

Improving steering convergence in autonomous vehicle steering control

Amir Ashraf Mohamad, Fadhlan Hafizhelmi Kamaru Zaman, Fazlina Ahmat Ruslan

Faculty of Electrical Engineering, Universiti Teknologi Mara, 40450 Shah Alam, Selangor, Malaysia

Article Info

Article history:

Received Aug 20, 2018

Revised Oct 28, 2018

Accepted Nov 12, 2018

Keywords:

Arduino mega

PID controller

PWM

Vexta motor

ABSTRACT

Steering control is a critical design element in autonomous vehicle development since it will determine whether the vehicle can navigate safely or not. For the prototype of UiTM Autonomous Vehicle 0 (UiTM AV0), Vexta motor is used to control the steering whereas Pulse Width Modulation (PWM) signal is responsible to drive the motor. However, by using PWM signal it is difficult to converge to the desired steering angle and furthermore time taken for steering angle to converge is much longer. Thus, Proportional Integral Derivative (PID) has been introduced in this autonomous vehicle steering controller to improve the convergence of the steering. Meanwhile a microcontroller was used to control the Vexta Motor direction and perform the calculation of the desired steering angle. Simulation results showed PID controller showed better time taken and precision of successful convergence of the desired steering angle compared to the PWM controller. Analysis results showed that PID controller significantly reduce the overshooting of steering angle and significantly improve the time taken for convergence by up to 37 seconds faster than PWM controller in UiTM AV0.

*Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Fadhlan Hafizhelmi Kamaru Zaman,
Faculty of Electrical Engineering,
Universiti Teknologi Mara,
40450 Shah Alam, Selangor, Malaysia.
Email: fadhlan@salam.uitm.edu.my

1. INTRODUCTION

Transport or transportation is the movement of humans, animals and goods from one location to another. Transport is a critical need for people in rural area, especially car. This is because, in rural areas, a car is the main transportation to go to work. Therefore, high demand from locals will increase the number of cars in the country and thus will increase the number of accidents occurs.

Nowadays, many countries compete against each other in developing their technologies especially in vehicles such as an implementation of the driverless vehicle technology [1]. The autonomous vehicle has already been started and becomes more advanced since fifty years ago [2]. It has been started by Carnegie Mellon University (CMU) in the 1980s in their presentation of the autonomous vehicle that can drive without a driver [3]. Then, the autonomous vehicle technology continues by Google and combine with the best engineers from DARPA that was led by Stanford University Professor Sebastian Thrun to implement fully interaction of the self-driving cars [4]. Moreover, in 2012s Google autonomous vehicle project has reached their goal which is their self-driving vehicle can drive more than 300,000 km with no accident [1].

The Autonomous Vehicle System is the new technologies that has been implemented with the concept of driveless car. The vehicles able to make decision on its own such as whether to turn left or right at the intersection. This is achieved with the help of a computer that act as the brain that controlling this driveless vehicle. In addition, Vexta motor was used to move the vehicle steering. The most critical element in the autonomous vehicle is the steering [5]. The conventional steering has been removed such as steering

and the steering shaft. Those components have been replaced by the electric motor to control the direction of the vehicle. There was a lot of benefit due to removing the mechanical parts of the steering such as the handling performance can be improved, vehicle's weight was reduced, impact force to the driver in frontal accidents can reduce and also there will be a large space in cabin [6], [7].

The rotary encoder was used to give the signal to the microcontroller in order to control the direction of the vehicle steering. The calibrations of the tires need to be done first to make sure the tire is at the center before any movement occurs. The maximum angle of the steering for UiTM AV0 is 25° degree left and 25° degree right. The 24V battery is needed to operate the Vexta motor. The Arduino MEGA was used to read the rotary encoder and send to the MATLAB using the USB serial data. The MATLAB is the computer software that will be used to calculating the data collected from the rotary encoder and IMU to performing the mapping.

