

**PI CONTROLLER DESIGN FOR VELOCITY CONTROL OF A MOBILE ROBOT**T. Sokunphal<sup>1,\*</sup>, W. A. F. W. Othman<sup>1</sup>, S. S. N. Alhady<sup>1</sup>, W. Rahiman<sup>2</sup><sup>1</sup>School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Malaysia<sup>2</sup>SPALAN Research Group, Engineering Campus Universiti Sains Malaysia, Malaysia

Published online: 18 April 2018

**ABSTRACT**

Velocity control of a ground robot is designed to obtain the current motion coordinate. Under the assumption of linear velocity and a fixed steering angle, the current motion coordinate of this robot can be defined. The closed loop control system is obtained by verification of DC motor model, design of speed controller by PI (proportional, and Integral) controller, and implementation of a kinematic model. A loop of PI controller is designed to provide a stable velocity to the system, and acquainted a current coordinate when the robot is started moving. The simulation and experimental results have demonstrated the functionality of velocity control of the mobile robot.

**Keywords:** Linear velocity; steering angle; PI; mobile robot**INTRODUCTION**

In the past years, mobile robots were designed with large size, heavy and require a high cost computer system in which they need to be connected via cable or wireless devices. Nowadays, the trend is to evolve with a small mobile robot in order to reduce in size, weigh, and cost of the system by using sensors, numerous actuators, and the controller that can be carried on-board (Bräunl, 2008). Mobile robots are built based on a good relation of both hardware and software. Generally, the mobile robots have the capability to move around in their environment without the need human intervention (Chen and Agrawal, 2013).

Author Correspondence, e-mail: [sokunphal@gmail.com](mailto:sokunphal@gmail.com)doi: <http://dx.doi.org/10.4314/jfas.v10i3s.77>

In this paper, velocity control of a mobile robot is designed by using a closed loop feedback mechanism such as PI controller, and mobile robot kinematic model. This robot moves on the straight line with a constant speed provided by the speed controller. Moreover, the robot kinematic model provides the current motion coordinate of the robot for the linear velocity by using speed controller and a fixed steering angle. The speed is set for the mobile robot in range between 9.99cm/s (0.36km/h) to 88.7cm/s (3.2km/h).

According to the mobile robot which is designed with a simple velocity control by adjusting the PWM normally faces problems during movement such as the mobile robot cannot keep moving on straight line and difficult to control direction such as during the cornering left or right. Therefore, the speed controller is introduced in this work to make the mobile robot to have high efficient on maneuver system.

Furthermore, this paper prepares three main procedures in order to control the velocity for the mobile robot such as DC motor model verification, design speed controller based on PI controller, and implementing a mobile robot with kinematic model to the system.

MATLAB Simulink is used for simulation purposes, and Arduino Mega is a main microcontroller which is used to do the real-time experiments. The error between simulation and experimental results are very important to confirm that the proposed method is right to use in a motion robot system. The experiment conducted mainly to study the velocity efficiency at a straight line and turning of the mobile robot.

### Vehicle Mathematical Model

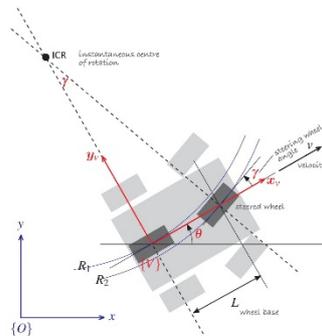
According to a high efficiency result, a realistic idea from large scale mobile robots is to focus on bicycle-like robots. The bicycle-like robot is one of the most effective motion control method used to control the wheels of the robot (Sanchez-lopez et al, 2012). The mathematical models are usually used in a fixed frame environment or in frame where the robot moves to the desired path (Kloetzer and Ghita, 2010). Vehicle mathematical model utilised throughout this paper can be written in as

$$\begin{aligned}x' &= V \cos \theta \\y' &= V \sin \theta\end{aligned}\quad (1)$$

and

$$\theta' = \frac{V}{L} \tan \gamma \quad (2)$$

Let assume the bicycle-like robot is moving with a longitudinal position  $x$ , a lateral position  $y$ , and an orientation  $\theta$ . The linear velocity  $V$  and a fixed steering angle  $\gamma$  are the main component in this motion model. The distance between front and back wheels of the robot is can be defined as  $L$ . The bicycle model is commonly used for a four-wheeled vehicle is shown in Fig. 1 (Siciliano et al, 2013).



