Design and Development of a Two Wheeled Self Balancing Robot.

Pranay Doble[1], Shradha Parkhi[2], Anagha Bodhe[3], Dr. D. S. Mantri[4]

Student, Department of Electronic and Telecommunication,
Sinhgad Institute of Technology, Lonavla, India[123].
Professor, Department of Electronic and Telecommunication
Sinhgad Institute of Technology, Lonavla, India[4].

Abstract: ***Two Wheeled Self Balancing Robot is a most popular research topic in the area of robotics and control engineering. This project deals with the theoretical principles and concept of inverted pendulum which is naturally unstable. The major focus on this paper is the hardware development of a two wheeled self-balancing robot. The main application of the project is to carry objects from one place to another. The modeling of the self-balancing robot is done in terms of the inverted pendulum. As the two wheeled self-balancing robot is unstable and nonlinear different types of controllers like PID are used.***

Keywords: Self balancing robot, Inverted pendulum, PID controller.

I. INTRODUCTION:

Robotics has always been played an integral part in the human life. The dream of creating a machine that replicates human thought and physical characteristics extends throughout the existence of mankind.

Developments in technology over the past fifty years have established the foundations of making these dreams come true. Robotics is now achievable through the miniaturization of the microprocessors which performs the processing and computations.

To make a self-balancing robot, it is essential to solve the inverted pendulum problem or an inverted pendulum on cart. While the calculation and expressions are very complex, the goal is quite simple: the position so that the inclination angle remains stable with in a pre- determined value, when the robot starts to fall in one direction, the wheels should move in the inclined direction with a speed proportional to angle and acceleration of falling to correct the inclination angle. So, we get an idea that when the deviation from and when the deviation is large, we should move more quickly. Self-balancing robot is an inverted pendulum example problem therefore it is difficult to balance [1].

The paper proposes the idea is to keep the robot upright by driving the wheels towards the leaning angle tilted.

The main objectives of this paper are

A. To get the robot to settle at the upright position in the shortest settling time and smallest over shoot.

B. To demonstrate the methods and techniques involved in balancing an unstable robotic platform on two wheels.

C. To move a predetermined distance along the horizontal whilst keeping its upright position.

The Complete paper is organized in different sections as,
Section II: Explanation about block diagram,
Section III: Working of self-balancing robot
Section IV: Flow Chart
Section V: Explore the functionality
Section VI: Hardware requirements
Section VII & VIII: Result is discussed and concluded.

II. BLOCK DIAGRAM:

The design of the system is quite challenging to bring the hardware and software to work together. The main components in the circuit of the two-wheel balancing robot are the MPU (6050), the Atmega328 controller, the stepper motor, motor driver, ultrasonic sensor, bluetooth and 12V battery.

Fig 1 shows the overall block diagram of the electronic system for the balancing robot. The MPU6050 is used to measure the acceleration and the angular rate of the robot and the output is processed into digital form. The raw inputs from the IMU are further processed to obtain the tilt angle of the robot. This tilt angle is then fed into the PID controller algorithm to generate the appropriate speed to the stepper motor in order to balance the robot. The ultrasonic sensor is used to measure the object distance and help the robot to prevent accident [2].

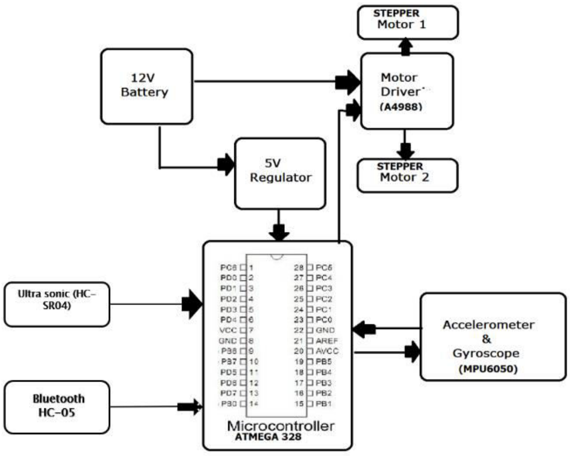


Fig 1. Block diagram of Self-balancing Robot

III. WORKING:

The basic component which is required for working of the self-balancing robot is a control system, PID algorithm, sensor and actuator. The inverted pendulum can’t control itself in an upright position as shown in Fig 2.

