eYSIP 2020

Model Based Robot Design

Project ID 17

Chinmay Palaye
Jai Garg
Shreya Rastogi
Avinash Dubey
Avijit Pandey
Duration of Internship: 08/05/2020 – 27/06/2020

2020, e-Yantra Publication

Model Based Robot Design

Abstract

A self-balancing bicycle robot based on the concept of an inverted pendulum is an unstable and nonlinear system. This system balances using a reaction wheel, also known as inertia wheel. Behaviour of this system is studied by deriving a mathematical model of the system using Euler Lagrange method. This model is used to balance the cycle robot using LQR control method. To apply LQR, system is first linearized about the equilibrium points using Jacobian method. This model was first simulated in the simulation software CoppeliaSim (VREP). A physical cycle robot was designed in Fusion360 software, manufactured, and balanced. Project Overview video

Completion status

The project has been completed successfully. Learning Modules were created for various topics of Control systems and LQR controller. Cyclebot was simulated in CoppeliaSim and also implemented using hardware.

1.1 Hardware parts

- GY87 10DOF 3axis accelerometer, gyroscope, magnetometer: Datasheet
- XBee Module: Datasheet
- Arduino Mega2560: Datasheet
- L298 motor driver: Datasheet
- GS5515MG Servo Motor: Datasheet

1.2 Software used

- CoppeliaSim:
  - Version 4.0.0
  - Download Link
  - User Manual
- GNU Octave:
  - version 5.1.0.0
  - Download Link
  - Installation Steps
- Fusion 360™:
  - Version 2.0.8156
  - Download Link
  - Installation Steps
- Arduino IDE:
  - Version 1.8.39.0
  - Download Link
  - Installation Steps

1.3 Assembly of hardware

Circuit diagram and Steps of assembly of hardware with pictures for each step.

Circuit Diagram

Figure 1.1 shows the circuit diagram of Cycle Bot.

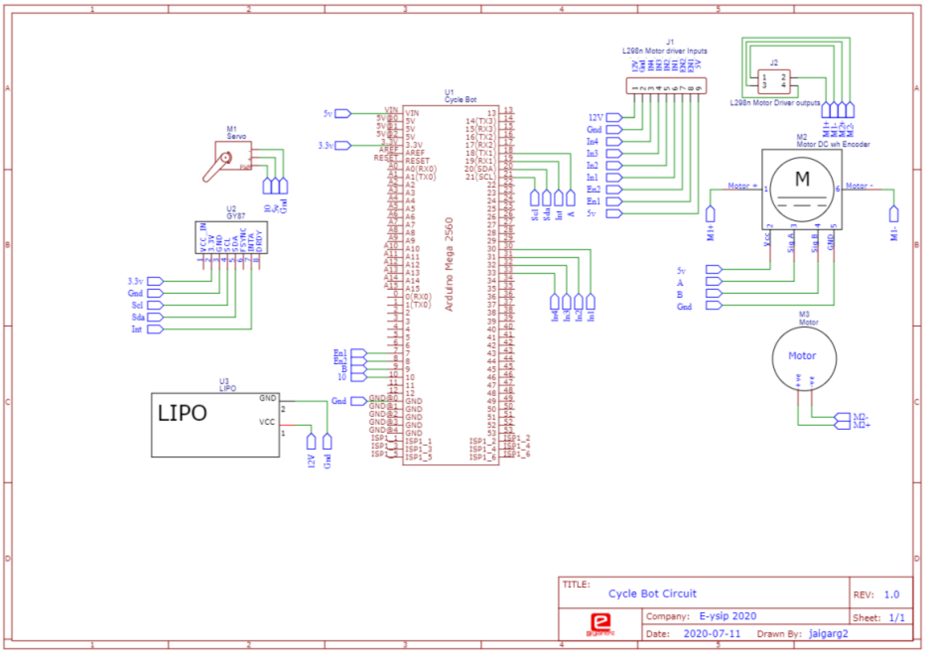

Figure 1.1: Circuit Diagram of Cycle Bot

Step 1

3D print all the 3D parts(Fig 1.1) in Week 6 in GitHub Repository (do not resize the models). Here is the link for design in fusion 360 video

