoot, because of non-minimum phase (right half plan zero)
- The force is larger than the simulation due to dry friction and when the system tracked its position the force goes to zero because of friction force which can hold the slider from slipping. So the IP is statically stable at that position.
- Figure 4.11 shows the response of the slider in meters.

4.4.5 Discreet Tracing controller (Experimentally)

After applying robust tracking analog controller, it is time to digitalize it in the z-plane, and design the controller to track a step position.

Recalling from chapter three:

$$\begin{bmatrix} \boldsymbol{x}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u}(k) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{r}(k)$$

Control system specifications

- Controlled system poles in Z-plane
  $[0.62 \quad 0.958 \quad 0.998 \quad 0.984 + 0.0047i \quad 0.984 - 0.0047i]$
  these locations are found by using the optimal method.
- Feedback gains $K_e = [110 \quad 45 \quad 371 \quad 68 \quad -0.099]$

  $k = K_e(1{:}4)$ $\qquad k_i = K_e(5)$

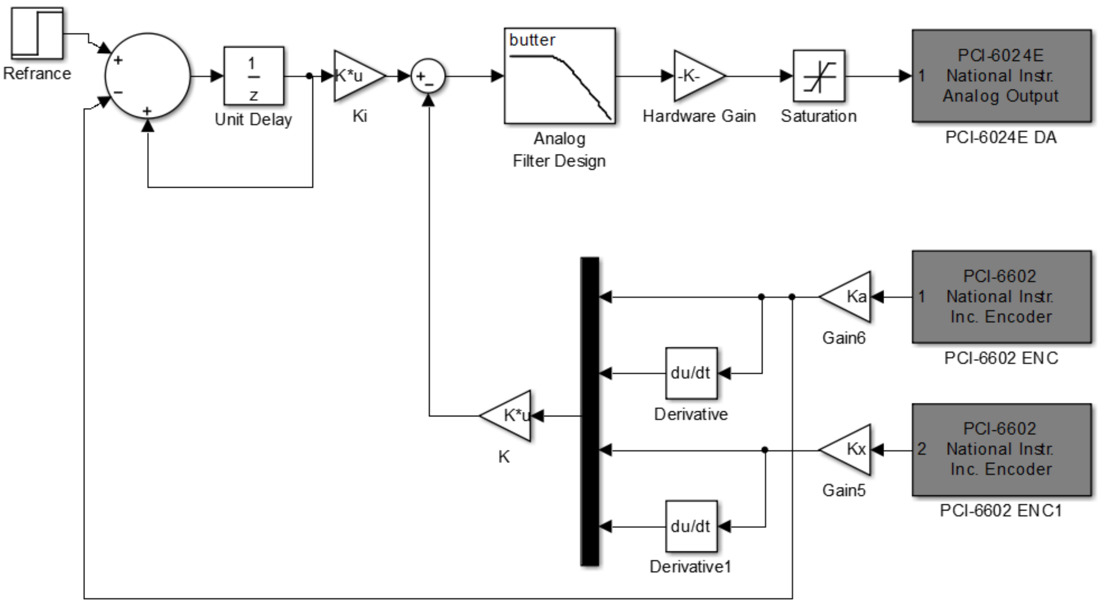

Figure 4.13: Experiment Simulink model for the discrete tracker

Experimental results are shown in following figures:

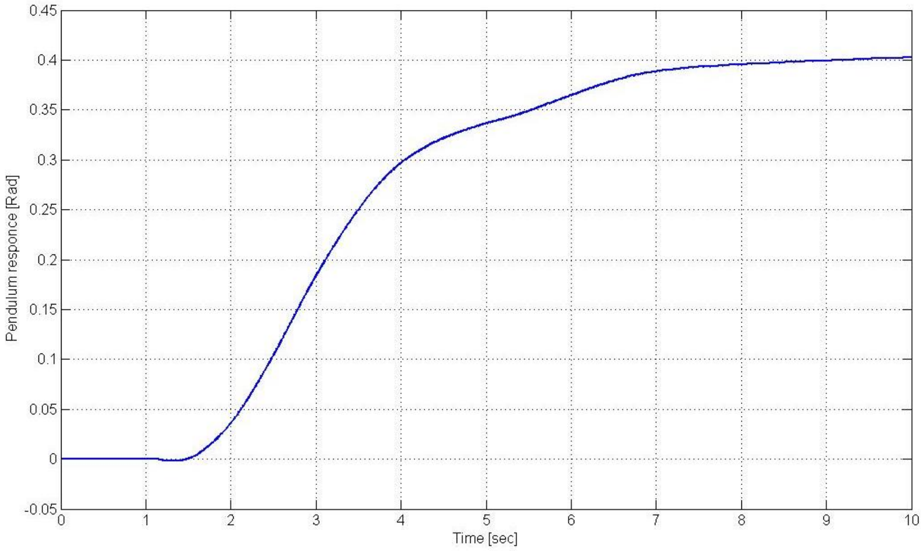

Figure 4.14: Pendulum response for the discreet tracker (experimentally)

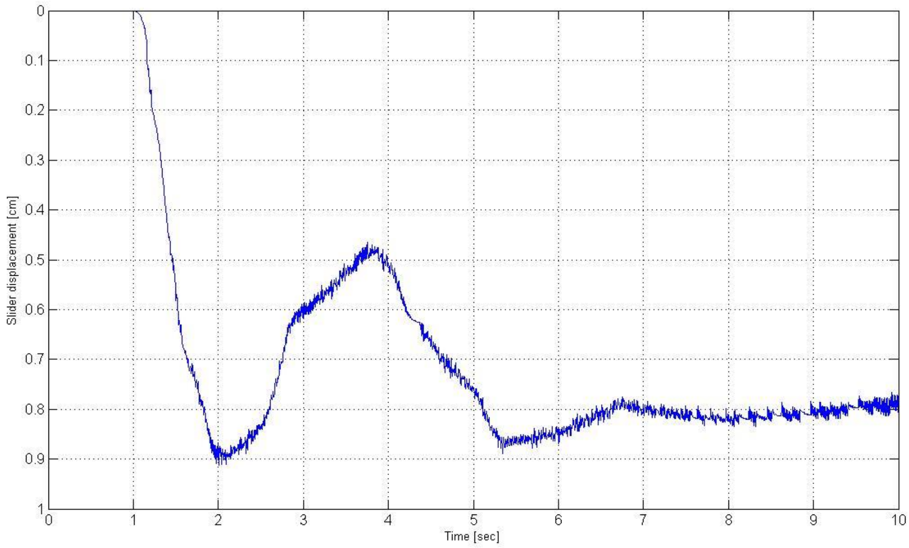

Figure 4.15: Slider response for the discreet tracker (experimentally)

From the results, it could be noted that:

- As noticed from figure 4.14 the percent overshoot and settling time 0.0% and 7 seconds respectively.
- To achieve 0.4 rad for the pendulum rod, the slider must move 0.8 centimeter.
- In final response of the pendulum rod there is a small error because of nonlinearities of the IP.
- The response is slower than simulation due to the tuned value of $k_i = 0.095$.

5 Embedded system, Design and control

5.1 Introduction

The next step of the project is to replace the x-pc target technology with a microcontroller. Microcontrollers are designed for embedded applications, so it is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.

5.2 Microcontroller

There were many choices to use, like PIC micro controller from microchip or Atmel microcontrollers, but in this project Arduino board based on Atmel is used; because its open source and one can deal with it easily, thus the Arduino Mega 2560 has been chosen to control the inverted pendulum.

Figure 5.1: Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 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 to get started [6].

For the software environment, the Arduino 1.5.6-r2 software used to verify and download the code on Arduino board.