**Fig.1.**The bicycle vehicle model

However, the real-time experiment is demonstrated by using Arduino Mega as a main microcontroller. There is another important mathematical model which is used to define the real-time motion coordinate of the robot according to the left and right position of the back wheels. This mathematical model is known as the odometry, as in equations(3) and (4). The current coordinate of the robot are  $x', y', \theta'$ . With  $D_r$  and  $D_l$  are the left and right position of the back wheels, respectively.  $D_c$  is the average position between left and right wheel.

$$\begin{aligned} \dot{x} &= x + D_c \cos \theta \\ \dot{y} &= y + D_c \sin \theta \end{aligned} \quad (3)$$

and

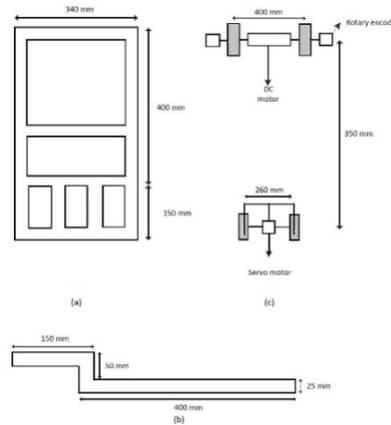
$$\dot{\theta} = \theta + \frac{D_r - D_l}{L} \quad (4)$$

## METHODOLOGY

### 1. Mobile Robot Concept

The mobile robot is designed with 340mm width, 550mm length, and 75mm height. The distance between front and back wheel is 350mm. The body and hardware attachment of mobile robot can be found as shown in Fig. 2, (a) top view, (b) side view, and (c) drive motor attachment. The total weight of mobile robot is about 13kg including one transaxle motor and 24v battery. The completed mobile robot for testing purpose is shown in Fig. 3, which the box

is used to carry microcontroller and driver motor. Furthermore, laptop is used as a power supply to microcontroller and a monitor to display data from microcontroller.

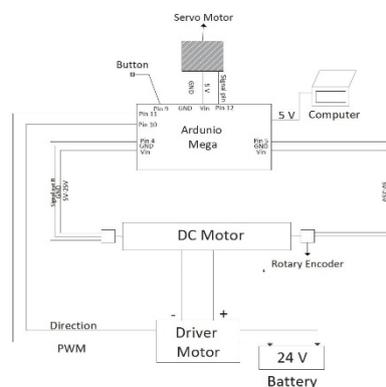


**Fig.2.** The body of mobile robot and drive motor attachment (a) Top view, (b) Side view, and (c) Drive motor attachment



**Fig.3.** The completed mobile robot for experiment

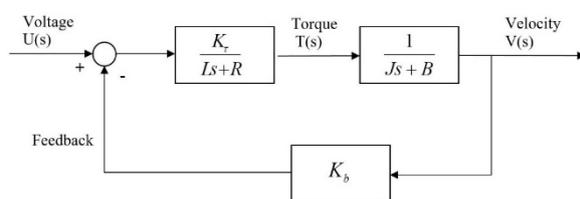
The robot is controlled by using Arduino Mega 2560 as a main microcontroller, and drive components such as motor driver, rotary encoder, and servo motor. The schematic diagram of microcontroller connection with other components are illustrated in Fig. 4.