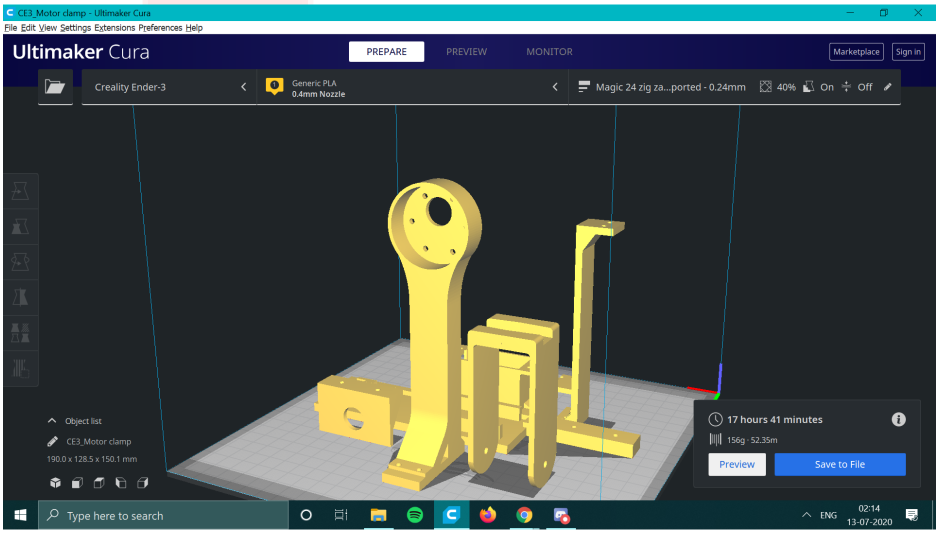

Figure 1.2: All 3D prints in Cura

Step 2

Attach Servo case and handle to the servo motor

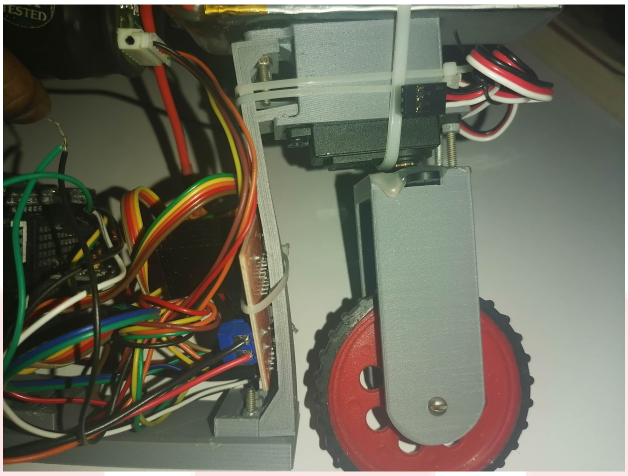

Figure 1.3: Front Assembly

Step 3

Attach middle slab and base using screws.

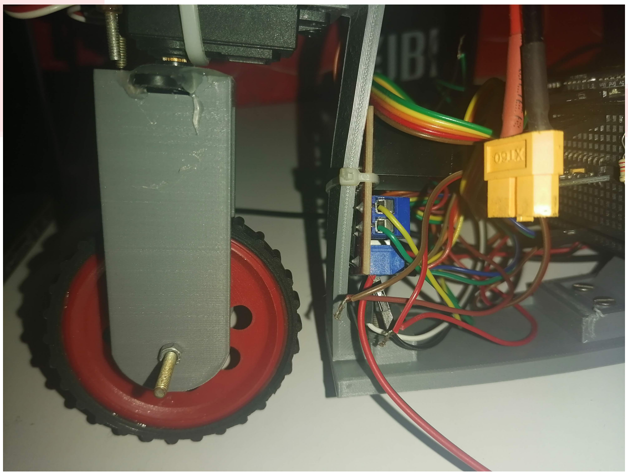

Figure 1.4: Base Connection

Step 4

Connect reaction wheel motor and base motor to reaction wheel stand and motor clamp respectively and then attach both on the base using screws(Fix reaction wheel tightly).

