UNIVERSITY OF CALIFORNIA, SAN DIEGO

LQG Controller Design of the Mobile Inverted Pendulum

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Engineering Science (Mechanical Engineering)

by

Zhu Zhuo

Committee in charge:

Professor Mauricio de Oliveira, Co-Chair
Professor Thomas Bewley, Co-Chair
Professor Sonia Martinez

2017

The Thesis of Zhu Zhuo is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

Co-Chair

_____

Co-Chair

University of California, San Diego

2017

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEGEMENT

I would like to express deepest appreciation to my advisor, Professor Mauricio de Oliveira, for helping and encouraging me to finish the master thesis. He never fails to steer me in the right direction and provide me with all the right tools to figure out such an interesting problem. I am also grateful that he answered my questions for countless hours with patience. Besides, he is a great mentor for my life.

I would also like to thank my parents for providing me with passionate support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This thesis would not have been possible without them.

ABSTRACT OF THE THESIS

LQG Controller Design of the Mobile Inverted Pendulum

by

Zhu Zhuo

Engineering Science (Mechanical Engineering)

University of California, San Diego, 2017

Mauricio de Oliveira, Co-Chair

Thomas Bewley, Co-Chair

This study aims at designing a controller that can stabilize a single inverted pendulum system using velocity feedback alone.

A state-space model for the EduMIP inverted pendulum system is obtained through dynamic analysis using a Lagrangian method. System identification experiments are proposed and performed to identify the parameters of the model. Based on the identified model, an LQG controller, which is a combination of optimal estimation and optimal control, is designed and implemented that can successfully stabilize EduMIP. At last, few tests are carried out to evaluate the performance of the controller, showing how the proposed LQG controller outperforms classical controller in many aspects.

Chapter 1
Introduction

Mobile robots have been the focus of study and research over few decades. The two-wheeled inverted pendulum robot is one of the most worth-mentioning configuration of mobile robots that receiving special attention. Due to its nonlinearity and intrinsically unstable dynamics property, a good controller is needed to keep the robot in upright position. Thus, two-wheeled inverted pendulum robots are ideal platforms to demonstrate and test the performance of different control strategies.

1.1 MIP

MIP, short for mobile inverted pendulum, is a robotic platform that resembles an autonomous Segway-like vehicle with two independently-actuated wheels. Several MIP prototype designs are developed at the UCSD Coordinated Robotics Lab, EduMIP (Figure 1) is one of the simplest of these designs. The simple structure, and stable running property makes it a perfect platform for our research.

Figure 1. EduMIP

We will introduce some important hardware of EduMIP and also the complementary filter algorithm in this section, which is used in IMU.

1.1.1 BeagleBone Black and Robotics Cape

Most small robotics projects in the educational space will have fairly minimal processing requirements, but will benefit greatly from having a wide variety of connectivity interfaces. This will allow a single embedded system to be used on different projects with unique requirements and functions. For this reason, we choose BeagleBone Black development board as it offers the power and functionality of a full Linux-based operating system with a small form factor that is well-suited for robotics projects.

The expansion headers on BeagleBone allow for accessory boards called Capes to be stacked on top [1]. Capes allow developers to add sensors and hardware specific to their application. We are using a Robotics Cape for this project, which adds IMU measurement, quadrature encoder counting and DC motor control to EduMIP.

1.1.2 IMU (MPU-9150)

An inertial measurement unit, known as IMU, works by detecting the rate of acceleration using accelerometers, and detects changes in rotational attributes like pitch, roll and yaw using gyroscopes. It collects angular velocity and linear acceleration data and sends them to main processor.

Attached on EduMIP is a MPU-9150 9-axis IMU, a popular sensor among developers. It combines a 3-axis gyroscope, a 3-axis accelerometer and magnetometers, which allow us to estimate the robot's orientation in space, on the same silicon die. Furthermore, the MPU-9150 contains an onboard digital motion processor (DMP). This microprocessor continuously runs digital low and high pass filters on the accelerometer and gyroscope signals respectfully in

addition to estimating orientation which can be read directly over an I2C bus and then interpreted as a quaternion vector.

1.1.3 Encoder

A rotary encoder, is an electromechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code. There are two main types: absolute and incremental. The output of absolute encoders indicates the current position of the shaft, making them angle transducers. The output of incremental encoders provides information about the motion of the shaft, which is typically further processed elsewhere into information such as speed, distance and position.

The encoders of MIP are quadrature encoders, which is a type of incremental encoders. They normally have at least two channel pulses, each of them will produce digital pulses when wheels are in motion. These pulses will follow a particular pattern that allows you to tell which direction the thing is moving, and by measuring the number of pulses per second, the angular speed can be derived. The encoder disks we use have 15 slots and therefore provide 60 counts per motor armature revolution [1].

1.1.4 Complementary filter

The raw data of θ can be obtained using accelerometer by calculating the relationship between two acceleration components of gravity. The raw data of $\dot{\theta}$ can be obtained directly using gyroscope.

However, we are not able to get an accurate and smooth angle of the body directly from IMU, but manage to do this with the help of complementary filter. The basic idea behind complementary filter is to take slow moving signals from accelerometer and fast moving signals from a gyroscope and combine them. As accelerometer gives a good indicator of

orientation in static conditions. While gyroscope gives a good indicator of tilt in dynamic conditions. So it quite reasonable to pass the accelerometer signals through a low-pass filter and the gyroscope signals through a high-pass filter, then combine them to give the final rate . The key point here is that the coefficient of low-pass and high-pass filters add up to 1 at all frequencies [2], we firstly split 1 into two parts:

$$\theta(s) = \frac{\lambda}{s+\lambda}\theta(s) + \frac{s}{s+\lambda}\theta(s) \tag{1}$$

Since

$$\dot{\theta}(s) = s\theta(s) \tag{2}$$

We therefore establish a relationship between θ and $\dot{\theta}$ in s domain:

$$\theta_{filtered}(s) = \frac{\lambda}{s+\lambda}\theta(s) + \frac{1}{s+\lambda}\dot{\theta}(s) \tag{3}$$

Choosing a proper value of parameter λ can optimize the estimate result.

1.2 Control Strategy

Motion of MIP is under-actuated, which implies the number of inputs to the system is less than the number of degree of freedoms to be stabilized [3]. More specifically speaking, MIP is a single-input and multi-outputs (SIMO) system, which makes it hard to apply conventional approaches to control the system. In classical control theorem, the way to overcome this problem is to divide SIMO system into successive single-input and single-output (SISO) system, and then stabilize the desired degree of freedom separately (e.g. PID-PID). However, we can directly deal with this SIMO system using modern control schemes (e.g. LQR), which gives us a better understanding of stabilizing the whole system.

Being familiar with classic controller for MIP, which is a position feedback controller will be introduced in next section. We are seeking for a modern control strategy with velocity feedback. There are three advantages of velocity feedback control over position control:

No need to know the balance offset

The IMU is attached to the robotics cape. However, one thing worth noticed is that when EduMIP is balanced, the cape is not upright. In other words, the equilibrium MIP reaches is not $\theta = 0$, but $\theta = 0 + offset$. It is necessary to measure the angle offset, in order to compensate the offset for the actual equilibrium.

Velocity control has no offset problem, the actual equilibrium will always happened when the velocity of body goes to zero. Instead of measuring the offset by experiment, we can simply focus on the unchanged zero velocity equilibrium.

Simpler controller

Velocity control allow us to drop one state from state-space, it is now a third-order state-space model rather than a four-order one. Since the LQG controller have the same order state-space model compared to the plant, the reduction on plant also implies that the order of controller is reduced by one.

It is often desirable to reduce the order of controller. On the one hand, higher order controllers may place many poles and zeros to achieve a good performance. However, a large number of poles and zeros may also affect other parts of frequency response of closed-loop system which may not be well modeled and lead to lack of robustness. On the other hand, a lower order controller is constrained to places less poles and zeros which tend to be focused on

shaping the important part of frequency response curve only. Reduced order controllers are also easier to implement.

Better measurement

The raw data of angle θ is always not accurate unless there is no accelerations except gravity. The more extra accelerations there are, the worse measurement we will get. However, the raw data of $\dot{\theta}$ is good even when IMU moving fast. In order to balance the MIP, we do care about the situation that large extra accelerations happens, it is a better choice to use the measured velocity as feedback signals rather than measured angle.

1.2.1 Classic Control

The scope of classical control theory is limited to single-input and single-output (SISO) system design. The system analysis is carried out in time domain using differential equations, in complex-s domain with Laplace transform or in frequency domain by transforming from the complex-s domain. Due to easier physical implementation of classical controller designs as compared to systems designed using modern control theory, these controllers are preferred in most industrial applications. The most common controllers designed using classical control theory are PID controllers.

In this section, we consider one input $V(t)$ (voltage) and two outputs of the system $\theta(t)$ (angle of the body) and $\phi(t)$ (angle of the wheels) to demonstrate how to design a simply successive loop closure controller to stabilize the MIP.

The transfer functions from voltage to two angles and two equilibriums of the system are given in section 3.3, which are:

$$TF_{V\to\theta} = \frac{-74.1s}{s^3 + 34.7s^2 - 128.4s - 1555.4}$$

$$TF_{V\to\phi} = \frac{113.8s^2 - 8418}{s\cdot(s^3 + 34.7s^2 - 128.4s - 1555.4)}$$

$$Equilibrium\ 1:\ \theta(t) = 0,\ \dot{\theta}(t) = 0,\ \dot{\phi}(t) = 0,\ V(t) = 0$$

$$Equilibrium\ 2:\ \theta(t) = \pi,\ \dot{\theta}(t) = 0,\ \dot{\phi}(t) = 0,\ V(t) = 0$$

Since there is a zero-pole cancellation in the first transfer function, we are not able to stabilize the whole system at equilibrium 1 by closing loop from voltage to the angle of the body only. MIP will move at a constant speed in this situation, which means $\dot{\phi}(t) \neq 0$. Thus, it is necessary to close the second loop from voltage to the angle of the wheel. We come up with a successive-loop structure as Figure 2 shows:

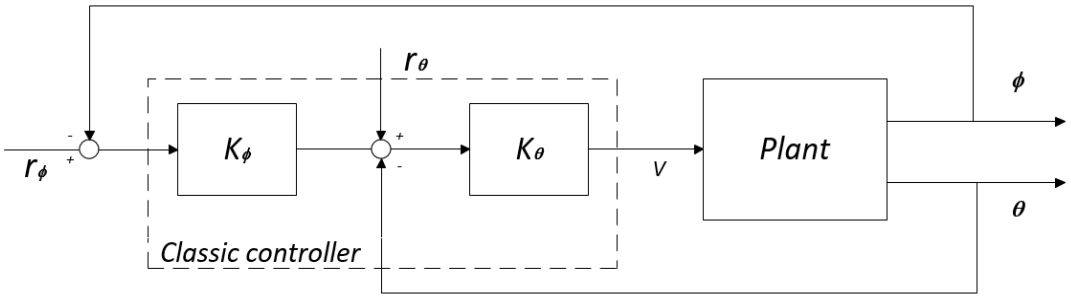

Figure 2. Block diagram of classical control

Inner loop

We firstly close the loop from $V(t)$ to $\theta(t)$, which is the inner loop of successive loops. By looking at the root locus plot, we know there is a zero-pole cancellation and we can't stabilize the inner loop without placing an unstable pole in the controller.

