UNIVERSITY OF NAIROBI

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

PROJECT TITLE: DESIGN AND IMPLEMENTATION OF A DIGITAL CONTROLLER FOR A WALKING ROBOT USING LEGO COMPONENTS

PROJECT INDEX: 74

NAME: WAMAI RUTH WANGU

REG. NO: F17/35462/2010

SUPERVISOR: PROF. M. MBUTHIA

EXAMINER: MR. P. MUSAU

A project submitted to the Department Of Electrical And Information Engineering in partial fulfilment of the requirements of BSc. Electrical and Electronics Engineering of the University of Nairobi.

Date of Submission: 24th of April 2015
DECLARATION OF ORIGINALITY

FACULTY/SCHOOL/INSTITUTE: Engineering

DEPARTMENT: Electrical and Information Engineering

COURSE NAME: Bachelor of Science in Electrical and Electronics Engineering

NAME OF STUDENT: WAMAI RUTH WANGU

REGISTRATION NUMBER: F17/35462/2010

COLLEGE: Architecture and Engineering

WORK: Design and Implementation of a Digital Controller for a Walking Robot Using Lego Components

1. I understand what plagiarism is and I am aware of the university policy in this regard.

2. I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people’s work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi’s requirements.

3. I have not sought or used the services of any professional agencies to produce this work.

4. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.

5. I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti-plagiarism policy.

Signature: ..................................................

Date: ..................................................
DEDICATION.

I dedicate this to my family.

I love you all.
ACKNOWLEDGEMENTS

All the Praise to The Lord Jesus Christ for the breath of life and the strength to pursue this course.

To my family, for the continual support in all I endeavor to do.

To Prof. Mbuthia Mwangi for guidance and encouragement during the project processes.

To my comrades, thank you for being there for me.
# TABLE OF CONTENTS

ACRONYMS ................................................................................................................................................... viii

ABSTRACT ........................................................................................................................................................ ix

CHAPTER 1 ..................................................................................................................................................... 1

1. INTRODUCTION ..................................................................................................................................... 1

1.1 THE INVERTED PENDULUM ................................................................................................................. 1

1.2 PROBLEM STATEMENT .......................................................................................................................... 1

1.3 OBJECTIVES .......................................................................................................................................... 1

1.3.1 OVERALL OBJECTIVE ...................................................................................................................... 1

1.3.2 SPECIFIC OBJECTIVES .................................................................................................................... 1

1.4 PROJECT JUSTIFICATION .................................................................................................................... 2

1.5 PROJECT SCOPE ................................................................................................................................ 2

1.6 REPORT ORGANIZATION .................................................................................................................... 2

CHAPTER 2 .................................................................................................................................................... 4

2. LITERATURE REVIEW ............................................................................................................................ 4

2.1 INVERTED PENDULUM ........................................................................................................................ 4

2.2 CONTROL SYSTEMS ............................................................................................................................ 8

2.2.1 DIGITAL CONTROL .......................................................................................................................... 9

2.2.2 TYPES OF CONTROLLERS ............................................................................................................. 10

2.3 HARDWARE ......................................................................................................................................... 11

2.3.1 LEGO MINDSTORM EV3 BRICK ................................................................................................... 12

2.3.2 SERVO MOTORS ............................................................................................................................. 12

2.3.3 GYROSCOPE ................................................................................................................................... 13

3. DESIGN AND METHODOLOGY .............................................................................................................. 15

3.1 SIMULATION ....................................................................................................................................... 15

3.1.1 SYSTEM PARAMETERS ..................................................................................................................... 15

3.1.2 S-DOMAIN INVERTED PENDULUM TRANSFER FUNCTION .......................................................... 17

3.1.3 Z-DOMAIN INVERTED PENDULUM TRANSFER FUNCTION ........................................................ 19

3.1.4 STUDY OF THE DISCRETE-TIME TRANSFER FUNCTION ........................................................... 21

3.2 DIGITAL CONTROLLER DESIGN ....................................................................................................... 23

3.2.1 PERFORMANCE CRITERIA ............................................................................................................. 23
LIST OF FIGURE

