Speed control system in mobile robot based on Bezier curve trajectory

M F Alfatih*, M A Riyadi and I Setiawan
Department of Electrical Engineering, Faculty of Engineering, Universitas Diponegoro, Semarang, Indonesia

*muhammadfaizalfatih@gmail.com

Abstract. The development of robot technology is currently growing rapidly. Mobile robot is one robot with many applications in everyday life. One of the important things in the field of mobile robots is the problem of trajectories or trajectories, in this case how robots can track approved paths. In this study, the mobile robot control system has been tested to track tracks. The path used in this study is the third-order Bezier Order, with control points and end points of the predetermined pathway. Based on the results of simulation tests on the development of mobile robot trajectories, based on the results of simulation tests on mobile robots, it is found that the performance of the track tracking control system is very dependent on the placement of the Bezier curve control points or in other words the actual path formed by the robot is greatly influenced by the shape of the Bezier trajectory that was designed.

1. Introduction
Robot is a mechanical device that can perform physical tasks [1], either using human supervision and control, using pre-defined programs or utilizing the principles of artificial intelligence [2]. Research on mobile robots has been intensively discussed [3–5]. Furthermore, many mobile robots have been used for various purposes such as industry [6], defense, places that have not been touched by humans (e.g. the Mars Rover Robot is used for planetary exploration, Mars) [7], or even for educational purposes, such as robots educative and also likes things like robot ball competition, grass cutting robot competition, fire source search robot competition [8].

One of the main tasks that must be completed by each mobile robot is to get to the expected locations and find out their position and orientation at all times (navigation system) [9]. A simple technique that can be used is by moving the robot to follow a straight path to the destination to rotate in place according to the expected orientation [10]. Because of the simplicity of control, in general this method is widely used in practical robots, such as those used in nursing robots (nursing robots) designed by Johann Borentein [11]. One of the real disadvantages of using this technique is that robots cannot move smoothly for points that must be passed. So as to avoid this, the robot trajectory designed must be continuous and smooth.

Bezier curves are commonly used in the field of computer graphics and in some literature have also been successfully implemented as robot trajectory [12,13]. Bezier curves can be used to optimize robot paths that are calculated with respect to length and curvature [14]. For robots to follow lane, it is important for the speed of the left and right wheel drive angle to adjust gently. The Bezier curve method can be applied to create curved lines to implementation continuous angular velocity [15].
The drive system basically functions to move to drive the robot from one location to another. In this case there are two types of driving systems on popular mobile robots: Ackerman driving which is commonly used in automobile vehicles and differential driving (often found in wheelchairs with automatic drive).

Based on previous research on mobile robots, there are still shortcomings in the form of reference tracks that are still inaccurate in providing reference links to mobile robots. This paper will present a construction of trajectories for mobile robots based on the Bezier curve method, with differential robot type mobile robots which generally have a degree of flexibility in maneuvering or rotating in their movements.

2. Methods

2.1. Mobile robot kinematics model

One type of mobile robot that is commonly used, especially for indoor operation is a mobile robot with a steering or differential drive system (differential drive) [16]. The main reason is because it is relatively more flexible in maneuvering and ease of control.

This type of robot basically has two main wheels, each of which is driven by a separate drive (generally in the form of a permanent magnet DC motor with gear-reducing which serves to strengthen the motor torque), as a counterweight generally this robot is also equipped with one or two castor wheels which is placed at the back of the robot. Figure 1 shows the robot architecture seen from the top: If the two cogs are rotating at the same speed, the robot will move in a straight direction, whereas if the speed of one of the wheels is slower than the robot will move to form a curve with the direction of the path to one the slower-moving wheels.

![Figure 1](image)

Figure 1. Mobile robot position and orientation in Cartesian coordinate systems [16].

By its nature, this type of mobile robot is included in the category of non-holonomic robot: The position and direction (orientation) are in a coupled state, in this case the robot cannot practically translate to the left (or right) without first changing the robot's orientation.

