PID for a Multi-wheeled Robot using a Six Axis Gyroscope as the Feedback Sensor.

Rigveda Koranne[1]

[1]*B.E. Student, Dept. of Electrical Engineering, Progressive Education Society's Modern College of Engineering Pune, Maharashtra, India*

---***---

Abstract - *This research paper is based on the regular problem that occurs in the locomotion of an autonomous multi-wheeled robot. We frequently observe that an autonomous robot is unable to traverse on a specified path to its target location; that is its locomotion has many errors. This problem usually occurs due to asymmetrical mechanical structures, or faults in assembling or manufacturing, or errors in electronic modules that cause imbalanced current distribution to the actuators, also this happens due to unbalanced load on the robot. This problem can be countered and reduced by using the concept of PID. PID takes input from a sensor which acts as the feedback about the position and orientation of the robot and computes the appropriate values of P, I and D and according to that the output is given to the connected actuators which indeed reduces the error and brings the robot on its programmed path. In this I have used a Six Axis Gyroscope as the sensor which gives the feedback of the current position of the robot to the microcontroller and by using the concept of PID the error is reduced.*

Key Words: **Autonomous, asymmetrical, locomotion, multi-wheeled, PID, six axis gyroscope, microcontroller.**

1.INTRODUCTION

There is a drastic increase in mobile robotic applications which include the use of automated multi-wheeled robots to perform various applications. Whilst performing the various applications the robot has to traverse in an environment on a pre-programmed path. Thus, in order to develop an autonomous robot, it should be equipped with a suitable locomotion system. The programmed path can include going in a straight line or moving along a curved path, or any other kind of motion. And the robot has to have accuracy and precision in its locomotion in order to precisely perform the objectives. Obtaining accuracy in locomotion could be difficult because of various factors like unbalanced load on the robot, fault in mechanical assembling or manufacturing, or even due to non-uniform current distribution to the actuators due to unideal electronic modules; Also, autonomous mobile robots have problems in determining its position and orientation in the environment. These problems lead to errors in the mobility of the robot leading to incompletion of desired objectives. Thus, having only, a suitable locomotion system is not much effective, it has to be supported by an apt control system. For that purpose, it is proposed to use a closed loop control system like the PID controller.

Thus, in order to reduce the errors, we use the concept of PID controller which is a well-established way of driving a system to its target position. PID control uses a closed loop feedback system to get close to the desired setpoint output. PID controller needs to have a feedback on the current position of the system in order to calculate the errors and correct them, this feedback is given by the sensors like encoders and other IMUs (Inertial Measurement Units).

In this we will be using an IMU that is a Six Axis Gyroscope (MPU6050) to give a feedback of the system's position to the PID controller for it to tune it and reduce the errors occurring in its motion.

This problem occurred to me when I was developing an Omnidirectional Autonomous Robot based on four wheeled(X-Configured) Holonomic drive, build on a rectangular chassis. It had a weight of approximately 9[Kg] and maximum speed of 1.8[m/s]. Power supply to the actuators was provided from a 12[V] LiPo battery having 8000[mAh] capacity. Arduino Mega 2560 was used as the microcontroller for the project. The robot had to perform certain objectives during which it had to traverse along a specified path that included going in a straight line and also moving along a curved path.

1.1 PID Controller

The PID algorithm is a common control algorithm which is universally established. It consists of three tuning parameters which can be calculated with respect to time. The basic concept is that the controller reads a sensor and then computes the desired actuator output by calculating the proportional, integral and derivative and then summing the components to obtain an optimal output. For calculating the present errors, the proportional term is used, for the previous errors the integral term is used, and the derivative term provides a prediction about the future errors

