BALANCING ROBOT CONTROL AND IMPLEMENTATION

A Thesis

by

HSIN-CHUAN SUNG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Gholamreza Langari
Committee Members,	Dezhen Song
	Sivakumar Rathinam
Head of Department,	Andreas A. Polycarpou

May 2015

Major Subject: Mechanical Engineering

Copyright 2015 Hsin-Chuan Sung

ABSTRACT

This thesis presents several control methods on one wheeled and two wheeled balancing robot and implementation of a two wheeled robot. The purpose is to balance the robot at a standstill position and let it move smoothly. The first step is to model the robot by Lagrangian method. After getting the equation of motion of the robot, some linear and nonlinear controllers-PID, LQR, and Sliding Mode etc. are tested on the model in Matlab. Second, build a balancing robot by LEGO-Mindstorm EV3 package then test the simulation result on the LEGO-Mindstorm EV3 robot to see how the simulation results work and perform. The simulation results show that PID controller controls only the body's angle but no wheel's position. LQR controller controls all the states which are body's angle and wheel's position. Sliding Mode controller also can coverage both body's angel and wheel's position at the same time. By adding a PID controller on a LQR controlled two wheeled robot, the robot can move to the front and back by control the state which is the wheel's speed. The LQR and PID controllers are tested on the LEGO-Mindstorm EV3 robot. The experiments show the simulation is valid and physically achievable. The robot can be used in many situations to help people reach stuffs and pass lanes that human can't. Robots can overcome many human's limitation. The more researches goes on the balancing robot, more the robot can help people in different ways.

ACKNOWLEDGEMENTS

For me to finish this thesis, first, thanks my professor Gholamreza Langari, committee members Dr. Dezhen Song, and Dr. Sivakumar Rathinam for the advices of the professional skills and their time. Secondly, thank my parents Yueh-Ying Gou, Wen-Chi Soong, Hsin-Wei Song, and my girlfriend Hui-Tzi Chu for giving me the full support physically and mentally. Third, Thanks to the Taiwanese Students Association for taking care of me since the first day I came to College Station. Thank my classmates and lab mates for all these days' works and discussions. They also helps me a lot with my research. Fourth, thanks all my friends and all other people not be mentioned. Last but not least, thanks God for leading me here to such a great school and country. I learn so much in this two years. Everyone in my life helps me and teaches me something in some ways that is hard to describe and mention. Without all these people, I couldn't finish this thesis and my degree. Thank you all.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

In modern society, people use robots in their life more and more. Robots are used in many ways, like improve people's living experience or help people solve problems. Robots can be designed for specific purpose, so they solve problems that people want them to solve. Robots are designed by many people and built by all kinds of materials, so they can overcome some of human's limitations like size, speed, sense ability or processing ability. Robots formations and designs are changing with the history. They interact with people more and more and assist people more and more. Robots are so important that they help the society so much and so wide that everyone need and be helped by some kind of robots in their daily life. Without robots, a lot of great achievements couldn't be done so soon and neat.

There are countless categories and styles of robots being designed and used every day. They all have their importance and purposes. One of the most important purpose to use robots is in rescue missions. During rescue missions, the environment could be very danger and difficult for everyone. It could be too hard or impossible for people to help others under extremely hard conditions. The efficiency and the success percentage could be low. Worst of all, people may get hurt or even die. Therefore, to prevent this kind of tragedy happen and increase the chances saving people, robots are designed and built to help people.

In some situation, rescuer need to pass a narrow bridge to help people or give supplies but the bridge may be too narrow or too danger for people to pass. In this kind of situation, a small balancing robot could help.

In this thesis, the balancing robot that can be used in a lot of rescuing situations is studied. There are hundreds of thousands of robots being used in rescue missions every day and all the time. They all have their special functions and missions. A balancing robot means a robot with one or two wheels contact to the ground and it needs some kind of balancing method to control the body's angle prevent itself from failing in pitch or roll angle. While the body's angle is controlled and stand upright, the wheels' position also should be controlled in certain situation. Controlling wheels' angle means to hold the wheel at the position or angle which we want.

A balancing robot needs to balance its body angle to stay up right. At the same time, the wheels' position should remain at the same spot. Furthermore, the balancing robot should be able to move to the front, back, right, and left. After all this movements have been done. The balancing robot leans to the front and the wheel stay in a range is required in this thesis.

In this thesis, a balancing robot with Linear Quadratic Regulator (LQR), Proportional-Integral-Derivative (PID) controller is used. The controllers are stabling the system and ask the system to go to some specific position or angle. The works are done in simulation and hardware.

CHAPTER II

LITERATURE REVIEW

Balancing robot has become an interesting topic and widely studied in the past decade. A balancing robot can be seen as an enhancement of an inverted pendulum. Many researches have been done on it. A balancing robot is a robot with body and wheels. The wheels are attached to the body and contact to the ground. The Body does not contact to the ground. The robot needs to balance its body angle to stay up right to avoid failing. At the same time, the wheels' position should remain at the same spot. Furthermore, the balancing robot should be able to move to the front, back, right, and left. After all this movements have been done. The balancing robot leans to the front and the wheel stay in a range is required in this thesis.

In this thesis, a balancing robot with Linear Quadratic Regulator (LQR), Proportional-Integral-Derivative (PID) controller is mainly used. The controllers are stabling the system and ask the system to go to some specific position or angle. The works are done in simulation and hardware.

A two wheel inverted pendulum robot has high nonlinear dynamics behavior [1] so it a good model to test the controllers to see whether they work or not. It has three degrees of freedom for the whole system but only the wheels are controlled by motors then control the whole system [1]. Because of the ability to turn at a point or on a spot and lean down to a specific angle, the robot can enter places that other kinds of robots can't.

To design and control a two wheel inverted pendulum robot, since knowing and testing the robot before actually building it is important and necessary, the simulation model of it is needed to be done at the very beginning. Then, test all the control theories and controllers on the simulation math model. Therefore, modeling is the first step to build a two wheel inverted pendulum robot. Getting the math model, the control strategy can be used and try to make sure it will work before applying on the robot. Also, the behavior of the robot can be seen and understand prior to the hardware design and experiment. In this thesis, all the movement are on two dimensional plan, so the robot can be modeled as an inverted pendulum with two wheels. Instead of a cart base, the wheels will be the base and the input for torque applying.

To get the math model of the two wheel inverted pendulum robot and knowing its behavior, the equation of motion must be obtained. The equation of motion is the equations which can represent the motion of the thing or the robot that being designed. The behavior of the robot at specific timing or places can be calculated then get the results. To obtain the equation of motion, there are methods to be used. At the beginning, an inverted pendulum on a cart model is the basic model [2].

The equation of motion for the inverted pendulum on a cart is shown below:

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

$$I\ddot{\theta} + ml^2\ddot{\theta} + mglsin\theta = -ml\ddot{x}cos\theta \quad (2)$$

After linearization, the equation is shown in linear form:

$$(I + ml^2)\ddot{\varphi} - mgl\varphi = ml\ddot{x} \quad (3)$$

$$(M + m)\ddot{x} + b\ddot{x} - ml\ddot{\varphi} = u \quad (4)$$

In [3], it provides the detail about how to design and model an inverted pendulum on a cart. It is in fact a device which is unstable without control. The pendulum would fall over if the cart does not have any input and control the system. The equation of the system is provided as following:

$$(M+m)\ddot{x}+b\dot{x}+ml\ddot{\theta}cos\theta-ml\dot{\theta}^{2}sin\theta=F \qquad (5)$$

$$(I+ml^{2})\ddot{\theta}-mglsin\theta=-ml\ddot{x}cos\theta \qquad (6)$$

After the equation of motion of an inverted pendulum on a cart is provided, a design of inverted pendulum like two wheel robot is going to be modeled. Take the pendulum as the body of the robot and switch the cart to the wheel.

After getting the right and suitable equation of motion, the controller will be tested on the robot model. The first controller to use is the Proportional-Integral-Derivative (PID) controller. PID controller is a controller that takes the value of desire state value minus the actual state value as the error value, then times a constant plus derivate of the error value, plus the integral of the error value as the input torque.

The PID controller attempts to minimize the error by adjusting the process through use of a manipulated variable. In the absence of knowledge of the underlying process, a PID controller has historically been considered to be the best controller [4] [5]. On the other hand, the PID controller doesn’t guarantee the optimal performance and the stability of the system. The PID control theory was first published by a Russian American engineer Nicolas Minorsky who designed the navigation system for the US navy. By observing the action of a helmsmen, he found out that the ship was controlled not only by the current error, but also on the past error and the rate of current error. The final form of the PID control algorithm is shown below [4]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \tag{7}$$

K_p is the proportional gain, K_i is the integral gain, and K_d is derivative gain. $e(t)$ is the error of the state which use the desire state value minus the actual state value.

The next step, applying PID controller on the robot model. Since PID controller is a controller for single-input-single-output (SISO) system, it can control only one state at once. Therefore, in this case, the main purpose for the inverted pendulum robot is to stay upright, the body's angle is the first and main state which need to be controlled. The PID controller takes the error value of body's angle (pitch angle) from stable position then input a torque value from PID controller to the motor which is directly linked with the wheel. At the meanwhile, since one PID controller controls only one state, the wheels' position is uncontrollable at this time. The wheels' behavior is kind of random and just moved to balance the pitch angle. As the PID controller input the value to the wheel and control it goes back and forth, the pitch angle will be controlled and stay balance.

The PID controller which is designed for the inverted pendulum system is introduced below. In this design process, the system is assumed to be a single-input–single-output system. In this case, the pendulum's angle is attempted to be controlled without regarding for the cart's position. The plant for the pendulum is shown below.

$$P_{pend}(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}} \left[\frac{rad}{N}\right] \tag{8}$$

Where

$$q = (M + m)(I + ml^2) - (ml)^2 \tag{9}$$

Through the Matlab, the PID controller is tested on the model and the results are shown below:

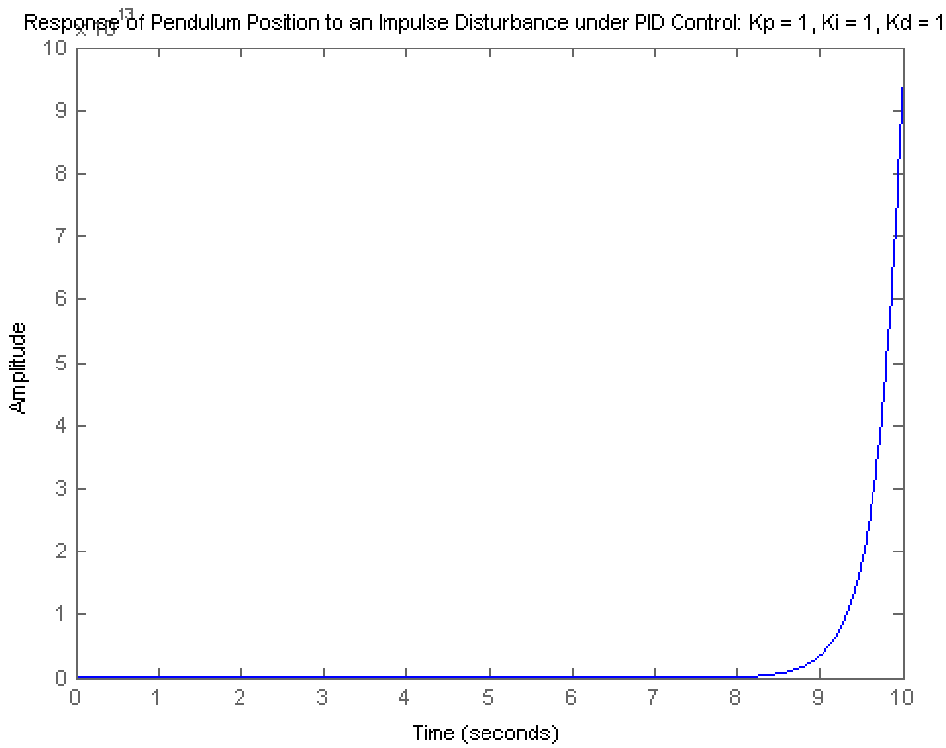

Figure1. Response of pendulum position to an impulse disturbance under PID control value 1, 1, and 1

Figure 1 above shows the impulse response under PID control value: kp=1, ki=1, and kd=1. It shows that the controller doesn't work for the system. With these values. The system is unstable and the inverted pendulum would fail.

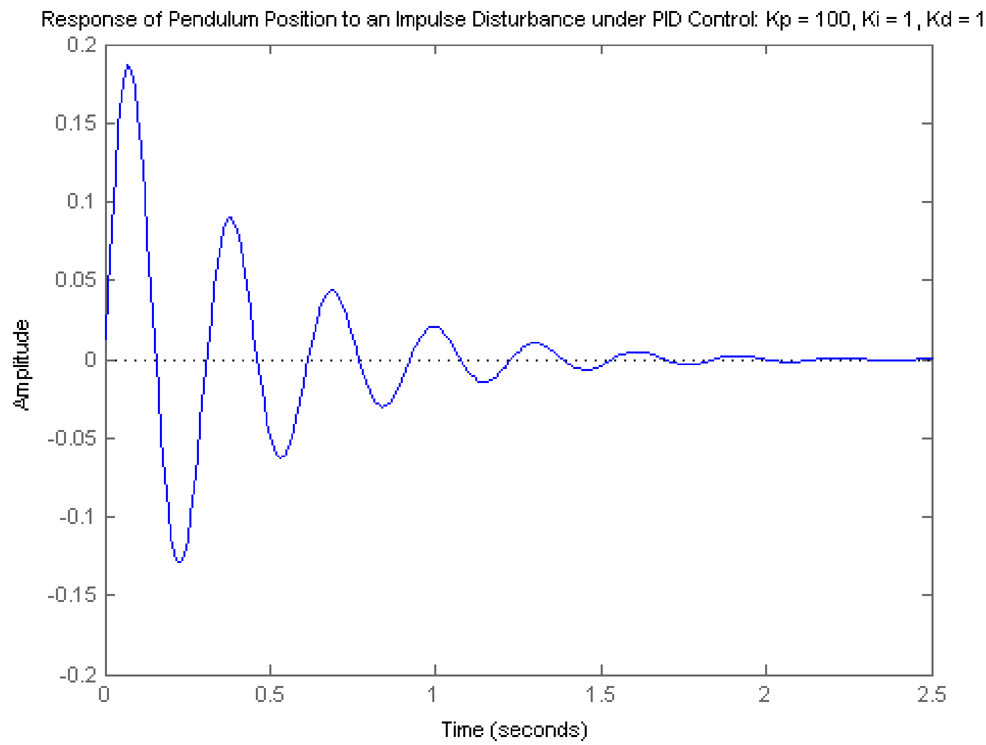

Figure 2. Response of pendulum position to an impose disturbance under PID control value

Figure 2 above shows the impulse response of the inverted pendulum under PID control value: kp=100, ki=1, and kd=1. The inverted pendulum is stabilized in a short time. The performance is good.

