Lakehead University

FUZZY ADAPTIVE CONTROL OF A TWO-WHEELED INVERTED PENDULUM

by

Tianqi Xi

Under the Supervision of Dr. Xiaoping Liu

For Fulfilling the Partial Requirements of the Master of Science in the Electrical and Computer Engineering

Lakehead University, Thunder Bay, Ontario, Canada

Jan. 2018

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Xiaoping Liu for his patience, motivation, immense knowledge and the continuous support of my master's thesis at Lakehead University. I was deeply affected by his strong passion and dedicated working ethics. His guidance helped me all the time academically and emotionally through the rough road to finish this thesis. I could not have imagined having a better supervisor and mentor for my Master study. Besides my supervisor, I would also like to show sincere thanks to my thesis reviewers: Dr. Natarajan and Dr. Bai for their valuable comments on my thesis, based on which my thesis has been improved a lot.

I would like to thank Zhengqi Wang, who as a good friend, was always willing to help and give his best suggestions. Many thanks to Jiajun Wang, Ning Che, and other colleagues in the laboratory for the sleepless nights we were working together, and for all the fun we have had in the last two years. My research would not have been possible without their helps.

Finally, but by no means least, thanks go to my family: my father and mother for almost unbelievable support and great understanding. They are the most important people in my world and I dedicate this thesis to them.

Sincerely

Tianqi Xi

Abstract

Recently, the two-wheeled inverted pendulum has drawn the attention of robotic community in view of a plethora of applications, such as transport vehicles: Segway, teleconferencing robots, and electronic network-vehicle. As a widely-used personal transportation vehicle, a two-wheeled inverted pendulum robot has the advantages of small size and simple structure. Moreover, with the advent of modern control technology, these kinds of platforms with safety features and sophisticated control functions can be cost down, so that they have high potential to satisfy stringent requirements of various autonomous service robots with high speed. At the same time, it is of great interest from control point of view as the inverted pendulum is a complicated, strongly coupled, unstable and nonlinear system. Therefore, it is an ideal experimental platform for various control theories and experiments.

To understand such a complex system, the Lagrangian equation has been introduced to develop a dynamic model. And following the mathematical model, linear quadratic regulator control and fuzzy adaptive method are proposed for upright stabilization, velocity control and position control of the system. However, sometimes these kinds of robots need to move on a slope, so an advanced linear quadratic regulator controller and a modified fuzzy adaptive controller have been proposed to achieve position control on a slope for the robot while stabilizing its body in balance. In addition, trajectory tracking control using proportional integral derivative control and sliding mode control with fuzzy adaptive backstepping method is also designed to make the robot autonomously navigate in two dimensional plane.

Simulation results indicate that the proposed controllers are capable of providing appropriate control actions to steer the vehicle in desired manners. Then, a couple of real time experiments have been conducted to verify the the effectiveness of the developed control strategies.

Contents

List of Figures

List of Tables

List of Abbreviations

TWIP – Two-Wheeled Inverted Pendulum.
PWM – Pulse Width Modulation.
LQR – Linear Quadratic Regulator .
PID – Proportional Integral Derivative .
RMP – Robotic Mobility Platform.
FLC – Fuzzy Logic Controller.
EN – V – Electric Networked-Vehicle.
DOF – Degree of Freedom.
USB – Universal Serial Bus.
ICSP – In Circuit Serial Programmable.
AC – Alternating Current.
DC – Direct Current.
IDE – Integrated Development Environment.
I2C – Inter-Integrated Circuit.
PCB – Printed Circuit Board.
OSP – Open-Source Projects.
APM – Arduino Pilot Module.
ESC – Electronics Speed Control.
IMU – Inertial Measurement Unit.
PPM – Pulse Position Modulation.

Chapter 1

Introduction

1.1 Overview of Two-Wheeled Inverted Pendulum Robot

The Two-Wheeled Inverted Pendulum (TWIP) is a pendulum vehicle with two wheels and consists of an inverted pendulum system and a mobile system. Generally, the tilting angle of the body is detected by a gyroscope and an accelerometer, and to keep itself balanced, a microcontroller is used to generate Pulse Width Modulation (PWM) signals to each motor. A motor drives each wheel independently, and the torque from the motors makes the vehicle move to balance the pitch angle of the pendulum. It can also move along curved paths by driving the motors at different speeds.

One of the first reported implementations of a TWIP robot was done by Kawamura in 1988 [1]. In 1996, Ha [2] developed an autonomous two-wheeled inverse pendulum type robot, called the "Yamabica Kurara". This robot was driven by two independent driving wheels on the same axle and had a gyro type sensor to measure the inclination angular velocity of the body. In the same year, Shiroma designed similar robots that coordinated with each other to carry a load cooperatively [3]. In 2003, Bui [4] developed a welding pendulum that was able to follow a specified welding trajectory.

Several approaches have been used to stabilize two-wheeled inverted pendulums. In the year of 2000, Ding used a sliding mode control scheme to deal with parametric and functional uncertainties [5]. Pathak [6] used partial feedback linearization to design a two level position-velocity controller in 2005. In 2007, Jeong and Takahashi [7] implemented a Linear Quadratic Regulator (LQR) state feedback controller for their mobile humanoid experimental robot. Meanwhile, Li implemented Proportional Integral Derivative (PID) control in an experimental vehicle with two reference inputs corresponding to a human transport mode and a goods transport mode [8]. In 2009, Li and Xu implemented an adaptive fuzzy controller, while Vlassis applied a Monte Carlo expectation-maximization algorithm to achieve balance by model free reinforcement learning [9]. In 2011, a fuzzy logic controller was designed by Huang to achieve stabilization and velocity control [10]. A TWIP robot provides massive opportunities for people to design applications and as a typical nonholonomic system, motion

control of TWIP systems is full of challenge.

Figure 1.1: Segway

The two-wheeled inverted pendulum has been proposed as a portable transporter due to its high maneuverability and simple structure. It has been suggested as a suitable unit for office and home environments. Controlling such a system is a challenging problem due to its nonlinearities and complex dynamics. The modeling of the system is also complex because of the slipping and rolling constraints of the wheels. In spite of its dynamic complexity, numerous two-wheeled inverted pendulums have been created by research institutions and companies.

Figure 1.2: Legway

In 2000, Segway was introduced to the world as the first human transporter, and it was marketed as the transporter alternative, as shown in Fig 1.1. And there are several different Segway models. The I167 model is shown in Fig 1.3. In this model, the driver can turn by twisting a grip located on the left of the handlebar. Another version of the Segway x2 in Fig 1.4 is intended for off-road use, which has wider tires and higher ground clearance.

Legway robot was designed by Lego company, which is capable of moving on the inclined surface or even on irregular ground as shown in Fig 1.2. In 2005, a TWIP robot was

Figure 1.3: Segway HT i167

Figure 1.4: Segway x2 SE

introduced as a service unit for reporting fire and intruders in indoor settings. Similar robots, such as the uBot series which have been shown in Fig 1.5 and the Segway Robotic Mobility Platform (Segway RMP) series, continue to be used as mobile research platforms for diverse studies within the field of robotics. Balance Bot was designed in Brno university. It is a low cost TWIP robot but can achieve obstacle avoiding.

In recent years, the TWIP has drawn more attention of robotic community in view of a plethora of applications, such as teleconferencing robots and Electric Networked-Vehicle (EN-V) shown in Fig 1.7 and Fig 1.8. The telepresence robot shown in Fig 1.6, also called "Anybot", has a panel showing the surrounding environment. Personal users can use the robot to help keep an eye on things when they cannot be there in person. Business clients can employ Anybot to strengthen in-office collaboration by expanding their employee's telecommuting options. More details can be seen in Table 1.1. EN-V is an electrically powered road vehicle created by Segway and adopted by General Motors as a concept vehicle representing the future of unban transportation, which is also built on the base of TWIP characteristics.

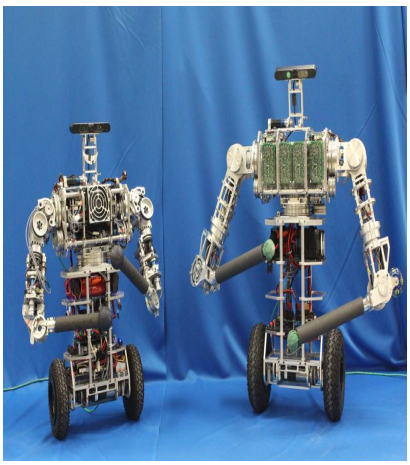


Figure 1.5: uBot-5

Figure 1.6: Anybots

1.2 Modeling of TWIP

Development of a dynamic model is necessary for understanding the physical phenomenon of a wheeled self-balancing inverted pendulum. Moreover, a dynamic model should be developed so that the controllers can be properly designed. Many approaches have been derived to describe a TWIP robot. In [11], a dynamic model of TWIP was constructed in the Newtonian reference frame using Kane's method [12]. With the kinematical information and the nonholonomic generalized active forces obtained, the equations of motion have been derived. In [13], the mathematical model of the inverted pendulum has been developed based on Lagrangian formulation. With the system kinetic energy and system potential energy, the TWIP system dynamics were presented in terms of two differential equations describing two DOF of the system. Furthermore, in [2], the system was modelled based on wheels' axles and its vertical axis. Using Lagrange's motion equation and linearizing in the neighbourhood of the upright state, the linearized model was obtained. In [14], the author used the similar method in [12] and developed a 2 DOF dynamic model for the inverted

Table 1.1: Anybots Specification

Top speed	5.6 km/h
The maximum vertical height	1.75 m
Display	320*240 mm
Camera	5 megapixels
Battery duration	8h
Weight	16 kg

pendulum on a slope.

To achieve the trajectory tracking control of a TWIP robot, a 3 DOF dynamic model has to be derived. In [6], the system model of the TWIP robot was described in the following state-space form using Euler-Lagrange equation.

$$M(q)\ddot{q} + V(q,\dot{q}) = E(q)\tau + A^T(q)\lambda \tag{1.1}$$

where τ is the control torques, λ is the constraint-force vector and $A(q)$ represents the three nonholonomic constraints. q is the state variable, which contains the position of the TWIP, pitch angle and yaw angle of the body.

1.3 Velocity and Position Control

Many different types of control schemes for velocity control and position control of inverted pendulum robots have been proposed over the years.

In [15], the author created a mathematical model for a nonlinear single link inverted pendulum-on-cart system by using Lagrange method. Based on Taylor series approximation, the nonlinear model can be linearized. LQR, double PID and pole placement control methods were developed to ensure the upright stabilization. The performances of the controllers were tested by real time experiments, and the LQR controller was more reliable based on the response time and level of disturbance rejection. In [16], the time specification performance between LQR controller and PID controller has been compared, and simulation study has been done in Matlab simulink environment. The results show that LQR produced better response compared to PID control method.

A similar LQR algorithm was introduced in [17] for the attitude and position control of the inverted pendulum robot. In this paper, Kalman filter was employed to process the angle signals from a gyroscope and an accelerometer.

Other control strategies are also proposed, such as sliding mode control and backstepping control in literature [14, 18–23]. In [20], the author aimed to elaborate a control law based on the combination of backstepping and PID controllers to steer the angle of the body. It can be concluded that the backstepping method is suitable for this kind of nonlinear systems. In recent year, an increasing number of approximator-based adaptive control

Figure 1.7: GMs EN-V Pride

Figure 1.8: GMs EN-V Magic

methods have been developed to extend the applicability of classical backstepping control scheme without full knowledge of system dynamics and to compensate model uncertainties as well. In [19], an adaptive integral backstepping controller with the velocity estimator was constructed to stabilize the wheeled inverted pendulum. To deal with the complex structure, a non-model-based differentiator based on the adaptive update law was proposed. The numerical simulation results show the effectiveness of the designed controller, though there are no signs of real experimental results. In [23], two sliding mode control (SMC) methods have been proposed for the inverted pendulum system and both of the strategies were capable of coping with parameter uncertainties and external disturbances.

Furthermore, in [24–28], a fuzzy logic control scheme has been implemented. In [24], a two-axis accelerometer and a rate gyroscope were used for collecting the actual angle of the robot, and the errors of the inverted pendulum's angle and angular velocity were used as the inputs to the fuzzy logic controller. The experimental results proved that the proposed controller can not only balance the angle of the pendulum but also track the desired speed successfully. An example of using a fuzzy controller to maintain proper control of an inverted pendulum has been presented in [26]. Control of the system was accomplished by the corresponding displacement of the carriage. At the stage of building a fuzzy controller, input and output linguistic variables with the corresponding membership functions and a rule base have been proposed. And the simulation results showed a relative satisfactory control performance. A novel design of a two-wheeled double inverted pendulum-like vehicle with a movable payload was presented in [28]. A hybrid fuzzy logic control approach has been

adopted to control and stabilize the inverted pendulum robot and different disturbances were applied to the system to test the robustness of the proposed control strategy.

1.4 Trajectory Tracking Control

The trajectory tracking control is an important part of the controller design for inverted pendulum robots. Plenty of researchers have devoted lots of time to this area of studies.

The widely used PID controller and LQR controller were studied and implemented in [2,29,30]. PD and PID methods were used to perform a line tracking task in [29]. The reference trajectory was captured by the camera and detected by image processing. The designed controller is capable of stable balancing and position tracking. In [2], the author focused on making the TWIP robot navigate autonomously in two dimensional plane with desired constant velocity while keeping balanced. A LQR control algorithm was used to keep the robot upright and track the given speed and trajectory.

The PID controllers have been used in industries as a dominant controller with advantages of simplicity of implementation and effectiveness of performance. Neural network and fuzzy method are very good tools for intelligent control under the circumstances with uncertainties and with no mathematical model. In [31], a neural network technique was applied to control a 2 Degree of Freedom (DOF) inverted pendulum on a x-y plane. The cart with an inverted pendulum moves on the x-y plane. Neural network works as an auxiliary controller to compensate for uncertainties of the system. Also, the controller was tested on the real inverted pendulum platform and effective experimental results were obtained. A dual-loop control method to realize the simultaneous balance and trajectory tracking control for the TWIP vehicle has been proposed in [32].

On the other hand, an adaptive backstepping sliding mode motion controller with fuzzy logic system has been developed in [33, 34]. Sliding mode control is a leading robust control technique as it is insensitive to those uncertainties implicit in the input channel, and it can be a very powerful method to solve the problem of trajectory tracking control in the presence of disturbances and uncertainties.

1.5 Research Motivation

Given the theory development in [2,4,6], range and variety of modern applications of inverted pendulum, a TWIP robot with emphasis on controller design is introduced in this thesis. Two modelling methods will be adopted to describe the TWIP system.

The first control scheme will be based on velocity and position control. Under the inspiration of [2, 15, 35], a LQR controller and a fuzzy adaptive backstpping controller will be utilized to achieve the velocity control and position control while maintaining the robot upright. Different from [2], the LQR controller and the fuzzy logic system will also be constructed

for the inverted pendulum robot on a slope, which meets the practical applications.

Enlightened by [29, 31, 33], a PID technique and a sliding mode backstepping control with fuzzy adaptive method will be used to achieve trajectory tracking control. To test the performance of the controllers, real time experiments will be conducted.

1.6 Thesis Outline

In this thesis, some brief descriptions of a hardware platform used in the experiments are given in Chapter 2. Chapter 3 introduces two different types of dynamic models. In Chapter 4, several controller strategies are developed and simulation results and experimental results are also conducted to show the effectiveness of the proposed control methods. Finally, Chapter 5 provides concluding remarks and some future work.

Chapter 2

Experimental Platform

The goal of this chapter is to describe the experimental setup used to implement the control strategy on the mobile inverted pendulum robot. It is vital to select the appropriate physical frames for the TWIP robot. Without a suitable microcontroller, a mobile two-wheeled robot is not nearly good enough for a complete controller design.

2.1 Platform Overview

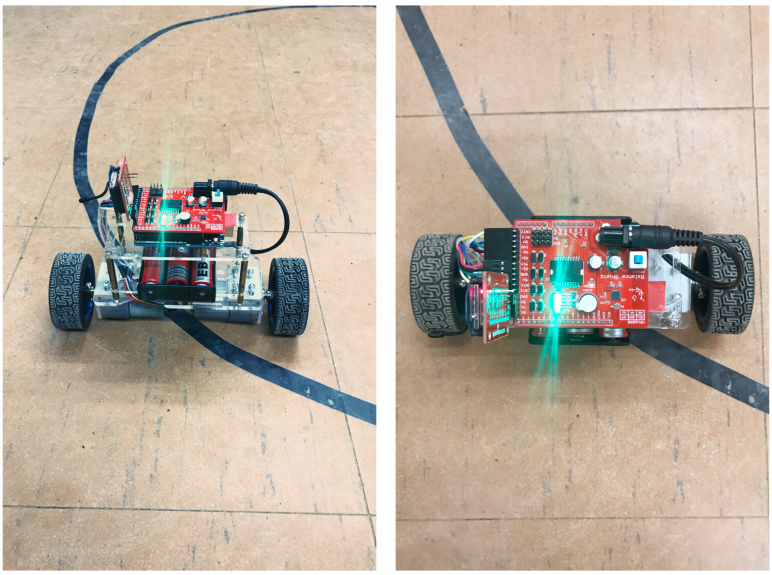


Figure 2.1: Experimental Platform

The experimental platform is based on the Balanbot provided by Maker Studio as shown in Fig 2.1. As you can see, the robot consists of a plastic body frame, two wheels and two 12V Direct Current (DC) motors attached at the bottom of the body frame. Three 3.7V

3000mAh rechargeable batteries are placed at the middle layer of the body frame. The Arduino UNO microcontroller, the balance shield and Xbee pro S1 wireless communication module are placed on the top of the frame. Fig 2.2 shows an overview of the experimental hardware.

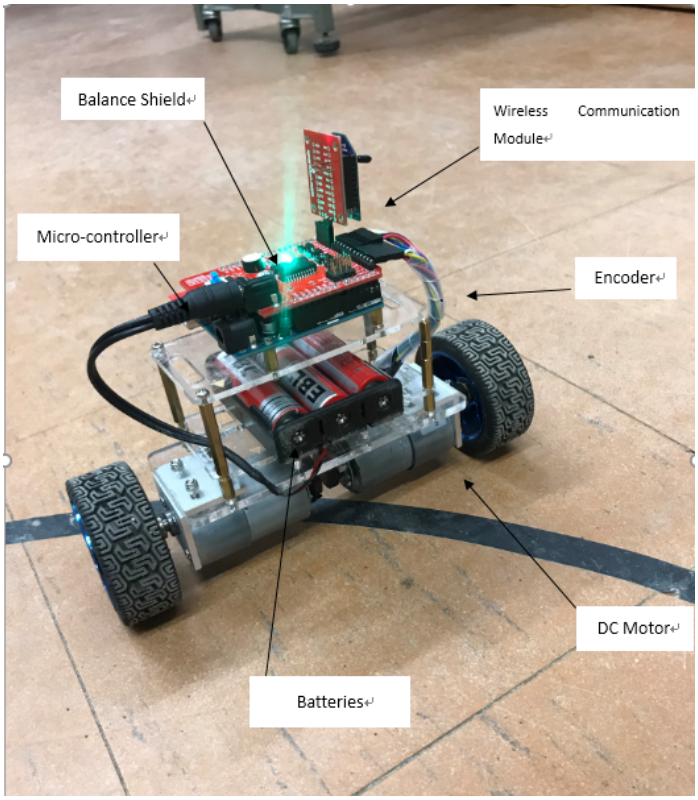


Figure 2.2: Overview of the TWIP Robot Hardware

2.2 Main Controller

2.2.1 Arduino Uno R3

Arduino [36] is an open-source microcontroller development board which can be used to read sensors and control things like motors and lights. This allows one to upload programs to this board which can then interact with things in the real world. With this, making devices which respond and react to the world at large is easy.

