eYSIP 2020

Model Based Robot Design

Project ID : 17-B
Prit Varmora
Aniket Nayak
Hruthwik K
Mentor 1:Avinash Dubey
Mentor 2:Avijit Pandey
Duration of Internship: 08/05/2020 – 27/06/2020

2020, e-Yantra Publication

Abstract

This paper discusses on the balance control of cycle bot.[1] A self-balancing cycle bot based on the concept of an inverted pendulum is an unstable and nonlinear system. To stabilize the system in this work, the following three main components are essential, i. e., an IMU sensor that detects the tilt angle of the cycle bot, a controller that is used to control motion of a reaction wheel, and a reaction wheel that is employed to produce reactionary torque to balance the cycle bot.[2] The PID controller is implemented to control the reaction wheel. The performance of the cycle bot is confirmed by the experiments.

Introduction

The development of technology for the convenience of humans has been evolving day by day. The leading sector in this development of technology is the robotics. This field is applied to many applications, including the industrial robots and unmanned robot. Among them, the research of unmanned robot is actively in progress. Most of unmanned robot was developed based on the four-wheel and six-wheel. The robot based on the four-wheel or six-wheel is hard to drive in the special environmental factor, such as the confined space. To solve the matter, a lot of research on the two-wheeled robot is in progress. The two-wheeled robot called the cycle bot/bicycle robot can freely drive in the narrow and the confined space, however, the robot has the characteristics of unstable structure. Many control methods have been proposed in order to improve the stability of the robot which has the unstable characteristics. This paper will also cover the control method for improving the stability of the robot. However, proving the dynamic model of robot is too complex to implement in real time. Therefore, the PID controller which don't requires mathematical modeling of system to implement is proposed to control the robot. To control the balance of robot, we assume that the robot's body and the reaction wheel as an inverted pendulum. The basic idea of working is when cycle bot starts falling in either direction the reaction wheel gets accelerated by PID controller to generate counter torque opposing the torque applied on cycle by gravity about pivot (contact point of wheel with ground). Due to the inertia of reaction wheel, reaction torque overcomes the gravity torque and cycle gets back to stable position vertically.

As mentioned above, cycle bot will behave like a inverted pendulum and,[3] there are many ways to balance an inverted pendulum. The most common way is to use a moving cart to move the pivot point to create counter moment. Another uncommon way of balancing is by rotating clockwise and anticlockwise to generate moment. Out of the many variations out there, the focus here is to balance an inverted pendulum by using reaction wheel. Unlike other type of inverted pendulum such as rotary or cart they require a runway or platform to balance itself because it is balanced by motion of the whole pendulum. However, reaction wheel inverted pendulum is balance by torque generated by the reaction wheel mounted on it self, this don't requires any platform or runway. To balance the cycle bot, The PID controller is used to not relying much on the dynamics of the cycle bot. And wireless communication is used to control the basic parameters of the cycle bot for speed controlling and left-right motion, for navigation process manually.

Reaction wheel is also mainly used in the satellite to provide precise rotation. To rotate the satellite in space, satellite rotated by spinning the reaction in the opposite direction. This allows the satellite to have rotational motion through its rotational axis according to newton's 3rd law.

Hardware parts

Electrical components are the vital part of the cycle bot. Table 1 shows the description of each component.

Table 1: Electronic components in the system

components	**Description of the components**
Actuators	6V DC Motors and Servo Motor
Microcontroller	Arduino Mega
Sensor	IMU MPU-6050 up to 6 DOF Accelerometer and Gyroscope
Motor Driver	L298N
Transmitter and Receiver	Xbee
Remote control	Joystick

Direct current (DC) motor is used due to the ability to provide high RPM and easy to control ability. The importance of reaction wheel is to create counter torque, and this is proportional to the acceleration of the wheel. if we keep reaction wheel at much height it requires less maximum torque to be generated on reaction wheel in order to balance cycle but it requires more RPM ratings and visa-versa if we keeps reaction wheel at less height we need more maximum torque but less RPM ratting of motor. So according to this tread off between motor torque and RPM ratings we have chose reaction wheel motor with 8.5 kg-cm stall torque and with 300 no load RPM as we are keeping our reaction wheel at height of 10.2 cm.

Another DC motor is used for navigation speed control. To measure the accurate tilt angle of the cycle bot, measurement of accelerometer and gyroscope sensors are fused together using kalman filter because accelerometer gives better results for long time interval but highly sensitive to noises and gyroscope can give better results in noisy environment but as we need to integrate data it generates problem of drifting over time it gives better result for small time span and . For navigation process i.e left/right motion DC servo motor is used. We are using Xbee's for wireless communication. Arduino Mega is used as a microcontroller due to its reliability and low cost.