Figure 1.1 Inverted pendulum .................................................................................................................... 5
Figure 2.1 Control System ............................................................................................................................ 9
Figure 3.2 Brick ........................................................................................................................................ 12
Figure 4.3 Servo motor .............................................................................................................................. 13
Figure 5.4 Gyro sensor .............................................................................................................................. 14
Figure 6.1 Pole Zero Map of the Open Loop System ................................................................................ 17
Figure 7.2 Step response .......................................................................................................................... 18
Figure 8.3 Rootlocus ................................................................................................................................. 18
Figure 9.4 Bode Plot ................................................................................................................................. 19
Figure 10.5 Pole zero map in z-domain ..................................................................................................... 21
Figure 11.6 Step response .......................................................................................................................... 22
Figure 12.7 Rootlocus ................................................................................................................................. 22
Figure 13.8 Rootlocus ................................................................................................................................. 24
Figure 14.9 Balancing Algorithm ................................................................................................................ 26
Figure 15. Walking algorithm ...................................................................................................................... 28
Figure 16.9.2 Vertical gyro sensor ............................................................................................................. 29
Figure 17.9.3 Assembly of the robot .......................................................................................................... 30
Figure 18.9.4: The robot after completion ................................................................................................. 30
Figure 4.19 The walking robot in progress (Side view) ............................................................................. 34
Figure 4.20.1 Robot in progress (Top view) ............................................................................................... 35
Figure 21 The two large motors ................................................................................................................. 39
Figure 22 Connecting the two large motors together ............................................................................... 39
Figure 23 Adding the brick to the robot ..................................................................................................... 40
Figure 24 Adding the gyroscope to the robot ............................................................................................ 40
Figure 25: Mindstorm EV3 components ................................................................................................... 41
Figure 26: Main balancing and walking program ..................................................................................... 44
Figure 27: Pid program .............................................................................................................................. 45
ACRONYMS

ADC-Analog to Digital Conversion

DAC-Digital to Analog Conversion

PD-Proportional plus Derivative

PI-Proportional plus Integral

PID-Proportion plus Integral plus Derivative

Kp-Proportion gain

Ki-Integral gain

Kd-Derivative gain

MATLAB-MATrix LABoratory
ABSTRACT

This project focuses on the design and implementation of a digital controller for a walking robot using LEGO components. The robot will be a wheeled robot that will require control for balancing. This project will include the design of a digital controller that can effectively balance the robot.

It will provide a detailed description for modeling the plant, design of the controller, simulating the final system and implementation into hardware. The LEGO mindstorm EV3 will be used to do this task. This is a series of kits that contain software and hardware to create automizable, programmable robots. They include an intelligent brick computer that controls the system, a set of modular sensors and motors, and LEGO parts to create a mechanical system.
CHAPTER 1

1. INTRODUCTION

1.1 THE INVERTED PENDULUM
The wheeled robot is an inverted pendulum moving on a cart. The inverted pendulum is a classical control problem, which involves developing a system to balance the pendulum. Whereas a normal pendulum is stable when hanging downwards, an inverted pendulum is inherently unstable since its mass is above its pivot point. This system must hence be actively balanced in order to remain upright by moving the pivot point (cart) horizontally as part of the feedback system. An ideal controller would keep the pendulum balanced.

1.2 PROBLEM STATEMENT
Stability is one of the vital system performance characteristics that control systems aims to achieve. To obtain a stable moving robot, a digital controller needs to be incorporated into the system. The controller compares the output of the plant to the input and compensates to correct for any errors.

1.3 OBJECTIVES

1.3.1 OVERALL OBJECTIVE
To design and implement a digital controller for a walking robot using LEGO components.

1.3.2 SPECIFIC OBJECTIVES
In order to accomplish the overall objective of this project, the following are the specific objectives:

- Model the two-wheeled robot as an inverted pendulum.
- Study the stability and performance of the system.
- Set the desired performance of the system.
- Design a digital controller to meet the desired specification.
- Study the compensated system.
• Implement a digital controller into hardware.

1.4 PROJECT JUSTIFICATION

Robots are essential in this century. It has been noted that robots can work tirelessly hence doing more work compared to humans who after a period of time tend to decrease in their work productivity. This consistency has proven important especially in manufacturing industries where production of quality material is a necessity.

The need to digitally control these robots is essential as most robots use analog controllers. The advantages of digital controllers have hence been outlined clearly in my literature review.

1.5 PROJECT SCOPE

Inverted pendulum, a highly non-linear unstable system, is used as a benchmark for implementing the control methods. Here the objective is to control the system such that the inverted pendulum stabilizes in upright position. The cart, depending upon the direction of the deflection of the pendulum (the angle of the inverted pendulum, θ) moves horizontally in order to bring the pendulum to rest in a vertical position.

MATLAB will be used to develop the appropriate algorithms for the control system, and then simulate to verify that algorithms work as intended.

1.6 REPORT ORGANIZATION

The project report will be organised as follows:

CHAPTER 2: Literature review will be discussed.

This chapter gives a detailed description of the inverted pendulum concept, control systems and hardware components required in the implementation.

CHAPTER 3: Design and Methodology

This gives a comprehensive description of MATLAB simulations and the software used to implement the walking robot.
CHAPTER 4: Results and Analysis

This chapter gives the results and analysis of system performance and the challenges encountered.

CHAPTER 5: Conclusion and Recommendation

This is the final chapter and entails the conclusion and recommendations for further work on topics to be studied.
CHAPTER 2

2. LITERATURE REVIEW

2.1 INVERTED PENDULUM

Beginning with the simplified free-body diagram of the cart and pendulum, the dynamics of the system can hence be modeled using Newton’s laws of motion. The mathematical expression was established to find the transfer function.

