

Received August 17, 2020, accepted August 26, 2020, date of publication September 1, 2020, date of current version September 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021057

Intuitive Planning, Generation, and Tracking of Trajectory for WMR With Mobile Computing Device and Embedded System

CELSO MÁRQUEZ-SÁNCHEZ¹, **VICTOR MANUEL HERNÁNDEZ-GUZMÁN**², (Member, IEEE),
RAMÓN SILVA-ORTIGOZA³, (Member, IEEE), **MAYRA ANTONIO-CRUZ**⁴,
AND CARLOS ALEJANDRO MERLO-ZAPATA⁵

¹Universidad Politécnica de Texcoco, Estado de México 56250, México

²Facultad de Ingeniería, Universidad Autónoma de Querétaro, Querétaro 76010, México

³Laboratorio de Mecatrónica and Energía Renovable, CIDETEC, Instituto Politécnico Nacional, Ciudad de México 07700, México

⁴UPIICSA, SEPI, Instituto Politécnico Nacional, Ciudad de México 08400, Mexico

⁵Departamento de Ciencias Básicas, Instituto Tecnológico de Iztapalapa, Tecnológico Nacional de México, Ciudad de México 09208, Mexico

Corresponding author: Victor Manuel Hernández-Guzmán (vmhg@uaq.mx)

This work was partially supported by the Comisión de Operación y Fomento de Actividades Académicas del I.P.N. The work of Celso Márquez-Sánchez was supported in part by the Sistema Nacional de Investigadores del Consejo Nacional de Ciencia y Tecnología, in part by the Beca de Estímulo Institucional de Formación de Investigadores (BEIFI) Program, and in part by the Consejo Nacional de Ciencia y Tecnología Scholarship. The work of Victor Manuel Hernández-Guzmán was supported by the Sistema Nacional de Investigadores, México. The work of Ramón Silva-Ortigoza was supported in part by the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional, in part by the Instituto Politécnico Nacional Programs Estímulos al Desempeño de los Investigadores and Sistema Institucional de Becas por Exclusividad, and in part by the Sistema Nacional de Investigadores, México. The work of Mayra Antonio-Cruz was supported in part by the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional via the Research Project 20200162, and in part by the Sistema Nacional de Investigadores del Consejo Nacional de Ciencia y Tecnología. The work of Carlos Alejandro Merlo-Zapata was supported by the Instituto Tecnológico de Iztapalapa.

ABSTRACT With the intention of delivering an intuitive and simple platform to plan, generate, and track trajectories with a WMR, this paper presents a technological integration of hardware and software. This technological integration consists of a mobile computing device with an app developed by authors for the trajectory planning and a differential drive WMR that possesses an embedded system, in which a trajectory generation algorithm and a trajectory tracking control algorithm –both proposed by authors– are programmed. The mobile computing device wirelessly drives the WMR through the embedded system. The app is developed on the basis of the sketch approach, since it allows drawing sketches of the trajectory intuitively on the mobile device. Also, the app geometrically scales the sketch coordinates to match them with the real WMR workspace. For the trajectory generation, these scaled coordinates are wirelessly sent to the embedded system, where the trajectory generation algorithm joins them by using Bézier polynomials and concatenation of straight-lines. Hence, the tracking task is carried out with the control algorithm, which is proposed departing from the dynamic model of the WMR and whose asymptotic stability proof is performed with the Lyapunov method. Experimental results verify that the technological integration successfully plans, generates, and tracks trajectories, even when the WMR is far from the initial coordinate of the desired trajectory. These results, corroborate the good performance and robustness of the proposed control algorithm. Thus, the introduced technological integration is intuitive and simple since users only have to draw a sketch on the mobile device to carry out the trajectory tracking task.

INDEX TERMS Wheeled mobile robot, trajectory tracking task, trajectory planning, trajectory generation, sketch approach, Bézier polynomial, mobile computing device, embedded system, tracking control.

The associate editor coordinating the review of this manuscript and approving it for publication was Wonhee Kim¹.