PWM signal was used to controlling the direction of the steering. There were a few problems encountered with using PWM which is the precision of the direction of steering of UiTM Autonomous Vehicle. This signal was not precise because the convergence to desired output value was not achievable during the required period of time. Besides, the time responds to obtain the desired output value of the steering until stable was very slow by using the PWM signal. The vibrations also occur while maintaining the system in order to get the output value. PWM signal only sends a high and low output signal to the Vexta motor driver to control the direction without having any calculation method. Previously, PID has been implemented as part of the Ziegler-Nichols based PID control system that controls robot [9], and Teleoperation Manipulators System [10]. Recently, Qi and Zhang also discuss about a study on adaptive PID control algorithm based on RBF Neural Network [11].

The work presented in this paper is carried out to analyze and improve the convergence of steering angle in the autonomous control of UiTM Autonomous Vehicle. The test area for this car is around Faculty of Electrical Engineering, University Teknologi MARA, Shah Alam, Selangor. The speed of the vehicle while performing this research will be maintained at a constant speed of 10KM/H. The mapping distance for this vehicle is only 100 meters. The objective of this research is to reduce overshoot in the steering control while using the PWM signal. Next is to implement the PID controller and compare the steering performance in term of time to convergence and occurrence of overshoot between PID and PWM. Previously he PID controller was used to solve the problem in controlling the desired output of the steering angle with the steering actuator [8]. Besides, it also can reduce the disturbances happened on the curvature which increases linearly respect to time [12]. This conventional PID control also can overcome precision when the angle error too large [13]. Rather than using PID controller, there also have many methods for controlling the steering [14]. The rest of this paper is arranged as follows: Section 2 elaborates the research methodology employed, while Section 3 presents the results and relevant analysis. Lastly, Section 4 concludes the paper.

2. RESEARCH METHOD

In this chapter will explain detail about the methodology of this project. This project development consists of both hardware and the software elements. In order to do the analysis of this autonomous vehicle mapping, there were few methods that will be used towards achieving the project outcomes.

Figure 1 shows the flowchart of the project for analyzing in improving the convergence of steering angle in autonomous control of UiTM Autonomous Vehicle. This project basically uses the rotary encoder compare with the set angle value from the MATLAB to control the steering and the direction of the autonomous vehicle.

The data from the rotary encoder was collected to the Arduino Mega to be converted into a degree of angle because the rotary encoder produces 500 pulses per rotation. The angle value sent to the Arduino Mega from the MATLAB to be compared with the angle of the rotary encoder. In this process, the PID controller is needed to perform the system.

The steering will turn to the right if the angle value from the MATLAB was bigger than the angle of the rotary encoder. While the steering will turn to the left if the angle value from the MATLAB was smaller than the angle of the rotary encoder. Then it will return to start to compare again until the value is equal. Furthermore, the steering will be maintained and not move if the angle value from the MATLAB was equal with the angle of the rotary encoder.

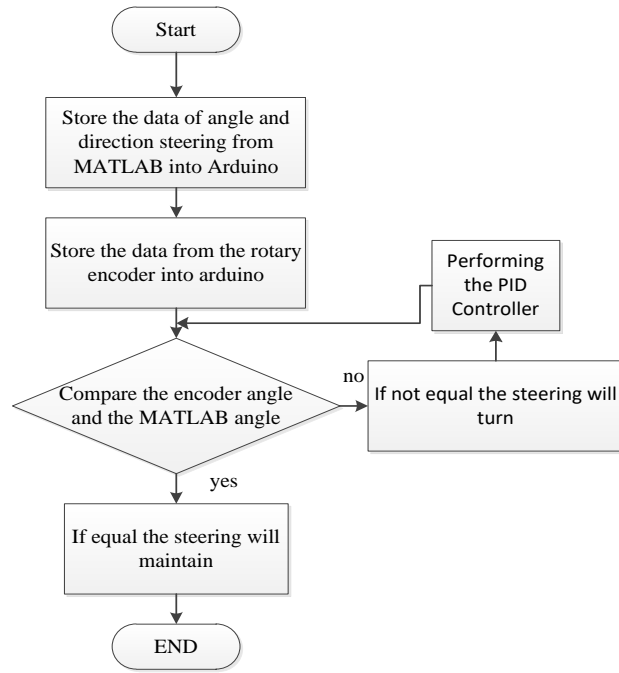


Figure 1. Flow chart of the system

2.1. PID Controller Design

In controlling the steering using the PWM signal, it produces a lot of vibration and disturbance on the steering. This problem can be solved by implementing the PID control system. It is because it can reject the disturbances happened on the curvature which increases linearly with respect to time [9]. This conventional PID control also can overcome with high precision when the angle error too large [10]. Figure 2 shows the PID system block diagram.

