A Low Cost Mobile Robot Based on Proportional Integral Derivative (PID) Control System and Odometer for Education

To cite this article: R Haq et al 2018 J. Phys.: Conf. Ser. 997 012046

View the article online for updates and enhancements.

Related content
- Comparison of PID and Fuzzy Controller for Position Control of AR.Drone
  A Prayitno, V Indrawati and IT Trusulaw
- A low cost indoor localization system for mobile robot experimental setup
  S Adinandra and A Syarif
- Tuning of PID controller using optimization techniques for a MIMO process
  S Thulasi dharan, K Kavyarasan and V Bagavaneeswaran
A Low Cost Mobile Robot Based on Proportional Integral Derivative (PID) Control System and Odometer for Education

R Haq¹, H Prayitno¹, Dzulkiflih¹, I Sucayho¹, E Rahmawati¹,a

¹Departement of Physics, Fakulty of Mathematics and Natural Sciences, Universitas Negeri Surabaya, Surabaya 60231, Indonesia

aE-mail: endahrahmawati@unesa.ac.id

Abstract. In this article, the development of a low cost mobile robot based on PID controller and odometer for education is presented. PID controller and odometer is applied for controlling mobile robot position. Two-dimensional position vector in cartesian coordinate system have been inserted to robot controller as an initial and final position. Mobile robot has been made based on differential drive and sensor magnetic rotary encoder which measured robot position from a number of wheel rotation. Odometry methodology use data from actuator movements for predicting change of position over time. The mobile robot is examined to get final position with three different heading angle 30°, 45° and 60° by applying various value of KP, KD and KI constant.

1. Introduction

Over the past decades mobile robots with various functions, designs and sizes have been developed for fulfilling the needs of education and research as well. Learning and developing robots involve multidisciplinary field of study including mechanical design, electronics, programming and control system, etc. Motivation in this research is to accommodate the needs of mobile robot in a wide area of scientific learning by experiments.

The mobile robot has been implemented as a learning media in control system topic which is a course of physics instrumentation in Physics Department of Universitas Negeri Surabaya. Based on previous research [1-4], in order to support theoretical concepts on control system that were explained by lecturers in classroom, experiment based works can be done by students must be provided. By conducting laboratory works, it is believed that the students will learn the theoretical concepts easily and grasp them more. Rickel [1] mentioned that students retain 25% of what they hear, 45% of what they hear and see, and 70% if they do what they hear and see or known as “learning by doing” method.

Currently, the evolution of technology achieved a significant reduction in the cost of prototype mobile robot that introduce new and more sophisticated capabilities. In the beginning, mobile robots used to line follower sensor for navigation by following a line and then develop rapidly by implementing various navigation system such as odometer for navigation system. The mobile robot of this research meets a low manufacturing cost about 800000Rp or about 60$.

Odometry is one of the most common relative position estimation methods. In most mobile robots, the offset from known starting position is computed by data of encoders which are mounted on wheels to monitor the wheel rotation steering or heading angle of robot wheels. Implementation of odometry can be independently without external information such as additional sensor or system such as camera. Therefore, odometry is simple, inexpensive and easy to be applied in real time [5-9].
However, odometry is strongly influenced by internal systemic factors and non-systemic factors which cause the accumulation of error. Systemic error has nothing to do with external environment which almost constant for a period of time. In order to reduce systemic error, PID control method have been implemented to model the systemic odometry errors and implemented practically on actual robots for validity. Calibration of error for for accurate odometry is necessary. In this article, the mobile robot is calibrated by straight line path and a turning on the spot experiments which only needs smaller experiment site [8].

2. Research Method
2.1. Determining position
One of the commonly used mobile robot drive controllers is the differential controller. This drive system consists of two wheels mounted in a single axis and each wheel can be independently controlled to rotate either clockwise or counter-clockwise.

![Figure 1. Position vector in cartesian coordinate](image)

The position vector is a quantity having a value and direction indicating the position of a point to the reference point of a plane. In this research using position vector on 2-dimensional plane which has 2 axes $x$ and $y$. The position of an object $P$ to point $O$ is written $(x, y)$ or $x\hat{i} + y\hat{j}$ and $R$ is the length of the vector.

$$R = \sqrt{x^2 + y^2}$$ (1)

$$\theta = \tan^{-1} \frac{y}{x}$$ (2)

To estimate the relative position of the robot, a constant variable required to convert the counter count generated by the rotary encoder sensor is the length of the path the robot takes. This constant is obtained by using the following equation:

$$C = \frac{K_{\text{wheel}}}{\text{resolution of encoder}}$$ (3)