The Arduino Uno is a microcontroller board based on the ATmega328 which has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a Universal Serial Bus (USB) connection, a power jack, an In Circuit Serial Programmable (ICSP) header and a reset button. It contains almost everything needed to support the main controller.

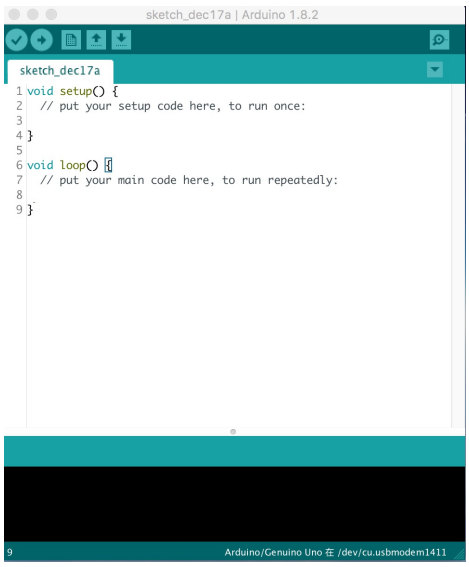


Figure 2.3: Arduino Interface

Figure 2.4: Arduino Uno R3

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform. For a comparison with previous versions, see the Arduino index of boards. The Arduino Uno R3 is shown in Fig 2.4 and technical specifications are given in Table 2.1. It is worth mentioning that Arduino is a developed platform which includes various of driver libraries for diverse sensors and actuators. It is intuitive, free to download, and compatible with Windows and Mac OS. More than 100 libraries are available for signal analysis and hardware accessories. The overview of the Arduino Integrated Development Environment (IDE) is shown in Fig 2.3.

Table 2.1: Arduino Uno R3

Microcontroller	ATmega328
Operating voltage	5V
Input voltage(Recommended)	7-12V
Input voltage(Limits)	6-20V
Digital I/O pins	14
Analog input pins	6
DC current per I/O pin	40mA
DC current for 3.3V Pin	50mA
Flash memory	32 KB of which 0.5KB used by bootloader
SRAM	2KB
EEPROM	1KB
Clock speed	16MHz

2.2.2 Balance Shield

The Balance Shield [37], as shown in Fig 2.5, uses an MPU6050 which is the world's first integrated 6-axis motion-tracking device that combines a 3-axis gyroscope and a 3-axis accelerometer. It connects to the UNO compatible board through the Inter-Integrated Circuit (I2C) interface. With this sensor, stable angle readings can be collected.

Figure 2.5: Balance Shield

MPU-6050 combines a 3-axis accelerometer and a 3-axis gyroscope on the chip. It offers a gyroscope with a full-scale range of ±250 degrees/sec, ±500 degrees/sec, ±1000 degrees/sec, ±2000 degrees/sec, and a user-programmable accelerometer with a full-scale range of ±2g, ±4g, ±8g, ±16g. Moreover, there is a configurable digital low-pass filter (DLPF), which provides a number of cut-off frequencies (see Table 2.2 for details).

The Balance Shield also integrates one L298P for driving motors. The L298P is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 package. It is a high voltage, high current dual full-bridge driver designed to accept standard Transistor-Transistor Logic (TTL)

Table 2.2: Low Pass Filter Specifications for the Accelerometer and Gyroscope

Accelerometer (Fs = 1KHz)		Gyroscope		
Bandwidth(Hz)	Delay (ms)	Bandwidth (Hz)	Delay (ms)	Fs(KHz)
260	0	256	0.98	8
184	2.0	188	1.9	1
94	3.0	98	2.8	1
44	4.9	42	4.8	1
21	8.5	20	8.3	1
10	13.8	10	13.4	1
5	19.0	5	18.6	1

logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic can work at a lower voltage.

2.3 Other Parts of the TWIP

2.3.1 Xbee Pro S1

The XBee [38] and XBee-PRORF Modules are engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The devices require minimal power and provide reliable delivery of data between devices. The devices operate within the Industrial Scientific Medical (ISM) 2.4 GHz frequency band and are pin-for-pin compatible with each other.

Figure 2.6: Xbee Pro S1

Fig 2.6 shows the very popular 2.4 GHz XBee XBP 24-AWI-001 module from Digi. The Pro

series have the same pinout and command sets as the basic series with an increase output power of 60 mW. These modules take the 802.15.4 stack (the basis for Zigbee) and wrap it into a simple to use serial command set. These modules allow a very reliable and simple communication between microcontrollers, computers, systems, really anything with a serial port. Point to point and multi-point networks are supported.

2.3.2 12V DC Motor with Encoder

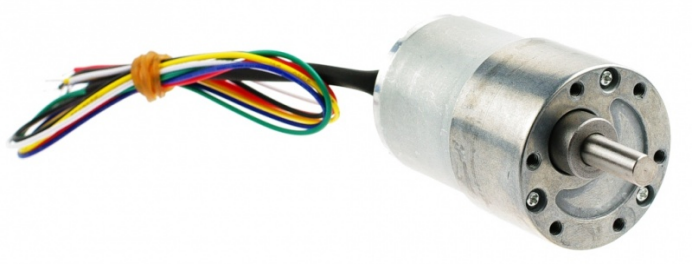


Figure 2.7: 12V DC Motor

This is a gear motor with encoder, model No.GB 37Y3530-12V-251R [39] shown in Fig 2.7. It is a powerful 12V motor with a 43.7:1 metal gearbox and an integrated quadrature encoder that provides a resolution of 64 counts per revolution of the motor shaft, which corresponds to 2797 counts per revolution of the gearbox's output shaft. These units have a 0.61 inches long, 6 mm-diameter D-shaped output shaft. This motor is intended for use at 12V, though the motor can begin rotating at voltages as low as 1V. The face plate has six mounting holes evenly spaced around the outer edge threaded for M3 screws. These mounting holes form a regular hexagon and the centers of neighboring holes are 15.5 mm apart, and all the features make the motor being an ideal option for mobile robot project (see Table 2.3 for details).

Table 2.3: Motor Specification

Gear ratio	43.7:1
No-load speed	251 + 10% RPM
No-load current	350mA
Start voltage	1.0V
Stall current	7A
Insulation resistance	20MΩ
EncoderOperating voltage	5V
Encoder type	Hall
Encoder resolution	16 CPR(motor shaft)/700 CPR(gearbox shaft)
Weight	205g

2.3.3 3.7V 3000mAh Rechargeable Batteries

Figure 2.8: Rechargeable Batteries

The Model 18650 3.7V 3000mAh rechargeable lithium batteries as shown in Fig 2.8, have large capacity and high quality. They have built-in Protection Circuit Board (PCB) protection boards making them long life batteries. Moreover, there is no memory effect and they can be recharged up to 1200 times when fully or partially drained. The improved low self discharged batteries have over current protection and over charge and discharge protection as well. They are good both in low and high temperature operations. Lastly, they provide excellent continuous power sources to the device.

Chapter 3

TWIP Dynamics

To steer an inverted pendulum robot, it is necessary to have its mathematical model such that a model-based controller can be successfully designed in order to achieve desired control goals. In this chapter, inspired by the work done in [2, 6, 40–42], two mathematical modelling methods are introduced. The first model is used for designing velocity control and position control for a TWIP robot while the second one for trajectory tracking control.

3.1 Dynamic Model with 2 DOF

The coordinate system of the TWIP robot is shown in Fig 3.1, where θ and ϕ are the rotation angle of the wheel and the tilting angle of the body, respectively. β is equal to the rotation angle θ minus the tilting angle ϕ, which is the wheel's relative rotation angle to the body [2]. α is the tilt angle of the slope.

Model assumptions are made as follows:

- Wheels do not slip on the slope;
- Body and the wheels are rigid;
- Robot has no lateral sliding.

Before giving a mathematical model for the TWIP robot in Fig 3.1, the physical meaning of the variables used in the modeling process is given in Table 3.1.

The total kinetic energy can be written into the following form.

Table 3.1: Parameters and Variables

Symbol and unit	Parameter and variable name
$M_b[kg]$	mass of the body
$M_w[kg]$	mass of the wheels
I_b	rotational intertia of the body
I_w	rotational intertia of the wheel
I_M	rotational intertia of the motor axis
$r[m]$	radius of the wheel
$l[m]$	length between the axle of wheel and gravitational center of the body
μ_s	viscous constant between the wheel axle including motor and gear
μ_g	viscous constant between the wheel and the ground
$g[m/s^2]$	gravitational acceleration
τ_i	torque constant of the motor
η	reduction ratio of gear
u	motor's input current

$$\begin{aligned} T &= \frac{1}{2}M_w r^2\dot{\theta}^2 + \frac{1}{2}M_b\left\{\left[\left(\dot{\theta}-\dot{\beta}\right)l\sin\left((\theta-\beta)+\alpha\right)\right]^2 + \left[r\dot{\theta}+\left(\dot{\theta}-\dot{\beta}\right)l\cos\left((\theta-\beta)+\alpha\right)\right]^2\right\} \\ &\quad + \frac{1}{2}I_w\dot{\theta}^2 + \frac{1}{2}I_b\left(\dot{\theta}-\dot{\beta}\right)^2 + \frac{1}{2}I_M\eta^2\dot{\beta}^2 \\ &= \frac{1}{2}M_w r^2\dot{\theta}^2 + \frac{1}{2}M_b\left[\left(\dot{\theta}-\dot{\beta}\right)^2 l^2 + r^2\dot{\theta}^2 + 2r\dot{\theta}\left(\dot{\theta}-\dot{\beta}\right)l\cos\left((\theta-\beta)+\alpha\right)\right] \\ &\quad + \frac{1}{2}I_w\dot{\theta}^2 + \frac{1}{2}I_b\left(\dot{\theta}-\dot{\beta}\right)^2 + \frac{1}{2}I_M\eta^2\dot{\beta}^2 \end{aligned} \tag{3.1}$$

The potential energy for the wheels is determined by

$$U_w = M_w g(r + \theta r\sin\alpha) = M_w gr(1+\theta\sin\alpha) \tag{3.2}$$

The potential energy for the body is given by

$$U_b = M_b gl(\cos\phi + \theta r\sin\alpha) = M_b gl(\cos(\theta-\beta) + \theta r\sin\alpha) \tag{3.3}$$

The total potential energy is calculated as

$$U = U_w + U_b = M_w gr(1+\theta\sin\alpha) + M_b gl(\cos(\theta-\beta) + \theta r\sin\alpha) \tag{3.4}$$

The dissipation energy function is

$$D = \frac{1}{2}\left(\mu_s\dot{\beta}^2 + \mu_g\dot{\theta}^2\right) \tag{3.5}$$

The external torques can be obtained as below.

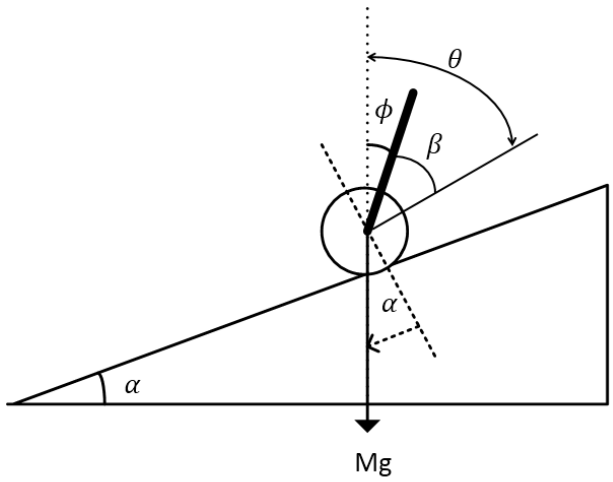


Figure 3.1: Schematic Diagram of TWIP Robot on Inclined Surface

$$Q_\beta = \eta\tau_i u,\ Q_\theta = 0 \tag{3.6}$$

Now, Lagrange's motion equation can be used to derive the dynamic model of the inverted pendulum system

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\beta}}\right) - \frac{\partial T}{\partial \beta} + \frac{\partial U}{\partial \beta} + \frac{\partial D}{\partial \dot{\beta}} = Q_\beta \tag{3.7}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) - \frac{\partial T}{\partial \theta} + \frac{\partial U}{\partial \theta} + \frac{\partial D}{\partial \dot{\theta}} = Q_\theta \tag{3.8}$$

where

$$\frac{\partial T}{\partial \beta} = M_b r\dot{\theta}\left(\dot{\theta} - \dot{\beta}\right) l\sin\left(\theta + \alpha - \beta\right) = M_b r\dot{\theta}\dot{\phi} l\sin\left(\phi + \alpha\right) \tag{3.9}$$

$$\frac{\partial T}{\partial \dot{\beta}} = -M_b lr\dot{\theta}\cos\left(\phi + \alpha\right) - M_b l^2\dot{\phi} + I_M\eta^2\left(\dot{\theta} - \dot{\phi}\right) - I_b\dot{\phi} \tag{3.10}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\beta}}\right) &= -M_b lr\ddot{\theta}\cos\left(\phi + \alpha\right) + M_b lr\dot{\theta}\sin(\phi + \alpha)\dot{\phi} - M_b l^2\ddot{\phi} \\ &\quad + I_M\eta^2(\ddot{\theta} - \ddot{\phi}) - I_b\ddot{\phi} \\ &= \left[-M_b lr\cos\left(\phi + \alpha\right) + I_M\eta^2\right]\ddot{\theta} - \left(I_M\eta^2 + M_b l^2 + I_b\right)\ddot{\phi} \\ &\quad + M_b lr\dot{\theta}\sin\left(\phi + \alpha\right)\dot{\phi}\end{aligned} \tag{3.11}$$

$$\frac{\partial U}{\partial \beta} = M_b g l \sin(\theta - \beta) \tag{3.12}$$

$$\frac{\partial D}{\partial \dot{\beta}} = \mu_s \dot{\beta} = \mu_s \left(\dot{\theta} - \dot{\phi}\right) \tag{3.13}$$

$$\frac{\partial T}{\partial \theta} = -M_b r \dot{\theta} \left(\dot{\theta} - \dot{\beta}\right) l \sin(\theta + \alpha - \beta) = -M_b r \dot{\theta} \dot{\phi} l \sin(\phi + \alpha) \tag{3.14}$$

$$\begin{aligned} \frac{\partial T}{\partial \dot{\theta}} &= M_w r^2 \dot{\theta} + \frac{1}{2} M_b \left[2l^2 \left(\dot{\theta} - \dot{\beta}\right) + 2r^2 \dot{\theta} + 2r(2\dot{\theta} - \dot{\beta}) l \cos((\theta - \beta) + \alpha)\right] \\ &\quad + I_w \dot{\theta} + I_b \dot{\phi} \\ &= M_w r^2 \dot{\theta} + M_b \left[l^2 \dot{\phi} + r^2 \dot{\theta} + r(\dot{\theta} + \dot{\phi}) l \cos(\phi + \alpha)\right] + I_w \dot{\theta} + I_b \dot{\phi} \end{aligned} \tag{3.15}$$

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) &= M_w r^2 \ddot{\theta} + M_b \left[l^2 \ddot{\phi} + r^2 \ddot{\theta} + rl(\ddot{\theta} + \ddot{\phi}) \cos(\phi + \alpha)\right] \\ &\quad - \left[r(\dot{\theta} + \dot{\phi}) l \sin(\phi + \alpha) \dot{\phi}\right] M_b + I_w \ddot{\theta} + I_b \ddot{\phi} \qquad (3.16) \\ &= \left[M_b l^2 + M_b r l \cos(\phi + \alpha) + I_b\right] \ddot{\phi} + \left[M_w r^2 + M_b r^2 + M_b r l \cos(\phi + \alpha) + I_w\right] \ddot{\theta} \\ &\quad - M_b r \left(\dot{\theta} + \dot{\phi}\right) \dot{\phi} l \sin(\phi + \alpha) \qquad (3.17) \end{aligned}$$

$$\begin{aligned} \frac{\partial U}{\partial \theta} &= M_w g r \sin\alpha + M_b g l (r \sin\alpha - \sin(\theta - \beta)) \\ &= M_w g r \sin\alpha + M_b g l (r \sin\alpha - \sin\phi) \end{aligned} \tag{3.18}$$

$$\frac{\partial D}{\partial \dot{\theta}} = \mu_g \dot{\theta} \tag{3.19}$$

Hence, with (3.7) and (3.8), the dynamics of a TWIP on the slope is derived as follows:

$$\begin{aligned} \left(I_M \eta^2 + M_b l^2 + I_b\right) \ddot{\phi} + \left[M_b l r \cos(\phi + \alpha) - I_M \eta^2\right] \ddot{\theta} - M_b r \dot{\theta} \dot{\phi} l \sin(\phi + \alpha) - M_b g l \sin\phi - \mu_s(\dot{\theta} - \dot{\phi}) \\ = -\eta \tau_i u \end{aligned} \tag{3.20}$$

$$\left[M_b l^2 + M_b r l \cos(\phi + \alpha) + I_b\right] \ddot{\phi} + \left[M_w r^2 + M_b r^2 + M_b r l \cos(\phi + \alpha) + I_w\right] \ddot{\theta} - M_b r \dot{\phi}^2 l \sin(\phi + \alpha)$$

$$+ M_w g r \sin\alpha + M_b g l (r \sin\alpha - \sin\phi) + \mu_g \dot{\theta} = 0 \tag{3.21}$$

The nonlinear model of the TWIP robot can be linearized based on Taylor series approximation

$$f(\phi) = f(\phi_0) + (\phi - \phi_0) f'(\phi_0) + ... + \frac{(\phi - \phi_0)^n}{n!} f^n(\phi_0) \tag{3.22}$$

The overall goal is to keep the robot rod in a vertical position with $\phi = 0$ (or near 0). The linearization might be performed about this point of equilibrium. Under these conditions, $\sin\phi \approx \phi$ and $\cos\phi \approx 1$, the linear model of the two-wheeled inverted pendulum can be yielded from (3.20) and (3.21).

$$\begin{aligned}\left(M_b l^2 + I_b + \eta^2 I_M\right)\ddot{\phi} + \left(M_b r l \cos\alpha - \eta^2 I_M\right)\ddot{\theta} - M_b g l \phi + \mu_s\left(\dot{\phi} - \dot{\theta}\right) \\ = -\eta \tau_i u\end{aligned} \tag{3.23}$$

$$\left(M_b r l \cos\alpha + M_b l^2 + I_b\right)\ddot{\phi} + \left(M_b r^2 + M_w r^2 + M_b r l \cos\alpha + I_w\right)\ddot{\theta} + M_w g r \sin\alpha$$

$$+ M_b g l (r \sin\alpha - \phi) + \mu_g \dot{\theta} = 0 \tag{3.24}$$

When the inclination angle of the slope α is equal to zero, the following nonlinear dynamics of the TWIP on a flat plane can be obtained.

$$\begin{aligned}\left(I_M \eta^2 + M_b l^2 + I_b\right)\ddot{\phi} + \left(M_b l r \cos\phi - I_M \eta^2\right)\ddot{\theta} - M_b r \dot{\theta}\dot{\phi} l \sin\phi - M_b g l \sin\phi - \mu_s(\dot{\theta} - \dot{\phi}) \\ = -\eta \tau_i u\end{aligned} \tag{3.25}$$

$$\begin{aligned}\left(M_b l^2 + M_b r l \cos\phi + I_b\right)\ddot{\phi} + \left(M_w r^2 + M_b r^2 + M_b r l \cos\phi + + I_w\right)\ddot{\theta} - M_b r \dot{\phi}^2 l \sin\phi \\ - M_b g l \sin\phi + \mu_g \dot{\theta} = 0\end{aligned} \tag{3.26}$$

Using the same method as in (3.22), the linear model of the TWIP on a flat plane is given as

$$\left(I_M \eta^2 + M_b l^2 + I_b\right)\ddot{\phi} + \left(M_b l r - I_M \eta^2\right)\ddot{\theta} - M_b g l \phi - \mu_s(\dot{\theta} - \dot{\phi}) = -\eta \tau_i u \tag{3.27}$$

$$\left(M_b l^2 + M_b r l + I_b\right)\ddot{\phi} + \left(M_w r^2 + M_b r^2 + M_b r l + I_w\right)\ddot{\theta} - M_b g l \phi + \mu_g \dot{\theta} = 0 \tag{3.28}$$

3.2 Dynamic Model with 3 DOF

To achieve the trajectory tracking control, another model which is described in a 3 DOF motion dynamic equation is chosen as follow.

