Integral Backstepping Approach for Mobile Robot Control

Bouzgou kamel¹,², Ibari benaoumeur¹,³, Benchikh laredj ², and Ahmed-foitih Zoubir¹

¹Laboratoire d’Électronique de Puissance, d’Énergie Solaire et d’Automatique. Université des sciences et
de la technologie d’Oran Mohamed Boudiaf.El Mnaouar, BP 1505, Bir El Djir 31000 Oran , Algérie.
²Laboratoire Informatique, Biologie Intégrative et Systèmes Complexes (IBISC). Université
d’Évry-Val-d’Essonne (UEVE), 40 rue d Pelvoux CE 1433 Courcouronnes. 91020. Evry Cedex, France.
³Département d’Électrotechnique, Université de mustapha stambouli, Mascara, Algérie

¹corresponding author, e-mail: bouzgou_kamel@hotmail.fr

Abstract

This paper presents the trajectory tracking problem of a unicycle-type mobile robots. A robust
output tracking controller for nonlinear systems in the presence of disturbances is proposed, the approach
is based on the combination of integral action and Backstepping technique to compensate for the dynamic
disturbances. For desired trajectory, the values of the linear and angular velocities of the robot are assured
by the kinematic controller. The control law guarantees stability of the robot by using the lyapunov theorem.
The simulation and experimental results are presented to verify the designed trajectory tracking control.

Keywords: Robot Mobile, Backstepping, Trajectory tracking, nonlinear systems.

1. Introduction

In the recent years, the mobile robots have been used to performing tasks that are too
dangerous or tedious for humans. They can also be found in: industry, science, education, med-
ical, domestic machines, entertainment and military applications [1]. They are used in a range of
applications [2, 3].

Many research articles have been proposed for the trajectory tracking problem in the
literature. Studies and approaches have been developed in this field [4, 5]. Most controllers
designed in control systems are based on kinematic models of the mobile robots and control
theory to achieve and accomplish the designing control laws.

For the path-following controller, in [6] the control algorithm of nonlinear PID-based adap-
tive control for the wheeled mobile robot (WMR) is developed. In [7], a controller based on distance
measures using the sonar sensors is proposed to follow a wall, and to guide a mobile robot
along a corridor while avoiding obstacles this algorithm is tested in the real implementation cases.

In the last-cited paper, the designed controllers do not include the dynamic model of mo-
bile robots. However, it is necessary to using the tools from control theory and dynamic systems
in order to perform at high speeds and heavy work. Thus, some controllers based on dynamic
modeling have been proposed. As an example in [8], A fuzzy Logic Controller is proposed to
track a desired trajectory of a unicycle mobile robot using Mamdani model and backstepping
technique, the dynamics parameters for tracking control of mobile robot are included using the
Euler-Lagrange method (masses, inertias, damping, etc.). Moreover, no experimental results
were reported.

In [9], an adaptive fuzzy logic based controller is designed to control mobile robot with
unknown dynamics parameters for tracking control of mobile robots, the proposed algorithm is
implemented on mobile robot system with PIC 16F877 microcontroller, but the real implementation
required high-performance computer architecture based on a multiprocessor system. An adaptive
trajectory-tracking controller based on the model dynamic of a unicycle-like mobile is designed in
[10], a stability analysis using Lyapunov theory is analyzed to ensure the robot stability, the
parameters are updated on-line where the parameters of the system are not known or change

Copyright © 2017 Universitas Ahmad Dahlan. All rights reserved.

Received January 27, 2017; Revised July 2, 2017; Accepted July 30, 2017
the task, this method of trajectory tracking control is tested on a Pioneer 3-DX mobile robot. Fuzzy adaptive trajectory tracking control of nonholonomic wheeled mobile robot is proposed in [11], to compensate for the kinematic and dynamic disturbances relying on the state variables, the proposed controller is tested only with simulation results and no experimental results were presented.

The robust trajectory tracking of mobile robots is proposed in [12] to compensate the unknown nonlinear dynamics of the robot. The control structure is based on the combination of the neural adaptation technique with sliding mode control with the use of a real implementation. In [13], the Backstepping approach is introduced to solve the tracking control problem, the commands of linear and angular velocities delivered to the robot servos are studied and tested with real implementation by giving the robot forward and angular reference velocities, collecting the actual velocities, the stability analysis of the system is proved via Lyapunov theory.