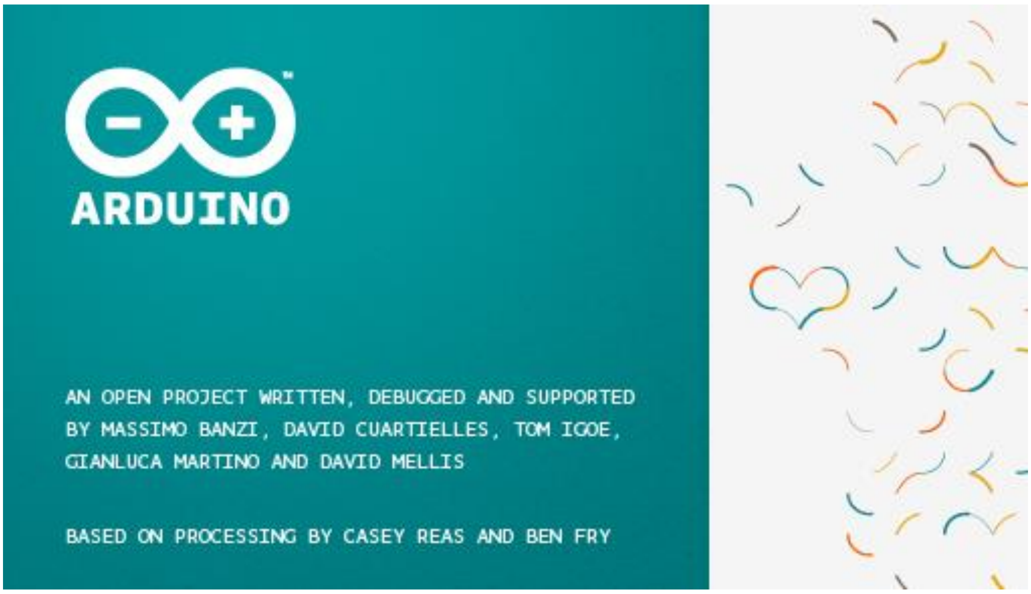

Figure 5.2: Arduino 1.5.6-r2

5.3 Controller

In this section, the discreet controller that applied on the IP using X-pc target technique is converted to c-code and applied on the microcontroller and the same gain used.

1) Reading the encoders

Since Arduino is open source and it had a very large community of people using it for all kind of projects, you can easily find and download library for reading encoders and include it in the code.

$\#include\ < Encoder.h >$

However, there is many choices to connect the channels of the encoder on the Arduino board, but to get the best performance, you must connect the channels to pins have interrupt capability, which are pin number 2, 3, 18, 19, 20 and 21 on Arduino mega.

$Encoder\ Slider(18, 19);$

So what is interrupt?

Interrupt is an event that forced the microcontroller to suspend processing the current instruction sequence and to begin an interrupt service routine. Which helps the microcontroller to provide real time response to events in the embedded system they are controlling.

Possible interrupt sources are device dependent, and often include events such as an internal timer overflow, completing an analog to digital conversion, a logic level change on an input such as from a button being pressed or from encoder, and data received on a communication link.

For example, to read the slider encoder you can use this function

$Slider.read\ ();$

2) Calculating the speed

The challenge here is to have an accurate measurement for both pendulum and slider speeds. Therefore, the best solution was to use timer interrupt and make the interrupt time equals to 0.004 seconds, which is the sampling time. However to calculate the speed you can do as follows:

- You should store the position before one sampling time (old position)
- Subtract the current position from the old one
- Divide the subtracted vale with sampling time.

$newRod\ =\ ((float)(Rod.read()))/2546.0;$

$newSlider\ =\ ((float)(Slider.read()))/50200.0;$

$Rod\_Speed = (newRod - oldRod)/(0.004);$

$Slider\_Speed = (newSlider - oldSlider)/(0.004);$

$oldRod = newRod;$

$oldSlider = newSlider;$

Where 50200 and 2546 are constants that convert the encoder pulses to distance.

3) Controller code:

After reading the position and calculating the speed for both pendulum and slider, it is time to start design the controller.

Recalling from discreet controller section

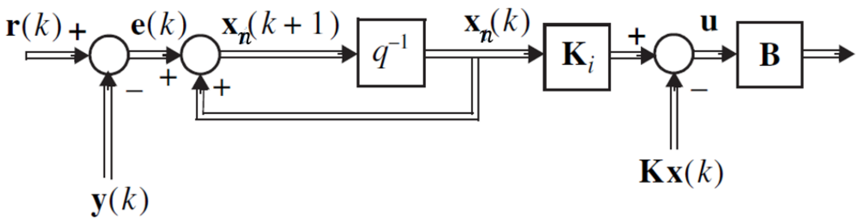

Figure 5.3: Discreet tracker

From figure 5.3:

$u = K_i x_n(k) - \boldsymbol{Kx}(k)$, (input force to the inverted pendulum) (5-1)

$x_n(k+1) = r(k) - y(k) + x_n(k)$ (5-2)

Now the gain $\boldsymbol{K}$ must be evaluated at sampling time equals to 0.004 second, so Matlab used and $\boldsymbol{K}$ found by using $place$ command as follows:

- Select a desired location of the system poles in Z domain

  $\boldsymbol{poles} = [0.62 \quad 0.958 \quad 0.998 \quad 0.984 + 0.0047i \quad 0.984 - 0.0047i]$

- Evaluate the system extended matrix in discreet form

  $\boldsymbol{F_e} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{I} \end{bmatrix}$, $\boldsymbol{G_e} = \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{0} \end{bmatrix}$ when sampling time equals to 0.004 second.

- Use $place$ command to find the gains

  $\boldsymbol{K_e} = place(Fe, Ge, poles)$

  $\boldsymbol{K_e} = [110 \quad 45 \quad 371 \quad 68 \quad -0.099]$

  $\boldsymbol{K} = \boldsymbol{K_e}(1:4)$

  $K_i = -\boldsymbol{K_e}(5)$

The goal here is to make the IP track some reference input (r) , which choose to be a potentiometer.

$r = (((float)\ (analogRead\ (A10))) - 512.0) * (0.4363/512.0);$

The potentiometer is on channel ten of the microcontroller and the above code used to convert input voltage into radians from $-0.4363\ to\ 0.4363\ rad$ ($\pm 25°$).

Equation 5-1 and 5-2 can be written in the microcontroller as shown below:

$$x_n = r - newRod + x_{np}$$

$$F = x_{np} * 0.095 - newSlider * 371.0 - Slider_{Speed} * 68.0 - newRod\ * 110.0$$
$$-Rod_{Speed} * 45.0$$

$$u = (F/4.2) * (255/5)$$

$$x_{np} = x_n$$

Where $255/5$ is a constant that converts the PWM duty cycle into output voltage.

However, the output signal (u) could be positive or negative value, so two pulse width modulation (PWM) pins were used. If u is positive number then pin 9 is activated, else pin 10 will be activated. The full code in appendix A. But the driver only receive analog signals not PWM and the microcontroller gives only positive voltage so an interfacing circuit needed to connect between the microcontroller and the driver, which explained in the coming section.

Furthermore, the Imposed direction should be the same as the real direction of the inverter pendulum or the system will surely be unstable and fails down.

4) Simulation results

The simulation results are shown in chapter three, figures 3.27 and 3.28 for the pendulum and slider respectively.

5.4 Interfacing circuit

1) Low pass filter

The low pass filter take the average of PWM, so the output is analog signal. The basic component of low pass filter are resistor and capacitor as shown in the figure below

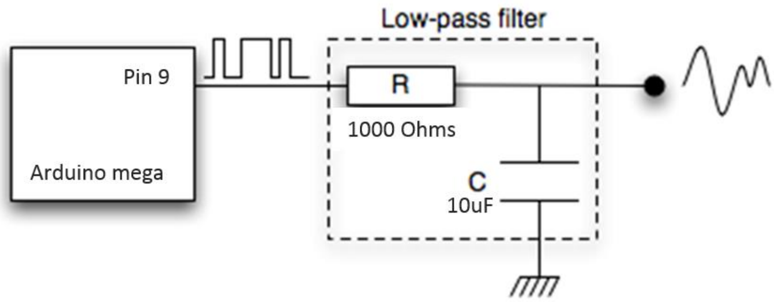

Figure: 5.4 Low pass filter

Calculations

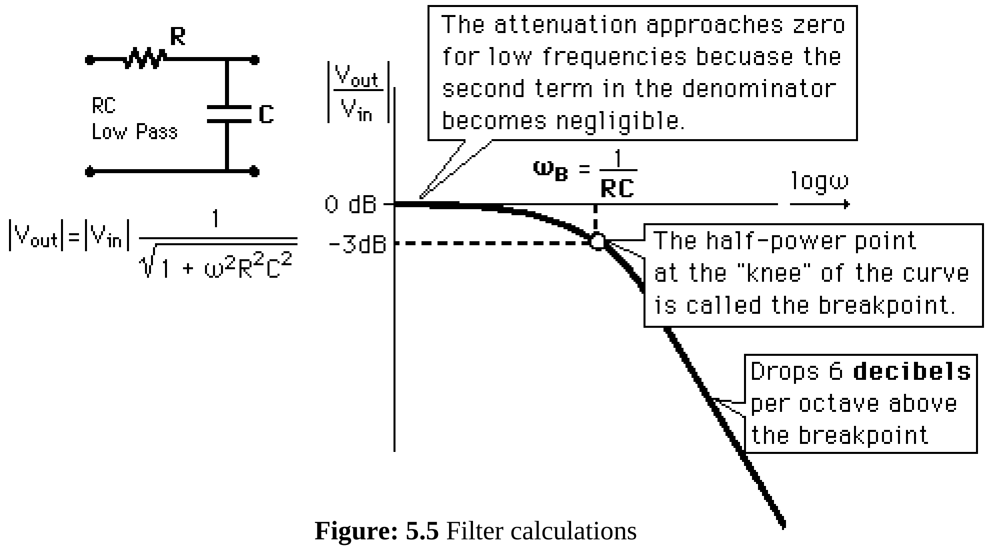

Figure: 5.5 Filter calculations

Where ω_B the breakpoint frequency.