Table 2: Circuit connection table

Hardware pin	**Arduino pin**
IMU:	
Vcc	5V
SDA	20
SCL	21

Hardware pin	**Arduino pin**
Xbee:	
Vcc	3.3V
Dout	RX1
Din	TX1

Hardware pin	**Arduino pin**
Motor Driver:	
E1 (rear wheel)	3
In1	27
In2	29
E2 (reaction wheel)	2
In3	25
In4	23

Hardware pin	**Arduino pin**
Servo:	
Vcc	5V o/p(motor driver)
signal	6

References for Hardware usage:

- Datasheet MPU 6050, page 5
- Guide for using Xbee

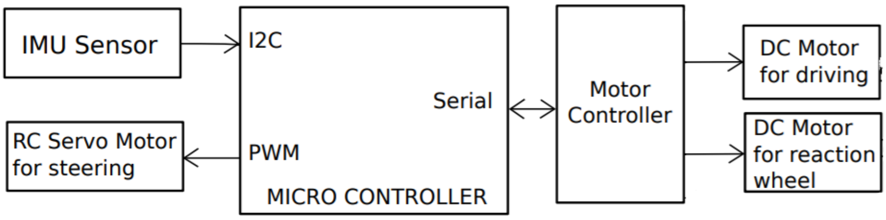

Figure 1: Block diagram of electronic system.

Software used

- List of software used:
  1. V-rep from coppeliaSim
  2. Arduino IDE
  3. Octave GNU

Assembly of hardware

Steps of assembly of hardware with pictures for each step:

Started from an aluminium pipe

While searching for right design for chessy, we found that aluminium will be solid and lighter. We used the Aluminium pipe as shown in fig.2.

