Path Tracking for Nonholonomic Mobile Robot by using Advance Lyapunov-based control laws


*Mechanical Engineering Department, College of Engineering- Al-Nahrain University, Baghdad-Iraq
^Mechanical Engineering Department, University of Technology, Baghdad-Iraq

Received 02 Jan 2019, Accepted 03 March 2019, Available online 04 March 2019, Vol.9, No.2 (March/April 2019)

Abstract

A differential drive wheeled mobile robots are a nonlinear multi input multi output systems. The kinematic model has been developed to control and to adjust the angular velocity of the right and left wheels. This velocity is used in the control law, to ensure stability and reduce the error by using the Lyapunov approach, where non-linear control laws are based on the development of the kinetic model in stability Lyapunov. A kinematic model by using four-state method is suggested for switching between the status of the robot and the status of the required system. It is shown that extending (four-state kinematic model) a well-known compare with a third order error model. The new kinematics control law has been introduced which allows Lyapunov-based control laws to achieve a global approximation of the system based on non-zero reference velocity requirements.

Keywords: Backstepping method, Lyapunov theory, Model Based Predictive Control

1. Introduction

The differential drive is a very simple driving mechanism that is used a lot in fact, especially in the case of small mobile robots. The robot’s body typically has one or more wheels to support the platform and prevent tilting. Two wheels are must be placed on a common axis and perpendicular to the axis of the platform. Each speed of this wheel is controlled by a separate by controller drive. Several research studies the solving of the central problems in order to track a mobile robot in terms of design and implementation of the steering control that the mobile robot must track to follow the desired path accurately and reduce the tracking error.

This paper presents one of the tracking strategies for mobile robots. For smoother movement when the tracking the desired path of the robot, the strategy controller will be divided into two levels when the robot moves on a predetermined path. First step, use the kinematic console to generate the required angular velocity control for the extraction and adjust the velocity of the left and right wheels. The control law uses this desired location with the real direction of the robot to generate the required velocity that will be used as input to the next level. In the second step, based on the Lyapunov approach, the dynamic controller is designed to track the desired path. The control Strategy is illustrated in block diagram figure (1), illustrates the two levels of the kinetic and dynamically controlling control. The desired speed provides by Kinematics controller it will be used to calculate the required torque for the right and right wheels.

![Fig.1 Mobile Robot control in general](image)

The paper is organized as follows. In Section 2, describe the Kinematic Mobile Robot and the error model. Section 3, introduces the expansion of the error model of the Lyapunov method for Mobile Robot. The Mobile Robot Control Act is presented in this section. In Section 4, simulation results are discussed and the conclusion is presented in Section 5.

2. Mobile Robot Kinematic and Error Model

Reducing the tracking error of the differential wheeled Mobile Robot is the fundamental essence motivating of this work in tracking the desired path by designing the perfect nonlinear kinematic control unit on line.
According to Figure (2), the main variables to be entered are the velocity of the left wheel $V_L(t)$ and the velocity of the right wheel $V_R(t)$. The other variables in Figure (2) are as follows: $r$ is the radius of the wheel; $L$ is the distance between the wheels.

$$\omega_R(t) = \frac{V_R(t)}{R(t) + \frac{L}{2}}$$
$$\omega_L(t) = \frac{V_L(t)}{R(t) - \frac{L}{2}}$$

(1)

Tangential or linear velocity can have calculated as

$$V(t) = \omega(t) \times R(t) = \frac{V_R(t) + V_L(t)}{2}$$

(2)

Wheel tangential or linear velocities for right and left are

$$V_R(t) = r \omega_R(t), V_L(t) = r \omega_L(t)$$

(3)

where $\omega_R(t), \omega_L(t)$, the left and right angular velocities of the two wheels are around its axes, respectively.