$$\omega_B = \frac{1}{RC} = \frac{1}{1000\times10\mu\times2\pi} = 16\ Hz$$

The output frequency of PWM is approximately 500 Hz, as mentioned low pass filter take the average of input, however the filter is tested at these values and gave good results.

2) Differential amplifier

A differential amplifier is a type of electronic amplifier that amplifies the difference between two voltages but does not amplify the particular voltages. Thus, it used here to subtract the signals that comes from pin 9 and 10 after the filter. The output of a differential amplifier is given by:

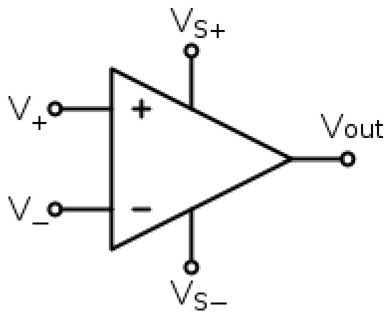

Figure 5.6: Differential amplifier

$$V_{out} = k(V_{in}^{+} - V_{in}^{-})$$

Where $V_{in}^{+} and\ V_{in}^{-}$ the input voltages and k are is the differential gain. Figure 5.7 shows the designed circuit with gain equal to one for differential amplifier.

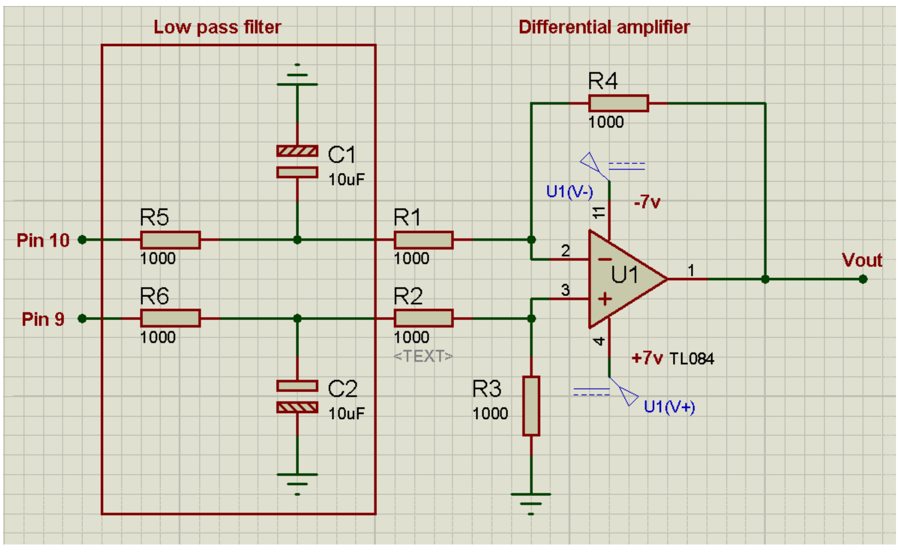

Figure 5.7: Circuit wiring diagram

5.5 Results

The complete circuit is designed as a shield, just plug it over the Arduino board and start controlling the inverted pendulum. The main advantages of this design were no missy wires and easy for diagnosing and fixing errors.

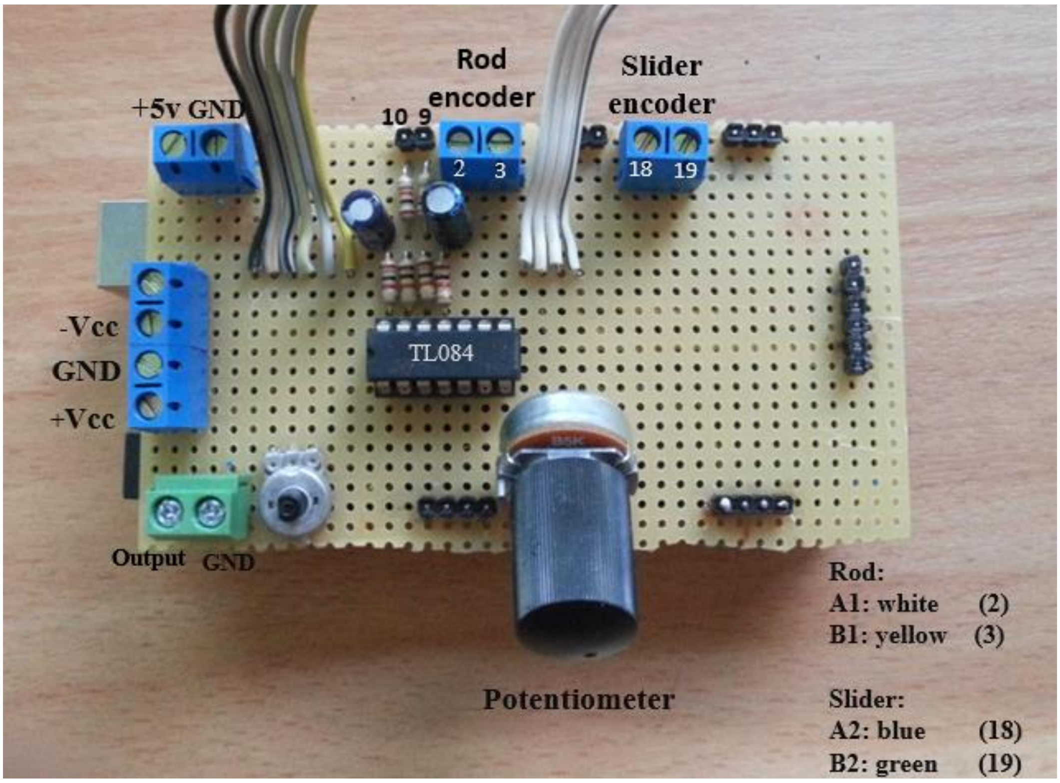

Figure 5.8: Full circuit practical

First the shield is tested as a regulator and shows it response to disturbances, after that a tracker controller applied and gave it a step command to rotate the pendulum 0.2 rad which equals to 22.92 degrees.

