Differential Drive Adaptive Proportional-Integral-Derivative (PID) Controlled Mobile Robot Speed Synchronization

S.M. Rubayat Islam[a] Shanjida Shawkat[b], Sumit Bhattacharyya[c], Md. Khaled Hossain[d]
[a b] Dept. of Electrical and Electronic Engineering, American International University-Bangladesh
[c] Lecturer, Atish Dipankar Biggan O Projukti Bissobiddaloy
[d] Lecturer, Robotics Bangladesh

ABSTRACT

In this paper we present the implementation of proportional integral derivative (PID) controlled mobile robot. we propose a mobile robot control method which based on optical encoder. First, the mobile robot counts the number of pulses from optical sensor. This system rejects the denouncing and optical sensor noises and makes a precise count of both the wheel rotation at the same time. Then through an adaptive PID control loop it controls the speed of both the motors to reach user defined set point. PID auto tuning methods are performed in different situations like on air, on ground and with load increase. This system detects the situation the robot at present and switches between its different modes to achieve the quickest maximum efficiency. Speed of each motor is controlled by Pulse Width Modulation (PWM). Finally, we experiment in a real environment, and verify the utility of the propose method.

Keywords

Differential Drive Robot (DDR), PID control, Mobile robot.

1. INTRODUCTION

The motion control problem of mechanical systems with non-linear constrain has been a heavily researched area due to both the theoretical nature of the problem and its practical implementation. Example of a non-holonomic system that has a large scope of research activity is the Differential Drive Robot (DDR). A differential-drive wheeled mobile robot is a highly nonlinear multi-input/multi-output system. Traditional techniques can not result in a control system design with satisfactory performance. [1] Some paper proposed solutions using wheel encoder [2]. But the problems with encoder are precise encoders cost a lot and cheap encoder has lower resolution which is not suitable for a robot control system. In this paper we have proposed a solution of this problem. Forward kinematics and inverse kinematics are heaviely dependent on encoder value. Encoder input part is not the only problem here, individual motors differ in their rpm with the change of pulse width modulation (PWM). A motor does not changes its rpm linearly with pulse width modulation. Thus we used Proportional Integral Differential (PID) control loop to solve this problem. But for a robot the parameters can not be fixed because robots has to change its speed continuously. For example to strait forward it goes at its best speed but at the point of turn it slows itself down and minimize the turning error. So the system has to be adaptive with these situations. Sometime the system goes to aggressive PID control mode when the error is higher and come back to its normal PID control mode when the error is small.

2. OVERVIEW

A differential drive is the most basic drive, which consists of two sets of wheels that can be driven independently. This is the most commonly used form of locomotion system used in robots. It has free moving wheels in the front accompanied with a left and right wheel. The two wheels are driven by different motors. Independent drives make it difficult for straight line motion. The differences in motors and frictional profile of the two wheels cause them to move with slight turning effect. The above drawback must be countered with appropriate feedback system.

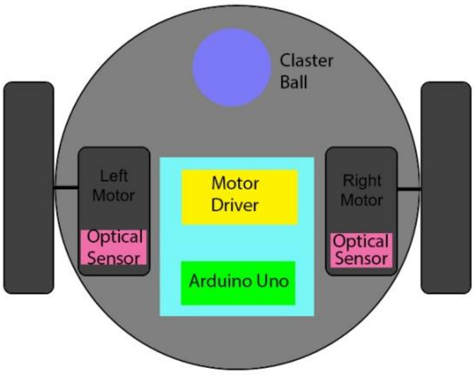


Fig1. Differential Drive Wheeled robot architecture

This differential drive wheeled robot consists of different portion. Controller of this wheel robot is 8 Bit microcontroller. In order to drive each motors a motor driver circuit is used as an encoder optical sensors are mounted closer to the wheels on top of motors. The whole system is powered by 12 V battery. In order to support the front part of the robot there is a cluster ball is attached which can move in any direction.