Some of the parameters of the systems are not modeled or are prone to change in time, with is often impossible to control. In this paper, a robust control of PI/Backstepping is proposed to compensate for the dynamic disturbances, and its stability is analyzed using the Lyapunov theory, the approach is based on the combination of nonlinear PI controller and a Backstepping technique. Integral action control is added to reduce the error of the trajectory tracking. A comparative study between the proposed approach and the backstepping control is presented by simulations in terms of accuracy and stability under different load conditions.

The rest of the paper is organized as follows. Section 2 gives the dynamic unicycle-like robot model. Section 3, presents and explains the proposed PI/backstepping controller design. Respectively, in section 4, we presented simulations and experimental results to verify the designed trajectory tracking control method. Finally, in section 5 we present the conclusion of this paper.

2. Robot model

In this section, a unicycle-like mobile robot proposed by De La Cruz and Carelli in [14] is considered. The mobile robot is illustrated in Figure 1, where $G$ is the center of mass, $h$ is the point of required to track a trajectory, $u$ and $\omega$ are, respectively, the linear and angular velocities, $\psi$ is the robot orientation, $a$, $b$, $c$ and $d$ are distances.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{The unicycle-like mobile robot.}
\end{figure}

The force and moment equations for the robot are written as [15]:

\begin{equation}
\begin{aligned}
\sum F_x' &= m(\ddot{u} - \ddot{u}\omega) = F_{rxx'} + F_{rrxx'} + F_{cx'} \\
\sum F_y' &= m(\ddot{u} + \ddot{u}\omega) = F_{rxy'} + F_{rrxy'} + F_{cy'} \\
\sum M_z &= I_z\dot{\omega}
\end{aligned}
\end{equation}
Where $F_{rx}$ and $F_{ry}$ are the longitudinal and lateral tire forces of the right wheel, $F_{lx}'$ and $F_{ly}'$ are the longitudinal lateral tire forces of the left wheel, $F_{cx}$ and $F_{cy}$ are the longitudinal and lateral force exerted on $C$ by the castor, $I_z$ is the robot moment of inertia at $G$. The kinematics of point $h$ is written in [14] as:

$$\begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\psi} \\
\dot{u} \\
\dot{\omega}
\end{pmatrix} =
\begin{pmatrix}
ucos(\psi) - usin(\psi) \\
usin(\psi) + ucos(\psi) \\
\frac{k_3}{k_2} \omega^2 - \frac{k_4}{k_2} u \\
-\frac{k_3}{k_2} \omega u - \frac{k_4}{k_2} \omega
\end{pmatrix} +
\begin{pmatrix}
0 & 0 \\
0 & 0 \\
\frac{1}{k_3} & 0 \\
0 & \frac{1}{k_2}
\end{pmatrix}
\begin{pmatrix}
u_{ref} \\
\omega_{ref}
\end{pmatrix}
$$

(2)

Where $u$ and $\bar{u}$ are the longitudinal and lateral velocities of the center of mass. The complete model of the unicycle-like mobile is given in [14] by De La Cruz and Carelli as:

$$\begin{pmatrix}
\dot{\chi} \\
\dot{\psi} \\
\dot{\omega}
\end{pmatrix} =
\begin{pmatrix}
\frac{a}{k_a} \cos(\psi) - awsin(\psi) \\
\frac{a}{k_a} \sin(\psi) + awcos(\psi) \\
\lambda_1 \omega \\
\lambda_2 \omega \\
-\lambda_3 \omega \\
\lambda_4 \omega
\end{pmatrix} +
\begin{pmatrix}
0 & 0 \\
0 & 0 \\
\frac{1}{k_3} & 0 \\
0 & \frac{1}{k_2}
\end{pmatrix}
\begin{pmatrix}
u_{ref} \\
\omega_{ref}
\end{pmatrix}
$$

(3)

