Analysis of DC Motor Speed Control and Interfacing it with Drowsiness Detection

Atharva Sulakhe
Department of Communication Engineering,
School of Electronics Engineering,
Vellore Institute of Technology,
Vellore 632014, Tamil Nadu, India.
sulakheatharv@gmail.com

Arjeet Anand
Department of Communication Engineering,
School of Electronics Engineering,
Vellore Institute of Technology,
Vellore 632014, Tamil Nadu, India.

Akshwin Kisore
Department of Communication Engineering,
School of Electronics Engineering,
Vellore Institute of Technology,
Vellore 632014, Tamil Nadu, India.

Nandha Kishore R B
Department of Communication Engineering,
School of Electronics Engineering,
Vellore Institute of Technology,
Vellore 632014, Tamil Nadu, India

Malayakumar Hota
Department of Communication Engineering,
School of Electronics Engineering,
Vellore Institute of Technology,
Vellore 632014, Tamil Nadu, India.
malayakumar.h@vit.ac.in

***Abstract*— Motors have the primary purpose of driving a load. As a result, they have a wide range of applications varying from vehicles and factories to numerous other traction systems. While some industries require constantly high torque, others prefer variable speeds and thus, the need for speed control arises. DC motors, in particular are being preferred over AC for such applications as they have relatively easy speed control systems. One such method is using controllers such as P, PI, PD and PID. In this paper, an attempt has been made to analyze the working of different controllers and prove why PID control is the preferred controller to vary the speed. Further, this control system has been interfaced with a drowsiness detection algorithm in order to reduce the speed of a car in case the driver is found to be drowsy.**

Keywords*— PID Control, DC Motor, Arduino, Drowsiness Detection, OpenCV.

I. INTRODUCTION

DC motors are used in some electric vehicles due to their higher torque and smooth handling of heavy loads. High starting torque and the ability to withstand sudden overshoot in load makes it a viable option [1]. There are quite a few ways to vary and control the speed of DC motors including –

1. Flux control method
2. Armature control
3. Voltage control

In this paper we focus on P, PI and PID controllers and analyse why PID is most broadly used [2].

The software used is MATLAB, which provides easy implementation of algorithms and superior graphics compared to other programming languages. This analysis has then been extended to devise an innovative solution to the problem of road accidents due to drowsiness while driving.

II. P CONTROL

The simplest and most straight-forward among these is Proportional or P control. In this, there is a constant relation between the actual output of the system and the error signal (deviation). It provides an output that is directly proportional to this error e(t). Here it compares the actual value with the required or desired value. The deviation in the output is then multiplied with a proportional constant to get an output. This constant is known as Kp [3]. It never reaches the steady state condition and thus it is generally used in biasing. Yes, it does provide stable operation but doesn't remain steady state error-free.

Since it has only one tuning parameter, it is simpler to tune for better performance in comparison with PI, PD and PID controllers which have multiple parameters. The main drawback of this kind of control is that it reaches its limit and cannot react to prolonged changes to a control loop's operation level. Thus, the lags in this method might not be suitable for a lot of applications.

The P-Only Algorithm-

The P-Only controller computes a CO action every loop sample time T as:

Controller Output = Biased Controller Output + Kp*e(t) (1)

Kp = controller gain (tuning parameter)

e(t) = controller error = SP – PV; SP = set point, PV = measured process variable. (2)

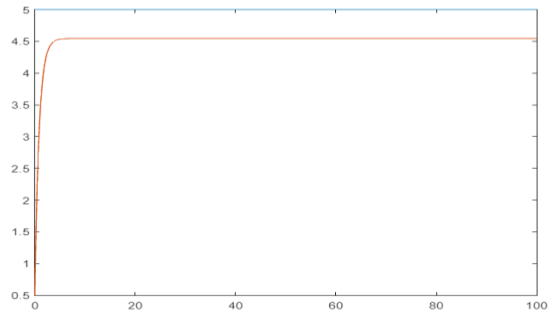

Fig. 1. Graphical output for P controller where the output was 4.5 even though the setpoint was set at 5.

III. PI CONTROL

PI controllers have 2 parameters that need to be calculated and they provide good results in most control systems. PI control is needed for non-integrating processes, that is any process that eventually returns to the same output considering that it was given the same input and disturbances. The integral action is mainly used to remove offset and is adjustable. To calculate the power drained from the battery, the current and voltage received by the controller is used. The difference between the set point and the measured power is used to find the error signal. Now the error signal passes into P.I control loop. After getting multiplied with proportional and integral multiple, the output is a power value.