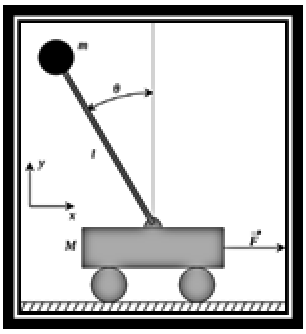


Fig 2: A inverted pendulum model

If we look at the Fig 2. when force is applied in a forwarding direction the pendulum moves in a backward direction so, therefore inverted pendulum is considered highly unstable. With the help of a microcontroller, the control system establishes a close feedback loop between the sensor and actuator. The data generated by the motor process and fed to the sensor then it is compared with the set point and detect for error with the help of the PID algorithm. After that, the output is fed to the microcontroller and then the actuator resets its position according to new data [3].

IV. FLOW CHART:

The Fig 3. represents the step by step working of the model.

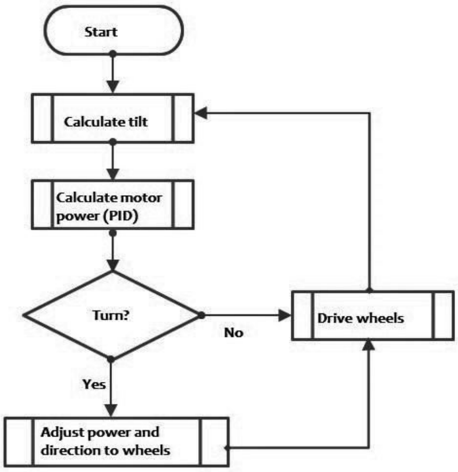


Fig 3. Flow Chart

Step 1: The robot is started when the power supply of 5v is applied to the microcontroller.

Step 2: by using the MPU sensor the tilt value is calculated.

Step 3: The title value is fed to the PID algorithm and is compared with the setpoint.

Step 4: After comparing the value with the setpoint it is feedback to the driving wheels and it adjusts its power and direction of the wheel.

Step 5: This process is followed in a continuous closed loop.

V. FUNCTIONAL OPERATION:

The detailed operation of self-balancing robot is divided in different parts as

A. CONTROL SYSTEM:

The purpose of a control system is to keep a system, within a specified range of elements and set variables. With the help of close loop feedback with the help of sensor and actuator. This could refer to numerous applications such as production, assembly and industrial plants through to computer, electrical and electronic systems. For this it refers to the control system charged with maintaining stability of the robot chassis as shown in Fig 4. Controlling the stability of the balancing robot requires sensors to detect the direction and rate of motion as well as a decision-based application that will provide the response signals to the actuator [4].

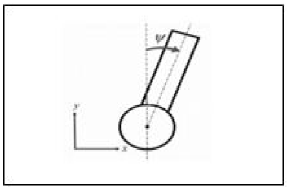


Fig 4: Simplified sketch showing the tilt angle, ψ and the position x

B. PID CONTROL SYSTEM:

The algorithm which was used to balance the self-balancing robot on its own was PID controller. The Proportional-Derivative-Integral (PID) is an instrument used in control applications such as to regulate temperature, flow, pressure, speed and other process variables. It is also a closed loop control system and also called as negative feedback system. PID is the most stable controller. The working principle behind a PID controller is that the proportional, integral and derivative terms must be individually adjusted or "tuned"[5].

A PID controller continuously calculates an error value e(t) which is subtraction of desired output value from the reference set-point value. The error is then given to the PID controller, where the error gets managed in three ways and summation of those is use to create the correcting signal. Again, the error value is given to the feedback loop and the process continues till it get stable. Fig 5. represents the concepts of PID

$$V(t) = Kp*e(t)+ Ki \int e(t)\, dt+ Kd\, de/dt \qquad \text{--(1)}$$

where,

V(t) is the voltage,
e(t) is the error signal,
Kp is the proportional gain,
Ki is the integral gain,
Kd is the derivative gain

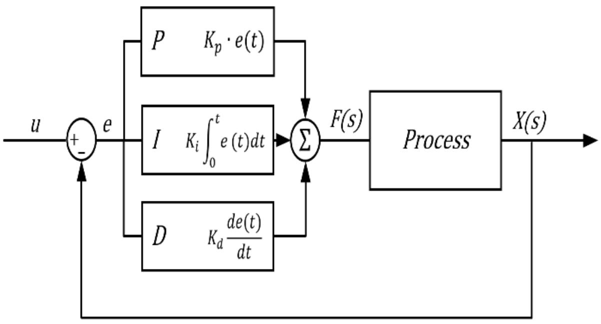


Fig 5: Concept of PID

C. EFFECT OF CONTROL PARAMETERS ON THE CLOSE LOOP SYSTEM:

Due to proportional controller, we will have reduced the rise time but no effect on steady state error. An integral control loop reduces the steady-state error for step input, but negative effect on rise time. A derivative increases the stability of the system as well as reduces the overshoot. Table 1. shows effects of PID

Table 1: PID parameter effect comparison.