Where the parameters of the dynamic model are defined in [14] as:

$$\lambda_1 = \frac{\frac{a}{k_a} \left(mR_t r + 2I_e + 2k_{DT}\right)}{(2k_{DT})}$$

$$\lambda_2 = \frac{\frac{a}{k_a} \left(I_c r^2 + 2R_t r (I_z + m b^2) + 2k_{DR} R_a \right)}{(2k_{DR})}$$

$$\lambda_3 = \frac{R_a}{k_a} \frac{w b h_r}{2k_{PT}}$$

$$\lambda_4 = \frac{\frac{a}{k_a} \left(\frac{1}{k_{PT}} \right) + B_e}{(r k_{PT})} + 1$$

$$\lambda_5 = \frac{\frac{a}{k_a} \left(\frac{1}{k_{PR}} \right) + B_e}{(r k_{PR})} + 1$$

$$\lambda_6 = \frac{\frac{a}{k_a} \left(\frac{1}{k_{PT}} \right) + B_e}{(r k_{PT})} + 1$$

where $m$ is the robot mass, $I_e$ and $B_e$ are the moment of inertia and the viscous friction coefficient, $r$ is the right and left wheel radius, and $R_t$ is the nominal radius of the tire, $k_a$ is the constant of torque, $k_b$ is electromotive constant of motors, $R_a$ is the electrical resistance of the motors. PD controllers are implemented with proportional gains $k_{PT}$ and $k_{PR}$, and derivative gains $k_{DT}$ and $k_{DR}$ to control the velocities of the right and left motor.

### 3. Robust Controller design

In this work, two controllers are used: the first one is a kinematics controller for external loop and the second one is a PI/backstepping for an internal loop as see in Figure2.

![Figure 2. Structure control of PI/backstepping.](image-url)
3.1. Kinematic controller

The kinematic controller is developed with the inverse kinematics equations of the robot mobile and it is designed in order to generate the desired values of the linear and angular velocities. From (3), the kinematic model can be written as:

\[
\begin{pmatrix}
  \dot{x} \\
  \dot{y}
\end{pmatrix} =
\begin{pmatrix}
  \cos(\psi) & -\sin(\psi) \\
  \sin(\psi) & \cos(\psi)
\end{pmatrix}
\begin{pmatrix}
  u_{\text{ref}} \\
  \omega_{\text{ref}}
\end{pmatrix}
\]

(4)

Where \( u_{\text{ref}} \) and \( \omega_{\text{ref}} \) are the desired values of the linear and angular velocities given by the kinematic controller.

The matrix inverse is

\[
A^{-1} =
\begin{pmatrix}
  \cos(\psi) & \sin(\psi) \\
  -\frac{1}{a}\sin(\psi) & \frac{1}{a}\cos(\psi)
\end{pmatrix}
\]

(5)

The control law can be chosen as:

\[
\begin{pmatrix}
  u_{\text{ref}} \\
  \omega_{\text{ref}}
\end{pmatrix} =
\begin{pmatrix}
  \cos \psi & \sin \psi \\
  -\frac{1}{a}\sin \psi & \frac{1}{a}\cos \psi
\end{pmatrix}
\begin{pmatrix}
  \dot{x} + \varepsilon_x \\
  \dot{y} + \varepsilon_y
\end{pmatrix}
\]

(6)

Where \( \varepsilon_x = x_d - x \) and \( \varepsilon_y = y_d - y \) the current are position errors, \( h(x, y) \) and \( h_d(x_d, y_d) \) are the current and the desired coordinates.

The development and stability analysis of the kinematic controller is detailed in [13].

3.2. Nonlinear PI-Based Backstepping controller design

The combination of integral action and backstepping technique is used to design a control law that improves the robustness when the dynamics parameters of the robot are not well-known. The dynamic part of equation (3) is:

\[
\begin{cases}
  \dot{u} = \frac{1}{\lambda_1} \omega^2 - \frac{1}{\lambda_2} \dot{u} + \frac{u_{\text{ref}}}{\lambda_1} \\
  \dot{\omega} = \frac{1}{\lambda_1} u \omega - \frac{1}{\lambda_2} \omega^2 + \frac{\omega_{\text{ref}}}{\lambda_1}
\end{cases}
\]

(7)