I. INTRODUCTION

Initially manipulators and hybrid and mobile robots allowed the mechanical automatization of the industrial activities

[1]–[3]. Currently, with the Information and Communication Technologies and the Internet of Things, the robots have been provided with better autonomy, functionalities, and capabilities, allowing them better interaction with different entities and streamlining their decision-making [4]. Thus, it can be found robots that are controlled remotely [5], that receive and send information in real time [6], that are capable of making “instantaneous” decisions in different scenarios [7], etc., contributing to the industrial globalization and giving rise to the industry 4.0. In that direction, this paper is focused on delivering an intuitive and simple platform to wirelessly plan, generate, and track trajectories with a WMR. This is achieved by integrating a mobile computing device and a differential drive WMR dotted with an embedded system, through a trajectory planning app, a trajectory generation algorithm, and a trajectory tracking control algorithm.

Literature dealing with the integration of a mobile computing device with a WMR is as follows. Chatterjee and Matsuno [8] introduced an off-line method for a WMR to follow a route drawn by hand on a scaled real scene map with predefined obstacles. Neural networks and a mobile device were used. Similarly, Skubic *et al.* [9], [10] and Chronis and Skubic [11], [12] presented a tool that allowed sketching, on a Personal Digital Assistant (PDA), a scaled real scene map and a route that the WMR had to perform. The route captured with the PDA was sent to a computer so that a fuzzy logic algorithm generated the real route for the WMR. Additionally, Skubic *et al.* [13] introduced a user interface prototype to drive a group of WMR with a tablet. The route to be followed and the obstacles to be avoided by the WMR were sketched in the tablet. The interface was validated via an usability study. Furthermore, Sakamoto *et al.* [14] developed an interface of stroke gestures to command a WMR in order to clean a house. Images of the house were collected via ceiling cameras and transmitted to a tablet, so that the user indicated with a hand-drawn sketch the route or area of the house to be cleaned by the robot. Moreover, Shah *et al.* [15] designed a human-machine interface based on the probabilistic multi-stroke sketching. The strokes were modeled with the variable duration hidden Markov method, whereas a search algorithm along with a heuristic found the most likely sketch until confirmation. Mission objectives were successfully executed for the WMR operated remotely. By their part, Jung *et al.* [16] reported a mobile interface based on sketch to accomplish telepresence in real static workspaces. For this, a WMR with monitor, cameras, and speakers was used to interact with the workspace, whereas a tablet was used to sketch routes and gestures for the WMR. Boniardi *et al.* [17], [18] introduced an approach for localization and navigation based on hand-drawn sketches of the environment. The Monte Carlo algorithm, along with the local deformation of the sketch, was used to estimate the robot state. An usability test was carried out to validate the approach. Lastly, Mielle *et al.* [19] carried out a method to interpret sketch-maps using Voronoi diagrams. Also, they introduced a graph matching method to find correspondences between

sketch-maps and ground truth maps. Both methods were successfully validated. Other important papers were reported by Negulescu and Inamura [20] and Samakrob *et al.* [21], who developed computer graphic interfaces where real scene maps, obstacles, and routes were drawn using a mouse.

On the other hand, the literature review associated with the implementation of trajectory tracking controllers for WMR was carried out in [22]. From this review, it is concluded that, in general, this implementation has been carried out by using personal computers, DSP, FPGA, and embedded systems of high capabilities, such as the ones from dSPACE, LabView, Sensoray, Xilinx, etc. This equipment requires license for specialized software as Matlab, ControlDesk, LabView, VHDL, etc. [23]–[25]. Hence, specialized knowledge and training are required for the programming related to the planning and generation of desired trajectories, as well as for programming and executing the tracking controller. Also, in such a case, every time a new trajectory is wanted to execute, it has to be programmed, load, and run with the specialized software. This implies that the WMR performing the trajectory tracking task for industrial applications [26] would have to be operated by specialized employees.

