Control for balancing line follower robot using discrete cascaded PID algorithm on ADROIT V1 education robot

Conference Paper · September 2015

DOI: 10.1109/ELECSYM.2015.7380849

4 authors, including:

Eko Henfri Binugroho
Electronics Engineering Polytechnic Institute of Surabaya

Derry Pratama

Dadet Pramadihanto
Electronics Engineering Polytechnic Institute of Surabaya

Control for Balancing Line Follower Robot using Discrete Cascaded PID Algorithm on ADROIT V1 Education Robot

Eko Henfri Binugroho[1], Derry Pratama[2], Akhmad Zackarya Rizqy Syahputra[3], Dadet Pramadihanto[2]
[1]Mechanical and Energy Engineering Department
[2]Computer and Informatics Engineering Department
[3]Electrical Engineering Department
Electronic Engineering Polytechnic Institute of Surabaya
sragen@pens.ac.id, derryprimero@gmail.com, z4ck.pott3r@gmail.com, dadet@pens.ac.id

Abstract*—Robotics has been widely used in education as a learning tool to attract and motivate students in performing laboratory experiments within the context of mechatronics, electronics, microcomputer, and control. In this paper we propose an implementation of cascaded PID control algorithm for line follower balancing robot. The algorithm is implemented on ADROIT V1 education robot kits. The robot should be able to follow the trajectory given by the circular guideline while maintaining its balance condition. The controller also designed to control the speed of robot movement while tracking the line. To obtain this purpose, there are three controllers that is used in the same time; balancing controller, speed controller and the line following controller. Those three controllers are cascaded to control the movement of the robot that uses two motors as its actuator. From the experiment, the proposed cascaded PID controller shows an acceptable performance for the robot to maintain its balance position while following the circular line with the given speed setpoint.

***Keywords*—ADROIT V1; Balancing robot; line follower; cascaded PID.**

I. INTRODUCTION

Recently, educational robotics has attracted the high interest of teachers and researchers as a valuable tool to develop cognitive and social skills for students. It has been widely developed for students from elementary school to high school to support learning in science, mathematics, technology, informatics and other school subjects or interdisciplinary learning activities [1-2]. Through the design, creation, assembly and operation of robots, educational robotics can support educational activities in strengthening specific areas of knowledge and skills developed by students. By using it, the student will try to learn the current trends in automation technology which is related to the use of mechanical, electronic, control, and computer-based programming.

Balancing robot would be an interesting model or tool to teach the student about the basic principle of the control. The robot could maintain its standing position by controlling its tilt angle by using tilt sensor. It will be an ideal object to teach the student about mechatronics, which includes electronic devices and embedded control algorithm. Two wheeled balancing robot has the ability to balance on its two wheels and spin on its center spot. This maneuverability allows navigation on the various terrains, traverse small step or curbs, and turn sharp corner easier. This robot also has a relatively smaller footprint compared with three or four wheeled robots, thus enable it to travel around corridors and tight corners more easily [3]. These capabilities make it also has a potential in solving many challenges in industry and society.

Balancing robot needs a controller to maintain its upright position without external support. Since its wheels can move on the plane surface, thus additional heading angle control to the robot makes the controller become more complex. In this paper we propose a controller that combines the balancing capability of the robot with the speed controller to make the robot move at the desired speed while maintaining its upright position. Line follower controller also added in the system to make the robot to be able to follow the given guide line. Those controllers are cascaded in the system to make the robot able to follow the given guide line at the desired speed, while still maintaining its balance condition. Control of the line follower balancing robot is quite challenging since a single input force has to control the balance, speed and steering in the same time. Thus, for successful control of the robot, three variables, namely a balancing angle, a steering angle and its speed should be controlled properly to drive the speed of DC motor connected to the wheels.

To make the proposed controller works, several sensors are used in the system, such as IMU sensor for balancing control, rotary encoder for speed sensor and array of reflective color sensor for the line sensor. The proposed controller is implemented for two wheel balancing robot model using ADROIT V1 education robot kit. The robot kit uses Atmel ATMega 128 as its main microcontroller [4-5]. Since the cascaded controller algorithm is implemented in the relatively small microcontroller, thus an efficient discrete control algorithm is needed to make the algorithm can be calculated at an acceptable rate.