1) Regulator

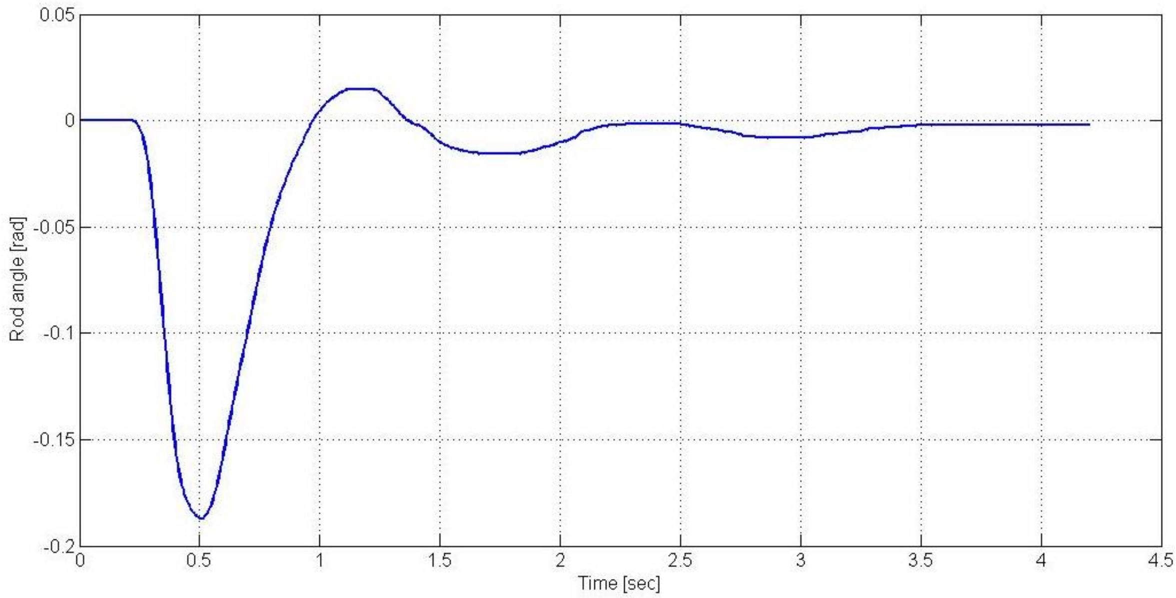

Figure 5.9: pendulum response to disturbances

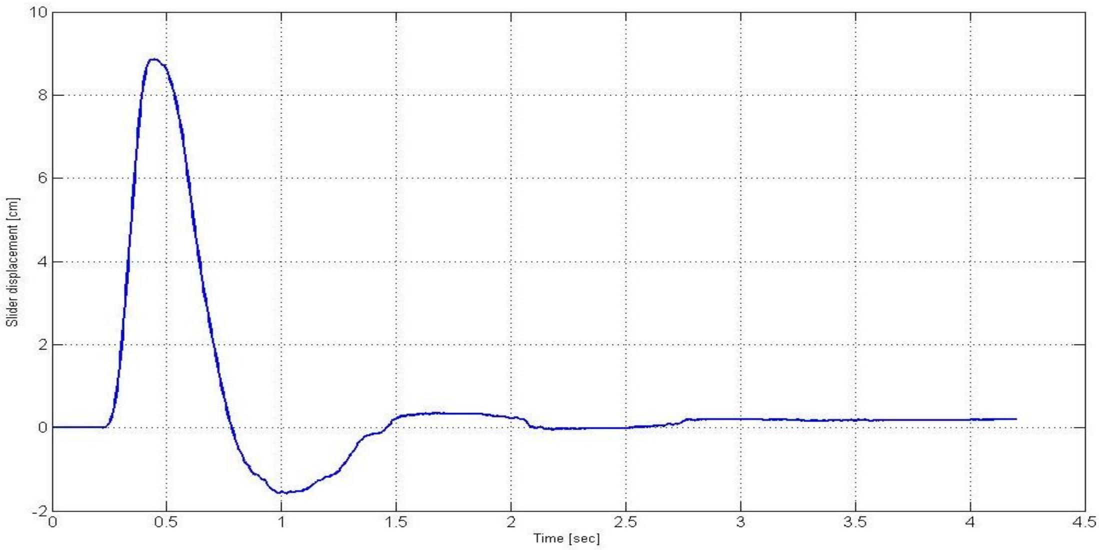

Figure 5.10: Slider response to disturbances

From the results, it could be concluded that:

- The percent over shoote is approximatly 7.75% and the setlling time is 0.372 second.
- The ocillations were less than the analog regulator which applied using x-pc target, because of the selection of poles.
- From figure 5.10 there is a steady state error approximately equal 0.135 degree due to frictions which were neglectable when driving the model.
- The controller rejects disturbances and manage to stabilize the IP.

2) Tracker

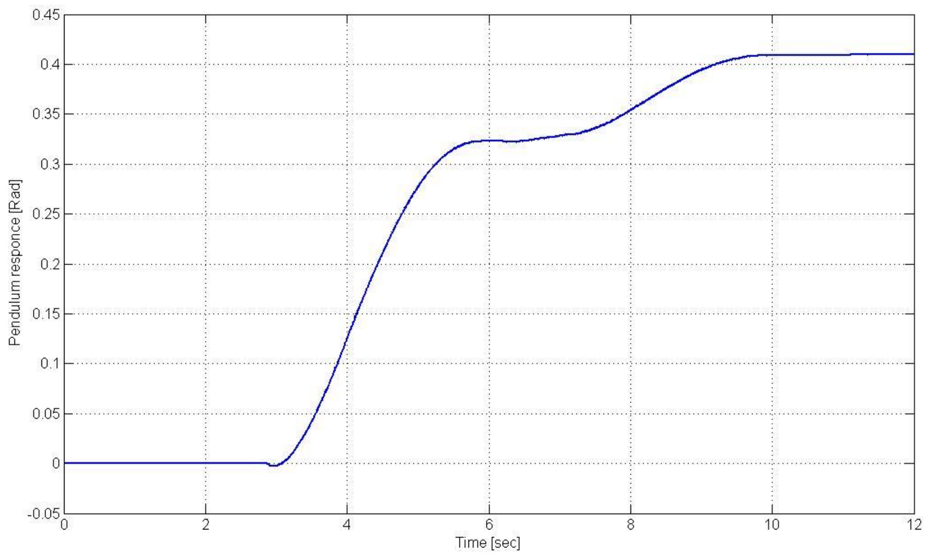

Figure 5.11: Pendulum response for the tracker

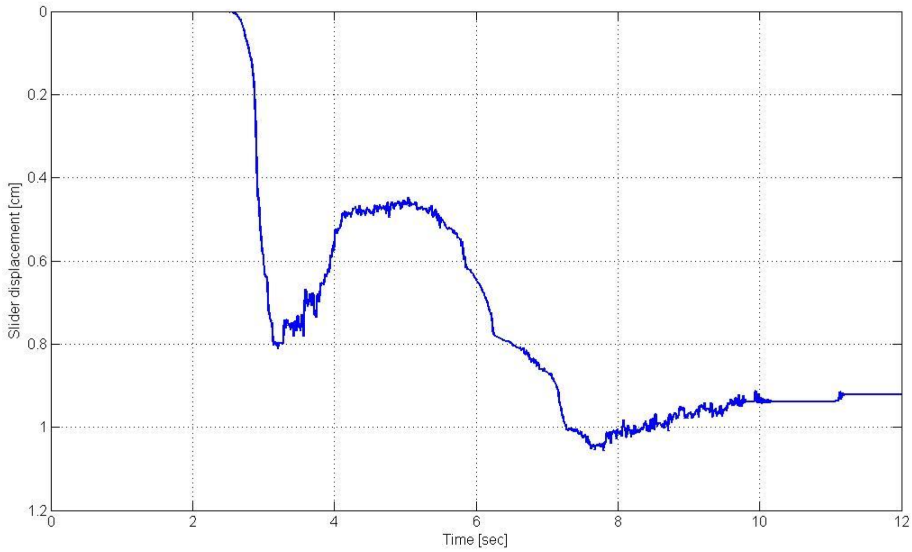

Figure 5.12: slider response for the tracker

From the results, it could be concluded that:

- The microcontroller was able to stabilized the IP and track a step input.
- The results is similar to results which obtained by using x-pc target.

Table 5.1 show a comparison between simulated results and experimental results for both x-pc target and embedded system.

Table 5.1: Discrete controller's comparison

	$\%Overshoot$	$Settling\ time$ $[sec]$	$Rising\ time$ $[sec]$	$\%steady\ state$ $Error$
Simulation	12.3	3.9	1.4	0.0
x-pc target	0.0	7	4.8	2.0
embedded system	0.0	6.76	5.2	3.0

The results for the embedded system were similar to x-pc target, so the embedded system works correctly and in effective way.

6 Feedback linearization

6.1 Introduction

Feedback linearization is an approach to control nonlinear process. The main idea is to algebraically transform nonlinear systems dynamics into linear ones in view of the control input, so that linear control techniques can be applied. This differs entirely from conventional (Jacobean) linearization, because feedback linearization is achieved by exact state transformation and feedback, rather than by linear approximations of the dynamics. However to be able to apply this method of control, all of the states must be measured [3].

6.2 Principle

Consider the following nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{v} \qquad \text{(6-1a)}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \qquad \text{(6-1b)}$$

Where:

- $\boldsymbol{f} \in R^n$: Nonlinear vector function.
- $\boldsymbol{G} \in R^{n\times m}$: Nonlinear vector function.
- $\boldsymbol{v} \in R^m$: Input vector.