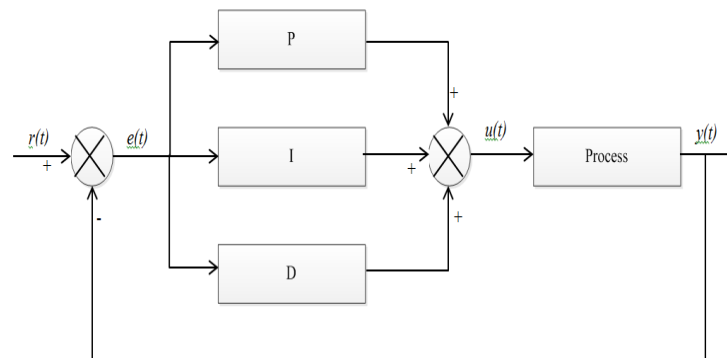


Figure 2. PID block diagram

For the PID controller design, some parameters need to be obtained first from the original graph. In this part will show the calculation method used for determining the transfer function $G(s)$ of the graph that was obtained from AV. This transfer function $G(s)$ was used to obtain the parameters of the PID controller which are K_p , K_i , and K_d . As shown in (1), (2), (3), and (4) shows the calculation of parameters for transfer function $G(s)$. As shown in (5) is the result of the transfer function $G(s)$. Figure 3 shows the parameters use to determine the transfer function.

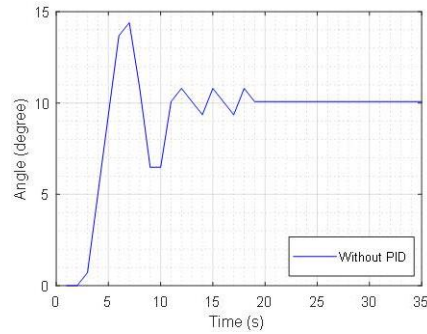


Figure 3. Parameters for higher order models

$$\text{Overshoot Ratio (OR)} = \frac{\Delta y_{OR}}{\Delta y}$$

$$\text{Overshoot Ratio (OR)} = \frac{4.4}{10} = 0.44 \quad (1)$$

$$\theta = \arctan\left(\frac{-\pi}{\ln OR}\right)$$

$$\theta = \arctan\left(\frac{-\pi}{\ln 0.44}\right) = 1.32 \quad (2)$$

$$\tau = t_{OR} \frac{\sin(\theta)}{\pi - \theta}$$

$$\tau = 1.0 \frac{\sin(1.32)}{\pi - 1.32} = 0.53 \quad (3)$$

$$\varepsilon = \cos\theta$$

$$\varepsilon = \cos 1.32 = 0.25 \quad (4)$$

$$G(s) = \frac{K}{(\tau^2 s^2 + 2\tau\varepsilon s + 1)} e^{-t_d s}$$

$$G(s) = \frac{1}{(0.53)^2 s^2 + 2(0.53)(0.25)s + 1} e^{-(1)s}$$

$$= \frac{1}{0.29s^2 + 0.27s + 1} \quad (5)$$

The MATLAB software was used to do the simulation based on the transfer function $G(s)$ that was obtained as shown in (5). In Figure 4 shows the transfer function $G(s)$ was inserted into the MATLAB pseudo code software. Figure 5 shows the graph of the transfer function after the execution of the pseudocode. The parameters which are K_p , K_i and K_d were obtained from the transfer function graph in Figure 5.

```
>> s=tf('s');
>> sys = 1/(0.29*s^2 + 0.27*s + 1);
>> sys

sys =
-----
      1
 0.29 s^2 + 0.27 s + 1

Continuous-time transfer function.

>> step(sys)
```