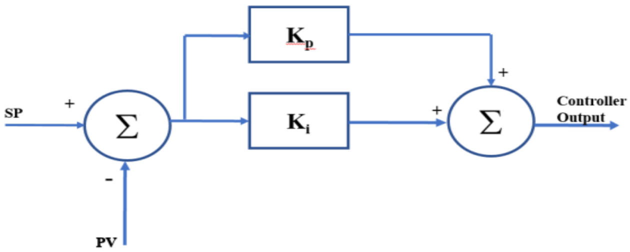

Fig. 2. Block diagram of PI Controller where the plant includes the Proportional and Integral parts.

The primary disadvantage of PI, however is that it has high starting overshoot and a sluggish response to sudden changes. It is also highly sensitive to controller gains.

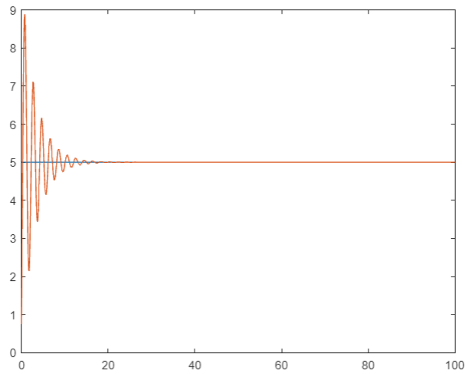

Fig. 3. Graphical output for PI control where we see gradual retardation to the setpoint=5.

IV. PID CONTROL

Proportional, Integral and Derivative control action is the PID Controller which is now most commonly used in dynamic works to regulate the time behavior and in part of Mechanical Control Frameworks. This PID Controller combines all the 3 algorithms - Proportional Mode, Integral Mode and Derivative Mode. It uses control loop feedback Mechanism to control the process variables and is considered to be the most stable controller. The Proportional Mode calculates the difference in current between the actual speed and the desired speed which is actually known as errors. The Integral Mode does the adjustment based on many recent errors. The Differential Mode determines response as indicated by the rate off blunder evolving. These three works together which contribute to control the environment i.e., P works on the past mistake, I on the gathering of past mistake and D anticipates the future.

The Main Goal of PID Controller is to make the actual motor speed match the desired motor speed.

Whenever the PID algorithm is activated, it will use the motor's sensors to monitor its actual speed. This actual speed is compared with the desired speed, and then the algorithm will calculate the power changes that are necessary to get the actual speed and will make it equal to the desired speed. Now the algorithm starts again and compares the new actual speed to the desired speed. Based on this, it actually makes further requirements if necessary to the motor's power. This now becomes a cycle where the speed of the motor is constantly checked against the desired speed and power level of the motor is being changed and to get the desired result [4,5].

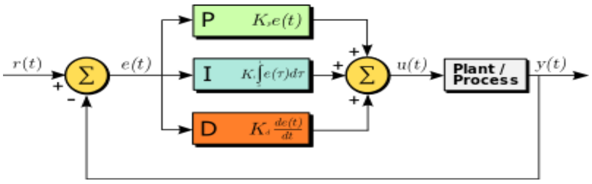

Fig. 4. PID Control Block diagram.

The above figure shows a feedback loop system where the plant is a DC Motor whose speed must be controlled. So here the PID controller is placed in the forward path, such that the output becomes the voltage applied to the motor's armature. The output velocity signal is summed with a reference to form the error signal. Finally, the error signal is the input to the PID controller [6,7].

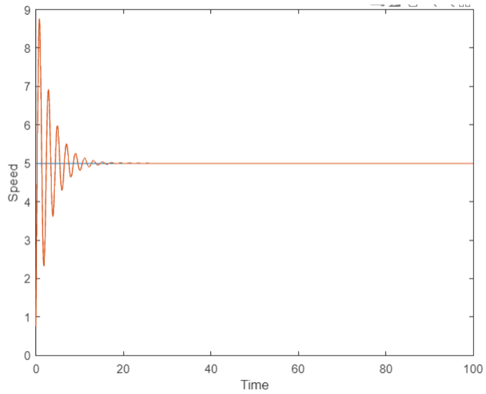

Fig. 5. Graphical output of PID Control.

The above graph was generated for Kp = 0.05, Kd = 0.1 and Ki = 0.1. Here, the graphical output clearly shows a constant retardation over time using PID Controller where the speed is brought down to 5.