According to the above relationships, internal robot movement (local coordinates) can be expressed as follows:

$${\dot{X}}_m(t) = V \cdot X_m(t)$$
$${\dot{Y}}_m(t) = V \cdot Y_m(t)$$
$${\dot{\Omega}}_m(t) = \begin{bmatrix} \omega_L(t) \\ \omega_R(t) \end{bmatrix}$$

(4)

External movement of the robot (in global coordinates) is given by

$${\dot{X}}_m(t) = \begin{bmatrix} \cos \phi(t) \\ \sin \phi(t) \end{bmatrix} V(t)$$
$${\dot{Y}}_m(t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \omega(t)$$
$${\dot{\Omega}}_m(t) = \begin{bmatrix} \omega_L(t) \\ \omega_R(t) \end{bmatrix}$$

(5)

Efforts have been directed to find a solution to the problem of movement under nonholonomic constraints using the kinematic model of the mobile robot. In addition, there is a lack of research on the problem of integration of nonholonomic kinematic control and mobile platform dynamics, previously mentioned that the figure (3) is error model and is inherently nonlinear. It is preferable to control, nonlinear systems using a nonlinear controller that takes into account all the characteristics of the original system during control design. The theory is based on the functions of Lyapunov (This is a Lyapunov's second method of stability) are scalar functions that can be used to prove the stability of nonholonomic systems. When nonlinear systems are present, the ordinary differential equations (ODE) are often used. Assuming that there is the same kinematic model for both of the robot movements (actual and reference) and according to equation (5), the error can be expressed as a deviation between the virtual reference robot and the actual robot as shown in Figure 3. The errors obtained are as follows:

$$e_x:$$ which gives the error in command direction.
$$e_y:$$ This gives the error in the vertical direction.
$$e_{\varphi}:$$ that gives the error in the orientation.

The posture error $e(t) = [e_x(t) \ e_y(t) \ e_{\varphi}(t)]^T$ is determined by using the actual posture

$$q(t) = [x(t) \ y(t) \ \varphi(t)]^T$$

of the real robot.

The reference posture

$$q_{ref}(t) = [x_{ref}(t) \ y_{ref}(t) \ \varphi_{ref}(t)]^T$$

of the virtual reference robot:

$$\begin{bmatrix} e_x(t) \\ e_y(t) \\ e_{\varphi}(t) \end{bmatrix} = \begin{bmatrix} \cos \phi(t) & \sin \phi(t) & 0 \\ -\sin \phi(t) & \cos \phi(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_{ref}(t) - q(t) \end{bmatrix}$$

(7)

The constant sampling interval $T_s$ represents the actual movements of the robot (actual and reference) and according to equation (5).
and getting into account the transformation equation (7) can be writing the posture error model can be written as follows:

\[
\begin{bmatrix}
\dot{e}_x \\
\dot{e}_y \\
\dot{e}_\phi
\end{bmatrix} =
\begin{bmatrix}
\cos \phi(t) & 0 & V_{ref} \\
\sin \phi(t) & 0 & 0 \\
0 & 1 & \omega_{ref}
\end{bmatrix}
\begin{bmatrix}
e_y \\
0 \\
-1
\end{bmatrix}
\begin{bmatrix}
V_{ref} \\
\omega_{ref} + \omega_f
\end{bmatrix}
\]  
(8)

where, \( v_f \) and \( \omega_f \) are the reference velocities from the reference trajectory given by \( x_{ref}(t) \) and \( y_{ref}(t) \):

\[
V_{ref}(t) = \sqrt{x_{ref}^2(t) + y_{ref}^2(t)}
\]
\( \omega_{ref}(t) = \frac{x_{ref}(t)y_{ref}(t) + y_{ref}(t)x_{ref}(t)}{x_{ref}^2(t) + y_{ref}^2(t)} \)  
(9)

The input \( u = [V, \omega]^T \) That is driven by the controller. Often control \( u \) is the console in the form as [2].

\[
u = \begin{bmatrix} V \\
\omega \end{bmatrix} = \begin{bmatrix} V_{ref} \cos e_\phi + v_f \\
\omega_{ref} + \omega_f \end{bmatrix}
\]  
(11)

The control velocities \( v_f \) and \( \omega_f \) are the feedback signals, while \( V_{ref} \) and \( \omega_{ref} \) are the forward signals, therefore the results in the tracking-error model:

\[
\begin{align*}
\dot{e}_x &= \omega_{ref} e_y + \omega_f e_x - v_f \\
\dot{e}_y &= -\omega_{ref} e_x - \omega_f e_y + V_{ref} \sin e_\phi \\
\dot{e}_\phi &= -\omega_f
\end{align*}
\]  
(12)

The control objective is to push error in equation (12) towards zero by selecting the appropriate controls \( v_f \) and \( \omega_f \).