Figure 4. MATLAB pseudo code

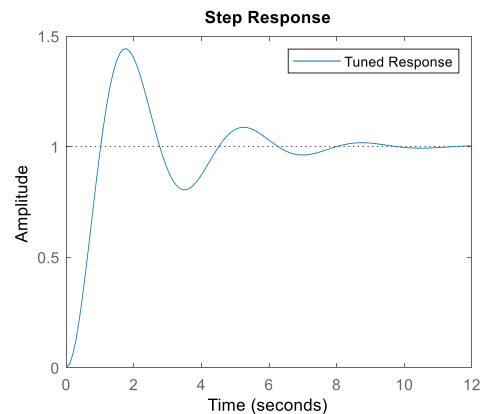


Figure 5. Graph of the transfer function

2.2. Hardware Implementation

Hardware part is the major element in designing the autonomous vehicle. Hardware parts need all electrical component to ensure that the autonomous vehicle functioning well. Hardware can be divided into two parts which are mechanical and electrical parts.

2.2.1. Mechanical Part

Example of the mechanical parts are body, steering itself, tires and motor. Some of the original steering set had been removed such as conventional steering and the steering shaft. This is because to replace the original part with the electrical motor which namely Vexta motor that was used to control the direction of the front wheel. Therefore, aluminium material was used for the mounting bracket for Vexta motor. It is designed so that the Vexta motor shaft is connected directly to the steering rack shaft. Next, the aluminum plate has been used to mount the bracket for the rotary encoder. The Rotary encoder was mount and the center of the steering rack to get the angle of the steering.

2.2.2. Electrical Part

Electrical part required two microcontrollers, rotary encoder, 24V lead acid battery, Vexta motor and Vexta motor driver. Both microcontrollers used were Arduino Mega. Arduino Mega was connected to the rotary encoder using the interrupt pin on Arduino Mega to collect the data from the rotary encoder and perform the calculation. Secondary Arduino Mega was used to send the signal to the Vexta motor driver (AXHD 450K) to control the direction of the Vexta motor. The Vexta motor was used to move the direction of the steering.

2.2.3. Software Implimentation

This part discusses on software that was used in this research to perform the analysis of improving convergence of steering angle in autonomous control of UiTM Autonomous Vehicle. The Arduino IDE software was used as a code compiler which later uploaded the code to the Arduino Mega microcontroller. Basically, the C/C++ programming language was used in this research. Then, the Microsoft Excel 2010 was used to store the data collected from the Arduino Mega to transform into the graph and do the analysis. The extension software for Microsoft Excel 2010 which is PLX-DAQ was used for connecting the Microsoft Excel with the Arduino Mega via USB Serial connection. This simple software was able to monitor or making the graph in real-time monitoring. It can directly analyze the output graph either the output is precise or not.

2.3. Project Setup

This project setup is about the process of all equipment or component which had been implemented on the project before doing the analysis. The Figure 6 shows all the hardware had already been install on the vehicle. The maximum angle of the steering had been taken first before doing the analysis. The maximum angle when turning to the left from the right is 60° which means left is 30° and right is 30°. Figure 7 shows the illustration of the maximum degree of turning steering angle.