II. RELATED WORKS

The principle of the self-balancing robot has been widely used for real world applications. One of them is for transportation tools, such as; Segway PT, Ninebot, Freego, Solowheel and many others. Even though many commercially products are already available, many researchers still eager to carry out the study the balancing robot related topics. Currently, various controllers has been implemented on two wheeled balancing robot such as Pole-Placement Controller [6] Linear Quadratic Regulator [6-7], Fuzzy Logic Controller [8], and Fuzzy Proportional Integrated Derivative [9-10], and Adaptive Neural Network [11]. But most of those paper are still focus on the balancing control it self.

Another research also try to control the speed and position of the balancing robot by using partial feed back linearization [12], but the set point still come from the predefined value. The two wheel balancing robot with the line following capability also has been proposed [13], but it only use two bits line sensor and without controlling the robot rated speed. Another balancing line follower robot also has ben developed, namely Balbot [13]. It uses CCD camera as it visual sensor and PID controller for balancing and visual servoing control.

In this research we use ADROIT V1 education robot kits [4] which is an alternative robot design kit that can be used for Robotics in Education program. Its modularity make it enables to be assembled and constructed by user in to several form. Balancing robot is one of the model that can be constructed. Adroit V1 uses a main 9 bit UART communication protocol to access its external sensor [3], such as reflective sensor array used as the line sensor in this paper. It also equipped with internal IMU sensor based on Ivensense MPU 6050 consisting 3 axis accelerometer and 3 axis gyroscope.

III. SISTEM MODELING

Two wheel balancing robot is a non-holonomic system, thus robot position x_R, y_R, and the heading angle φ of the robot could be realized by using DDMR model [13].

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}cos\varphi - \frac{Rd}{L}sin\varphi & \frac{R}{2}cos\varphi + \frac{Rd}{L}sin\varphi \\ \frac{R}{2}sin\varphi + \frac{Rd}{L}cos\varphi & \frac{R}{2}cos\varphi - \frac{Rd}{L}sin\varphi \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (1)$$

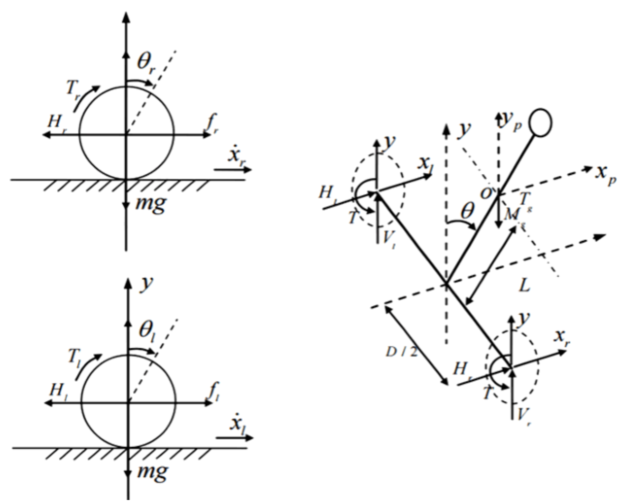
Where:

R : wheel radius
d : distance between center rotation and wheel axis
L : distance between left and right wheel
ω_r : angular speed of the right wheel
ω_l : angular speed of the left wheel

Since the center of rotation is aligned with the wheel axis, then d will be equal to 0, thus Eq. 1 could be simplified as

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}cos\varphi & \frac{R}{2}cos\varphi \\ \frac{R}{2}sin\varphi & \frac{R}{2}cos\varphi \\ \frac{rR}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (2)$$

The dynamic model of the robot in this paper is divided in to two parts, lower body (wheel) and the upper body.



(a) Lower Body (robot's wheel) (b) Overall body
Fig. 1 Free body diagram of two wheels balancing robot

m : weight of the driving wheel
M : weight of the robot body
J_ω : inertial moment of the driving wheel
J_b : inertial moment of the robot body
D : distance between left and right wheel
L : distance between CoG to rotation axis
θ : tilt angle of the body
x_r : displacement of the right wheel in x axis
x_l : displacement of the left wheel in x axis
x_m : displacement on the x direction
T_r : torque of the right driving wheel
T_l : torque of the left driving wheel