3. Extend Error Model of Lyapunov method

Lyapunov stability gives appropriate conditions for stability Convergence of the equilibrium points in a nonlinear dynamic system given by:

\[
\dot{x} = f(x), x \in \mathbb{R}
\]

Assume that the equilibrium results from:

\[
x = f(x)
\]

The positive definite scalar function is based approach

\[
V(x) : \mathbb{R}^n \rightarrow \mathbb{R}, \text{Which meet the following :} \\
(x) = 0 \text{ if } x = 0, \text{ and } V(x) > 0 \text{ if } x \neq 0.
\]

When verifying the derivative of the V function, it can be analyzed and stability of the equilibrium point. It is very important that the derivative is obtained via the system differential equation:

\[
\dot{V} = \frac{\partial V}{\partial x} \dot{x} = \frac{\partial V}{\partial x} f(x)
\]  
(13)

This equation gets three case:

1- \( \dot{V} < 0 \) when the \( V \) is negative semi definite, the equilibrium is locally stable.

2- \( \dot{V} \leq 0 \) except at \( x=0 \) and \( V \) is negative definite, the equilibrium asymptotically stable.

3- \( \lim_{x\rightarrow \infty} V(x) = \infty \), the results are global.

Therefore, the approach depends on the search for functions with the above characteristics referred to as Lyapunov functions. Often the square filter for the Lyapunov function is selected, and if we are able to show its derivatives at least negative or zero, the system is stabilized. The proposed control law has been verified by using the Lyapunov approach, a simple and successful method for stabilizing kinematics. And the control point of the view that we often want to track for any robot pose that is different from the 360-degree reference. Equation (12) does not make this problem easier because the routing error should normally be driven to 0. This can be accomplished by extending the state vector of one element. The variable \( \varphi \) of the original kinematic equation (5) is replaced with two new variables:

\[
\varphi(t) = s(t) + c(t)
\]  
(14)

where:

\[
s(t) = \sin(\varphi(t))
\]
\[
c(t) = \cos(\varphi(t))
\]

Their derivatives are:

\[
\dot{\varphi}(t) = \dot{s}(t) + \dot{c}(t) = \cos(\varphi(t)) \dot{\varphi}(t) - \sin(\varphi(t)) \dot{s}(t)
\]
\[
\dot{s}(t) = \cos(\varphi(t)) \dot{c}(t) - \sin(\varphi(t)) \dot{c}(t)
\]

The new kinematic model is then obtained:

\[
\dot{q} = \begin{bmatrix}
\dot{x}(t) \\
\dot{y}(t) \\
\dot{s}(t) \\
\dot{c}(t)
\end{bmatrix} = \begin{bmatrix}
\cos \varphi(t) & 0 \\
0 & \sin \varphi(t) \\
0 & 0 & \cos \varphi(t) \\
0 & -\sin \varphi(t)
\end{bmatrix}
\begin{bmatrix}
V \\
\omega
\end{bmatrix}
\]  
(16)

The new error states are defined as follows:

\[
\begin{align*}
e_x &= \cos(x_{ref} - x) + \sin(y_{ref} - y) \\
e_y &= -\sin(x_{ref} - x) + \cos(y_{ref} - y) \\
e_s &= \sin(\varphi_{ref} - \varphi) = \cos(\varphi_{ref} - \varphi) = \sin(\varphi_{ref} - s \cos \varphi_{ref}) \\
e_c &= \cos(\varphi_{ref} - \varphi) = \cos \varphi_{ref} + s \sin \varphi_{ref}
\end{align*}
\]  
(17)

differentiation of Equation (18) and some manipulations, the following system is obtained:

\[
\begin{align*}
\dot{e}_x &= V \cos - V + e_y \omega \\
\dot{e}_y &= V_{ref} e_s + e_x \omega \\
\dot{e}_s &= \omega_{ref} e_{cos} - e_{cos} \omega \\
\dot{e}_c &= -\omega_{ref} e_s + e_s \omega
\end{align*}
\]  
(18)
while $V = V_{ref} e_{cos} + V_{fb}$ and $\omega = \omega_{ref} + \omega_{fb}$