Closed loop response	Rise time	Over shoot	Settling time	Steady state error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	eliminate
Kd	Small Change	Decrease	Decrease	No change

VI. HARDWARE REQUIREMENTS:

A. MPU6050

Fig 6: MPU 6050

- MPU6050 sensor module is 6-axis Motion Tracking Device as shown in Fig 6.
- It has 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor.
- It has I2C bus interface to communicate with the microcontrollers 328P.

  Specifications:
  a. Supply voltage: 2.3-3.4 V
  b. Accelerometer
    1.Measures ranges: ±2g ±4g ±8g ±16g
    2.Calibration tolerance: ±3%
  c. Gyroscope
    1.Measuring ranges: ±250/500
    2.Calibration tolerance: ±3%

B. STEPPER MOTOR

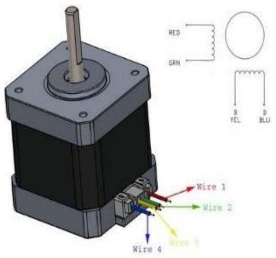


Fig 7: Stepper Motor

- Above Fig 7. shows stepper motor whose operating voltage is 12v, it is 3 phase brushless DC motor and divides full rotation 200 stepper revolution with step angle is equal to 1.8.
- It has high torque and low vibration at low speed.

represent stepper motor.

Specifications:

a. Weight: 350g
b. Size: 42.3mm square x 48 mm
c. Voltage rating: 4V

C. STEPPER MOTOR DRIVER

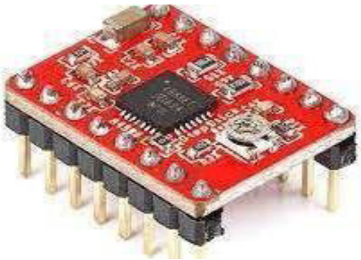


Fig 8: A4988 Stepper Motor Driver

- Fig 8 shows the module of stepper motor driver it has output drive capacity of up to 35 V and ±2A.
- Has built-in translator for easy operation.
- This reduces the number of control pins to just 2, one for controlling the steps and other for controlling spinning direction.
  Specification:
  a. Maximum operating voltage: 35V
  b. Minimum operating voltage: 8V
  c. Current per phase: 2A
  d. Micro step resolution: Full step,1/2 step,1/4 step,1/8 step and 1/16 step

D. ATMEGA 328P

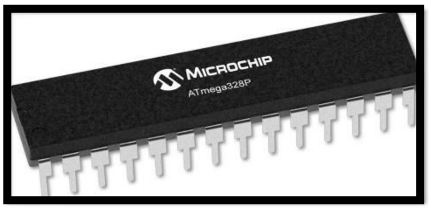


Fig 9: A4988 Stepper Motor Driver

- At mega 328P as in Fig 9 is an 8-bit AVR RISC-based microcontroller.
- It has 32kb flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general-purpose input-output lines, 32 general purpose working registers, and three flexible timer/counter.
- SPI- serial port,6-channel 10-bit analog to digital converter and the device is operating in a range of 1.8-5.5 V