For the first step, we consider the output errors between the reference and actual controls are given by:

\[
\begin{align*}
  \varepsilon_{1u} &= u_{\text{ref}} - u \Rightarrow \dot{\varepsilon}_{1u} = \dot{u}_{\text{ref}} - \dot{u} \\
  \varepsilon_{1\omega} &= \omega_{\text{ref}} - \omega \Rightarrow \dot{\varepsilon}_{1\omega} = \dot{\omega}_{\text{ref}} - \dot{\omega}
\end{align*}
\]

(8)

And the following Lyapunov function defined as

\[
V(\varepsilon_{1u}) = \frac{1}{2} \varepsilon_{1u}^2, \quad V(\varepsilon_{1\omega}) = \frac{1}{2} \varepsilon_{1\omega}^2
\]

(9)

The time derivative of the Lyapunov candidate functions is calculated as

\[
\dot{V}(\varepsilon_{1u}) = \varepsilon_{1u} \dot{\varepsilon}_{1u}, \quad \dot{V}(\varepsilon_{1\omega}) = \varepsilon_{1\omega} \dot{\varepsilon}_{1\omega}
\]

(10)

The stabilization of the dynamics errors system can be obtained by introducing a virtual controls input:

\[
\begin{cases}
  u_{\text{w}u} = \dot{u}_{\text{ref}} + K_{1u} \varepsilon_{1u} + \alpha_u \chi_u \\
  u_{\text{w}\omega} = \dot{\omega}_{\text{ref}} + K_{1\omega} \varepsilon_{1\omega} + \alpha_\omega \chi_\omega
\end{cases}
\]

(11)

Where the integral actions are:

\[
\begin{cases}
  \chi_u = \int \varepsilon_{1u}(\tau) d\tau \\
  \chi_\omega = \int \varepsilon_{1\omega}(\tau) d\tau
\end{cases}
\]

(12)

With \( K_{1u}, K_{1\omega}, \alpha_u \) and \( \alpha_\omega \) are design parameters.

At the second step the new errors given by:

\[
\begin{align*}
  \varepsilon_{2u} &= \dot{u}_{\text{ref}} - \dot{u} + K_{1u} \varepsilon_{1u} + \alpha_u \chi_u \\
  \varepsilon_{2\omega} &= \dot{\omega}_{\text{ref}} - \dot{\omega} + K_{1\omega} \varepsilon_{1\omega} + \alpha_\omega \chi_\omega
\end{align*}
\]

(13)
And

\[ \begin{align*}
\dot{\varepsilon}_{2u} &= \ddot{u}_{ref} - \ddot{u} + K_{1u} \varepsilon_{1u} + \alpha_u \varepsilon_{1u} \\
\dot{\varepsilon}_{2w} &= \ddot{\omega}_{ref} - \ddot{\omega} + K_{1w} \varepsilon_{1w} + \alpha_w \varepsilon_{1w}
\end{align*} \]  

(14)

From (11) and (14), it follows that:

\[ \begin{align*}
\dot{\varepsilon}_{1u} &= - K_{1u} \varepsilon_{1u} - \alpha_u \chi_u + \varepsilon_{2u} \\
\dot{\varepsilon}_{1w} &= - K_{1w} \varepsilon_{1w} - \alpha_w \chi_w + \varepsilon_{2w}
\end{align*} \]  

(15)

And

\[ \begin{align*}
\dot{\varepsilon}_{2u} &= - K_{2u} \varepsilon_{2u} - \varepsilon_{1u} \\
\dot{\varepsilon}_{2w} &= - K_{2w} \varepsilon_{2w} - \varepsilon_{1w}
\end{align*} \]  

(16)

Where \( K_{2u}, K_{2w} \) are positive constant of stability. So that is result, the integral backstepping control laws of elevation, linear and angular velocities are:

\[ u_u = \begin{bmatrix} \lambda_1 (\ddot{u}_{ref} + (K_{1u} + K_{2u}) \varepsilon_{2u}) \\ + (1 - \alpha_u \varepsilon_{1u} - \alpha_w \chi_w) \\ + 2 \lambda_3 \omega - \lambda_4 \ddot{u} \end{bmatrix} \]  

(17)