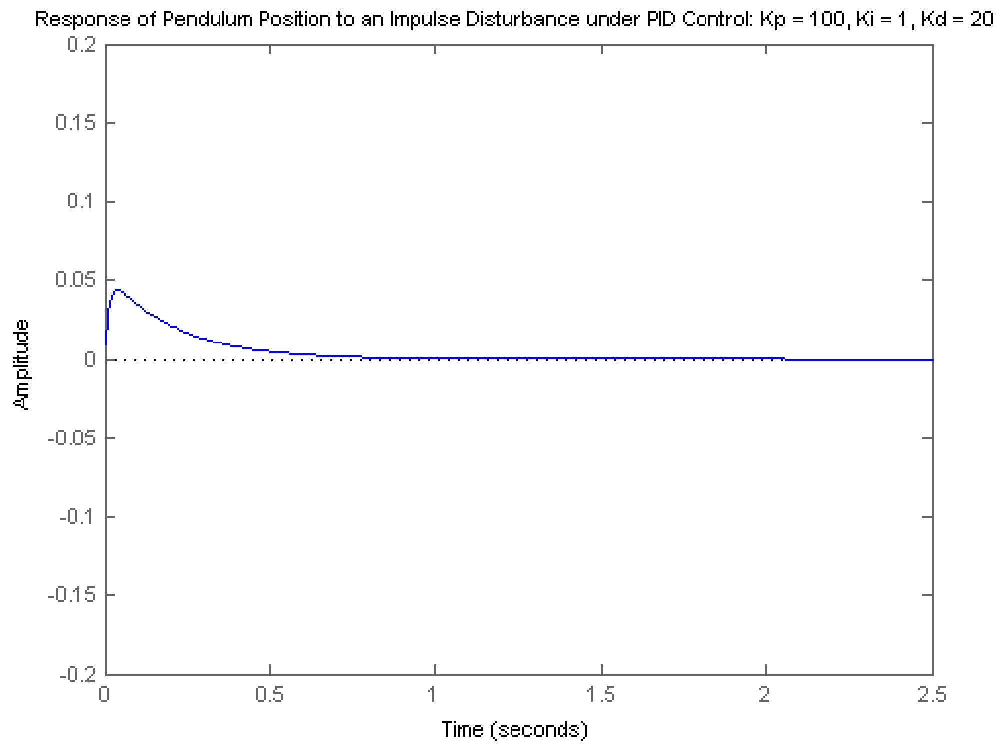

Figure 3. Response of pendulum position to an impulse disturbance under PID control value 100, 1, and 20

Figure 3 above shows the impulse response of the inverted pendulum under the PID control value: kp=100, ki=1, kd=20. The vibration is decreased, and the peak is also decreased. Overall, the performance is better than the previous set of PID value.

However, with only one PID controller to control the inverted pendulum, the cart's position is not controlled. Figure 4 below shows the behavior of the cart when the inverted pendulum is controlled by the PID controller. [6]

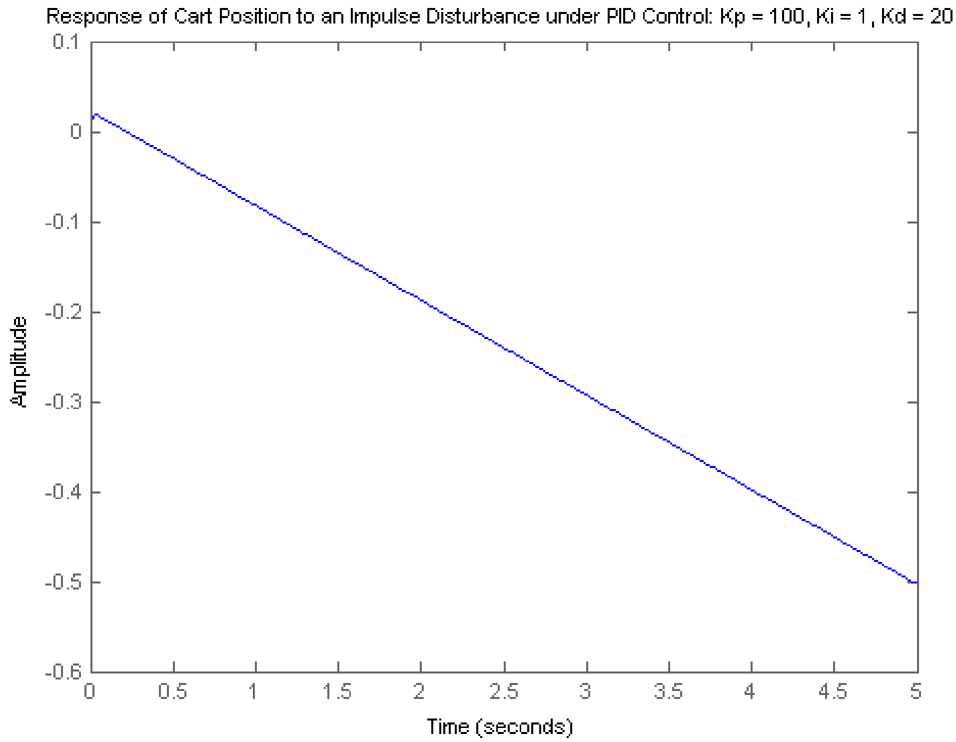

Figure 4. Response of cart position to an impulse disturbance under PID control value

Since there is no control for the cart, the cart would just keep moving to the same direction to the infinite. This result is not valid since it won't work physically. The implementation of this design will not work.

LQR controller is a control method that can control multiple state at the same time. Following the method to find the correct series of gain value the times them with the right state. After that, add them together. Through the LQR controller, the system can be stabilized.

By using the one PID controller, the pitch angle can be stabilized. However, a good control performance for the two wheel balancing robot should control both body's angle (pitch angle) and wheel's position. A good controller should be able to balance the inverted pendulum and control the wheels' position at the same time. The LQR controller become a good choice. Due to LQR controller's multi-input-multi-output (MIMO) system control ability, it can control multiple states at the same time. By a certain series of gain values, feedback the desired states and times the gain values each by each then add them together to become the torque value which input to the wheels. By finding a correct and suitable series of gain value, the wheels and the body can be controlled at the same time. The LQR controller will take them to the stable position and state.

For a linear time continuous (LTI) system, the system is represented as:

$$\dot{x} = Ax + Bu \tag{10}$$

The quadratic cost function is defined as:

$$J = \frac{1}{2}x^T(t_1)F(t_1)x(t_1) + \int_{t_0}^{t_1}(x^TQx + u^TRu)dt \tag{11}$$

Then, the feedback control law which can minimizes the value of the equation above is:

$$u = -Kx \tag{12}$$

Where $K = R^{-1}B^TP(T)$ and P can be found through solving the Riccati equation:

$$A^TP(t) + P(t)A - P(t)BR^{-1}B^TP(t) + Q = -\dot{P}(t) \tag{13}$$

The boundary condition is $P(t_1) = F(t_1)$[7]

The inverted pendulum system is modeled and LQR controller is tested on the model. By designing the feedback gains for the states, the system can be stabilized. In different initial conditions, say, different start angle for the inverted pendulum, the system has different behavior. Due to the limitation of the using of the LQR controller, the system must be presented in state space form because the LQR controller is a modern control theory controller. Hence, the objective of the LQR controller must be modeled in state space form. Then, give the matrices of the state space A and B and the Q and R matrices for LQR controller finding the P matrix from Riccati equation. The Q and R matrices are chosen as below:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{14}$$

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ R = 1 \tag{15}$$

From the Q matrix, the performance of body's angle and wheels' position is the important part so the value of Q_{11} and Q_{22} is larger than Q_{33} and Q_{44}. After that, solve the equation below to find the P matrix.

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{16}$$

Then, plug in the P matrix to the equation below to get the best gain value K.

$$K = T^{-1}(T^T)^{-1}B^T P = R^{-1}B^T P \tag{17}$$

The K matrix is: $K = [\,-31.6228 \quad -39.2019 \quad -170.4550 \quad -41.8113\,]$

Comparing the performance of PID and LQR controller, the LQR controller gives a better result for the two wheel inverted pendulum robot. The LQR controller controls both body's angle and wheel's position. The LQR controller makes both of them stable. The settling time and the peak are also better than the PID controller. Therefore, the LQR controller definitely a good choice for controlling the robot.

However, there are still some points and problems for the LQR controller to improve and fix. While stabilizing the body's angle and the wheels' position, the body's angle can be stabilized without steady state error but the wheels may have steady state error. In other words, the body can stay upright all the time but the wheels are staying at a small distance away from the initial position.

To fix this problem, adding a PID controller to the system becomes a solution that is worthy to try. By adding a PID controller to a LQR stabilized system, take the wheels' position as the state for PID controller to control, the steady state error of the wheels can be reduced. It means that the wheels can stay more close to the initial position while the body is stabilized. [8]

To decrease the steady state error, adding a PID controller to the LQR controller is the choice. Through this way, the controllers can eliminate steady state error and improve the response of the system at the same time. Due to the errors occur on the wheel's position and other wheel's performance, the PID controller is added to control the states relative to the wheel. In [8], the PID controller is added to control the wheel's position. Through the PID controller, the steady state error could be eliminate and the peak and the settling time can also be decreased. Overall, the response is better. [8]

CHAPTER III

BALANCING ROBOT MODEL AND CONTROLLERS DESIGN

In this thesis, the inverted pendulum two wheel balancing robot model is designed. The math dynamic model of the robot and the controllers' math model are designed. Since all the movements are designed in two dimensional, which means the yaw angle and the roll angle are not used in this model and thesis. The equation of motion is in two dimensional, too. The angle mainly considered is the pitch angle. The idea picture of the model is shown below. Figure 5 and figure 6 show the side view of the two wheel robot. There are two parts of the robot. The first part is the body, which has the center of mass at the center of the body. The second part of the robot is the wheel which also has the center of mass at the center of the wheel. Since the robot moves only in two dimensional and the pitch angle is the only considered angle, there is only one wheel can be seen in the figure shown below.

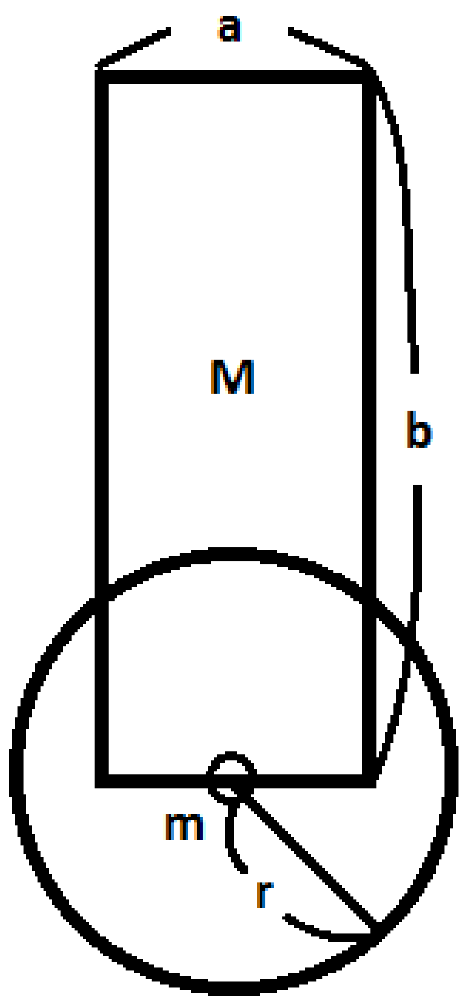

Figure 5. Simple idea side view figure of a two wheel robot

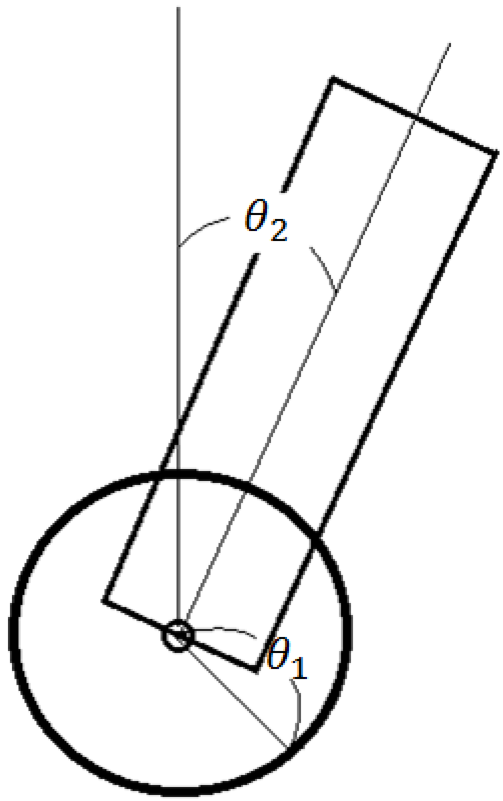

Figure 6. Side view of the two wheel robot with defined angles

Use the wheel as input and the inverted pendulum is replaced by the robot's body. The body is free to lean to the front and back if there's no extra force on the body. The wheel can rotate clockwise and counter clockwise. The wheel rotates without slipping relative to the ground. The body can be effected by the wheels' movement if the wheel goes to the front or back. The motion of the system is presented by the equation of motion below. The method for finding the equation of motion of the system is shown below:

The kinetic energy for the system is shown below:

$$T = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}I_1\dot{\theta_1}^2 + \frac{1}{2}M\left(\dot{x_g}^2 + \dot{y_g}^2\right) + \frac{1}{2}I_2\dot{\theta_2}^2 \quad (18)$$

Note that $x = r\theta_1$, $\dot{x} = r\dot{\theta_1}$ and $x_g = x + lsin\theta_2$, $y_g = lcos\theta_2$ then, take the derivative and square of x_g and y_g, the new kinetic energy can be found as below:

$$T = \frac{1}{2}mr^2\dot{\theta}^2 + \frac{1}{2}I_1\dot{\theta_1}^2 + \frac{1}{2}M\left(r^2\dot{\theta_1}^2 + 2r\dot{\theta_1}lcos\theta_2\dot{\theta_2} + l^2\dot{\theta_2}^2\right) + \frac{1}{2}I_2\dot{\theta_2}^2 \quad (19)$$

The next step is finding the potential energy. The potential energy is shown below:

$$V = Mglcos\theta_2 \quad (20)$$

Assume L equal to T - V

$$L = T - V = \frac{1}{2}mr^2\dot{\theta}^2 + \frac{1}{2}I_1\dot{\theta_1}^2 + \frac{1}{2}M\left(r^2\dot{\theta_1}^2 + 2r\dot{\theta_1}lcos\theta_2\dot{\theta_2} + l^2\dot{\theta_2}^2\right) + \frac{1}{2}I_2\dot{\theta_2}^2 - Mglcos\theta_2 \quad (21)$$

Take the partial derivative of L relative to $\dot{\theta_1}$

$$\frac{\partial L}{\partial\dot{\theta_1}} = mr^2\dot{\theta_1}^2 + I_1\dot{\theta_1} + Mr^2\dot{\theta_1} + Mrlcos\theta_2\dot{\theta_2} \quad (22)$$

Take the time derivative of the equation above

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta_1}}\right) = mr^2\ddot{\theta_1} + I_1\ddot{\theta_1} + Mr^2\ddot{\theta_1} + Mrl(-sin\theta_2\dot{\theta_2}^2 - cos\theta_2\ddot{\theta_2}) \quad (23)$$