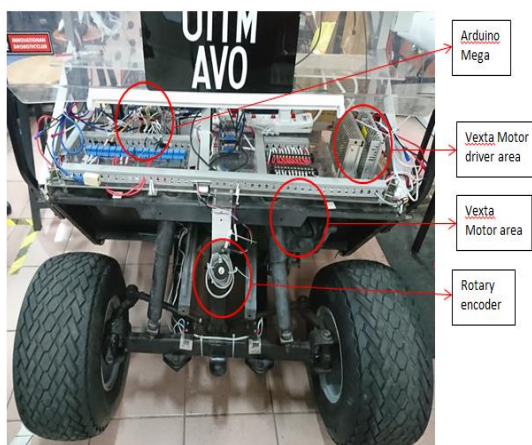


Figure 6. The setup of the project

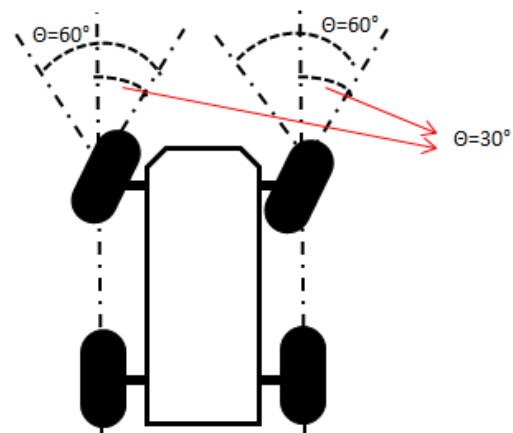


Figure 7. Angle of turning steering

3. RESULTS AND ANALYSIS

In this section we compare the steering convergence performance to a specific steering angle using both implementation of PID controller and without PID controller. Three results with three different targeted steering angle which are 10°, 15° and 25° were collected. Figure 8, Figure 9, and Figure 10 show the comparison in term of steering convergence to targeted steering angle with PID controller and without PID controller.

Figure 8, 9 and 10 shows that the overshoot has been reduced with the implementation of PID controller. For example, Figure 8 shows that the amplitude of the overshoot is 14.4° at time 2.2s is without the PID controller compared with Figure 11 shows that the amplitude of the overshoot is 10.8° at time 2.0s with PID controller. The convergence time achieved by PID control is also significantly faster, as shown by Figure 8. PID steering control converges to the desired angle of 10° after 7 seconds but, PWM controller took 19 seconds to converge to the same angle. From Figure 9, the PID controller convergence took 19 seconds faster than PWM controller while in Figure 10, the PID controller convergence took 37 seconds faster than PWM controller.

There was some problem while carrying out this analysis which is coming from the mechanical parts. There was a gap on the steering rack that makes the steering become vibrates. This situation makes the value of encoder is less accurate. Besides, this Vexta motor is not suitable for controlling the steering because it does not have fixed angle in performing the direction. The Vexta motor drivers have some problem which is the load too high for the Vexta motor to support. The Vexta motor driver will stop working and need reset at the Vexta motor driver. There are two ways to reset the motor driver which is turn off the power supply to the Vexta motor driver and give the signal high to the reset pin on the Vexta motor driver.

The speed of the Vexta motor also one of the important aspect that needs to be considered while performing this project. The experimental of speed test need to be done several times to get the accurate speed of the Vexta motor before performing the analysis. Other than that, the values of the graph also need to be collected few times for accurate results.

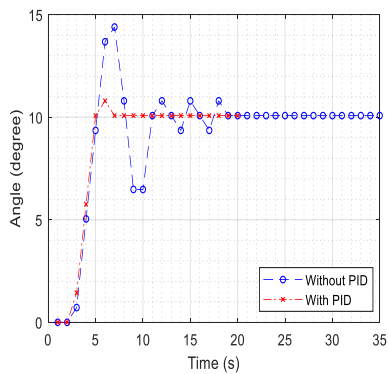


Figure 8. Comparison of steering convergence to 10° angle without using PID controller and using PID controller

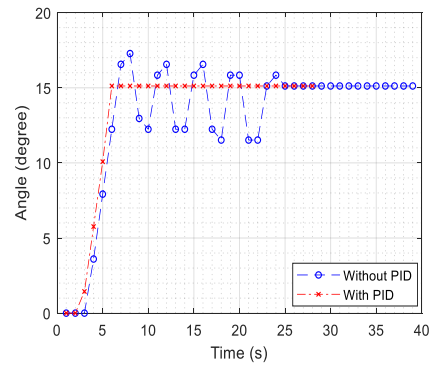


Figure 9. Comparison of steering convergence to 15° angle without using PID controller and using PID controller