E. LITHIUM POLYMER BATTERY

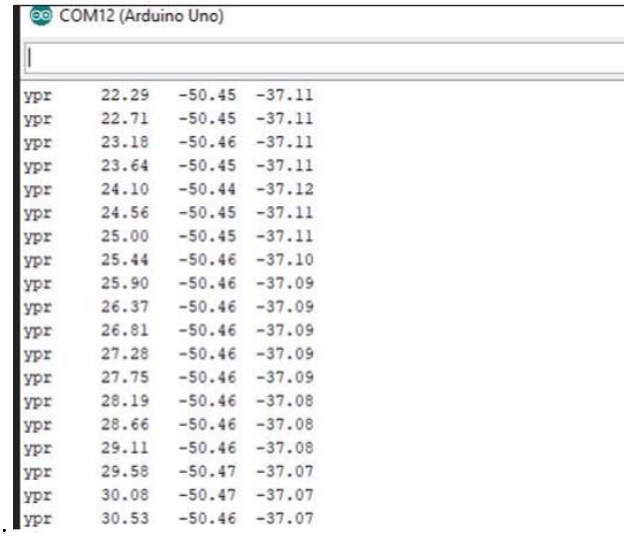
Specification:
Voltage: 11.1V
Constant Discharge: 30C (10.5A)
Max discharge: 40C (10 sec)
Dimensions: 74 x 34 x 22 (LxWxH)(mm)
Weight: 107 gm
Charge Rate: 1-3C Recommended, 5C Max

Fig 10: LiPo battery

VII. RESULT:

By doing the simulation using Arduino IDE we are able to get the values of yaw, pitch, and roll as shown in the Fig 11. The graphical representation of yaw (pink) and pitch (sky blue) and roll (yellow) are shown in Fig 11. For the initial time, the value is oscillated after that it becomes stable and it is followed in a close loop pattern after deriving this value for self-balancing robot works well.

After simulation, the final working model is implemented as in Fig 13, which consist of all the component mention in the section VI



```
ypr    22.29   -50.45  -37.11
ypr    22.71   -50.45  -37.11
ypr    23.18   -50.46  -37.11
ypr    23.64   -50.45  -37.11
ypr    24.10   -50.44  -37.12
ypr    24.56   -50.45  -37.11
ypr    25.00   -50.45  -37.11
ypr    25.44   -50.46  -37.10
ypr    25.90   -50.46  -37.09
ypr    26.37   -50.46  -37.09
ypr    26.81   -50.46  -37.09
ypr    27.28   -50.46  -37.09
ypr    27.75   -50.46  -37.09
ypr    28.19   -50.46  -37.08
ypr    28.66   -50.46  -37.08
ypr    29.11   -50.46  -37.08
ypr    29.58   -50.47  -37.07
ypr    30.08   -50.47  -37.07
ypr    30.53   -50.46  -37.07
```

Fig 11: Readings of yaw, pitch and roll

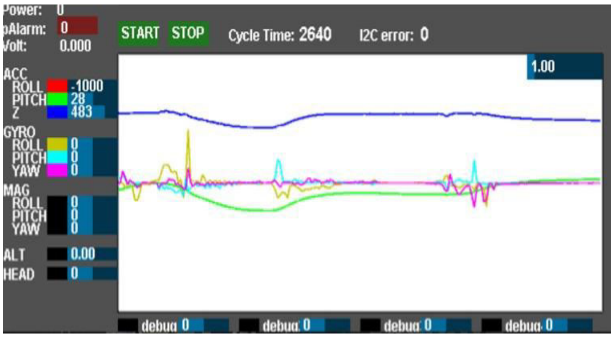


Fig 12: Representation of poles

Fig 13: Self balancing robot

VIII. CONCLUSION:

This paper has presented implementation of a two-wheel self-balancing robot equipped with two supporting wheels. The structure as well as hardware components has been introduced. The self-balancing robot is able to hold its upright position in the shortest settling time and can move a predetermined distance along a horizontal line. And can carry objects from one place to another.

REFERENCE

[1] Yun Su, T. Wang, K. Zhang, C. Yao, Z. Wang , "Adaptive Nonlinear Control Algorithm for a Self-Balancing Robot", in IEEE access vol. 8, pp. 3751 - 3760, 2019.

[2] Xiao WU, Yulei GONG and Huijiao MA ,"Research on Control Strategy of Two- wheeled Self-balancing Robot", in 2015 International Conference on Computer Science and Mechanical Automation.

[3] Hau-Shiue Juang and Kai-Yew Lum , "Design and Control of a Two-Wheel Self- Balancing Robot using the Arduino Microcontroller Board", in 2013 10th IEEE International Conference on Control and Automation (ICCA)

[4] Ruan Xiaogang, Cai Jianxian, Li Xinyuan, Zhao Jianwei. Research and "Design of Two- wheeled Self-balancing Robot". Beijing: Science Press, 2012.

[5] Kealeboga Mokonopi, "BALANCING A TWO WHEELED ROBOT", A dissertation report Courses ENG 4111 and ENG4112 Research Project (Issue: November, 2006).