By placing only an unstable pole at 26.5, and a gain of -26, we successively stabilized the loop from $V(t)$ to $\theta(t)$. The controller is:

$$K_\theta = \frac{-26}{s - 26.5} \tag{4}$$

Closed-loop root locus plot of $\theta(t)$:

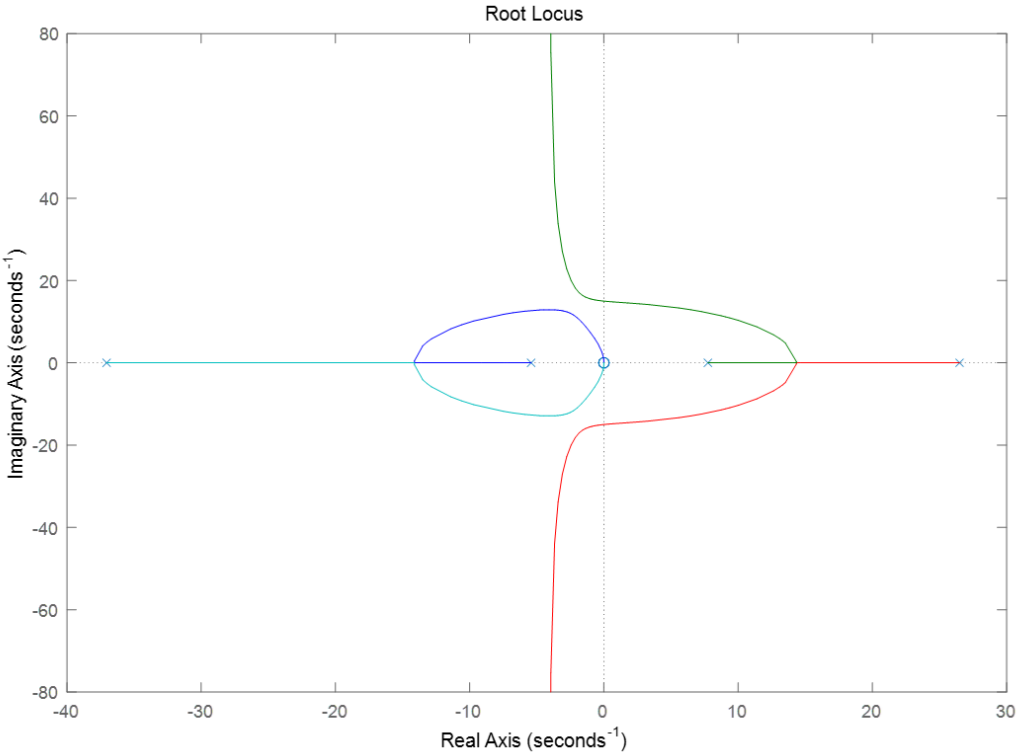

Figure 3. Root locus plot of classical control

Closed loop Nyquist plot also implies a good gain margin and phase margin of the inner loop controller:

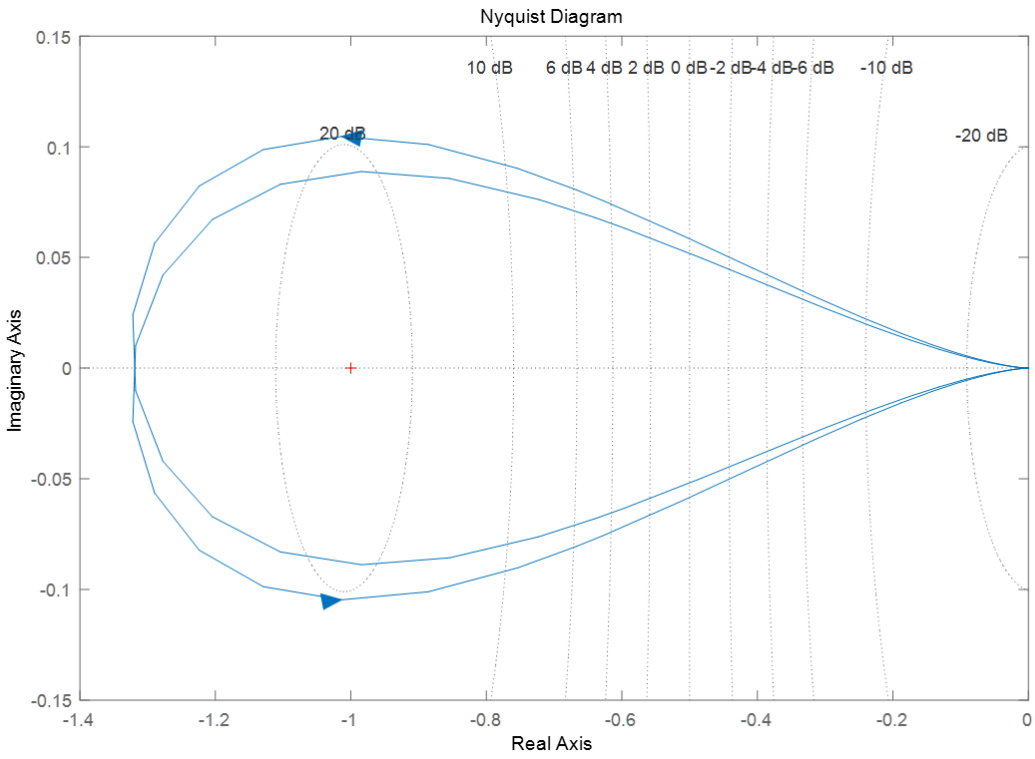

Figure 4. Nyquist plot of classical control

Outer loop

We then close the loop from $V(t)$ to $\phi(t)$, which is the outer loop of successive loop. By looking at the open loop root locus plot, we can observe that choosing K_ϕ as a little gain is enough to stabilize $\phi(t)$. The controller we design is:

$$K_\phi = 0.05 \tag{5}$$

Rearranging the block diagram gives a better structure of the controller like Figure 5, we are able to stabilize the system with (4) and (5):

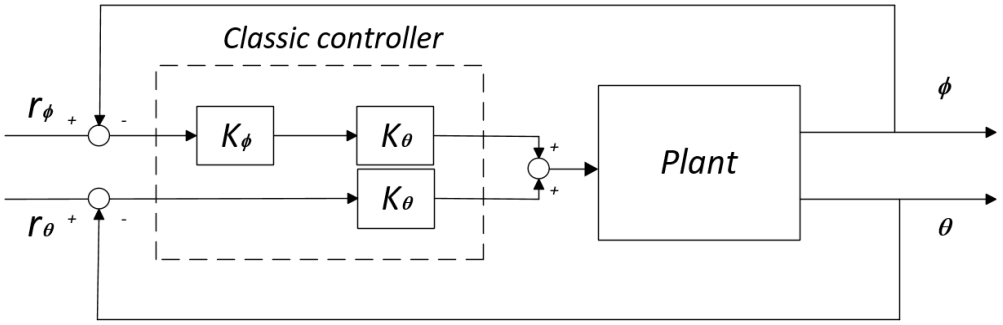

Figure 5. Rearranged block diagram of classical control

Essentially, this classic double-loop controller is trying to imitate the behavior of modern controller by tuning two gains for two feedback, then adding then to become one input to the system. The only difference is that classic method can't determine multiple gains at the same time, it will divide the single-input and multi-output (SIMO) system into successive SISO systems, then determine gains separately. The problem of such a process is that the inherent relationships between these outputs to input will not be considered, and thus will lead to an unstable controller in our case.

1.2.2 Modern Control

Modern control theory is carried out strictly in the complex-s or the frequency domain, and can deal with multi-input and multi-output (MIMO) systems. This overcomes the limitations of classical control theory in more sophisticated design problems. In modern design, a system is represented as a set of first order differential equations defined using state variables. What is more, modern control theory can also deal with nonlinear system and time-variant systems.

While the modern controller deal with all the outputs at the same time. By looking into the inherent relationships of the system, all the gains (K_θ and K_ϕ in our case) can be determined together by certain rules as Figure 6, which may give us a stable controller.

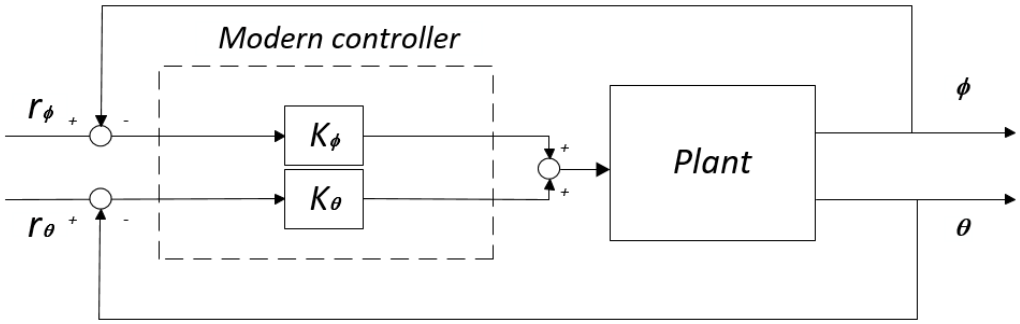

Figure 6. Block diagram of modern control

The goal of this thesis is to design a well-performed LQG controller for EduMIP. In next chapter, we will establish a state-space model of the EduMIP, this is the first step to solving a control problem using modern control theory.

Chapter 2
Modeling

Before looking into control system of EduMiP, it is critical to get an accurate model that characterizes EduMiP reasonably. An accurate model will help us design a controller fit for the system, thus makes it possible to implement the controller on EduMIP.

Two assumptions are made here to idealize the model:

1. Both body and wheels are considered as rigid body.
2. No sliding between wheels and ground and thus no sliding friction.

2.1 Equations of Motion

We use Lagrangian method to obtain equations of motion for the system [4].

Simplify the MIP to Figure 7 below:

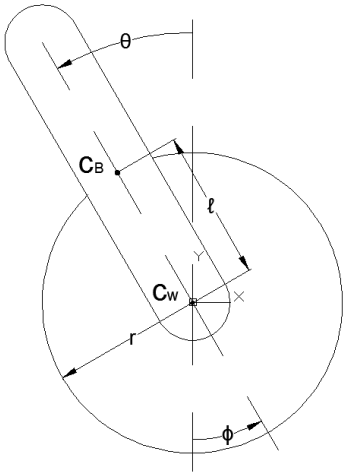

Figure 7. Simplified MIP model

Take the center of mass of wheels as origin of a static coordinate system, X and Y direction are given as shown. θ and ϕ are angles of body and wheels, and both are positive in counter-clockwise direction.

The translational velocity of wheels is

$$v_W = \dot{c}_W = (-r\dot{\phi}(t), 0) \tag{6}$$

Where:

r	radius of wheels
$x(t) = -r\phi(t)$	translation displacement of wheels
$c_W = (x(t), 0)$	center of mass of wheels

The translational velocity of body is

$$v_B = \dot{c}_B = (-r\dot{\phi}(t) - l\cos\theta(t)\cdot\dot{\theta}(t), -l\sin\theta(t)\cdot\dot{\theta}(t)) \tag{7}$$

Where:

l	distance between c_W and c_B
$c_B = (x(t) - l\sin\theta(t), l\cos\theta(t))$	center of mass of body

The kinetic energy of system can be described as

$$K = \frac{1}{2} m_B \cdot {v_B}^2 + 2\cdot\frac{1}{2} m_W \cdot {v_w}^2 + \frac{1}{2} I_B \cdot \dot{\theta}^2 + 2\cdot\frac{1}{2} I_W \cdot \dot{\phi}^2 \tag{8}$$

Where:

m_W	mass of single wheel
m_B	mass of body

G_r	gearbox ratio
I_m	motor armature inertia
I_B	inertia of body
$I_W = m_W r^2 / 2 + G_r^2 \cdot I_m$	inertia of single wheel with gearbox

The potential energy of system can be described as

$$U = m_B g l \cdot \cos\theta \tag{9}$$

Lagragian, which is the difference between the kinetic and potential energy of the system is therefore:

$$L = K - U \tag{10}$$

According to Lagrange equations of motion in Cartesian coordinates for a point mass subject to conservative forces, namely,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_i}\right) - \frac{\partial L}{\partial x_i} = 0 \tag{11}$$

Any non-conservative forces acting on the point mass would show up on the right hand side.

So equation of motions are:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} &= -\tau(t) \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) - \frac{\partial L}{\partial \phi} &= \tau(t) \end{aligned} \tag{12}$$

Substitute equations (6) – (10) into (12), we have equation of motions:

$$
\begin{aligned}
&(I_B + m_B l^2)\ddot{\theta}(t) + m_B l(-g\sin\theta(t) + r\cos\theta(t)\cdot\ddot{\phi}(t)) = -\tau(t) \\
&m_B rl\cos\theta(t)\cdot\ddot{\theta}(t) - m_B rl\sin\theta(t)\cdot\dot{\theta}(t)^2 + (2I_W + (m_B + 2m_W)r^2)\ddot{\phi}(t) = \tau(t)
\end{aligned}
\tag{13}
$$

τ is the torque provided by the motor, can be calculated as:

$$
\begin{aligned}
\tau(t) &= 2G_r(\bar{s}\cdot u - k\cdot\omega_m) \\
\omega_W &= \dot{\phi} - \dot{\theta} = \omega_m / G_r
\end{aligned}
\tag{14}
$$

Where:

$V(t)$	motor voltage
$V_{\max}$	maximum voltage of motor
$u = V(t)/V_{\max}$	normalized motor duty cycle
$\bar{s}$	motor Stall Torque
k	motor Constant
ω_m	motor armature speed
ω_W	wheel speed

Furthermore, we set up following substitution rules for simplicity:

a	$2I_W + (m_B + 2m_W)r^2$
b	$m_B rl$
c	$I_B + m_B l^2$
d	$m_B gl$
e	$2G_r\bar{s}/V_{\max}$
j	$2G_r{}^2 k$

Thus equation (13) is now:

$$\begin{aligned} d\sin\theta(t) + j\dot{\phi}(t) &= eV(t) + j\dot{\theta}(t) + c\ddot{\theta}(t) + b\cos\theta(t)\cdot\ddot{\phi}(t) \\ j\dot{\phi}(t) + b\cos\theta(t)\cdot\ddot{\theta}(t) + a\ddot{\phi}(t) &= eV(t) + \dot{\theta}(t)(j + b\sin\theta(t)\cdot\dot{\theta}(t)) \end{aligned} \tag{15}$$