MOTION	LEFT WHEEL	RIGHT WHEEL
Right Turn	Counter Clockwise	Clockwise
Left Turn	Clockwise	Counter Clockwise
Forward	Clockwise	Clockwise
Backward	Counter Clockwise	Counter Clockwise

2.1 Circuit Diagram

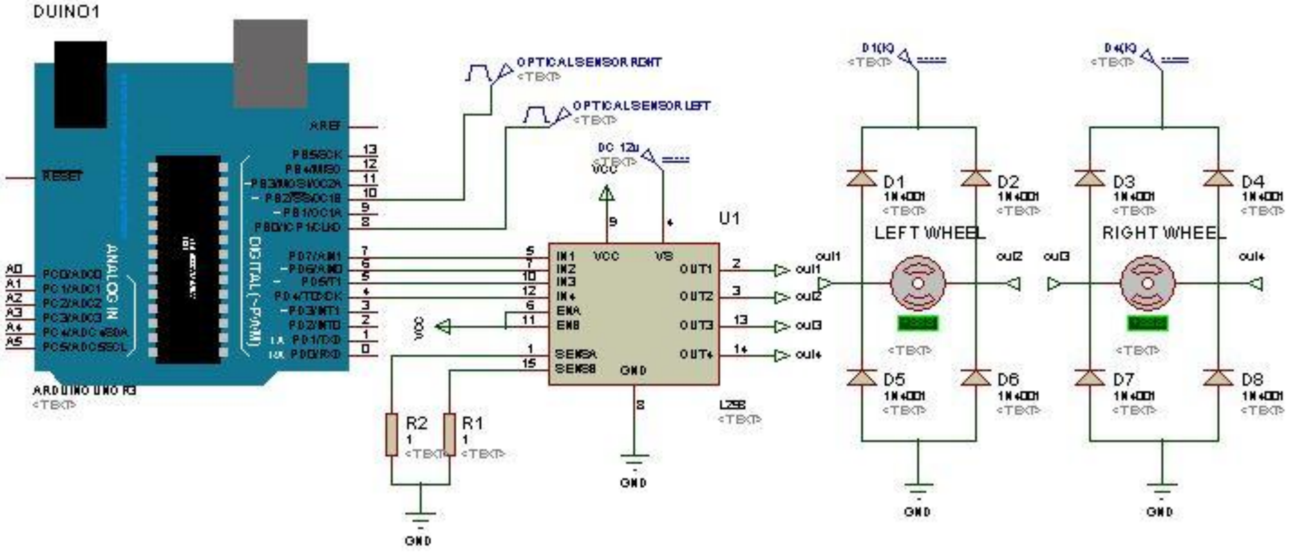


Fig.2 Circuit Diagram of Wheeled Robot

Arduino Uno consist of multiple digital and analog input output pins among this digital pins there are six pulse width modulation pins. Optical sensors are connected to general digital input pin no 8 and 10. Microcontroller is capable of counting digital pulses through these two pins. Direction of motors can be controlled by any digital pin but in order to control the speed of motors they must be connected to pulse width modulation pins. Arduino pulse width modulation pins are 3, 5, 6, 9, 10, and 11. All of these pins can produce 8 Bit pulse width modulation output at same frequency. So the resolution of pulse width modulation is from 0 to 255. In order to control the speed of a motor in both direction one of the controlling pin must be connected to a pulse width modulation pin. Thus controlling pin of both the motors are connected to 4, 5, 6, and 7 respectively. Each motors rating is 12V 1A. To supply this amount of current L298H-Bridge IC is used. There are four free will diode connected with each motor to protect the IC from flyback current.

Here two digital optical sensors are used as encoder. A printed 16 black and white strips are connected with each wheel. Whenever IR light reflects from a white strip IR receiver generates an analog voltage which is feed to a comparator operational amplifier. A voltage reference is feed to the other pin of the comparator. Bias voltages of this comparator are +5V and 0V. Output of this comparator is pulled down by a 10K resistor. So whenever IR reflects back from a white strip comparator generates +5V to its output and whenever IR doesn’t reflect from black strip it produce 0V to its output. Each output of the comparators are connected to pin 8 and 10.