Since modern control theory is based mostly in controlling linear systems, the inverted pendulum control problem is challenging due to the nonlinearities inherent to the system. These nonlinearities are generally ignored in many control schemes by replacing the nonlinear model of the system with an approximate linear one (linearizing the model). This process is generally valid because the controller is only controlling the system in a small range about an equilibrium position.

In order to obtain the system dynamics, the following assumptions were made:

1. The system starts in an equilibrium state i.e. that the initial conditions are assumed to be zero
2. The pendulum does not move more than a few degrees away from the vertical to satisfy a linear model. By linearizing the system around the origin where the pendulum is up, a very good approximation of the actual non-linear system can be obtained.
3. An assumption was made that the rod is massless. The cart mass and point mass at the upper end of the inverted pendulum are denoted as M and m respectively.
4. There is an externally x-directed force on the cart, u(t) and a gravity force acts on the point mass at all times
Summation of all the forces in the free-body diagram of the cart in the horizontal direction generates:

\[ M\ddot{x} + b\dot{x} + N = F \] \hspace{1cm} \text{(1)}

Summation of the forces in the free-body diagram of the pendulum in the horizontal direction also yields:

\[ N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \] \hspace{1cm} \text{(2)}

Substituting equation (2) into (1), we get:

\[ (M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \] \hspace{1cm} \text{(3)}

Summation of the forces perpendicular to the pendulum:

\[ P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \] \hspace{1cm} \text{(4)}

The moments about the centroid of the pendulum are summed to yield the following:

\[ -Pl\sin\theta - Nl\cos\theta = l\ddot{\theta} \] \hspace{1cm} \text{(5)}
Combination of equation (4) and (5):

\[(l + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\dot{x}\cos\theta\]  \hspace{1cm} (6)

The inverted pendulum is adherently a non-linear system due to the cosine and sine terms generated while modeling the dynamics of the system. Approximating the non-linear transfer function to a linear one allows for controller design methods that use linear system techniques. This enables the system to be controlled in a specific range. Most control techniques hinge on the assumption that the system is linear.

\[
\cos \theta = \cos(\pi + \phi) \approx -1 \]  \hspace{1cm} (7)

\[
\sin \theta = \sin(\pi + \phi) \approx -\phi \]  \hspace{1cm} (8)

\[
\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \]  \hspace{1cm} (9)

After substituting the above approximations into the non-linear governing equations, we arrive at the two linearized equations of motion. \( u(t) \) has been substituted for the input \( F \).

\[(l + ml^2)\phi - mgl\dot{\phi} = ml\ddot{x} \]  \hspace{1cm} (10)

\[(M + m)x + bx - ml\dot{\phi} = u \]  \hspace{1cm} (11)

Taking the Laplace transform of the system equations assuming zero initial conditions results in the transforms shown below.

\[(l + ml^2)\phi(s)s^2 - mgl\phi(s) = mlX(s)s^2 \]  \hspace{1cm} (12)

\[(M + m)X(s)s^2 + bX(s)s - ml\phi(s)s^2 = U(s) \]  \hspace{1cm} (13)

Making \( X(s) \) subject of the formula from equation (12) yields:

\[X(s) = \left[\frac{l + ml^2}{ml} - \frac{\phi(s)}{s^2}\right] \]  \hspace{1cm} (14)

Substitution of the above equation into equation (13) gives us:
\[(M + m) \left[ \frac{l + ml^2}{ml} - \frac{g}{s^2} \right] \phi(s)s^2 + b \left[ \frac{l + ml^2}{ml} - \frac{g}{s^2} \right] \phi(s)s - ml\phi(s)s^2 = U(s) \] .................(15)

Rearranging, the transfer function becomes:

\[
\frac{\phi(s)}{U(s)} = \frac{\frac{ml}{q}s^2}{s^4 + \frac{b(l + ml^2)}{q}s^3 - \frac{(M + m)mg}{q}s^2 - \frac{bmg}{q}s} \] ....... (16)

Where:

\[ q = [(M + m)(l + ml^2) - (ml)^2] \] ........... (17)

The pole and zero at the origin can be canceled and the TF becomes the following.

\[
G_1(s) = \frac{\phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(l + ml^2)}{q}s^2 - \frac{(M + m)mg}{q}s - \frac{bmg}{q}s} \left[ \frac{\text{rad}}{N} \right]
\]

A transfer function represents the relationship between a single input and a single output at a time. Since there are two outputs and a single input, it is possible to generate another transfer function illustrating the relationship of the input \(u(s)\) and the output \(X(s)\). This becomes.

\[
G_2(s) = \frac{x(s)}{y(s)} = \frac{\frac{(l + ml^2)s^2 - gmt}{q}}{s^4 + \frac{b(l + ml^2)}{q}s^3 - \frac{(M + m)mg}{q}s^2 - \frac{bmg}{q}s} \left[ \frac{m}{N} \right] [1]
\]

It is evident from the above two equations, the inverted pendulum system is single input/multi-output (SIMO). This means that a controller will attempt to control both pendulum’s angle and cart position, but for the classical control methods e.g. rootlocus controller design interest is only to control the pendulums position. This is because the techniques used will be best suited for single input/single output systems (SISO).
2.2 CONTROL SYSTEMS

To control a physical system or process using a digital controller, the controller must receive measurements from the system, process them, and then send control signals to the actuator that effects the control action.