2.2 Nonlinear Model

Solve for $\ddot{\theta}(t)$ and $\ddot{\phi}(t)$ in equations (15), we have

$$\begin{aligned} \ddot{\theta}(t) &= -\frac{1}{ac - b^2\cos\theta(t)^2}(-ad\cdot\sin\theta(t) + ae\cdot V(t) + be\cdot\cos\theta(t)\cdot V(t) + aj\dot{\theta}(t) \\ &+ bj\cdot\cos\theta(t)\cdot\dot{\theta}(t) + b^2\cos\theta(t)\cdot\sin\theta(t)\cdot\dot{\theta}(t)^2 - aj\cdot\dot{\phi}(t) - bj\cdot\cos\theta(t)\cdot\dot{\phi}(t)) \\ \ddot{\phi}(t) &= -\frac{\sec\theta(t)}{b^2 - ac\cdot\sec\theta(t)^2}(-bd\cdot\sin\theta(t) + be\cdot V(t) + ce\cdot\sec\theta(t)\cdot V(t) + bj\dot{\theta}(t) \\ &+ cj\cdot\sec\theta(t)\cdot\dot{\theta}(t) + bac\cdot\tan\theta(t)\cdot\dot{\theta}(t)^2 - bj\cdot\dot{\phi}(t) - cj\cdot\sec\theta(t)\cdot\dot{\phi}(t)) \end{aligned} \tag{16}$$

From equations (16), we have nonlinear model of MiP:

$$\begin{pmatrix} \dot{\theta}(t) \\ \dot{\phi}(t) \\ \theta(t) \\ \phi(t) \end{pmatrix} = \begin{pmatrix} -\frac{1}{ac - b^2\cos\theta(t)^2}(-ad\cdot\sin\theta(t) + ae\cdot V(t) + be\cdot\cos\theta(t)\cdot V(t) + aj\dot{\theta}(t) \\ +bj\cdot\cos\theta(t)\cdot\dot{\theta}(t) + b^2\cos\theta(t)\cdot\sin\theta(t)\cdot\dot{\theta}(t)^2 - aj\cdot\dot{\phi}(t) - bj\cdot\cos\theta(t)\cdot\dot{\phi}(t)) \\ -\frac{\sec\theta(t)}{b^2 - ac\cdot\sec\theta(t)^2}(-bd\cdot\sin\theta(t) + be\cdot V(t) + ce\cdot\sec\theta(t)\cdot V(t) \\ +bj\dot{\theta}(t) + cj\cdot\sec\theta(t)\cdot\dot{\theta}(t) + bac\cdot\tan\theta(t)\cdot\dot{\theta}(t)^2 - bj\cdot\dot{\phi}(t) - cj\cdot\sec\theta(t)\cdot\dot{\phi}(t)) \\ \dot{\theta}(t) \\ \dot{\phi}(t) \end{pmatrix} \tag{17}$$

2.3 Linearized Model

We choose four states, inputs and outputs of MiP as following:

$$\begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} = \begin{pmatrix} \dot{\theta}(t) \\ \dot{\phi}(t) \\ \theta(t) \\ \phi(t) \end{pmatrix} \qquad \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} \dot{\theta}(t) \\ \dot{\phi}(t) \end{pmatrix} \qquad u_1(t) = V(t) \tag{18}$$

Equilibriums can be obtained by solving equations (15) when all angular speeds and accelerations are 0.

$$\begin{aligned} &\textit{Equilibrium } 1\text{: } \theta(t)=0,\ \dot{\theta}(t)=0,\ \dot{\phi}(t)=0,\ V(t)=0 \\ &\textit{Equilibrium } 2\text{: } \theta(t)=\pi,\ \dot{\theta}(t)=0,\ \dot{\phi}(t)=0,\ V(t)=0 \end{aligned} \tag{19}$$

There are two equilibriums of such an inverted pendulum system. Equilibrium 1 is an unstable equilibrium, which we are interested in for control. While equilibrium 2 is a stable equilibrium can be achieved without any control.

Thus, we linearized equations (17) at equilibrium 1 with notation rules (18):

$$\begin{pmatrix} \dot{x}_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} = \begin{pmatrix} -\dfrac{(a+b)j}{-b^2+ac} & \dfrac{(a+b)j}{-b^2+ac} & \dfrac{ad}{-b^2+ac} & 0 \\ -\dfrac{(b+c)j}{b^2-ac} & \dfrac{(b+c)j}{b^2-ac} & \dfrac{bd}{b^2-ac} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} + \begin{pmatrix} -\dfrac{(a+b)e}{-b^2+ac} \\ \dfrac{(b+c)e}{-b^2+ac} \\ 0 \\ 0 \end{pmatrix} \cdot u(t) \tag{20}$$

We will have system identified in next chapter to get a more accurate model in numerical values. However, in order to have an ideal of the structure of the system, we plug in all the data from Appendix A for now, which is provided by James Strawson:

$$\begin{pmatrix} \dot{x}_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} = \begin{pmatrix} -13.528 & 13.528 & 175.457 & 0 \\ 17.735 & -17.735 & -115.561 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} + \begin{pmatrix} -90.456 \\ 118.585 \\ 0 \\ 0 \end{pmatrix} \cdot u(t) \tag{21}$$

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot u(t) \tag{22}$$

Equations (21) together with (22) are our linearized model.

Where

$$A=\begin{pmatrix} -13.528 & 13.528 & 175.457 & 0 \\ 17.735 & -17.735 & -115.561 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad B=\begin{pmatrix} -90.456 \\ 118.585 \\ 0 \\ 0 \end{pmatrix}$$

$$C=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad D=\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Therefore, we are ready to check the controllability and observability of MIP model for further work.

The controllable matrix C_o is of form:

$$C_o=[B \;\; AB \;\; A^2B \;\; A^3B]$$

The observable matrix O_b is of form:

$$O_b=\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix}$$

Using Matlab command ctrb(A,B) and obsv(A,C) we know both of matrixes are of full rank, which implies the system is controllable as well as observable.

2.4 Reduced Linearized Model

One thing we can observe from linearized model is the state $x_4(t)$ ($\phi(t)$) has no impact on any other states. It is also what happens in reality, MIP can balance up if the body is upright, no matter where the wheel is. What is more, $x_4(t)$ will not be used as an output in the velocity feedback. So it would be a good choice to reduce $x_4(t)$ from the four-state state-space model.

After reduction, the state-space model is:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{pmatrix} = \begin{pmatrix} -13.528 & 13.528 & 175.457 \\ 17.735 & -17.735 & -115.561 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} -90.456 \\ 118.585 \\ 0 \end{pmatrix} \cdot u(t) \quad (23)$$

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot u(t) \quad (24)$$

Where

$$A = \begin{pmatrix} -13.528 & 13.528 & 175.457 \\ 17.735 & -17.735 & -115.561 \\ 1 & 0 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} -90.456 \\ 118.585 \\ 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

It is also easy to verify that this reduced-model is still controllable and observable.

Chapter 3
System Identification

System identification is a methodology for building mathematical models of dynamic systems using measurements of the system's input and output signals.

Since we can derive model of EduMIP from Lagranigian principles, but are not so sure about numerical values of its parameters. We can estimate the value of its parameters from experiment date, which is known as grey-box modeling approach [5], to ensure that we are working on the correct model. We would like to conduct 2 separate experiments, the body experiment (θ Experiment) and the wheel experiment (ϕ Experiment), to identify the real parameters of the system.

3.1 θ Experiment

The first experiment we can perform is constraining the movement of the wheels (fix the wheels by force) and design a chirp signal as input to collect data of body's angular velocity. This experiment help us to identify parameters I_B, k and $\overline{s}$.

Since we constrained the movement of two wheels, it is obvious that:

$$\phi(t) = 0, \quad \dot{\phi}(t) = 0, \quad \ddot{\phi}(t) = 0$$

So there is only one equation of motion left:

$$eV(t) + j\dot{\theta}(t) + c\ddot{\theta}(t) = d\sin\theta(t) \tag{25}$$

Linearized the model around the stable equilibrium since it was where we performed the test. Which gave us a reduced state-space model:

$$\begin{pmatrix} \dot{\theta} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{d}{c} & -\frac{j}{c} \end{pmatrix} \cdot \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ -\frac{e}{c} \end{pmatrix} \cdot u(t) \tag{26}$$

This model describe the relationship between the voltage V and angular velocity of the body $\dot{\theta}$. Convert the state-space model to a transfer function from V to $\dot{\theta}$:

$$TF_{\dot{\theta}\_Model} = \frac{-\frac{e}{c}s}{s^2 + \frac{j}{c}s + \frac{d}{c}} \tag{27}$$

Plug in data from Appendix A, (27) is now:

$$TF_{\dot{\theta}\_Model} = \frac{-38.93s}{s^2 + 5.822s + 125.2} \tag{28}$$

The chirp signal properties are 1.85V as magnitude and start at 1 Hz, gradually increase logarithmically to 20 Hz in 30 seconds, and decreased back to 1 Hz logarithmically in 30 seconds. We apply this chirp signal few times, part of the response is shown as figure 8:

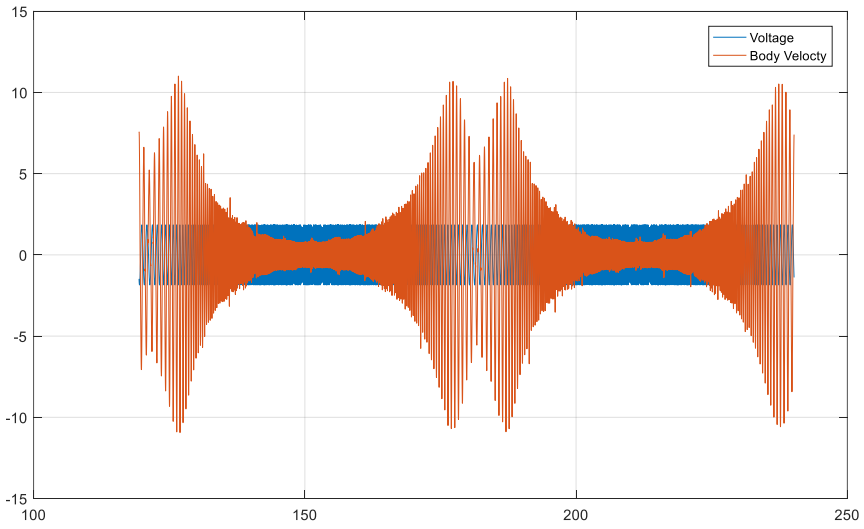

Figure 8. θ Experiment response

Using the plot above, we are able to estimate transfer function from the voltage V and angular velocity $\dot{\theta}$ given by the experiment. Constrained the estimation by setting the constant term of numerator to zero will help us get a transfer function in the same shape of (28):

$$TF_{\dot{\theta}\_Exp} = \frac{-34.81s}{s^2 + 6.432s + 99.54} \tag{29}$$

Which is close to transfer function (28).

Plot both of them at the same time for comparison, as figure 9:

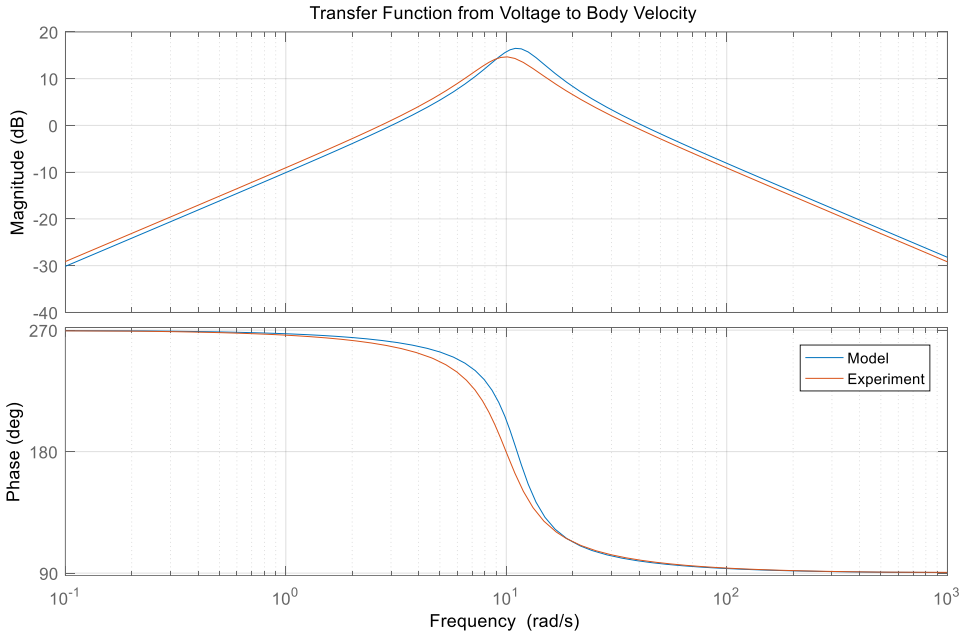

Figure 9. θ Experiment comparison

3.2 ϕ Experiment

The second experiment we can perform is a free run test, we can place the back of MIP to the table and observe the wheels velocity. This experiment help us to identify parameters I_W. One thing to mention, instead of consider

$$a = 2I_W + (m_B + 2m_W)r^2$$

We now consider

$$a = 2I_W$$

Since the body is no longer moving and there is no translational velocity of wheels, thus no m_B and m_W terms.

