Advances in Engineering Research, volume 208
Proceedings of the International Conference on Innovation in Science and Technology (ICIST 2020)

Modelling and Control of DC Motor Speed and Position for Wheel Mobile Robot Application

B. Supriyo*, A. Suharjono, K. W. Atmaja
Magister Terapan Teknik Telekomunikasi
Politeknik Negeri Semarang
Semarang, Indonesia
*bsupriyo7763@yahoo.com

***Abstract*—This paper presents the simulation works of PID based speed and position controllers for the DC motor used in mobile robot applications. The mathematical model of the DC motor modelling is developed using Matlab System Identification Toolbox based on its input and output signals obtained during open loop experiments under full load condition. The PID controllers are proposed to regulate the speed of the motor for the maximum of 100rpm. The initial PID parameters are obtained using relay feedback experiment method and Ziegler-Nichols formula. These parameters are then further fine tuned based on practical knowledge in tuning PID. The best PID speed controller is then selected and used in designing the DC motor position controller which utilizes both speed and position feed backs. The simulation results show that both speed and position controllers have a very good performance.**

Keywords*—wheel mobile robot application, PID speed controller, and two-wheel mobile robot

I. Introduction

DC motors have been used extensively as actuators for wheel mobile robot. The wheel mobile robot is a kind robot that uses wheel to move. It is controlled by programmed embedded inside the microcontroller or computer to work automatically by recognizing its environmental conditions based on its sensor information and to navigate around its working area to carry out its programmed tasks. Generally, the robot can have two wheels [1], three wheels [2] or four wheels [3]. Each of the robot active wheels is driven by the DC motor. The smooth maneuvering of the robot depends on its DC motor controller. Two general control applications widely used for the DC motor are speed and position controllers. The speed controller is used to regulate how fast the robot to move, while the position controller is used to regulate how accurate the robot to move to its predetermined position.

Various works related to the methods of controlling the DC motor speed have been introduced lately to stabilize. Some of these works combined the PID based controllers with other methods such as fuzzy logic controller [4]. Genetics algorithm [5] and Particle swam optimization [6] to stabilize the DC motor speed performance during operation. Some of the prostitution controller works combine PID based controller with other artificial intelligent method such as Fuzzy [7], Genetic algorithm [8] and artificial neural Network [9] to improve its position accuracy.

For this study, one of the DC motor of the two wheeled mobile robot is used as the plant. The mathematical model of the DC Motor is carried out by first obtaining the input and output signal from open loop experiment of the DC motor by applying the working voltage (+5V dc) of the DC motor system as input signal and measure its speed electronically as output signal. The step response of the mathematical model resulted from the identification process is compared to that of the original one to evaluate its validity. The PID based speed controller is proposed with its initial PID parameters obtained from relay feedback experiment method and Ziegler-Nichols formula. These parameters are then fine tuned manually based on practical knowledge in PID tuning to get the good results with no overshoot and no steady state error. The best PID based speed controller is then slightly modified and used in the DC motor based position controller by utilizing both speed and position feedbacks. This position controller is tested for several position set points. The controller performance is evaluated based on the overshoot and steady state error values.

II. System Identification

System identification is a way of developing a mathematical model of the DC motor as a plant by analyzing the input and output signals of the process during experiment or normal running. The construction of the model is primarily based on the observed data. There is a discrepancy between the real system and the mathematical model representation of the system, so a perfect mathematical model of the process cannot be obtained. The system identification process needs experimental planning to obtain the input and output data as the observed data, selection of the model structure, a certain criterion needed to measure how well a model fits the experiment data, parameter estimation, and model validation. This system identification is performed using identification tool in Matlab.

In this study, the system identification process starts by obtaining the experimental input-output data from open loop DC motor. The unit step input data is a 5 volts DC voltage used to drive the DC motor. The output data is a measured rotational speed of the DC motor shaft using encoder sensor. The measurement process is performed by Arduino Uno microcontroller which reads both the 5 Volt unit step input and encoder pulses which converted to its respective speed. Next, the Arduino Uno transfers these experiment data to the computer, then processed using Excel software. Finally, the data is saved as a file consisting of time, input voltage and speed of the shaft motor. This file is used by system identification toolbox for obtaining mathematical model of the DC motor.