Let $v = \alpha u(t) + \beta$, then select $\alpha \; and \; \beta$ such that the resulted model linear for the input u.

Recalling the nonlinear differential equations for the IP from chapter 2

$$m_1\ddot{x} + m_1 l_0\ddot{\theta} - m_1 x\dot{\theta}^2 - m_1 g \sin\theta + f_2\dot{x} = F(t) \qquad \text{(2-8)}$$

$$[J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2]\ddot{\theta} + 2m_1 x\dot{x}\dot{\theta} + m_1 l_0\ddot{x} - (m_1 l_0 + m_2 l_c)g \sin\theta - m_1 g x \cos\theta + f_1\dot{\theta} \; = \; PL \qquad \text{(2-10)}$$

Let $\sin(\theta) = \theta$, $\cos(\theta) = 1$ about operation point (because the operation angle is small due to physical limits of the device) and niggling friction in equations 2-8 and 2-10

$$m_1\ddot{x} + m_1 l_0\ddot{\theta} - m_1 x\dot{\theta}^2 - m_1 g\theta = F(t) \quad (6\text{-}2a)$$

$$[J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2]\ddot{\theta} + 2m_1 x\dot{x}\dot{\theta} + m_1 l_0\ddot{x} - (m_1 l_0 + m_2 l_c)g\theta - m_1 gx = PL \quad (6\text{-}2b)$$

If the control chosen to be

$$F(t) = \alpha_1 u(t) + \beta_1 \quad (6\text{-}3a)$$

Where $\alpha_1 = 1$ *and* $\beta_1 = -m_1 x\dot{\theta}^2$

And

$$PL = \alpha_2 N + \beta_2 \quad (6\text{-}3b)$$

$\alpha_2 = 1$ *and* $\beta_2 = 2m_1 x\dot{x}\dot{\theta}$

Thus, the nonlinear term could be canceled. This cancellation results in the system

$$m_1\ddot{x} + m_1 l_0\ddot{\theta} - m_1 g\theta = u(t) \quad (6\text{-}4a)$$

$$J(x)\ddot{\theta} + m_1 l_0\ddot{x} - m_o g\theta - m_1 gx = N \quad (6\text{-}4b)$$

Where

- $J(x) = J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2$
- $m_o = m_1 l_o + m_2 l_c$

Simplifying equations 6-4a and 6-4b in terms of $\ddot{x}$ *and* $\ddot{\theta}$ yields:

$$\ddot{x} = \frac{u(t)}{m_1} - l_o\ddot{\theta} + g\theta \quad (6\text{-}5a)$$

$$\ddot{\theta} = \frac{[N + m_o g\theta + m_1 gx - m_1 l_0\ddot{x}]}{J(x)} \quad (6\text{-}5b)$$

Substitute (6-5a) in (6-4b) and (6-5b) in (6-4a), this results

$$J\ddot{\theta} - m_2 l_c g\theta - m_1 g x = N - l_0 u(t) \quad \text{(6-6a)}$$

$$J\ddot{x} - (J - m_2 l_c l_0) g\theta + m_1 l_0 g x = \frac{J_0}{m_1} u(t) - l_0 N \quad \text{(6-6b)}$$

Where $J = J(x) - m_1 l_o^2$ (6-6c)

But $J(x) = [J_1 + J_2 + (x^2 + l_0^2)\, m_1 + m_2 l_c^2]$

Where J_1 and J_2 are the polar moment of inertias around joint (O) for the slider and pendulum respectively. However the value of x is small so it can be neglected, thus $J(x) = J_o$ *and* $J = J^*$ which there values found in table 2-1.

Convert equations 6-6a and 6-6b to state space model by choosing the states to be $x, \dot{x}, \theta$ *and* $\dot{\theta}$ respectively

$$q_1 = x \quad \rightarrow \quad \dot{q}_1 = q_2$$

$$q_2 = \dot{x} \quad \rightarrow \quad \dot{q}_2 = \frac{J(x)}{Jm_1} u(t) - \frac{l_0}{J} N - \frac{m_1 l_0 g}{J} q_1 + \frac{(J - m_2 l_c l_0) g}{J} q_3$$

$$q_3 = \theta \quad \rightarrow \quad \dot{q}_3 = q_4$$

$$q_4 = \dot{\theta} \quad \rightarrow \quad \dot{q}_4 = \frac{1}{J} N - \frac{l_0}{J} u(t) + \frac{m_1 g}{J} q_1 + \frac{m_2 l_c g}{J} q_3$$

So the state space model is:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ m_1 l_o g/J & 0 & g - m_2 l_c l_o g/J & 0 \\ 0 & 0 & 0 & 1 \\ m_1 g/J & 0 & m_2 l_c g/J & 0 \end{bmatrix} \boldsymbol{q} + \begin{bmatrix} 0 \\ J(x)/Jm_1 \\ 0 \\ -l_o/J \end{bmatrix} u + \begin{bmatrix} 0 \\ -l_0/J \\ 0 \\ 1/J \end{bmatrix} N$$

Substituting the values of Table 2.1 into state space model matrices yields

$$\dot{\boldsymbol{q}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -19 & 0 & 14.5 & 0 \\ 0 & 0 & 0 & 1 \\ 57.56 & 0 & -14.23 & 0 \end{bmatrix}}_{A} \boldsymbol{q} + \underbrace{\begin{bmatrix} 0 \\ 7.7 \\ 0 \\ -9.1 \end{bmatrix}}_{B} u + \underbrace{\begin{bmatrix} 0 \\ -9.2 \\ 0 \\ 27.8 \end{bmatrix}}_{E} N$$

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_{C} \boldsymbol{q}$$

6.3 Controller

The stabilization problem for the nonlinear system has been reduced to a stabilization problem for a controllable linear system. Thus, the designed controller proceed to stabilizing linear state feedback control. The controller applied here is discrete tracker, which explained early in chapter three.

First, the system matrices are found in the z-plane at sampling time equals one millisecond:

$$F_n = I + T_s A = \begin{bmatrix} 1 & 0.001 & 0 & 0 \\ -0.019 & 1 & 0.0144 & 0 \\ 0 & 0 & 1 & 0.001 \\ 0.058 & 0 & -0.0141 & 1 \end{bmatrix}$$

$$G_n = T_s B = \begin{bmatrix} 0 \\ 0.0077 \\ 0 \\ -0.0091 \end{bmatrix}$$

The extended matrices found as follows:

$$\boldsymbol{F_{ne}} = \begin{bmatrix} \boldsymbol{F_n} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} 1 & 0.001 & 0 & 0 & 0 \\ -0.019 & 1 & 0.0144 & 0 & 0 \\ 0 & 0 & 1 & 0.001 & 0 \\ 0.058 & 0 & -0.0141 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

$$G_{ne} = \begin{bmatrix} G_n \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0077 \\ 0 \\ -0.0091 \\ 0 \end{bmatrix}$$

Controller gains were found by using place command as follows:

- Select locations of the poles in z-plane inside unity circuit to achieve stability

$$P_n = [0.8876 \quad 0.9884 \quad 0.9952 \quad 0.9986 + 0.0016i \quad 0.9986 - 0.0016i]$$

- $K = place(F_e,\ G_e,\ P_n)$

  $$K = [\underbrace{447.7 \quad 81.8 \quad 134.3 \quad 54.6}_{k} \quad \underbrace{-0.1}_{k_n}]$$

6.4 Simulink model

Figure 6.1 shows the Simulink model for the feedback linearization with discrete tracker controller, as noticed the input force equals to $u - m_1 x \dot{\theta}^2$.