V. INFERENCE

Table 1. The effect of the 3 controllers on different parameters, whilst there is a variation in rise time [8].

Parameter	P Control	PI Control	PID Control
Stability	Decreases	Decreases	Better if Kd is small
Overshoot	Increases	Increases	Almost same
Rise time	Decreases	Decreases	Almost same
Settling time	Almost same	Increases	Almost same
Steady state error	Decreases	Significant change	Same

Thus, on analysing the 3 controllers, it can be inferred that PID controllers provide a better response as compared to P and PI. In PID, there is no change in steady state error over time, whereas the same cannot be said about the other two. The overshoot, rise time and settling time also remain almost the same with only minor changes in PID.

VI. HARDWARE

In the Hardware, we implemented an Arduino-based control of DC motor.

The components involved were-

1. Arduino UNO
2. L293D Motor Driver
3. DC Motor
4. Wires

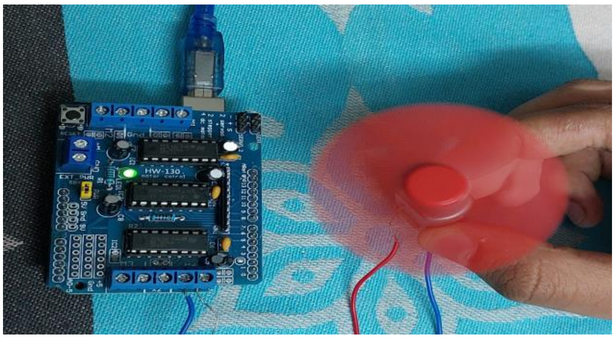

Fig. 6. Hardware

In the hardware made, the L293D motor driver was placed directly on the Arduino. This was to enable simpler connections. We wrote a code on Arduino IDE using the motor driver library and managed to gradually decrease the speed from 225 to 15 (as shown in Fig. 7). This process kept on happening in a loop [9].

```
255
225
195
165
135
105
75
45
15
255
225
195
165
```

Fig. 7. Gradual reduction in motor speed

VII. DROWSINESS DETECTION

By definition, drowsiness is a state of sleepiness where the person needs to rest. As a result, it leads to lack of awareness, slowed response time or microsleeps (blinks of over 0.5s) [10]. Numerous surveys in the past have shown that around 20% or a fifth of all fatal accidents take place due to drowsiness of the driver. These surveys also predict that around 60% of all drivers have driven a vehicle while feeling drowsy [11,12]. Looking at these statistics, we felt that there was an acute need to make an attempt to solve this long-standing problem using technology.

Hence, we have tried to find an innovative solution to improve road safety by automatically reducing the speed of the vehicle if the driver is detected to be feeling drowsy or has low alertness level [13].

The idea is to have a miniature camera near the speedometer of the vehicle which is programmed to detect alertness levels of the driver. If this level goes below a certain limit, then the above motor speed control system is activated which automatically reduces the speed of the car, thus minimizing the risk from crashes.

In terms of working of the system, images captured by the camera are fed as input (Deep learning) into a dataset. From the face captured, a Region of Interest (ROI) will be created. From the ROI, the position of the eyes will be detected and be fed into a classifier. This classifier will then determine whether the eyes of the person are open or closed and calculate the drowsiness score. If this score is beyond a certain threshold, then in that case, the motor speed control system gets triggered and the speed of the vehicle will be automatically reduced.

For the code, the following packages in Python were used –

OpenCV – Face and eye detection

TensorFlow – Keras uses TensorFlow as backend

Keras – To build our classification model

Pygame – To play alarm sound

A. STEPS/ ALGORITHM

1. To access the webcam, an infinite loop is made which will capture each frame. This loop can be created by the OpenCV in-built function cv2.VideoCaopture(0) which will access the camera and set the capture object. The function cap.read() will read and store each frame as an image.

2. Since the OpenCV algorithm takes images in the grayscale only, we have to convert our captures into the grayscale format. Using the Haar Cascade Classifier, a Region of Interest (ROI) is created [14].

3. This procedure of detecting faces is used to detect the eyes. Then, we feed this to the Classifier which will determine if the eyes are closed or open. This task will take place for each eye one by one.

4. Now, the Classifier will resize the image to the conventional 24*24 pixels first and then categorize the eyes into open or closed.

5. A drowsiness score is then calculated as follows-

If both eyes are open, the score will keep on decreasing. If both are closed, the score will start increasing. Now, if this score crosses a certain threshold, we can conclusively say that the person is drowsy.