When digital computer is used as a controller for a physical system, the process of signal conversion is essential so that the digital and analog components can be interfaced in the same system. The output signals of a device first undergo an analog-to-digital (A/D) conversion before they can be processed by a digital computer. Similarly, the coded signal from a digital controller must be decoded by a digital-to-analog (D/A) converter before it can be processed by an analog device. [2]

In most cases, both the plant and the actuator are analog in nature hence the need to use ADC and DAC. DAC is the translation from controller language (digital) to physical process language (analog). Conversely ADC is the translation from physical process language to controller language. The DAC and ADC converters are used simultaneously. The analogue-to-digital converter samples the output signal periodically and converts this sample to a digital word to be processed by the digital computer and thus generate a control strategy in the form of a number. The DAC then converts the numerical control strategy generated by the digital computer from a digital word to an analogue signal.

Feedback control will be used in this project. The system will maintain a prescribed relationship between the output and the reference input by comparing them and using the difference as a means of control i.e. in classical design philosophy, we decide at the outset that there should be a feedback from the output to the input reference so that the error between the two signals is formed; the controller then operates on the error signal in such a way that the design specifications is met;
2.2.1 DIGITAL CONTROL

Digital controllers are becoming more and more common. They have gained popularity and importance in all industries due to the advances made in digital computers and more recently in microcomputers. In fact many new digital controller applications are being stimulated by microprocessor technology including control of various aspects of automobiles and household appliances. [2]

Digital control offers distinct advantages over analog control that explain its popularity. These include:

Digital controllers are capable of performing complex computations with constant accuracy at high speed. Digital computers can have almost any desired degree of accuracy in computations at relatively little increase in cost. On the other hand, the cost of analogue computers increases rapidly as the complexity of the computations increases if constant accuracy is to be maintained. Digital controllers are extremely versatile than analogue controllers. The program which characterizes a digital controller can be modified to accommodate design changes, or adaptive performances, without any variations on the hardware. By merely issuing a new program, one can completely change the operations being performed. This feature is particularly important if the control system is to receive operating information or instructions from some computing center, where economic analysis and optimization studies are being made. [3]

Digital components in the form of electronic parts, transducers and encoders, are often more reliable, more rugged in construction, and more compact in size than their analogue equivalents.
Digital processing of control signals involves addition and multiplication by stored numerical values. The errors that result from digital representation and arithmetic are negligible. By contrast, the processing of analog signals is performed using components such as resistors and capacitors with actual values that vary significantly from the nominal design values. [3]

An analog controller is difficult to modify or redesign once implemented in hardware. A digital controller is implemented in firmware or software and its modification is possible without a complete replacement of the original controller. Furthermore, the structure of the digital controller need not follow one of the simple forms that are typically used in analog control. More complex controller structures involve a few extra arithmetic operations and are easily realizable. [2]

These and other glaring comparisons are rapidly converting the control system technology into a digital one.

In general, control systems are designed to perform specific tasks. The requirements imposed on the control systems are usually spelled out in the performance specification. They relate to accuracy, relative stability and speed of response.

In order to achieve a specific performance specification, a controller/compensator needs to be incorporated.

2.2.2 TYPES OF CONTROLLERS

The types of controllers used in digital control system are: P, PI, PD and PID controllers

P Controller:

A proportional controller increases gain hence reduces Steady State Error (SSE) but may not give the necessary SSE performance needed in a system because it may make the system unstable. The proportional term produces an output value that is proportional to the error value. The proportional response can be adjusted by multiplying the error by a constant $K_p$, called the proportional gain constant.
PI Controller:

An integral controller may give the required SSE performance but has a negative impact in terms of the speed of the response and overall stability of the system. This controller has a proportional term and an integral term. The controller produces an output value that is proportional to the error value and another value that is proportional to the integral of the error signal. In the z-domain, the PI controller introduces a pole at $z=+1$ and a zero at $z=0$. The location of the zero, being to the left side of the pole, contributes a phase lag to the system under consideration. [4]

PD controller:

The PD controller also increases the stability of the system, by decreasing overshoot and increasing the speed of response. Therefore, it improves the transient response of the system. This controller has a proportional term and a derivative term. The controller produces an output value that is proportional to the error value and another value that is proportional to the derivative of the error signal. Adding a derivative term to the system contributes a phase lead because of the positions of the pole and zero added.

