Research Inventy: International Journal of Engineering And Science
Issn (e): 2278-4721, Issn (p):2319-6483, PP: 38-40
www.researchinventy.com

Self Balancing Robot

Nayana Jain[1], Pinaki Gupta[2] , Sameeksha Jain[3] ,Simmi Agrawal[4]
[1]*Computer Science Engineering, Arya Institute of Engineering and Technology, Jaipur*
nayanajain854@gmail.com
[1]*Computer Science Engineering, Arya Institute of Engineering and Technology, Jaipur*
guptapinaki26@gmail.com
[1]*Computer Science Engineering, Arya Institute of Engineering and Technology, Jaipur*
sameekshajain09@gmail.com
[1]*Computer Science Engineering, Arya Institute of Engineering and Technology, Jaipur*
simmiagrawal07@gmail.com

Abstract— *Two-wheeled balancing robots are a growing subject of research that could one day provide locomotion for everyday robots. The specific stability control required to keep the robot upright distinguishes traditional generations of robotics. The inverted pendulum idea allows for mathematical modeling of the naturally unstable system. This data is then utilised to build and implement a suitable stability control system that is responsive, timely, and effective in achieving the desired outcome.*

We make a robot which is self balancing. The approach is very effective and plays a key role in the field of robotics and its evolution. The theory used here is of the inverted pendulum. The main purpose is to balancing the system at the initial level and control the system. The robot determines its exact position in three-dimensional geometry using accelerometer and gyroscope sensor values and delivers them to the microcontroller. The microcontroller, on the other hand, uses programming to instruct the motor driver module how to rotate the wheels, which aids in the robot's equilibrium. This robot outperforms traditional four-wheeled robots by being able to turn quickly and manoeuvre in tighter places, making it a useful machine for a range of industrial applications.

Keywords— *ATMEGA328P, Self-Balancing, Accelerometer, Gyroscope, PID, Complementary filter*

I. INTRODUCTION

The designing, building, and controlling system of a self-balancing robot having two- wheels are described here. The components drives and their instructions in this robot: DC motors, a microcontroller, a 3-axis gyroscope, and a 3-axis accelerometer for determining the attitude. A supplemental is implemented for dealing the problem of horizontal motions that happens in an instant and drifts in sensors. This project's feedback technique is PID (proportional integral derivative).

To know the accurate position of the robot, we must use sensor values. The accelerometer is used for determining the orientation in static conditions where as gyroscope is an effective indication of tilt angle in dynamic conditions. Both the accelerometer's slow moving signals and the gyroscope's fast moving signals must be combined. Because the gyroscope is largely made up of a drift, the results are frequently incorrect. The accelerometer, generates genuine numbers when the acceleration is progressive, but it is prone to vibrations, resulting in inaccurate angle measurements. A complementary filter is used, which is a math filter that combines both signal values, to achieve the proper results.

II. *Proposed System:-*

Our project will be completed in three parts. 1) Mechanical 2) Electrical 3) Coding. While working on the mechanical level, we initially developed the robot's mechanical chassis. Screwing and fastening cut acrylic sheets together, as well as installing motors and tyres, were all part of this operation. On the electrical level, we built a simple L293D/L298 motor driver circuit. IC 7805 and 7809 were among them. The ATmega 328 microcontroller board from Arduino Duemilanove was used. This board requires the IC 7809 for power (constantly at 9 Volts). The Sharp GP2D120 sensors we utilised required a 5V supply. As a result, the IC 7805 was required, which generates a constant 5V output voltage.

This 5V is also used to enable the motor driver and as a logical high. We created a code that would allow the Arduino to give the motor driver appropriate PWM and PWM direction based on the input data provided by the sensors. The difference between the analogue readings provided by the sensors to the Arduino defined the PWM signal's value and direction fully. The higher the PWM, the larger the difference. The PWM application's direction was determined by detecting whether the output PWM (produced by the PID control mechanism) is positive or negative.

For the purpose of simplicity, I assume that the robot's movement is limited to one axis (i.e., it can only move forward and backward), and that both wheels move at the same speed in the same direction. Because we just have to worry about sensor values on a single plane, the calculations become substantially easier. We'll have to operate each wheel separately if we want the robot to travel sideways. The general concept remains the same, albeit with less complexity, because the robot's falling direction is still constrained to a single axis.

Components and Working :-

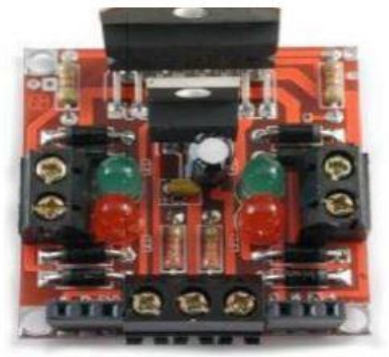
Controller: The controller I used in this project was the Arduino UNO because it is really simple to use. You can also use an Arduino Nano or Arduino Mini, but I recommend the UNO because it can be programmed without any additional hardware.