Same as the Equation (11) it will be used in the control law to be derive a $e_x, e_y, e_s$ to 0. The $e_x$ variable is obtained when the cosine direction is a fault of 1. For this reason, a new error will be defined as $e_c = e_{cos} - 1$ and the final error model of the system is now:

\[
\begin{align*}
\dot{e}_x &= \omega_{ref} e_y - v_{fb} + e_y \omega_{fb} \\
\dot{e}_y &= -\omega_{ref} e_x + v_{ref} e_s - e_x \omega_{fb} \\
\dot{e}_s &= -e_c \omega_{fb} - \omega_{fb} \\
\dot{e}_c &= e_s \omega_{fb} \\
\end{align*}
\]

(19)

The error model in equation (19) achieves relative stability, and based on the Lyapunov approach will be developed. The very clear idea is to use Lyapunov’s type function as flows:

\[
\begin{align*}
V_0 &= \frac{1}{2}(e_x^2 + e_y^2) + \frac{1}{2} e_s^2 + \frac{k}{2}(e_x^2 + e_y^2) + \frac{1}{2}(e_s^2 + e_c^2) \\
\end{align*}
\]

(20)

An exponentially stable controller will be developed based on Lyapunov approach. The states $e_x, e_y, e_s$ should be driven towards 0, while $e_c$ should converge to 1 if we want to achieve perfect tracking. The function $V_0$ can be simplified by using the following:

\[
\begin{align*}
e_s^2 + e_c^2 &= e_{cos}^2 = (e_{cos} - 1)^2 = 2 - 2e_{cos} = -2e_c
\end{align*}
\]

(21)

Taking into account the equations of the error model (19) and (21), the derivative of $V_0$ in (20) is:

\[
\begin{align*}
V_0 &= k_y e_x e_y + k_y e_y e_x - e_c \\
\dot{V}_0 &= k_y e_x(\omega_{ref} e_y - v_{fb} + e_y \omega_{fb}) + k_y e_y(\omega_{ref} e_x - \omega_{fb} e_s + v_{ref} e_s) \\
\end{align*}
\]

(22)

sub $V_{fb} = V - V_{ref} e_{cos}$ and $\omega_{fb} = \omega - \omega_{ref}$ in equation (22) get:

\[
\begin{align*}
\dot{V}_0 &= -k_y v_{fb} e_x + k_y v_{ref} e_{cos} e_x + k_y v_{ref} e_y e_s - \omega e_s + \omega_{ref} e_x \\
V_0 &= k_y e_x(\omega_{ref} e_x - V) + e_s(k_y e_y v_{ref} - \omega + \omega_{ref}) \\
\end{align*}
\]

(23)

Since the values in the parentheses in Equation (23) must be chosen to make Lyapunov derivative of a negative semidefinite function, the control law that achieves it is:

\[
\begin{align*}
V &= V_{ref} e_x + k_x e_x \\
\omega &= \omega_{ref} + k_y V_{ref} e_y + k_x e_s \\
\end{align*}
\]

(24)

when adding (V and $\omega$) in Equation (24) get:

\[
\begin{align*}
V_0 &= k_y e_x(\omega_{ref} e_x - V e_x - k_x e_x) + e_x(k_y e_y V_{ref} - k_x e_y + e_ref) \\
\dot{V}_0 &= -k_y k_x e^2_s - k_x e^2_s \leq 0 \\
\end{align*}
\]

(25)

The design parameters $k_x, k_y$ and $k_s$ are proposed with value $> 0$.

By The comparing between the actual and the desired characteristic polynomial equations can be determined the controller gains $(k_x, k_y, k_s)$. Take the desired characteristic polynomial for the following [5]:

\[
(Z + z_1)(Z + z_2)(Z + z_3)
\]

(26)

where:

\[
\begin{align*}
z_1 &= -e^{-2\xi \omega_{ref} T_s} \\
z_2, z_3 &= -e^{-2\xi \omega_{ref} T_s \pm \sqrt{1 - 4\xi^2}} \\
\end{align*}
\]

(27)

The desired damping coefficient $\zeta \in (0, 1)$ and the characteristic frequency $\omega_{ref} > |\omega_{ref, max}|$ are selected, where $|\omega_{ref, max}|$ is the maximum allowed mobile robot angular velocity.