**Fig.4.** The schematic diagram of microcontroller system

## 2. DC Motor Model Verification

DC motor model is very necessary to design the speed controller. The main purpose of DC motor model verification is to define a proper transfer function for DC motor, which is a relation between input voltage and velocity at the output. Transfer function of DC motor is a main section for modeling of DC motor, that can be demonstrated in closed loop block diagram in Fig. 5. The electromagnetic force gain,  $K_b$ , and motor torque gain,  $K_t$ . While  $R$  is a resistance,  $L$  is an inductance,  $B$  is viscous friction, and  $J$  is a rotor inertia (Babuska and Stramigioli, 1999). The transfer function of DC motor can be defined as shown in equation (5), which is in relation between DC gain  $K$  and time constant  $\tau$ .



**Fig.5.** Block diagram of DC motor

The first-order system may be described in terms of transfer function

where  $K$  is DC gain and  $\tau$  is time constant of

$$G(s) = \frac{K}{1 + \tau s} \quad (5)$$

DC motor. There are two experiments must be

conducted in order to obtain the value of DC gain and time constant of DC motor transfer function. The first experiment is carried out by testing PWM versus 500 samples of velocity in cm/s. Then the data between those 500 samples of velocity and running time is plotted by using MATLAB in order to get a steady state velocity. The time constant can be defined at 63% of the steady state velocity.

The second experiment is used to define DC gain by testing 10 different PWM versus 500 samples of velocity per each PWM. MATLAB is used to plot data in relation between PWM and velocity, then the steady state velocity for each PWM can be defined. The linear regression equation is used as a specific method to get the fitting curve between the steady state velocity and PWM, the linear regression equation is

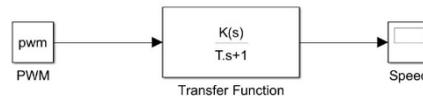
$$Y = a + bX \quad (6)$$

where  $Y$  is the dependent variable on  $Y$  axis steady state velocity and  $X$  is the independent variable on  $X$  axis PWM.  $n$  is the number of collected sample,  $a$  is  $y$ -interception and  $b$  is the

slop of the line as shown in equation (7). Furthermore, DC gain  $K$  is the average value of between velocity and PWM can be calculated which is equal to  $b$ . The DC motor model on Simulink is shown in Fig. 6.

$$a = \frac{\sum y \sum x^2 - \sum x \sum xy}{n \sum x^2 - (\sum x)^2}$$

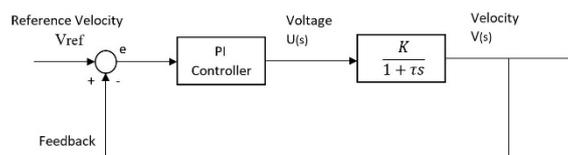
$$b = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$



**Fig.6.** Block diagram of DC motor in Simulink

3. *The Design of Speed Controller based PI Controller*

In this research, the PI controller has been implemented as a main controller to control the speed from DC motor for the mobile robot. The proportional control  $K_P$  is used to control signal respond to the error immediately. However, the error is not reduced to zero and the offset error is still inherently present. The integral control  $K_I$  has been used to remove the entire offset error in the system (Aly, 2011). Moreover, this design is also used to demonstrate the stability of mobile robot as suggested in Visioli, (2012). The block diagram of speed controller based on PI controller is shown in Fig. 7.



**Fig.7.** Block diagram of speed controller

Based on the digital control formula, the PI controller gain  $K_P$  and  $K_I$  can be calculated by using the relation of PI controller transfer function and the DC motor transfer function. Furthermore, the settling time, the peak time, and the percent of overshoot are very necessary for the design of speed controller based on PI controller. The block diagram of speed controller in Simulink is shown in Fig. 8. The PI controller transfer function can be computed by

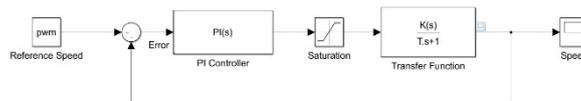
$$C(s) = K_p + \frac{K_I}{S} \tag{8}$$

The relation between PI controller transfer function and DC motor transfer function is

$$T(s) = \frac{G(s)C(s)}{1+G(s)C(s)} = \frac{K(K_pS+K_I)}{\tau S^2 + (1+K_pK)S+K_I K} \tag{9}$$