Also:

$$\dot{\theta}(t) = 0, \quad \ddot{\theta}(t) = 0$$

So there is only one equation of motion (one state state-space model) left:

$$\ddot{\phi}(t) = -\frac{j}{a}\dot{\phi}(t) + \frac{e}{a}u(t) \tag{30}$$

This model describe the relationship between the voltage V and angular velocity of wheels $\dot{\phi}$. Convert the state-space model to a transfer function from V to $\dot{\phi}$:

$$TF_{\dot{\phi}\_Model} = \frac{\frac{e}{a}}{s + \frac{j}{a}} \tag{31}$$

Plug in data from Appendix A, (31) is now:

$$TF_{\dot{\phi}\_Model} = \frac{235.8s}{s + 35.27} \tag{32}$$

The setting of this experiment is shown as Figure 10:

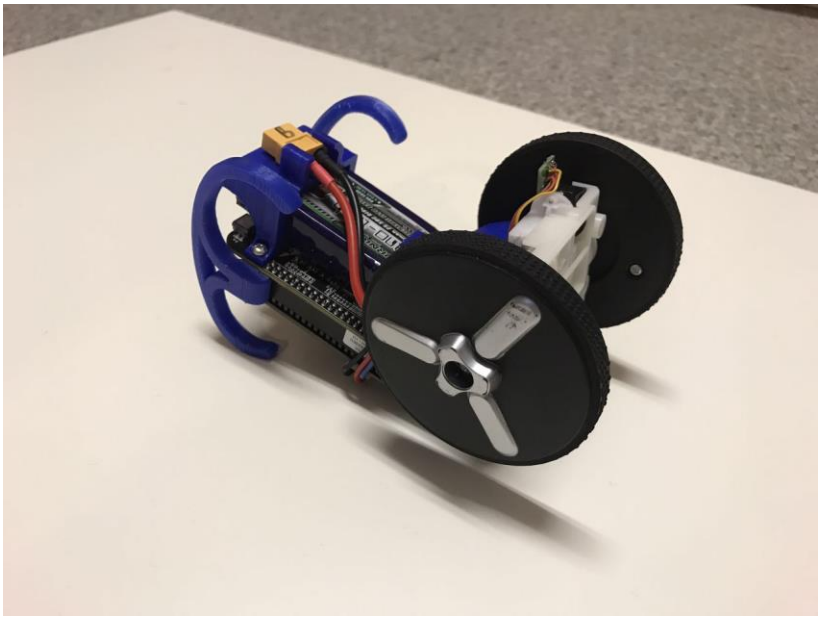

Figure 10. ϕ Experiment setting

The chirp signal properties are 1.85V as magnitude and start at 0.3 Hz, gradually increase logarithmically to 30 Hz in 30 seconds, and decreased back to 0.3 Hz logarithmically also in 30 seconds. We apply this chirp signal few times, part of the response is shown as figure 11:

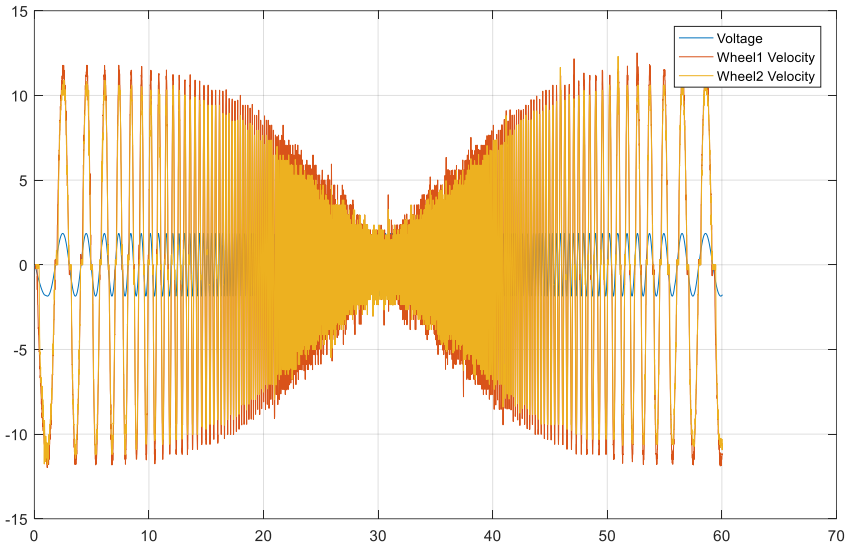

Figure 11. ϕ Experiment setting

Using the plot above, we are able to estimate transfer function from the voltage V and angular velocity $\dot{\theta}$ given by the experiment. Constrained the estimation by setting the constant term of numerator to zero will help us get a transfer function in the same shape of (32):

$$TF_{\dot{\phi}\_Exp} = \frac{252.7s}{s+45.24} \tag{33}$$

Which is close to transfer function (32).

Plot both of them at the same time for comparison, as Figure 12:

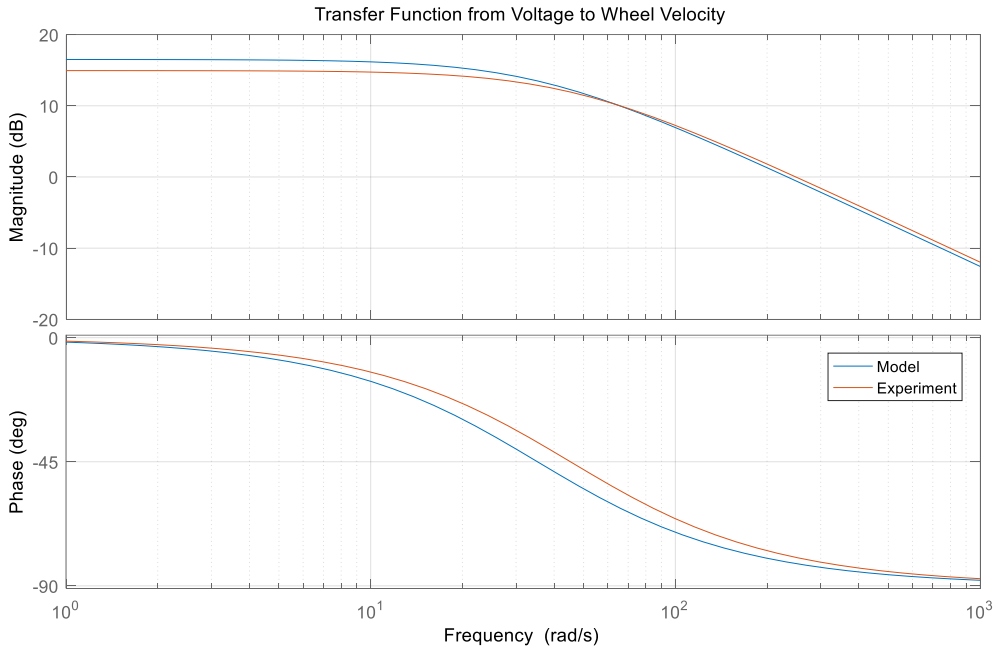

Figure 12. ϕ Experiment comparison

3.3 Identified Model

We firstly determine parameters that we want to identify, they are I_B, k, $\overline{s}$ and I_W. Other parameters are either easy to measure, such as m_W, or unnecessary to identify, like G_r. Then we build equations from transfer function (27), (29), (31) and (33):

$$\begin{aligned}
\frac{e}{c} &= \frac{2Gr^2 \cdot \overline{s}}{V_{\max}(I_B + m_B l^2)} &= 34.81 \\
\frac{j}{c} &= \frac{2Gr^2 k}{I_B + m_B l^2} &= 6.432 \\
\frac{d}{c} &= \frac{m_B g l}{I_B + m_B l^2} &= 99.54 \\
\frac{e}{a} &= \frac{Gr \cdot \overline{s}}{I_W V_{\max}} &= 252.7 \\
\frac{j}{a} &= \frac{Gr^2 k}{I_W} &= 45.24
\end{aligned}$$

Reversely substitute substitution rules showed in section 2.1 into equation above, we are able to solve for undetermined parameters as following expressions (though there are 5 equations above, we can only solve for 4 unknowns, since they are not totally independent):

$$\begin{aligned}
I_B &= \frac{m_B g l - m_B l^2 \cdot (d/c)}{d/c} &= 5.91\times10^{-4} \\
k &= \frac{m_B g l \cdot (j/c)}{2Gr^2 \cdot (d/c)} &= 2.37\times10^{-6} \\
\overline{s} &= \frac{m_B g l V_{\max} \cdot (e/c)}{2Gr \cdot (d/c)} &= 0.003375 \\
I_W &= \frac{m_B g l \cdot (e/c)}{2(d/c) \cdot (e/a)} &= 6.42\times10^{-5}
\end{aligned}$$

We can directly calculate the value of 4 unknown based on expressions, since m_B and l are all easy to measure and g, Gr and $V_{\max}$ are all known to us.

After substituting these expressions and other known parameters into the full state-space model, we obtained the identified linearized state-space model:

$$\begin{pmatrix} \dot{x}_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} = \begin{pmatrix} -13.692 & 13.692 & 128.381 \\ 21.023 & -21.023 & -83.514 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} -74.101 \\ 113.775 \\ 0 \end{pmatrix} \cdot u(t) \quad (34)$$

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \cdot u(t) \quad (35)$$

Where

$$A = \begin{pmatrix} -13.692 & 13.692 & 128.381 \\ 21.023 & -21.023 & -83.514 \\ 1 & 0 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} -74.101 \\ 113.775 \\ 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

We can still verify that identified state-space model is not only controllable, but also observable.

Bode plot of identified MIP is shown in Figure 13.

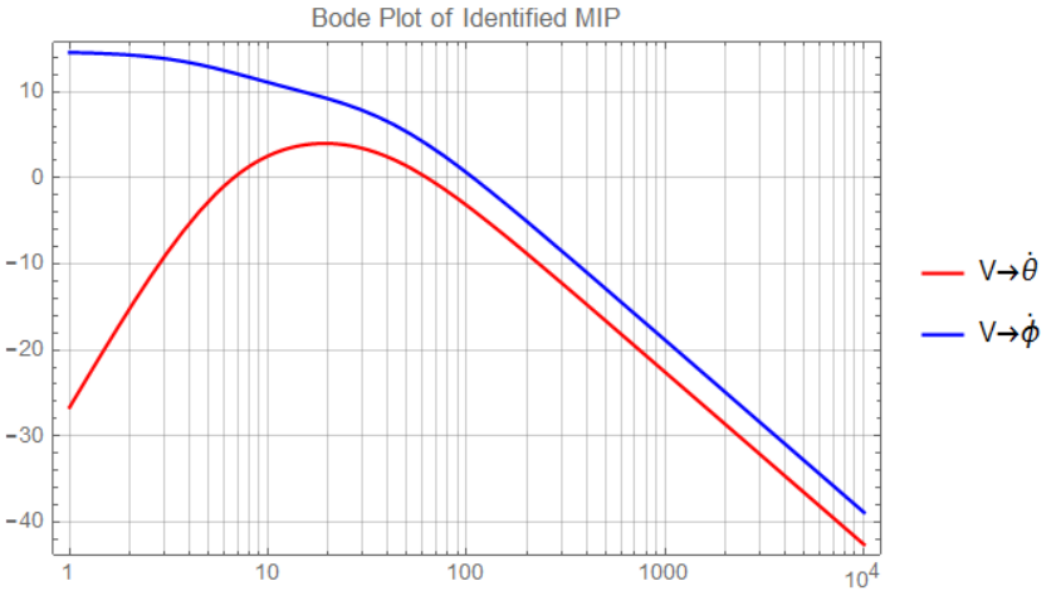

Figure 13. Bode plot of identified MIP

In addition, the transfer functions of plant can also be obtained:

$$TF_{V\to\theta} = \frac{-74.1s}{s^3 + 34.7s^2 - 128.4s - 1555.4}$$

$$TF_{V\to\phi} = \frac{113.8s^2 - 8418}{s\cdot(s^3 + 34.7s^2 - 128.4s - 1555.4)}$$

$$TF_{V\to\dot{\theta}} = \frac{-74.1s^2}{s^3 + 34.7s^2 - 128.4s - 1555.4}$$

$$TF_{V\to\dot{\phi}} = \frac{113.8s^2 - 8418}{s^3 + 34.7s^2 - 128.4s - 1555.4}$$

The first two transfer functions are used in section 1.2 for classic controller design, and it is also good to know poles of last two transfer function are -37.05, -5.42 and 7.75.