PID controller:

The PID controller combines all the elements of the PI and the PD controller. It improves both the transient and steady state responses of the system.

2.3 HARDWARE

Lego bricks can be assembled and connected in many ways, to construct such objects as vehicles, buildings, and even working robots. Anything constructed can then be taken apart again, and the pieces used to make other objects. [5]

The LEGO Mindstorm EV3 will be used to implement the walking robot. This is a third set of mindstorm kits that were recently released. This set includes motors, sensors, the ARM9-based microcontroller brick and 550+ LEGO Technic elements. I will look at the components at a time:

1. LEGO mindstorm EV3 brick
2. Servo motors
3. Gyroscope (Gyro sensor)
The robot in this project will use feedback from a single axis gyro sensor to maintain vertical balance while it moves around in motorized wheels.

2.3.1 LEGO MINDSTORM EV3 BRICK

In Lego software, the Linux operating system is referred to as a firmware. The intelligent brick can be programmed using official software available for Windows and Mac computers, and is downloaded onto the brick via Bluetooth or a USB cable. This brick can be programmed using various languages e.g. the mindstorm EV3 language which is a box programming, Simulink and ROBOT C.

![Image of LEGO EV3 brick](image)

Figure 3.2 Brick

2.3.2 SERVO MOTORS

A servo motor is an automatic device that uses error-sensing feedback to correct the error in the mechanism. Servo motors are used more frequently because they give much more precision and rapidity to the movements of a mechanical system.

These motors have a built-in rotation sensor that enables the control of the robot’s movement precisely. The rotation sensor measures the motor rotations in degrees or full rotations (accurately +/- one degrees). Each motor in the EV3 set has a built-in rotation sensor to measure the motor’s position and speed. [7]
2.3.3 GYROSCOPE

Gyroscope is a device for measuring and maintaining orientation, based on the principles of angular momentum. In short, gyro sensors sense the rotational motion and changes in orientation in a single axis.

In the mindstorm EV3, the gyro sensor contains a single axis gyroscopic sensor that detects rotation and returns a value representing the number of degrees per second of rotation. It can measure up to +/- 360 per second of rotation. The sensor can be read 300 times per second (an interval of 3.33 times per second), which correspond to a sample rate of 300Hz.

Its design consists of a freely-rotating disk called a rotor, mounted onto a spinning axis in the center of a larger and more stable wheel.
Figure 5.4 Gyro sensor
CHAPTER 3

3. DESIGN AND METHODOLOGY

The design of the walking robot was done in the following ways:

- Implementation of the transfer function of the inverted pendulum.
- Study the stability and performance of the system.
- Set the desired performance of the system.
- Design a digital controller to meet the desired specification.
- Study the compensated system.
- Implement a digital controller into hardware.

3.1 SIMULATION

For simulation purposes of the system, MATLAB was used to study the response and performance of the control design.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and a programming environment. Furthermore, it has sophisticated data structures which contain built-in editing and debugging tools, and supports object-oriented programming. These factors make it an excellent tool for teaching and research.

3.1.1 SYSTEM PARAMETERS

For the project under consideration, we will have the following parameters of the inverted-pendulum system:
| \( I \) | 0 \( \text{kgm}^2 \) | Moment of inertia |
| \( l \) | 0.3m | Length of the pendulum |
| \( b \) | 0.1N/m/s | Coefficient of friction |
| \( \phi \) | - | Vertical pendulum angle |
| \( M \) | 0.5kg | Pendulum weight |
| \( m \) | 0.03kg | Cart weight |
| \( G \) | 9.81m/sec\(^2\): | Gravity acceleration |

And were applied in the transfer function below:

\[
G_1(s) = \frac{\phi(s)}{U(s)} = \frac{\frac{ml}{q} s}{s^3 + \frac{b(I + ml^2)}{q} s^2 - \frac{(M + m)mgl}{q} s - \frac{bmgl}{q} s} \quad \left[ \frac{\text{rad}}{N} \right]
\]

Substituting the parameters into the equation yields, the analog plant below:

\[
G_1(s) = \frac{6.667s}{s^3 + 0.2s^2 - 45.73s - 6.533}
\]
3.1.2 S-DOMAIN INVERTED PENDULUM TRANSFER FUNCTION

3.1.2.1 POLE ZERO MAP OF THE OPEN LOOP SYSTEM

The pole position of the linearized model of the inverted pendulum in open loop configuration indicates that the system is unstable, as one of the poles of the transfer function is located on the right half of the s-plane. That pole is located at $s = 6.7347$ thus the system is absolutely unstable.

![Pole Zero Map of Open Loop System](image)

Figure 6.1 Pole Zero Map of the Open Loop System
3.1.2.2 STEP RESPONSE OF THE OPEN LOOP SYSTEM

A step response of the system also shows instability as shown in the figure 3.2 below. The response grows unbounded.