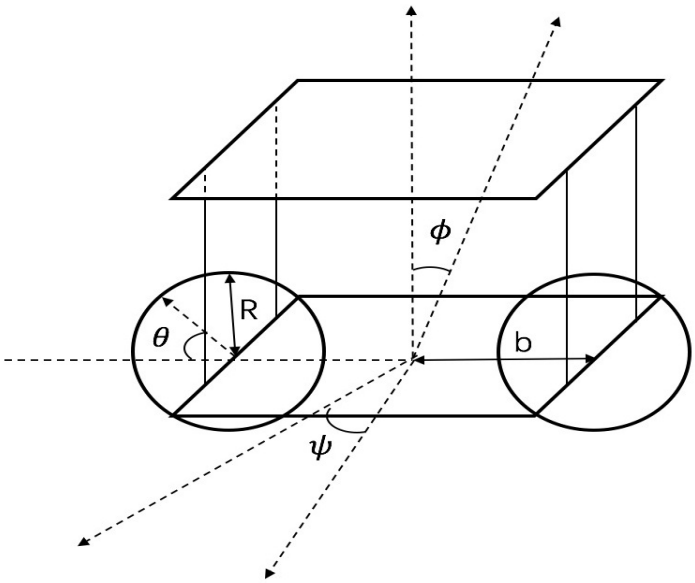


Figure 3.2: Free Body Diagram of the TWIP

Fig 3.2 shows the body diagram of the TWIP whose mathematical model has been established in detail in [6]. In order to briefly recall the dynamic model, Table 3.2 lists all the symbols and their definitions. With the Euler-Lagrange equation, the dynamic model of the inverted pendulum robot has been described in [41].

Define the following notations and the augmented vector $\mathbf{x}$,

$$q_r = \begin{bmatrix} x \\ y \\ \psi \\ \phi \end{bmatrix} \tag{3.29}$$

$$V = \begin{bmatrix} \dot{\phi} \\ v \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_\phi \\ v \\ \omega \end{bmatrix} \tag{3.30}$$

$$\mathbf{x} = \begin{bmatrix} q_r \\ V \end{bmatrix} \tag{3.31}$$

The following model for the TWIP shown in Fig 3.2 is taken from [33].

Table 3.2: Parameters and Variables

Symbol and unit	Parameter and variable name
I_{xx}, I_{yy}, I_{zz}	Moment of inertia of the pendulum body with respect to the x,y,z axis
$v_r[m/\sec]$	reference linear velocity
$\omega_r[m/\sec]$	reference angular velocity
$R[m]$	radius of both wheels
c_x, c_z	the center of mass of the vehicle is at coordinate $OG_b = (c_x, 0, c_z)$ in β
u_r, u_l	input control applied to the right motor and the left motor
I_{wa}, I_{wd}	moment of inertia of a wheel about its axis and about a diameter
$\theta_r, \theta_l[rad]$	angles of the right and left wheel
$\psi[rad]$	yaw angle
$M_b[kg]$	mass of the pendulum body
$M_w[kg]$	mass of the each wheel
$\phi[rad]$	tilt angle of the inverted pendulum
$b[m]$	half of the distance between both driving wheels
$(x, y)[m/\sec]$	position of the inverted pendulum
v, ω	linear velocity and angular velocity of the inverted pendulum

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \tag{3.32}$$

with $f(x)$ and $g(x)$ being defined below.

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}, g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \end{bmatrix}, u = \begin{bmatrix} u_r \\ u_l \end{bmatrix} \tag{3.33}$$

where

$$f_1(x) = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} \tag{3.34}$$

$$f_2(x) = \begin{bmatrix} f_{21}(x) \\ f_{22}(x) \\ f_{23}(x) \end{bmatrix} \tag{3.35}$$

$$g_1(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.36}$$

$$g_2(x) = \begin{bmatrix} g_{21}(x) & g_{21}(x) \\ g_{22}(x) & g_{22}(x) \\ g_{23}(x) & -g_{23}(x) \end{bmatrix} \tag{3.37}$$

$$\begin{aligned} f_{21}(x) &= \frac{1}{D_\alpha}(\sin(2\phi)\dot{\psi}\bar{H}) + \frac{1}{D_\alpha}(M_b^2 c_z^2 R^2 \sin(2\phi)(\phi)^2) \\ &\quad + \frac{1}{2D_\alpha}(-2M_b^2 R^2 c_z - 4I_{wa} M_b c_z - 4M_w R^2 M_b c_z) g \sin\phi \end{aligned} \tag{3.38}$$

$$\begin{aligned} f_{22}(x) &= K_\phi^2 \dot{\psi} + \frac{1}{2D_\alpha}(M_b^2 c_z^2 R^2 g \sin(2\phi)) - \frac{I_{yy} M_b R^2 c_z}{D_\alpha} \\ &\quad - \frac{1}{4D_\alpha} R^2 M_b^2 c_z^3 \sin(\phi)\dot{\phi}^2 \end{aligned} \tag{3.39}$$

$$\begin{aligned} f_{23}(x) &= -\frac{1}{G_\alpha}(I_{xx} - I_{yy}) R^2 \sin(2\phi)\dot{\phi}\dot{\psi} - \frac{1}{G_\alpha} M_b c_z^2 R^2 \sin(2\phi)\dot{\phi}\dot{\psi} \\ &\quad - \frac{1}{G_\alpha}(\sin(\phi) R^2 M_b c_z v \dot{\psi}) \end{aligned} \tag{3.40}$$

$$g_{21}(x) = \frac{1}{D_\alpha}(M_b R^2 + 2M_w R + 2I_{wa} + M_b \cos(\phi) c_z R) \tag{3.41}$$

$$g_{22}(x) = -\frac{R}{D_\alpha}(M_b \cos(\phi) c_z R + I_{yy} + M_b c_z^2) \tag{3.42}$$

$$g_{23}(x) = \frac{Rb}{G_\alpha} \tag{3.43}$$

The detailed expressions for D_α, G_α, K_ϕ and $\bar{H}$ are shown as follows:

$$\begin{aligned} D_\alpha &= M_b^2 \cos^2\phi c_z^2 R^2 + ((-M_b^2 - 2M_w M_b) c_z^2 - 2I_{yy} M_w - I_{yy} M_b) R^2 \\ &\quad - 2M_b c_z^2 I_{wa} - 2I_{yy} I_{wa} \end{aligned} \tag{3.44}$$

$$G_\alpha = (-M_b c_z^2 + I_{zz} - I_{xx}) R^2 \cos^2\phi + \left(M_b c_z^2 + I_{xx} + 2I_{wd}\right) R^2 \tag{3.45}$$

$$+ 2b^2 M_w R^2 + 2b^2 I_{wa} \tag{3.46}$$

$$\begin{aligned} K_\phi &= \frac{(-4I_{yy} M_b R^2 c_z - 3R^2 M_b^2 c_z^3 + M_b R^2 c_z (I_{xx} - I_{yy})) \sin\phi}{4D_\alpha} \\ &\quad + \frac{(M_b R^2 c_z (I_{xx} - I_{zz}) + R^2 M_b^2 c_z^3) \sin 3\phi}{4D_\alpha} \end{aligned} \tag{3.47}$$

$$\begin{aligned}\bar{H} &= \frac{1}{2}M_b R^2 I_{zz} + I_{wa} I_{zz} - M_w R^2 I_{xx} - I_{wa} I_{xx} \\ &\quad -M_b c_z^2 M_w R^2 - M_b c_z^2 I_{wa} - \frac{1}{2}M_b R^2 I_{xx} + M_w R^2 I_{zz}\end{aligned} \tag{3.48}$$

Chapter 4

Controller Design

In this chapter, various control methods are discussed. Under the inspiration of the [2, 19, 29, 43–45], the controllers are mathematically derived.

The proposed control strategies are tested by simulations and then put into real time tests. Both the simulation results and experimental results are being reviewed at the end of each section.

4.1 Velocity and Position Control on a Slope

The purpose of this section is to derive the control law to make the TWIP robot track the desired speed and stop at a given point while keeping balance of its body.

4.1.1 LQR Control

At this stage, LQR control is implemented. The LQR theory has been one of the oldest and most mature state space design methods with the development of modern control theory. It is a powerful technique used in control system design for complex systems.

In practical applications, robots are often required to drive on a non-flat plane, e.g, on the slope. Because of the influence of the angle of the slope, when the inverted pendulum cart becomes stable on this non-flat plane, the pitch angle of the body will not be zero anymore.

In this section, the LQR control strategy for the TWIP robot on a slope is developed firstly. By setting the slope angle to zero, the LQR controller for the TWIP robot on a flat plane can be obtained.

Controller Design

In order to transform (3.23) and (3.24) to the state space representation, define the state variable

$$\mathbf{x} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} \tag{4.1}$$

and define the following matrices

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} M_b l^2 + I_b + \eta^2 I_M & M_b r l \cos\alpha - \eta^2 I_M \\ M_b r l \cos\alpha + M_b l^2 + I_b & (M_b + M_w) r^2 + M_b r l \cos\alpha + I_w \end{bmatrix} \tag{4.2}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -\eta\tau_i \\ 0 \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} -M_b g l & \mu_s & -\mu_s \\ -M_b g l & 0 & \mu_g \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 0 \\ (M_w + M_b l)\, g r \sin\alpha \end{bmatrix} \tag{4.5}$$

Then, it follows from (3.27) and (3.28) that

$$\begin{aligned} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \left(- \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \mathbf{x} - \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \mathbf{u} \right) \\ &= \begin{bmatrix} \bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\ \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \end{bmatrix} + \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \end{bmatrix} \mathbf{u} \end{aligned} \tag{4.6}$$

By differentiating x, taking the equation above into consideration, the following equation can be derived.

$$\dot{\mathbf{x}} = A\mathbf{x} + D + B\mathbf{u} \tag{4.7}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\ \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} \end{bmatrix}, \tag{4.8}$$

$$B = \begin{bmatrix} 0 \\ \bar{b}_1 \\ \bar{b}_2 \end{bmatrix}, \tag{4.9}$$

$$D = \begin{bmatrix} 0 \\ \bar{d}_1 \\ \bar{d}_2 \end{bmatrix} \tag{4.10}$$

By setting $\dot{x} = 0$ and $\dot{\theta} = \dot{\theta}_{ref}$, it follows from (4.7) that the steady state can be determined by

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\ \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} \end{bmatrix} \begin{bmatrix} \phi_s \\ 0 \\ \dot{\theta}_{ref} \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{d}_1 \\ \bar{d}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{b}_1 \\ \bar{b}_2 \end{bmatrix} u_s \tag{4.11}$$

By solving (4.11), it is obtained that ϕ_s and u_s satisfy the following equations.

$$\phi_s = \frac{\left(\bar{a}_{23}\bar{b}_1 - \bar{a}_{13}\bar{b}_2\right)\dot{\theta}_{ref} + \bar{b}_1\bar{d}_2 - \bar{b}_2\bar{d}_1}{\bar{a}_{11}\bar{b}_2 - \bar{a}_{21}\bar{b}_1} \tag{4.12}$$

$$u_s = \frac{\left(\bar{a}_{11}\bar{a}_{23} - \bar{a}_{13}\bar{a}_{21}\right)\dot{\theta}_{ref} + \bar{a}_{11}\bar{d}_2 - \bar{a}_{21}\bar{d}_1}{\bar{a}_{21}\bar{b}_1 - \bar{a}_{11}\bar{b}_2} \tag{4.13}$$

Now, define $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_s$ and $\Delta u = u - u_s$ with

$$\mathbf{x}_s = \begin{bmatrix} \phi_s \\ 0 \\ \dot{\theta}_{ref} \end{bmatrix} \tag{4.14}$$

Then, the following state equation for Δx and Δu can be obtained by differentiating Δx.

$$\Delta\dot{\mathbf{x}} = A\boldsymbol{\Delta}\mathbf{x} + B\boldsymbol{\Delta}\mathbf{u} \tag{4.15}$$

A LQR controller with an integrator is constructed below. In order to introduce integral control to the LQR controller, define a new state variable z as

$$z = \int_0^t (\dot{\theta} - \dot{\theta}_{ref})dt \tag{4.16}$$

The augmented control system is represented as follows:

$$\Delta\dot{\hat{\mathbf{x}}} = \hat{A}\Delta\hat{\mathbf{x}} + \hat{B}\Delta\hat{\mathbf{u}} \tag{4.17}$$

where

$$\boldsymbol{\Delta}\hat{\mathbf{x}} = \begin{bmatrix} \Delta\mathbf{x} \\ z \end{bmatrix}, \Delta\hat{\mathbf{u}} = \Delta\mathbf{u} \tag{4.18}$$

$$\hat{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}, \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \tag{4.19}$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{4.20}$$

The quadratic performance index function with respect to $\Delta\hat{x}$ and $\Delta\hat{u}$ is given as

$$J = \int_0^\infty (\Delta\hat{\mathbf{x}}^T\hat{Q}\Delta\hat{\mathbf{x}} + \Delta\hat{\mathbf{u}}^T\hat{R}\Delta\hat{\mathbf{u}})dt \tag{4.21}$$

where $\hat{Q} = \hat{Q}^T \geq 0$ (positive semi-definite) and $\hat{R} = \hat{R}^T > 0$ (positive definite) are weighting matrices. Linear quadratic regulators consist in finding a state feedback gain $\hat{K}$ such that the cost function J is minimized. $\hat{K}$ can be determined by

$$\hat{K} = \hat{R}^{-1}\hat{B}^T\hat{P} \tag{4.22}$$

where $\hat{P}$ is a symmetric positive definite solution to the following Riccati equation.

$$\hat{P}\hat{A} + \hat{A}^T\hat{P} + \hat{Q} - \hat{P}\hat{B}\hat{R}^{-1}\hat{B}^T\hat{P} = 0 \tag{4.23}$$

As a result, the control input is given by

$$u = u_s - \hat{k}_1(x - x_s) - \hat{k}_2\int_0^t (\dot{\theta} - \dot{\theta}_{ref})dt \tag{4.24}$$

where $\hat{K} = \begin{bmatrix} \hat{k}_1 & \hat{k}_2 \end{bmatrix}$ with $\hat{k}_1$ being a 1x 3 matrix and $\hat{k}_2$ being a 1x1 matrix.

The block diagram of the close-loop system for the velocity and position control is shown in Fig 4.1.

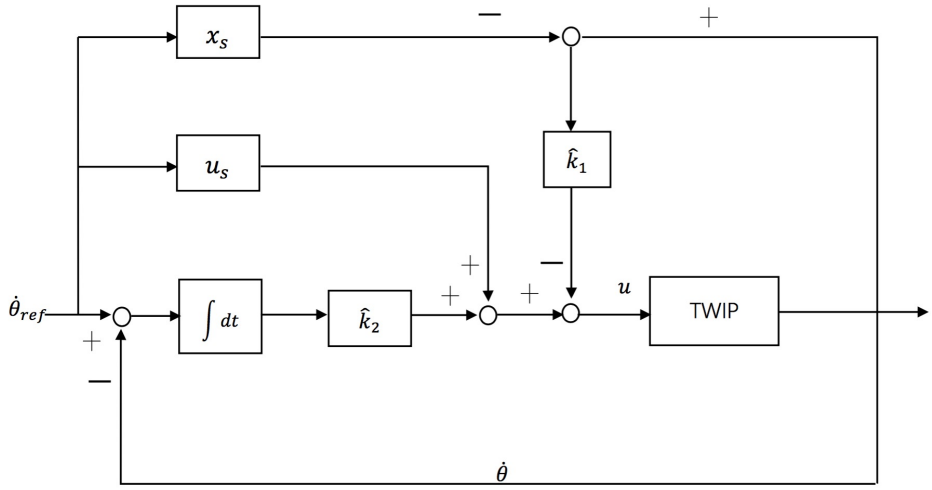


Figure 4.1: Block Diagram for LQR Control

Simulation Results

In this section, the performance of the velocity control and position control will be represented via simulation results. After several times of testing, a couple of relative satisfactory gains have been obtained.

By selecting $\hat{Q}$ and $\hat{R}$ as

$$\hat{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \hat{R} = \begin{bmatrix} 1000 \end{bmatrix} \tag{4.25}$$

the gain matrix $\hat{K}$, as a result, can be calculated as

$$\hat{K} = \begin{bmatrix} -15.2 & -4.5 & -0.35 & -0.32 \end{bmatrix} \tag{4.26}$$

In this simulation, the reference pitch angle, angular velocity of the body and the wheel speed are assumed as $\phi_s \approx 12.5\,\text{deg}$, $\dot{\phi}_s = 0\,\text{deg}/s$ and $\dot{\theta}_{ref} = 250\,\text{deg}/s$.

The simulation results represented in Fig 4.2 indicate the fast convergence in ϕ, $\dot{\phi}$, and $\dot{\theta}$. The pitch angle of the body can be stabilized within 5 seconds, and it takes about 2 seconds for the angular velocity to the desired value. The convergence time of $\dot{\theta}$ is about 7 seconds, meanwhile, $\dot{\theta}$ settles at 250 deg $/s$ which matches the assumption.

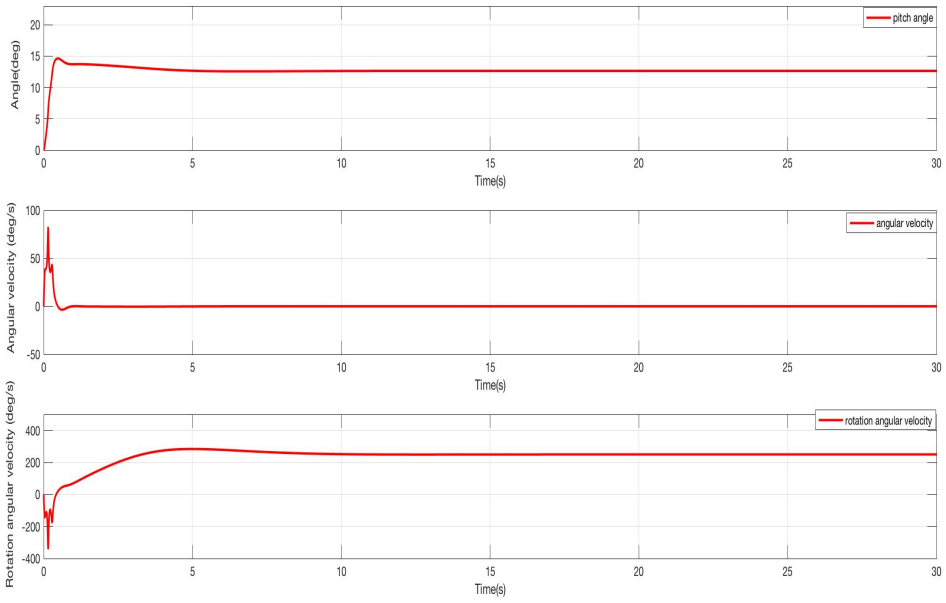


Figure 4.2: Simulation Results of LQR Control for the TWIP on a Slope

Experimental Results

In order to test the real time performance of the proposed LQR controller, several experiments have been done on the real system.