The response of the second order system transfer function is defined as

$$H(s) = \frac{W_n^2}{S^2 + 2\zeta W_n S + W_n^2} \tag{10}$$

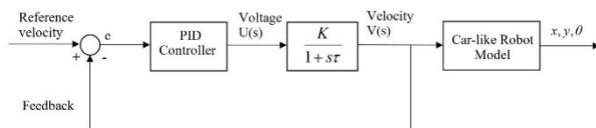


**Fig.8.** The block diagram of speed controller in Simulink

The real-time experiment is proposed in this research to verify the accurateness of the PI controller. The Arduino code is used for the experiment. Due to real-time experiment, the comparison between simulation and real-time testing by using PI controller is needed to verify that PI controller can be used as the speed controller.

4. *Mobile Robot Kinematic Model*

There are two main inputs for this kind of robot kinematic model such as linear velocity and a fixed steering angle. Both simulation and real-time experimental result are necessary to be done in this research. The block diagram of mobile robot model is shown in Fig. 9.

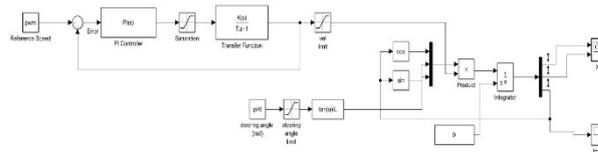


**Fig.9.** The block diagram of mobile robot

In addition, the block diagram of mobile robot kinematic model is implemented by using MATLAB Simulink as shown in Fig.10, where the linear velocity is provided by using PI controller is 20cm/s (0.72km/h) and a fixed steering angle is 30°. Due to the mechanical design on mobile robot, the fixed initial point in at angle 80°. This mean that the fixed steering angle 30° is equal to 50° in coding to control servo motor direction. Moreover, the velocity

limit is started from 0.18km/h (5cm/s) to 1.8km/h (50cm/s) and the steering angle block is started from  $-45^\circ$  to  $45^\circ$ .

Furthermore, the real-time experiment is done by using Arduino code as a main microcontroller to control steering angle and linear velocity. The comparison between simulation and real-time testing are very necessary in this research. According to the tolerance design, the acceptable error must be less than 10%, and the motion curves must be in the same pattern.



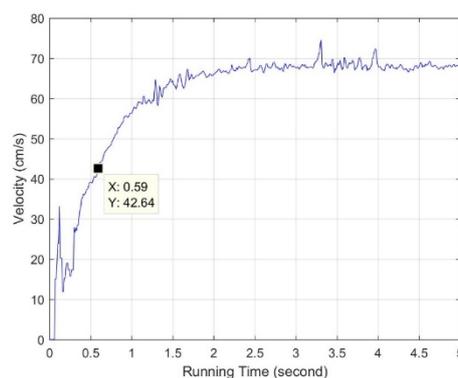
**Fig.10.** The bicycle model of mobile robot in Simulink

## RESULTS AND DISCUSSION

### 1. Transfer Function of DC Motor System

#### a) Time Constant

The experiment is done by running mobile robot for 5s to obtain 500 samples of relevant velocity. The relation graph between 500 samples of speed and running time for 5s is shown in Fig. 11, in which time constant  $\tau$  can be calculated at 63% of steady state velocity equals to 0.59s. This time constant is represent the speed with which the system is respond to change to reach 63% of its final value.



**Fig.11.** The relation between velocity and running time

## b) DC gain

**Table 1. Steady state velocity corresponding to PWM**

PWM (x)	Steady State Velocity [cm/s] (y)	xy	x <sup>2</sup>	y <sup>2</sup>
40	9.99	399.6	1600	99.8
60	17.44	1046.4	3600	304.15
80	25.71	2056.8	6400	669.00
100	33.28	3328	10000	1107.55
120	39.93	4791.6	14400	1594.4
140	47.60	6664	19600	2265.76
160	53.65	8584	25600	2878.32
180	62.54	11257.2	32400	3911.25
200	69.37	13874	40000	4812.19
220	74.74	16442.8	48400	5586.06