Chapter 4
Controller Design

In this section, we will firstly introduce some basic concepts of LQG controller, which is a LQR and a LQE [6]. Then the process of design a LQG controller is presented in few steps and the simulation closed-loop responses are also attached.

4.1 Linear Quadratic Regulator (LQR) Design

Linear Quadratic Regulator (LQR) problem is one fundamental problem of optimal control.

For a Linear Time Invariant (LTI) system:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{36}$$

Where x and u are the states and control.

LQR essentially generated an automated way of finding appropriate feedback controller:

$$u(t) = Kx(t) \tag{37}$$

Not only to stabilize the system (32), but also minimize a quadratic cost function, which of the form:

$$J = \int_0^\infty [x(t)^T Qx(t) + u(t)^T Ru(t)]dt \tag{38}$$

Where Q and R are weighting matrix.

The gain is K given as:

$$K = -R^{-1}B^T X \tag{39}$$

Where X is the solution to an Algebraic Riccati Equation (ARE)

$$A^T X + XA - XBR^{-1}B^T X + Q = 0 \tag{40}$$

Some assumptions here are:

1. $Q \geq 0, R > 0$
2. (A, B) is stabilizable.

Q and R are weighting matrix that penalize the states and control work, respectively. Tuning elements of Q will change performance of the states correspondingly. In the same way, we can tuning R to determine the ability of control for each input. The relative ratio of Q and R elements will emphasize whether we care more about control or system performance.

Generally speaking, weighting matrix differs based on different design goals and need some judgments to decide if "optimization" is reached. There is no obvious relationship can be observed between these two matrix and the behavior of the controller, though it does affect the behavior. So it is very important to understand how to tradeoff between performance and control, and choose nice weighting matrix in a LQR design.

4.2 Linear Quadratic Estimator (LQE) Design

In LQR problem, all the states are assumed to be available by sensor measuring and being ready to feedback. However, it is always not the case, the limit of sensors and budget won't give us all states. Therefore, an estimator is needed to provide with reasonable estimation of all the states, one of the estimators that also perform an "optimal" procedure is called Kalman Filter.

For a LTI system with noise:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bw(t) \\ y(t) &= C_y x(t) + v(t) \\ z(t) &= C_z x(t) \end{aligned} \tag{41}$$

Where $w(t)$ and $v(t)$ are both Gaussian zero mean white noise with variance W and V.

LQE essentially generated an automated way of finding appropriate estimation gain L for:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + L(\hat{y}(t) - y(t)) \\ \hat{y}(t) &= C_y \hat{x}(t) \\ \hat{z}(t) &= C_z \hat{x}(t) \end{aligned} \tag{42}$$

Not only stabilize the state estimation errors, which leads to a stable estimator, but also minimize a quadratic cost function, which of the form:

$$J = \lim_{t \to \infty} E[(z(t) - \hat{z}(t))^T (z(t) - \hat{z}(t))] \tag{43}$$

The gain L is given as:

$$L = -YC_y{}^T V^{-1} \tag{44}$$

Where Y is the solution to an Algebraic Riccati Equation (ARE):

$$AY + YA^T - YC_y{}^T V^{-1} C_y Y + W = 0 \tag{45}$$

One assumptions here is (A, C_y) should be detectable.

W and V are the process and measurement noise variance, respectively. Tuning W and V is also needed in order to achieve a high-performance estimate gain. Tuning elements of V will change the weights of how estimated states depending on each output, while tuning W will change how much the estimation depending on outputs. However, it is worth-mentioning

that W and V being close to what true variance guarantees only a good estimator, but not a good controller. We will talk about this later in tuning LQG controller.

Kalman filter is essentially a recursive estimator that implements states prediction, followed by a correction to produce estimations of unknown variables that tend to be more precise than those based on a single measurement alone.

4.3 Linear Quadratic Gaussian (LQG) Controller Design

The Linear Quadratic Gaussian (LQG) controller is simply the combination of a LQE with a LQR. The separation principle guarantees that they can be independently designed and computed [6].

So for the LTI system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_w w(t) \\ y(t) &= C_y x(t) + v(t) \\ z(t) &= C_z x(t) + Du(t) \end{aligned} \tag{46}$$

We can combine K and L obtained in equation (4) and (9) to design an observer-based controller:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + B_u u(t) + L(\hat{y}(t) - y(t)) \\ \hat{y}(t) &= C_y \hat{x}(t) \\ u(t) &= K\hat{x}(t) \end{aligned} \tag{47}$$

Which stabilize the closed loop system and minimize the cost

$$J = \lim_{t\to\infty} E[z(t)^T z(t)] \tag{48}$$

Here is a block diagram of closed-loop plant with LQG controller as Figure 14:

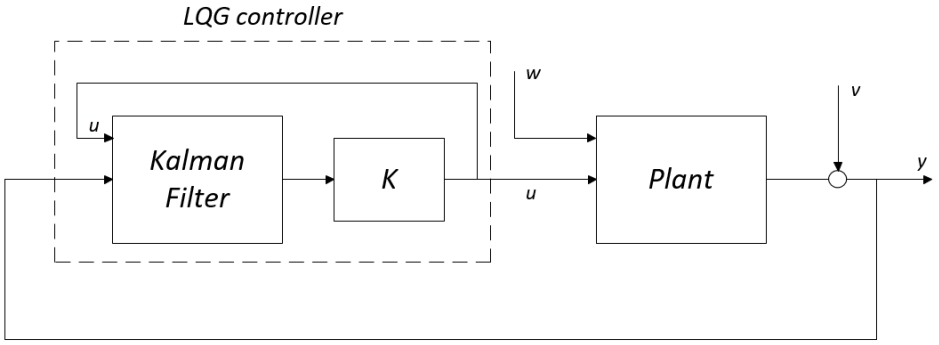

Figure 14. Block diagram of LQG control

4.3.1 Weighting Matrix Choice

After many trials, we determined a set of weighting matrix:

$$Q=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\qquad R=1\qquad W=4\qquad V=\begin{pmatrix}0.01&0\\0&0.1\end{pmatrix} \tag{49}$$

The transfer function of controller calculated based on these matrix is:

$$\begin{aligned}TF_{controller\_\dot{\theta}}&=-\frac{2199.26\cdot(s+8.535)\cdot(s+50.264)}{(s+4.754-3.074i)\cdot(s+4.754+3.074i)\cdot(s+1769.02)}\\TF_{controller\_\dot{\phi}}&=\frac{305.123\cdot(s+1.875)\cdot(s-112.967)}{(s+4.754-3.074i)\cdot(s+4.754+3.074i)\cdot(s+1769.02)}\end{aligned} \tag{50}$$

All the poles of this controller are with a negative real part, which implies the controller we designed is a stable controller. This is one of the advantages that LQG controller possesses, by choosing appropriate weighting matrix we are able to get a stale controller. This stable controller allowed us to set the initial condition θ_0 to a large non-zero value, while the initial condition θ_0 of classical controller can't exceed around 10 degrees.

Connect this controller to the system and simulate the close-loop response with initial condition θ_0.

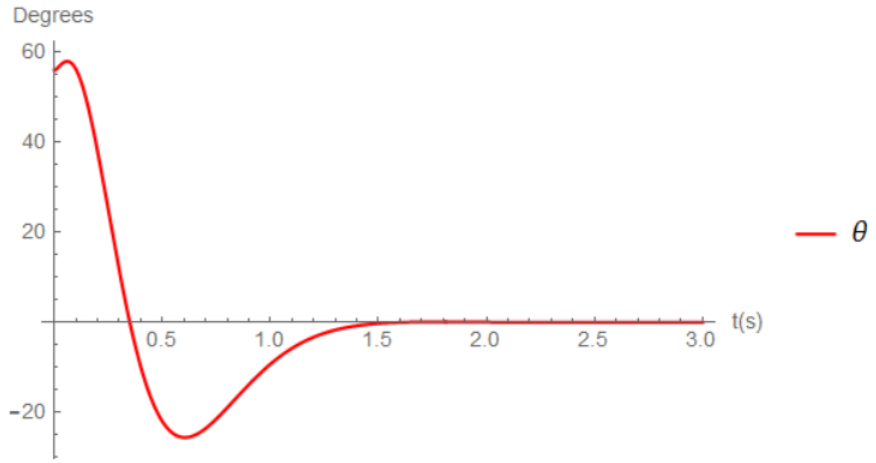

Figure 15. θ Response with LQG controller

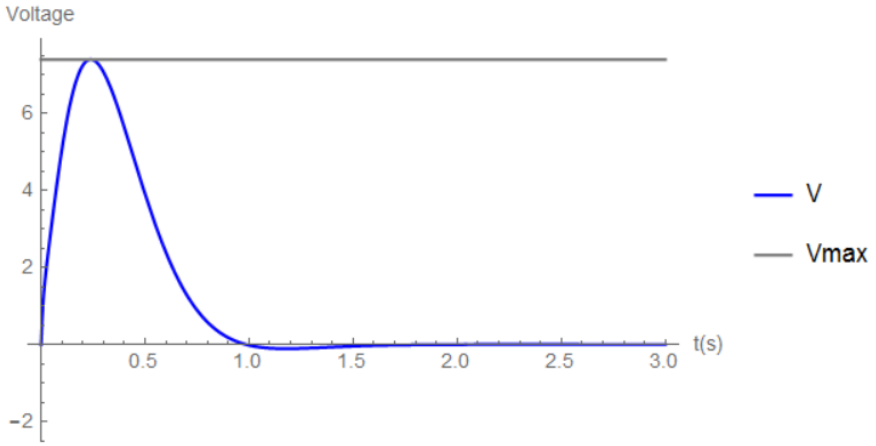

Figure 16. Voltage Response with LQG controller

According to the simulation result Figure 15 and 16, the maximum value of θ_0 could be balanced without exceeding maximum voltage is around 56 degrees. What's more, θ will converge to equilibrium in 2 seconds.

4.3.2 Controller Order Reduction

We would like to discretize the controller at 100Hz, which is also the frequency that the MIP is running at. Though we obtained a well-performed controller in simulation, it would still be problematic in implementation due to the large poles and zeros. All the zeros and poles corresponding to a frequency over or close to 100Hz may mess up the discretization and result in a totally different discrete controller. So we removed the large poles and zeros (we have tried balanced truncation method but it was not suitable for our case since it messed up low frequency response). It wouldn't be a problem for the system since the system barely response to a signal over 100Hz (see Figure 13), which can be observed from MIP's Bode diagram.

Poles we removed: -1769.02

Zeros we removed: -112.967

The reduced controller is a second-order controller instead of a third-order controller:

$$TF_{controller\_\dot{\theta}\_reduced} = -\frac{1.243\cdot(s+8.535)\cdot(s+50.264)}{(s+4.754-3.074i)\cdot(s+4.754+3.074i)}$$
$$TF_{controller\_\dot{\phi}\_reduced} = \frac{19.485\cdot(s+1.875)}{(s+4.754-3.074i)\cdot(s+4.754+3.074i)} \quad (51)$$

The comparison between (50) and (51) is shown in Figure 17 and 18:

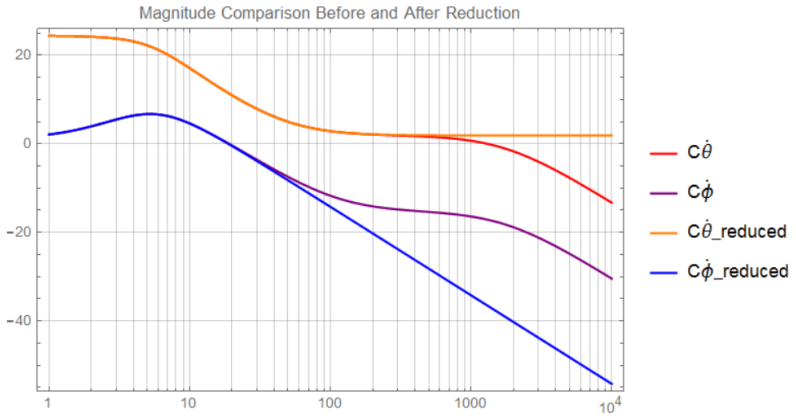

Figure 17. Bode plot comparison of controllers (Magnitude)

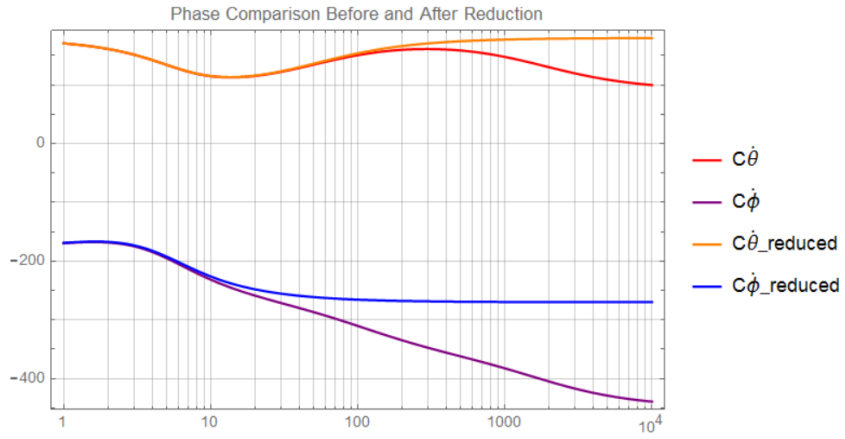

Figure 18. Bode plot comparison of controllers (Phase)

Connect reduced controller (51) to the system and simulate the close-loop response with initial condition θ_0. One thing good to know is the closed-loop system has 5 poles at -107.11, -10.858+0.849*i*, -10.858-0.849*i*, -3.759+2.481*i* and -3.759+2.481*i*.