The linearization of the equation (13) while adding $V = V - V_{ref} e_{cos}$ and $\omega_{fb} = \omega - \omega_{ref}$ get:

\[
\begin{align*}
\begin{bmatrix} e_x \\ e_y \\ e_s \end{bmatrix} &= \begin{bmatrix} -k_x e_x + \omega_{ref} e_y \\ -\omega_{ref} e_x + v_{ref} e_s \end{bmatrix} - \begin{bmatrix} k_x e_y \\ -k_y V_{ref} e_y - k_x e_s \end{bmatrix} \\
\end{align*}
\]

(28)

By using equation (28) it can be determined the closed loop characteristics of the system:

\[
\det(sI - A) = 0 \\
\det(sI - A) = \det \begin{bmatrix} S + k_x & -\omega_{ref} & 0 \\ \omega_{ref} & S & -V_{ref} \\ 0 & V_{ref} k_y & S + k_s \end{bmatrix} = 0 \\
S^3 + (k_x + k_s)S^2 + (k_y k_x + V_{ref} k_y + \omega_{ref}^2)S + V_{ref} k_y k_x + \omega_{ref}^2 k_s = 0 \\
\]

(29)

The continuous-time characteristic equation $S$, convert to the discrete-time characteristic equation $Z$, by using the forward difference method as follows:

\[
\begin{align*}
\frac{c}{c_{s(z)}} &= \frac{c}{c_{s(z)}} \bigg|_{s = \frac{z - 1}{T}} \\
\frac{c}{c_{z(\omega)}} &= Z^2 + (T_s k_x + T_s k_x - 3)Z^2 + (T^2_s k_x + T^2_s V_{ref} k_y + T^2_s \omega_{ref}^2 - 2T_s k_x + 2T_s k_x - 3)Z + (T_s k_x - T_s k_x - T^2_s k_x - T^2_s \omega_{ref} + T^2_s V_{ref} k_y - T^2_s V_{ref} k_y - 1 \\
\end{align*}
\]

(30)

Comparing coefficients at the same power of $z$ in Equation (26) with Equation (30) results in

\[
\begin{align*}
T_s k_x + T_s k_x - 3 &= z_1 + z_2 + z_3 \\
T^2_s k_x + T^2_s V_{ref} k_y + T^2_s \omega_{ref}^2 - 2T_s k_x + 2T_s k_x - 3 &= z_2 z_3 + z_2 z_3 + z_2 z_3 \\
T_s k_x - T_s k_x - T^2_s k_x - T^2_s \omega_{ref} + T^2_s V_{ref} k_y + T^2_s \omega_{ref}^2 - 2T_s k_x + 2T_s k_x - 3 &= z_2 z_3 + z_2 z_3 + z_2 z_3 \\
T^2_s \omega_{ref}^2 + T^2_s V_{ref} k_y + T^2_s \omega_{ref}^2 - 2T_s k_x + 2T_s k_x - 3 &= z_2 z_3 + z_2 z_3 + z_2 z_3 \\
\end{align*}
\]

(31) (32) (33)
Let \( k_x = k_y \) to find the solution of Equation (30), as follows:

\[
k_x = k_y = \frac{z_1 + z_2 + z_3 + 3}{2T_s}
\]

(34)

and \( k_y \) is determined from Equation (32) or Equation (33) as follows:

\[
k_y = \frac{z_1 z_2 z_3 + z_1 z_3 + z_2 z_3 - T_s^2 \omega_{ref}^2 + 2(z_1 + z_2 + z_3 + 3)}{4}
\]

(35)

When \( V_{ref} \) is close to zero or \( T_s \) is very small sampling time, \( k_y \) goes to infinity and we will lose the stability of the mobile robot system, therefore, \( V_{ref} > 0 \), as we have proved in the Lyapunov function.

4. Simulation Results

To demonstrate the effectiveness of the proposed control strategy will be used mobile robot that appears in Figure 4 where Table 1 shows the dynamic parameters of this mobile robot.

**Table 1** General specifications of the mobile robot

<table>
<thead>
<tr>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>0.375×0.313×0.267(L×W×H) m</td>
</tr>
<tr>
<td>Weight</td>
<td>4.2 Kg</td>
</tr>
<tr>
<td>Inertia</td>
<td>0.15 Kg.m²</td>
</tr>
<tr>
<td>Wheels Distance</td>
<td>0.28 m</td>
</tr>
<tr>
<td>Radius of Wheels</td>
<td>0.143 m</td>
</tr>
<tr>
<td>Linear Speed</td>
<td>0.8 m/s</td>
</tr>
</tbody>
</table>

where the linear velocity and the reference angular velocity are chosen as \( V_{ref} = 0.8 \) m/s and \( \omega_{ref} = 5.59 \) rad/s, respectively.