Based on Figure 1 for the length of the wheel radius \( r \), as well as the rotation speed of each right wheel \( \omega_R \) and left wheel \( \omega_L \) respectively, then the linear speed of the right wheel \( v_R(t) \) and the left wheel \( v_L(t) \) can be found using equation (1) and (2) following [17]:

\[
\begin{align*}
v_R(t) &= r\omega_R(t) \\
v_L(t) &= r\omega_L(t)
\end{align*}
\]
When the robot rotates for a moment with the length of the radius $R$ measured from the center of rotation and the center of the two wheels (Figure 3 can be used as a reference), the rotation speed at each point of the robot will always be the same (the robot is a rigid mechanical system), so the following equations (3) and/or (4) apply to calculate the rotational speed of the robot.

\[
\omega(t) = \frac{v_y}{R + \frac{L}{2}}
\]

(3)

\[
\omega(t) = \frac{v_y}{R - \frac{L}{2}}
\]

(4)

Based on equations (3) and (4) the rotation speed of the robot can be calculated only based on information from the two linear wheels’ speeds of the robot:

\[
\omega(t) = \frac{v_y(t) - v_x(t)}{L}
\]

(5)

Whereas the radius of the path can be searched by substituting equation (5) into equation (6), and solving it for $R$:

\[
R = \frac{L}{2} \left( v_x + v_y \right)
\]

(6)

Seen from equation (6), the radius of the momentary circle trajectory is inversely proportional to the difference in the speed of the two robot wheels, the smaller the difference between the two wheel speeds, the radius of the momentary circle formed by the robot trajectory is getting longer, and vice versa, whereas if $v_y = v_x$, then $R = \infty$, or practically the robot will move to form a straight path. For the robot to rotate at the center of its axis ($R = 0$), based on equation (6), the speed of the two wheels must be the same in the opposite direction or $v_y = -v_x$.

Based on equations (5) and (6), the linear velocity of the robot can be calculated using the following equation (7):

\[
v(t) = \frac{v_y(t) + v_x(t)}{2}
\]

(7)

For simplicity, equations (6) and (7) can be collected in the form of matrix-vector equations as shown in equation (8) below:

\[
\begin{bmatrix}
\dot{v}(t) \\
\dot{\omega}(t)
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{2} & \frac{1}{2} \\
\frac{1}{L} & -\frac{1}{L}
\end{bmatrix}
\begin{bmatrix}
v_x(t) \\
v_y(t)
\end{bmatrix}
\]

(8)

Equation (8) above shows the direct kinematics relation between the linear velocity of the robot wheels and the linear and angular speed of the robot, while equation (9) below shows the opposite relation:

\[
\begin{bmatrix}
v_x(t) \\
v_y(t)
\end{bmatrix} =
\begin{bmatrix}
1 & \frac{L}{2} \\
1 & -\frac{L}{2}
\end{bmatrix}
\begin{bmatrix}
\dot{v}(t) \\
\dot{\omega}(t)
\end{bmatrix}
\]

(9)

By knowing the linear and angular velocity of this robot at any time, then the speed at each Cartesian axis can be found by projecting the robot velocity vectors to these axes, this is shown in equation (10) below:

\[
\begin{bmatrix}
x(t) \\
y(t) \\
\dot{\theta}(t)
\end{bmatrix} =
\begin{bmatrix}
\cos(\theta(t)) & 0 & v(t) \\
\sin(\theta(t)) & 0 & \dot{v}(t) \\
0 & 1 & \dot{\omega}(t)
\end{bmatrix}
\]

(10)

From equation (10) it can be seen that for each moment the position and orientation of the robot in the Cartesian coordinate system can be calculated by integrating each equation term:
\[ x(t) = \int_{0}^{t} v(\tau) \cos(\theta(\tau)) d\tau \]  
\[ y(t) = \int_{0}^{t} v(\tau) \sin(\theta(\tau)) d\tau \]  
\[ \theta(t) = \int_{0}^{t} \omega(\tau) d\tau \]  