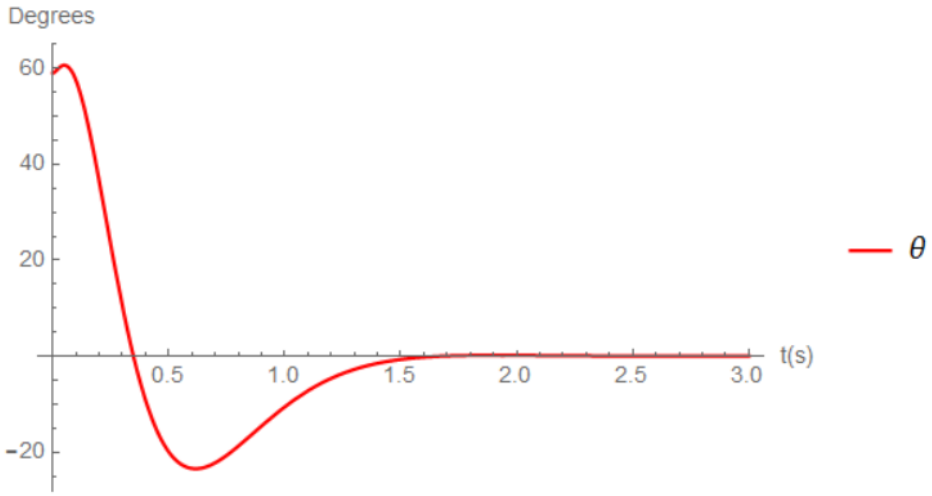

Figure 19. θ Response with reduced LQG controller

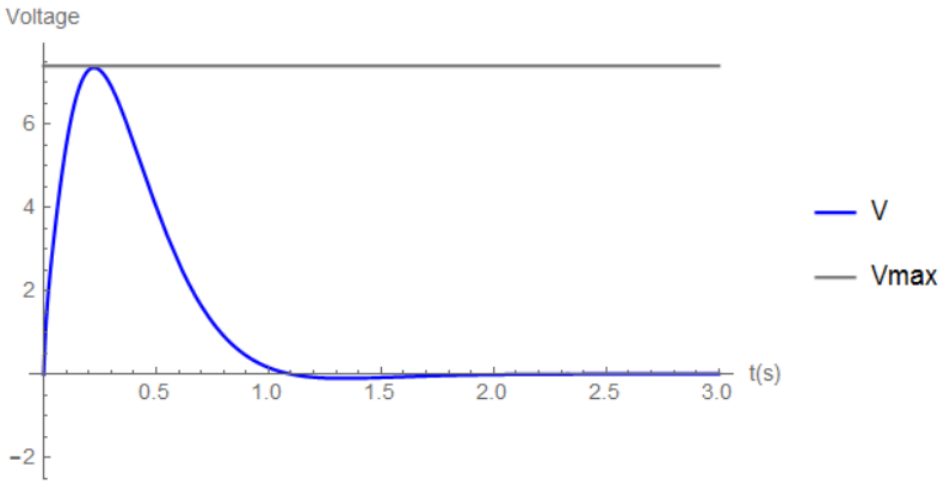

Figure 20. Voltage Response with reduced LQG controller

According to the simulation result Figure 19 and 20, the maximum value of θ_0 could be balanced without exceeding maximum voltage is around 59 degrees for reduced controller. What's more, θ will converge to equilibrium in 2 seconds.

It is obvious that after the controller order reduction, the closed-loop performance is still as good as before, even a little increasing of the maximum θ_0. So this is a successful order reduction, which almost keeps the closed-loop response unchanged.

4.3.3 Discretized Controller

Discretization concerns the process of transferring continuous models into discrete counterparts. This process is usually carried out as a first step toward making model suitable for numerical evaluation and implementation on digital computers.

We will discretize the order reduced controller at 100Hz using zero order hold method, which generates an exact discretization in the time domain for staircase inputs. We can constructs the continuous signal back by holding each sample value as a constant over one sample period [7]. The state-space model of discretized controller is:

$$SS_{Discrete\_LQG\_reduced} = \left(\begin{array}{cc|cc} 0.9129 & 0.0378 & 0.002835 & -0.0005967 \\ -0.0655 & 0.9933 & 0.001463 & -0.001286 \\ \hline -380.821 & 322.448 & -1.24321 & 0 \end{array}\right)$$

In order to simulate the discrete performance, we discretized the system at 100 Hz, too. We have:

$$SS_{Discrete\_MIP} = \left(\begin{array}{ccc|c} 0.89 & 0.1159 & 1.157 & -0.6274 \\ 0.1743 & 0.8222 & -0.6376 & 0.9621 \\ 0.0094 & 0.000613 & 1.006 & -0.00332 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right)$$

Connect this discretized reduced controller to the discretized system and simulate the close-loop response with initial condition θ_0.

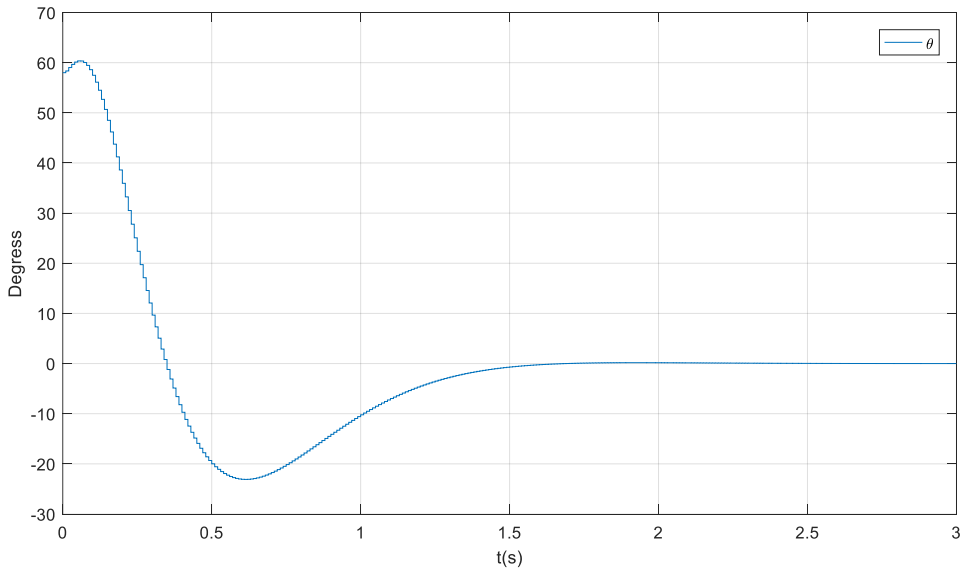

Figure 21. θ Response with discretized LQG controller

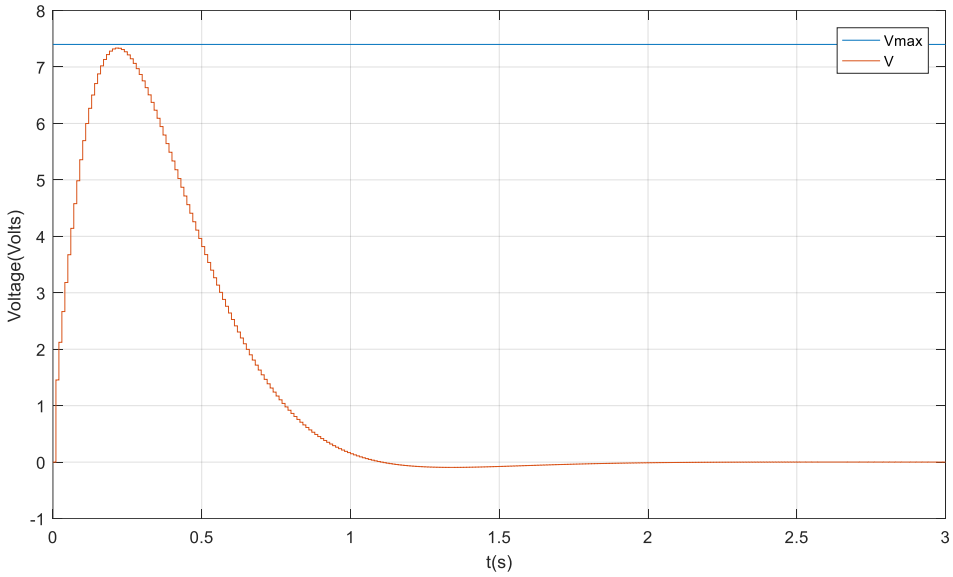

Figure 22. Voltage Response with discretized LQG controller

According to the simulation result Figure 21 and 22, the maximum value of θ_0 could be balanced without exceeding maximum voltage is around 58 degrees for discretized reduced controller. Conversion time is still 2 seconds.

Chapter 5
Controller Implementation

In this chapter, we briefly introduce how to implement the controller using Python and Jupyter Notebook. Then we conduct 4 tests to observe the performance of the MIP and also put some comments based on our test results.

5.1 Setup for Python server

We installed a Python 3.2.3 in the MIP for implementation use. Few steps help us to implement designed controller on the MIP, here are some brief instructions that can be followed to finish the setup.

Step1: Get root access into MIP

Keep the MIP and your device in the same network, SSH login the MIP

(Replace /etc/network/interfaces with your Wifi connection setting file, and replace ip_address with your device IP):

```
vim /etc/network/interfaces
ssh root@ip_address
```

Step2: Run the server on MIP

Locate at the directory containing the server file, then start the server in MIP

(Replace beaglebone/python with your directory, and replace ip_address with your device IP, Ts with your sampling frequency

```
cd beaglebone/python

python3 server.py –H ip_address –m mip3 –t Ts
```

Step3: Run the Notebook

First, start Notebook:

```
jupyter notebook
```

Then import useful heading files and functions, construct control structure in Notebook by creating and connecting signals and filters, import designed controller in form of state-space model and modify the codes based on your requests, run the notebook.

5.2 Implementation Tests and Results

5.2.1 Balancing Test 1 (Start close to equilibrium)

Balancing test staring near from equilibrium demonstrate the ability of the MIP to balance up and be stable. We conduct this test first since maintaining stable is the prerequisite for all further movements.

We started the test with the controller on, kept the MIP upright and released it. Here are responses of some important parameters.

Figure 23 is the θ response. We can tell that MIP has a strong ability of keeping stable uprightly. We noticed that the mean value of theta is about -23 degrees (marked out as a yellow line), which is the angle offset when it is balanced that we have mentioned in section 1.2. What is more, an oscillation of ± 10 degrees from the mean value is observed.

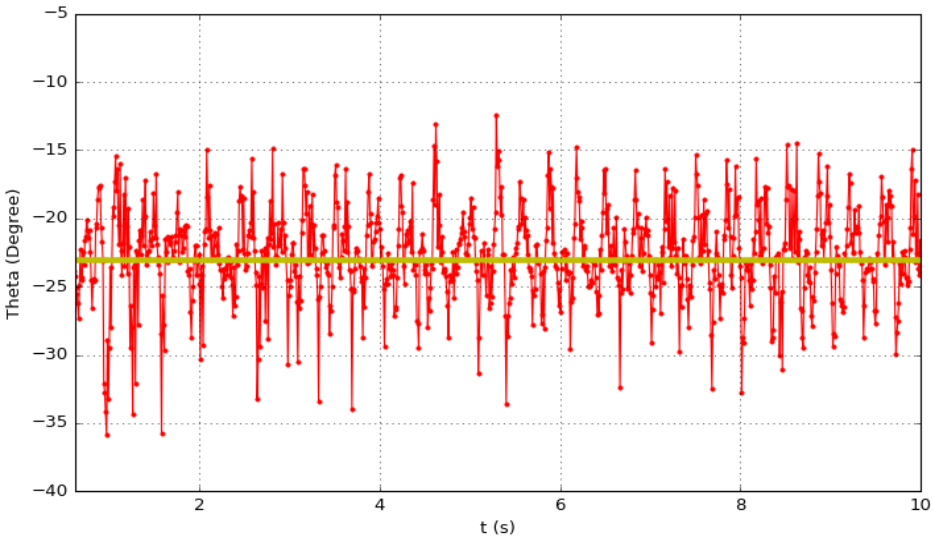

Figure 23. θ Response of balancing test 1

However, the response was quite noisy. Thus we got the response through a complementary filter as equation (3), with 2 as the value of λ. Then we got Figure 24, filtered θ Response. It is obvious that there were some oscillation periodically happened, a close look was desired as Figure 25 showed, filtered θ Response from 2s to 4s.