The L298 is an integrated monolithic H-Bridge motor controller circuit. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

In 1, In 2, In 3, In 4 pins of L289 IC are connected with pin no. 7, 6, 5, and 4 respectively. EnableA (ENA) and enableB (ENB) controls the outputs. These two pins are shorted and connected to +5V. This makes the H-Bridge circuit active. SensA and SensB pins controls the maximum amount of output current. Two 1 Ohm resistors are connected to these pins for maximum current output. As the two motors are rated +12V, VS pin is connected to a 12V source. To protect this IC in total 8 diodes are used in two motors. For example if left motors generates a flyback voltage greater than +12V diode D1 gets turn on and the over voltage goes into the batteries. If motor generates lower voltage than -12V then diode D3 turns on and extra reverse voltage goes to ground. In this way Out1, Out2, Out3, Out4 all these pins are protected. Arduino Uno runs on +5V, so a voltage regulator IC (7805) is mounted on the board. Vin pin of Arduino Uno is connected with positive terminal of +12v Battery and the negative terminal is connected to ground pin.

3. HARDWARE IMPLEMENTATION

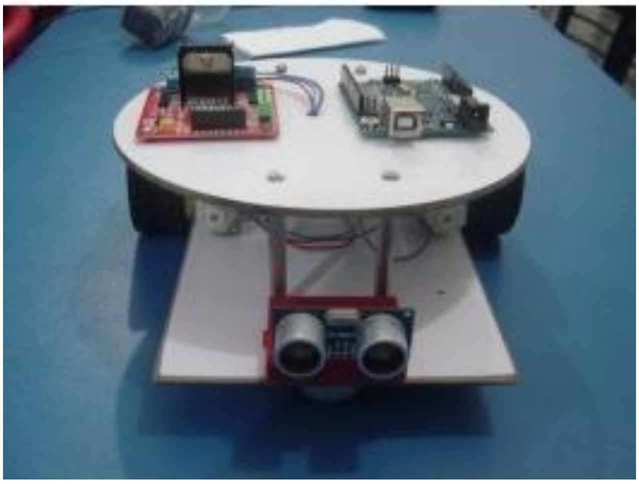


Fig3. Hardware Implementation

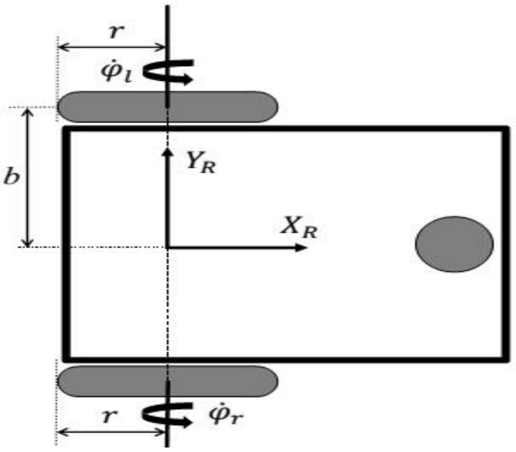


Fig4. Hardware Architecture of Robot

Radius of each wheel, r= 3.15cm and the distance of each wheel from the center, b=11cm. Forward and inverse kinematics equation for differential robot are as follows.

$$\begin{bmatrix}\dot{\varphi}_r\\ \dot{\varphi}_l\end{bmatrix} = \begin{bmatrix}1/r & 0 & b/r\\ 1/r & 0 & -b/r\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}$$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}r/2 & r/2\\ 0 & 0\\ r/2b & -r/2b\end{bmatrix}\begin{bmatrix}\dot{\varphi}_r\\ \dot{\varphi}_l\end{bmatrix}$$

So plugin the values of r and b we get the following values

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}1.6 & 1.6\\ 0 & 0\\ 0.14 & -0.143\end{bmatrix}\begin{bmatrix}\dot{\varphi}_r\\ \dot{\varphi}_l\end{bmatrix}$$

$$\begin{bmatrix}\dot{\varphi}_r\\ \dot{\varphi}_l\end{bmatrix} = \begin{bmatrix}0.31 & 0 & 3.49\\ 0.31 & 0 & -3.49\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}$$

These two equitation's are used to calculate the forward and inverse kinematics of the robot.

3.1 Optical Encoder

To measure the rpm of each wheel a custom made encoder is used. A 16 strip disk is attached to the wheel in the following manner.

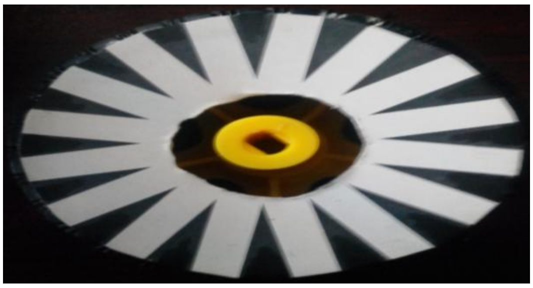


Fig5. Encoder disk mounted on a wheel

Optical sensor is mounted on top of the motor which provides a clear reflection from the disk. The optical sensor is a 3 pin digital sensor. The figure of optical sensor mounted on motor is shown below.