Since $\frac{\partial L}{\partial\theta_1} = 0$,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta_1}}\right) - \frac{\partial L}{\partial\theta_1} = (mr^2 + I_1 + Mr^2)\ddot{\theta_1} + Mrl(-sin\theta_2\dot{\theta_2}^2 - cos\theta_2\ddot{\theta_2}) \quad (24)$$

The second part of the equation of motion is shown below:

$$\frac{\partial L}{\partial\dot{\theta_2}} = Mr\dot{\theta_1}lcos\theta_2 + Ml^2\dot{\theta_2} + I_2\dot{\theta_2} \quad (25)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) = (Ml^2 + I_2)\ddot{\theta_2} + Mrl(-sin\theta_2 + cos\theta_2) \tag{26}$$

$$\frac{\partial L}{\partial \theta_2} = -Mr\dot{\theta}_1 lsin\theta_2 \dot{\theta_2}^2 + Mglsinsin\theta_2\dot{\theta}_2 \tag{27}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) - \frac{\partial L}{\partial \theta_2} = (Ml^2 + I_2)\ddot{\theta_2} + Mrl(-sin\theta_2 + cos\theta_2) + Mr\dot{\theta}_1 lsin\theta_2 \dot{\theta_2}^2 - Mglsin\theta_2\dot{\theta}_2 \tag{28}$$

The final equation of motion is given to present the model of inverted pendulum two wheel robot:

$$\begin{bmatrix} Mr^2 + mr^2 + I_1 & Mrlcos\theta_2 \\ Mrlcos\theta_2 & Ml^2 + I_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta_1} \\ \ddot{\theta_2} \end{bmatrix} = \begin{bmatrix} Mrlsin\theta_2 \dot{\theta_2}^2 \\ Mrlsin\theta_2 \dot{\theta}_2 \dot{\theta}_1 - Mrlsin\theta_2 \dot{\theta_2}^2 \dot{\theta}_1 + Mglsin\theta_2 \dot{\theta}_2 \end{bmatrix} \tag{29}$$

The equation of motion of the two wheel robot is shown. To test the certain controllers on the robot, the linear model is required due to the controllers' limitations. The original equation of motion is nonlinear, therefore, linearizing the model is the next step. After linearization, the controllers like PID or LQR can be used on it. Note that the controllers should also be tested on the nonlinear model after they are tested on the linear model.

To use the model on finding the value of LQR controller. The model must be linearized and putted into state space form since LQR controller is a linear controller. Therefore, the linearization must be done here so that the process can keep going.

To linearize the nonlinear model, assume the robot is at upright position. Note that the position is a singular solution for the system. The other angles around it are unstable positions. The assumption is that the angle of the body and the wheels are very small, so the angles are approximately zero. Therefore, $cos\theta \to 1$ and $sin\theta \to 0$. The two equations from the equation of motion is shown below:

$$(Mr^2 + mr^2 + I_1)\ddot{\theta}_1 + Mrl\ddot{\theta}_2 = u \tag{30}$$

$$(Ml^2 + I_2)\ddot{\theta}_2 + Mrl\ddot{\theta}_1 - Mgl\theta_2 = 0 \tag{31}$$

After getting the linearized equation of motion, the next step is to put the equation of motion into matrix form for the sake of easy calculation. The equation is shown below:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & & 0 & 1 & 0 \\ 0 & & 0 & 0 & 1 \\ & 0 & \frac{-MrlMgl\theta_2}{(Mr^2+mr^2+I_1)(Ml^2+I_2)+Mrl^2} & 0 & 0 \\ & 0 & \frac{(Mr^2+mr^2+I_1)(Mgl\theta_2)}{-(Mr^2+mr^2+I_1)(Ml^2+I_2)+(Mrl)^2} & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} +$$

$$\begin{bmatrix} 0 \\ 0 \\ \frac{1}{(Mr^2+mr^2+I_1)-\frac{(Mrl)^2}{Ml^2+I_2}} \\ \frac{1}{-(Mr^2+mr^2+I_1)(Ml^2+I_2)/_{Mrl}+Mrl} \end{bmatrix} \tag{32}$$

In this part, the PID controller is tested on the model provided above. There are two experiments for the PID controller on the model. The first task is to use one PID controller to control the body's angle and see if the PID controller can stabilize the body or not. At the same time, observe the behavior of the wheel, to see what will happen to the wheel. The second experiment for the PID controller is to use two PID controllers on the system. The purpose for the two PID controllers is trying to balance the body and control the wheel's position at the same time. Each controller would control a state, one controller control wheels' position, the other controls body's angle. The two-PID controller are expected to fully control two states without any effects from other states.

The first part of the PID controller experiment is placing one PID controller on the body and trying to balance the body's angle. Due to there's no control for the wheels' position, the wheel can have different uncontrolled behaviors that is needed to be test. The block diagram of the one PID controller on the robot's body is shown as figure 7 below:

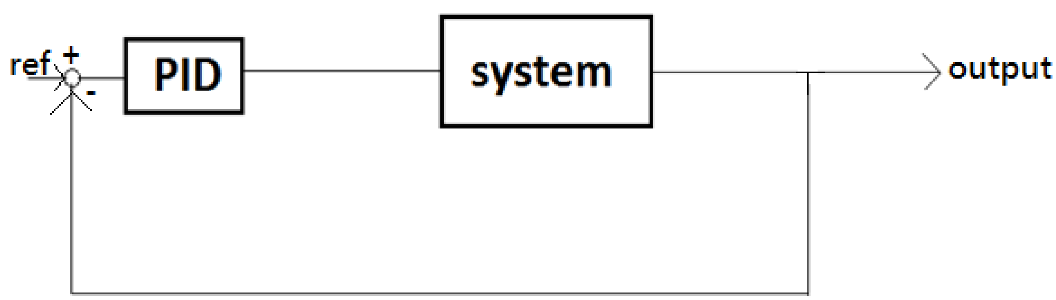

Figure 7. Block diagram of PID control with the system

The values of the gains for the PID controller is set as $k_p = -38, k_i = -60, k_d = 3$ at the beginning then use trial and error method to tune the values. The result is shown in the result chapter.

Figure 8 below shows the dual PID controller on the robot. Each controller controls a state.

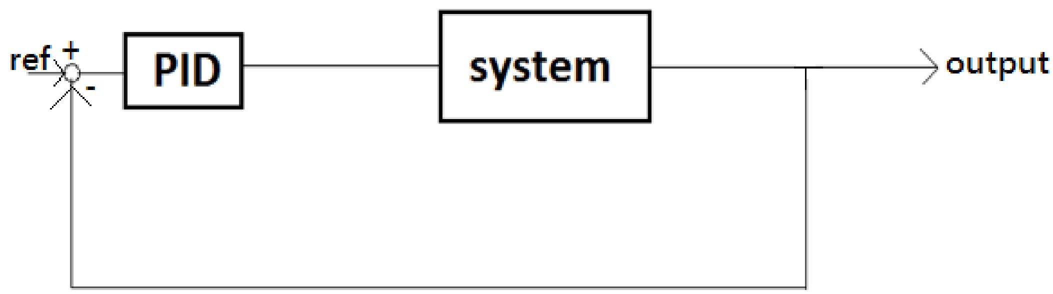

Figure 8. Block diagram of two PID control with the system

In this part, the LQR controller is tested on the robot. From the equation of motion which is provided above, take wheels' position, angular velocity, body's angle, and body's angular velocity as system's states. Take the quadratic cost function:

$$J = \frac{1}{2}x^T(t_1)F(t_1)x(t_1) + \int_{t_0}^{t_1}(x^T Q x + u^T R u)dt \quad (33)$$

Deciding the Q and R matrices to find the suitable gain matrix K which can stabilize all the states. The LQR controller should can stabilize all the states at the same time. The K matrix is depended on the Q and R matrices, so the way to define Q and R matrices is important. In this thesis, the way to decide Q and R matrices is trial and error method. Observe the performance of the robot and the physical limitation of the hardware then change the Q and R matrices to fit the simulation and implementation requirement. The block diagram of the LQR controller on the robot is shown as figure 9 below:

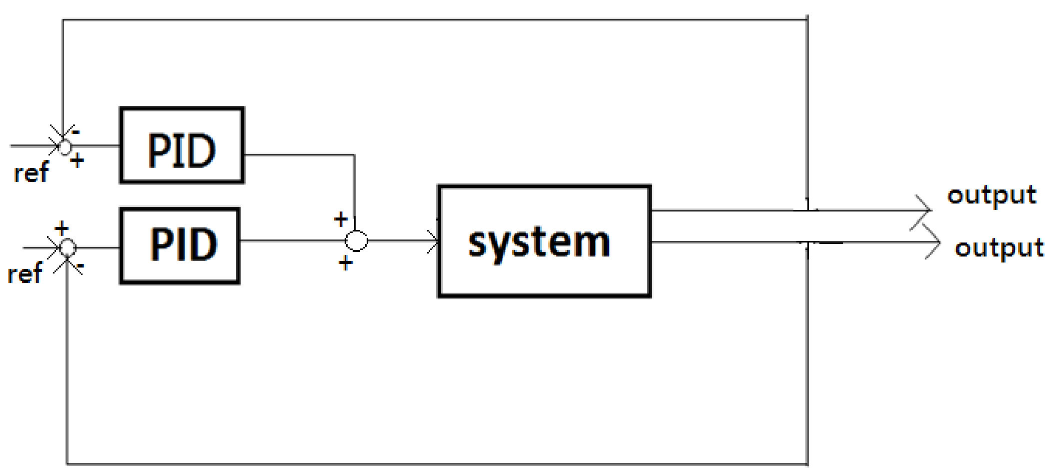

Figure 9. Block diagram of LQR control with the system

The matrices is set as shown below:

$$Q = \begin{bmatrix} 30 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, and\ R = [100] \tag{34}$$

The gain matrix is obtained and shown below: $K =$

$[-0.5477 \quad -0.6275 \quad -36.8896 \quad -7.7534]$

In this part, the LQR plus PID controller is tested. From the previous tests, the PID controller controls the body's angle and the body's angle only. LQR controller controls all the states. However, the performance of the LQR controller can be improved by adding a PID controller for a certain state. Therefore, in this thesis, the PID controller is set for the wheel or body depends on the situation. For different purpose, the PID controller can control the wheels' angle or angular velocity. Through this way, the robot can have different behavior and performance. The block diagram of the LQR controller plus PID controller is shown as figure 10 below:

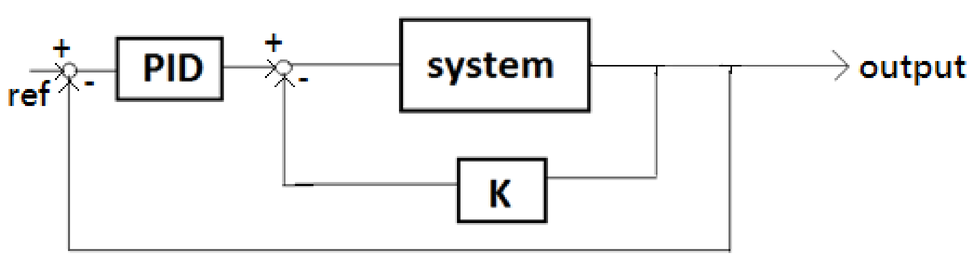

Figure 10. Block diagram of LQR plus PID control with the system

The LQR controller maintain the system around stable position which is upright position. Under this condition, the PID controller can be added and controls certain state to let the robot have different behavior. The value of the LQR controller and PID controller is important since the LQR controller tent to maintain all the states' value at zero but the PID controller tent to bring certain state to different value. The different value of these two controllers can be seen as giving different weight. A good set of value can let the robot move just right.

CHAPTER IV

BALANCING ROBOT MOVEMENT DESIGN

In this chapter, the object is designing the two wheel balancing robot's movement. By using the PID and LQR controller, the behavior of the robot can be seen and designed. The LQR controller can stabilize the whole system and the PID controller can controller certain state. What's more, the LQR controller plus PID controller can control the wheels' speed by PID controller and the body staying upright under the condition of LQR stabilize the system. By using these properties, the robot can have different kind of movement and performance. Interact between these control theories and control strategies helps the robot under different situation. Giving different order to the robot on different timing also helps the robot act different ways.

In this part, the robot's body is required lean to a specific angle and the wheels should rotate at the same time. Due to the physical limitation and the coupling property. The robot's body angle and the wheels' rotation are related to each other. It needs to be tested while simulation and implementation.

By using the LQR controller, the robot can stay at a spot which means the wheel would stay in a small range and the body can stay upright which means the body would come back and forth a little but mainly around the upright position. At the same time, a PID controller is added to the system to control the wheel's speed. The block diagram is shown as figure 11 below:

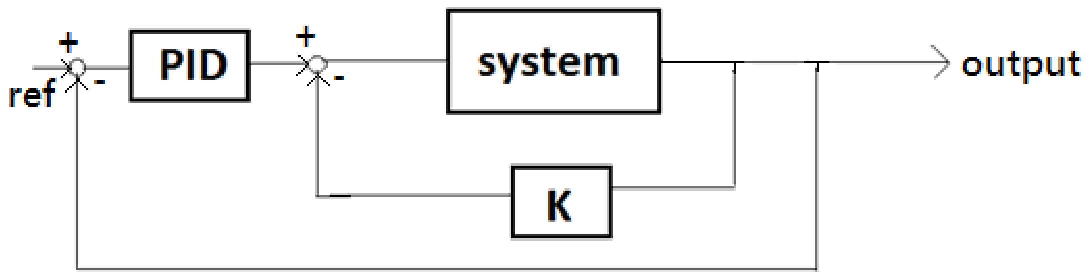

Figure 11. Block diagram of PID plus LQR control with the system

Through this way, the robot can stay balanced while the wheels are rotating at a specific speed. The robot's body would be balanced through the LQR controller, and the PID controller would control the wheel to the required speed.

The object of this part is to lean the body to a specific angle and trying to make the wheel stay at a spot at the same time. Defining the purpose, the robot's body should lean to a specific angle and comes back and forth just like vibrating. At the same time, the wheel should come back and forth around the initial position. To approach this objective, the LQR controller and PID controller is going to be used. Since any position away from the upright position is an unstable position for the body, it becomes very difficult to use one controller or one control strategy to reach the requirement. For now, the behavior of the LQR and PID controller on the robot is known. By switching the controller in different situations, the robot is hoped to act like what the requirement is.

In this part, the brake ability is going to be discussed and tested. The initial condition will be set as the two wheel inverted pendulum robot moving with a specific leaning angle. The brake will be applied in different strength and see the difference that robot act.

The robot set to lean ten degrees to the front (pitch angle) while moving. The brake will be applied under different speed condition. By tuning the controller, the strength of the brake can be controlled. The reacting angle of the robot and the distance needed can be found. The relationship between speed, brake distance, and robot maximum leaning degree can be understood.