III. Controller Design

A. Initial PID Parameters

Initial PID parameters of the speed controller are determined based on Relay Feedback method and Ziegler-Nichols formula. This method was originally proposed by Åström and Hägglund [10]. The Simulink program for Relay Feedback experiment is shown in figure 1. The resulted output signals produce a stable oscillation of a and relay output amplitude of h as shown in figure 2. The values of critical period (T_c) is obtained from the stable period time of the relay output. The critical gain (K_c) of the controller can be computed using equation (1). While the proportional gain (K_p), integral gain (K_i) and derivative gain (K_d) can be determined using, Ziegler Nichols formula given in table 1.

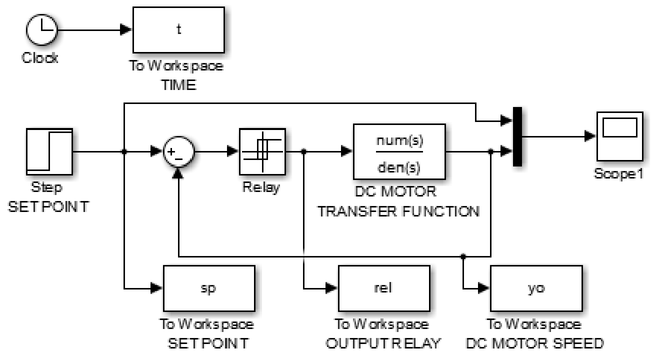

Fig. 1. Relay feedback experiment.

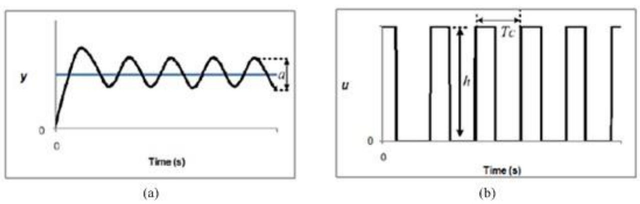

Fig. 2. (a) Oscillation output, (b) Relay output.

$$K_c = 4h/\pi a \quad (1)$$

TABLE I. Ziegler-Nichols Formula

Controller	PID Parameters		
	K_p	K_i	K_d
PID	$0.33K_c$	$2K_p /T_c$	$K_p\ T_c/8$

B. PID Speed Controller

The simulink program developed for PID speed controller is shown in figure 3. The initial PID parameters consisting of proportional gain (K_p), integral gain (K_i) and derivative gain (K_c) are obtained from relay feedback experiment and calculated Ziegler Nichols formula. These initial values are then tuned manually based on the practical knowledge in tuning PID. Firstly we work on P controller and tune the proportional gain (K_p) such that the output results in only one overshoot and smooth constant value without oscillation, but still having steady state error, for the rest of the simulation process. Next we work on PD controller using the selected K_p and tune the derivative gain (K_d) such that the output produces no overshoot and no oscillation, but still having steady state error, for the rest of the simulation process. Finally, we work on PID controller using the selected K_p and K_d, and tune integral gain (K_i) such that the output produces no overshoot and no steady state error, for the rest of the simulation process. In this case, the PID parameters have been successfully tuned.

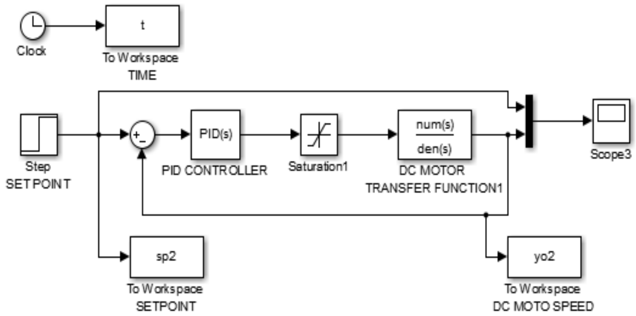

Fig. 3. PID speed controller.