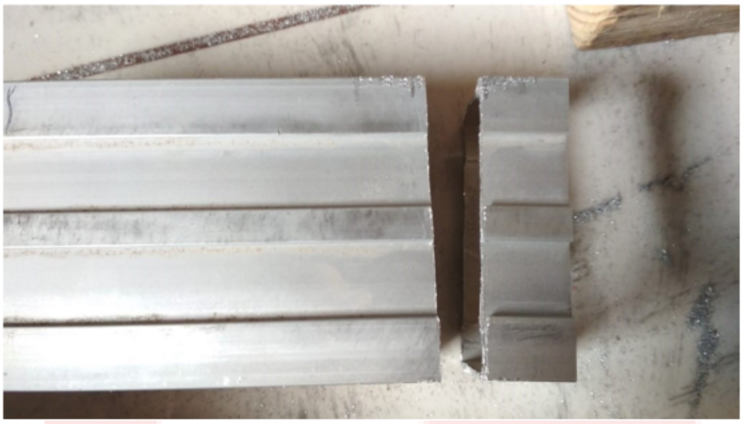

Figure 2: raw aluminium pipe

Step 1

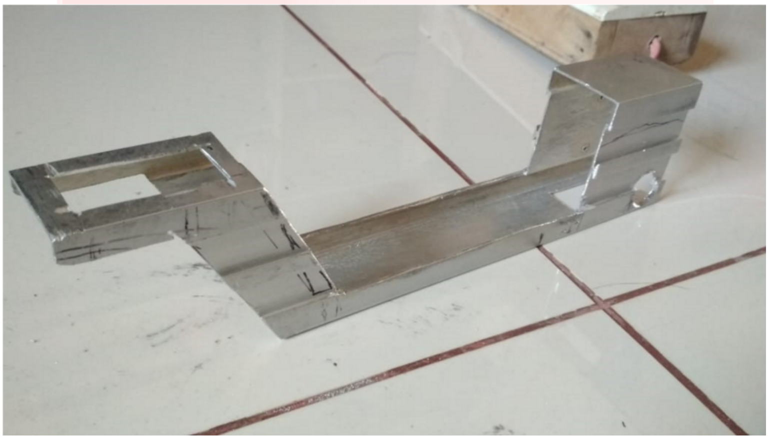

Figure 3: After cutting slots according to plane

Step 2

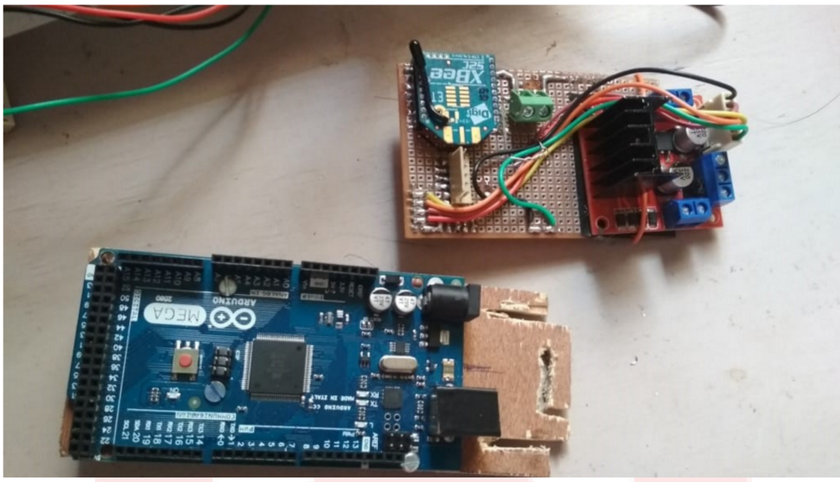

Figure 4: Building Circuit on wooden base

Step 3

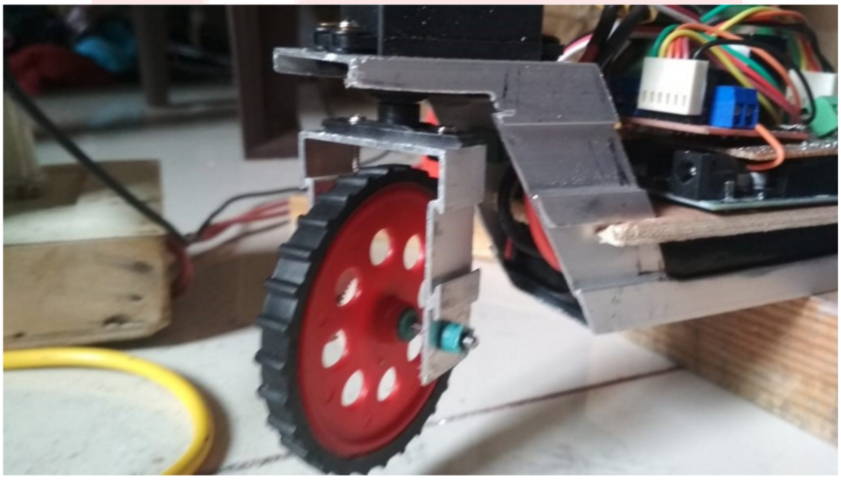

Figure 5: Building Steering wheel from pice of same pipe

Step 4

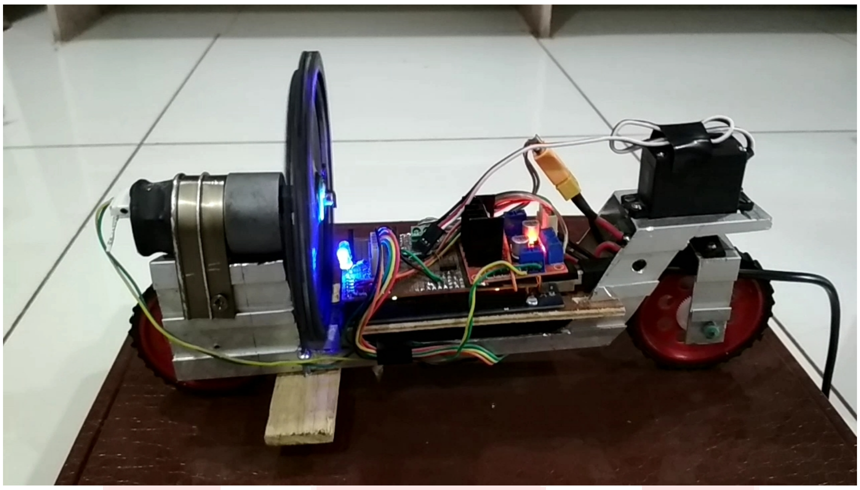

Figure 6: Final cycle Bot after assembling all together

Software and Code

- Github link for the repository of code
- Link for libraries to be installed to use above code

Navigation control

[5]To control the direction of the bicycle robot, a servomotor is used, which is directly connected to the front wheel and changes the directional angle. A 5V DC power supply is used to drive the servomotor to control the angle using the PWM duty cycle.

Explanation of various parts of code

full execution program is containing 1 main execution files and 4 local header files

- take_4.ino
- motor_run.h
- mpu_header.h

- remote_x.h
- testing_print.h

take_4 contains main setup and loop function. setup function configures the Arduino timers, initiates the communication with sensors and also resets them. loop function iterates continuously and receives and processes data from Xbee to control maneuvering the bot. To achieve uniform sampling we have pipeline sampling of IMU data and reaction wheel controlling tasks using timer interrupts.

motor_run.h file contains function definitions to control various motors used in cycle.

mpu_header.h file contains functions to initialise IMU. it also contains necessary function definitions for sampling IMU and fusing its data through complementary filter.

remote_x.h file contains function definitions for receiving Xbee data and extracting and analysing remote command data from packages for maneuvering the cycle bot.

testing_print.h header file is is not having use for controlling cycle bot but it contains function definitions for analysing various data and tuning PID controller through serial monitor.

