[[1]](#footnote-1)

Turtle-bot with PID using ROS Velocities Messages

*Mohammed Omair Siddiqui M00841284*

***Abstract*—In the past few years, there has been a massive growth in the field of computer science and robotics, due to which mobile robots or autonomous robots has been a big talk in the market. This document provides a guide a detailed explanations of a two-wheel motor driven wheel with an additional support wheel robot car with a speed control using PID (proportional, integral, and derivative). The setup is consisted of two DC motors which are attached to two wheels, an Arduino board, Proto- shield, an H-bridge and a battery. The PID is used in the speed control mechanism which will offer accurate control of the speed of the robot to the PC/user. The experiment shows the two wheeled robot moving in linear velocity and angular velocity as per the received twist message from ROS.**

***Keywords- PID, Encoder counts, H-bridge, Arduino UNO, DC motors***

# I. INTRODUCTION

There has been a significant advancement in the field of autonomous robot technology during the past 4 decades. Mobile robot technology has been established in the early 1966 which is roughly more than 55+ years. Nils Nilssen was the person at Stanford Research Institute behind the development of ‘Shakey’ which is an autonomous robot which was able to communicate with the environment and act as a responsive robot to the environment. Because of the high rate of accelerated growth in the field of computer science and robotics, the application of mobile robotics has extended from industries, factories to public spaces for instance schools, roads, shopping centers etc. [1]

In today’s date, numerous types of autonomous are available, but based on the locomotion of the robot, they are divided in three; wheeled, chain, floating and walking. [2] Wheeled robot has some advantages, such as light-weight object, In addition to their light weight its simple mechanism, appropriate drive, and flexibility, wheeled mobile robots are frequently used in factories which is filled with human force, delicate missions of space exploration, where their advantages give them way to work more efficiently in the areas which is time and cost sensitive. In this experiment we will be working on a three wheeled robot, which is mainly consist of two wheels which are connected to the motors, and are motor driven and act as a force. These two wheels can achieve differential speed, these wheels can be run at different speeds. The third wheel is rather a steel ball which is not connected to any motor and is not driven by any force and just acts as a support. The steering of this robot is all dependent on the speed provided to the two motors, which can result in linear or angular movement or both at the same time as well in easy terms ‘Speed Control’. PID plays the main role while providing speed control which has been used in many applications and in many companies constantly over the past several years. As can be observed from the Figure 1, the basic setup of the robot can be understood.

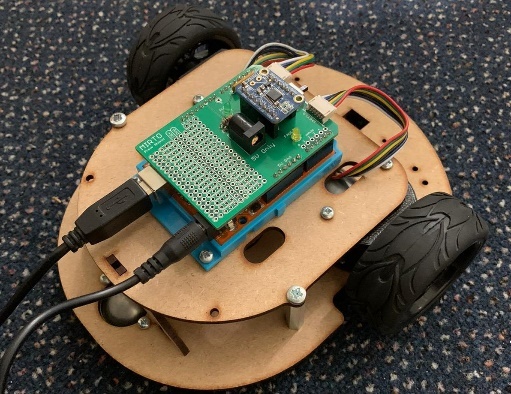


Figure 1: Photo of the Robot

II. Structure of the Robot

The structure of the robot which we are working on, can be defined in layers, each layer consists of equipment according to the need. This robot basically is a two-wheel motor driven robot with a support wheel attached at the rear of the robot for support purpose.

To the left motor and to the right motor, chips are attached in order to record the encoder counts whenever the wheel turns as the result, the speed of the left wheel and the right wheel can be identified and send to the micro-controller. This setup also consists of a proto-shield fixed on top of the Arduino board ProtoShield is designed to facilitate prototyping – this equipment makes it easy to connect wires between a breadboard and an Arduino. Shields are designed to sit on top of an Arduino, extending the functionality of the popular microprocessor platform. An H-bridge is attached on top of the Protoshield which is a simple circuit that lets you control a DC motor to go backward or forward at the same time. The Figure two illustrates where the motor driven wheels are located and the support wheel is located.

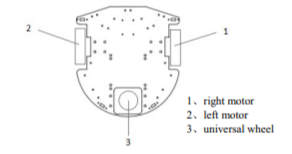


Figure 2: Structure of the robot

III. Encoder Counts

One of the initial stages of this experiment is getting the ***Encoder Counts*** per revolution. And then the circumference of the wheel has to be calculated using formula

~ ,

Where value is 3.14,

And value is the diameter of the wheel which is 0.06m in this case.

~3.14\*0.06

=0.2m

Circumference of the wheel is 0.2m.

The distance between the two wheels is also measured such that the diameter for angular movement generated with a 360 degree turn of the robot car is measured in meters as 0.165m. Now using interrupts and counters, the encoder counts per wheel rotation has been calculated 6 times in order to minimize the margin of error while rotation of the wheel, and hence the encoder counts per wheel rotation is found to be 852 ticks.