Fig6. Optical encoder

Optical encoder supposed to give high pulse during white strip and low pulse during black strip. But in practical optical sensor debounces a lot. Fast type of debounce occurs at the transition from white to black strip. Second type of debounces occurs from the printed black strip. Logic analyzed figure of the bounces are shown below.

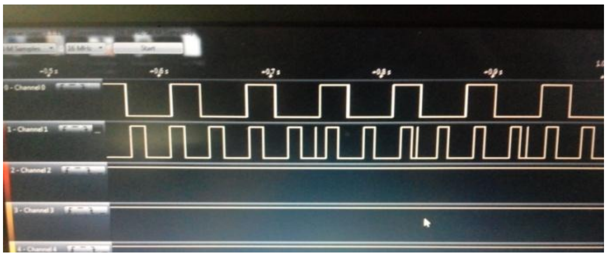


Fig7. Logic Analysis of optical sensor

In typical encoders this debounces are calculated as high pulse which causes accumulated errors in pulse counting. This debounces needs to be handle precisely. A debouncing delay may not work if the debouncing occurs at the middle of the pulse. So we proposed a new way of elimination of the bouncing error. We calculate the pulse high time duration. From our observation we found that the duration of an actual pulse is from 10ms to 20ms. But the duration of debounces are less than 1ms. The following figure shows the time duration of a pulse and the debounce.

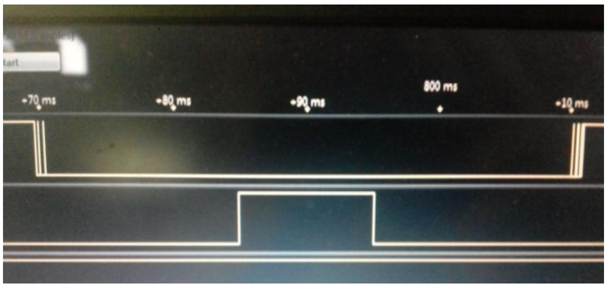


Fig8. Analysis of debouncing duration

So elimination of all the pulses whose duration is less than 1ms makes the pulse counting error less.

3.2 Speed Control of DC Motor

The speed of DC motor is controlled through pulse width modulation. Arduino Uno is capable of generating 8 bit pulse width modulation. So the resolution of pulse width modulation is 0 to 255. When the value of PWM is 255 motor runs at full speed. When the PWM is half of its maximum value motor rotates at half of its maximum speed. But a fix value of PWM does not generate a fix rpm of a motor. As the battery voltage decays the speed of the motor also decrease. To keep the motor at a constant speed a close loop PID control system needs to be applied. The equations of PID control loop are shown below.

Error = Setpoint – Input ; // Calculate error

Errorsum = Errorsum + Error ;

Derror = (Input – Lastinput) ; // Calculate slope of the input

Pout = Kp * Error ; // Calculate PID gains

Iout = Ki * Errorsum ;

Dout = Kd * Derror ;

Output = Pout + Iout + Dout ;

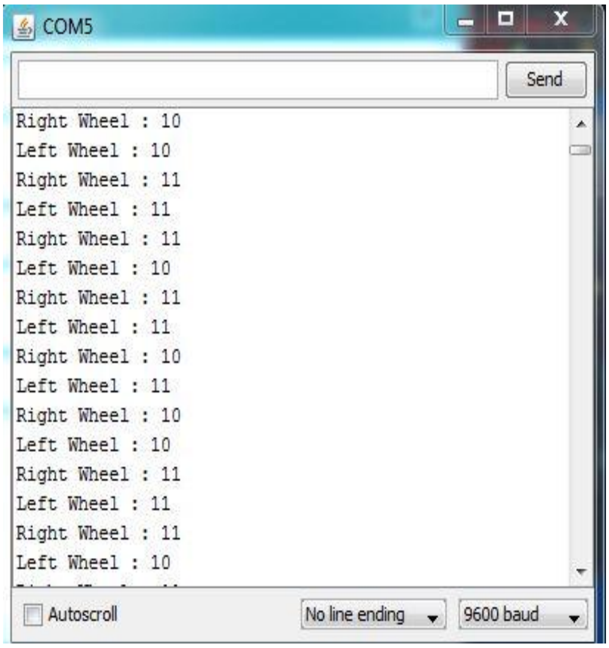
The value of Kp, Ki and Kd are tuned through software auto tuning mechanism. Value of Kp = 8, Ki = 3 and Kd = 1.