By understanding the relationship between robot's speed, brake distance, and robot's maximum leaning degree, the data of those numbers can be used to the robot while it is on the road or go out for missions. While the robot is driving on the road, it comes to a red light, the robot can use those data to measure how much distance does it has for stopping in front of the red light and what is the degree it will lean to. The minimum degree is always a better option. If there is a man driving the robot, smaller the leaning degree it is, the comfort the man would feel. Combining these information, the way to control or drive the robot can be known.

In this part, the speed control is applied. The block diagram of the system is shown in figure11. The PID controller is going to control the wheel's speed. By using the LQR and PID controller, the optimal movement can be designed. The LQR controller is going to let the system has the optimal performance to reach the required speed.

The last part of the movement design is to combine the movements which is designed previously and have an optimal design of the velocity and trajectory of the two wheel robot. The goal of this optimal design is to let the robot go from point A to point B in the shortest time by controlling its speed. There are problems needed to be solved for the robot to acquire the optimization. First of all, a two wheel robot can't act like a four wheel car or the machines that doesn't require balancing. The robot can move only under the condition of the body's angle is balanced. Secondly, while moving the robot can't not simply accelerate with its maximum acceleration, reach the maximum velocity, then decelerate with its maximum deceleration to stop. There will be problems happened during every part of the movement.

First of all, before speeding up, the robot has to lean to the front with a specific angle. Then, if the robot is speeding up with its maximum acceleration, when the robot reaches its maximum velocity, the robot has to remain the velocity, since the speed becomes a constant, the robot's body angle has to stay zero which is vertical to the ground. However, to pull back the body's angle from a front leaning angle, the robot has to speed up but the robot is already at its maximum velocity and acceleration, so there is no room for the robot to do so. The robot will fall by this way to move. Secondly, if the robot is moving with its maximum speed constantly and the body's angle is zero, the same problem will happen while slowing down. To slow down the robot, the robot has to lean to the back to stay balancing while decelerating. Without leaning back the robot will fall to the front while slowing down. To let the robot to lean to the back, the robot needs to speed up. However, at the moment, the robot is already at its maximum moving velocity, it cannot speed up anymore. There is no way for the robot to stop without falling.

A best way to control the robot from point A to point B is shown below. First, the time of the maximum acceleration which is needed to bring the body's angle back to vertical from front leaning and to bring the body from vertical to back leaning is preserved. The maximum speed that the robot can reach is the actual maximum speed minus the preserved time times the maximum acceleration. From here, the maximum velocity that the robot can do can be defined. Therefore, the robot should go like this: use an acceleration which is smaller than the maximum acceleration to reach the defined maximum velocity, before reaching the defined max velocity, accelerate with the max acceleration to bring the body's angle back to vertical and reach the defined max velocity then stay with the velocity. To slow down the robot, the robot has to accelerate again to bring the body leaning to the back and reach the actual maximum velocity then start to slow down with the maximum deceleration. While almost getting to the destination, the braking control is applied to slow down the robot and let the robot to stop at the destination and stay vertical. That is the whole optimal velocity control of the robot. The idea figure is shown below:

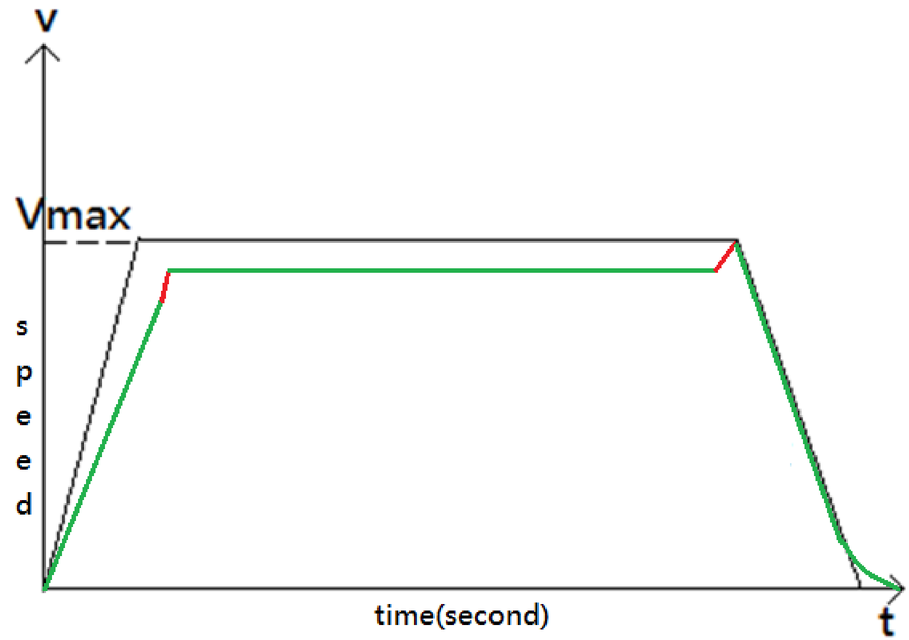

Figure 12. Ideal velocity control of the robot

Figure 12 above shows the ideal velocity control of the robot. In realistic, the figure should look like the figure 13 below. The curve is smoother in every turning point. Also, at the beginning, to let the robot lean to the front, the wheel has to roll to the back a little bit.

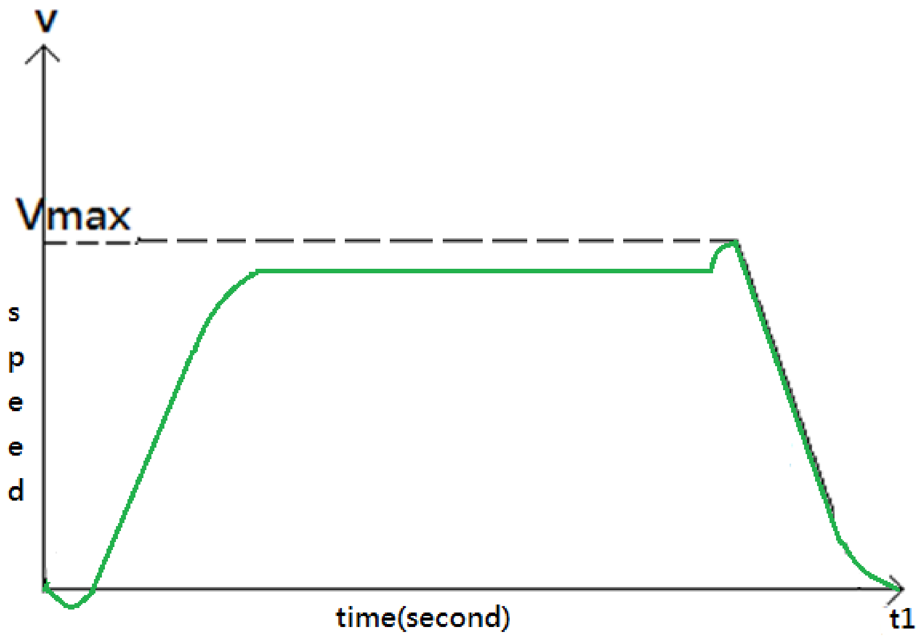

Figure 13. Actual velocity control of the robot

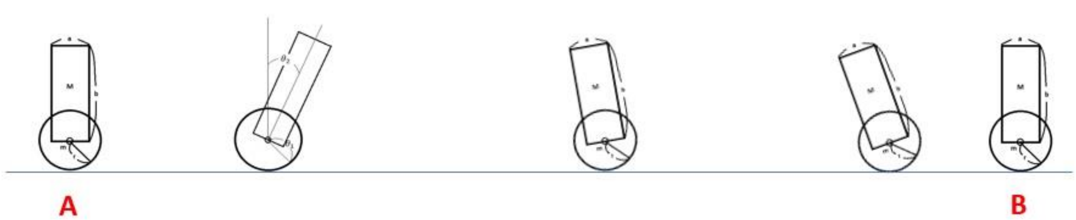

Figure 14. Idea of the robot goes from A to B

Figure 14 above shows the idea of how the robot would act when it goes from point A to point B. From here, the optimization velocity controlling is shown. One of the interesting part that is needed to be discussed is the beginning of the movement before reaching the defined max velocity. To optimize the way to reach the defined velocity is discussed below.

To have the shortest time to reach the defined velocity, the line is seen as an equation. Since the line has to curves, the equation of the line would be a cubic equation.

$$y = ax^3 + bc^2 + cx + d \tag{35}$$

By the property of the cubic equation, the lower bound and the upper bound can be presented as

$$y = \frac{2b^3 - 9abc \pm (2b^2 - 6ac)\sqrt{b^2 - 3ac}}{27a^2} \tag{36}$$

The time of those point can be presented as

$$x = \frac{-b \pm \sqrt{b^2 - 3ac}}{3a} \tag{37}$$

The maximum acceleration is the turning point of the cubic equation, so it can be presented as

$$\frac{d^2y}{dx^2} = 0 \rightarrow x = \frac{b}{-3a} \tag{38}$$

$$\frac{dy}{dx} = 3ax^2 + 2bx + c \tag{39}$$

Also, since the initial point would be zero speed

$$x(0) = 0 \tag{40}$$

By using the condition equations (36), (39), and (40), the cubic equation can be solved and get the values of a, b, c, and d.

Optimizing the line's upper bound's x value means to have the shortest time to reach the maximum defined velocity. Therefore take the x value as the cost function:

$$Cost = \int_0^t \frac{-b \pm \sqrt{b^2 - 3ac}}{3a} \tag{41}$$

Use the upper bound, lower bound, and turning point's value as constraint. The constraint of turning point would be

$$Constraint: -Acc \leq \frac{dy}{dx} = 3\text{a}x^2 + 2bx + c \leq \text{Acc} \tag{42}$$

By using these conditions to minimize the cost in Matlab optimization tool, the shortest time to reach the maximum defined velocity can be found. The result of the optimization is shown in the result chapter.

CHAPTER V

HARDWARE IMPLEMENTATION

In this chapter, the object is to show the design of the hardware for the two wheel inverted robot. In the previous chapter, the simulation has been done. Therefore, in this chapter, the hardware will be designed to test the result which got from the simulation. The parameters and the properties should fit the simulation variables. For example, the size of the body, the radius of the wheel, and the mass of every part.

For the purpose of costing down and experiment efficiency, LEGO Mindstorm EV3 package is decided to be used. The LEGO packages have been used for a lot of experiments and projects since it is easy to build, depart, or change the design of a robot. What's more, it is safe while testing since the materials are plastics. Also, the program of the LEGO Mindstorm EV3 is easy to learn and it is designed by LABVIEW. The interface and the logic is very similar to LABVIEW. In this way, by encode the program to the EV3 brick, the robot can be given orders. The changing of code is quick and easy, there is no need for the hardware changing or replace any part due to the code changing.

The hardware design of the robot will base on the LEGO Mindstorm EV3 official gyro boy.

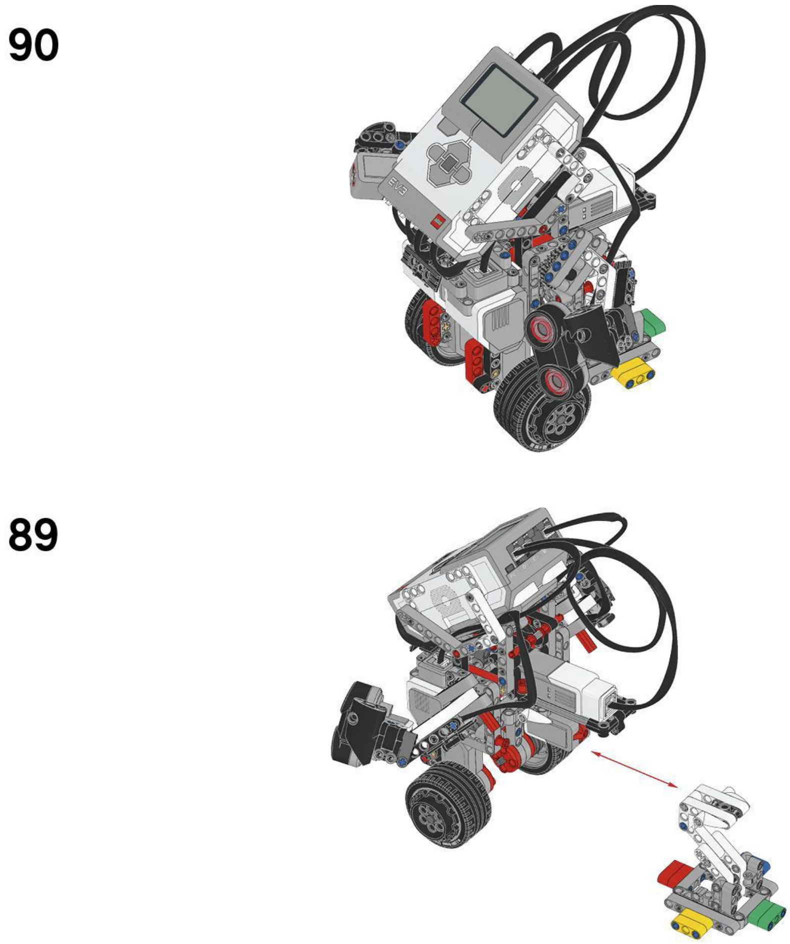

Figure 15. LEGO Mindstorm Ev3 official Gyro boy

Base on the original design shown in figure 15 above, there are little changes being made. For the sake of convenience of changing the design, the part which is not being used is taken off. The changes can't affect the position of the mass center, the mass center should always on the top of the wheel. At the back of the robot, in the original design, there is a small motor placed at the back to make the mass center on the top of the wheel and at the middle of the robot, but it is faced inside. In the original design, the small motor at the back is not used at all. It is placed there only for the balancing reason. Since the motor will be useless if it face inside, for the sake of future develop and experimental reasons, the small motor is turned facing outside. In the future, if the robot is needed to go with one wheel, the small motor can help to balance the robot.