In the experiments, the tilting angle of the slope is 15 degrees. The desired pitch angle of TWIP robot can be obtained from (4.12), $\phi' = 12.5$ degrees. The desired speed for the wheel is set to 250 deg/s and the given distance is set to 0.6 meters. The gain coefficients used here are the same as the ones used in the simulation. At the beginning of the test, the TWIP robot has been running in a straight line at a speed of 250 deg/s while keeping itself upright. When the robot reached the set point of 0.6 meters, it would stop at the given point for 5 seconds, and then, run back to the starting point.

From the experimental results shown in Fig 4.3, the TWIP robot could track the desired speed with small error and travel back and forth between the starting point and the given point, in the meantime, keep the pitch angle about 12.5 degrees.

4.1.2 Fuzzy Adaptive Backstepping Control

Recently, fuzzy logic controllers (FLCs) have been used widely for nonlinear system control since they possess a simple structure and good approximation performance. Apparently, these FLCs have attracted increasingly attention in solving practical complex problems and successfully applied to many nonlinear systems.

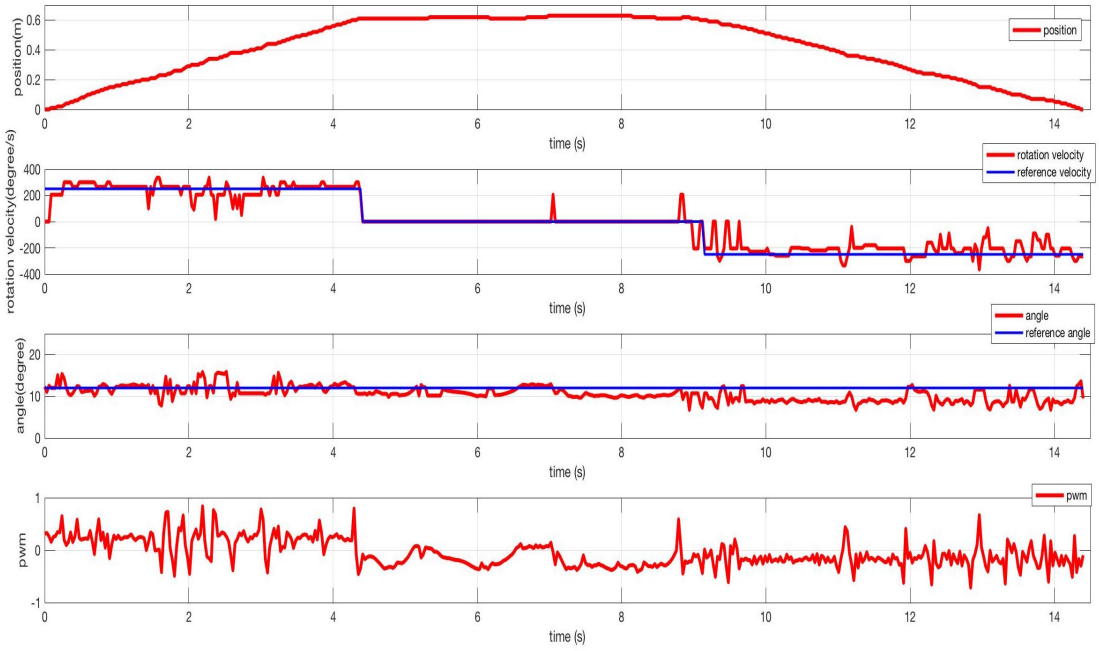


Figure 4.3: Experimental Results of LQR Control of TWIP on a Slope

Controller Design

Motivated by [35, 43, 46–50], a backstepping control with fuzzy adaptive method has been proposed to handle the nonlinearities. The fuzzy adaptive method is proposed to learn the uncertain terms on-line, thus achieving adaptive capability.

A fuzzy logic controller consists of four parts: the knowledge base, the fuzzifier, the fuzzy inference engine, and the defuzzifier. The knowledge base is composed of a collection of fuzzy IF-THEN rules in the following form

$$R^k : \mathbf{IF}\ \mathbf{x_1} \in \mathbf{A_1^k}, \mathbf{x_2} \in \mathbf{A_2^k}, \cdots, \mathbf{x_n} \in \mathbf{A_n^k},\ \mathbf{THEN}\ \mathbf{y} \in \mathbf{B^k} \tag{4.27}$$

where $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$ is the input vector, y is the output variable, $\mu_{A_i^k}(x_i)$ is the fuzzy membership function of fuzzy set A_i^k, $\mu_{B^k}(y)$ is the fuzzy membership function of fuzzy set B^k, $k = 1, 2, \cdots, N$ with N being the number of the IF-THEN rules.

By using singleton fuzzifier, product inference engine, and center average defuzzification, the fuzzy logic system can be formulated as follows:

$$y^* = \frac{\sum_{k=1}^{N} \bar{y}_k [\Pi_{i=1}^{n} \mu_{A_i^k}(x_i)]}{\sum_{k=1}^{N} [\Pi_{i=1}^{n} \mu_{A_i^k}(x_i)]} \tag{4.28}$$

where $\bar{y}_k$ is the maximum value of $\mu_{B^k}(y)$, that is, $\bar{y}_k = \max_{y \in R} \mu_{B^k}(y)$.

Define the fuzzy basic function as

$$\xi_k(x) = \frac{\Pi_{i=1}^{n}\mu_{A_i^k}(x_i)}{\sum_{k=1}^{N}[\Pi_{i=1}^{n}\mu_{A_i^k}(x_i)]},\ k = 1, 2, ..., N \tag{4.29}$$

Then, the fuzzy logic system can be rewritten as

$$f(x) = \hat{S}^T\xi(x) \tag{4.30}$$

where $\hat{S} = \begin{bmatrix} \bar{y}_1 & \bar{y}_2 & \cdots & \bar{y}_N \end{bmatrix}^T$ and $\xi(x) = \begin{bmatrix} \xi_1(x) & \xi_2(x) & \cdots & \xi_N(x) \end{bmatrix}^T$.

In this thesis, the following trapezoidal membership function is used to reduce the computation burden.

$$\mu(x; a, b, c, d) = \begin{cases} 0, & x < a \ or \ x > d \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \end{cases} \tag{4.31}$$

It follows from (3.20) and (3.21) that the nonlinear model of the system can be rewritten as

$$\begin{aligned} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \left(- \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u \right) \\ &= - \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \end{bmatrix} u \qquad (4.32) \\ &= F + Gu \qquad (4.33) \end{aligned}$$

where

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} I_M\eta^2 + M_b l^2 + I_b & M_b rl\cos(\phi+\alpha) - I_M\eta^2 \\ M_b l^2 + M_b rl\cos(\phi+\alpha) + I_b & (M_b + M_w)r^2 + M_b rl\cos(\phi+\alpha) + I_w \end{bmatrix} \tag{4.34}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -\eta\tau_i \\ 0 \end{bmatrix} \tag{4.35}$$

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} -M_b r\dot{\theta}\dot{\phi} l\sin(\phi+\alpha) - M_b gl\sin\phi - \mu_s(\dot{\theta} - \dot{\phi}) \\ -M_b lr\dot{\phi}^2\sin(\phi+\alpha) + M_w gr\sin\alpha + M_b gl\,(r\sin\alpha - \sin\phi) + \mu_g\dot{\theta} \end{bmatrix} \tag{4.36}$$

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = - \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \end{bmatrix}$$

Firstly, considering the angle of the body produces

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ G_1 u + F_1 \end{pmatrix} \tag{4.37}$$

where

$$x_1 = \phi,\ x_2 = \dot{\phi}$$

Here, c_1 and c_2 are considered as unknown, so the nonlinear function F is assumed to be unknown which can be estimated by the following fuzzy logic system

$$F_1 = \hat{S}^T \xi + \varepsilon \tag{4.38}$$

where ε is the approximation error, which is bounded by a positive constant $\varepsilon_{\max}$, that is $\varepsilon \leq \varepsilon_{\max}$.

In this thesis,

$$\xi = [\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \xi_7, \xi_8, \xi_9]^T \tag{4.39}$$

where

$$\xi_1 = \frac{\mu_N(\phi)\mu_N(\dot{\phi})}{D}, \xi_2 = \frac{\mu_N(\phi)\mu_Z(\dot{\phi})}{D}, \xi_3 = \frac{\mu_N(\phi)\mu_P(\dot{\phi})}{D}$$

$$\xi_4 = \frac{\mu_Z(\phi)\mu_N(\dot{\phi})}{D}, \xi_5 = \frac{\mu_Z(\phi)\mu_Z(\dot{\phi})}{D}, \xi_6 = \frac{\mu_Z(\phi)\mu_P(\dot{\phi})}{D}$$

$$\xi_7 = \frac{\mu_P(\phi)\mu_N(\dot{\phi})}{D}, \xi_8 = \frac{\mu_P(\phi)\mu_Z(\dot{\phi})}{D}, \xi_9 = \frac{\mu_P(\phi)\mu_P(\dot{\phi})}{D}$$

and

$$\begin{aligned} D = {} & \mu_N(\phi)\mu_N(\dot{\phi}) + \mu_N(\phi)\mu_Z(\dot{\phi}) + \mu_N(\phi)\mu_P(\dot{\phi}) \\ & + \mu_Z(\phi)\mu_N(\dot{\phi}) + \mu_Z(\phi)\mu_Z(\dot{\phi}) + \mu_Z(\phi)\mu_P(\dot{\phi}) \\ & + \mu_P(\phi)\mu_N(\dot{\phi}) + \mu_P(\phi)\mu_Z(\dot{\phi}) + \mu_P(\phi)\mu_P(\dot{\phi}) \end{aligned}$$

The fuzzy sets N,Z,P are defined for ϕ and $\dot{\phi}$ as shown in Fig 4.4 and Fig 4.5.

The fuzzy rule base for the nonlinear term is given as follows:

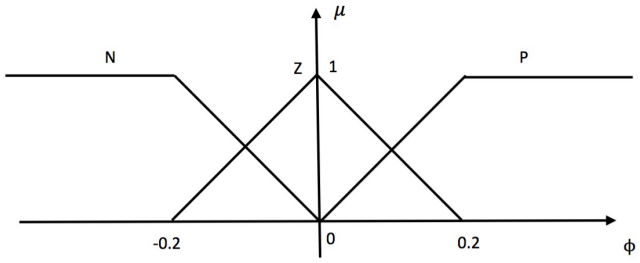


Figure 4.4: The Fuzzy Set for ϕ

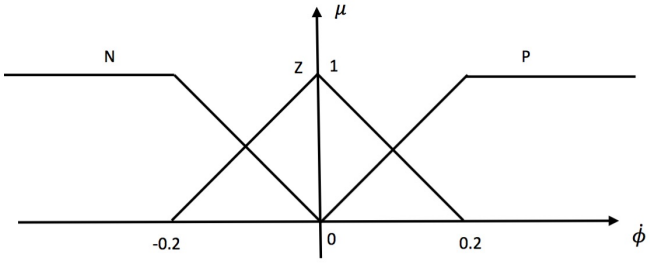


Figure 4.5: The Fuzzy Set for $\dot{\phi}$

$$
\begin{aligned}
R^1 &: \textbf{IF } \phi \in \mathbf{N}, \dot{\phi} \in \mathbf{N}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^1} \\
R^2 &: \textbf{IF } \phi \in \mathbf{N}, \dot{\phi} \in \mathbf{Z}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^2} \\
R^3 &: \textbf{IF } \phi \in \mathbf{N}, \dot{\phi} \in \mathbf{P}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^3} \\
R^4 &: \textbf{IF } \phi \in \mathbf{Z}, \dot{\phi} \in \mathbf{N}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^4} \\
R^5 &: \textbf{IF } \phi \in \mathbf{Z}, \dot{\phi} \in \mathbf{Z}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^5} \\
R^6 &: \textbf{IF } \phi \in \mathbf{Z}, \dot{\phi} \in \mathbf{P}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^6} \\
R^7 &: \textbf{IF } \phi \in \mathbf{P}, \dot{\phi} \in \mathbf{N}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^7} \\
R^8 &: \textbf{IF } \phi \in \mathbf{P}, \dot{\phi} \in \mathbf{Z}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^8} \\
R^9 &: \textbf{IF } \phi \in \mathbf{P}, \dot{\phi} \in \mathbf{P}, \textbf{ THEN } \mathbf{y} \in \mathbf{B^9}
\end{aligned}
$$

Then, with (4.38), (4.37) can be rewritten as

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ G_1 u + \hat{S}^T \xi + \varepsilon \end{bmatrix} \tag{4.40}
$$

The block diagram of the adaptive fuzzy logic estimator is shown in Fig 4.6.

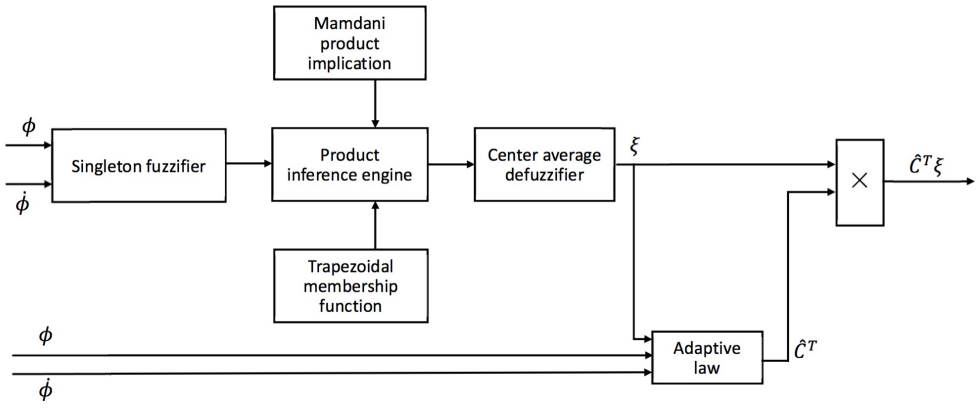


Figure 4.6: Fuzzy Logic Estimator

By using backstepping, a fuzzy adaptive controller is designed as follows.

Step 1

Define the tracking error for pitch angle

$$e_1 = x_1 - \phi_s \tag{4.41}$$

where ϕ_s is the steady state for the pitch angle. By setting $\dot{\phi} = 0$, $\ddot{\phi} = 0$, and $\ddot{\theta} = 0$, it follows from (3.21) that ϕ_s can be calculated as

$$\phi_s = \arcsin \frac{M_w g r \sin\alpha + M_b g l r \sin\alpha + \mu_g \dot{\theta}_{ref}}{M_b g l} \tag{4.42}$$

The following positive definite Lyapunov function is introduced.

$$V_1 = \frac{1}{2} e_1^2 \tag{4.43}$$

Differentiating it with respect to time gives

$$\dot{V}_1 = e_1 x_2 = -k_1 e_1^2 + e_1 (x_2 - \alpha_1) \tag{4.44}$$

where α_1 is a virtual control defined by

$$\alpha_1 = -k_1 e_1 \tag{4.45}$$

with k_1 being a positive gain.

Define $z_2 = x_2 - \alpha_1$, then

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha}_1 = G_1 u + \hat{S}^T \xi + \varepsilon + k_1 x_2 \tag{4.46}$$

Step 2

Define a positive Lyapunov function V_2

$$V_2 = V_1 + \frac{1}{2} z_2^2 + \frac{1}{2} \tilde{S}^T \Gamma \tilde{S} \tag{4.47}$$

where $\tilde{S} = \hat{S} - S$, and Γ is a positive definite matrix.

The derivative of V_2 with respect to time becomes

$$\begin{aligned}
\dot{V}_2 &= -k_1 e_1^2 + e_1 z_2 + z_2 \dot{z}_2 - \tilde{S}^T \Gamma \dot{S} \\
&= -k_1 e_1^2 + z_2 (e_1 + G_1 u + \hat{S}^T \xi + \varepsilon + k_1 x_2) - \tilde{S}^T \Gamma \dot{S} \\
&= -k_1 e_1^2 + z_2 (e_1 + G_1 u + S^T \xi + \varepsilon + k_1 x_2) - \tilde{S}^T (\Gamma \dot{S} - z_2 \xi) \\
&= -k_1 e_1^2 + z_2 (e_1 + G_1 u + S^T \xi + k_1 x_2) - \tilde{S}^T (\Gamma \dot{S} - z_2 \xi) + z_2 \varepsilon
\end{aligned} \tag{4.48}$$

The following adaptation law is introduced

$$\dot{S} = \Gamma^{-1} z_2 \xi - \sigma S \tag{4.49}$$

where σ is a positive constant.

Then, (4.48) can be rewritten as

$$\begin{aligned}
\dot{V}_2 &= -k_1 e_1^2 + z_2 (e_1 + G_1 u + S^T \xi + k_1 x_2) + \tilde{S}^T \sigma S + z_2 \varepsilon \\
&= -k_1 e_1^2 + z_2 (e_1 + G_1 u + S^T \xi + k_1 x_2) + \tilde{S}^T \sigma \hat{S} - \tilde{S}^T \sigma \tilde{S} + z_2 \varepsilon
\end{aligned} \tag{4.50}$$

Using Young's inequality $x^T y \leq \frac{1}{2} x^T x + \frac{1}{2} y^T y$ gives

$$\tilde{S}^T \sigma \hat{S} \leq \frac{1}{2} \tilde{S}^T \sigma \tilde{S} + \frac{1}{2} \hat{S}^T \sigma \hat{S} \tag{4.51}$$

and

$$z_2\varepsilon \leq \frac{1}{2}z_2^2 + \frac{1}{2}\varepsilon^2 \tag{4.52}$$

Substituting (4.51) and (4.52) into (4.50) yields

$$\begin{aligned}\dot{V}_2 \leq & -k_1e_1^2 + z_2(e_1 + G_1u + S^T\xi + k_1x_2) - \frac{1}{2}\tilde{S}^T\sigma\tilde{S} \\ & + \frac{1}{2}\hat{S}^T\sigma\hat{S} + \frac{1}{2}z_2^2 + \frac{1}{2}\varepsilon^2\end{aligned} \tag{4.53}$$

To get the control law, set

$$\begin{aligned}u &= -\frac{1}{G_1}(e_1 + S^T\xi + k_1x_2 + \frac{1}{2}z_2 + k_2z_2) \\ &= -\frac{1}{G_1}\left[e_1 + S^T\xi + k_1x_2 + \frac{1}{2}(x_2 + k_1e_1) + k_2(x_2 + k_1x_1)\right]\end{aligned} \tag{4.54}$$

Using (4.54), (4.53) can be simplified as

$$\begin{aligned}\dot{V}_2 &\leq -k_1e_1^2 - k_2z_2^2 - \frac{1}{2}\tilde{S}^T\sigma\tilde{S} + \frac{1}{2}\hat{S}^T\sigma\hat{S} + \frac{1}{2}\varepsilon^2 \\ &\leq -aV_2 + b\end{aligned} \tag{4.55}$$

where

$$a = \min\{2k_1, 2k_2, \Gamma^{-1}\sigma\}$$

$$b = \frac{1}{2}\hat{S}^T\sigma\hat{S} + \frac{1}{2}\varepsilon^2$$

As a result, all the signals in V_2 are bounded [51]. The detailed discussion for this result is given in [52]. Therefore, the error terms in V_2 are bounded.