Where $C$ is the variable constant (mm / counter), $K_{\text{wheel}}$ is the circumference of the wheel used (mm) and the encoder resolution is the number of counters in one turn. The distance of the right wheel and the left wheel are respectively $d_{\text{right}}$ and $d_{\text{left}}$ and the distance between the two wheels is given by $l$, the distance distance and the orientation angle $\theta$ can be formulated by the following equation:

$$d_{\text{right}} = \text{count}_{\text{right}} \times C$$ (4)

$$d_{\text{left}} = \text{count}_{\text{left}} \times C$$ (5)

$$\text{distance} = \frac{(d_{\text{right}} + d_{\text{left}})}{2}$$ (6)

$$\theta_{\text{rad}} = \frac{(d_{\text{right}} + d_{\text{left}})}{l}$$ (7)
Furthermore, $\theta$ in equation (7) is in radians form and should be converted to degree by equation:

$$\theta_{heading} = \theta \times \frac{180}{\pi}$$  \hspace{1cm} (8)

By knowing distance and orientation of angle $\theta_{heading}$ hence can be determined coordinate of position of robot by using concept of position vector in real time.

![Figure 2. Determining mobile robot position](image)

$$x_{pos} = distance \times \sin \theta_{heading}$$ \hspace{1cm} (9)

$$y_{pos} = distance \times \cos \theta_{heading}$$ \hspace{1cm} (10)

2.2 Hardware and control system

This mobile robot uses two DC motors as its motion device. This DC motor which connected to the rencoder were installed across the rear of the robot. The speed comparison of these two DC motors will determine the direction of robot motion. When the right motor and left motor rotate at the same speed then the robot will run straight. When the right motor speed is greater than the left motor speed then the robot will tend to turn left. When the right motor speed is smaller than the left motor speed then the robot will tend to turn right.

![Figure 3. Hardware of mobile robot](image)
Target in the form of coordinate \( x\hat{i} + y\hat{j} \) is inserted on the microcontroller and will be stored on EEPROM. From the data \((x, y)\) the microcontroller will calculate the angle \(\theta\) the destination which is then regarded as a set of points, then drive the drive system connected to the rotary encoder sensor. The data from the rotary encoder will be fed back to the microcontroller and converted into position data \((x_{pos}, y_{pos}, \theta_{pos})\). The data will be compared with the set points so that the error is obtained. Error will be entered into the PID equation on the microcontroller which will then affect the increase or decrease the speed of the drive system.

In this research manipulation variable is the value of the constant PID is \(K_P\), \(K_I\), and \(K_D\). The control variable used is the diameter of the wheel and the distance between the two wheels mobile robot. Respond variable is the value to be obtained in a study. Variable respond in this research is the position of ahir robot \(x\hat{i} + y\hat{j}\) which has been passed by mobile robot.

2.3. Calibration

The data retrieval process begins with rotary encoder sensor calibration using equation (1). By inserting \(x\) and \(y\) position, then the mobile robot will move towards the coordinates automatically. When the mobile robot has reached the intended coordinates then the mobile robot will stop and display will inform final coordinates \((x', y')\).

 Rotary encoder sensor testing is done by moving the mobile robot with a distance of 10 cm to 100 cm and compare the value of the distance read by the sensor with the actual distance measured by the ruler and then repeated as much as 3 times the measurement for each mobile robot mileage. This measurement aims to test the accuracy of rotary encoder sensor precision. Figure 5 shows that the right and the left wheel data are fit together and there is almost no difference.

3. Result

Testing of mobile robot control system is done on a flat field that has been given a pattern of diesi kartesius to make it easier to observe the position of mobile robot. At first mobile robot is placed at the center of Cartesian diagram with the front of the robot parallel to the \(Y\) axis, then the researcher gives data input in the position of \(x\hat{i} + y\hat{j}\) which becomes the end of mobile robot. From this data the
controller will calculate the angle of heading robot and the distance that will be taken mobile robot, then mobile robot will move towards that position automatically. The test is repeated five times for each manipulation.

3.1. **Heading error**

Final position is the position when the robot has stopped. Error is percentage of mobile robot position difference to set point. Different tuning of PID constant induce different respons of robot to attain the set point.

![Image](image_url)

**Figure 6.** Error of heading for 30°

Heading error is the difference between the robot heading angle in any time with the heading of the robot as set point. Figure 6, 7 and 8 show the change in heading error value every 50 ms when mobile robot starts to move until stop. When $K_p = 0, K_i = 0, K_d = 0$ (without PID controller) the heading error value is randomly changed and can not reach zero. This happens because mobile robots try to reach set point without feedback for correcting its position. While the PID controller is applied to the robot, initially the error heading is at a maximum value of 30° and continues to fall to close to or around zero. This result showed that the control system will try to maintain the value of the heading robot according to the set points and the heading error is zero.