Figure 16. Front view of Unicycle robot

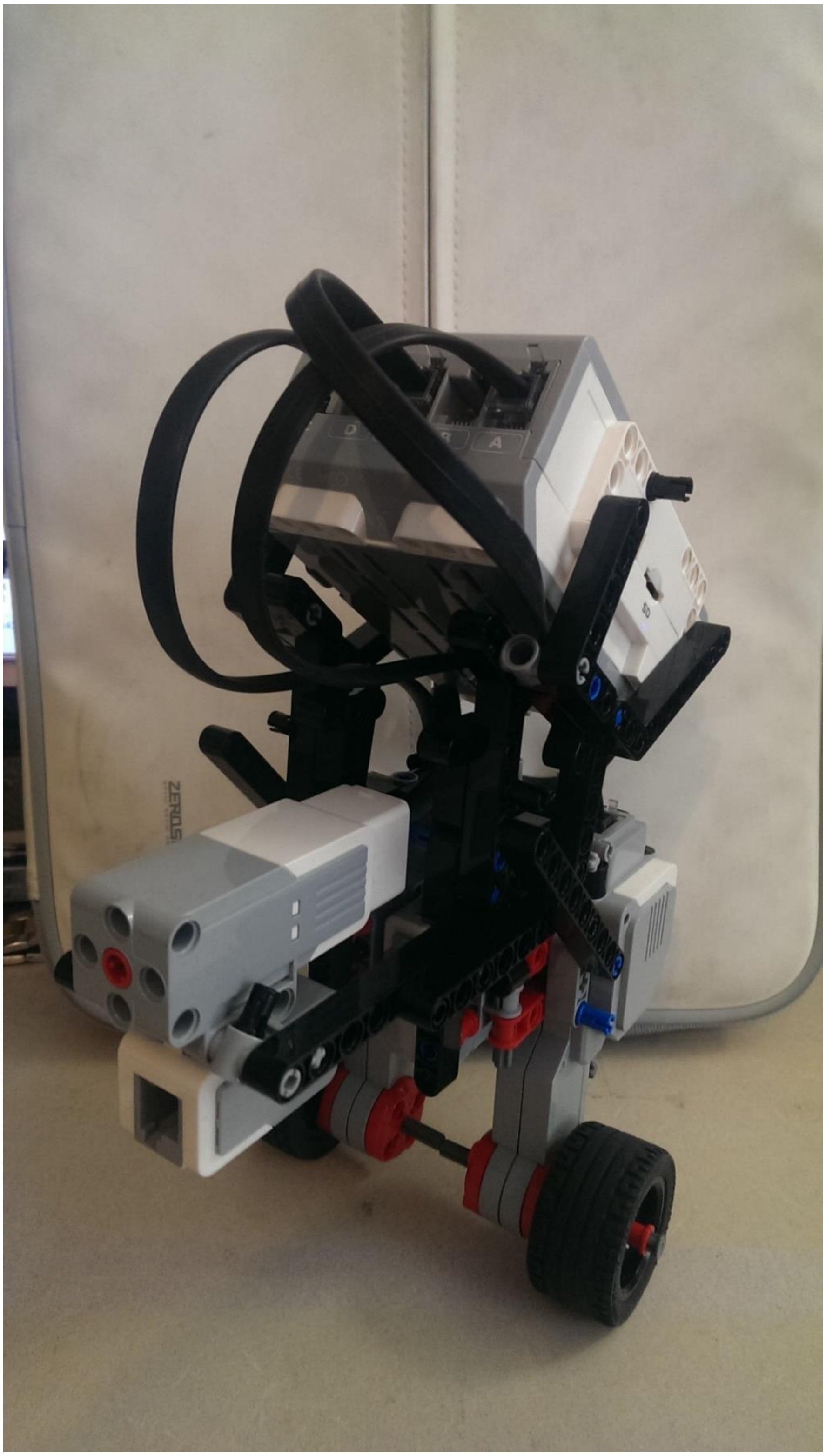

Figure 17. Back view of Unicycle robot

There are two large motors at the bottom of the robot to control two wheels. Each motor controls a wheel. The two robot is connected with the brick body. The motors rotate in pitch angle clockwise or counter clockwise. To balance the robot in pitch angle, the two motors are set to rotate synchronized. The motor will need to rotate differently while the robot needs to turn.

The body is set on the two wheels' large motors and leaning back around forty degree. In this way, while pushing the button to activate the robot it is easier to see the screen and proceed. Figure 16 and figure 17 above show the robot which is built in this thesis.

CHAPTER VI

SIMULATION RESULTS

The PID controller controls the body's angle as shown in figure 18 below. The body's angle can be controlled to stay upright all the time. However, due to there's no control for the wheel, the wheel moves randomly. In this model, the wheel would just rolling all the time under the condition of the body is stable. Therefore, the robot would go to somewhere that can't be predicted.

In this model, under the condition of controlling the body's angle, the robot will keep going in one direction all the time.

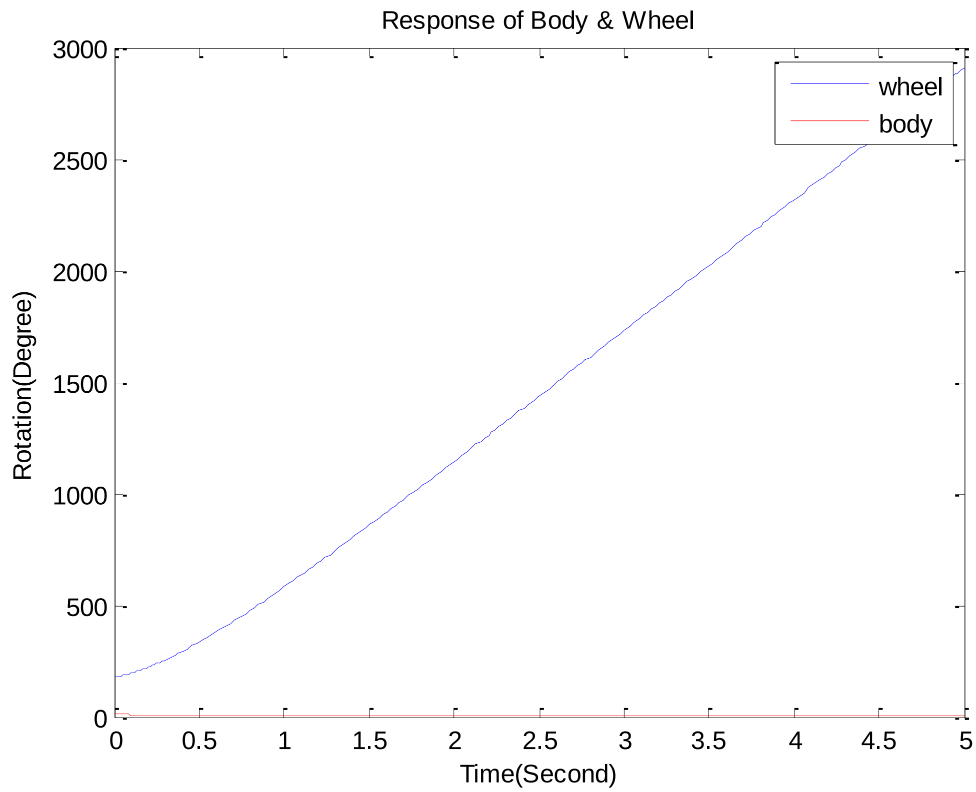

Figure 18. Response of robot's body and wheel

Figure 19 shows only the body's behavior from figure 18. There's no problem that the body can remain stable. The body would be stabilized by the PID controller around three seconds and the initial angle is ten degrees leaning to the front.

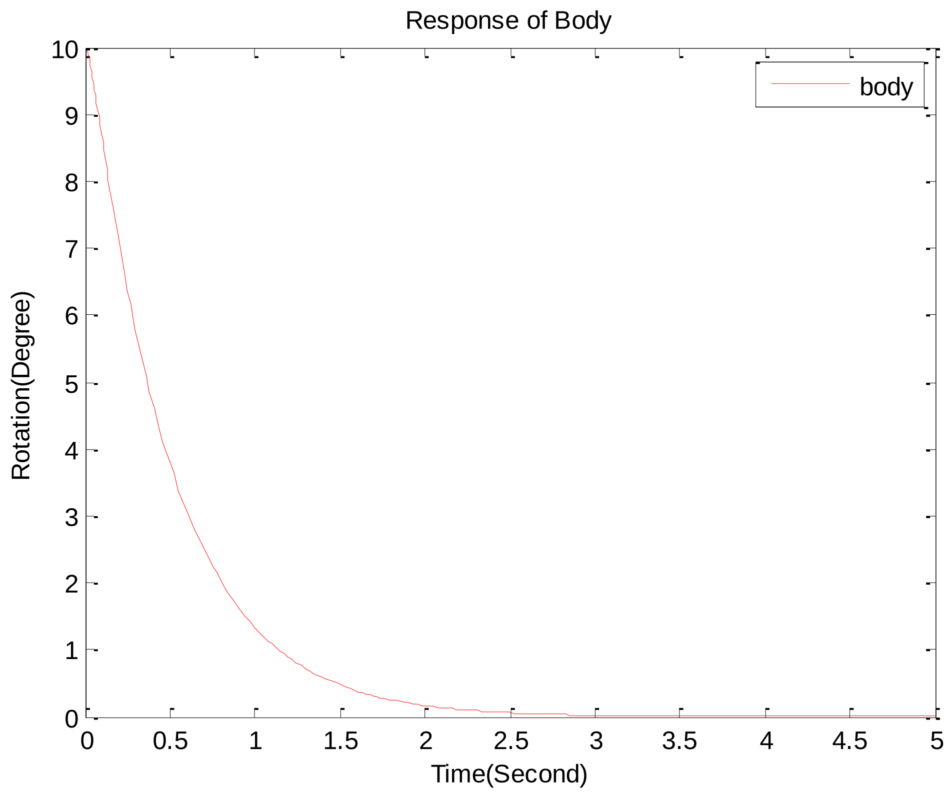

Figure 19. Response of robot's body angle test1

The second test is to put two PID controllers on the robot, the first controller controls the body's angle and the second controller controls the wheel's position. Then, take the summation of the PID controller's signal and input the signal to the wheel.

From figure 20 below, the wheel seems vibrating around some random point and the body is stabilized. However, for the wheel, the distance away from the initial point is very large and the vibration is also too large. Hence, the control strategy doesn't seem work.

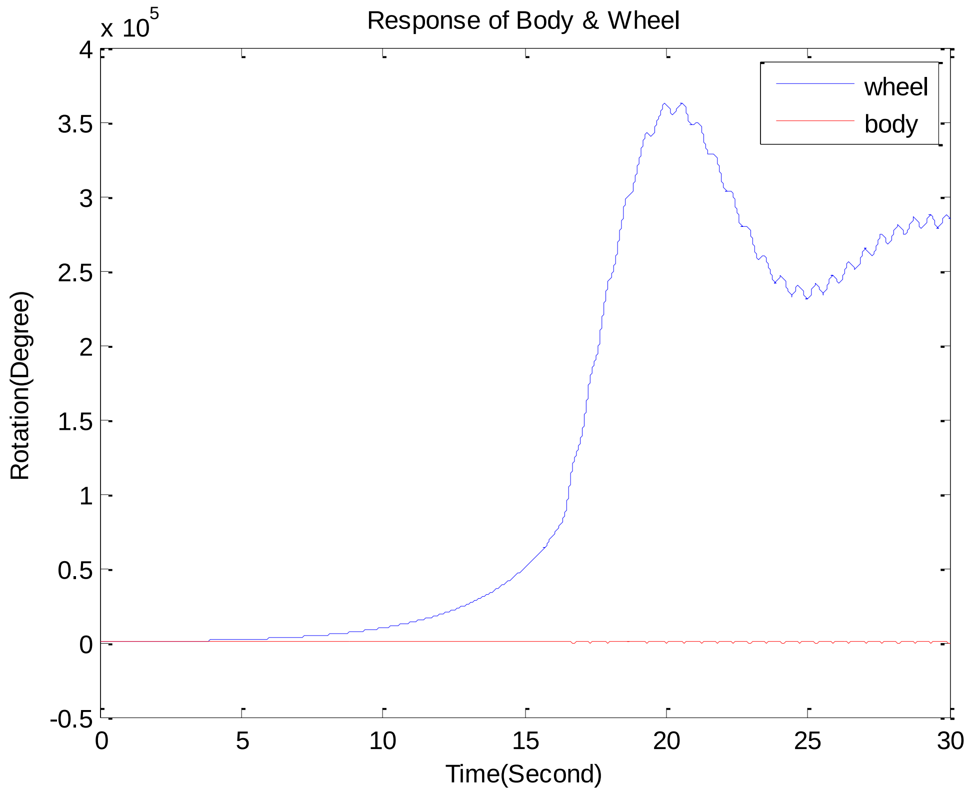

Figure 20. Response of robot's body angle and wheel angle test1

Figure 21 below shows only the body's angle for the dual-PID-controller controls. It shows that although from the previous figure 20, the body seems stable, but in fact, it still vibrates with a very big value. Hence, the control strategy is invalid.

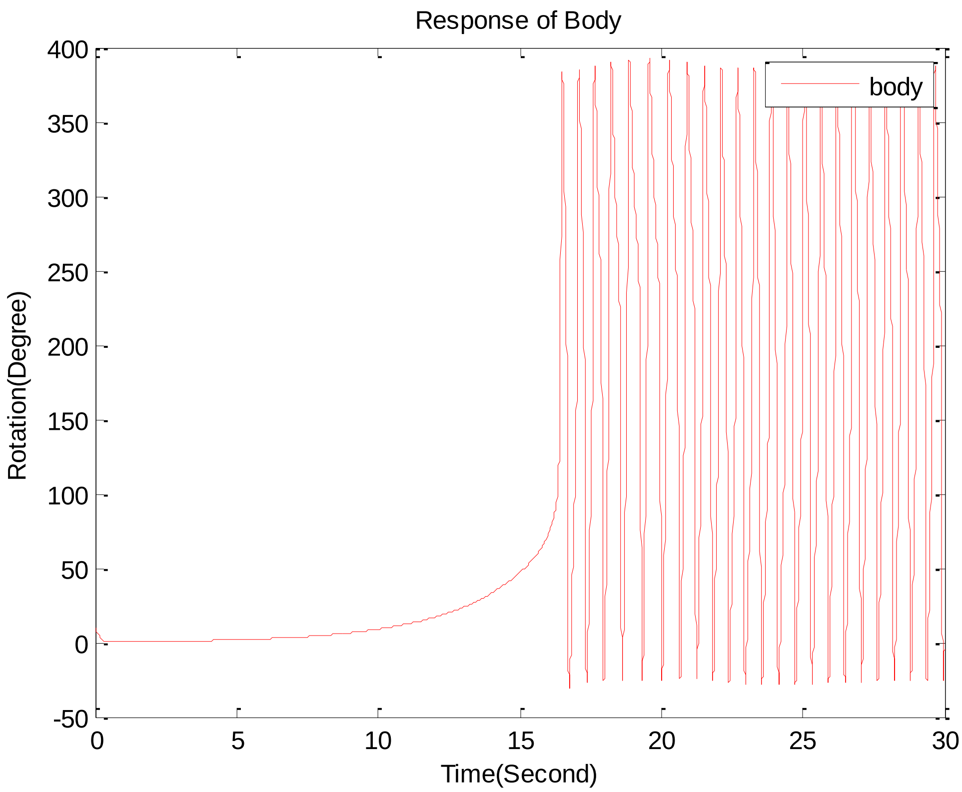

Figure 21. Response of robot's body angle test 2

The two figures figure 22 and figure 23 below show the same result as the previous two figures. The PID controllers are tuned to different values, so the behaviors of the body and the wheel is a little different than the previous one. Still, the result is quite the same. The wheels still roll to somewhere unpredictable and the body is not even stabilized in this case.