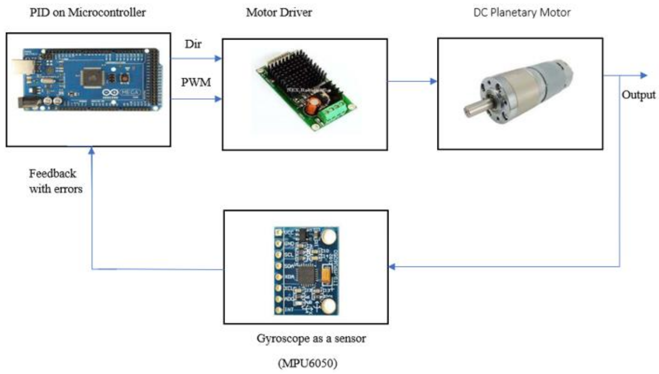

Fig.-1: Flowchart of the PID Controller process

The PID algorithm is implemented which will allow us to change the speed of rotations of each motor connected to the microcontroller. The motor will give the respective output and change the position and orientation of the robot; And the change in the position or orientation of the robot is sensed and taken as an input by the sensor (here a gyroscope) and given to the microcontroller as feedback. The microcontroller then tunes the appropriate Kp, Ki and Kd values to correct the error that is generated.

1.2 Six Axis Gyroscope (MPU6050)

MPU6050 is an Inertial Measurement Unit that contains a MEMS 3-axis accelerometer and MEMS 3-axis gyroscope in a single chip; It uses standard I2C Communication Bus. It has Six 16-bit Analog Digital Converter (Three for accelerometer and three for gyroscope) inbuilt in the module for higher accuracy; Also, it has an integrated Digital Motion Processor (DMP) on chip. It works on +5(V) VCC and the SDA and SCL pins on the module which are connected to the Arduino's SDA and SCL pins respectively. Tri-angular rate sensor or Gyroscope has a full-scale range of ±250, ±500, ±1000, and ±2000 dps.

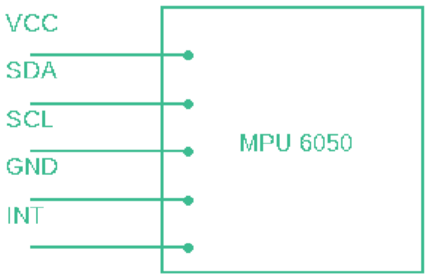

Fig-2: MPU6050 Connections

The DMP processor is embedded in the chip which computes the values of accelerometer and gyroscope together to minimize the effects of errors in each sensor.

Pin on MPU6050	Pin on Arduino
Vcc	+5 (V) VCC
GND	GND
SDA	SDA
SCL	SCL
INT	Pin 2(Interrupt pin)

The program for DMP (Digital Motion Processor) which is included in the MPU6050 library of Arduino was used, the program calculated raw values of pitch, roll and yaw that is rotation about the x, y and z axes respectively which are also known as the Euler Angles.

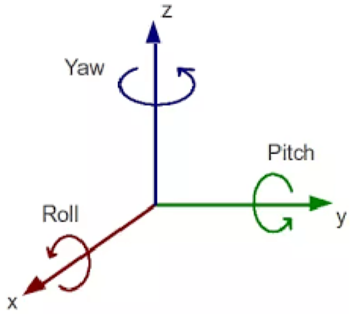

Fig-3: Yaw Pitch and Roll Axes

2. MPU 6050 Gyroscope and PID Interfacing

The MPU 6050 Six Axis Gyroscope sensor will be mounted on the chassis of the robot as close to the ground level as possible for greater accuracy. The IMU sensor gives the feedback about the current position and orientation of the robot in terms of pitch, roll and yaw values on the x, y and z axes respectively. Now we have to integrate the DMP program that gives the values of position, with the PID algorithm and get appropriate values for Kp, Ki and Kd.