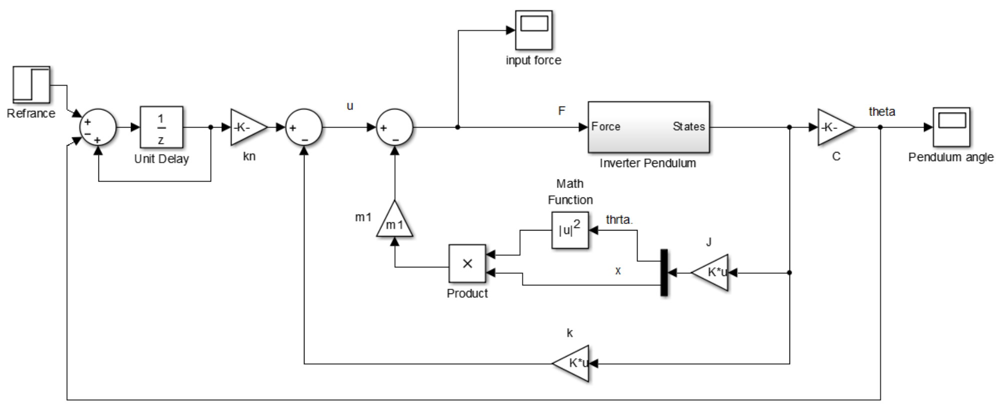

Figure 6.1: Feedback linearization Simulink model

6.5 Simulation results

The pendulum rod commanded to rotate 0.4 rad with zero initial conditions, so the results for pendulum, slider and input force as follows:

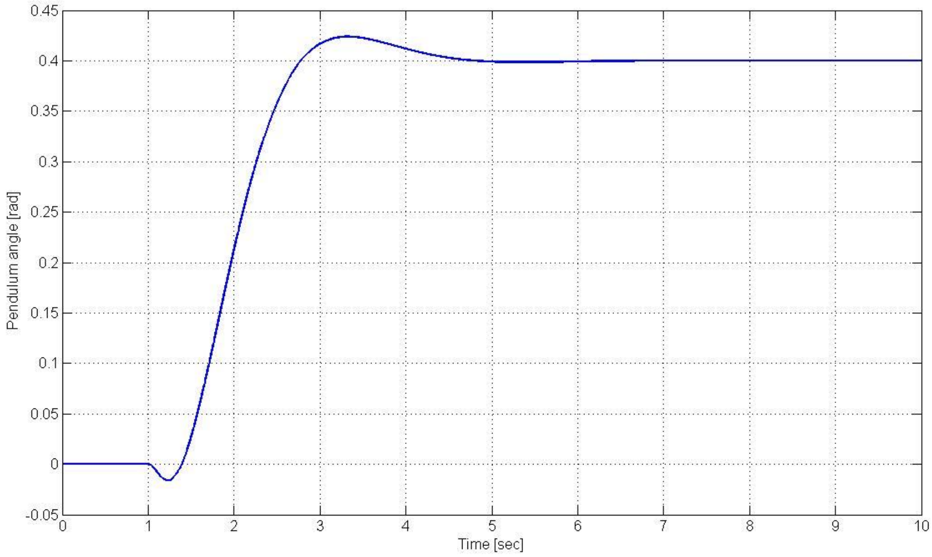

Figure 6.2 Pendulum response (Feedback linearization)

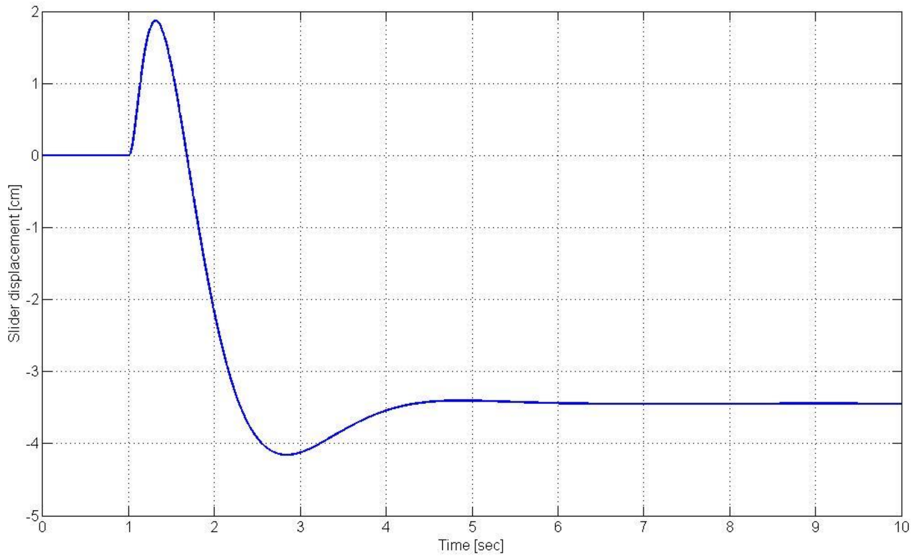

Figure 6.3 Slider response (Feedback linearization)

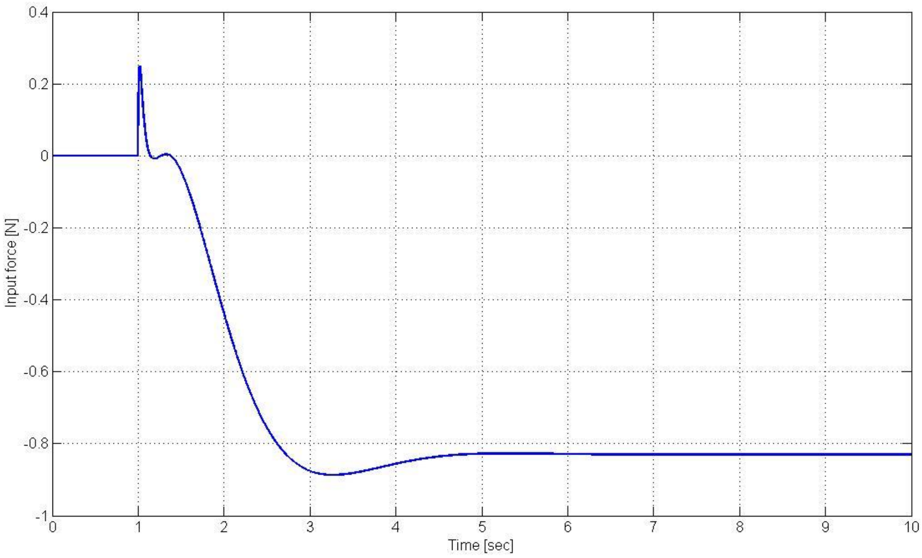

Figure 6.4 Input force (Feedback linearization)

From the simulated results, it could be noted that:

- In figure 6.2 the percent overshoot and the stilling time is 5.95 % and 3.183 *seconds* respectively.
- Figure 6.4 shows the input force and because of neglecting the friction in the model the IP needs 0.8 newton's to hold the slider from slipping.

6.6 Experimental results

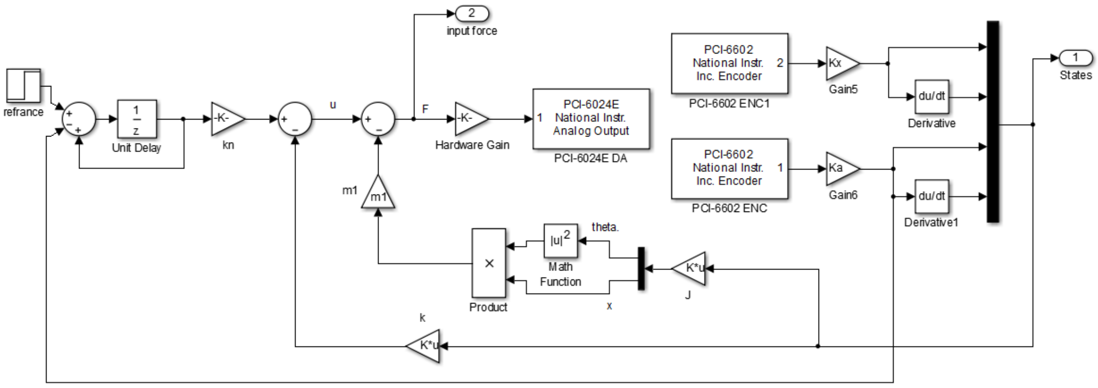

Figure 6.5 x-pc target model (Feedback linearization)

The controller applied to the IP by using x-pc target technique and the Simulink model is shown in figure 6.5, and the experimental results as follows:

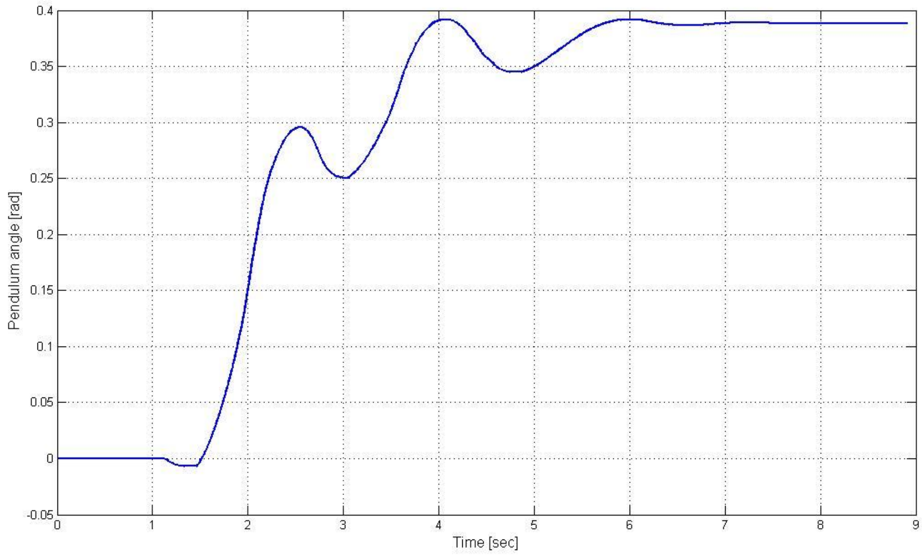

Figure 6.6 Experimental pendulum response

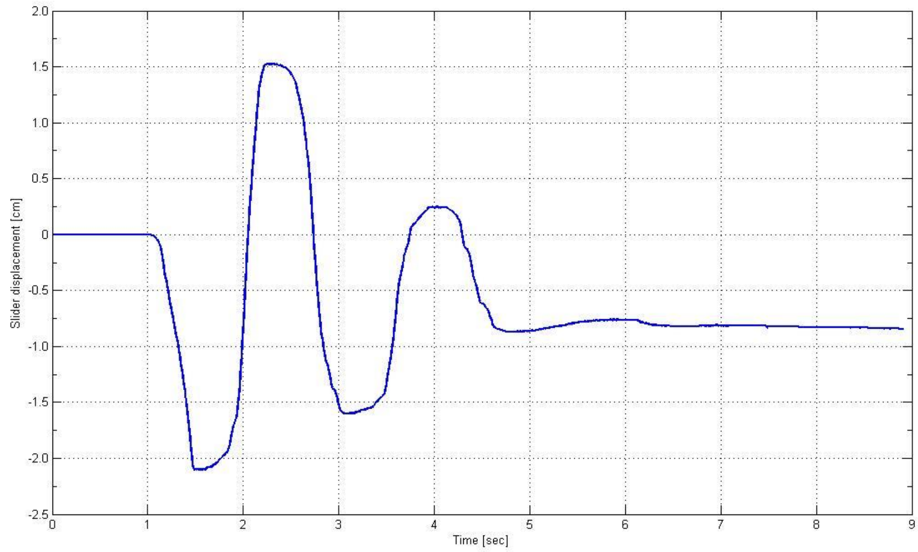

Figure 6.7 Experimental Slider response

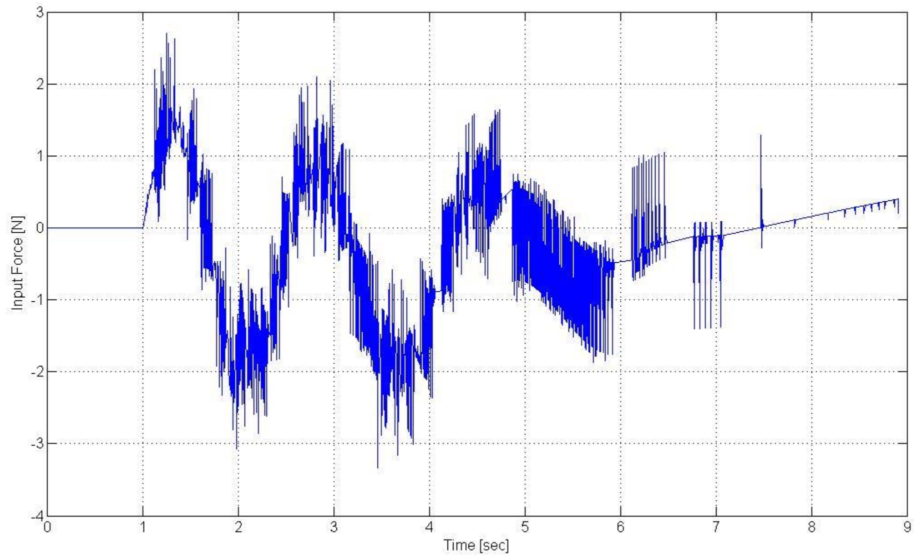

Figure 6.8 Input force experimentally

From the previous results, it could be noted that:

- From figure 6.6 the settling time equals to $4.6\ seconds$ which is slower than simulation result.
- The negative overshoot in pendulum response (figure 6.6) is due to right half plane zero.
- The real input force (figure 6.8) is greater than force expected in simulation because of friction, moreover when the pendulum reaches its position the input force goes to zero due to friction that holds the slider from slipping.
- In figure 6.8, the input force is noisy, but the driver tacks the average and drive the motor smoothly.
- The main advantage of this controller it was able to achieve the desired pendulum angle with less force than previous controllers.

7 Conclusion & Suggestion

7.1 Conclusion

Different control technique were applied to the inverted pendulum, and it could be concluded that all of state feedback controllers were able to stabilize the IP, furthermore tracking some reference angle for the pendulum rod. However it was difficult to have the same results as simulation, because of the pendulum rod cannot be controlled directly, just by controlling the slider position you were able to control the pendulum rod.

In general, the designed controllers were limited by the physical limits of the device, like the length of the slider, so the speed of the response cannot be increased so far or the system will be unstable.

Embedded system based on micro controller was able to control the IP and gave results similar to discrete controller which applied by using x-pc target technique.

The feedback linearization controller helps to deal effectively with the nonlinearity of the IP, and that appears in the input force to the system that is less than other controllers.

7.2 Suggestion for the Future Work

Although the controllers has been successfully stabilized the system, controller technique should be improved so that a robust controller and a better response can be achieved. More control theories can be applied and tested, like fuzzy logic control and neural networks.

On the other-hand embedded controller need more development and to tests more controllers.

Appendix A

Appendix A

Controller design (m-file)

A.1 PID controller

```
%==================================================================%
%=======       Inverted Pendulum ECP 505 model              ======%
%=======                    PID                             ======%
%==================================================================%
                            clc; clear;

%% definitions of the basic parameters:
Ts=0.001;        % Sampling time
Kx=1.9920e-005;  % sliding rod encoder          m / increment
Ka=3.9277e-004;  % pendulum angular encoder     rad/increment
Kv_f=1/4.2;      % Hardware gain                v/N

%% Controller Gains: (using sisotool function)
Kd=11.25;
Kp=422;
Ki=86;
```

A.2 Regulator controller

```
%==================================================================%
%=======        Inverted Pendulum ECP 505 model             ======%
%=======                   Regulator                         ======%
%==================================================================%
                            clc; clear;

%% definitions of the basic parameters:
Ts=0.001;       % Sampling time
Kx=1.9920e-005; % sliding rod encoder          m / increment
Ka=3.9277e-004; % pendulum angular encoder     rad/increment
Kv_f=1/4.2;     % Hardware gain                v/N
%% System [ theta theta. x  x.]'

A=[0 1 0 0;-14.23 0 57.562 0;0 0 0 1;14.5 0 -19 0];%system matrix
B=[0 -9.2 0 7.75]';                                %input matrix
C=[1 0 0 0];                                       %output matrix
D=0;
Cm=ctrb(A,B);
Rc=rank(Cm);

%% Regulator Controller
Wn1 = 10;
z1 = 0.9;
Wn2 = 14.32;
z2 = 0.93;

p1 =-z1*Wn1+1i*Wn1*sqrt(1-z1^2);
p2 =-z1*Wn1-1i*Wn1*sqrt(1-z1^2);
p3 =-z2*Wn2+1i*Wn2*sqrt(1-z2^2);
p4 =-z2*Wn2-1i*Wn2*sqrt(1-z2^2);

poles=[p1 p2 p3 p4];

K_Reg=place(A,B,poles);
```