The following are the outputs of program- [15]

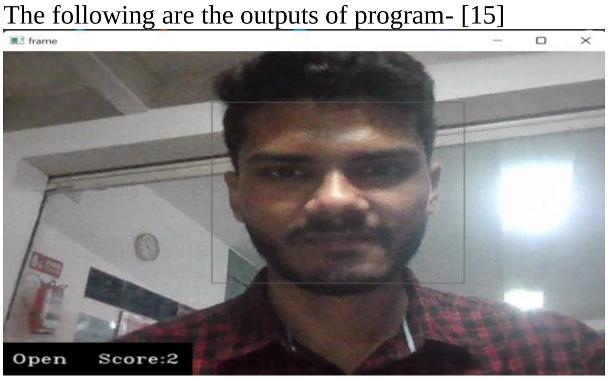

Fig. 8. Eyes open. Not drowsy

Fig. 9. Eyes closed. Drowsy

In Fig. 8, the eyes of the driver are open and he is alert. Thus, we get an output saying open score: 2.
In Fig. 9, the eyes of the driver are completely closed, showing that he is feeling drowsy. Thus, we get an output saying closed score: 76 and a red border shows an alert. A higher score corresponds to lower alertness level.

VII. CONCLUSION

In this paper, we implemented an analysis of P, PI and PID controllers using MATLAB software and concluded that PID controller is better than the others. We also constructed an Arduino-based hardware using a motor driver to control the speed of the DC motor. Finally, we used this concept to innovate a solution to reduce car crashes due to driver drowsiness wherein a python-based algorithm would detect the alertness level of the driver and if it is below a certain threshold, the speed of the motor will automatically be reduced and risk of accidents will be minimized, thus enhancing road safety.

REFERENCES

[1] Khan M.R., Khan, A.A., Ghazali, U.: Speed Control of DC Motor under Varying Load Using PID Controller. International Journal of Engineering (IJE) (2015)

[2] Achanta, R.K., Pamula, V.K.: DC motor speed control using PID controller tuned by java optimization algorithm. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI) (2017)

[3] Rao, K.S., Mishra, R.: Comparative study of P, PI, and PID controller for speed control of VSI-fed induction motor. IJEDR Volume 2, Issue 2, June (2014)

[4] Nayak, A., Singh, M.: Study of Tuning PID Controller by using Particle Swarm Optimization. International Journal of Advanced Engineering research and Studies, E-ISSN2249-8974.

[5] Hamoodi, S.A., Mohammed, R.A., Salih, B.M.: DC Motor Speed Control Using PID Controller Implementation by Simulink and Practical. International Journal of Electrical Engineering. ISSN 0974-2158 (2018)

[6] Singh, A.P., Narayan, U., Verma, A.: Speed Control of DC Motor using PID Controller Based on Matlab. Inoovative Systems Design and Engineering, ISSN (2013)

[7] Tesfamikael, H.H., Fray, A., Mengsteab, I., Semere, A., Amanuel, Z.: Construction of Mathematical Model of DC Servo Motor Mechanism with PID controller for Electric Wheel Chair Arrangement. Journal of Electronics and Informatics (2021)

[8] Loucif, F.: DC Motor Speed Control Using PID Controller. ICCAS (2005)

[9] Vikhe, P.S., Punjabi, N., Kadu, C.B.: DC motor speed control using PID controller in lab view. International Journal of Innovative Science and Modern Engineering (IJISME) (2015)

[10] Magán, E., Sesmero, M.P., Alonso-Weber, J.M.; Sanchis, A.: Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images. Appl. Sci. (2022)

[11] Navya-Kiran, V.B., Raksha, R., Rahman, A., Varsha, K.N., Dr. Nagamani N.P.: Driver Drowsiness Detection. INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT (2020)

[12] Saini, V., Saini, R.: Driver drowsiness detection system and techniques: a review. International Journal of Computer Science and Information Technologies (2014)

[13] Hong, T., Qin, H.: Drivers Drowsiness Detection in Embedded System. IEEE Xplore (2008)

[14] Jain, M., Bhagerathi, B., Sowmyarani, C.N.: Real-Time Driver Drowsiness Detection using Computer Vision. IJEAT, Volume 11, Issue-1 (2021)

[15] Forsman, PM., Vila, BJ., Short, RA., Mott. CG., Van Dongen, HP.: Efficient driver drowsiness detection at moderate levels of drowsiness. (2013).