C. Position Controller

The simulink program for position controller is shown in figure 4. The position controller is developed based on the previous PID speed controller in which the speed output is converted to its linear displacement by multiplying the speed with the wheel circumference. The set point is the desired position, while the output is actual position of the robot which is the distance travelled by the wheel from its point of origin. The difference between the set point and the actual position is then multiply by the position gain. The output of this gain is fed to the speed controller unit as the current speed set point.

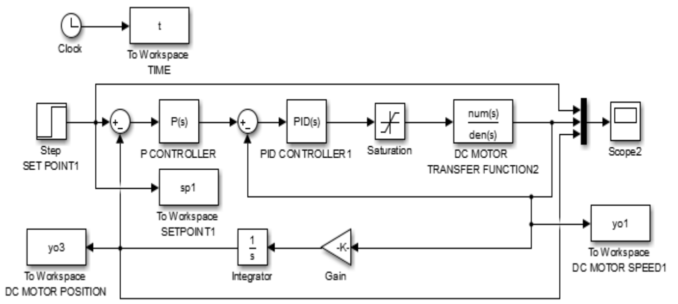

Fig. 4. Position controller.

IV. Results and Discussion

A. System Identification

System identification process starts by collecting the input-output data from open loop DC motor experiment with the help of Arduino Uno microcontroller. The input data is a 5 volts DC voltage used to drive the DC motor, while the output data is a rotational speed of the DC motor shaft measured using encoder and processed and converted to its revolution per minute (rpm) by Arduino Uno. These experiment data are transferred to computer from Arduino Uno via serial USB and then processed using Excel software. The data consists of time, input voltage and speed of the shaft motor. By using the system identification toolbox in Matlab, the respected input-output data which is saved as a file is loaded to the system for identification purpose. By selecting the second order system in the form of transfer function, the mathematical model obtained from system identification process is presented in (2), where u is the voltage input and y is the speed output of the motor.

The step input of 5 volts is fed to the input of the DC motor plant resulted from the system identification process. The open loop speed responses of the real DC motor (Exp.) and the mathematical model one resulted from system identification process (SI) is shown in figure 5. The real DC motor has the maximum speed of about 115 rpm, while the mathematical model one has maximum speed of about 112 rpm. Based on data in figure 5, the average percentage error of the open loop performance between the real motor and the mathematical model of the motor is about 4.55%.

$$y(s) = \frac{26.43}{s^2 + 1.97s + 1.18}\, u(s) \qquad (2)$$

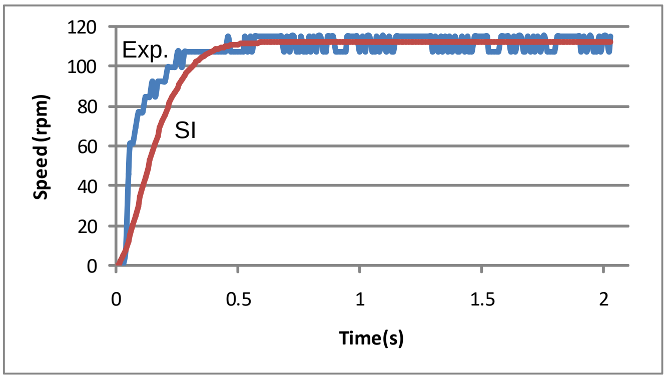

Fig. 5. Step response of the open loop DC motor.

B. Initial PID Parameters

The results of the relay feedback experiment consist of oscillation output (y) and relay output, as shown in figure 6 and figure 7, respectively.

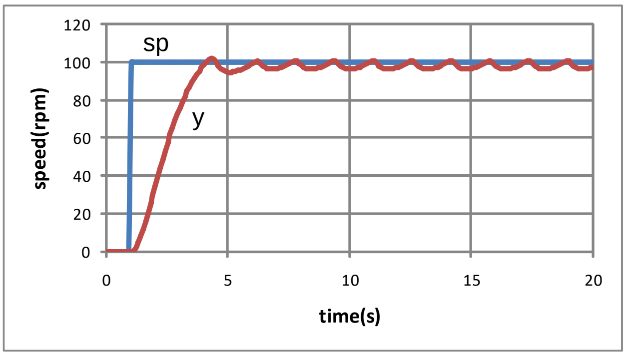

Fig. 6. Oscillation output.