By using Newton analysis, the system and operation of the forces can be written in the following kinetics equations:

$$\frac{(T_r+T_l)}{R} = \ddot{x}_m\left(\frac{M+2m+2J_\omega}{R^2}\right) + \left(\ddot{\theta}cos\theta - \dot{\theta}^2\ \sin\theta\right)ML \quad (3)$$

$$\ddot{\theta}J_b = M\left(g - L\ddot{\theta}\ \sin\theta - L\dot{\theta}^2\ \cos\theta\right)L\sin\theta - \left[\ddot{x}_m M + \left(\ddot{\theta}\ \cos\theta - L\dot{\theta}^2\ \sin\theta\right)ML\right]L\cos\theta - (T_r + T_l) \quad (4)$$

The model shown in Eq. 3,4 are not linear system. A linearization can be used in order to simplify the model by using assumption that the operation point of the robot is at its straight up position, thus tilt angle angle θ is in zero position. This linearization will simplify Eq. 3, 4 in to Eq. 5,6 respectively.

$$\frac{(T_r+T_l)}{R} = \ddot{x}_m\left(\frac{M+2m+2J_\omega}{R^2}\right) + ML\ddot{\theta} \quad (5)$$

$$\ddot{\theta} = \left[\left(\ddot{x}_m M + ML\ddot{\theta}\right)L - (T_r + T_l)\right]/J_b \quad (6)$$

Since the robot uses a brushed DC motor gearbox, thus torque generated by the motor can be calculated as follow:

$$T = \frac{K_m K_e N^2 \omega_m + N K_m V}{R_a} \quad (7)$$

Where K_m and K_e are the motor constants, R_a is the motor armature resistance, N is the reduction ratio, ω_m is the motor rotation speed, T is the torque of the output shaft, and and V is the motor voltage. From Eq.7, motor voltage that needed to generate torque T can be calculated, as shown in Eq. 8.

$$V = \frac{T R_a - K_m K_e N^2 \omega_m}{N K_m} \quad (8)$$

IV. SYSTEM DESIGN

The proposed controller is implemented using ADROIT V1 Education robot depicted in the Fig. 2.

Fig. 2 Line Follower Balancing Robot using ADROIT V1

Robot uses array of 8 line reflective sensor module as depicted in Fig. 3. This sensor is connected to the main controller using UART communication interface.

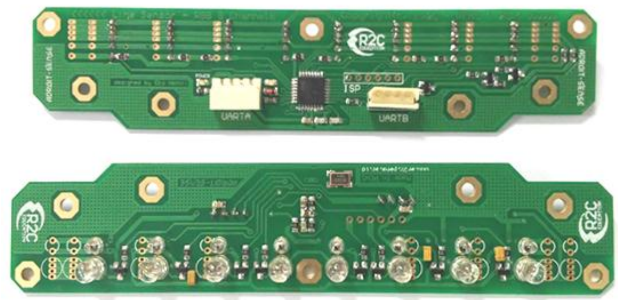


Fig. 3 RGB Line Sensor Module

To control the the robot a cascaded PID controller designed. The overall proposed control diagram is shown in Fig.4. The value of *Center of Balance offset* is found using steady balancing condition. It is done by removing the line follower control and the speed control from the system, and then adjust its value so the robot standing still in this position.

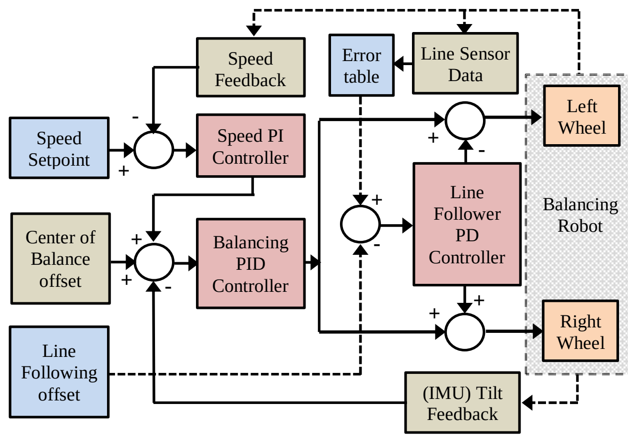


Fig. 4 Overall Cascaded PID Diagram

The balancing controller use PID method to calculate the balancing control signal $u_{B(k)}$. The center of balance of this controller uses the accumulation of *center of balance offset* value with the speed control signal $u_{S(k)}$ from the speed controller.

$$u_{B(k)} = K_{pB} e_{B(k)} + K_{iB} i_{B(k)} + K_{dB} G_{y(k)} \quad (9)$$

$$i_{B(k)} = i_{B(k-1)} + \left(e_{B(k)} + e_{B(k-1)}\right)\frac{T_s}{2} \quad (10)$$

Instead of differentiating the balance error value e_B, the rate error of the balancing control uses the pitch rate from the gyroscope sensor data in the y axis (G_y) directly. The value of integral balancing error from the controller is calculated using trapezoid method shown in Eq. 10.