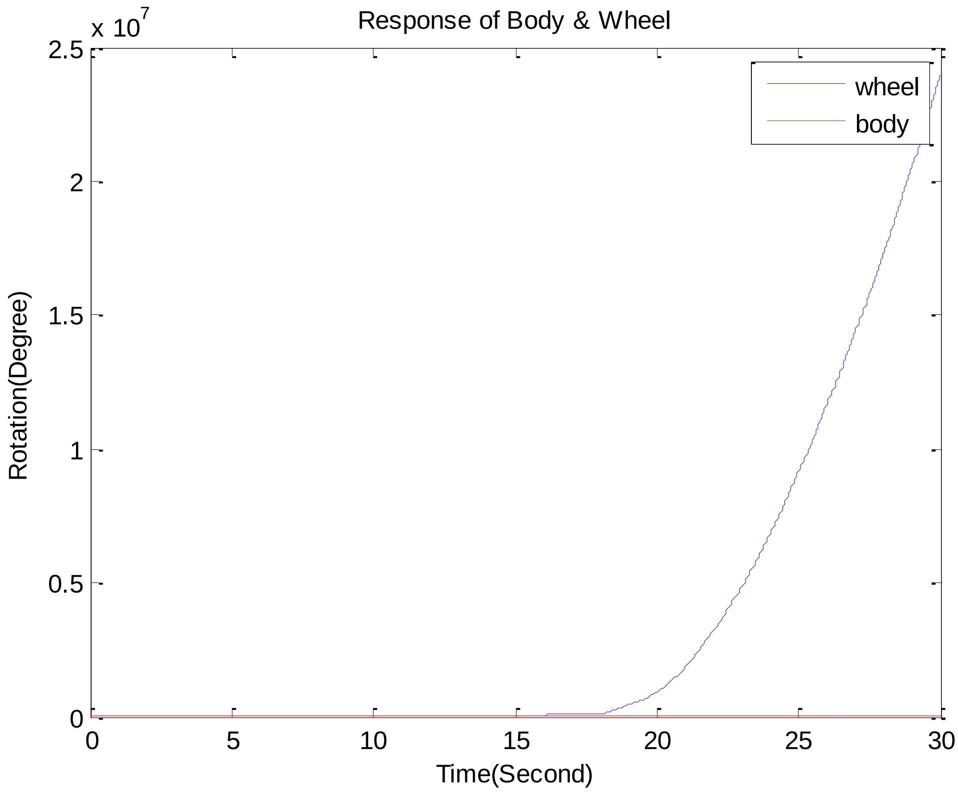

Figure 22. Response of robot's body angle and wheel angle test 2

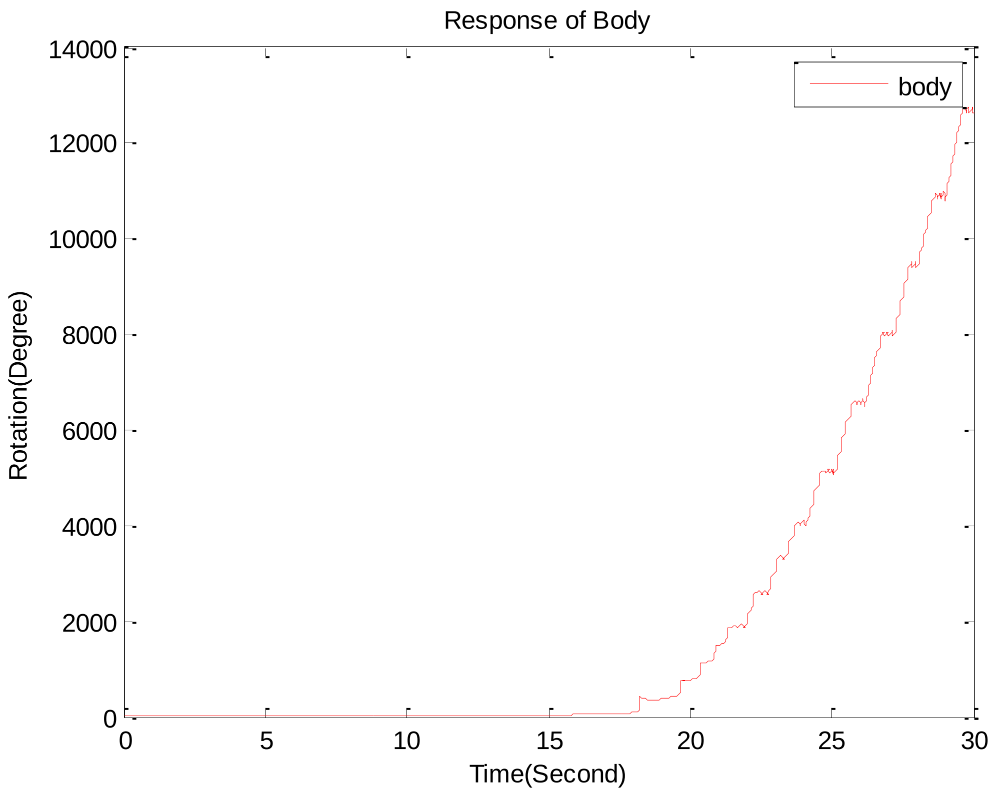

Figure 23. Response of robot's body angle test 3

In this part, the result of the LQR controller controls the robot is shown. The LQR controller controls all the states of the robot, which are body's angle, body's angular velocity, wheel's angle, and wheel's angular velocity. Figure 24 below shows the LQR controller controls the whole system. It can stabilize both wheel and body's angle. It takes around five seconds to stabilize both states and both of them would stay around the stable position.

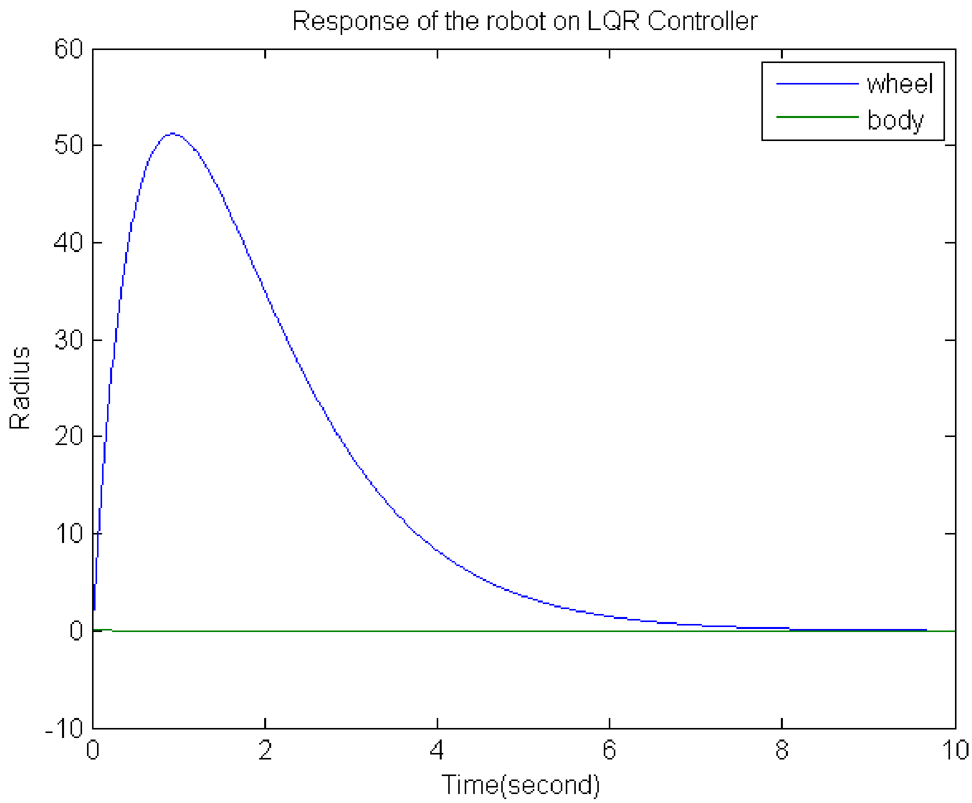

Figure 24. Response by LQR controller of robot's body angle and wheel angle

Figure 25 below shows the body's response for the LQR controller. It takes around five seconds to stabilize the body's angle and the body's angle doesn't not go away from the stable position in any time. Therefore, the experiment is valid and the LQR controller is useful for the two wheel inverted pendulum robot.

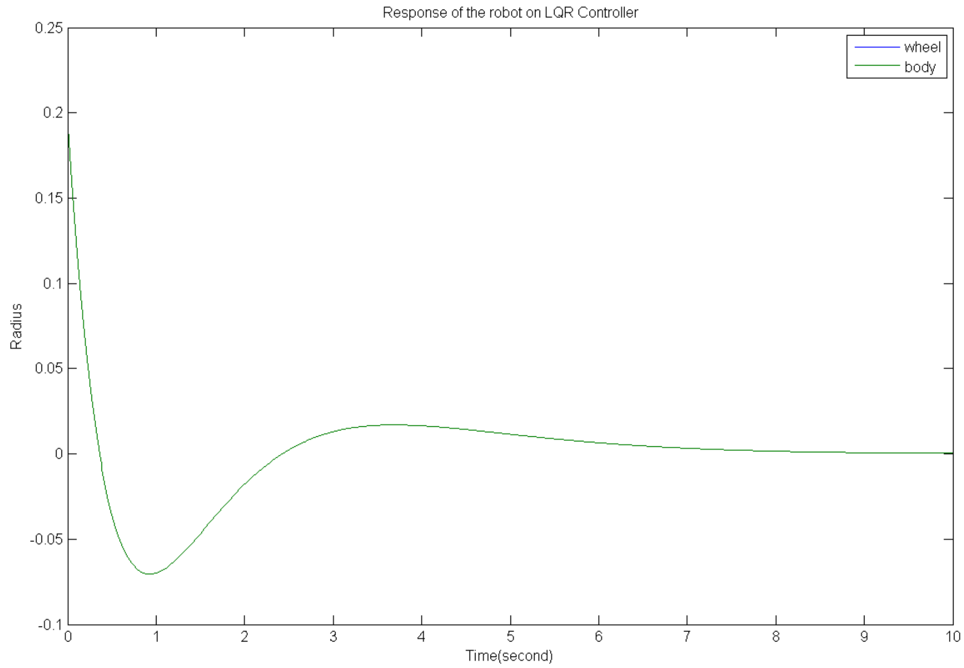

Figure 25. Response of LQR controller of robot's body angle

The result for LQR plus PID controller for the robot is shown in this part. The combination of these two controller helps the robot have a better performance. The LQR controller would control the robot's body and the wheel. However, there is steady state error for the wheels' position and some other performance can be improved. By adding a PID controller to the system, the steady error can be decreased and the other performances can be adjusted to match the requirements.

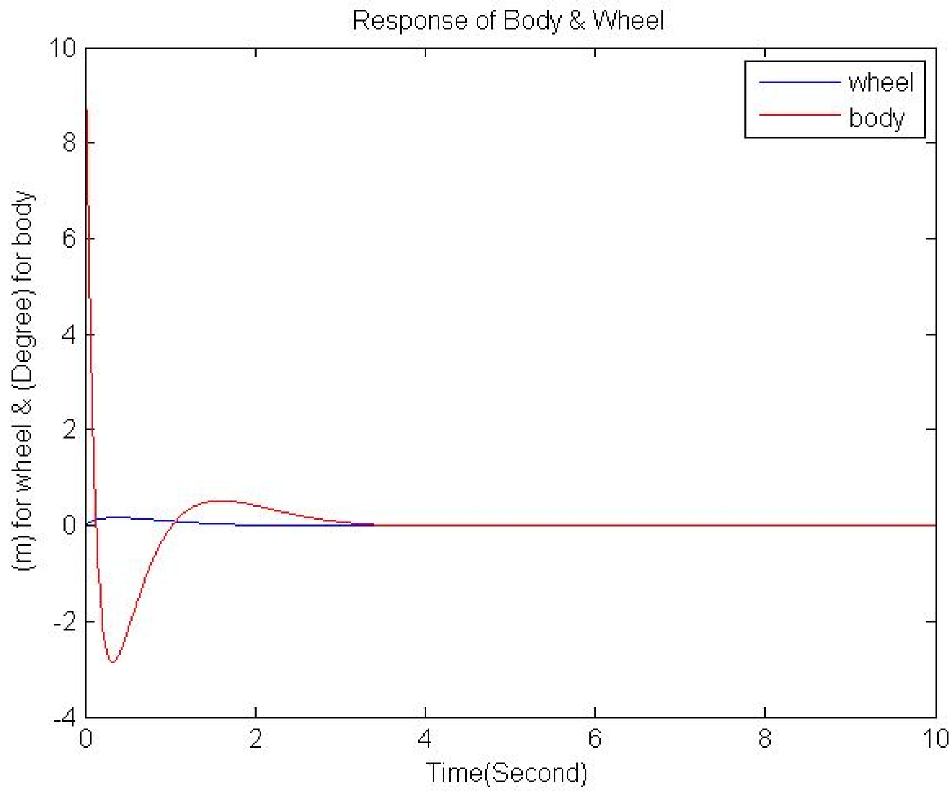

Figure 26. Response of LQR plus PID controller on nonlinear model of robot's body angle and wheel angle test 1

Figure 26 above shows the LQR controlled system. The two figures figure 27 and figure28 below shows the LQR controller plus differently tuned PID on the robot. Through a well set LQR controller, the system can perform nicely. There has no big problem with it. Through the PID controller, the body's angle or the wheel's position movement both can be controlled. Figure 27 below shows how the system act when it is added a PID controller to control the wheels' position. Since the wheels have no big peak or steady state error and the performance is good already, the effect on the wheel is small. However, while improving the wheels' performance, the body's angle is effected, too, since the system is coupled.

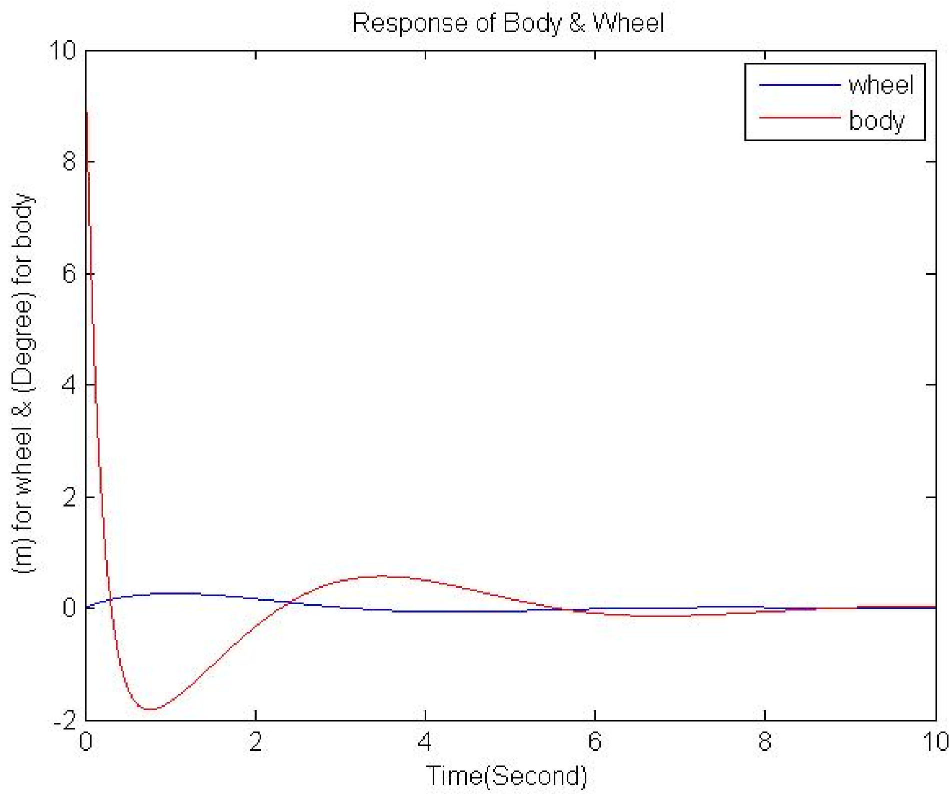

Figure 27. Response of LQR plus PID controller on nonlinear model of robot's body angle and wheel angle test 2

Figure 27 above shows the system with LQR controller and tuned PID controller. Both body's angle and the wheel's position is controlled. The point would be the performance of the system. With the value KP=1, KI=1, and KD=1, the body's leaning angle is decreased, but the wheels will come back and forth a little bit more than before. Therefore the PID can't be said well-tuned.