The results of solving the equations above are given below:

\[ \theta(t) = \frac{(v_x(t) - v_y(t))}{L} \theta_0 \]  
\[ x(t) = x_0 + \frac{v(t)}{\omega(t)} \left[ \sin(\theta(t)) + \theta_0 \right] - \sin(\theta_0) \]  
\[ y(t) = y_0 - \frac{v(t)}{\omega(t)} \left[ \cos(\theta(t)) + \theta_0 \right] - \cos(\theta_0) \]  

2.2. Motion control

As mentioned earlier, the main purpose of a robot control system is to move the robot from its initial state (P0) to its final location (P3) by following the expected trajectory. This is illustrated in Figure 2:

![Figure 2. Illustration of robot trajectory.](image)

To achieve these objectives a motion control system is needed. At the lowest level, this system basically regulates the rotational speed of each robot's wheels in accordance with the Reference as shown in Figure 3:

![Figure 3. Block diagram control of DC motor in robot mobile.](image)

As discussed earlier, the feedback control system that will be used to control the DC motor is the PID control, with the equation of the transfer function.
\[ CO(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \]  
(17)

With  
\( CO(t) \) = Output controller  
\( K_P \) = Proportional constant  
\( T_I \) = time integral  
\( T_D \) = time derivative  
\( K_I \) = gain integral \((K_P/T_I)\)  
\( K_D \) = gain derivative \((K_P T_D)\)

While the DC motor itself basically has the following order 1 transfer function:

\[ H(s) = \frac{b}{T s + 1} \]  
(18)

with  
\( b \) = gain static DC motor  
\( T \) = time constant DC motor

![Figure 4. Flowchart simulation trajectory.](image)

When the system starts or starts, the operator must input the values of X and Y starting position, end position and control points and robot travel time. The system will calculate the robotic path, curvature and rotational speed of the robot from the start point to the end point. After that the system will run the Bezier curve trajectory generator algorithm to graph the robot's trajectory based on the input value from the operator.

2.3. Simulation of the overall mobile robot system control block
In general, the control system model that was simulated is shown in Figure 5.
In this simulation the quantities of the PID control amplifier was searched empirically for the values of the given DC motor parameters. The ultimate goal of this simulation is basically to look at the dynamics of the robot’s trajectory to reference the expected trajectory.

Bezier curves are parametric curves that are often used in graphical applications and related fields [4,14,18–20]. This curve is determined by several control points and always the initial and final control points are determined. The shape can be changed by moving the control point. The two-dimensional Bezier curve of order n is represented in the following equation:

$$P(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} P_i, \quad t \in [0,1]$$

Where $P(t) = [x(t), y(t)]^T$ and $P_i$ are two-dimensional Bézier curves and control points. Bezier curves have geometrical properties that are useful for path planning:

- Each Bezier curve starts at $P_0$ and ends at $P_n$.
- The curve is tangent to the line connected by $P_0$ and $P_1$ and the line connected by $P_n$ and $P_{n-1}$ respectively at $P_0$ and $P_n$.

$$T(0) = n(P_1 - P_0)$$
$$T(0) = n(P_n - P_{n-1})$$

- The second derivative in $P_0$ must be determined by $P_0, P_1$ and $P_2$, and the second derivative in $P_n$ must be determined by $P_{n-2}$ and $P_{n-1}$.

$$S(0) = n(n - 1)(P_2 - 2P_1 + P_0)$$
$$S(1) = n(n - 1)(P_n - 2P_{n-1} + P_{n-2})$$

According to the above properties, the control points of the Bezier curve are calculated below. The curvature at any point on the Bezier curve is determined by the first and second derivatives with respect to the parameter $u$ as [21]:

$$P(u) = P_0(1-u)^3 + P_1u(1-u)^2 + 3P_2u^2(1-u) + P_3u^3, \quad u \in [0,1]$$

To achieve the path of continuous curvature, we mainly focus on ensuring continuity of curvature at the waypoint connecting two curves.