NOTE: Functionality of all functions are briefly explained in the code files by well-commenting it.

Use and Demo

YouTube Link of demonstration video in V-rep simulation of:-

- Self- Balancing Inverted Pendulum with Revolute Joint.
- Self- Balancing Inverted Pendulum without revolute Joint.
- Balancing and Maneuvering of Cycle bot in V-rep.
- And Partial balancing of Real cycle bot in real environment.

The final One:

YouTube Link of demonstration video of Complete balancing of Real cycle bot in real environment.

Future Work

- To make the system more robust.
- Will try to increase the recovery angle.
- Maneuvering the Cycle bot in Real environment.
- Try to add some extra element like sensors for perception to avoid obstacle while maneuvering.
- Partial or fully autonomous to make decision on its own or with some little help of users to make the ride more smooth.

Bug report and Challenges

- In making Reaction wheel of perfect shape, size and weight.
- Getting the correct Angle with minimum noise at good rate.
- Uniform sampling with interrupts and timers.

  The major problem that we faced in handling timers for uniform sampling is that the timer that we are using and the timer used by the servo.h library to generate pwm signal is same. That's why the code is not compiling at that time. Later, the problem is solved using alternate servo library that uses Timer2.
- Choosing the right Sampling Time.
- Tuning PID gains.
- Balancing the Cycle Bot.

Work Completed

Table 3: Work Completed

	Task	Deadline
1.	Model and balance a simple inverted pendulum robot with reaction wheel (V-REP).	13^{th} *May* 2020
2.	Model and balance an inverted pendulum robot with chisel-like base (V-REP).	15^{th} *May* 2020
3.	Design and Model a cycle robot in V-REP.	16^{th} *May* 2020
4.	Balance the cycle robot in steady condition.	18^{th} *May* 2020
5.	Path following using cycle robot.	18^{th} *May* 2020
6.	Experiments and explanations of given topics.	24^{th} *May* 2020
7.	Choosing robot design, choosing hardwares and 3D modelling in V-REP	27^{th} *May* 2020
8.	GY-87, Filter design and communication protocols	30^{th} *May* 2020
9.	Shifting from Continuous to Discrete using transfer function	3^{rd} *June* 2020
10.	Studying the effect of sampling frequency, Ensuring uniform sampling in arduino by Writing ISRs	6^{th} *June* 2020
11.	Studying the effect of P Controller and Why we need Integral and derivative controller	9^{th} *June* 2020 .
12.	Implementing PID in Arduino and Tuning the Parameters	11^{th} *June* 2020
13.	Strategy for navigation via remote control and system debugging	13^{th} *June* 2020
14.	Balancing the Robot.Documentation and submissions.	27^{th} *June* 2020

Bibliography

[1] Kiattisin Kanjanawanishkul, “LQR and MPC controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel” Kybernetika, Vol. 51 (2015), No. 1, 173–191.

[2] Hyun-Woo KIm , Jae-Won An ,Han ong Yoo ,an Jang-Myung Lee, “Balancing Control of Bicycle Robot Using PID Control,” 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013) Oct. 20-23, 2013 in Kimdaejung Convention Center, Gwangju, Korea.

[3] Yon Yaw Lim1, Choon Lih Hoo2, and Yen Myan Felicia Wong1 ”Stabilising an Inverted Pendulum with PID Controller,” MATEC Web of Conferences 152, 02009 (2018) Eureca 2017.

[4] ”https://arduinoplusplus.wordpress.com/2017/06/21/pid-control-experiment-tuning-the-controller/”.

[5] Yunki Kim, Hyunwoo Kim and Jangmyung Lee1, ”Stable control of the bicycle robot on a curved path by using a reaction wheel”, Journal of Mechanical Science and Technology 29 (5) (2015) 2219 2226.

[6] Jepsen, F., Søborg, A., Pedersen, A. R., & Yang, Z. (2009). ”Development and Control of an Inverted Pendulum Driven by a Reaction Wheel”, In The 2009 IEEE International Conference on Mechatronics and Automation (pp. 2829-2834). IEEE. https://doi.org/10.1109/ICMA.2009.5246460.

[7] Pratik Raut, Shabbir Karjatwala, Shishir Kadam, ”Dynamic Modelling, Simulation & Control Design of Drive & Reaction Wheel Balancing Bot”, DOI:10.15680/IJIRSET.2016.0505112.

[8] Olfa Boubaker, ”The inverted Pendulum: A fundamental Benchmark in Control Theory and Robotics”, National Institute of Applied Sciences and Technology INSAT, Centre Urbain Nord BP. 676 – 1080 Tunis Cedex, Tunisia.