![Step Response of Open Loop](image)

Figure 7.2 Step response

3.1.2.3 ROOTLOCUS ANALYSIS

The rootlocus exists on the right half of the s-plane indicating instability.

![Root Locus Of Uncompensated System](image)

Figure 8.3 Rootlocus
3.1.3 Z-DOMAIN INVERTED PENDULUM TRANSFER FUNCTION

A large number of control systems in operation in the industry are continuous-time systems. Rather than carrying out a completely new design when a digital controller is to be designed, it is possible to redesign the system to arrive at an equivalent digital system. Transformation of an analog design to digital domain is hence required. Three methods of discretization include:

- ZOH equivalence
- Pole-zero mapping
- Bilinear transformation

The plant needs to be converted from continuous time to discrete time. The continuous system was hence sampled at the Nyquist rate. The zero-order sample of hold was used to achieve this conversion.

A band limited signal can be fully reconstructed from its samples, provided that the sampling rate exceeds twice the maximum frequency in the band limited signal. This minimum sampling frequency is called the Nyquist rate. This result, usually attributed to Nyquist and Shannon, is known as the Nyquist–Shannon sampling theorem. [9]

The bandwidth of the plant was hence found from the bode pole.

![Bode Diagram](image.png)

Figure 9.4 Bode Plot
The corresponding frequency to the -3dB point from the maximum gain magnitude was found to be 4.64 rad/sec.

The Nyquist rate:

\[ \omega_s \geq 2\omega_b \]

\[ \omega_b = 4.64 \text{rads/sec} \]

\[ \omega_s \geq (2 \times 4.64) \]

\[ \omega_s \geq 9.28 \text{rads/sec} \]

The sampling period:

\[ \omega_s = 2\pi f_s \]

\[ f_s = \frac{w_s}{2\pi} \]

But:

\[ T = \frac{2\pi}{f_s} = \frac{2\pi}{w_s} \]

\[ \frac{2\pi}{9.28} = 0.677 \text{sec} \]

The sampling time should be less than or equal to 0.677s. Hence 0.1 is a good choice. The discretization method we will use is the zero-order hold (‘zoh’).

\[ G1(z) = \frac{0.0344z^2 - 0.0002324z - 0.03416}{z^3 - 3.454z^2 + 3.427z - 0.9802} \]

Sample time (T): 0.1 seconds
3.1.4 STUDY OF THE DISCRETE-TIME TRANSFER FUNCTION

3.1.4.1 POLE ZERO MAP OF THE DISCRETE SYSTEM

From figure 3.5, it is observed that the discrete plant has two zeros at \( z=1 \) and \( z=-0.9932 \). The open loop poles are situated at \( z=1.9610, z=0.9858 \) and \( z=0.5070 \). It is evident that the system is unstable as there is a pole located outside the unit circle.
3.1.4.2 STEP RESPONSE.

A step response of the system also shows that the system is highly unstable as phi diverges very rapidly. This runaway nature of the response indicates instability.

![Step Response in Z Domain](image)

Figure 11.6 Step response

3.1.4.3 ROOTLOCUS OF THE DISCRETE PLANT

The rootlocus of the discrete plant is as shown in figure 3.7. Most of the rootlocus is outside the unit circle.

![Root Locus of the Discrete System](image)

Figure 12.7 Rootlocus
3.2 DIGITAL CONTROLLER DESIGN

It is therefore apparent from the analysis that some sort of control will need to be designed to improve the stability and response of the system. The control system is to be designed in digital domain.

3.2.1 PERFORMANCE CRITERIA

Settling time of less than 5 seconds. This implies that the pendulum will achieve stability in less than 5 seconds. This ensures that the pendulum does not take too long to obtain an upright position.

Pendulum should not move far away from the point of equilibrium.

3.2.1.1 PID CONTROLLER

PID stands for Proportional-Integral-Derivative. This is a type of feedback whose output, a control variable, is generally based on the error between some defined set point and some measured process variable. Each element of the PID controller refers to a particular action taken on the error. The error is then used to adjust some input to the process in order to achieve its defined set point.

Three parameters are designed in the PID controller and each parameter as an effect on the error. The transfer function of the PID controller is written as [4]:

\[ G(z) = \frac{(K_p + K_i + K_d)z^2 - (K_p + 2K_d)z + K_d}{z(z - 1)} \]

Where:

\[ K_p = \text{proportional gain} \]
\[ K_i = \text{integral gain} \]
\[ K_d = \text{derivative gain} \]
The PID controller has two poles: \( z = 0 \) and \( z = 1 \) and two zeros which are both somewhere at the right half plane.