Then, another experiment is conducted to obtain DC gain, which the relation graph between PWM and velocity can be plotted by using MATLAB. Therefore, steady state velocity of each PWM can be obtained as shown in Table 1. Moreover, the linear regression parameter which is in relation between steady state velocity and PWM as

$$a = \frac{\sum y \sum x^2 - \sum x \sum xy}{n \sum x^2 - (\sum x)^2} = -3.37 \quad (11)$$

$$b = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} = 0.35$$

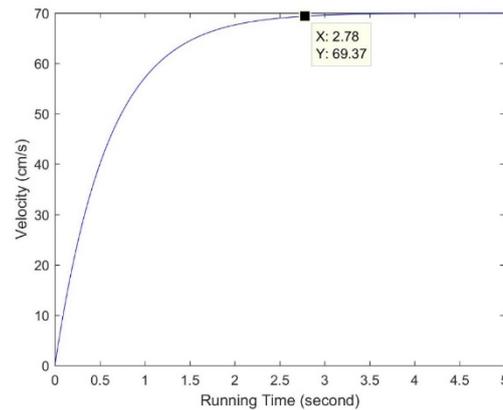
Therefore, the linear regress regression equation of PWM and steady state velocity can be obtained in equation (12), which DC gain can be known as the average value between the steady state velocity and PWM. In this case, DC gain is equal to 0.35.

$$Y = 0.35X - 3.37 \quad (12)$$

Finally, time constant  $\tau$  and DC gain  $K$  can be found. Therefore, the final DC motor transfer function can be written as

$$G(s) = \frac{0.35}{1 + 0.59s} \quad (13)$$

Furthermore, the relation graph between velocity and PWM = 200 can be plotted by using MATLAB Simulink, which is expressed the DC motor transfer function for mobile robot system as shown in Fig. 12.



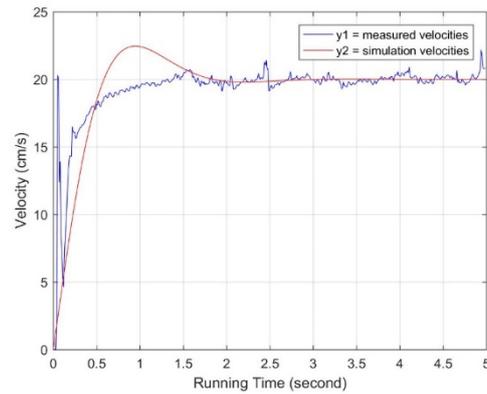
**Fig.12.** The transfer function of DC motor curve

Based on the graph in Fig. 11, the steady state speed of the input at PWM = 200 is 69.37cm/s. This steady state velocity value is demonstrated that the DC motor model verification is right for the system due to the comparison between the defined steady state velocity to the steady state velocity by using PWM= 200 in Table 1. Therefore, DC motor model is verified to use for mobile robot system.

## 2. The Design of Speed Controller

Speed controller is one of the main controller in this project. The speed controller is designed by using PI controller as a main controller, which has ability to provide the stable speed to the system. There are four main things of speed controller such as defining PI controller gains, PI controller simulation, real-time testing by using PI controller, and making a comparison between simulation and real-time testing result.

By using the digital control formula for PI controller which is relation between DC motor transfer function and PI controller transfer function, the PI gains can be calculated such as  $K_P = 3.77$  and  $K_I = 17.03$ . Moreover, the comparison between simulation and real-time experiment is done by plotting simulation result together with real-time experiment result is shown in Fig. 13. According the error calculation by using the root mean square error (RMSE) function in MATLAB, the error can be found as 1.91%. Due to the error rate is very small and less than 10%, therefore the speed controller is successfully designed for the system.