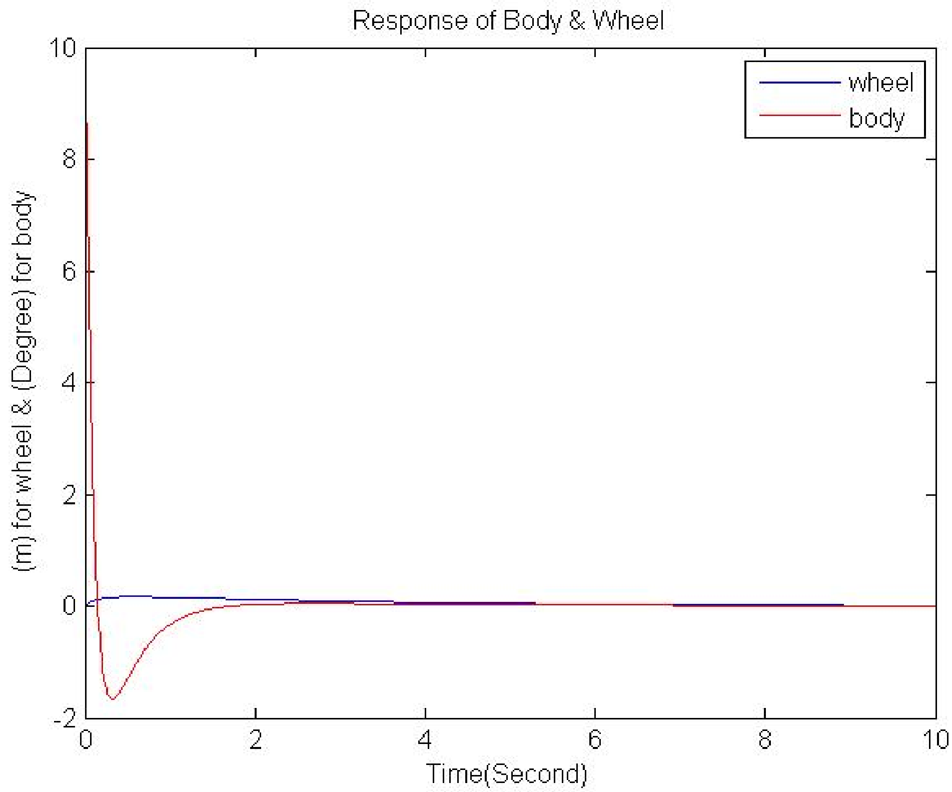

Figure 28. Response of LQR plus PID controller on nonlinear model of robot's body angle and wheel angle test 3

Figure 28 above shows the system with LQR controller and tuned PID controller. The value of the parameters are KP=1, KI=0.1, and KD=0.01. The system. The result shows that the body's leaning angle is decreased and the wheels' rolling is not effected much. The performance of the system is better and improved.

In this part, the simulation of movement of the robot is shown. Figure 29 below shows the result for the body leaning to a specific angle and the wheel stay at a spot. From the first figure, the wheel is rolling back and forth from zero to six hundred degree. Under the condition, the body hold an leaning angle for as long as it can.

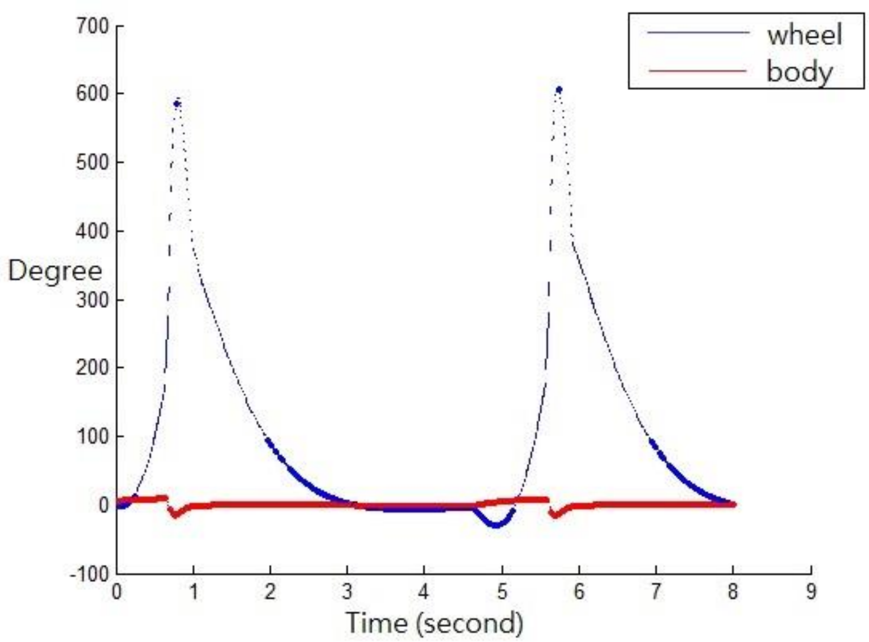

Figure 29. Response of LQR plus PID controller of robot's body angle and wheel angle test 1

Figure 30 shows the body's angle along. The body's angle is changing all the time but it would remain around five degrees as long as it can.

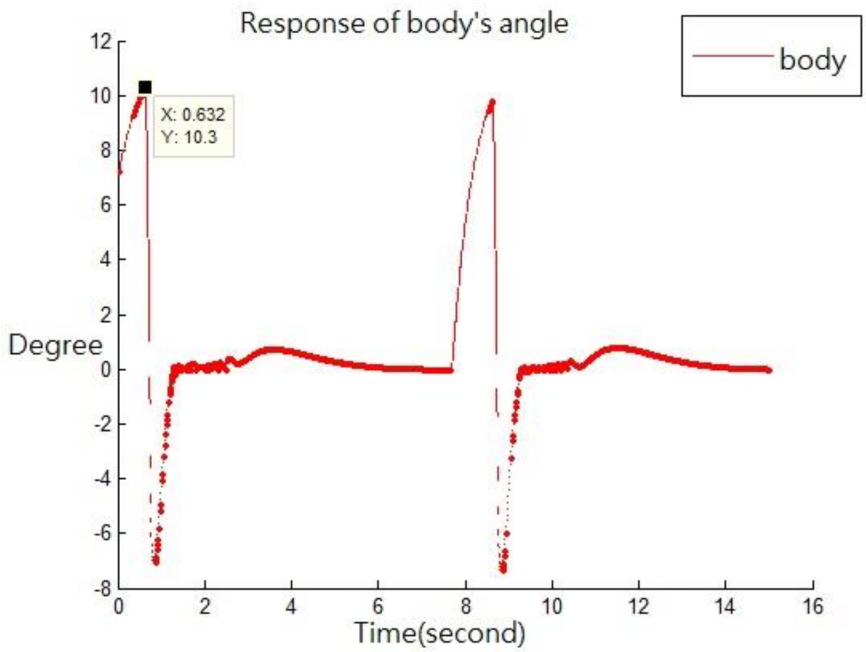

Figure 30. Response of LQR plus PID controller of robot's body angle test 1

Figure 30 above shows a clearer figure of how the body react while the wheels are rolling.

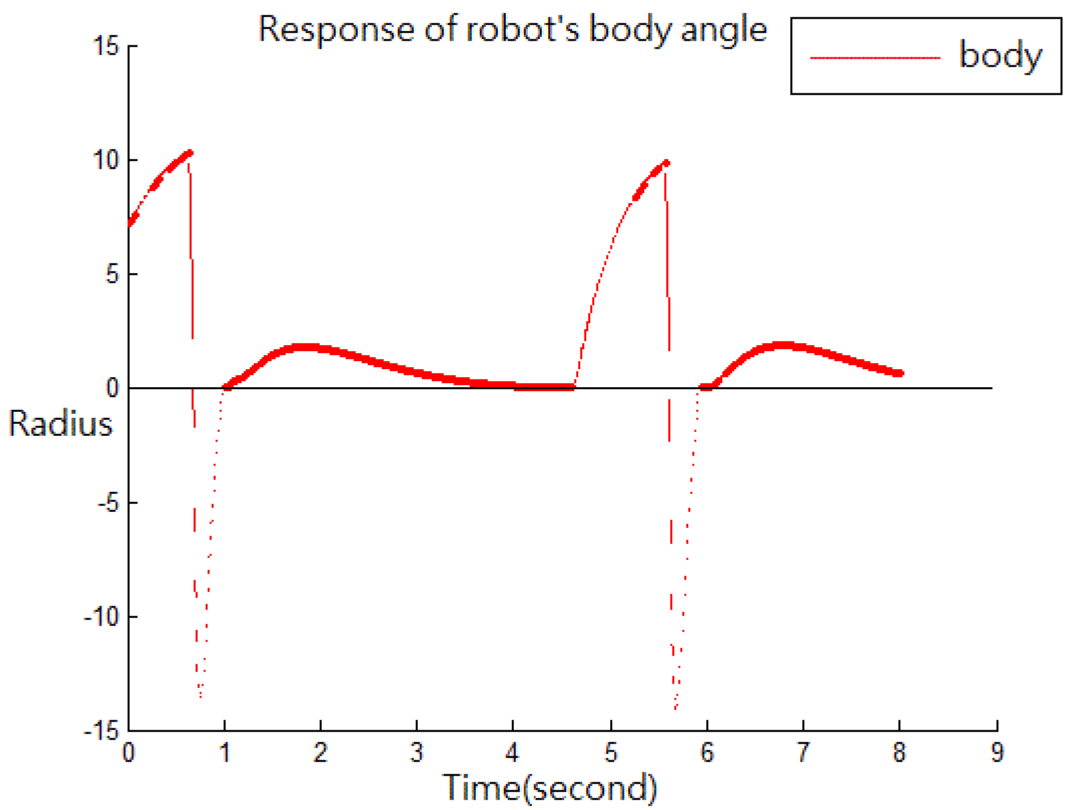

Figure 31. Response of LQR plus PID controller of robot’s body angle test 2

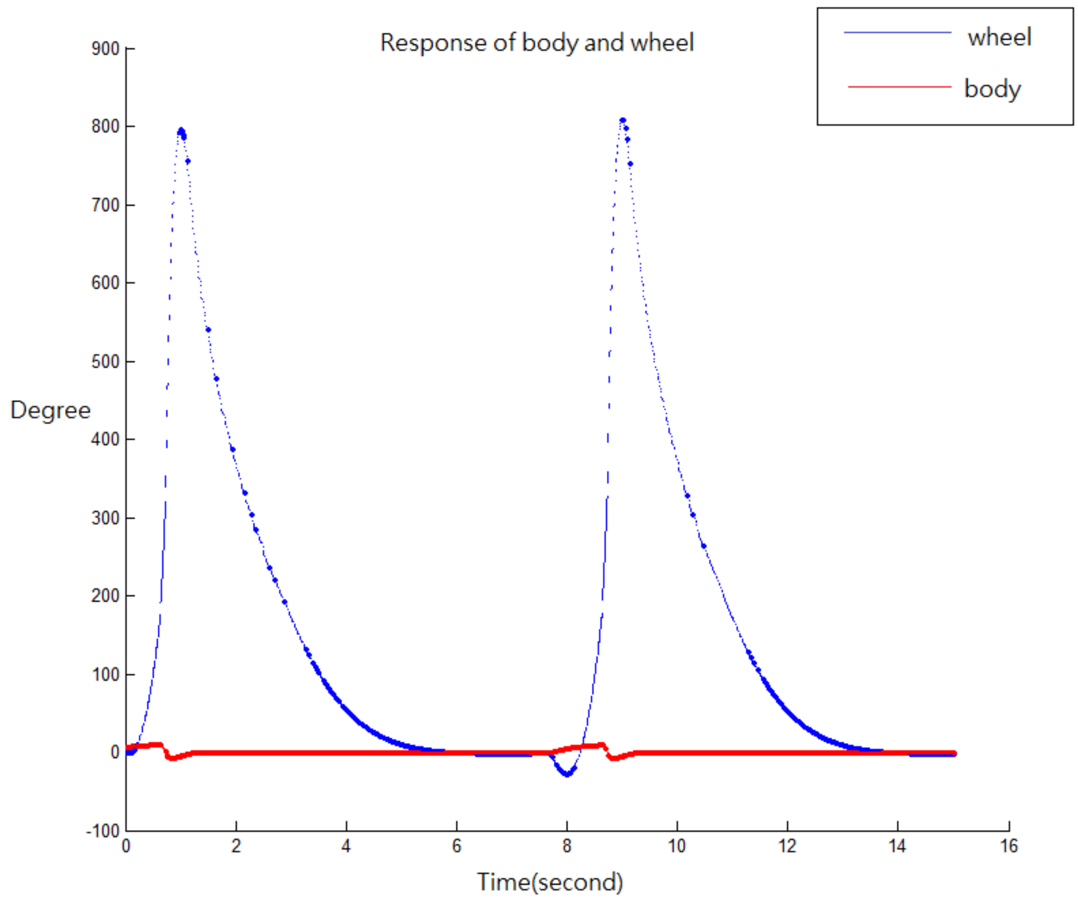

Figure 32. Response of LQR plus PID controller of robot's body angle and wheel angle test 2

The two figures above figure 31 and figure 32 show the performance of the two wheel robot. The body's angle and the wheel's position is shown. The figures indicate that the wheel will comes back and forth periodically and try to remain in a certain range. At the beginning, the wheel rolls back a little bit to lean the body to the required angle, then rolls the opposite direction to remain the body's angle. If the body's angle goes the other way too much, the wheel will roll opposite direction again. Once the wheel's position reaches the set limitation, it will start to roll back. At that time, the body's angle will goes to the other side for a short moment. The body's angle will be averagely at the required angle which is set. The body's angle can't be kept at a specific angle with a small motion of the wheel. The body's angle would come back and forth in a certain range that is set previously.

In this part, the result of braking strategy is shown. Through the LQR controller and PID controller, the distance needed for stopping and the maximum leaning degree can be found for each driving speed. By using these data, the robot can choose different controller parameters for stopping to fit the requirement.