Speed feedback of robot v is calculated from the calculation of the each motor speed v_r , v_l by sampling the elapsed rotary encoder data E mounted on the robot's motor using fixed time sampling T_s.

$$v_{(k)} = \frac{v_{r(k)}+v_{l(k)}}{2} \tag{12}$$

$$v_{r_{(k)}} = \frac{\left(E_{r(k)}-E_{r(k-1)}\right)}{T_s} \tag{13}$$

$$v_{l_{(k)}} = \frac{\left(E_{l(k)}-E_{l(k-1)}\right)}{T_s} \tag{12}$$

The speed controller uses PI method. The speed control signal $u_{S_{(k)}}$can be calculated in Eq. 12. The speed error $e_{S_{(k)}}$is calculated by subtracting speed setpoint S_{sp} with the robot's actual speed $v_{(k)}$.

$$u_{S_{(k)}} = K_{pS}e_{S_{(k)}} + K_{iS}i_{S_{(k)}} \tag{13}$$

$$e_{S_{(k)}} = S_{sp} - v_{(k)} \tag{14}$$

$$i_{S_{(k)}} = i_{S_{(k-1)}} + e_{S_{(k)}}T_s \tag{15}$$

The value of integral speed error from the controller is calculated using rectangular method shown in Eq. 15.

The line follower controller use PD method. The line follower error $e_{F_{(k)}}$ is obtained using quantitation of the line position with regards to Line Following offset F_{os}. This quantitation error values are shown in TABLE 1. The line follower control signal $u_{F_{(k)}}$can be calculated from Eq. 15

$$u_{F_{(k)}} = K_{pF}e_{F_{(k)}} + K_{dF}\left(\frac{e_{F(k)}-e_{F(k-1)}}{T_s}\right) \tag{16}$$

TABLE I. QUANTITATION ERROR OF LINE FOLLOWER SENSOR

Sensor Data (binary)	Quantitation Error $e_{B_{(k)}}$
10000000	-12
11000000	-7
01000000	-8
01100000	-5
00100000	-3
00110000	-2
00010000	-1
00011000	0
00001000	1
00001100	2
00000100	3
00000110	5
00000010	7
00000011	9
00000001	12
Others	$e_{B_{(k-1)}}$

The rated distance of each reflective sensor shown in Fig. 3 is 1.5 cm, with the guideline width at 2 cm. In this set up 1 sensor or 2 sensors can detect the line in the same time.

The line follower control signal $u_{F_{(k)}}$ will give a steering signal to the robot's wheel when combined with balancing control signal $u_{B_{(k)}}$. The the driving signal for left motor $V_{l_{(k)}}$ and right motor $V_{r_{(k)}}$ can be calculated using Eq. 17 and Eq. 18 accordingly. These driving signals will be used to generate motor torque for the robot as shown in Eq. 7.

$$V_{l_{(k)}} = u_{B_{(k)}} - u_{F_{(k)}} \tag{17}$$

$$V_{r_{(k)}} = u_{B_{(k)}} + u_{F_{(k)}} \tag{18}$$

V. EXPERIMENT RESULTS

The experiments are done by running the robot in the circular shape guide line with 40 cm radius. The measurement data is sent using telemetry system via Bluetooth to PC. The tilt angle error of the balancing control is measured in degree, and the line follower error is measured in unit (based on the quantitation error shown in TABLE 1). Robot speed is also measured in speed unit, since it use data sampling from the encoder directly with the rotary encoder attached in each rotor of the motor. One speed unit of the speed controller is equal to 6.6937 cm/s. This value can be calculated from Eq. 19, with wheel radius R = 3cm, gearbox ratio N=17.6:1, and encoder resolution E_{Res}=16 pulse per rotation (ppr).

$$S_{(k)} = v_{(k)}2\pi R/NE_{Res} \tag{20}$$

The sampling time for telemetry data and the control update T_s is 10 ms. This time interval T_s is generated by using timer interrupt from the robot's main microcontroller.