Since the adaptive backstepping controller is derived for the body angle, the controller is needed to stabilize the wheel angle. Motivated by [19], a PD-type controller is employed to complete the velocity control of the TWIP robot.

The whole adaptive controller is given as follows:

$$u = -\frac{1}{G_1}\left[e_1 + S^T\xi + k_1 x_2 + \frac{1}{2}(x_2 + k_1 e_1) + k_2(x_2 + k_1 e_1)\right] + k_p e(t) + k_d \frac{de(t)}{dt} \tag{4.56}$$

where $e(t) = \theta_d - \theta(t)$.

Simulation Result

The simulation results for the fuzzy adaptive backstepping with PD control on a slope are shown in Fig 4.7. The gain coefficients of k_1, k_2, k_p, k_d used for the simulation are found by the trail and error method. The values are $k_1 = 10, k_2 = 13, k_p = -0.01, k_d = -0.09$. In this simulation, the initial tilting angle is set to be $\phi(0) = 0\,\text{deg}$. The desired pitch angle of the robot is set to 12.5 degrees, the desired angular velocity of the wheel $\dot{\theta}_{ref}$ is set to be $180\,\text{deg}/s$, respectively. For the TWIP robot to run at reference speed, the pitch angle and angular velocity displacement stabilization takes approximately 1.5 seconds. For speed stabilization, it takes 5 seconds for the wheeled inverted pendulum to come to the desired angular velocity.

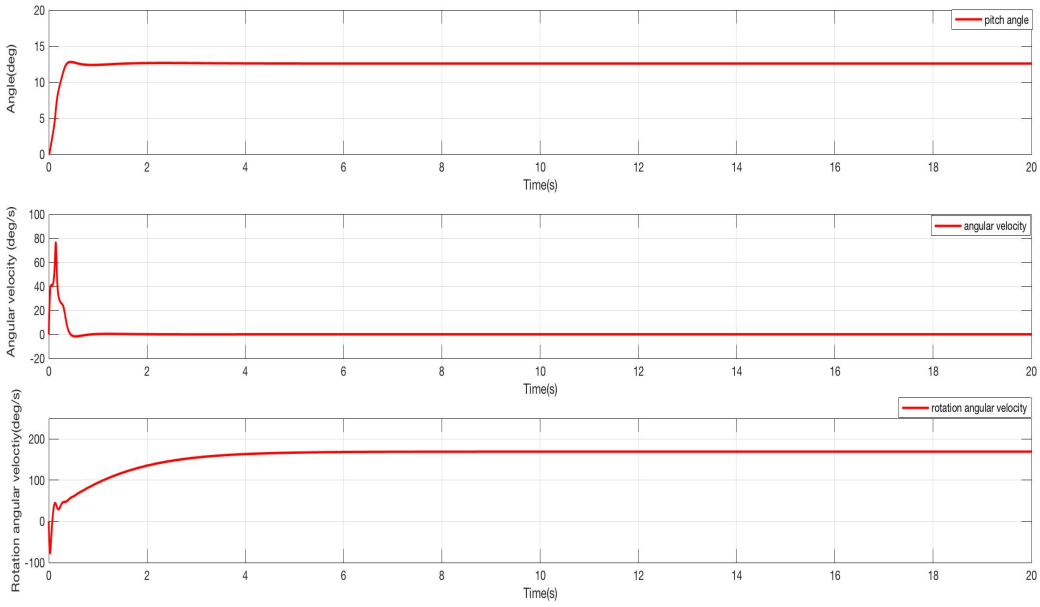


Figure 4.7: Simulation Results of Fuzzy Adaptive with PD Control on a Slope

Experimental Results

In this experiment, a test of travelling back and forth on the slope with angle of 15 degrees has been conducted. The gain coefficients are $k_1 = 10, k_2 = 13, k_p = -0.01, k_d = -0.09$, which are the same as the ones used in the simulation. The experimental results shown in Fig 4.8 prove a functional control scheme for the inverted pendulum robot. The desired speed is 180 deg/s and the given position to stop is 0.75 meters away from the starting point. From the graphs, the pitch movements are controlled within 5 degrees and the angular velocity is pretty much locked onto the desired speed.

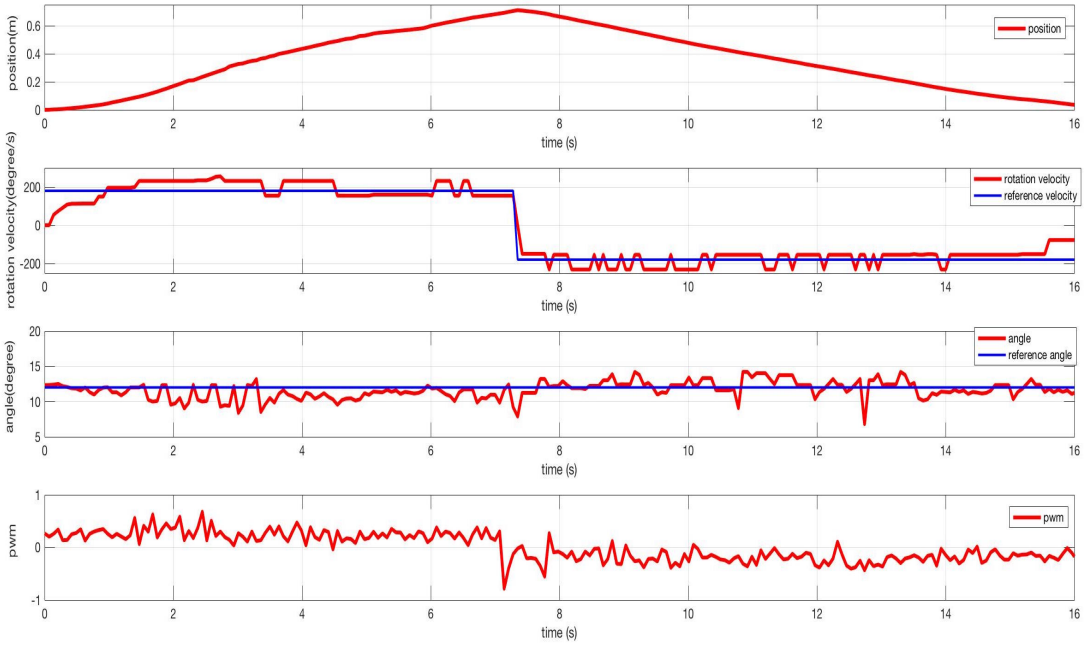


Figure 4.8: Experimental Results of Fuzzy Adaptive Control of TWIP on a Slope

4.2 Velocity and Position Control on a Flat Plane

4.2.1 Controller Design

When the slope angle α is equal to zero, the problem can be treated as controller design on a flat plane.

LQR Control

Substituting $\alpha = 0$ into (4.2) yields

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} M_b l^2 + I_b + \eta^2 I_M & M_b rl - \eta^2 I_M \\ M_b rl + M_b l^2 + I_b & (M_b + M_w) r^2 + M_b rl + I_w \end{bmatrix} \tag{4.57}$$

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.58}$$

Because of the slope angle α is changed, it is obvious that ϕ_s and u_s will be changed.

$$\phi_s = \frac{\left(\bar{a}_{23}\bar{b}_1 - \bar{a}_{13}\bar{b}_2\right)\dot{\theta}_{ref} + \bar{b}_1\bar{d}_2 - \bar{b}_2\bar{d}_1}{\bar{a}_{11}\bar{b}_2 - \bar{a}_{21}\bar{b}_1} \tag{4.59}$$

$$u_s = \frac{\left(\bar{a}_{11}\bar{a}_{23} - \bar{a}_{13}\bar{a}_{21}\right)\dot{\theta}_{ref} + \bar{a}_{11}\bar{d}_2 - \bar{a}_{21}\bar{d}_1}{\bar{a}_{21}\bar{b}_1 - \bar{a}_{11}\bar{b}_2} \tag{4.60}$$

Then, the output of the proposed LQR controller (4.24) can be updated as

$$u = u_s - \hat{k}_1(x - x_s) - \hat{k}_2 \int_0^t (\dot{\theta} - \dot{\theta}_{ref})dt \tag{4.61}$$

Simulation Results

The simulation results for LQR control of the TWIP robot on a flat plane are shown in Fig 4.9.

By selecting $\hat{Q}$ and $\hat{R}$ as

$$\hat{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \hat{R} = 480$$

$\hat{K}$ as a result, can be calculated as

$$\hat{K} = \begin{bmatrix} -11 & -1.7 & -0.05 & -0.045 \end{bmatrix}$$

The reference tilting angle, angular velocity of the body and the wheel speed are assumed as $\phi_s \approx 0\,\text{deg}$, $\dot{\phi}_s = 0\,\text{deg}/s$ and $\dot{\theta}_{ref} = 180\,\text{deg}/s$. The simulation results represented in Fig 4.9 indicate the fast convergence in ϕ, $\dot{\phi}$, and $\dot{\theta}$. Both the angle and the angular velocity of the body can be stabilized within 7.5 seconds. And the convergence time of $\dot{\theta}$ is about 8 seconds, meanwhile, $\dot{\theta}$ settles at 180 deg $/s$ which matches the assumption.

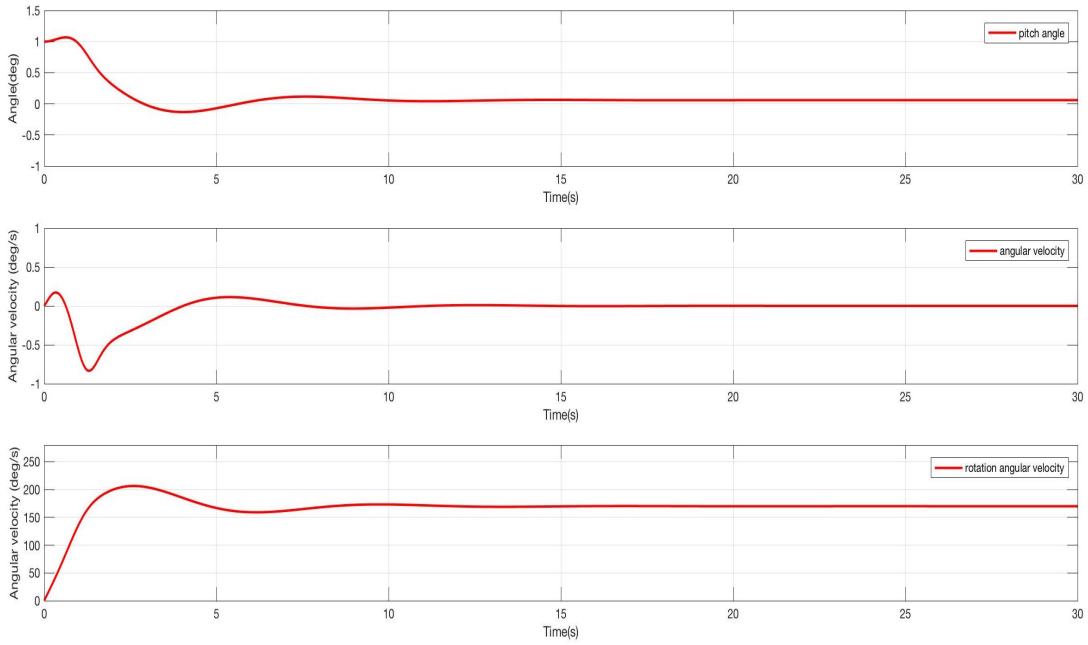


Figure 4.9: Simulation Results of LQR Control For the TWIP

Experimental Results

In this experiment, the TWIP robot was placed on the ground. The desired rotation velocity of the wheel is set to 180 deg/s and the given distance is set to 1 meter. The gain coefficients used here are the same as the ones used in the simulation. At the beginning of the test, the robot has been running in a straight line at a speed about 180 deg/s while keeping itself upright. After the robot travelled the given distance of 1 meter, the reference speed was reversed to -180 deg/s, making the inverted pendulum cart return to the starting point and keep stable. From the experimental results shown in Fig 4.10, the TWIP robot could track the desired speed with small error and travel back and forth between the starting point and the given point, in the meantime, keep the tilting angle about 0.

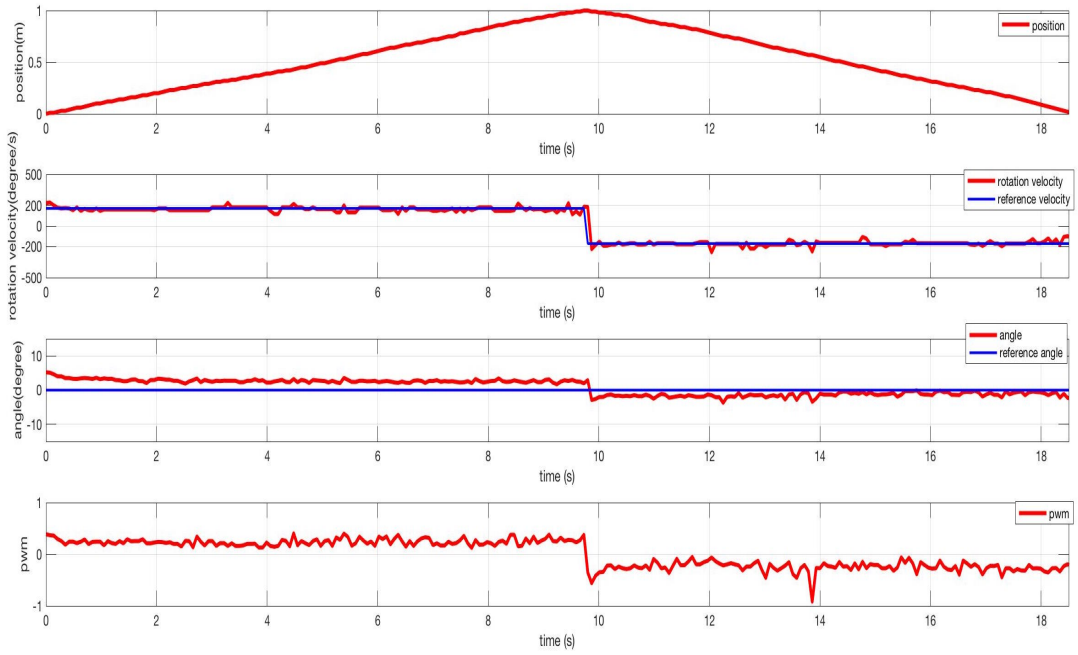


Figure 4.10: Experimental Results of LQR Control for the TWIP

Fuzzy Adaptive Control

When $\alpha = 0$, from (4.42), ϕ_s can be derived as

$$\phi_s = \arcsin \frac{\mu_g \dot{\theta}_{ref}}{M_b g l} \tag{4.62}$$

The tracking error is updated as

$$e_1 = x_1 - \phi_s$$

Using the similar method, the fuzzy adaptive control law can be developed as below.

$$\begin{aligned} u = -\frac{1}{G_1} \left[e_1 + \hat{C}^T \xi + k_1 x_2 + \frac{1}{2}(x_2 + k_1 e_1) + k_2 (x_2 + k_1 e_1) \right] \\ + k_p e(t) + k_d \frac{e(t)}{dt} \end{aligned} \tag{4.63}$$

Simulation Results

For the fuzzy adaptive backstepping with PD control on a flat plane, the gain coefficients are $k_1 = 30, k_2 = 33, k_p = -0.1, k_d = -0.9$, which are found by the trial and error method. In the simulation, the reference pitch angle and angular velocity of the body are both set to 0, and the angular velocity of the wheel is set to 180 deg/s. The results from Fig 4.11 indicates the fast convergence in $\phi, \dot{\phi}$ and $\dot{\theta}$.

4.2.2 Experimental Results

In this experiment, a similar back and forth test has been conducted on a flat plane for the fuzzy adaptive backstepping controller. The experimental results shown in Fig. 4.12 prove a functional control scheme for the inverted pendulum robot. The desired speed is 180 deg/s and the given position to stop is 1 meter. The gain coeffcients are $k_1 = 30, k_2 = 33, k_p = -0.1, k_d = -0.9$. From the graphs, the pitch movements are controlled within 5 degrees and the rotation angular velocity is pretty much locked onto the desired speed.

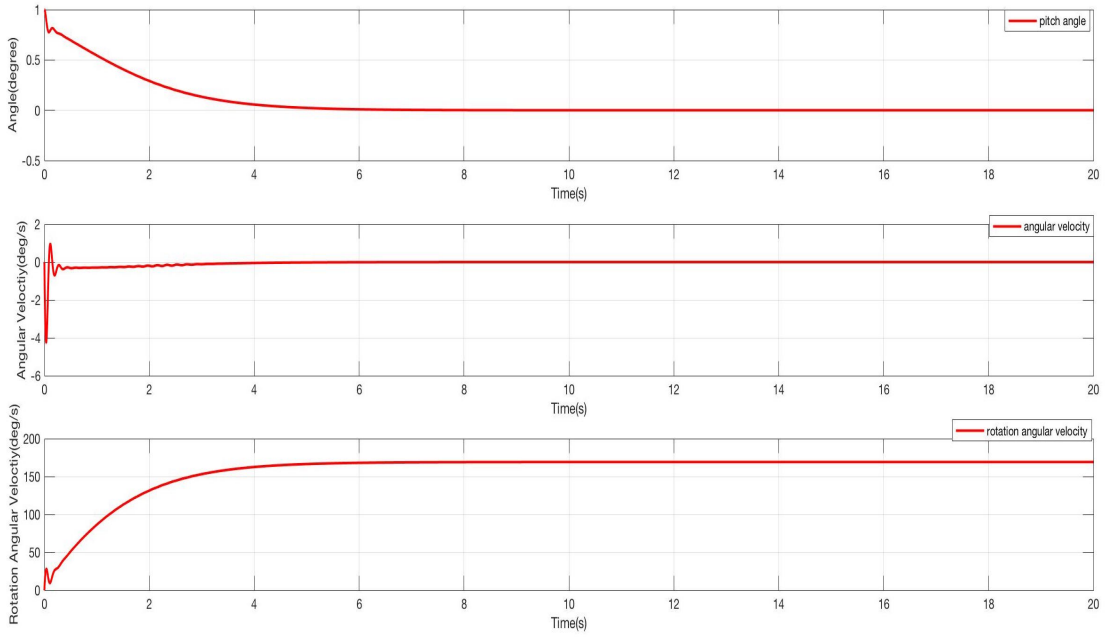


Figure 4.11: Simulation Results of Fuzzy Adaptive with PD Control

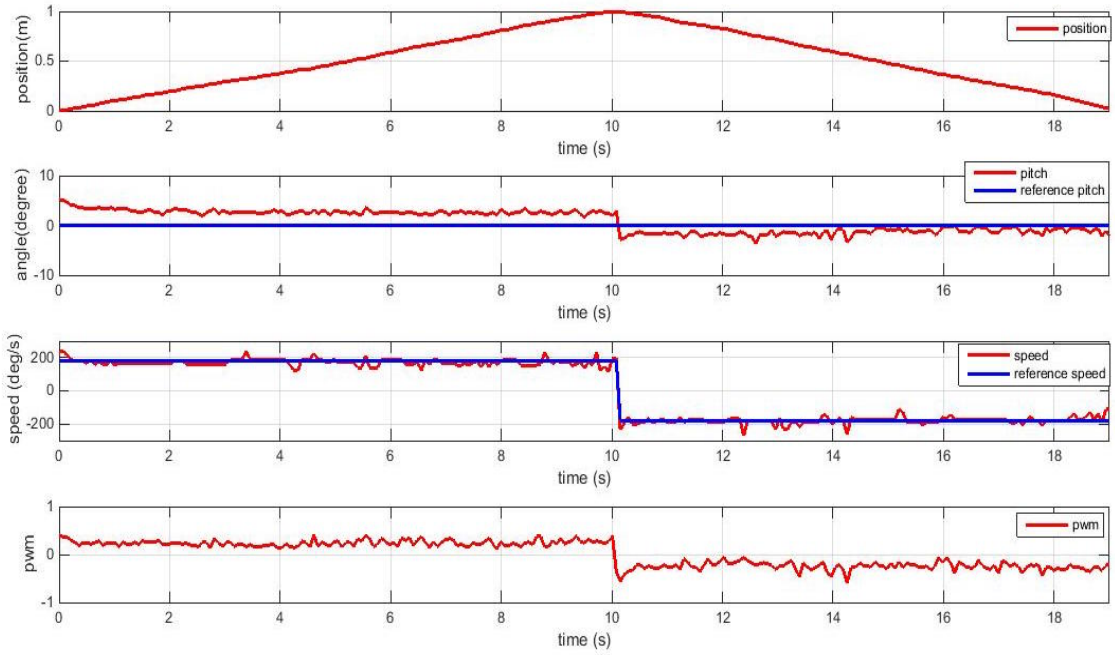


Figure 4.12: Experimental Results of Fuzzy Adaptive with PD Control

4.3 Trajectory Tracking Control

This section presents the control scheme of the wheeled inverted pendulum robot to perform a trajectory tracking control task.