In the program we have to set a base position which will be considered as the origin position of the robot with coordinates (0,0,0) on the x, y and z axes respectively. The origin position will be the position at which the robot is initially placed and switched on. The orientation at which the robot is switched on will be considered as the origin position and will be set to (0,0,0) coordinates, and as the robot is initially static to its position there will be no rotations around any axes; Thus, we will be getting some stable readings of the Euler angles(yaw, pitch and roll), that readings would be taken as the origin readings and will be set to 0. The DMP program prints the values of yaw, pitch and roll on the Serial monitor of the Arduino IDE so we can see the feedback values that are being provided as the output by the IMU.

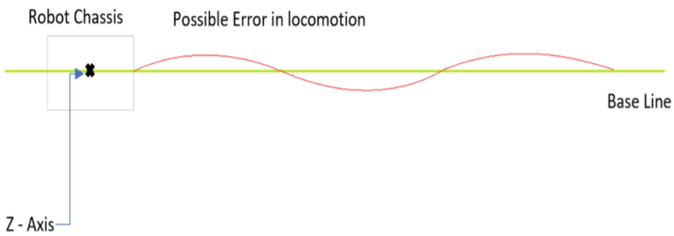

Fig-3: Locomotion of the Robot and possible errors.

The robot is ideally supposed to move along the base line (Green line in figure 3) when it is programmed to move straight, but there could be possible errors (Red line in figure 3) in its movement. Therefore, from the (Figure 3) we can say that any rotation about the Z-axis will be considered as an error when the robot is traversing in a straight line.

The movement of the robot will only be along the Y plane so there will not be any rotation about the Y axis and X axis, considering the robot does not topple or fall. So, the values of pitch and roll should be ideally zero or practically very close to zero considering the mounting of the sensor is proper and stable. There will only be change in the values of yaw as there will be rotations about the Z axes as the robot changes its direction or if there's an error while going along a straight line. If the command given to the robot is to go straight forward then any value of yaw other than 0 will be considered as an error because in traversing straight there should not be any rotation about the Z axis. If the value is anything other than 0 then the PID controller will compute the values so that the motor speeds vary to make the orientation of the moving robot straight and the value of yaw to be as close to 0 as possible.

When the robot is traversing along a curve the yaw value could be set according to the orientation of the robot along the curve which will be priorly fed in the program. Now if the yaw value is anything other than that particular pre-fed value then it will be considered as an error; And the PID controller will tune the motors to get a feedback value of yaw to be close to the pre-fed value. The robot has to perform certain trial and errors on the turning angle of the curve in order to determine the pre-fed yaw value that the IMU provides when the robot is taking the turn.

3. CONCLUSIONS

In this paper, the problem regarding the locomotion of an autonomous robot on a specified path precisely and with accuracy with the help of a closed looped control system that is the PID controller and interfacing it with the Six Axis Gyroscope IMU for feedback of the position and orientation of the robot is discussed. The faults in mechanical assembling or manufacturing of a robot, non-uniform distribution of current to the actuators due to unideal electronic modules, the unequal load distribution on the robot and other problems causing the instability and divergence in the robot's ideal motion, which are considered as errors can be reduced and eliminated by interfacing the PID controller with the MPU6050 gyroscope. By using this interfacing, we can achieve optimal movement paths for autonomous robots to reach its target setpoint locations.

REFERENCES

Helder P. Oliveira, Armando J. Sousa, A. Paulo Moreira and Paulo J. Costa in (2014): Dynamical Models for Omni-directional Robots with 3 and 4 wheels. In Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias s/n 4200-465, Porto, Portugal. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

Alexandru BÂRSAN: Position Control of a Mobile Robot through PID Controller. In Lucian Blaga University, Sibiu, Romania.

Ziad Tarik Al-Dahan2, Nasseer K. Bachache1, and Lina Nasseer Bachache2: Design and Implementation of Fall Detection System Using MPU6050 Arduino. In 1University College of Humanities Studies, Najaf, Iraq and 2Al-Nahrain University, Baghdad, Iraq.

BIOGRAPHY

Mr. Rigveda Koranne is pursuing graduation in the domain of Electrical Engineering and is currently working in the field of Robotics Research and Automation.