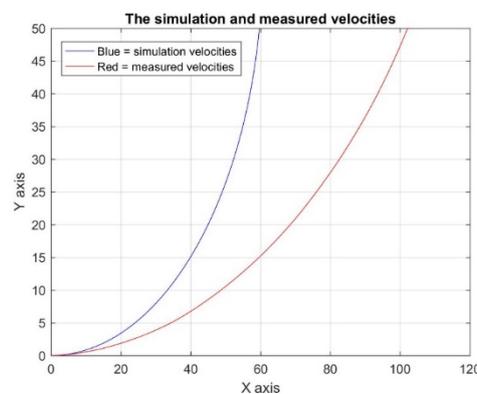


**Fig.13.** The comparison between simulation and real-time testing using PI controller

The blue curve is shown about the measured velocity curve which is obtained from real-time testing result. Besides, the red curve is dedicated to velocity curve which is done by simulation. Both curves are almost stick together in the same pattern. This graph is proved that the steady state velocity between simulation and real-time testing are really almost the same.

### 3. Mobile Robot Kinematic Model

By obtaining the result for simulation and real-time experiment. The comparison between simulation and real-time testing result is very compulsory to verify that the motion robot kinematic model can be used as a maneuver control system method. Moreover, the error of both results is very important in this section. The comparison between simulation and the real-time testing is done by plotting both results together. According to the obtained result for simulation and real-time testing, the comparison graph between simulation and real-time testing can be plotted as shown in Fig. 14.



**Fig.14.** The comparison between simulation and real-time testing velocities

Both curves are shown that the simulation and real-time testing has the same pattern curve within 10 seconds of mobile robot movement. Beside the same curve pattern, both curves are not fit together. The simulation result is demonstrated by the blue curve, and the real-time testing is plotted with red curve. Based on both curve, the movement in simulation is faster than the movement in real-time testing.

However, the error between the simulation and the real-time testing can be found by using MATLAB code. By using MATLAB code, the error rate can be defined as 86.02%. According to the error rate is higher than 10%, the design of mobile robot kinematic model is needed some improvements in order to become a maneuver control method for mobile robot system.

## CONCLUSIONS

In conclusion, the velocity control of a mobile robot is successfully designed by using speed controller and mobile robot kinematic model. The current motion coordinate can be defined, however the improvement should be made on mobile robot kinematic model to make simulation and real-time testing more accurate than this.

## ACKNOWLEDGEMENT

This research was supported by Development of Sustainable Platform for Land, Air and Naval (SPALAN) System Grant. (Grantnumber: 1001/PELECT/814265).

## REFERENCES

- Bräunl T. (2008). Robots and Controllers. Mobile Robot Design and Application with Embedded Systems, 3rd ed., Berlin: Springer, pp. 1–7.
- Chen X. and Agrawal S.K. (2013). Assisting versus repelling force-feedback for learning of a line following task in a wheelchair. IEEE Trans. Neural Syst. Rehabil. Eng., vol. 21, no. 6, pp. 959–968.
- Sanchez-lopez J.L., Compoy P., Olivarez-Mendez M.A., Mellado-Bataller I. and Galindo-Gallego D. (2012). Adaptive Control System based on Linear Control Theory for the Path-Following Problem of a Car-Like Mobile Robot(4), pp. 1–6.
- Kloetzer M. and Ghita N. (2010). Motion Planning and Control of a Car-Like Robot in an Environment Cluttered. Autom. Calc., vol. 3, pp. 81–94.
- Siciliano B., Khatib O. and Groen F. (2011). Mobile robot vehicles. Robotics, Fundamental Algorithms in MATLAB®, vol. 73, Springer, Ed. Berlin, Germany, 2011, pp. 66–77.

---

BabuskaR. and Stramigioli S. (1999). *Matlab and Simulink for Modeling and Control*. TU Delft, Delft, Netherland,

Aly A.A. (2011). PID Parameters Optimization Using Genetic Algorithm Technique for Electrohydraulic Servo Control System. *Intell. Control Autom.*, vol. 2, no. 2, pp. 69–76.

Visioli A. (2012) “Research Trends for PID Controllers,” vol. 52, no. 5, pp. 1–7.

**How to cite this article:**

Sokunphal T, Othman W A F W, Alhady S S N, Rahiman W. Pi controller design for velocity control of a mobile robot. *J. Fundam. Appl. Sci.*, 2018, *10(3S)*, 890-902.