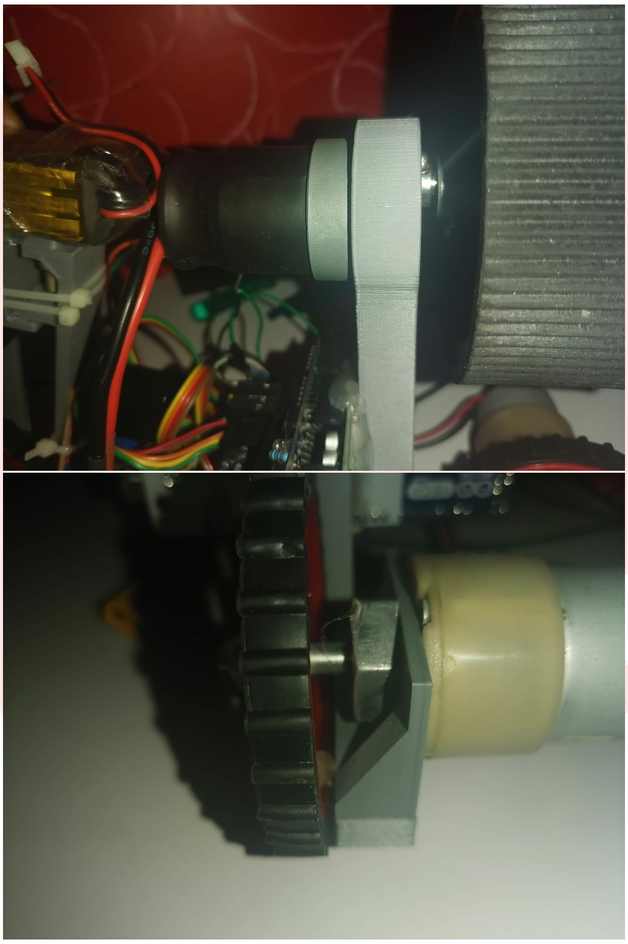

Figure 1.5: Reaction Wheel and Motor Clamp

Step 5

Then Connect all the electronic Components according to the circuit diagram.

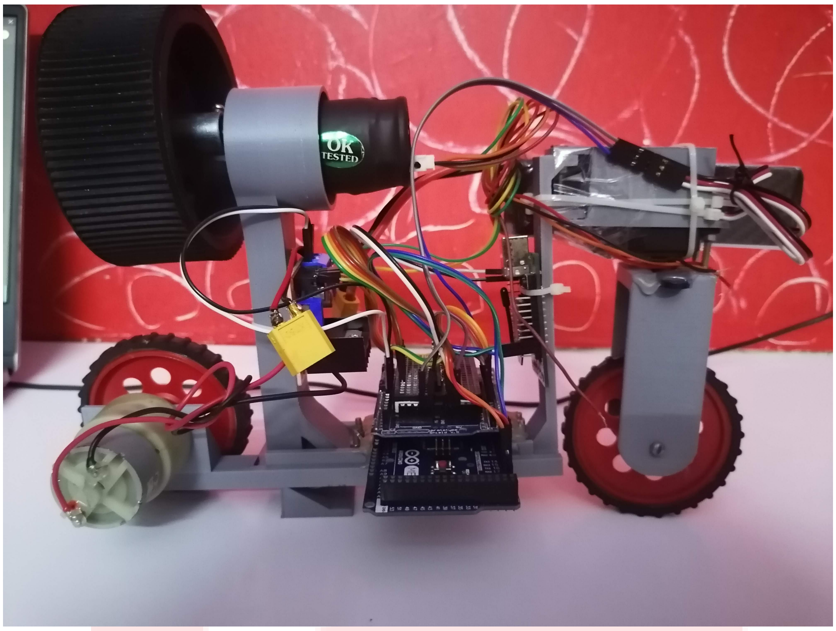

Figure 1.6: Fully assembled bot

1.4 Software and Code

GitHub link for the repository of code
Brief explanation of various parts of code:

- **Main_loop.ino:** The main code where the rest of the header files and functions have been called.
- **motor.h:** The file where the functions and working of the motors and related calculations take place.
- **controller_lqr.h:** The header file containing the controller of the system. The LQR controller has been implemented in this.
- **DMP.h:** This code gives the readings from the MPU6050 i.e. the roll and the pitch of the system.