And

\[ u_w = \begin{bmatrix} \lambda_2 (\ddot{\omega}_{ref} + (K_{1w} + K_{2w}) \varepsilon_{2w}) \\ + (1 - \alpha_u \varepsilon_{1u} - \alpha_w \chi_w) \\ - \lambda_5 (\ddot{\omega} + \omega) - \lambda_6 \omega \end{bmatrix} \]  

(18)

The stability of the equilibrium at the origin of the error system can be obtained by Lyapunov theory. Thus, choosing the Lyapunov candidate functions as follows:

\[ \begin{align*}
V(\varepsilon_{1u}, \varepsilon_{2u}) &= \frac{\alpha_u \chi_u^2 + \varepsilon_{1u}^2 + \varepsilon_{2u}^2}{2} \\
V(\varepsilon_{1w}, \varepsilon_{2w}) &= \frac{\alpha_w \chi_w^2 + \varepsilon_{1w}^2 + \varepsilon_{2w}^2}{2}
\end{align*} \]  

(19)

Whose first time derivative

\[ \begin{align*}
\dot{V}(\varepsilon_{1u}, \varepsilon_{2u}) &= - K_{1u} \varepsilon_{1u}^2 - K_{2u} \varepsilon_{2u}^2 \leq 0 \\
\dot{V}(\varepsilon_{1w}, \varepsilon_{2w}) &= - K_{1w} \varepsilon_{1w}^2 - K_{2w} \varepsilon_{2w}^2 \leq 0
\end{align*} \]  

(20)

are negative defined, which means that, the tracking error is asymptotically stable.

4. Simulation and experiment results

In this section, the proposed control law of the robot was implemented on a Arduino Robot Mobile (Radius of 185 mm, height of 85 mm and weight of 1.50 kg), which has two processors based on the ATmega32u4, see Figure 3. The wheels are driven by DC motors having rated torque 30 mNm at 15000 rpm, each motor is equipped with an encoder output of 500 ticks/revolution, the sample time of the robot is 0.1 s. Arduino Yun with highly integrated and cost effective IEEE 802.11n 1x1 2.4 GHz System is mounted on the robot to transmit data from matlab to the robot using the Wireless network protocol. The odometric sensors are used to sensing the point h of robot position. The controller is implemented under the Windows operating system using Matlab. In order to test the proposed PI/Backstepping control law of this paper, several experiments were carried out with circular track reference trajectory of 2 m.

The initial position for the mobile robot is as \( P_0 = (x, y, \psi) = (0, 0, 0) \). The gains of the implemented controller are selected as:

\[ \begin{align*}
K_{1u} &= 120, K_{2u} = 75 \\
K_{1w} &= 14, K_{2w} = 2 \\
\alpha_u &= 12.75, \alpha_w = 3.5
\end{align*} \]
The robot mobile tracks the circular trajectory with a good performance (smaller tracking error) as see in Figure 4a, the evolutions of the desired and actual linear and angular velocities are plotted in Figure 4b, from these figures, it is clear that the tracking of the reference trajectory is very accurate and the tracking errors tend to zero.

Figure 5a illustrates the control the robot’s wheels speed signal generated by the Motor board processor. The control law of the linear and angular velocities are presented in Figure 5b, these velocities are sent to the robot through wireless communication using the Yun Arduino card.

4.1. Comparative study
The second experiment is designed to compare between the Backstepping approach proposed in [13] and PI/backstepping, we carried out the experiment with same circular track reference trajectory of 2 meters of radius and same initial posture $P_0=(x, y, \theta) = (0, 0, 0)$.

Figure 6 shows the evolution distance errors for experiments using the proposed approach and backstepping controller, it can be see that the PI/backstepping is the most stable and accurate method comparing the backstepping approach.

Now in table 1, we have the numerical results: the mean error, variance and standard deviation of trajectory using backstepping and PI/backstepping approaches, we notice that the PI/backstepping presents the best performance in term of accuracy, the numerical results proved
the effectiveness of this algorithm. One of the advantages of this approach is that they can usually compensate for the dynamics not modeled as hysteretic damping, wheel tire diameters and vibrations.

5. Conclusion

This paper addressed the problem of tracking control of unicycle-like robot. The proposed approach is based on the backstepping technique combined with integral action; the stability of the system is analyzed using the Lyapunov theory, and the design controller is implemented on an Arduino mobile robot. The integral part of algorithm is added to eliminate the tracking errors and
the disturbances and dynamics not modeled as wheel and tire diameters, mass, inertia, etc, the comparative results demonstrate the effectiveness the proposed controller, in term of accuracy in term of accuracy, stability and convergence.