3. Results and discussion

The application of the Bezier curve method for the trajectory of the mobile robot trajectory was tested with different starting points ($P_0$), control points 1 and 2 ($P_1, P_2$) and end points ($P_3$). The observed results are the accuracy of the track shape with the starting point ($P_0$), control points 1 and 2 ($P_1, P_2$) and end points ($P_3$) entered, the robot’s rotational speed profile, robot wheel speed and robot travel time from starting point ($P_0$) to end point ($P_3$).
3.1. Simulation with starting point \( P_1 = 0.0 \) and end point \( P_3 = 400.400 \) with some varied control points \( P_1, P_2 \), and speed 10 cm/s

![Bezier curve paths with P0 (0,0) and P3 (400,400) and velocity of 10 cm/s.](image)

With reference to equation (24):

\[
P(q) = P_0(1-q)^3 + P_1q(1-q)^2 + 3P_2q^2(1-q) + P_3q q \in [0,1]
\]

Figure 6 shows that the Bezier curve is formed with P0 (0,0), P3 (400,400) and P1 and P2 which are varied and the robot speed is 10 cm/s. From this simulation the system that was built successfully created a robot car reference path based on input parameters x and y. The results of tracing the actual trace of the curve attempted experienced significant differences. As shown in the figure, the final coordinates of each experiment differ. for the first experiment with P1 (150.0), p2 (250,400) detected endpoints (350,410). for the second experiment with P1 (200.0), p2 (300,400) the endpoints detected were (420,320), then for the third experiment with P1 (170.0), p2 (150,400) detected endpoints (300,470).

![Step response in the PID control scheme with the track reference in Figure 6.](image)

In Figure 7, referring to the PID control equation in equation 17:

\[
CO(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}
\]

And with equation 18 for a first order DC motor equation:

\[
H(s) = \frac{0.0274}{0.1 + 1}
\]
The response control system that is applied to the dynamics of the trajectory that is designed has a good and fast level of lift.

3.2. Simulations on variations in parameters P3 with fixed speed and control points (P1, P2)

Figure 8. Bezier curve paths with variation parameters P3 with fixed speed and control points (P1, P2), with speeds of 10 cm/s.

Figure 8 shows the Bezier curve with the endpoint P3 that must be traveled by a robot that is located differently. with the same control points, namely P1 (150.0), P2 (250,400) and P0 (0.0). and also the robot's fixed speed is 10 cm/s. the graph shows that the reference path can be made accurately, but the robot is not able to get to the final point correctly. for the first experiment with P3 (400,400) the robot goes to P3 (348,448), for the second experiment with P3 (400,200) the robot goes to P3 (389,210) and for the third experiment with P3 (300,200) the robot goes to P3 (210,151).

Figure 9. Step response in the PID control scheme with the track reference in Figure 8.

Figure 9 the response of the control system applied to the dynamics of the designed path has a stable lifting rate, and the results are the same as from the first experiment.
3.3. Simulations with variations in speed, but with the same control points (P1 and P2) and end points (P3)

Figure 10. Bezier curve paths with parameters P0 (0.0), P1 (150.0), P2 (250.400), P3 (400.400) and with variation of speeds.

Figure 6 shows that the Bezier curves are formed with P0 (0,0), P1 (150,0), P2 (250,400), P3 (400,400) and different robot speeds. In this experiment the system that was built succeeded in making the reference robot car line in accordance with the input parameters x and y. The results turned out to have significant similarities, so the difference in speed of the robot did not affect the shape of the bezier curve.

Figure 11. Step response in the PID control scheme with the track reference in Figure 10.

In Figure 11, the control system response that is applied to the dynamics of the designed path has a smooth lift rate and different response times in the stability of the control system. For a speed of 10 cm / s with an amplitude of 0.08 at 0.4 seconds, then for a speed of 6 cm / s the output amplitude is 0.20 at 0.5 seconds, and for a speed of 3 cm / s the output amplitude is 0.50 at 0.6 seconds.