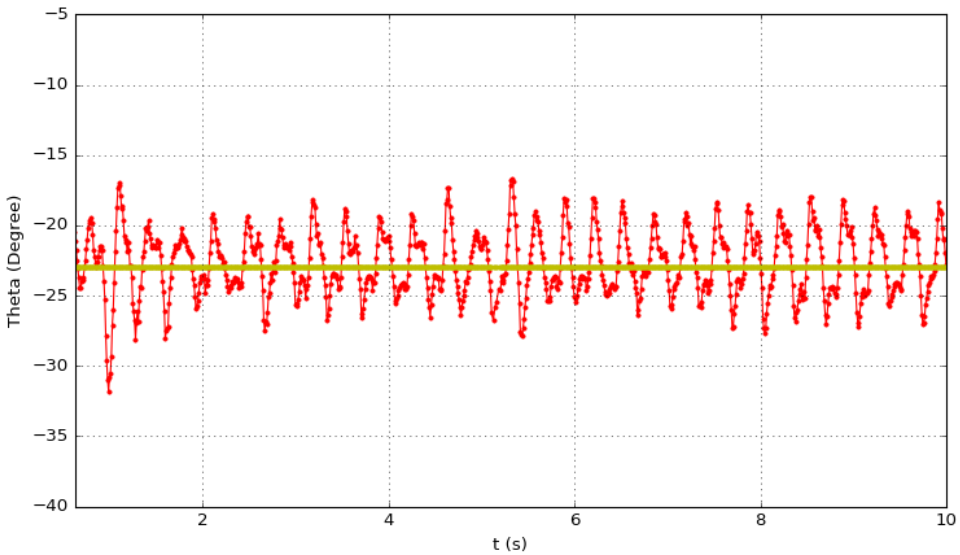

Figure 24. Filtered θ Response of balancing test 1

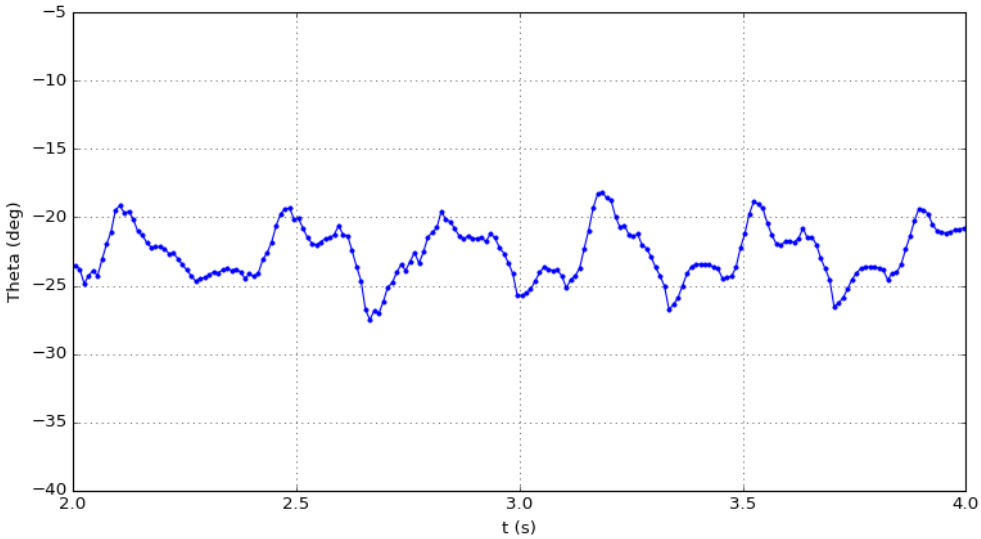

Figure 25. Close look of filtered θ Response of balancing test 1

Figure 26 is the voltage response, which is the response of motor voltage applied. It turned out that we had plenty of voltage to do more than stabilizing, because the voltage is bound by ± 2 V and far from the limit ± 7.4 V.

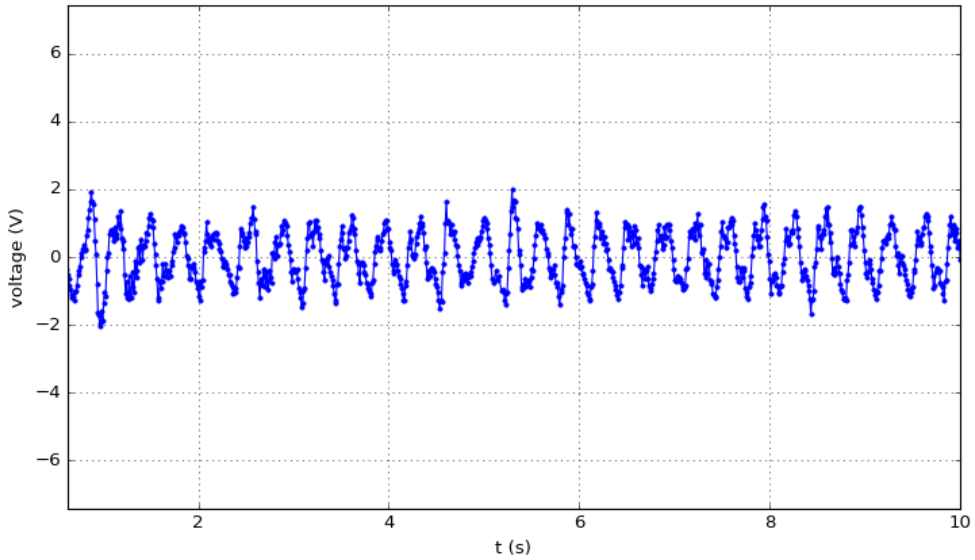

Figure 26. Voltage Response of balancing test 1

5.2.2 Balancing Test 2 (Start away from equilibrium)

Balancing test staring away from equilibrium demonstrate the ability of the MIP to balance up when it is not starting near the equilibrium. We conduct this test to see if the advantages that MIP can start from an unstable mode is true. We expected this the case for the modern controller since LQG controller we designed is a stable controller, therefore possible to correct the unstable initial conditions.

We started the test with the controller on, kept the MIP inclined with a certain angle and released it at 0.8 second (Orange line). Here are responses of some important parameters.

Figure 27 is the θ response. We can tell that MIP start at almost 23 degrees away from the desired equilibrium. However, it converged the equilibrium really quick in less than 2 seconds and since then became the same situation as last test. Figure 28 was a close look at what happened in first 2.5 seconds, from 0.8s to 1.8s is the transition period from an unstable mode to the equilibrium.

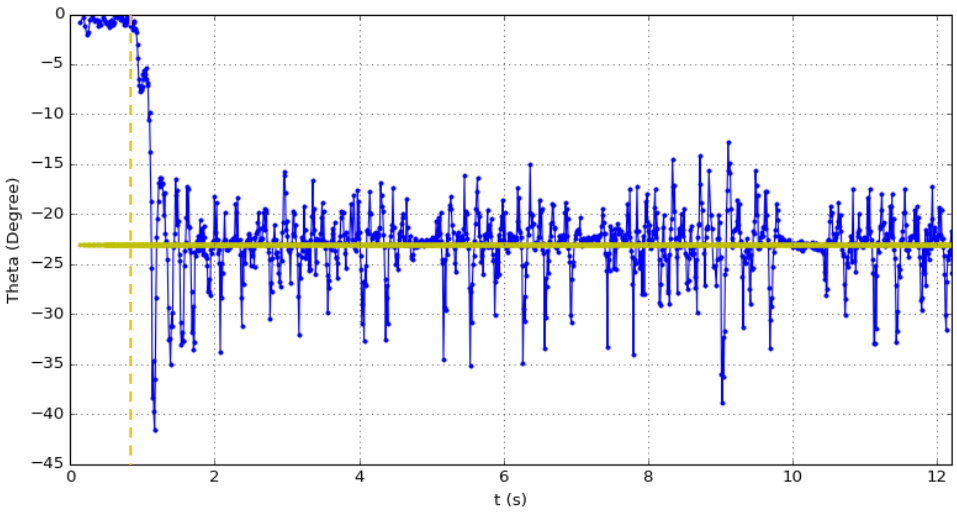

Figure 27. θ Response of balancing test 2

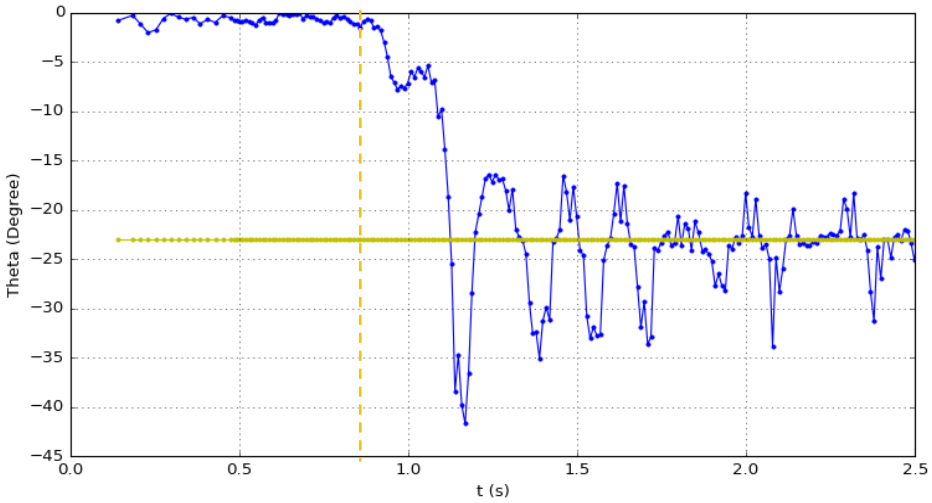

Figure 28. Close look of θ Response of balancing test 2

Figure 29 is the $\dot{\phi}$ response, which is the response of wheels angular velocity. According to our analysis, when the MIP tried to be upright from a certain angle, the wheels would rush to the same side of the direction of falling to reach a balance. In our case, $\dot{\phi}$ is 0 before 0.8 second as we haven't released it. The rush movement is obvious as a peak of wheels angular velocity appears since 0.8 second, which is way higher than normal stable velocity.

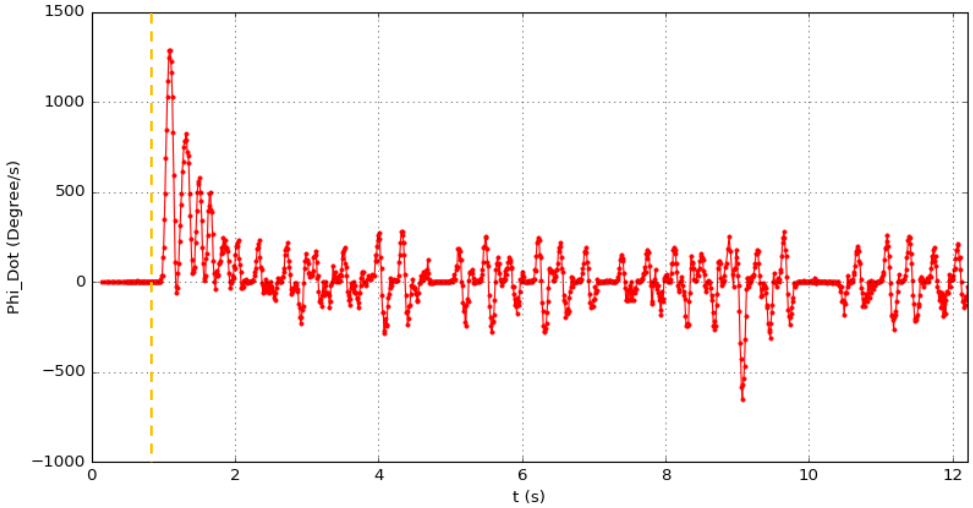

Figure 29. $\dot{\phi}$ Response of balancing test 2

Figure 30 is the voltage response. The peaks of voltage, which is generated by a request of peak $\dot{\phi}$ velocity, is about 4.6 V. In other words, we can even inclined the MIP for more than 23 degrees and finally balanced it up, because the maximum voltage we could apply is 7.4 V and yet not reached.