All orders and programs are produced by microprocessor through wired and wireless connections with other devices by using Python software programs language. After determining the desired path and then the processor is programmed to get the commands to the DC motors, the encoder gets the update of platform mobile robot’s location, after that the comparison and extraction of the error occurs to use in the control equations.

The reference initial position of the mobile platform is \( q_{ref} = [0 \ 0 \ 56]^T \), same as the actual initial position is \( q_0 = [0 \ 0 \ 0]^T \) and using circle path shown in Figure (6).

4. Simulation Results

To demonstrate the effectiveness of the proposed control strategy will be used mobile robot that appears in Figure 4 where Table 1 shows the dynamic parameters of this mobile robot.

**Table 1** General specifications of the mobile robot

<table>
<thead>
<tr>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>0.375×0.313×0.267(L×W×H) m</td>
</tr>
<tr>
<td>Weight</td>
<td>4.2 Kg</td>
</tr>
<tr>
<td>Inertia</td>
<td>0.15 Kg.m²</td>
</tr>
<tr>
<td>Wheels Distance</td>
<td>0.28 m</td>
</tr>
<tr>
<td>Radius of Wheels</td>
<td>0.143 m</td>
</tr>
<tr>
<td>Linear Speed</td>
<td>0.8 m/s</td>
</tr>
</tbody>
</table>

where the linear velocity and the reference angular velocity are chosen as \( V_{ref} = 0.8 \) m/s and \( \omega_{ref} = 5.59 \) rad/s, respectively.

All orders and programs are produced by microprocessor through wired and wireless connections with other devices by using Python software programs language. After determining the desired path and then the processor is programmed to get the commands to the DC motors, the encoder gets the update of platform mobile robot’s location, after that the comparison and extraction of the error occurs to use in the control equations.

The reference initial position of the mobile platform is \( q_{ref} = [0 \ 0 \ 56]^T \), same as the actual initial position is \( q_0 = [0 \ 0 \ 0]^T \) and using circle path shown in Figure (6).

Regarding the simulation results, a good tracking error and tracking trajectory in X position are shown in Figure (7), which shows the related tracking errors...
which be reduce after 26 second shows a good tracking of the real tracking with desired after 26 second along circular path with showing in figure (8).

For the $\phi$- direction, the tracking of the desired trajectory is presented in Figure (11) It is clear from Figure (12) that the related tracking converges to zero, but in the path turn zones the error ratio increases to 2 cm. This is logical for the effect of friction on the path turn. Again, there is a good tracking.

In the Y-position, Figure (9) shows a good tracking error in Y position because the tracking errors which be reduce after 26 second. Also show a good tracking trajectory of real and desired path after 26 second in figure (10).

Finally, the simulation results show that the errors are very small and converge to zero otherwise the arrive at the state of stability in a very short time, indicating effective control in the performance of the nonholonomic wheeled mobile robot.

Conclusions

This paper presents a control strategy based on the approach of Lyapunov’s for nonholonomic wheeled mobile robots. First, the kinematic controller is designed to generate the required speed for the left and right wheels. Based on the dynamic control of the mobile robot platform, the Dynamic Controller design has been developed to ensure good tracking of the desired path. After obtaining the simulation results showed the effectiveness of the proposed control method in mobile robots. In practice, the proposed control strategy has been validated with a mobile robot produced by the NUXES Robot Company, type 2WD mobile Arduino robotics car 10004.

As a future work, using the accessories of additional devices and systems with mobile robots such as image processing, GPS, laser techniques to take advantage of these devices to obtain an additional source to obtain the real coordinates and extract the error rate and then apply the control strategy proposed on these values.
References

Kanayama, Yutaka, (1990), A Stable Tracking Control Method for an Autonomous Mobile Robot, IEEE
Spyros G. Tzafestas, (2017), Introduction to Mobile Robot Control, Elsevier USA.
Emine Canigur, Metin Ozkan, (2012), Model Reference Adaptive Control of a Nonholonomic Wheeled Mobile Robot for Trajectory Tracking, 978-1-4673-1448-2012 IEEE.
Jun Ye, (2008), Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique, Elsevier.