In this cascaded PID method, the PID constants were tuned separately. The balancing control needs to be tuned first to make the robot stands balanced, then the speed control was tuned to make the robot moves according to the speed set-point input. After that, the line follower control is tuned to control the balance steering. The balance control tuning started with only P constant in the first time. P constant was used and tuned until the robot oscillates a little. Then the D constant was used to reduce the oscillation. The integral constant was added to correct the robot steady state error.

The speed control was added after the balancing robot successfully stands balanced without moving. The P constant was used to make the robot moves based on speed input, it was tuned until the robot moves correctly. After that, the integral constant was added to reduce the steady error which was causing the robot unable to stop. The integral constant tuned by observing the robot displacement from the starting point while the speed input is zero. After the speed control was finely tuned. The line following control was added. The P control was tuned by experiments on straight line. It was tuned until the robot moves along the line with small oscillation, and the D constant was added to reduce that oscillation. Finally, the PID gain constants for proposed controller used in this experiment can be seen from TABLE 2.

TABLE II. PID GAIN CONSTANTS

Gain Constant	Value
Proportional Balancing Control K_{pB}	14
Integral Balancing Control K_{iB}	40
Differential Balancing Control K_{dB}	0.1296
Proportional Speed Control K_{pS}	0,6
Integral Speed Control K_{iS}	5
Proportional Line Follower Control K_{pF}	30
Differential Line Follower Control K_{dF}	0.5

The experiments record the balancing tilt angle, error quantitation for line sensor data, and the robot speed during its travel following the line for 5 second long. In the first experiment, the given speed setpoint is 2 unit. The tilt and line sensor error can be seen from Fig. 4, while the speed response is shown in Fig. 5. From Fig.4, we found that the line sensor quantitation error is quite small. This means that the robot can follow the guide line well. But the speed respond has relatively big error up to 100% from its set point.

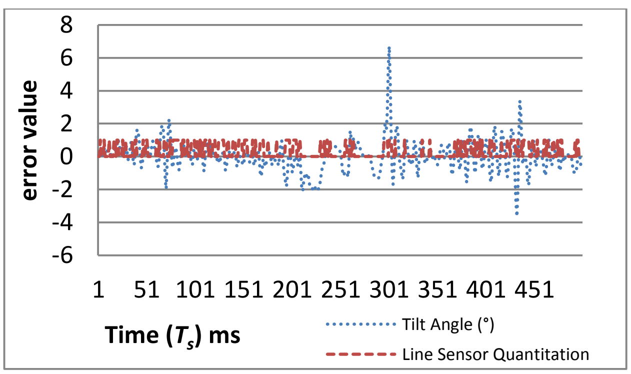


Fig. 4 Error of Tilt Angle and Line Sensor Quantitation during Speed setpoint = 2 unit speed (13.38 cm/s)

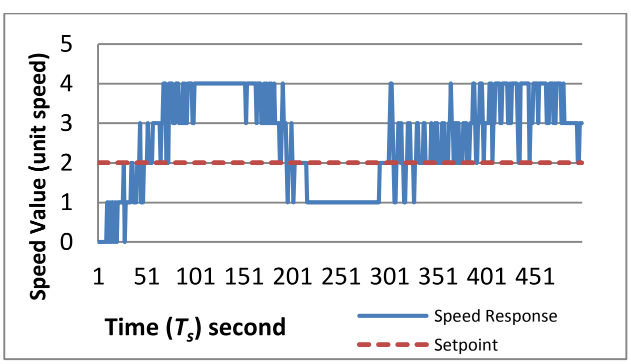


Fig. 5 Speed response during setpoint = 2 unit speed (13.38 cm/s)

I the second experiment, speed setpoint = 5 unit is given to the controller. The tilt and line sensor quantitation error for this speed can be seen from Fig. 6, while the speed response is shown in Fig. 7.