4.3.1 PID Control

Controller Design

The goal of the wheeled mobile robot is to control the pitch angle ϕ to be upright while the robot tracking the desired trajectory.

The center of the mass of the TWIP robot is assumed to be located on the center of the wheel axis, and the following relationship can be obtained

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4.64}$$

where ψ is the yaw angle of the TWIP robot and x, y are the coordinates of the position. The relationship between the Cartesian velocities and wheel angular velocities is shown as below.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} \tag{4.65}$$

where R is the radius of the wheel and L is the distance between two wheels. $\dot{\theta}_R$ is the angular velocity of the right wheel and $\dot{\theta}_L$ is the angular velocity of the left wheel. Combining (4.80) and (4.81) yields

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}\cos\psi & \frac{R}{2}\cos\psi \\ \frac{R}{2}\sin\psi & \frac{R}{2}\sin\psi \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} \tag{4.66}$$

By the inspiration of the [29], the following PID controller is derived based on (4.66).

$$u_\phi = k_{p\phi}\phi + k_{d\phi}\dot{\phi} \tag{4.67}$$

$$u_v = k_{pv}e_v + k_{iv}\int e_v dt \tag{4.68}$$

$$u_\psi = k_{p\psi} e_\psi + k_{i\psi} \int e_\psi dt + k_{d\psi} \frac{e_\psi}{dt} \tag{4.69}$$

where

$e_v = v_d - v_{car}$, $e_\psi = \psi_d - \psi_{car}$, v_d is the desired speed of the wheel, v_{car} is the actual wheel speed, ψ_d is the desired heading angle, and ψ_{car} is the actual heading angle.

Then, the control inputs of the right and left wheels are given by

$$u_R = u_\phi + u_v + u_\psi$$

$$u_L = u_\phi + u_v - u_\psi \tag{4.70}$$

Fig 4.13 shows the block diagram for PID control .

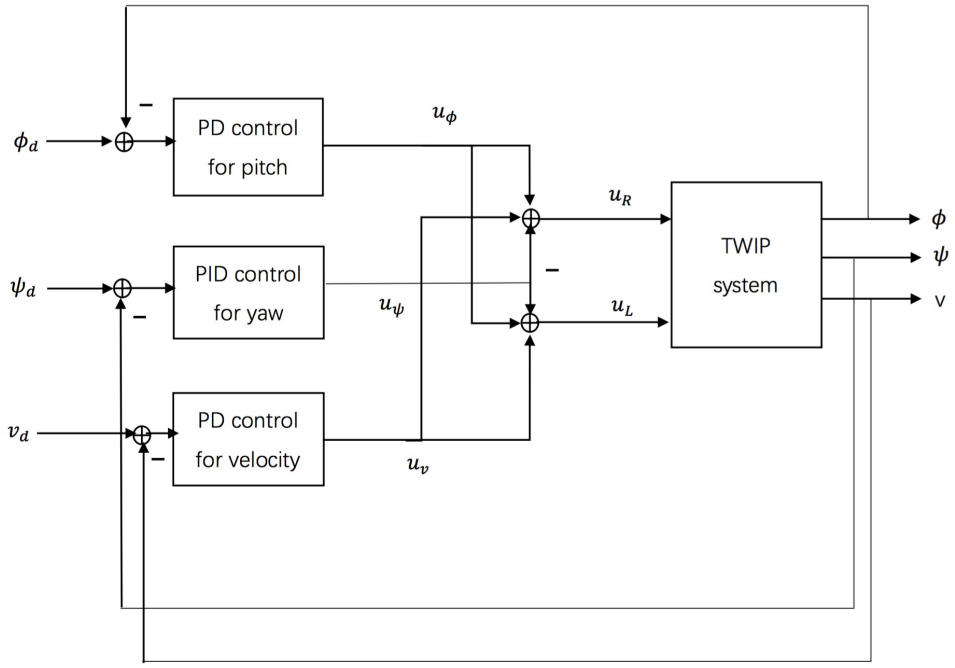


Figure 4.13: Block Diagram for PID Control

Experimental Results

The experimental results shown in Fig 4.14 and Fig 4.15 prove a functional control scheme for the inverted pendulum robot. Controller gains listed in Table 4.1 are used for experimental studies, which are found by trial and error. The reference trajectory is a circle with a radius of 0.6 meters and the desired speed is 180 deg/s. It takes about 36 seconds for the TWIP robot to complete the given circular trajectory. The balancing angle is well maintained within ±5 degrees as shown.

Table 4.1: Controller Gains

	P	D	I
Angle	25.0	3.0	0.0
Velocity	-25.0	0.0	-0.005
Yaw	-8.0	-0.03	-0.001

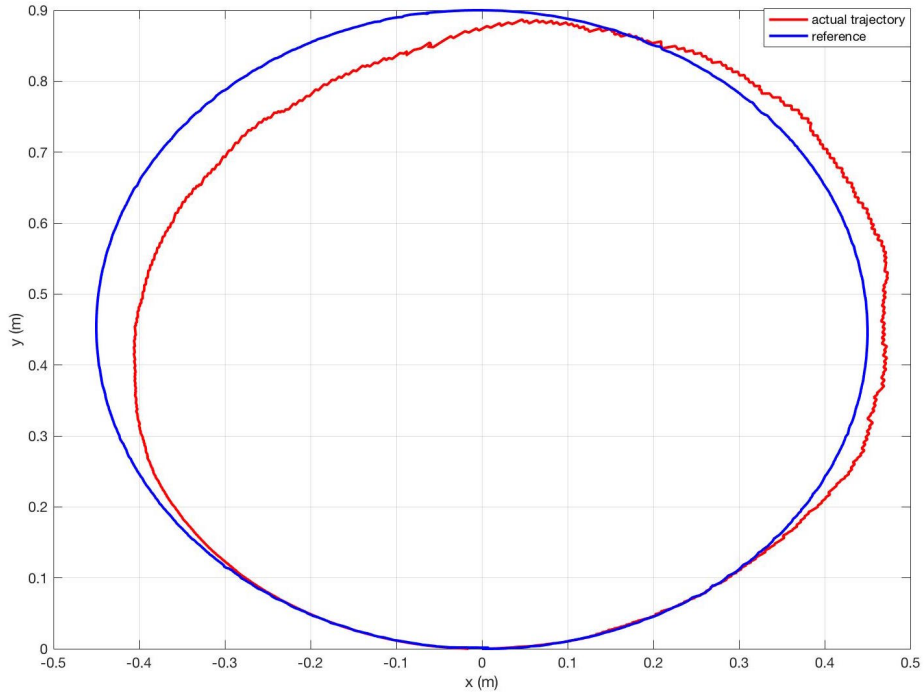


Figure 4.14: Circular Trajectory Tracking for PID Control

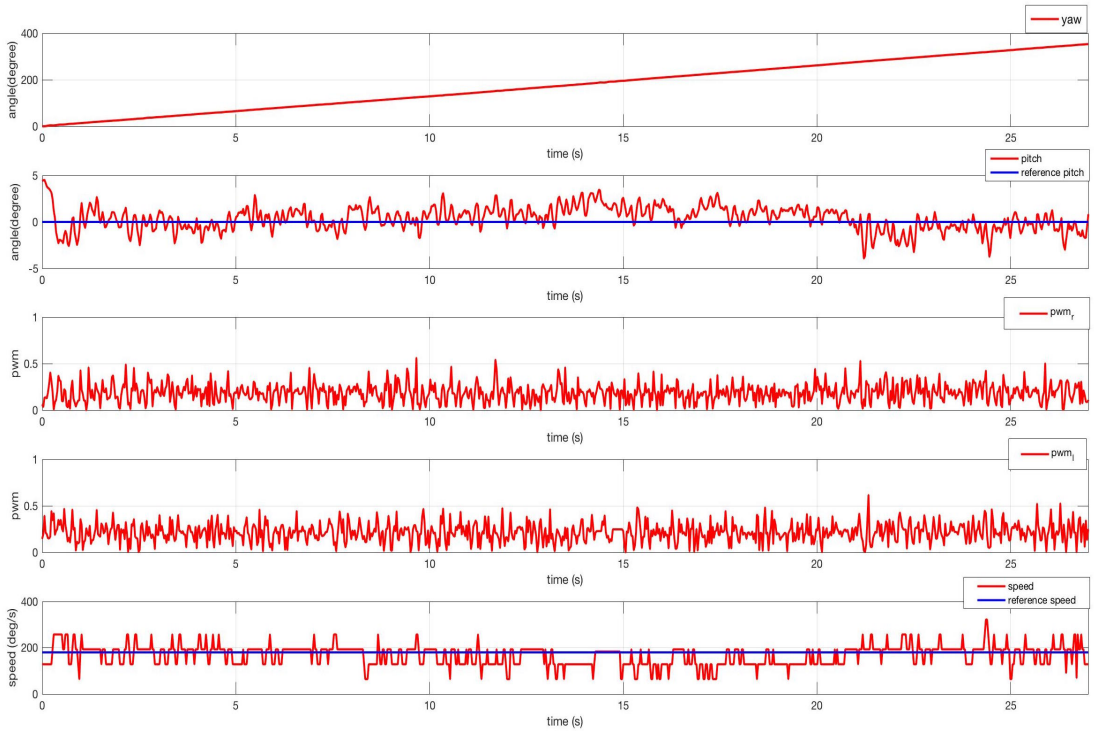


Figure 4.15: Experimental Results of PID Control

4.3.2 Sliding Mode Control with Fuzzy Adaptive Backstepping Method

This section presents a sliding mode controller with fuzzy adaptive backstepping method for trajectory tracking of a TWIP robot. A decoupling approach is proposed to decouple the dynamic model of the TWIP robot such that the controller can be synthesized using sliding mode and backstepping control in both kinematic and dynamic levels. The fuzzy adaptive method is proposed to deal with the uncertain terms.

Sliding Mode Backstepping Controller Design

With the transformation of the torques C_ψ and C_y into the wheel torques u_r and u_l, it can be obtained that

$$\mathbf{u} = \begin{bmatrix} u_r \\ u_l \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} C_\psi \\ C_y \end{bmatrix} \tag{4.71}$$

$$\begin{aligned} \dot{\mathbf{V}} &= \begin{bmatrix} \dot{\omega}_\phi \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = f_2(x) + g_2(x) \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} C_\psi \\ C_y \end{bmatrix} \\ &= \begin{bmatrix} f_{21}(x) \\ f_{22}(x) \\ f_{23}(x) \end{bmatrix} + \begin{bmatrix} g_{21}(x) & 0 \\ g_{22}(x) & 0 \\ 0 & g_{23}(x) \end{bmatrix} \begin{bmatrix} C_\psi \\ C_y \end{bmatrix} \end{aligned} \tag{4.72}$$

To formulate the problem, let $\tilde{x}(t)$ and $\tilde{y}(t)$ and $\tilde{\psi}(t)$ be the differences between the real position $x(t)$, $y(t)$ and $\psi(t)$ and the reference trajectory.

$$\tilde{x}(t) = x_r(t) - x(t), \tilde{y}(t) = y_r(t) - y(t), \tilde{\psi}(t) = \psi_r(t) - \psi(t) \tag{4.73}$$

From (3.30), the linear velocity $v(t)$, the angular velocity $\omega(t)$ and the angular velocity of the pitch angle $\omega_\phi(t)$ are regarded as virtual control. On one hand, in order to stabilize the pitch angle of the body, it is easy to design the virtual control.

$$\omega_\phi(t) = \lambda_1(\phi) = -k_\phi \phi \tag{4.74}$$

On the other hand

$$\begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\psi} \end{bmatrix} \tag{4.75}$$

where $e_1(t)$ is the tangential error, $e_2(t)$ is the normal error and $e_3(t)$ is the orientation error. Differentiating the error vector produces

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 - v + v_r \cos e_3 \\ -\omega e_1 + v_r \sin e_3 \\ \omega_r - \omega \end{bmatrix} \tag{4.76}$$

Define a new auxiliary variable $\bar{e}_3(t)$

$$\bar{e}_3(t) = e_3 + k_3 e_2 \tag{4.77}$$

Taking the time derivative of $\bar{e}_3(t)$ yields

$$\dot{\bar{e}}_3 = \dot{e}_3 + k_3 \dot{e}_2 = \omega_r - \omega + k_3(-\omega e_1 + v_r \sin e_3) \tag{4.78}$$

Choose the following Lyapunov function

$$V_1 = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{2}\bar{e}_3^2 + \frac{1}{2}\phi^2 \tag{4.79}$$

The derivative of V_1 can be expressed as

$$\begin{aligned} \dot{V}_1 &= e_1\dot{e}_1 + \dot{e}_2 e_2 + \bar{e}_3\dot{\bar{e}}_3 + \phi\omega_\phi \\ &= e_1\left(\omega e_2 - v + v_r\cos e_3\right) + e_2\left(-we_1 + v_r \sin e_3\right) \\ &+ \bar{e}_3\left[\omega_r - \omega + k_3\left(-\omega e_1 + v_r \sin e_3\right)\right] - k_\alpha \alpha^2 \\ &= e_1(v_r\cos e_3 - v) + \bar{e}_3\left[\omega_r - \left(1 + k_3 e_1\right)\omega + k_3 v_r \sin e_3 + \frac{e_2 v_r \sin e_3}{\bar{e}_3}\right] - k_\phi \phi^2 \end{aligned} \tag{4.80}$$

To stabilize the $e_1, e_2, \bar{e}_3$, the following control laws for v and ω are proposed,

$$v = \lambda_2 = v_r \cos e_3 + k_1 e_1 \tag{4.81}$$

$$\omega = \lambda_3 = \frac{1}{1 + k_3 e_1}\left(k_2\bar{e}_3 + \frac{e_2 v_r \sin e_3}{\bar{e}_3} + k_3 v_r \sin e_3 + \omega_r\right) \tag{4.82}$$

Substituting (4.81) and (4.82) into (4.80) gives

$$\dot{V}_1 = -k_1 e_1^2 - k_2 \bar{e}_3^2 - k_\phi \phi^2 \tag{4.83}$$

To steer the TWIP robot to track the desired heading angular velocity, from (4.72), the yaw rate controller can be designed based on the following yaw motion model.

$$\dot{\omega} = f_{23}(x) + \frac{Rb}{G_a} C_y \tag{4.84}$$

Define the sliding surface S_η as below.

$$S_\eta = \omega - \lambda_3 \tag{4.85}$$

Differentiating S_η gives

$$\dot{S}_\eta = \dot{\omega} - \dot{\lambda}_3 = f_{23}(x) + \frac{Rb}{G_a} C_y - \dot{\lambda}_3 \tag{4.86}$$

Consider the Lyapunov function as

$$V_2 = \frac{1}{2} S_\eta^2 \tag{4.87}$$

Differentiating it with respect to time gives

$$\begin{aligned} \dot{V}_2 &= S_\eta \dot{S}_\eta \\ &= S_\eta \left[f_{23}(x) + \frac{Rb}{G_a} C_y - \dot{\lambda}_3 \right] \end{aligned} \tag{4.88}$$

Let the yaw rate control law be

$$C_y = \frac{G_a}{Rb} \left[-f_{23}(x) + \dot{\lambda}_3 - k_{\eta\omega 1} sgn(S_\eta) - k_{\eta\omega 2} S_\eta \right] \tag{4.89}$$

such that,

$$\dot{V}_2 = -k_{\eta\omega 1} |S_\eta| - k_{\eta\omega 2} S_\eta^2 \leq -k_{\eta\omega 1} |S_\eta| \tag{4.90}$$

A torque control law for C_ψ can be determined by using the following simplified and coupled 2-state equation

$$\begin{bmatrix} \dot{\omega}_\phi \\ \dot{v} \end{bmatrix} = \begin{bmatrix} f_{21}(x) \\ f_{22}(x) \end{bmatrix} + \begin{bmatrix} g_{21}(x) \\ g_{22}(x) \end{bmatrix} C_\psi \tag{4.91}$$

The following two first-layer sliding surfaces are defined as

$$S_\phi = \eta_\phi = \omega_\phi - \lambda_1$$

$$S_v = \eta_v = v - \lambda_2 \tag{4.92}$$

whose derivatives are respectively given by

$$\dot{S}_\phi = \dot{\omega}_\phi - \dot{\lambda}_1 = f_{21}(x) + g_{21}(x)C_\psi - \dot{\lambda}_1$$

$$\dot{S}_v = \dot{v} - \dot{\lambda}_2 = f_{22}(x) + g_{22}(x)C_\psi - \dot{\lambda}_2 \tag{4.93}$$

And the second-layer sliding surface is defined as

$$S_1 = r_1 S_\phi + r_2 S_v \tag{4.94}$$

where r_1 and r_2 are two real parameters.

To stabilize the S_1, the following Lyapunov function candidate is chosen

$$V_3 = \frac{1}{2} S_1^2 \tag{4.95}$$

whose time derivative is

$$\begin{aligned} \dot{V}_3 &= S_1 \dot{S}_1 \\ &= (r_1 S_\phi + r_2 S_v) \left[r_1 \left(f_{21}(x) + g_{21}(x) C_\psi - \dot{\lambda}_1 \right) + r_2 \left(f_{22}(x) + g_{22}(x) C_\psi - \dot{\lambda}_2 \right) \right] \end{aligned} \tag{4.96}$$

from which the control law for C_ψ can be derived as

$$C_\psi = \frac{1}{r_1 g_{21}(x) + r_2 g_{22}(x)} \left[-r_1 f_{21}(x) - r_2 f_{22}(x) - r_1 k_\phi \dot{\phi} + r_2 \dot{\lambda}_2 - K_{s1} sgn(S_1) - K_{s2} S_1 \right] \tag{4.97}$$

such that,

$$\dot{V}_3 = -k_{s1}|S_1| - k_{s2}S_1^2 \leq -k_{s1}|S_1| \tag{4.98}$$

Sliding Mode with Fuzzy Adaptive Backstepping Controller Design

A sliding mode control with fuzzy adaptive backstepping is proposed to handle the nonlinearities. The unknown nonlinear terms can be estimated by the following fuzzy logic systems

$$f_{23}(x) = C_1^T \xi_1 + \varepsilon_1 \tag{4.99}$$

$$\frac{r_1 f_{21}(x) + r_2 f_{22}(x) - r_1\dot{\lambda}_1 - r_2\dot{\lambda}_2}{r_1 g_{21}(x) + r_2 g_{22}(x)} = C_2^T \xi_2 + \varepsilon_2 \tag{4.100}$$

where ε_1 and ε_2 are the approximation errors, which are assumed to satisfy $|\varepsilon_1| \leq g_{y\max 3}$ and $|\varepsilon_2| \leq g_{y\max 4}$.