Figure 6 shows four graph of heading error for heading angle of 30° with different PID constant include of; $K_p = 0, K_i = 0, K_d = 0$ (without controller); $K_p = 10, K_i = 2, K_d = 10$; $K_p = 10, K_i = 0, K_d = 10$ and $K_p = 15, K_i = 0, K_d = 0$. PID constant of $K_p = 10, K_i = 0, K_d = 10$ reach the set point 100 ms faster then the other constants and has the smallest error of heading angle that is 0,44%. Data from five times running test for each tuning of PID constant shows a good repeatability.
Figure 7. Error of heading for 45°

Gambar 8. change of error for heading 60°

Figure 7 shows four graphs of heading error for heading angle of 45° with different PID constant tuning include of; $K_P = 0, K_I = 0, K_D = 0$ (without controller); $K_P = 10, K_I = 2, K_D = 0$; $K_P = 10, K_I = 0, K_D = 10$ and $K_P = 15, K_I = 0, K_D = 0$. PID constant of $K_P = 10, K_I = 0, K_D = 10$ and of $K_P = 10, K_I = 0, K_D = 0$ cause the same error of heading of 0.42%. Heading error of PID constant $K_P = 10, K_I = 0, K_D = 10$ try to reach set point without oscillation, the other experienced it before get
the set point.

By manipulating the destination position of 1200i + 700j, we get the heading angle of robot about 60° and the different responses by four different PID constant tuning was depicted in figure 8. Maximum and minimum heading error respectively are 10% (6°) and 0.06% (less than 1°)

3.2. PID response

In general, the response of a control system shows a response characteristic such as rising time, settling time and error. Rising time is the time it takes the system to pass through 90% set point value, settling time is the time it takes the system to reach the smallest error value.

Figure shows the comparison of robots heading response to set point at any time for each PID constant. Initially the heading value is 0° and will rise towards the set of points that have been determined. This illustrates the mobile robot that originally faces parallel to the y-axis and then gradually toward the set of points. It appears that the system works fairly quickly with rising time less than 200 ms for all PID constants values. Rising time is influenced by the magnitude of KP, the greater the value of KP then rising time will be faster.

![Figure 9. PID response for heading 30°, 45° and 60°](image)

The rising time indicated by curve with $K_P = 10$ and $K_P = 15$ is almost similar about 600 ms. After reaching the heading corresponding to the set of points, the system will experience overshoot. The magnitude of the overshoot value is influenced by the value of $K_P$, but at the same $K_P$ value the value of $K_I$ and $K_D$ also affect. Seen in the figure 9 the largest overshoot is achieved when $K_P = 10$, $K_I = 2$, $K_D = 0$.

The time required to achieve the smallest error varies for each PID constant. Settling time is influenced by $K_I$ and $K_D$. The correct tuning of $K_I$ and $K_D$ will result in a stable system response. At set point 30° with $K_I = 2$, $K_D = 0$ obtained settling time 900 ms, when $K_I = 0$, $K_D = 10$ obtained a fast settling time of 400 ms, whereas when $K_I = 0$, $K_D = 0$ obtained 600 ms settling time.

3.3. Implementation in Education

The last a year the mobile robot were used for the laboratory exercises of an undergraduate robotics learning at Physisc Department of Universitas Negeri Surabaya. Typical undergraduate exercise consist
of electrical and programming control system for navigation. The robot also help the students to have a research for a competition. The robot proved user friendly and helped the undergraduate students to quickly learn how use them and adapt them to another projects. The hardware and software design of the robot is a supporting tool for students to comprehend and practice about robotics.

4. Conclusion
In this article, the development of a prototype mobile robot for education is presented. The low cost of mobile robot has been fulfilled the needs of navigation system of robot in laboratory scale. This robot already proved to be reliable for experiments and research for period a year both for laboratory exercising and research for competition preparation. The use of odometry system and PID controller have been implemented for mobile navigation. The implementation provides the position and orientation of mobile. Through PID controller, the error of heading and position can be corrected in real time. Maksimum and minimum error for several PID tuning respectively are less then 10% ($3^\circ$) and 0,42%($0,5^\circ$). So, the approach proposed in this article is viable to reduce the error of navigation or localization.

References
[1] Rickel J W 1989 IEEE Trans. Syst. Man, Cybern 19 40
[2] Rahmawati E, Ekawita R, Budiman M, Abdullah M and Khairurrijal 2012 Telkomnika 10 733
[3] Rahmawati E, Sucahyo I and Khairurrijal 2014 ICAET Proc. 1 155
[4] Khairurrijal, Abdullah M and Budiman M 2011 Comp. Appl. Eng. Educ. 19 10
[5] Ardilla F, Marta B S and Besari A R A 2011 IES Proc. 13 1
[6] Dudek G and Jenkin M 2010 Computational Principles of Mobile Robotics second edition (Cambridge University Perss: New York)
[7] Ang K H, Chong G C Y and Li Y 2005 IEEE Trans. Control Syst. Technol. 13 14 559
[8] Chen W and Zhang T 2017 Int. J. Adv. Robot Syst. 13
[9] Martinelli A 2002 IEEE Trans. on Robot. Autom. 18 1