***Now Converting encoder count per wheel*** rotation into speed which is per 100 milliseconds. The formula used for conversion is;

ECLinear = lx\*(ticksPerWR/wheelCirc)/10;

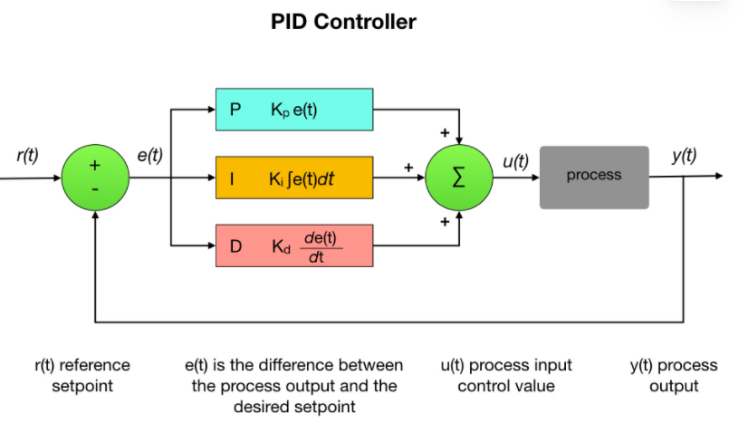
Where, ECLinear is Encoder counts linear velocity, lx is the linear velocity as per the twist message, wheelCirc is the wheel circumference.

ECAng = az\*(ticksPerWR\*(circleCirc/wheelCirc)/(2\*Pi))/10;

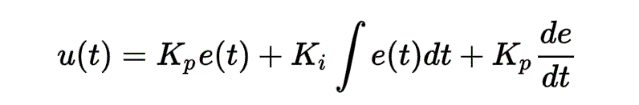
Where, ECAng is Encoder counts angular velocity, az is the angular velocity as per the twist message, wheelCirc is the wheel circumference.

IV.PID

The PID stands for (proprtional, integral, derivative) can be defined as a controller that computes the difference between a desired setpoint and the actual output from a process, and applies the correction based on that difference to the process. [3]



From the above figure PID controller can be understood more clearly. PID is extensively used in solving various major problems in todays teechnical world, the PID expression is defined as;

 [4]

On the robot we are working, as discussed before two highly-sensitive encoders are attached onto the wheels in order to calculate the encoder counts and later converting that to speed. These encoder counts are being sent to the micro-controller, upon receiving the speed (encoder counts), the micro-controller observes and compares the received speed from the encoders with the target speed given by PC/user via serial communication, hence the error of speed control will be identified. The error-margin of speed control is sent to the PID controller, PID controller calculate PWM signal in order adjust the voltage of the motor and hence forming a good control over the motor.

Directing the drive of the motor, the speed of the robot is denoted by ECLinear (linear velocity) and ECAng (Angular velocity) which is measured by the encoder and passes to the micro controllers, which is equated or compared with set-points; SetpointL (set point of the left wheel) and SetpointR (set point of the right wheel) after comparison between the speed and the setpoint,

The set-point of the left and right motor is calculated by the formula;

SetpointL = ECLinear + ECAngL/2 - ECAngR/2;

SetpointR = ECLinear - ECAngL/2 + ECAngR/2;

The error-margin is being sent to the PID controller as discussed before, and in order to fill the error margin, PWM signal of both the motors using the expression formula is measured and directed to the motors.

Using Arduino 1.8.16 software on a computer, serial plotter under tools is opened, the waveforms can be observed on the computer display by opening serial plotter, the PID will do its work to reduce the overshoot of the speed of the motor over the setpoint and smoothens the waveform.

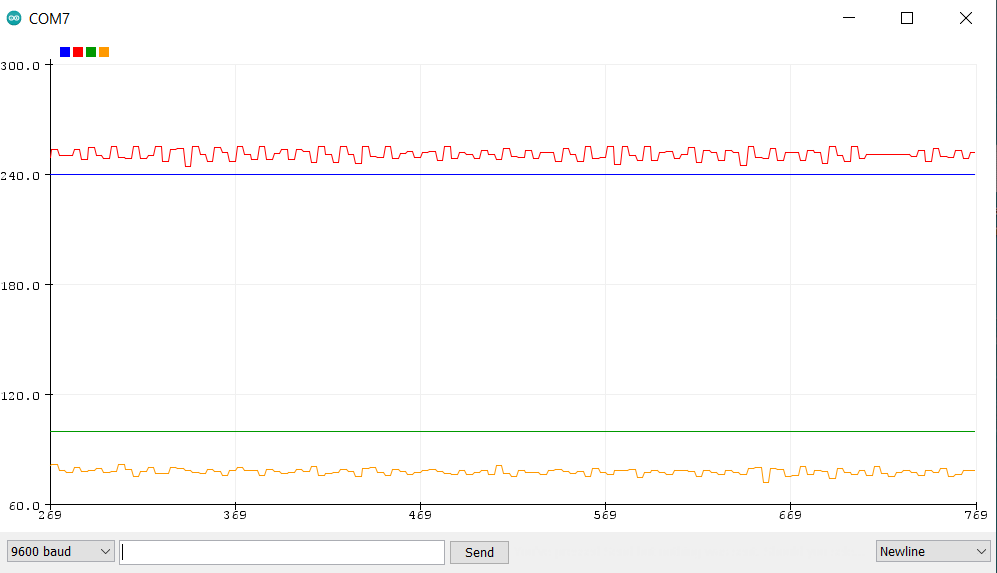


Figure 3: Both motors spinning at speed (0.4,4) (linear velocity, angular velocity)

From the Figure 3, it can be observed that both motors are spinning linearly and angularly. The linear and angular speed of the motor is maintained near the specified setpoint.