Then,

$$\dot{S}_\eta = \dot{\omega} - \dot{\lambda}_3 = f_{23}(x) + \frac{Rb}{G_a}C_y - \dot{\lambda}_3 = C_1^T \xi_1 + \varepsilon_1 + \frac{Rb}{G_a}C_y - \dot{\lambda}_3 \tag{4.101}$$

$$\tilde{C}_1^T = C_1^T - \hat{C}_1^T \tag{4.102}$$

Define the following Lyapunov function candidate

$$V_2 = \frac{1}{2}S_\eta^2 + \frac{1}{2}\tilde{C}_1^T \Gamma_1 \tilde{C}_1 \tag{4.103}$$

whose time derivative is given by

$$\begin{aligned}
\dot{V}_2 &= S_\eta \left(C_1^T \xi_1 + \varepsilon_1 + \frac{Rb}{G_a}C_y - \dot{\lambda}_3 \right) - \tilde{C}_1^T \Gamma_1 \dot{\tilde{C}}_1 \\
&= S_\eta \left[\left(\tilde{C}_1^T + \hat{C}_1^T \right) \xi_1 + \varepsilon_1 + \frac{Rb}{G_a}C_y - \dot{\lambda}_3 \right] - \tilde{C}_1^T \Gamma_1 \dot{\hat{C}}_1 \\
&= S_\eta \left(\hat{C}_1^T \xi_1 + \varepsilon_1 + \frac{Rb}{G_a}C_y - \dot{\lambda}_3 \right) + S_\eta \tilde{C}_1^T \xi_1 - \tilde{C}_1^T \Gamma_1 \dot{\hat{C}}_1 \\
&= S_\eta \left(\hat{C}_1^T \xi_1 + \varepsilon_1 + \frac{Rb}{G_a}C_y - \dot{\lambda}_3 \right) + \tilde{C}_1^T \left(S_\eta \xi_1 - \Gamma_1 \dot{\hat{C}}_1 \right)
\end{aligned} \tag{4.104}$$

Using the inequality $A + B \leq A + |B|$, one obtains

$$\begin{aligned}\dot{V}_2 &= S_\eta \left(\hat{C}_1^T \xi_1 + \frac{Rb}{G_a} C_y - \dot{\lambda}_3 \right) + S_\eta \varepsilon_1 + \tilde{C}_1^T \left(S_\eta \xi_1 - \Gamma_1 \dot{\hat{C}}_1 \right) \\ &\leq S_\eta \left(\hat{C}_1^T \xi_1 + \frac{Rb}{G_a} C_y - \dot{\lambda}_3 \right) + |S_\eta||\varepsilon_1| + \tilde{C}_1^T \left(S_\eta \xi_1 - \Gamma_1 \dot{\hat{C}}_1 \right) \\ &\leq S_\eta \left(\hat{C}_1^T \xi_1 + \frac{Rb}{G_a} C_y - \dot{\lambda}_3 \right) + |S_\eta| g_{y\max 3} + \tilde{C}_1^T \left(S_\eta \xi_3 - \Gamma_1 \dot{\hat{C}}_1 \right) \end{aligned} \tag{4.105}$$

from which, the control law can be rewritten as follows:

$$C_y = \frac{G_\alpha}{Rb} \left(\dot{\lambda}_3 - \hat{C}_1^T \xi_1 - g_{y\max 3} sgn(S_\eta) - k_{\eta 3} sgn\left(S_\eta\right) - k_{\eta 4} S_\eta \right) \tag{4.106}$$

$$\dot{\hat{C}}_1 = \Gamma_1^{-1} S_\eta \xi_3 \tag{4.107}$$

From (4.106) and (4.107), (4.105) can be expressed by

$$\dot{V}_2 \leq -k_{\eta 3} |S_\eta| - k_{\eta 4} S_\eta^2 \tag{4.108}$$

On the other hand,

$$S_1 = r_1 S_\phi + r_2 S_v \tag{4.109}$$

Substituting the proposed fuzzy logic function into $\dot{S}_1$ yields

$$\begin{aligned}\dot{S}_1 &= r_1 \left[f_{21}(x) + g_{21}(x) C_\theta - \dot{\lambda}_1 \right] + r_2 \left[f_{22}\left(x\right) + g_{22}(x) C_\psi - \dot{\lambda}_2 \right] \\ &= \left[r_1 g_{21}\left(x\right) + r_2 g_{22}\left(x\right) \right] C_\psi + r_1 f_{21}(x) + r_2 f_{22}(x) - r_1 \dot{\lambda}_1 - r_2 \dot{\lambda}_2 \\ &= \left[r_1 g_{21}(x) + r_2 g_{22}(x) \right] \left(C_\psi + C_2^T \xi_2 + \varepsilon_2 \right) \end{aligned} \tag{4.110}$$

Then, the following Lyapunov function candidate is proposed

$$V_3 = \frac{1}{2} S_1^2 + \frac{1}{2} \tilde{C}_2^T \Gamma_2 \tilde{C}_2 \tag{4.111}$$

which leads to

$$\begin{aligned}\dot{V}_3 &= S_1\dot{S}_1 - \tilde{C}_2^T\Gamma_2\dot{\hat{C}}_2 \\ &= S_1\left[r_1 g_{21}(x) + r_2 g_{22}(x)\right](C_\psi + C_2^T\xi_2 + \varepsilon_2) - \tilde{C}_2^T\Gamma_2\dot{\hat{C}}_2 \\ &= S_1[r_1 g_{21}(x) + r_2 g_{22}(x)][C_\psi + (\tilde{C}_2^T + \hat{C}_2^T)\xi_2 + \varepsilon_2] - \tilde{C}_2^T\Gamma_2\dot{\hat{C}}_2 \\ &= S_1[r_1 g_{21}(x) + r_2 g_{22}(x)]\left(C_\psi + \hat{C}_2^T\xi_2 + \varepsilon_2\right) + \tilde{C}_2^T\left\{\xi_2 S_1\left[r_1 g_{21}(x) + r_2 g_{22}(x)\right] - \Gamma_2\dot{\hat{C}}_2\right\}\end{aligned} \tag{4.112}$$

Using the inequality $A + B \geq A - |B|$, one obtains

$$\begin{aligned}S_1\left(C_\psi + \hat{C}_2^T\xi_2 + \varepsilon_2\right) &= S_1(C_\psi + \hat{C}_2^T\xi_2) + S_1\varepsilon_2 \\ &\geq S_1(C_\psi + \hat{C}_2^T\xi_2) - |S_1||\varepsilon_2| \\ &\geq S_1\left(C_\psi + \hat{C}_2^T\xi_2\right) - g_{y\max 4}|S_1|\end{aligned} \tag{4.113}$$

Note that,

$r_1 g_{21}(x) + r_2 g_{22}(x)$ can be made negative if both r_1 and r_2 are properly chosen.

Moreover, from (4.112), it follows that

$$\begin{aligned}\dot{V}_3 \leq\ & [r_1 g_{21}(x) + r_2 g_{22}(x)][S_1(C_\psi + \hat{C}_2^T\xi_2) - g_{y\max 4}|S_1|] \\ & + \tilde{C}_2^T\{\xi_2 S_1[r_1 g_{21}(x) + r_2 g_{22}(x)] - \Gamma_2\dot{\hat{C}}_2\}\end{aligned} \tag{4.114}$$

Then, the control law can be derived as

$$C_\psi = -\hat{C}_2^T\xi_2 - k_3 sgn(S_1) - k_{s4}S_1 + g_{y\max 4}sgn(S_1) \tag{4.115}$$

$$\dot{\hat{C}}_2 = \Gamma_2^{-1}\xi_2 S_1\left[r_1 g_{21}(x) + r_2 g_{22}\left(x\right)\right] \tag{4.116}$$

(4.114) can be simplified as

$$\dot{V}_3 \leq \left[r_1 g_{21}(x) + r_2 g_{22}(x)\right]\left[-(k_3 + g_{y\max 4})|S_1| - k_{s4}S_1^2\right] \tag{4.117}$$

where

$$(k_3 + g_{y\max 4}) < 0, k_{s4} < 0 \tag{4.118}$$

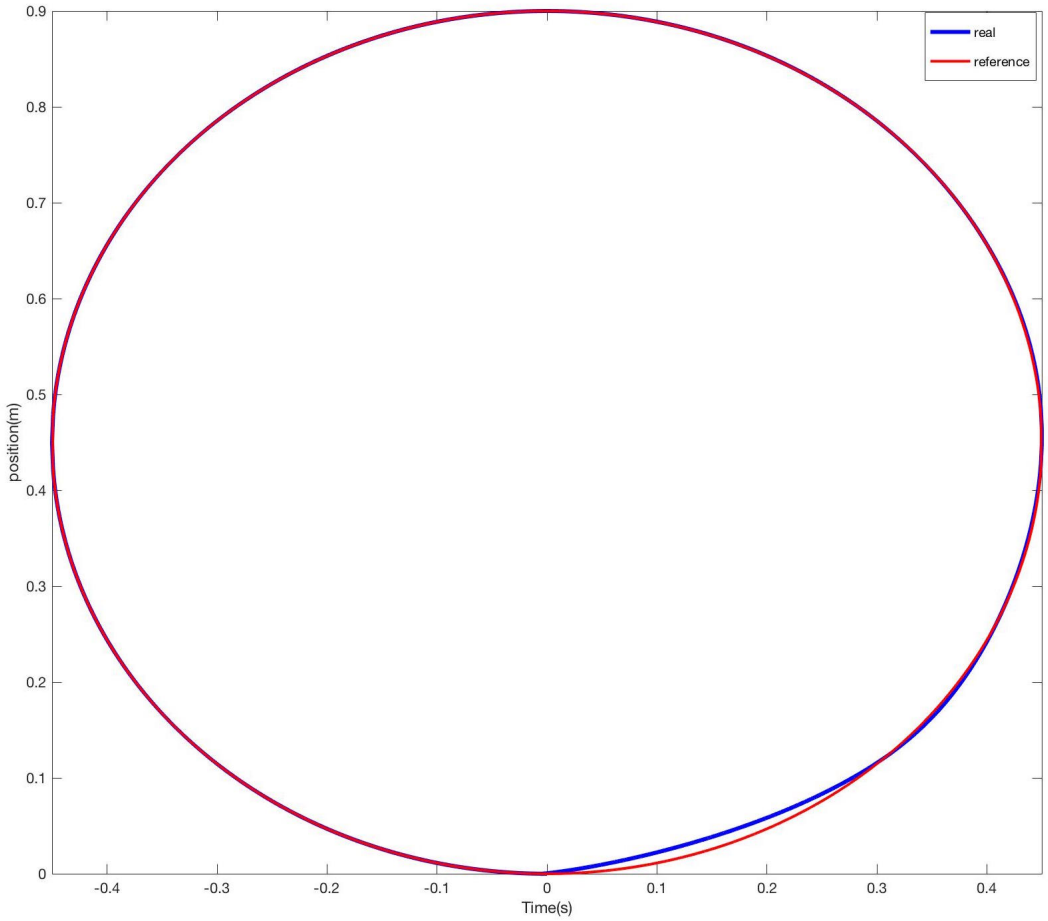


Figure 4.16: Circular Trajectory Tracking

As a result,

$$\dot{V}_3 < 0 \tag{4.119}$$

Then, the control inputs of the right and left wheels are given by

$$u_r = 0.5C_\psi + 0.5C_y \tag{4.120}$$

$$u_l = 0.5C_\psi - 0.5C_y \tag{4.121}$$

which can be derived from (4.87) Fig 4.17 shows the block diagram for the proposed sliding mode fuzzy adaptive control.

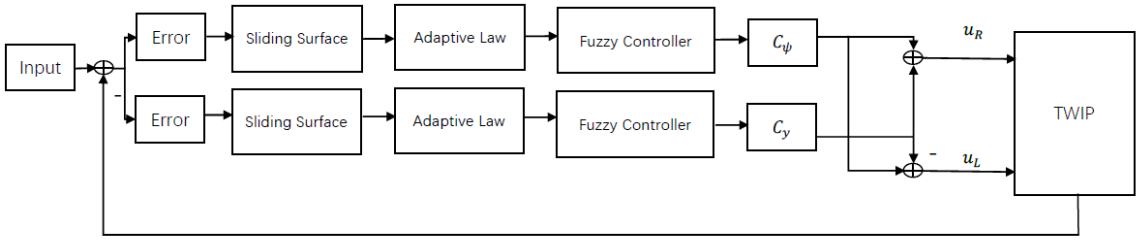


Figure 4.17: Block Diagram for Sliding Mode with Fuzzy Adaptive Control

Simulation Results

Fig 4.16 and Fig 4.18 present the simulated results of the sliding mode fuzzy adaptive controller for tracking a circle. In the simulation, the desired speed is set to be $\frac{\pi}{30}m/s$ and the desired yaw rate is set to be $\frac{\pi}{13.5}rad/s$. The radius of the circular trajectory is 0.45 meters. The parameters are $k_3 = -2, k_{s4} = -1, r_1 = -25, r_2 = 3, k_{\eta 3} = -2, k_{\eta 4} = -5, g_{y\max 3} = 1, g_{y\max 4} = 1$. It turns out that the proposed controller guarantees that the wheeled inverted pendulum robot is able to track the desired trajectory, desired speed and yaw angle while keeping body to be upright.

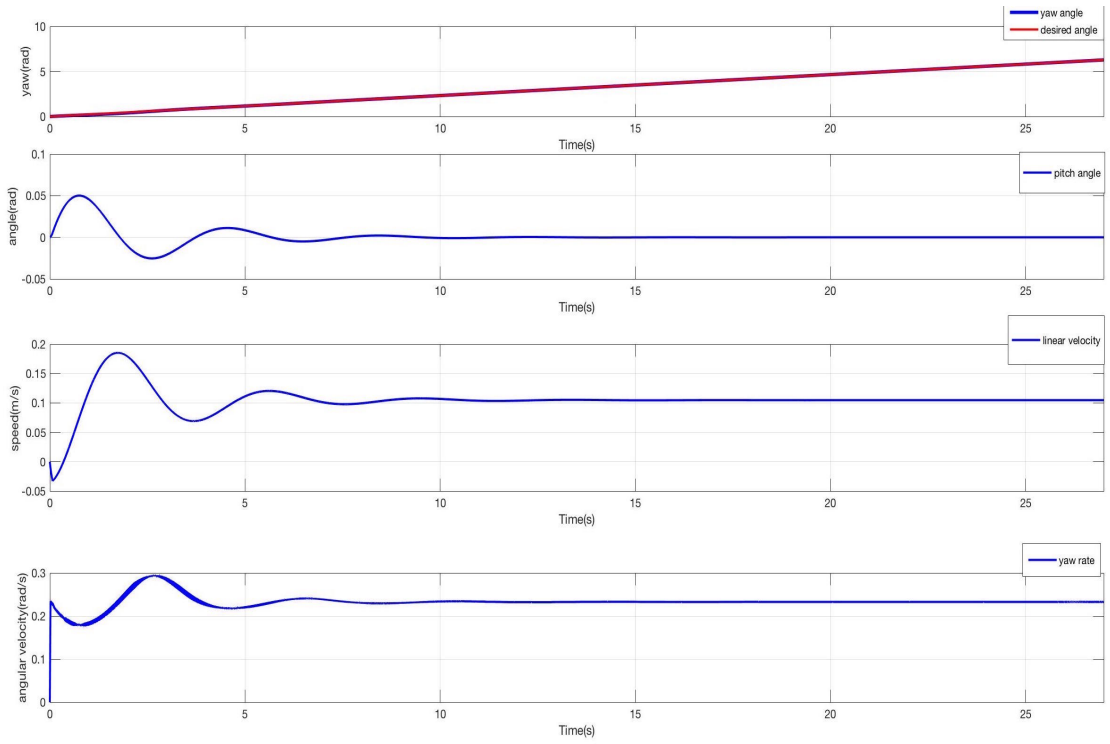


Figure 4.18: Simulation Results of Sliding Mode Fuzzy Adaptive Control

Experimental Results

In order to explore the real time performance of the sliding mode fuzzy adaptive controller, the experiment is performed on the TWIP platform. Fig 4.19 shows the actual trajectory of the TWIP. Other experimental results are shown in Fig 4.20. The desired linear velocity is chosen as 0.1 m/s. And the gain coefficients are chosen as $k_3 = -2, k_{s4} = -1, r_1 = -25, r_2 = 3, k_{\eta 3} = -2, k_{\eta 4} = -5, g_{y\max 3} = 1, g_{y\max 4} = 1$. It can be observed that the inclined angle is maintained around 0, and the speed is controlled to be about 0.1 m/s.

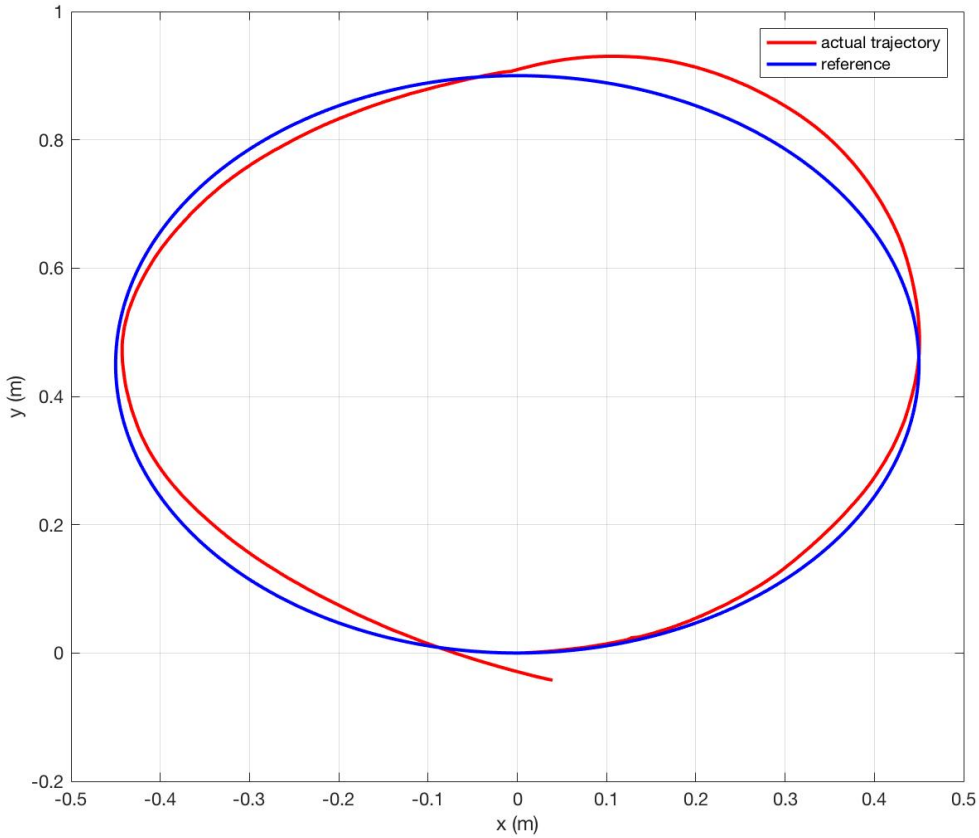


Figure 4.19: Circle Trajectory Tracking for Sliding Mode Fuzzy Adaptive Control

From the experimental results, it is evident that the proposed sliding mode fuzzy adaptive control method could ensure that the inverted pendulum system is able to achieve the expected tracking trajectory task with small errors.