The proposed controller is a Proportional, Integral and Derivative (PID) controller. This is the best suited controller due to the following reasons:

- It increases the system type by 1 thereby improving the steady error tracking
- It can contribute a phase that can range between ±180
- It improves both the transient and steady state responses of the system

![Root Locus](image)

Figure 13.8 Rootlocus

Adding the poles of the PID which are located at \( z=0 \) and at \( z=1 \) yields the rootlocus in figure 3.8 above. The two zeros that come with the PID controller therefore need to be placed strategically to ensure that the rootlocus is pushed to the right. One zero was chosen to be at \( z=1.9610 \) so that it can cancel out the pole of the discrete plant that is outside the unit circle. The other was chosen to cancel out the pole at \( z=0.5070 \) which would cancel the other plant pole.
3.3 SOFTWARE CHOICE

The mindstorm EV3 software was chosen to program the hardware. This software is readily available at no cost and uploading of the software onto the robot is via USB.

The EV3 programs consist of programming blocks. Each block instructs the robot to do something different. Programs run the blocks one by one, beginning with the first block on the left.
3.3.1 FLOW CHARTS.

The program for the robot has two algorithms:

- Balancing algorithm
- Walking algorithm

3.3.1.1 BALANCING ALGORITHM

![Flow Chart: Balancing Algorithm](image)

Figure 14.9 Balancing Algorithm
**Balancing algorithm.**

The robot uses a PID controller to keep balance. Four physical signals are repeatedly measured and controlled to stay close to zero at all times. These signals are:

1. Robot angle relative to the expected balance. This is the angle of the robot. This value is positive if the robot is leaning forwards and negative if leaning backwards. The units is degrees.
2. Robot angular velocity. This indicates the rate at which the angle is changing. The units are in degrees/sec.
3. Motor encoder position. This indicates the position of the robot at a particular instant. This is an important reading in the balancing algorithm because the robot should be able to balance at one point without much deviation from its initial position i.e. it is the term that keeps the robot in a particular place. This position is in degrees of the motor.
4. Motor encoder speed. This gives the change in position in time. It is the speed of the motors.

Variance relative to the desired position is the error. This error is the added total of the above four error signals, each multiplied by a gain.

The total added error was kept close to zero. The power applied to the driving motors was the sum of the total error (Proportional). The accumulated error in time (Integral), and the change of the error in time (Derivative). Each of these was multiplied by a relative weighing factor $K_i$, $K_d$ and $K_p$.

The values of $K_p$, $K_i$ and $K_d$ were found using trial and error. The table below was used to determine which parameter to change. These values were varied interactively while observing the behavior of the robot. It was a time consuming activity.

Power to the motors = $K_p \cdot \text{error} + K_i \cdot \text{integral} + K_d \cdot \text{derivative}$.

The sampling is done at 20 milliseconds.
3.3.1.2  WALKING ALGORITHM.

Table: Parameter Changes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Rise Time</th>
<th>Overshoot</th>
<th>Settling Time</th>
<th>Steady State</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Decrease</td>
<td>Increase</td>
<td>Small Change</td>
<td>Decrease</td>
</tr>
<tr>
<td>I</td>
<td>Decrease</td>
<td>Increase</td>
<td>Increase</td>
<td>Eliminate</td>
</tr>
<tr>
<td>D</td>
<td>Small Decrease</td>
<td>Decrease</td>
<td>Decrease</td>
<td>None</td>
</tr>
</tbody>
</table>

Figure 15. Walking algorithm
**Walking algorithm**

Once the balancing algorithm worked, the walking algorithm was incorporated in the code to make the robot move around. This was done by powering the motors of the robot after 10 seconds of running the balancing algorithm. A wait command was used to pause the program for a set amount of time.

This allows the robot to get the desired position before powering the robot. Since the robot will be walking on a prescribed path, the time that the robot would need to walk before meeting an obstacle was measured. This time was found to be 15 seconds. After this time has elapsed, the robot was programmed to make an about turn and the robot was to be stopped manually by pressing the stop button on the EV3 block.

The mode that this robot works on is called multitasking. It is executing two algorithms at the same time.

### 3.4 BUILDING.

The robot components were put together to build the robot. This was a very fun and interesting part of the entire project. It involves putting together the different parts of the technic elements into the desired robot shape.

A gyro sensor was placed at the top of the robot where the change in angular velocity will be felt maximally. The axis of measurement is in the vertical plane with the gyro sensor positioned with the black cap facing upwards as shown: The gyro sensor was connected to the EV3 sensor port 2 using a standard EV3 wire.

![Figure 16.9.2 Vertical gyro sensor](image)
Two large motor are connected to the output ports on the EV3 labelled B and C.

Figure 17.9.3 Assembly of the robot

Figure 18.9.4: The robot after completion
CHAPTER 4

4. RESULTS AND ANALYSIS

4.1 SIMULATION RESULTS

4.1.1 EFFECTS OF THE PID ON THE SYSTEM.

Effects of adding a pole at $z = 0$ and $z = 1$ on the rootlocus of the system.

From the reshaping of the rootlocus, it was observed that the system can be stabilized by a PID controller. The rootlocus for some range of gain is within the rootlocus.