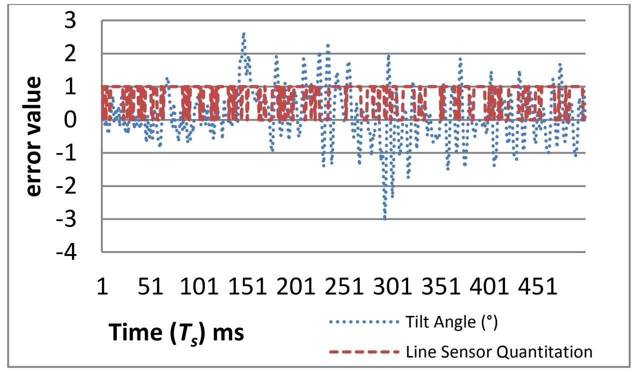


Fig. 6 Error of Tilt Angle and Line Sensor Quantitation during Speed setpoint = 5 unit speed (33.4517 cm/s)

From Fig.6, we found that the line sensor quantitation error is still quite small. It is oscillated between 0 and 1 with higher frequency that the previous experiment. This oscillation is caused by the response of the line sensor when following the circular guide line with 40 cm radius. This experiment shows that the, in this speed the robot can follow the guide line very well. The speed respond shown in Fig. 7 is better than the previous one, but overshoot is still quite high up to 120%. There also steady state error remains from the speed controller.

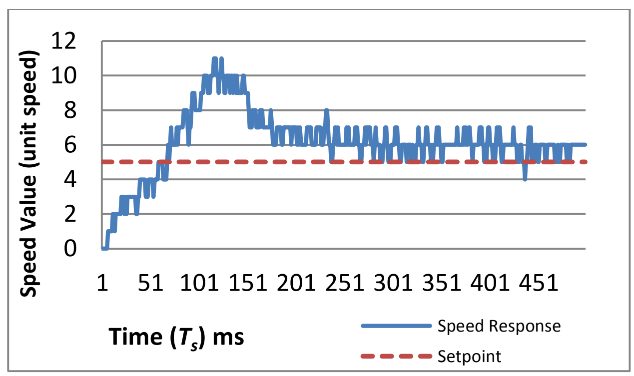


Fig. 7 Speed response during setpoint = 5 unit speed (33.4517 cm/s)

I the last experiment, speed setpoint = 8 unit is given to the controller. The tilt and line sensor quantitation error for this speed can be seen from Fig. 8, while the speed response is shown in Fig. 9. Fig. 8 shows that the line sensor quantitation error become bigger than the previous speed. The values start to oscillate and cross the center sensor position (zero value). Even though the error is bigger, the robot still can follow the line quite well at this speed. The sensor quantitation error never exceeds ±3 unit.

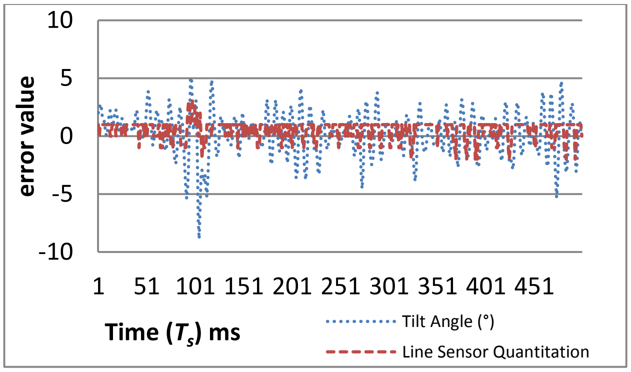


Fig. 8 Error of Tilt Angle and Line Sensor Quantitation during Speed setpoint = 8 unit speed (53.5227 cm/s)

The speed response at 8 speed unit as shown in Fig. 9 is a not as good as at 5 speed unit. Even though the overshoot value is smaller, it has bigger steady state error.

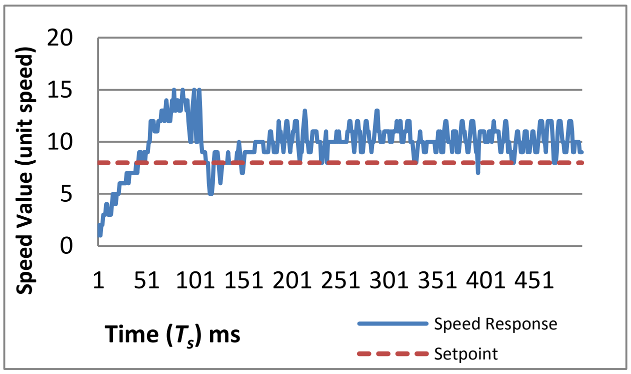


Fig. 9 Speed response during setpoint = 8 unit speed (53.5227 cm/s)

VI. CONCLUSIONS

In this paper, a discrete cascaded PID for self-balancing line follower robot was presented. The line follower and the balancing controller of the proposed controller applied in the ADROIT V1 robot shows good results. The controller can make the robot to be able to follow the given guide line, while still maintain its upright position. The robot is able to work well in following 40 cm radius of circular path in the speed up to 53.52 cm/s. The down side of the controller is its speed response. The proposed controller still has a big overshoot and a relatively high steady state speed error. But overall, the proposed controller shows and acceptable result to be implemented for line follower self-balancing robot application.