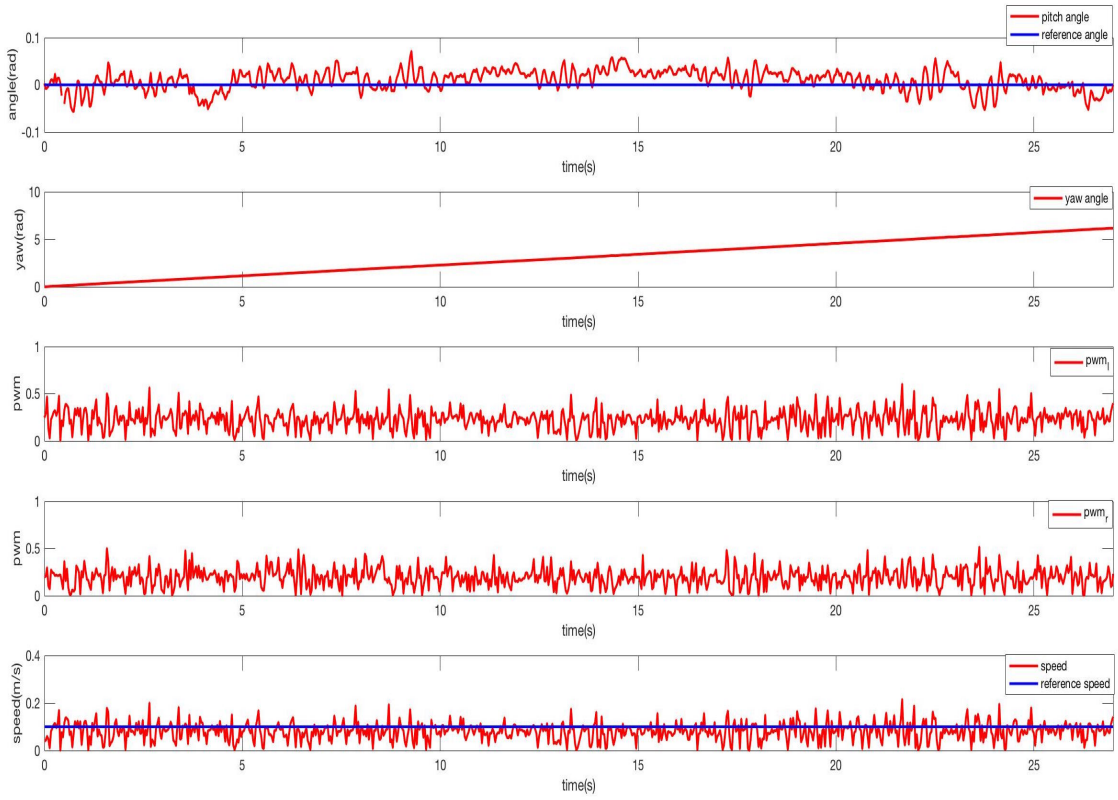


Figure 4.20: Experimental Results of Sliding Mode Fuzzy Adaptive Control

Chapter 5

Conclusion

5.1 Achievements of the Thesis

In this thesis, a trajectory tracking task with velocity and position control has been considered, and various controllers have been investigated and implemented to steer a TWIP robot.

Prior to the controller design, two different dynamic models have been introduced. It is a vital step to derive the mathematical model in order to accomplish a good control scheme. However, most of the existing works ignore the kinetic energy produced by the motors during building the dynamics. In this thesis, the motor kinetic energy has been considered as a part of the total kinetic energy. Based on Lagrange's motion equation, the dynamic model with 2 DOF is derived for the velocity control and position control. To achieve the trajectory tracking control scheme, another model with 3 DOF is also introduced.

Later on, aiming at steering the robot to run at desired speed and keep itself upright, a LQR controller and a fuzzy adaptive backstepping controller have been employed. The proposed fuzzy logic control design is carried out to approximate the uncertainties.

To suit the needs of the practical applications, an advanced LQR controller and a new fuzzy adaptive backstepping controller have been proposed to stabilize the inverted pendulum robot on a slope.

The trajectory tracking controller is designed after achieving a stable velocity and position control. The sliding mode backstepping control and fuzzy adaptive method have been synthesized to achieve self-balancing and motion tracking of desired trajectory. The adaptation laws of the proposed system are derived based on Lyapunov stability analysis.

Through the simulation results, the proposed controllers have been shown useful and effective in providing appropriate control actions to steer the vehicle.

The experimental tests demonstrate that the proposed controllers are straightforward to implement and exhibits good tracking performance.

The following comparisons are made in order to evaluate the performance of the LQR controller, the fuzzy adaptive controller, the PID controller and the sliding-mode with fuzzy adaptive controller. In Table 5.1, the average absolute error of the pitch angle of the body and the average absolute error of the rotation angular velocity for LQR controller and fuzzy adaptive controller are given. The results indicate that the performance of the fuzzy adaptive controller is a little bit better than the LQR controller on both a flat plane and a slope.

Table 5.1: Average Absolute Error of Pitch Angle and Rotation Angular Velocity

Control method	Pitch angle	Angular velocity
LQR control on a flat plane	3.225 deg	25.5 deg/s
Fuzzy adaptive control on a flat plane	3.024 deg	20.6 deg/s
LQR control on a slope	0.975 deg	50 deg/s
Fuzzy adaptive control on a slope	0.765 deg	22.32 deg/s

As for the trajectory tracking control, the average absolute error of the displacement on x axis and y axis is shown in Table 5.2. From the table, it can be confirmed that the performance of the sliding-mode with fuzzy adaptive controller is much better than the PID controller.

Table 5.2: Average Absolute Error on X Axis and Y Axis

Control method	x axis	y axis
PID	0.072 m	0.076 m
Sliding-mode	0.027 m	0.030 m

5.2 Future Work

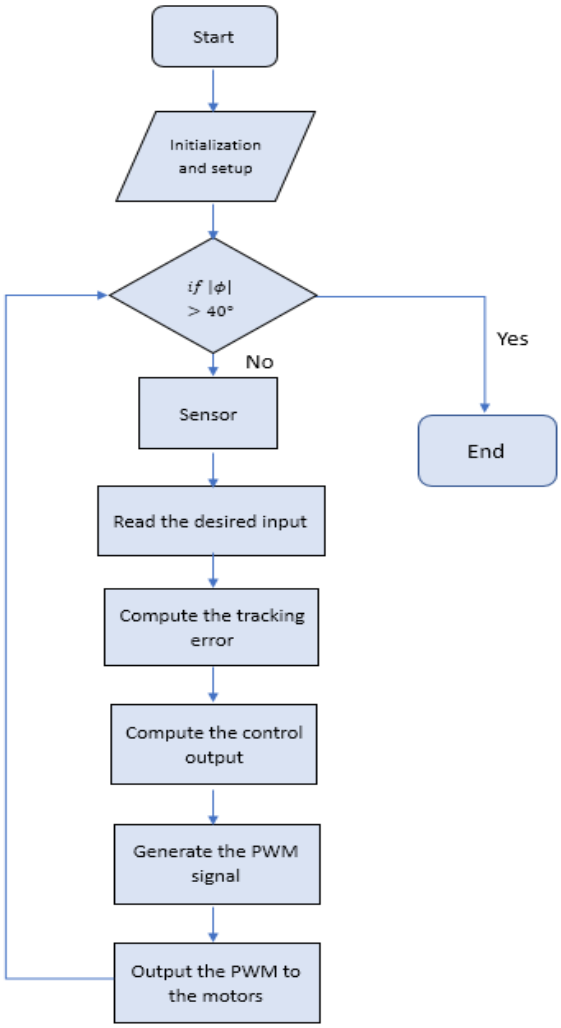
Probing deeper, the results in this thesis also provide a strong foundation for future work. Several new areas can yet be explored and many opportunities for extending the scope of this thesis remain. Moreover, some improvements can be enriched to my thesis. This section presents some of these directions.

- The trajectory tracking control should be implemented on a slope. The dynamic models used in Chapter 2 is obtained by omitting the spinning motion. However, a more complicated dynamic equation is necessary since both roll and yaw angles are required when the inverted pendulum robot spinning on a non-flat plane.
- A more extensive and complex membership function should be embedded into the fuzzy logic systems. Due to the limitation of the computational power, a trapezoidal membership function is used in the proposed fuzzy adaptive controllers. And more fuzzy rules can also be added to improve the accuracy of the estimations.
- The performance of the traditional controller is below expectations when the wheeled inverted pendulum robot running under different travelling conditions. Hence, one

should take more uneven terrains into consideration in system modelling and controller design.

Appendix A

C Language Flow Chart



Bibliography

[1] K. Yamafuji and T. Kawamura, "Study on the postural and driving control of a coaxial bicycle," in *Proceeding of the Meeting of Japan Society of Mechanical Engineering*, vol. 100, pp. 1114–1121, 1988.

[2] Y.-S. Ha *et al.*, "Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot," *Robotics and Autonomous Systems*, vol. 17, no. 1, pp. 65–80, 1996.

[3] N. Shiroma, O. Matsumoto, and K. Tani, "Cooperative behavior of a mechanically unstable mobile robot for object transportation," *JSME International Journal Series C*, vol. 42, no. 4, pp. 965–973, 1999.

[4] T. T. Bui, T. L. Nguyen, and S. B. Kim, "A simple nonlinear control of a two-wheeled welding mobile robot," *International Journal of Control, Automation and Systems*, vol. 1, no. 1, pp. 35–42, 2003.

[5] F. Ding, J. Huang, Y. Wang, T. Matsuno, T. Fukuda, and K. Sekiyama, "Modeling and control of a novel narrow vehicle," in *Proceeding of the IEEE International Conference on Robotics and Biomimetics*, pp. 1130–1135, 2010.

[6] K. Pathak, J. Franch, and S. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearizaiton," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505–513, 2005.

[7] J. Takahashi, "Wheeled inverted pendulum type assistant robot: inverted mobile, standing and sitting motions," in *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1932–1937, 2007.

[8] J. Li, X. Gao, Q. Huang, Q. Du, and X. Duan, "Mechanical design and dynamic modeling of a two-wheeled inverted pendulum mobile robot," in *Proceeding of the IEEE International Conference on Automation and Logistics*, pp. 1614–1619, 2007.

[9] Z. Li and C. Xu, "Adaptive fuzzy logic control of dynamic balance and motion for wheeled inverted pendulums," *Fuzzy Sets and Systems*, vol. 160, no. 2, pp. 1787–1803, 2009.

[10] H. Wang and C. Chiu, "Velocity control realisation for a self-balancing transporter," *IET Control Theory & Applications*, vol. 5, pp. 1551–1560, 2011.

[11] K. Yeonhoon, K. Soo, H. Kwak, and Y. Keun, "Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot," *Journal of Intelligent and Robotic Systems*, vol. 44, no. 1, pp. 25–46, 2005.

[12] R. K. Thomas and A. L. David, *Dynamics, Theory and Applications*. McGraw Hill, 1985.

[13] S. Agouri, O. Tokhi, A. Almeshal, O. Sayidmarie, and K. Goher, "Dynamic modelling of a new configuration of two wheeled robotic machine on an inclined surface," in *Proceeding of the 17th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 315–318, 2012.

[14] M. Key, Changgook, and D. Yoo, "Sliding mode control for a two-wheeled inverted pendulum mobile robot driving on uniform slopes," in *Proceeding of the International Conference on Control Automation and Systems*, pp. 2159–2162, 2012.

[15] M. Shehu and M. R. Ahmad, "LQR, double PID and pole placement stabilization and tracking control of single link inverted pendulum," in *Proceeding of the IEEE International Conference on Control System, Computing and Engineering*, pp. 218–223, 2015.

[16] A. N. K. Nasir, Ahmad, M. Ashraf, and Rahmat, "Performance comparison between LQR and PID controllers for an inverted pendulum system," in *Proceeding of the International Conference on Power Control and Optimization*, vol. 1052, pp. 124–128, AIP, 2008.

[17] F. Sun, Z. Yu, and H. Yang, "A design for two-wheeled self-balancing robot based on Kalman Filter and LQR," in *Proceeding of the International Conference on Mechatronics and Control (ICMC)*, pp. 612–616, 2014.

[18] J. Ha and J. Lee, "Position control of mobile two wheeled inverted pendulum robot by sliding mode control," in *Proceeding of the International Conference on Control Automation and Systems*, pp. 715–719, 2012.

[19] Y. Maruki, K. Kawano, H. Suemitsu, and T. Matsuo, "Adaptive backstepping control of wheeled inverted pendulum with velocity estimator," *International Journal of Control, Automation and Systems*, vol. 12, pp. 1040–1048, 2014.

[20] C. Ghorbel, A. Tiga, S. Rannen, and N. Benhadj, "Combined backstepping-PID control of inverted pendulum," in *Proceeding of the 14th International Multi-Conference on Systems, Signals and Devices (SSD)*, vol. 12, pp. 779–784, 2017.

[21] C. Chen and W. Chen, "Robust adaptive sliding-mode control using fuzzy modeling for an inverted-pendulum system," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 2, pp. 297–306, 1998.

[22] M. Park and D. Chwa, "Swing-up and stabilization control of inverted-pendulum systems via coupled sliding-mode control method," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3541–3555, 2009.

[23] H. Jian, Z. Guan, M. Takayuki, F. Toshio, and S. Kosuke, "Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 750–758, 2010.

[24] M. E. James, H. Brian, and D. Dan, *Inverted pendulum design with hardware fuzzy logic controller*. PhD thesis, South Dakota School of Mines and Technology, Rapid City, 2009.

[25] C. Ji and L. Fang, "Fuzzy logic controller for an inverted pendulum system," in *Proceeding of the IEEE International Conference on Intelligent Processing Systems*, vol. 1, pp. 185–189, 1997.

[26] A. Robert, "The use of fuzzy logic in the control of an inverted pendulum," in *Dynamical Systems: Theoretical and Experimental Analysis*, pp. 71–82, Springer, 2016.

[27] Y. Tang, D. Zhou, and J. Wen, "A new fuzzy-evidential controller for stabilization of the planar inverted pendulum system," *PloS One*, vol. 11, no. 8, 2016.

[28] A. Almeshal, M. Tokhi, and K. Goher, "Robust hybrid fuzzy logic control of a novel two-wheeled robotic vehicle with a movable payload under various operating conditions," in *Proceeding of the UKACC International Conference on Control*, pp. 747–752, 2012.

[29] T. Jung and S. Jung, "Line tracking control of a two-wheel balancing mobile robot: Experimental studies," in *Proceeding of the IEEE International Conference on Industrial Technology (ICIT)*, pp. 91–95, 2014.

[30] S. J. Mou and L. Feng, "Trajectory tracking simulation of the planar inverted pendulum based on LQR," in *Proceeding of the 15th International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pp. 697–701, 2015.

[31] H. Cho and S. Jung, "Neural network position tracking control of an inverted pendulum an X-Y table robot," in *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1210–1215, 2003.

[32] H. Chih and L. Chiang, "Adaptive fuzzy hierarchical sliding-mode control for the trajectory tracking of uncertain underactuated nonlinear dynamic systems," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 2, pp. 286–299, 2014.

[33] C.-C. Tsai, S.-Y. Ju, and S.-M. Hsieh, "Trajectory tracking of a self-balancing two-wheeled robot using backstepping sliding-mode control and fuzzy basis function networks," in *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3943–3948, 2010.

[34] C.-C. Tsai and Z.-Z. Wang, "Intelligent adaptive trajectory tracking using fuzzy basis function networks for self-balancing two-wheeled mobile robots," in *Proceeding of the International Conference on System Science and Engineering*, pp. 143–148, 2011.

[35] C. Huang, W. Wang, and C. Chiu, "Design and implementation of fuzzy control on a two-wheel inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 2988–3001, 2011.

[36] https://www.arduino.cc/en/main/arduinoBoardUno/.

[37] https://www.indiegogo.com/projects/balanbot-best-arduino-self-balancing-robot-ever/.

[38] www.sparkfun.com/products/8742/.

[39] http://www.dfrobot.com/wiki/index.php/12VDCMotor251rpmw/.

[40] N. Zheng, Y. Zhang, Y. Guo, and Z. Xiaohua, "Hierarchical fast terminal sliding mode control for a self-balancing two-wheeled robot on uneven terrains," in *Proceeding of the 36th Chinese Control Conference*, pp. 4762–4767, 2017.

[41] Y. Hosoda, S. Egawa, and M. Togami, "Basic design of human-symbiotic robot EMIEW," in *Proceeding of the IEEE International Conference on Intelligent Robots and Systems*, pp. 5079–5084, 2006.

[42] W. H. Gage, D. A. Winter, J. S. Frank, and A. L. Adkin, "Kinematic and kinetic validity of the inverted pendulum model in quiet standing," *Gait and Posture*, vol. 19, no. 2, pp. 124–132, 2004.

[43] S. Mu and J. Gao, "Adaptive fuzzy control in the application of primary inverted pendulum," in *Proceeding of the 2nd International Conference on Measurement*, vol. 01, pp. 732–736, 2013.

[44] F. Dai, X. Gao, S. Jiang, Y. Liu, and J. Li, "A Multi-DOF two wheeled inverted pendulum robot climbing on a slope," in *Proceeding of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1958–1963, 2014.

[45] T. Nomura, Y. Kitsuka, H. Suemitsu, and T. Matsuo, "Adaptive backstepping control for a two-wheeled autonomous robot," *2009 ICCAS-SICE*, pp. 4687–4692, 2009.

[46] M. Yue, C. An, D. Yu, and S. Jianzhong, "Indirect adaptive fuzzy control for a nonholonomic/underactuated wheeled inverted pendulum vehicle based on a data-driven trajectory planner," *Fuzzy Sets and Systems*, vol. 290, pp. 158–177, 2016.

[47] S. Tong, S. Sui, and Y. Lia, "Fuzzy adaptive output feedback control of MIMO nonlinear systems with partial tracking errors constrained," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 729–742, 2015.

[48] L. Wang, "Stable adaptive fuzzy controllers with application to inverted pendulum tracking," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 5, pp. 677–691, 1996.

[49] Y. Takeshi, "Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system," *Fuzzy sets and Systems*, vol. 32, no. 2, pp. 161–180, 1989.

[50] J. Yi, N. Yubazaki, and H. Kaoru, "A new fuzzy controller for stabilization of parallel-type double inverted pendulum system," *Fuzzy Sets and Systems*, vol. 126, no. 1, pp. 105–119, 2002.

[51] G. Kondalsamy, *Stability and oscillations in delay differential equations of population dynamics*, vol. 74. Springer Science & Business Media, 2013.

[52] K. Zhao, "Attitude control for a quadrotor UAV using adaptive fuzzy backstepping," *Master Thesis, Lakehead University, Ontario, Canada*, 2017.