A.3 Tracker controller

```
%================================================================%
%=======       Inverted Pendulum ECP 505 model              ======%
%=======               Tracker & Observer                   ======%
%================================================================%
                            clc; clear;
%% definitions of the basic parameters:
Ts=0.001;        % Sampling time
Kx=1.9920e-005; % sliding rod encoder          m / increment
Ka=3.9277e-004; % pendulum angular encoder     rad/increment
Kv_f=1/4.2;      % Hardware gain                v/N

%% System [ theta theta. x  x.]'
A=[0 1 0 0;-14.23 0 57.562 0;0 0 0 1;14.5 0 -19 0];%system matrix
B=[0 -9.2 0 7.75]';                                 %input matrix
C=[1 0 0 0];                                        %output matrix
D=0;

%% Tracker Controller
Wn1 = 10;
z1 = 0.9;
Wn2 = 14.32;
z2 = 0.93;
p1 =-z1*Wn1+1i*Wn1*sqrt(1-z1^2);
p2 =-z1*Wn1-1i*Wn1*sqrt(1-z1^2);
p3 =-z2*Wn2+1i*Wn2*sqrt(1-z2^2);
p4 =-z2*Wn2-1i*Wn2*sqrt(1-z2^2);

Ae=[A zeros(4,1);-C 0];
Be=[B;0];
Ce=[C 0];
Cme=ctrb(An,Bn)
Rce=rank(Cme);

poles=[p1 p2 p3 p4 -10];
k_Trac=place(Ae,Be,poles);
K=k_Trac(1:4);
Ke=-k_Trac(5);
```

```
%% Observer Controller (faster 10 times than controller)
Observer=[p1 p2 p3 p4]-10;
Cm=[1 0 0 0;0 0 1 0]; %output matrix
Om=rank(obsv(A,Cm));
L=place(A',Cm',Observer)';
```

A.4 Discrete controller

```
%=================================================================%
%=======        Inverted Pendulum ECP 505 model             ======%
%=======                     Discrete                       ======%
%=================================================================%
                            clc; clear;
%% definitions of the basic parameters:
Ts=0.004;         % Sampling time
Kx=1.9920e-005; % sliding rod encoder          m / increment
Ka=3.9277e-004; % pendulum angular encoder     rad/increment
Kv_f=1/4.2;      % Hardware gain                v/N


%% System [ theta theta. x  x.]'
A=[0 1 0 0;-14.23 0 57.562 0;0 0 0 1;14.5 0 -19 0];%system matrix
B=[0 -9.2 0 7.75]';                                %input matrix
C=[1 0 0 0];                                       %output matrix
D=0;


%% Discrete Controller
F=eye(4)+Ts*A;
G=Ts*B;
Fe=[F zeros(4,1);-C eye(1)];
Ge=[G;0];
Q=eye(4)*100;
R=1;
Q(1,1)=10000;
Q(3,3)=10000;
k_d=dlqr(F,G,Q,R)
Pe=eig(Fe-Ge*[k_d -0.099])

ke=place(Fe,Ge,Pe)

k=ke(1:4)
ki=ke(5)
```

A.5 Feedback linearization

```
%================================================================%
%=======        Inverted Pendulum ECP 505 model             ======%
%=======             Feedback linearization                 ======%
%================================================================%
                             clc; clear;
%% definitions of the basic parameters:
Ts=0.001;        % Sampling time
Kx=1.9920e-005; % sliding rod encoder          m / increment
Ka=3.9277e-004; % pendulum angular encoder     rad/increment
Kv_f=1/4.2;      % Hardware gain               v/N
m1=0.213;
m2=1.785;
m0=0.1204;
lo=0.33;
lc=0.028;
g=9.81;


%% System [x  x. theta theta.]'
A=[0 1 0 0;-19 0 14.45 0;0 0 0 1;58 0 -14.1 0]; %system matrix
B=[0 7.75 0 -9.2]';                              %input matrix
C=[0 0 1 0];


%% Controller
F=eye(4)+Ts*A;
G=Ts*B;
Fe=[F zeros(4,1);-C eye(1)];
Ge=[G;0];
Q=eye(4)*100;
R=1;
Q(1,1)=10000;
Q(3,3)=10000;

kk=dlqr(F,G,Q,R)
Pe=eig(Fe-Ge*[kk -0.1])

K=place(Fe,Ge,Pe')
k=K(1,:,4)
kn=K(5)
```

A.6 microcontroller c code

```
/*LCD CONNECTION
LCD 1- ARDU GND + POTENTIOMETER
LCD 2- ARDU 5V + POTENTIOMETER
LCD 3- POTENTIOMETER MIDDLE
LCD 4- ARDU PIN 34
LCD 5- ARDU GND  + POTENTIOMETER
LCD 6- ARDU PIN 36
LCD 7/8/9/10 - NOT USED
LCD 11- ARDU PIN 44
LCD 12- ARDU PIN 42
LCD 13- ARDU PIN 40
LCD 14- ARDU PIN 38
LCD 15- ARDU 5V
LCD 16- ARDU GND
*/
#include <LiquidCrystal.h>
#include <Encoder.h>
#include <TimerOne.h>
// Red +5v
// Black Gnd
Encoder Rod(2, 3);
// A1: white
// B1: yellow
Encoder Slider(18, 19);
// A2: blue
// B2: green

LiquidCrystal lcd(34, 36, 44, 42, 40, 38);

int cw=9;       //   + |\
int ccw=10;     //   - |/

int Ts = 4000;


void setup() {

  pinMode(cw, OUTPUT);
  pinMode(ccw , OUTPUT);

  Timer1.initialize(Ts); // set a timer of length 100000
microseconds (or 0.1 sec - or 10Hz => the led will blink 5 times, 5
cycles of on-and-off, per second)
  Timer1.attachInterrupt( timerIsr ); // attach the service routine
here

  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  lcd.print("InvertedPendulum");

}

float oldRod   = 0.0;
float oldSlider = 0.0;
float Rod_Speed=0.0;
float Slider_Speed=0.0;
```

```
float u=0.0,F=0.0;
float newRod=0.0, newSlider=0.0;
float sensorValue = analogRead(A0);
float r=0.0 n=0.0, np=0.0;


void loop() {


  if(u >= 0.0){
  analogWrite(ccw, 0);
  analogWrite(cw, u);

  }
  if(u < 0.0) {
  analogWrite(cw, 0);
  analogWrite(ccw, 0.0-u);

  }
  lcd.setCursor(0, 1);
  lcd.print(r*180*7/22);

  lcd.setCursor(10, 1);
  lcd.print(newRod*180*7/22);
  }



 void timerIsr()
{
  r = (((float)(analogRead(A10)))-512.0)*( 0.4363/512.0);
  newRod = ((float)(Rod.read()))/2546.0;
  newSlider = ((float)(Slider.read()))/50200.0;
  Rod_Speed=(newRod-oldRod)/(0.004);
  Slider_Speed=(newSlider-oldSlider)/(0.004);
  oldRod=newRod;
  oldSlider=newSlider;

  n=r-newRod+np;
  F=np*0.095-newSlider*390.7318-Slider_Speed*75.4469-newRod*56.181-
Rod_Speed*35.896;
  u=(F/4.2)*(255/5);
  np=n;
}
```

References

[1] Elbert Hendricks and others, Linear Systems Control Deterministic and Stochastic Methods, Springer-Verlag Berlin Heidelberg, 2008.

[2] Norman S. Nice, Control System Engineering, John Wiley & Sons, Inc. Sixth Edition, California State Polytechnic University, Pomona.

[3] J. K. Hedrick and A. Girard, Control of Nonlinear Dynamic Systems: Theory and Applications, 2005.

[4] Khalil Sultan, “Inverted Pendulum, Analysis, Design and Implementation”, Institute of Industrial Electronics Engineering, Karachi, Pakistan, 2003.

[5] Ratchatin Chanchareon and others, “Stabilizing of an Inverted Pendulum Using Computed Feedback Linearization Technique”, The 19th Conference of Mechanical Engineering Network of Thailand, October 2005, Phuket, Thailand.

[6] Arduino web site
http://www.arduino.cc/

[7] xPC Target User’s Guide, Math Works, Inc.
http://www.mathworks.com/support/product/XP/productnews/xpc_target_ug_Nov_07_2003.pdf

[8] Manual for Model 505 Inverted Pendulum
http://maelabs.ucsd.edu/mae171/controldocs/pendulum.htm

[9] Wikipedia, the free encyclopedia
http://www.wikipedia.org/