ACKNOWLEDGEMENTS

This research was conducted by authors with EEPIS Robotics Research Center (ER2C) in Electronic Engineering Polytechnic Institute of Surabaya (EEPIS). This work was supported by EEPIS Center for Research and Community Service (*Pusat Penelitian dan Pengabdian kepada Masyarakat)* Politeknik Elektronika Negeri Surabaya and Ditlitabmas Dikti.

REFERENCES

[1] Mataric, M. J., “Robotics Education for All Ages”, Proceedings, *AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education*, Spring 2004, Palo Alto, CA.

[2] Johnson, J., “Children, robotics, and education”, *Artificial Life and Robotics*, 7 (1-2), pp. 16-2, 2003.

[3] Vivien Coelho, Stanley Liew, Karl Stol, Guangyu Liu. “Development of a Mobile Two-Wheel Balancing Platform for Autonomous Applications”. Department of Mechanical Engineering, the University of Auckland. 15th International conference on Mechatronics, 2008.

[4] Eko Henfri B., Achmad S. KH., Dadet P., Rahardhita W. S., "ADROIT V1: Design of Low-cost Modular Educational Robot Platform”, International Electronics Symposium (IES), pp. 26-31, 2014.

[5] Eko Henfri B., Endah S. N., Adnan Rachmad A. B., “Design of Communication Protocol based on 9 Bit UART on ADROIT Education Robot”, International Electronics Symposium (IES), pp. 32-37, 2014.

[6] Junfeng Wu, Wanying Zhang, “Research on Control Method of Two-wheeled Self-balancing Robot”, Proceedings of the 2011 Fourth International Conference on Intelligent Computation Technology and Automation, vol. 01, pp. 476-479, 2011.

[7] Chenxi Sun, Tao Lu, Kui Yuan, “Balance control of two-wheeled self-balancing robot based on Linear Quadratic Regulator and Neural Network”, International Conference on Intelligent Control and Information Processing (ICICIP), pp. 862-867, June 2013.

[8] Junfeng Wu, Wanying Zhang, “Design of fuzzy logic controller for two-wheeled self-balancing robot”, The 6th International Forum on Strategic Technology (IFOST), Vol. 2, pp. 1266 – 1270, 22-24 Aug. 2011.

[9] Abdullah, W. Hassan, Rahim, “PID control behavior and sensor filtering for a self balancing personal vehicle”, International Conference on Robotics and Artificial Intelligence (ICRAI), 22-23 Oct. 2012, pp. 7-10.

[10] Congying Qiu, Yibin Huang, The Design of Fuzzy Adaptive PID Controller of Two-Wheeled Self-Balancing Robot, International Journal of Information and Electronics Engineering, vol. 5, no. 3, pp. 193-197, May 2015.

[11] C.C. Tsai, H.C. Huang, S. C. Lin, ”Adaptive neural network control of a self-balancing two-wheeled scooter”, IEEE Transaction on Industrial Electronics Society, Vol. 57, No. 4, pp. 1420–1428, 2010

[12] L. Pathak, J. Franch, S. K. Agrawal, “Velocity and position control of a wheeled inverted pendulum by partial feedback linearization”, IEEE Transaction on Robot and Automation, Vol. 21, No. 3, pp. 505–513, 2005.

[13] Nor Maniha Abdul Ghani, Faradila Naim, Tan Piow Yon, Two Wheels Balancing Robot with Line Following Capability, World Academy of Science, Engineering and Technology, vol.5, 27-07-2011.

[14] G. H. Lee, Seul Jung, "Line Tracking Control of a Two-Wheeled Mobile Robot Using Visual Feedback", International Journal of Advanced Robotic Systems, vol. 10, no. 177, pp. 1-8, 2013.