![Rootlocus graph](image)

Figure 4.1 Rootlocus
Figure 4.2 Compensated step response

From the results of the step response, the settling time was found to be 1.76 seconds which is very acceptable. The overshoot percentage is 27.7% which was also acceptable.

This results indicate that the robot after it has undergone some displacement it goes back to its original position at equilibrium after a very short time. As in the figure 4.2 above, the overshoot is minimal which indicates that the pendulum does not move far away from the vertical. Since all of the given design requirements have been met, no further iteration is needed.

4.2 HARDWARE RESULTS

After trial and error the optimum values of the PID controller were:

\[ K_p = 0.6 \quad K_i = 13 \quad K_d = 0.005 \]

The robot was started at an upright position to ensure that the gyroscope was in a vertical position. The balancing algorithm works for 10 seconds before the robot starts moving. The robot covers a distance of approximately 1.2 meters for 20 seconds before it makes a roundabout. At this point the robot stops when the enter button on the EV3 is pressed.
4.2.1 LIMITATIONS

When starting the robot it keeps on toppling over and several attempts need to be made before it balances. Once it balances, it manages to balance throughout the balancing and walking algorithm. This can be rectified by always ensuring the robot is always at the vertical position when starting but due to human errors this is difficult.
Figure 4.19 The walking robot in progress (Side view)
Figure 4. 20 Robot in progress (Top view)
CHAPTER 5

5. CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

It is indeed possible to stabilize an inherently non-linear unstable system as observed from both the simulation of the system dynamics and the actual implementation of the walking robot. The objectives of my project were hence met.

5.2 RECOMMENDATION

5.2.1 SIMULATION

Linear Quadratic Regulator (LQR) controller should be implemented in the simulation portion as it easily caters for a multiple-input/multiple-output system unlike the classical method that was used in this project.

5.2.2 HARDWARE

A behavioural system can be implemented into the robot that will allow the robot to act autonomously and make its own decisions based on its surroundings. This can be in a form of an obstacle avoidance system that would aid the robot in making movements around a room without any problem. An ultrasonic sensor can be used.

The robot’s movements can also be tested on a sloppy area and a rough terrain.

For easier tuning of the PID controller, development tools can be used instead of tuning the PID after building it. It would take less time but extensive knowledge of the plant parameters would be needed.

5.3 APPLICATIONS

The inverted pendulum concept is a very important in the teaching of control system analysis. It sets a very good base for students to understand feedback systems and how controllers work.
The two wheeled concept has been used to build the Segway. These are two-wheeled, self-balancing, battery powered electric vehicles used in Europe mostly used for patrolling e.g traffic.

Figure 4.5 Segway
REFERENCES


[11]
APPENDIX

5.4 HARDWARE ASSEMBLY PROCEDURE

Figure 21 The two large motors

Figure 22 Connecting the two large motors together
Figure 23 Adding the brick to the robot

Figure 24 Adding the gyroscope to the robot
5.5 THE MINDSTORM EV3 COMPONENTS

Figure 25: Mindstorm EV3 components
5.6 MATLAB CODE COMMANDS

s = tf('s');
M=0.5;
m=0.2;
b=0.1;
l=0.3;
I=0;
g=9.8;
q = ((M+m)*(I+m*(l^2))-(m*l)^2);
G = (((m*l)/q)*s)/(s^3 + (((b*(I + m*(l^2)))/q)*s^2) - (((M+m)*m*g*l)/q)*s) - ((b*m*g*l)/q));

% Pole zero map
pzmap(G)
title('Pole Zero Map of Open Loop System');

% To get the position of the poles and zeros of the discrete plant
[zeros poles] = zpkdata(G,'v')

% step response
step(G,t)
title('Step Response of Open Loop');

% rootlocus
rlocus(G)
title('Rootlocus Of Uncompensated System');

% Bobe plot of the analog for analysis of sampling time
bode(G)

% Conversion to a digital plant
G1=c2d(G,0.01,'zoh')

% To get the position of the poles and zeros of the discrete plant
[zeros poles] = zpkdata(G1,'v')

% to get the plot of the poles and zeros on the z-plane
pzmap(G1)
title('Pole Zero Map in Z Domain');

% to get the step response
step(G1);
title('Step Response in Z Domain');

% to plot the rootlocus
rlocus(G1)
title('Rootlocus of the Discrete System');
% Poles of the pid
num = [0 0 1]
den = [1 -1 0]
Ts = 0.1;
pid = tf(num, den, Ts)

% Poles and zeros for the PID
num = [1 -2.531 1.11777]
den = [1 -1 0]
Ts = 0.1;
pid1 = tf(num, den, Ts)

GC = G1 * pid1;
rlocus(GC)

pidtool(G1, 'pid')
5.7 THE MAIN PROGRAM

Figure 26: Main balancing and walking program
Figure 27: Pid program