Table1. Relationship between controller's value and stopping distance

Velocity(m/s)	Distance(m)	Maximum Degree	PID Value			
5	5.7	18.29	-2000,10,80			
	7.84	12.9	-3500,10,80			
	10.37	9.2	-5500,10,80			
	12.67	7	-7500,80,200			
	13.74	6.4	-8500,10,80			
7	5.427	33.48	-1000,10,80			
	8.76	22.25	-2500,10,80			
	10.73	18.05	-3500,10,80			
	12.57	15.14	-4500,10,80			
	14.31	13.03	-5500,10,80			
	15.96	11.42	-6500,10,80			
	17.55	10.17	-7500,10,80			
10	6.79	44.94	-1000,10,80			
	10.28	34.7	-2000,10,80			
	14.78	25.32	-3500,10,80			
	17.48	21.25	-4500,10,80			
	21.23	17.21	-6000,10,80			
	26.89	12.98	-8500,10,80			
12	7.54	51.11	-1000,10,80			
	11.68	40.51	-2000,10,80			
	15.48	32.88	-3000,10,80			
	18.94	27.47	-4000,10,80			
	22.13	23.51	-5000,10,80			
15	7.73	58.12	-1000,10,80			
	13.29	48.12	-2000,10,80			
	18.25	39.83	-3000,10,80			
	22.73	33.64	-4000,10,80			
	26.84	28.95	-5000,10,80			
	30.67	25.33	-6000,10,80			
	34.29	22.45	-7000,10,80			
17	7.77	61.35	-1000,10,80			
	16.98	48.05	-2500,10,80			
	24.97	37.51	-4000,10,80			
	29.73	32.43	-5000,10,80			
	34.15	28.47	-6000,10,80			

Table1 Continued

Velocity(m/s)	Distance(m)	Maximum Degree	PID Value			
20	11.19	61.28	-1500,10,80			
	18.25	53.56	-2500,10,80			
	24.82	46.16	-3500,10,80			
	30.83	40	-4500,10,80			
	41.54	31.13	-6500,10,80			
	51.03	25.28	-8500,10,80			
	55.45	23.09	-9500,10,80			
	59.69	21.2	-10500,10,80			
22	18.83	56.58	-2500,10,80			
	32.83	43.23	-4500,10,80			
	55.39	27.64	-8500,10,80			
	60.3	25.26	-9500,10,80			
	65.01	23.25	-10500,10,80			
25	19.36	60.13	-2500,10,80			
	35.38	47.65	-4500,10,80			
	49.3	37.79	-6500,10,80			
	61.57	31.08	-8500,10,80			
	67.25	28.46	-9500,10,80			
	72.69	26.22	-10500,10,80			

By using the data above, the information is given for the braking distance needed with different initial speed. Also the maximum degree would reach while braking is also shown in the table.

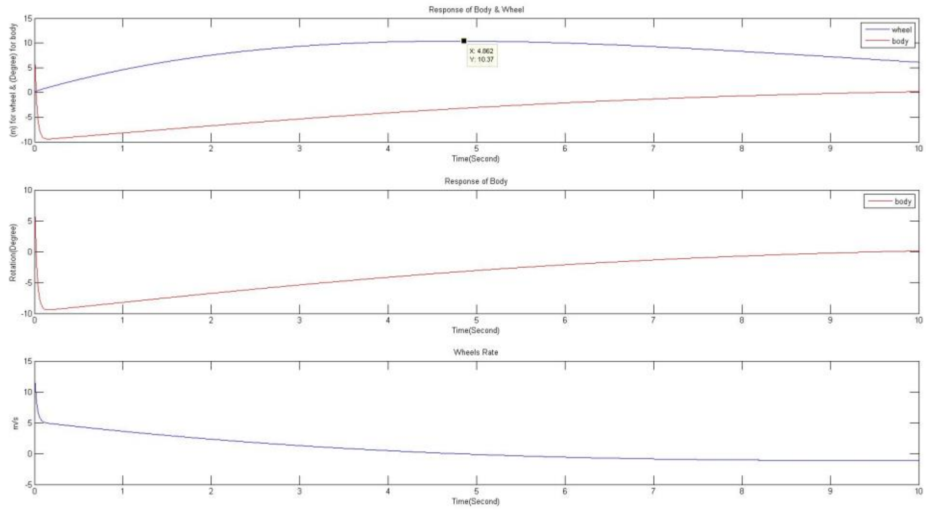

Figure33. Example of braking of the system

Figure 33 above shows an example of how the braking works. The robot has an initial leaning angle and an initial speed. While the robot start to brake, the wheel will speed up and the body will lean to the back. After reaching the maximum back leaning angle, the robot starts to decrease the leaning angle and slow down. While the robot is almost stop, the leaning angle will approach to zero, although there is still a small angle.

In this part, the result of speed control is shown.

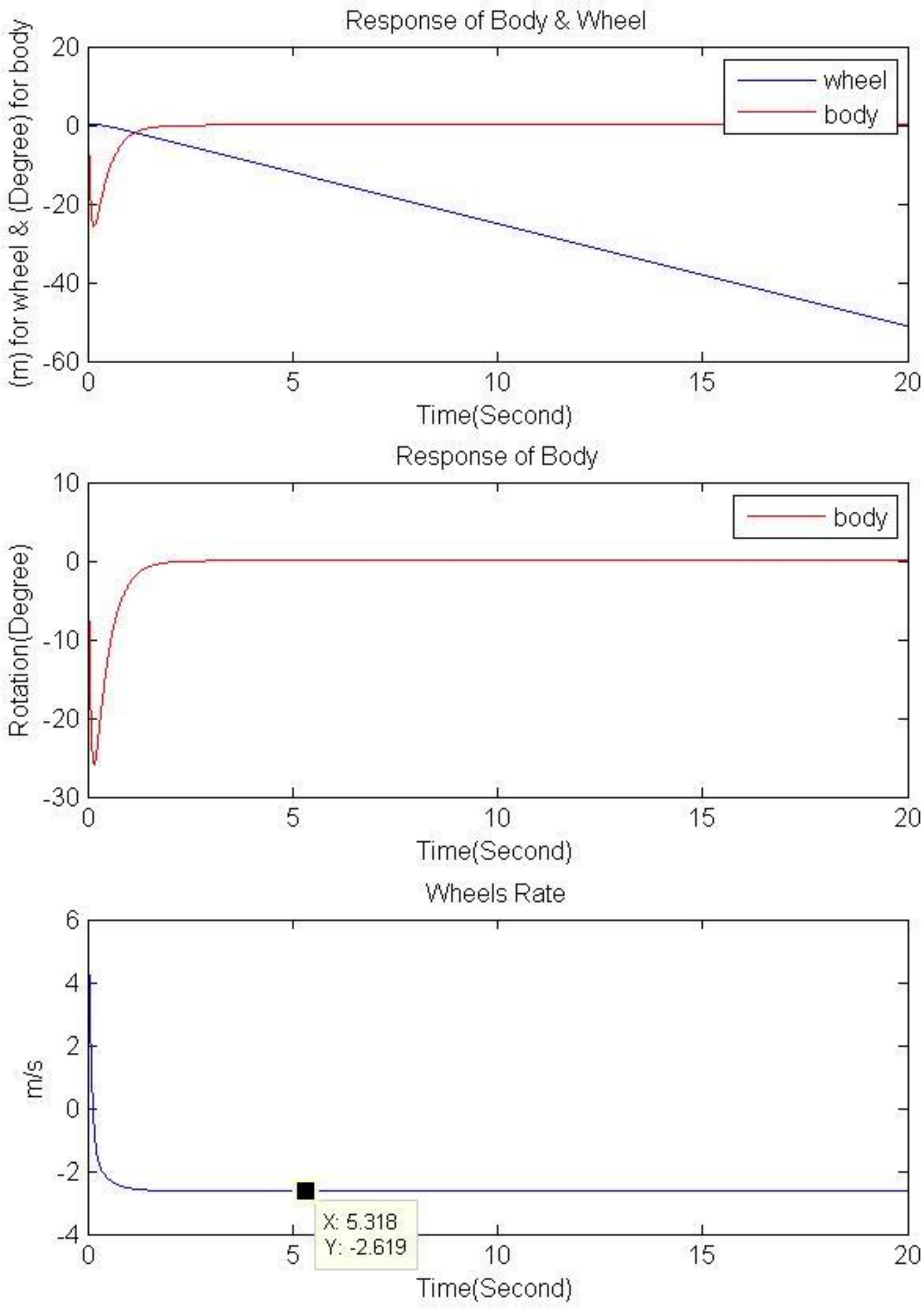

Figure34. Example of controlling speed

Figure 34 above shows the speed is controlled to reach to a certain value. At the same time the body's angle would remain zero when the speed reach constant. At the beginning, the wheel would rolls back a little bit to cost the robot lean to the front. After that, the robot can start to speed up. While speeding up the acceleration always remain positive but with different values. The acceleration getting smaller and smaller when the system is reaching the required speed. While the system reach the required speed, the acceleration becomes zero and the speed remain the value. Since the leaning angle is relative to the acceleration, when the acceleration decrease, the leaning angle decrease. When the acceleration goes to zero, the leaning angle gradually goes to zero.

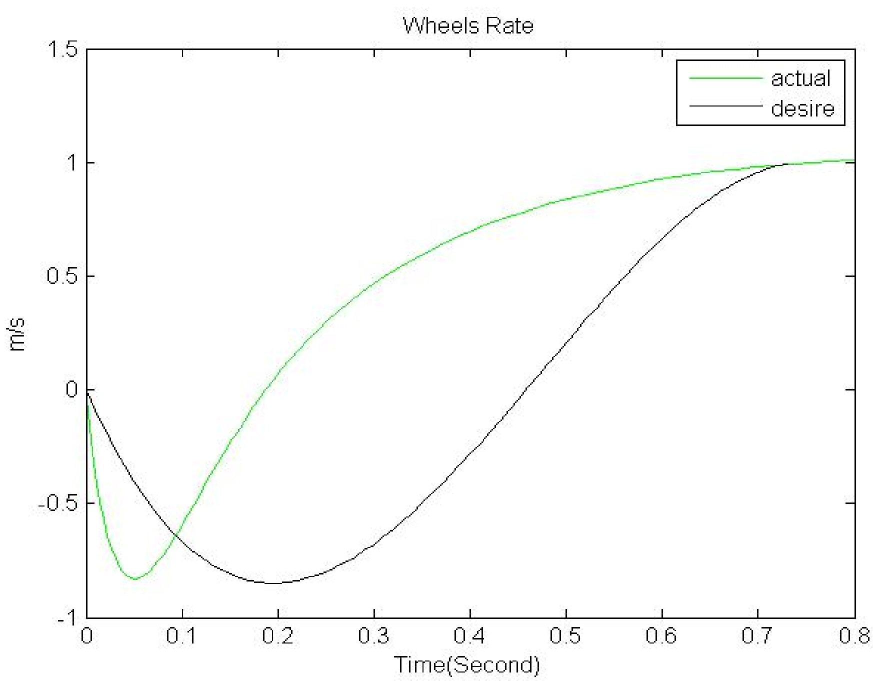

Figure35. Optimized velocity control

Figure 35 above shows the optimized velocity and the actual velocity. The black line is the desired velocity and the green line is the actual one for the robot. There are slightly different between the desired one and the actual one since the desired one does not guarantee the balance of the body. However, for the robot, the body's balance is the first priority, so the actual velocity cannot absolutely follow the desired velocity.

CHAPTER VII

SUMMARY

In this thesis, the simulation of the two wheel inverted pendulum and hardware implementation is shown. It shows the same results on both simulation and hardware experiment. The result from the work of this thesis could be useful but some part can still be improved.

The one PID controller can controls the body's angle but no control of the wheels position. The body would stay upright all the time, but the wheel is uncontrollable. In this case, the using of only one PID controller is not a good idea to control the robot. Therefore, a dual PID control strategy is tested. The purpose is to control both body's angle and wheel's position at the same time. However, since the two states are not decoupled, the PID signal for each state is seen as disturbance to the other state, the system can't be stabilized in this way. Even set one PID controller for a very small value for a state, it still cause the other state unstable and diverse in the end. Hence, the use of one or two PID controllers to control the system is invalid. There is the other control strategy needed to control the robot.

The second controller being tested is the LQR controller. Due to the property for LQR controller, it can control multi input multi output system. For this robot, there are one input and two outputs. There are four states of the system, the LQR controller consider all the states then comes out a gain value to control the whole robot. The robot can be stabilized both the body's angle and the wheels' position around four seconds. However, there would be steady state error for the wheel's position. In this case, to reduce the steady state error of the wheel's position, a PID controller is added to the system to help control the wheel's position. Through this way, the LQR controller can stabilize the whole system and the PID controller can help improve the system's performances

Through the previous work, the robot can be stabilized. The third step is to control the movement of the robot by the LQR controller and PID controller. The first movement is leaning to the front and move to the front which is the same direction as the body leaning. The result show that the robot can lean to a direction and move to the same direction at the same time. Due to the wheel's movement and the body's angle is not decoupled, the angle of the body is related to the wheel's speed. Generally, the robot can lean and move to the same direction.

The next task is to let the robot's body lean to an angle and the wheel stay at a spot at the same time. To achieve this requirement, the switching between LQR controller and PID controller is used. The robot is hold at an angle for most of the time and the wheel comes back and forth around a small region. In this way, the robot seems leaning to an angle and stay at the place.

The last task is to have the robot go from point to point within a shortest time. By optimizing the velocity of the robot due to the balancing needed. The robot would go with a different speed during the way. The robot start from initial point with zero speed and zero leaning angle to the final point with zero speed and zero leaning angle. The time is the shortest and optimized.

The work has been done and the results are satisfied although there are still some part can improve and there are still works needed to be done. There are a lot of experiments and simulations can test on the two wheel robot. The ability of movement can be designed and created by people. For example, decoupled the wheel and the body. Let them do different things without interrupt each other. On the other hand, the combination of the robot with a glasses remote controller could be interesting. The use of the glasses controller is for people who is disable in some function. The combination of this two product can really help disable people have a better life and improve living experience.

REFERENCES

[1] Amir A. Bature, Salinda Buyamin, Mohamed. N.Ahmad, Mustaphpa Muhammad (2014) A Comparison of Controllers for Balancing Two Wheeled Inverted Pendulum Robot. *International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS Vol:14 No:03*

[2] Osama Jamil, Mohsin Jamil, Yasar Ayaz, Khubab Ahmad (2014) Modeling, Control of a Two Wheeled Self-Balancing Robot. *2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)Islamabad, Pakistan, April 22-24, 2014*

[3] Wikipedia PID controller http://en.wikipedia.org/wiki/PID_controller accessed on Nov. 3, 2014

[4] Bennett, Stuart (1993). *A history of control engineering, 1930-1955*. IET. p. p. 48. ISBN 978-0-86341-299-8.

[5] Mathworks: Inverted Pendulum: PID Controller Design http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlPID accessed on Nov. 14, 2013

[6] Wikipedia Linear-quadratic regulator http://en.wikipedia.org/wiki/Linear-quadratic_regulator accessed on Mar. 12, 2015

[7] Kuo-Chun Wu (2007) Analysis and comparison of PID and state feedback controller in inverted pendulum system. 2007 Master's thesis, National Kaohsiung University Department of Electrical Engineering, Kaohsiung, Taiwan

[8] Bernhard Mahler, Jan Haase (2013) Mathematical Model and Control Strategy of a Two-wheeled Self-balancing Robot. *978-1-4799-0224-8/13/$31.00 IEEE*