Motor: Stepper motors are unquestionably the best motor choice for a self-balancing robot.
However, to keep things simple, I used a DC gear motor.Yes, a stepper motor isn't required; instead, the bot can run on these cheap yellow-colored DC gear motors.

MPU6050:

MPU6050 is a Micro Electro-mechanical system with a three-axis accelerometer and three-axis gyroscope (MEMS). It is used to calculate velocity, direction, acceleration, displacement, and other motion-related parameters.

Motor Driver: You can use the L298N driver module, if you have selected the DC gear motors. L293D should also work in this.



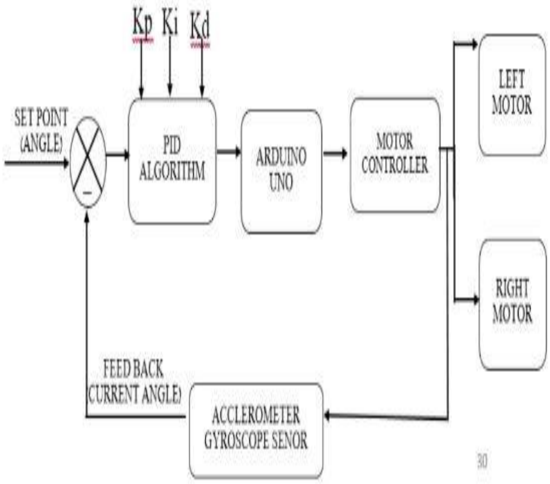
Wheels: Check closely, your grip should never allow wheels to skit on the floor. Also make sure that wheels have good grip on the floor we are using with.

Accelerometer and Gyroscope: The MPU6050 is the perfect accelerometer and gyroscope for your bot.So don't try to build one with a conventional accelerometer like the ADXL345 or something like.

Battery: To power our Arduino without using a boost module, we'll need a battery that's as light as possible and has a higher operational voltage than 5V.The best option is a 7.4V Li-polymer battery. Because it was readily available, I utilised a 7.4V Li-ion battery. Keep in mind that a Li-Po battery is preferable to a Li-ion battery.

Working Principle:

The Digital Motion Processor (DMP) took this information and turned them into a more comprehensible collection of variables like yaw, pitch, and roll. Because it is used for creating the tilt in the axis which is needed to be taken under notice, just pitch was necessary in this case. The controller received this information and delivered feedback to the microcontroller. The microcontroller used program-defined ways to process the values collected from the DMP.



Applications:-

Self balancing robots have application in a variety of areas. These are:-

- Self balancing Unicycle is a kind of device which is efficient enough in balancing itself on three dimensions.
- The 2-Monocycle Is The World's First Self-Balancing Bike: The Monocycle is a fantastic city bike that can be used for short trips and parked in the tiniest of spaces. The bicycle also has a powerful electric motor that allows users to accelerate without expending too much effort.
- 3 self-balancing electric scooters/balance bikes with remote switch: This is a self-balancing electric bike with two wheels that balance themselves.
- Two-wheeled balancing robots have a variety of applications. These includes agricultural sector where the farmer can use the product for the purpose of effective gardening, automatic trolley system at airports, in the hospitals, malls, and at several workplaces.

III. CONCLUSION

In this study, the concept of inverted pendulum is used, and its application was expanded to make a robot which is self-balancing. When pushed forward or backward, the robot balances itself using this concept. The needs, functions, and linkages of the components are discussed in a deeper manner. We can improve the idea even further by using quadrature optical encoders to improve the precision of motor speed measurements and hence improve stability. Potentiometers is an effective device by which one can fine-tune the error constants. Some of the requested features were not implemented because of the limiting time constraints, however, we can be utilized them as a tweak to the current system while developing a better and a more effective solution.

REFERENCS

[1]. E. Ferreira, S. T. (June 2000). Advanced Robotics. Vol 14 No 6
[2]. F. Grasser, A. D'Arrrigo, S. Colombi, and A. C. Rufer, "JOE: A mobile, inverted pendulum," IEEE Transactions on Industrial Electronics, vol. 49, no. 1, pp. 107 – 14, 2002.
[3]. karthik Raja, B. S. (2018, July). Development of Self Balancing Robot. International Journal of Robotics and Autonomous System Volume3 Issue 1 .
[4]. Shircliff, D 2002, Build a Remote-Controlled Robot, McGraw-Hill, Sydney.
[5]. Larson, T, 2008, Balancing Robot Project - Bender, viewed 20th March 2008