4. Conclusions

Based on testing and analysis conducted in this study, the performance of a track tracking mobile robot control system is strongly influenced by the shape of the Bezier trajectory or trajectory given as a reference. For the coordinates of the endpoint to which Xfinal, Yfinal, are, the relatively consistent control points that give the final result with minimal errors are P1=Xfinal/2, 0 and P2=Xfinal/2, Yfinal For differential
type Mobile Robot, there is a restraint of the Y axis's final coordinate destination which is the maximum $X_{\text{final}}/2$.

References
[1] Adriansyah A 2014 Perancangan Localization Menggunakan Metode Dead Reckoning Sinergi 18 25–30
[2] Bounini F, Gingras D, Pollart H and Gruyer D 2017 Modified artificial potential field method for online path planning applications IEEE Intelligent Vehicles Symposium (IV) 180–185
[3] Yu W, Shuo W, Rui W and Min T 2017 Generation of temporal – spatial Bezier curve for simultaneous arrival of multiple unmanned vehicles Inf. Sci. (Ny) 418–419 34–45
[4] Simba K R, Uchiyama N and Sano S 2013 Real-Time Trajectory Generation for Mobile Robots in a Corridor-Like Space Using B´ezier Curves 37–41
[5] Dinçer Ü and Çevik M 2019 Improved trajectory planning of an industrial parallel mechanism by a composite polynomial consisting of B´ezier curves and cubic polynomials 132 248–263
[6] Li H and Savkin A V 2018 An algorithm for safe navigation of mobile robots by a sensor network in dynamic cluttered industrial environments Robot. Comput. Integr. Manuf. 54 65–82
[7] Yang G J and Choi B W 2013 Implementation of joint space trajectory planning using RTO with considering velocity constraints for mobile robots IEEE ISR 1-3
[8] Xianglin Y, Zhiwen Z, Junhao X and Zhiqiang Z 2015 Trajectory Planning for RoboCup MSL Mobile Robots Based on B´ezier Curve and Voronoi Diagram 2552–2557
[9] Dixon J and Henllich O 1997 Mobile Robots Navigation
[10] Illah R 2004 Autonomus Mobile Robots (London, England: MIT Press)
[11] Borenstein J and Koren Y 1985 A Mobile Platform for Nursing Robots IEEE Trans. Ind. Electron. IE-32 2 158–165
[12] Huang J, Farritor S M, Qadi A and Goddard S 2006 Localization and Follow-the-Leader Control of a Heterogeneous Group of Mobile Robots 11 2 205–215
[13] Hwang J H, Arkin R C and Kwon D S 2003 Mobile robots at your fingertips: Bezier curve on-line trajectory generation for supervisory control IEEE Int. Conf. Intell. Robot. Syst. 2 1444–1449
[14] Costanzi R, Fanelli F, Meli E, Ridolﬁ A and Allotta B 2016 Mobile Robots Based on Bézier Curves IFAC-PapersOnLine 49 15 145–150
[15] Ishikawa H, Noguchi K, Maki R and Naitoh H 2009 Path generation with clothoid curve using image processing for two-wheel-drive autonomous mobile robots 42 16
[16] Koren Y and Borenstein J 1991 Potential field methods and their inherent limitations for mobile robot navigation Proc. - IEEE Int. Conf. Robot. Autom. 2 1398–1404
[17] Jusuf Dwi Kurnianto N H, Ali H A and Fernando A 2012 Navigasi Mobile Robot Berbasis Trajektori dan Odometry dengan Pemulihan Jalur Secara Otomatis 66 37–39
[18] Xu L, Wang D, Song B and Cao M 2017 Global smooth path planning for mobile robots based on continuous Bezier curve Proc. - 2017 Chinese Autom. Congr. CAC 2081–2085
[19] Zhou F, Song B and Tian G 2017 Bezier Curve Based Smooth Path Planning for Mobile Robot
[20] Hassani V and Lande S V 2018 Path Planning for Marine Vehicles using Bézier Curves IFAC-PapersOnLine 51 29 305–310
[21] Zhang L, Sun L, Member I, Zhang S, Liu J and Member I 2015 Trajectory Planning for an Indoor Mobile Robot Using Quintic Bezier Curves 757–762