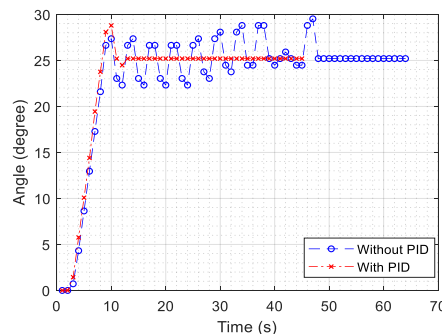


Figure 10. Comparison of steering convergence to 25° angle without using PID controller and using PID controller

4. CONCLUSION

In conclusion, this paper has proposed an implementation of PID controller on UiTM AV0 steering control. Results showed that the overshooting of PWM controller has been reduced after implementation of the PID controller. The PID controller had shown satisfactory performance in all assessments including time response, stability and overshoot of the system. From experiments, PID controller can achieve convergence to desired steering angle up to 37 seconds faster than PWM controller, due to its better stability. Thus, the steering control for UiTM AV0 which possesses good stability and fast convergence has been successfully developed.

ACKNOWLEDGEMENTS

This research work was supported by Institute of Research Management and Innovation (IRMI), Universiti Teknologi MARA (UiTM), Shah Alam, Selangor, Malaysia. The authors also wish to acknowledge and thanks the Faculty of Electrical Engineering their logistic support.

REFERENCES

- [1] K. R. Li, G. T. Lin, L. Y. Lee, and J. C. Juang, "Application of Particle Filter Tracking Algorithm in Autonomous Vehicle Navigation," pp. 250–255, 2013.
- [2] R. E. Fenton and R. J. Mayhan, "Automated Highway Studies at The Ohio State University—An Overview," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 100–113, 1991.
- [3] C. Thorpe, M. Hebert, T. Kanade, and S. Shaffer, "Toward Autonomous Driving: The CMU Navlab Part I — Perception," *IEEE Expert. Syst. their Appl.*, vol. 6, no. 4, pp. 31–42, 1991.
- [4] E. Guizzo, "How Google's Self-Driving Car Works," on Ligne. p. 5.p, 2011.
- [5] J. Huang and H. Tan, "Design of an Automatic Steering Controller for Bus Revenue Service Based on Drivers' Steering Mechanism," pp. 3930–3935, 2014.
- [6] S. A. Saruchi, H. Zamzuri, S. A. Mazlan, M. Hatta, M. Ariff, and M. Afandi, "Active Front Steering for Steer-by-Wire Vehicle via Composite Nonlinear Feedback Control," 2015.
- [7] S. M. H. Fahami, H. Zamzuri, S. A. Mazlan, and M. A. Zakaria, "Modeling and simulation of vehicle steer by wire system," in SHUSER 2012 - 2012 *IEEE Symposium on Humanities, Science and Engineering Research*, 2012, pp. 765–770.
- [8] J. K. Subosits and J. C. Gerdes, "A Synthetic Input Approach to Slip Angle Based Steering Control for Autonomous Vehicles," pp. 2297–2302, 2017.
- [9] Kim Seng Chia, "Ziegler-Nichols Based Proportional-Integral-Derivative Controller for a Line Tracking Robot", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 9, No. 1, January 2018, pp. 221-226
- [10] Gao Wei, Zhou Miaolei, "Fuzzy PID Control Method for Internet-based Teleoperation Manipulators System", *TELKOMNIKA*, Vol.11, No.11, November 2013, pp. 6381-6389.
- [11] Qi Xuemei, Zhang Jingdong, "Study on Adaptive PID Control Algorithm Based on RBF Neural Network", *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol. 13, No. 2, February 2015, pp. 287-291
- [12] R. Marino, S. Scalzi, G. Orlando, and M. Netto, "A Nested PID Steering Control for Lane Keeping in Vision Based Autonomous Vehicles," pp. 2885–2890, 2009.
- [13] C. S. Namuduri, S. Gopalakrishnan, B. V Murty, R. Bolio, R. Feller, and M. Rd, "Robust Control of Low-Cost Actuator for Automotive Active Front Steering Application," pp. 2108–2114, 2009.
- [14] C. Urmson et al., "Autonomous driving in Urban environments: Boss and the Urban Challenge," in *Springer Tracts in Advanced Robotics*, 2009, vol. 56, pp. 1–59.