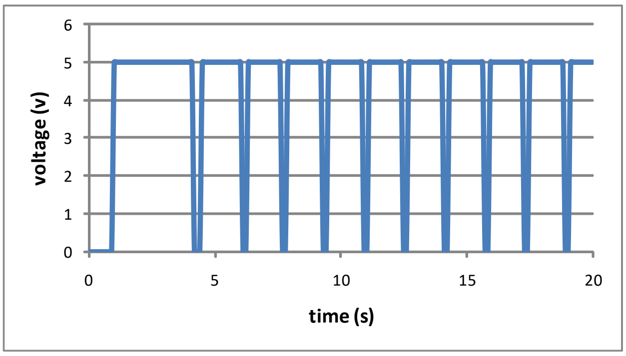

Fig. 7. Relay output.

From the relay feedback experiment, the following parameters are obtained:

- a= 4.27;
- h=5;

- Tc = 1.5.

While From Ziegler-Nichols formula these following initial PID parameters are determined:

- K_p = 0.9;
- K_i = 1.2;
- K_d = 0.17.

C. Speed Controller Performance

After the fine tuning process it was found that the final K_p = 0.05; K_i = 0.034 and K_d = 0.026. The initial value of K_p = 0.9 gives big oscillation output. The fine tuning is performed to adjust the K_p , such that the P controller output results in only one overshoot and no oscillation, although steady state error is still not zero. This final value of K_p is 0.05.

Next, the process continues with the PD controller with the K_p = 0.05 and initial K_d = 0.17. The PD is fine tuned by adjusting the K_d parameter, such that the PD controller output results in no overshoot and no oscillation, although the steady state error is still not zero. This final value of K_d is 0.026.

Finally, the process continues with the PID controller with the K_p = 0.05, K_d = 0.026 and initial K_i = 1.2. The PID is fine tuned by adjusting the K_i parameter, such that the PID controller output results in no overshoot, no oscillation and zero steady state error. This final value of K_i is 0.034. The performance of the PID speed controller using the final parameters can be shown in figure 8. The output of PID controller reaches the 100 rpm set point in about 8 second, has no overshoot and has very small (near zero) steady state error. This PID speed controller is used as main component in developing the position controller.

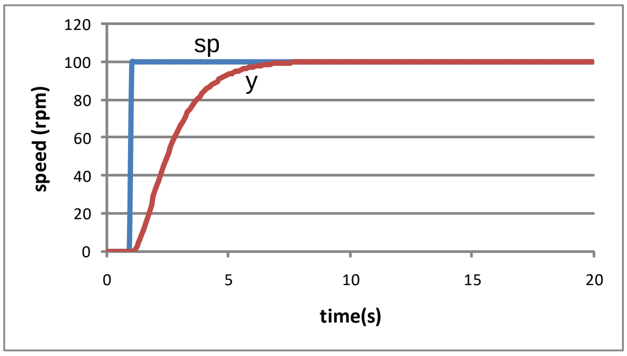

Fig. 8. Speed controller performance.

D. Position Controller Performance

The position controller uses the previous PID speed controller as main controller. However, in this case the speed output is converted to its linear displacement or position by multiplying the speed with the wheel circumference. The difference between the position set point and the calculated position is fed to the position Proportional controller. The output of this controller is used to regulate the speed of the DC motor, such that the actual position can reach the desired position set point. The actual position of the robot will be calculated as the distance travelled by the wheel from its point of origin.

By using the wheel having diameter of 6 cm, the proportional gain of the position controller is fine tuned manually. The best tuning was obtained with the proportional gain position of 0.08. The simulation works perform the position controller based on the set points of 10, 50 and 100m. The output response for position controls for these positions are very good. The response have no overshoot and no steady state errors as shown in figure 9, where X10, X50 and X100 are the output response for set points (Sp) of 10cm, 50cm and 100cm respectively.