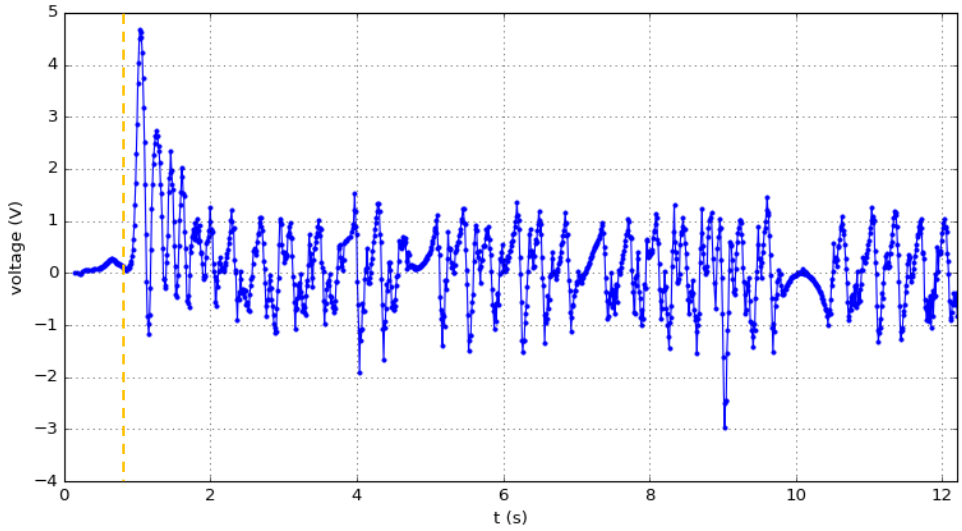

Figure 30. Voltage Response of balancing test 2

5.2.3 Moving Test

Moving test demonstrate the ability of the MIP to move forward and backward according to our command, so we can remote control the MIP rather than just balancing it. The idea of controlling moving is setting reference signal of $\dot{\phi}$ to a non-zero value, a positive value drives MIP moving forward and a negative value drives MIP moving backward. The absolute value of reference determines the moving speed.

We started the test with the controller on, after the MIP is stabilized, we send our moving command out in Notebook. Here are responses of some important parameters.

Figure 31 is the ϕ response, which is the wheel position response. There are three lines here in blue, green and red. They stand for the Phi information for left wheel, right wheel and the average of wheels, separately. It is reasonable that they assembled each other since they share the same movement when moving forward and backward. We can tell that MIP stabilized itself in the first 5 seconds. Then from 5s to 12s, our command was sent to the processor and MIP moved forward. Subsequently, it went backward from 16s to 21s.

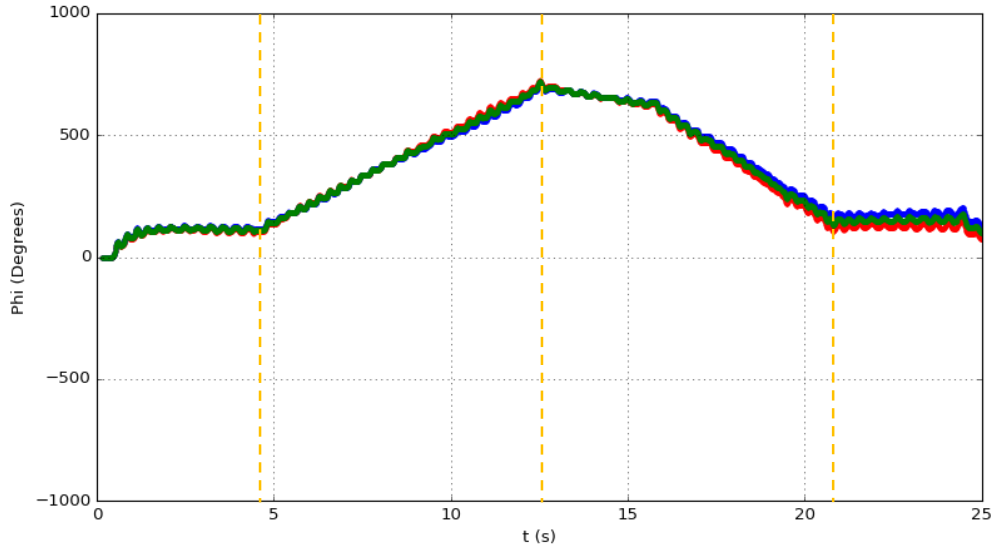

Figure 31. ϕ Response of moving test

Figure 32 is the $\dot{\phi}$ response, which is the response of wheels angular velocity. It is easy to observe that $\dot{\phi}$ is mostly positive from 5s to 12s, as it was moving forward. While mostly negative from 16s to 21s due to moving backward.

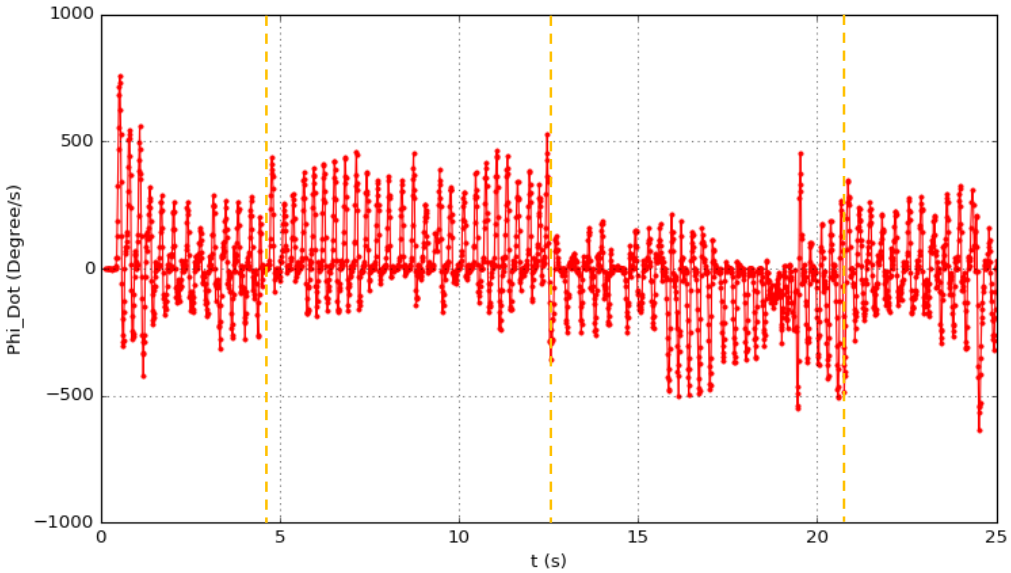

Figure 32. $\dot{\phi}$ Response of moving test

5.2.4 Steering Test

Steering test demonstrate the ability of the MIP to turn left and right in place according to our steering command. The idea of controlling steering is setting voltages of two motors into different values. The ratio of two voltages determines the steering speed.

We started the test with the controller on, after the MIP is stabilized, we send our steering command out in Notebook. Here are responses of some important parameters.

Figure 33 is the ϕ response, which is the wheel position response. Blue for left wheel and red for the right. We can tell that MIP stabilized itself in the first 6 seconds. Then from 6s to 18s, left wheel span forward and right wheel almost kept in place, which led to right turns in place. Then from 20s to 30s, MIP perform left turns and went back to initial states.

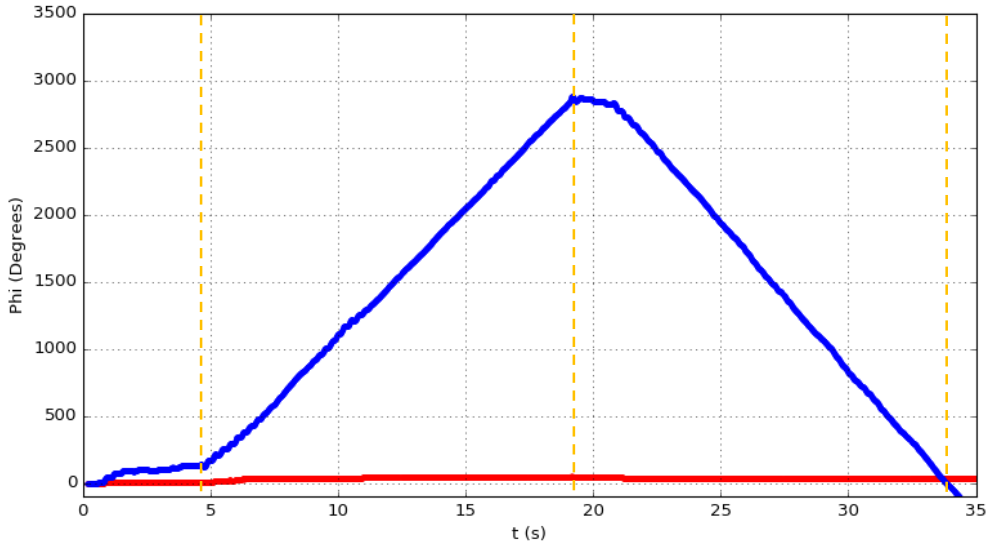

Figure 33. ϕ Response of steering test

Figure 34 is the $\dot{\phi}$ response of left wheel (blue line in Figure 33). Left wheel had positive speed from 6s to 18s, negative speed from 20s to 30s, while right wheel stay in place.

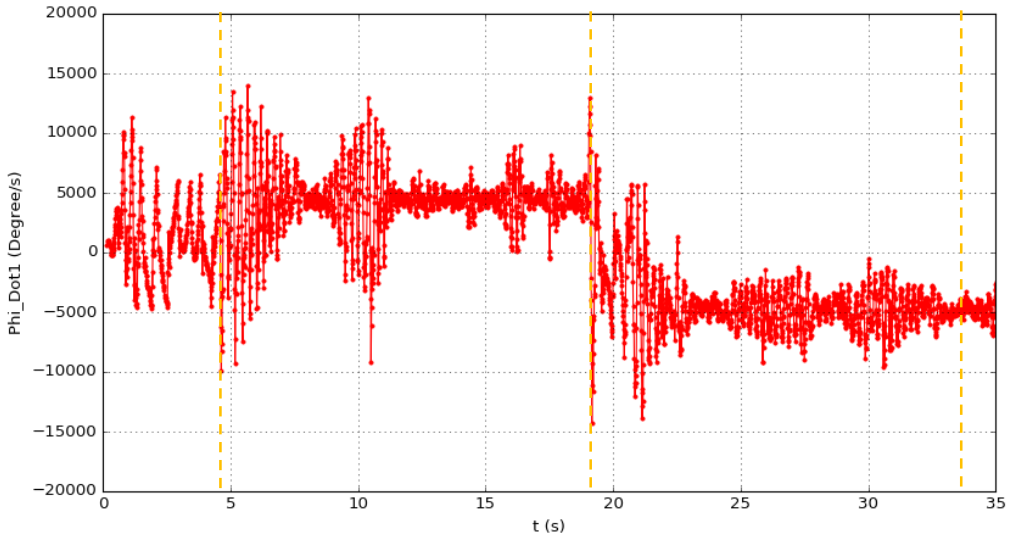

Figure 34. Left wheel $\dot{\phi}$ Response of steering test

Chapter 6
Conclusion

In this paper, we firstly extracted state-space model of an inverted pendulum system based on a Lagrangian method, two system identification experiments were followed to achieve the better accuracy of the model. Then we managed to design a second-order and stable LQG controller based on velocity control alone. After successfully implementing the controller on the robot EduMIP, it turned out that the closed-loop system is not only a stable robot that could balance itself, but also a stable robot that could move, steer and start with itself inclined.

Appendix A

All the data are given by James Strawson. Please refer to:

https://github.com/StrawsonDesign/EduMiP/blob/master/HW4%20Balancing%20EduMiP.pdf

Symbol	Parameter	Value
r	Radius of wheels	$34mm$
l	Body center of mass to wheel axis	$36mm$
m_W	Mass of a wheel	$27g$
m_B	Mass of the body	$263g$
G_r	Gearbox ratio	35.57
I_m	Motor armature inertia	$3.6\times10^{-8}kg\cdot m^2$
I_B	Body inertia	$4\times10^{-4}kg\cdot m^2$
$V_{\max}$	Nominal battery voltage	$7.4V$
$\overline{s}$	Motor stall torque	$0.003Nm\ at\ V_{\max}$
ω_f	Motor free run speed	$1760rad/s\ at\ V_{\max}$

References

[1] “Beagle bone robotics”. Retrieved from https://github.com/StrawsonDesign

[2] “Complementary filter - My IMU estimation experience”. Retrieved March 08, 2017, from https://sites.google.com/site/myimuestimationexperience/filters/complementary-filter

[3] A. N. K. Nasir, M. A. Ahmad, R. M. T. Raja Ismail. “The Control of a Highly Nonlinear Two-wheels Balancing Robot: A Comparative Assessment between LQR and PID-PID Control Schemes,” International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, Vol 4, No 10, 2010.

[4] “Using the Lagrangian to obtain Equations of Motion” Lecture Notes, Indiana University-Purdue University Indianapolis, 2008. Retrieved March 08, 2017, from http://www.engr.iupui.edu/~skoskie/ECE680/ECE680_l3notes.pdf

[5] “System identification”. Retrieved March 08, 2017, from https://www.mathworks.com/help/ident/gs/about-system-identification.html#bq47jv3-1

[6] M.C. de Oliveira. “Linear system design” Lecture Notes, University of California, San Diego, 2016.

[7] “Zero-order hold”. (2016, November 14). Retrieved March 08, 2017, from https://en.wikipedia.org/wiki/Zero-order_hold