- **mpu.h:** This code gives us the readings from the MPU6050 using filters.
- **torque_model.m:** This is the octave code which fives us the value of the gain matrix i.e. K matrix for our system.
- **xbee.h:** This is for the communication between our bot and the remote via Zigbeee communication.

1.5 Use and Demo

Final Setup Image

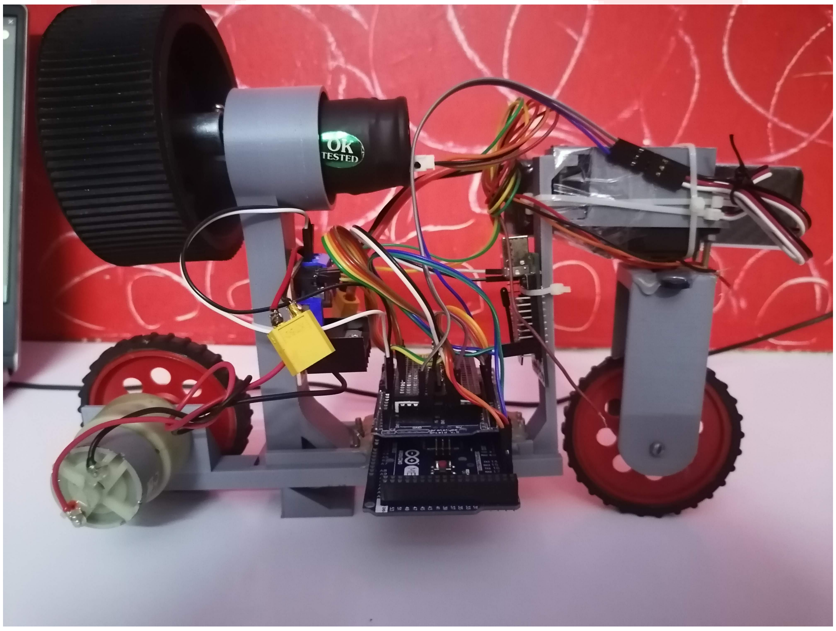

Figure 1.7: Fully assembled bot

Youtube Link of demonstration video

1.6 Future Work

Presently we have managed to balance the cyclebot for a very short duration i.e. a few seconds. In the future the first thing to be implemented would be it's successful balancing. Then more work can be done regarding it's movement. The cyclebot should be able to move forward, take turns, climb an inclined plane etc. in the future projects.

1.7 Bug report and Challenges

Failure or challenges faced during project:

- The hardware wasn't available for making the cyclebot so we had to take help of a 3D printer.
- The implementation of LQR controller to balance the cyclebot was a huge challenge.
- Main challenge faced was in balancing the cyclebot.
- Usage of timers in the code was slightly tricky as well.

Bibliography

[1] Kiattisin Kanjanawanishkul, *LQR and MPC controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel*, 2015.

[2] Adam Owczarkowski, Dariusz Horla and Joanna Zietkiewicz, *Introduction of feedback linearization to robust LQR and LQI control – Analysis of results from an unmanned bicycle robot with reaction wheel*, 2019.

[3] Jepsen, Frank; Søborg, Anders; Pedersen, Anders R.; Yang, Zhenyu, *Development and Control of an Inverted Pendulum Driven by a Reaction Wheel*, 2009.

[4] Yon Yaw Lim, Choon Lih Hoo, and Yen Myan Felicia Wong, *Stabilising an Inverted Pendulum with PID Controller*, 2018.