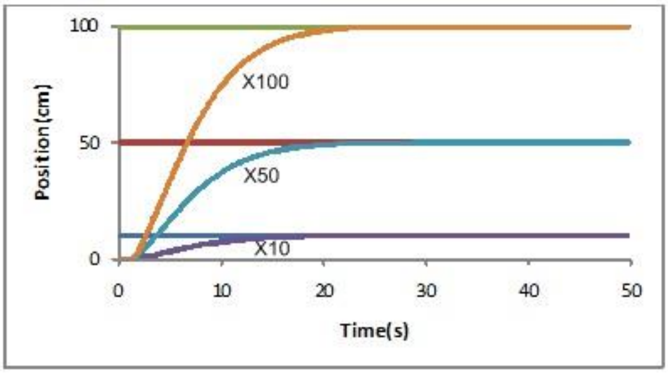

Fig. 9. Position controller performance.

V. Conclusion

The system identification process is carried out to obtain the mathematical model of the DC motor for two-wheeled mobile robot. This identification process has been performed satisfactorily which results in the second order system transfer function representing the DC motor plant under studied. Based on this plant mathematical model, the PID based speed controller is designed. The initial PID parameters are obtained from relay feedback process and calculated using Ziegler-Nichols formula. Since the PID controller with its initial parameters performs unsatisfactorily, then these parameters were fine tuned manually with the help of practical knowledge based in tuning PID. The results are very satisfactory in which the speed response for 100 rpm has no overshoot and no steady state error. Furthermore, the position controller which is derived from the speed controller with the proportional gain of 0.08 has performed satisfactorily for set points of 10cm, 50cm and 100cm. The position output response mostly has no overshoots and no steady state errors.

References

[1] M. Deng, A. Inoue, A. Skiguchi, and L. Jiang, “Two-wheeled mobile robot motion control in dynamic environments,” Robotics and

Computer-Integrated Manufacturing, vol. 26, no. (1), pp. 268–272, 2010.

[2] M.A. Sharbafi, C. Lucas, and R. Daneshvar, “Motion Control of Omni-Directional Three-Wheel Robots by Brain-Emotional-Learning-Based Intelligent Controller,” IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 40, no. (6), pp. 630-638, 2010.

[3] Y. Hao, J. Wang, S.A. Chepinskiy, A.J. Krasnov, and S. Liu, “Backstepping Based Trajectory Tracking Control for a Four-Wheel Mobile Robot with Differential-Drive Steering,” Proceedings of the 36th Chinese Control Conference, pp. 4918-4923, 2017.

[4] U.K. Bansal and R. Narvey, “Speed Control of DC Motor Using Fuzzy PID Controlle,” Advance in Electronic and Electric Engineering, vol. 3, no. (99), pp. 1209-1220, 2013.

[5] V.K. Giri and S.K. Suman, “Speed control of dc motor using optimization techniques based PID controller,” The 2nd IEEE International Conference on Engineering and Technology (ICETECH), pp. 581-587, 2016.

[6] R.G. Kanojiya and P.M. Meshram, “Optimal Tuning of PI Controller for Speed Control of DC motor drive using Particle Swarm Optimization,” International Conference on Advances in Power Conversion and Energy Technologies (APCET), pp. 1-6, 2012.

[7] E. Natsheh and K.A. Buragga, “Comparison between Conventional and Fuzzy Logic PID Controllers for Controlling DC Motors,” JCSI International Journal of Computer Science, vol. 7, no. (5), pp. 128-134, 2010.

[8] N.P. Adhikari, M. Choubey, and R. Singh, “Dc Motor Control Using Ziegler Nichols and Genetic Algorithm Technique,” International Journal of Electrical, Electronics andComputer Engineering, vol. 1, no. (1), pp. 33-36, 2012.

[9] M. Aamir, “On replacing PID controller with ANN controller for DC motor position control,” International Journal of Research Studies in Computing, vol. 2, no. (1), pp. 21-29, 2013.

[10] K.J. Åström and T. Hägglund, “Automatic tuning of simple regulators with specifications on phase andamplitude margins,” Automata, vol. 20, no. (5), pp. 645-651, 1984.