The force estimator design can be added to guarantee the robustness of the controller. This work can be applied to remote control using the virtual reality [16], and augmented reality [17]. We intend also to improve the system in an environment with obstacles.

References
[1] T. Lozano-Perez, I. J. Cox, and G. T. Wilfong, Autonomous robot vehicles. Springer Science & Business Media, 2012.
[2] K. Karam and A. Albagul, “Dynamic modelling and control issues for a three wheeled vehicle,” Proc. 5th Int. Con. on Control, Automation, Robotics and Vision, Singapore, 1998.
[3] D. W. Gage, “Ugv history 101: A brief history of unmanned ground vehicle (ugv) development efforts,” DTIC Document, Tech. Rep., 1995.
[4] P. Antonini, G. Ippoliti, and S. Longhi, “Learning control of mobile robots using a multiprocessor system,” Control Engineering Practice, vol. 14, no. 11, pp. 1279–1295, 2006.
[5] S. Adinandra and D. A. Ratnawari, “A practical coordinated trajectory tracking for a group of mixed wheeled mobile robots with communication delays,” TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 14, no. 4, 2016.
[6] T.-S. Jin and H.-H. Tack, “Path following control of mobile robot using lyapunov techniques and pid controller,” International Journal of Fuzzy Logic and Intelligent Systems, vol. 11, no. 1, pp. 49–53, 2011.
[7] R. Carelli and E. O. Freire, “Corridor navigation and wall-following stable control for sonar-based mobile robots,” Robotics and Autonomous Systems, vol. 45, no. 3, pp. 235–247, 2003.
[8] O. Castillo, L. T. Aguilar, and S. Cárdenas, “Fuzzy logic tracking control for unicycle mobile robots.” Engineering Letters, vol. 13, no. 2, pp. 73–77, 2006.
[9] T. Das and I. N. Kar, “Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots,” IEEE Transactions on Control Systems Technology, vol. 14, no. 3, pp. 501–510, 2006.
[10] F. N. Martins, W. C. Celeste, R. Carelli, M. Sarcinelli-Filho, and T. F. Bastos-Filho, “An adaptive dynamic controller for autonomous mobile robot trajectory tracking,” Control Engineering Practice, vol. 16, no. 11, pp. 1354–1363, 2008.
[11] D. Chwa, “Fuzzy adaptive tracking control of wheeled mobile robots with state-dependent kinematic and dynamic disturbances,” IEEE transactions on Fuzzy Systems, vol. 20, no. 3, pp. 587–593, 2012.
[12] F. G. Rossomando, C. Soria, and R. Carelli, “Sliding mode neuro adaptive control in trajectory tracking for mobile robots,” Journal of Intelligent & Robotic Systems, vol. 74, no. 3-4, pp. 931–944, 2014.
[13] I. Benaoumeur, B. Larejd, H. E. A. Reda, and A.-f. Zoubir, “Backstepping approach for autonomous mobile robot trajectory tracking,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 2, no. 3, 2016.
[14] C. De La Cruz and R. Carelli, “Dynamic modeling and centralized formation control of mobile robots,” in IECON 2006-32nd Annual Conference on IEEE Industrial Electronics. IEEE, 2006, pp. 3880–3885.
[15] F. D. Boyden and S. A. Velinsky, “Dynamic modeling of wheeled mobile robots for high load applications,” in Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on. IEEE, 1994, pp. 3071–3078.
[16] I. Benaoumeur, A.-f. Zoubir, and H. E. A. Reda, “Remote control of mobile robot using the virtual reality,” International Journal of Electrical and Computer Engineering (IJECE), vol. 5, no. 5, pp. 1062–1074, 2015.
[17] B. Ibari, K. Bouzgou, Z. Ahmed-Foith, and L. Benchikh, “An application of augmented reality (ar) in the manipulation of fanuc 200ic robot,” in Innovative Computing Technology (INTECH), 2015 Fifth International Conference on. IEEE, 2015, pp. 56–60.