V. Velocities and Direction

There are two types of velocities involved in this experiment, linear velocity and angular velocity, which results the robot to move in forward or curve or both at the same time leading to a curve drive.

1. Straight Drive

In order to get the straight drive of the robot, we will run some tests where the two wheels will rotate in opposite direction such that; if one-wheel spins clock wise the other wheel should be spinning anti-clockwise in order to get the straight drive, both speeds should be exact same to avoid any angular movement. Added to that, in the program written in Arduino 1.8.16, if the linear velocity from the twist message in positive value the movement will be in forward direction, and if negative the direction will be backward direction.

1. Curve drive

Curve drive is a bit more complex than straight drive; curve drive is combination of linear movement and angular movement. In the program, we ran some test giving the robot both the velocities in twist message. In order to make the robot to take a curve drive the speed of the wheels has to be different such that, one wheel has to possess higher speed than the other wheel which will in return give an angular movement (curve drive) in radians along with linear velocity. When the angular velocity from the twist message is given to robot the setpoint is defined in a way that half of the angular velocity of given value gets added to one wheel and half of the same value is reduced from the other wheel in order to get the angular movement.

The expression to find the setpoint is as follows:

SetpointL = ECLinear + ECAngL/2 - ECAngR/2;

SetpointR = ECLinear - ECAngL/2 + ECAngR/2;

In Arduino 1.8.16 software, if the linear velocity from the twist message is given in positive and the angular velocity is positive as well then, the direction of the robot will be forward with clockwise curve, if linear velocity positive and angular velocity negative, then the direction of the robot will be forward with anti-clockwise curve.

Whereas if the if we keep the linear velocity negative and enter angular velocity positive and negative respectively then the direction of the robot will be backward with clock wise and backward with anti-clockwise directions respectively.

But when the linear velocity was given 0 and the angular velocity was entered with some radians, instead of two motors spinning in two different directions, only one wheel was rotating. This is an issue yet to be resolved.

VI. My Work

My contribution in this experiment was calculating encoder counts per wheel rotations and converting into speed. As we divided all the work into parts and assigned them to each of the group members. I recorded the number of encoders counts per wheel rotations on both the wheels which was 300-400, however later after some group sessions and interactions with the other group members recalculated to 852 encoder counts per wheel rotations and converted that to speed using a formulas;

ECLinear = lx\*(ticksPerWR/wheelCirc)/10;

Where, ECLinear is Encoder counts linear velocity, lx is the linear velocity as per the twist message, wheelCirc is the wheel circumference.

ECAng = az\*(ticksPerWR\*(circleCirc/wheelCirc)/(2\*Pi))/10;

Where, ECAng is Encoder counts angular velocity, az is the angular velocity as per the twist message, wheelCirc is the wheel circumference. And hence the encoder counts per 100 milliseconds was measured. Furthermore, several group discussions, brainstorming and meetings at the university library and public library in order to overcome the challenges to resolve issues related to the PID.

V. Gitlab link

https://labcode.mdx.ac.uk/msc-robotics/turtle-twister

VI. Conclusion

In this experiment, a robot that can move linearly or angularly or both at the same time such that; curve drive is created. The encoders are attached to the motors which record the encoder counts, and this data is sent to the micro-controller. The PID controller now compares the received encoder speed with the specified setpoint and then supply appropriate voltage. The robot successfully moved in straight direction and curve drive.

But when the linear velocity was given 0 and the angular velocity was entered with some radians, instead of two motors spinning in two different directions, only one wheel was rotating. This is an issue yet to be resolved.

# References

|  |  |
| --- | --- |
| [1] | J. Meng, A. Liu, Y. Yang, Z. Wu and Q. Xu, "Two-Wheeled Robot Platform Based on PID Control," in *5th International Conference on Information and Control Engineering* , 2018. |
| [2] | C. X. P. C. a. M. S. Nie, "Robots on the move: Versatility complexity in mobile robot locomotion," *IEEE Robotics Automation Magazine ,* vol. 4, pp. 72-82, 2013. |
| [3] | M. WHITE, Programming Robots, 13 7 2019. [Online]. Available: https://mjwhite8119.github.io/Robots/pid-control. |
| [4] | D. K. Alexis, "PID Control," [Online]. Available: http://www.kostasalexis.com/pid-control.html#:~:text=PID%20Control%20stands%20for%20Proportional,used%20controllers%20used%20in%20industry.&text=The%20proportional%20term%20corresponds%20to,derivative%20to%20the%20last%20one.. |

1. [↑](#footnote-ref-1)