Before applying PID control loop both the motors speed need to be measured quickly. A time limitation of 500ms gives a fine resolution of motor speed. We got around 20 pulses in 500ms for motor 1 and 24 pulses for motor 2. Here calculating both the pulses at the same time frame is very important. Instead of counting the pulses separately we measured the pulses from both the pins at the same time. The debouncing elimination process also performed within 500ms. This technique makes the PID calculation time difference constant. Which gives us higher stability abs accuracy in PID control loop.

To get a quick and stable output we choose adaptive PID control loop technique. For Different amount of errors different values of Kp, Ki and Kd are chosen. If the difference between set point and present rpm is large we approach with aggressive PID control loop. And for small error the values of Kp, Ki and Kd are much smaller. This feature gives us the freedom to take sharp turn with the robot.

4. RESULTS

For different set point we tasted the PID control loop. For a time interval of 1000ms at PWM 250 we get around 24 pulses. Sampling for 1000ms is a large amount for differential drive robot. So we took sample for 500ms time period. So the input of PID loop is number of pulses in 500ms and the output is PWM. For a set point of 10 pulses we ran PID loops for both the motors and found the following results.



```
Right Wheel : 10
Left Wheel : 10
Right Wheel : 11
Left Wheel : 11
Right Wheel : 11
Left Wheel : 10
Right Wheel : 11
Left Wheel : 11
Right Wheel : 10
Left Wheel : 11
Right Wheel : 10
Left Wheel : 10
Right Wheel : 11
Left Wheel : 11
Right Wheel : 11
Left Wheel : 10
```

Fig9. Output of PID loop

From this results we can come to a conclusion that both the motors speed are pretty close. For a 16 strip disk we could not minimize the fluctuation of 1 pulse.

5. CONCLUSION

PID control loop is the most efficient way of controlling speed of motor. But nullifying errors from the input is also crucial. All the future work of robot like odometry, inertial navigation system, object tracking, simultaneous localization and mapping depends on this work. We can develop our output using better motors. The number of strips can be increase also. Instead of using 8 Bit PWM, use of 16 Bit PWM will certainly make the performance of this control loop better. This control loop can be implemented in four wheel drive robots. Any steeper motor can also implement this system.

6. REFERENCES

[1] Shouling He, Feedback control design of differential-drive wheeled mobile robots in Advanced Robotics, 2005. ICAR '05. Proceedings. 12th International Conference on 18-20 July 2005 pages135 – 140.

[2] Xiaokang Song ; State Key Lab. of Robot., Chinese Acad. of Sci., Shenyang ; Yuechao Wang ; Zhenwei Wu. Kinematical model-based Yaw calculation for an all-terrain mobile robot in Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on 2-5 July 2008.

[3] W. E. Dixon, D. M. Dawson, E. Zergeroglu and F. Zhang "Robust tracking and regulation control for mobile robot" on International journal of robust and nonlinear control.

[4] W. E. Dixon, D. M. Dawson, E. Zergeroglu and Aman Behal "Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system" on IEEE transactions on systems, man and cybernetics-Part B: cybernetics, Vol. 31, No. 3, June 2001.

[5] Gregor Klancar, Drago Matko and Saso Blazic "A control strategy for platoons of differential drive wheeled mobile robot" on Robotics and Autonomous System-Journal of ELSEVIER.

[6] Kiam Heong Ang, Yokogawa Eng, Chong, G. and Yun Li "PID control system analysis, design, and technology" on Control Systems Technology, IEEE Transactions.

[7] Chen Guilan ; Wen Xuhui and Zhang Liwei "A coordinate double-loop PID pressure control method for PEMFC system" on Electrical Machines and Systems, 2003. ICEMS 2003. Sixth International Conference