From the literature review two fashions are found: *i*) The one concentrated on developing interfaces based on the sketch approach for mapping environments and drawing routes, which does not include the implementation and performance of tracking controllers for WMR. *ii*) The one focused on the design, implementation, and performance of tracking controllers for WMR, where intuitive and simple user interfaces were not used. Also, some observations have to be made regarding the papers dealing with fashion *i*: Not all of the developed interfaces were connected to a real WMR and none of them described the algorithm to carry out the planning and generation of the routes for the real workspace of the WMR. Thus, this paper combines fashion *i* and *ii* to deliver an intuitive and simple platform that allows carrying out the planning, generation, and tracking of trajectories with a WMR. This, through the integration of a mobile device that includes a trajectory planning app and a differential drive WMR, which has an embedded system dotted with a trajectory generation algorithm and a trajectory tracking control algorithm. The trajectory planning app is developed on the basis of the sketch approach, the trajectory generation algorithm is based on Bézier polynomials and concatenation of straight-lines, and the tracking control algorithm is proposed departing from the dynamic model of the WMR. An advantage of this technological integration of hardware and software is that specialized training and expertise are not required for the platform operation, so that any user could intuitively and easily control the WMR through the mobile device.

The rest of the paper is organized as follows. The technological integration of hardware is described in Section II. The trajectory planning app developed by authors is introduced in Section III, whereas the own authorship trajectory genera-

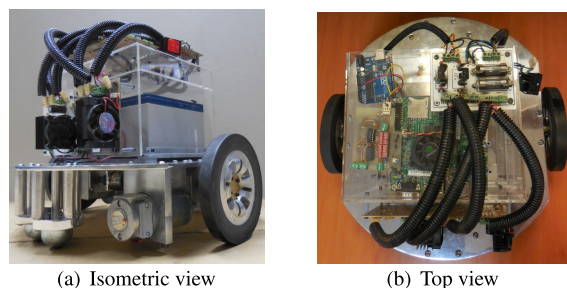


FIGURE 1. Differential drive WMR prototype with embedded system.

tion algorithm is detailed in Section IV. Then, the trajectory tracking control algorithm is proposed in Section V. The experimental results of the whole platform, i.e. of the technological integration of hardware and software, are presented in Section VI. The conclusion is given in Section VII and a comparative assessment of the performance of the proposed trajectory tracking control algorithm is presented in Appendix.

II. TECHNOLOGICAL INTEGRATION OF HARDWARE

The hardware involved in the technological integration is described below.

- 1) **A mobile computing device.** This is a SM-T800 tablet from Samsung, which has a screen of 10.5", a CPU Samsung Exynos Octa, and a 3 GB RAM. The tablet allows sketching by hand a representation of the trajectory to be tracked by the WMR through an app developed in Android, which shall be described in Section III.
- 2) **An embedded system.** It is a Jetson TK1 board from NVIDIA, wirelessly communicated with the mobile device, which is part of a differential drive WMR prototype and receives a set of coordinates of the hand-drawn sketch that represents the desired trajectory for a WMR prototype, but geometrically scaled to the real workspace. With these scaled coordinates, the Jetson TK1 board executes the trajectory generation algorithm that delivers the desired trajectory for a WMR prototype in the real workspace. Also, the Jetson TK1 board computes the trajectory tracking control algorithm for a WMR prototype to perform the tracking task. The most important characteristics of this board are its ARM Cortex A15 CPU, RAM of 2 GB, and Operating System (OS) Ubuntu 14.04.
- 3) **A differential drive WMR prototype.** This was built at the Mechatronics Laboratory of CIDETEC-IPN and is shown in Fig. 1. The rest of the parts of the prototype are: *a) Base*, which is made of aluminum and whose dimensions are 422.84 mm \times 350 mm from end to end with 6 mm of thickness. *b) Actuators*, are two DC motors from MicroMo of the model GNM3150, with the G2.6 gearbox. *c) Sensors*, two E6B2-CWZ6C incremental encoders from Omron are used to measure the angular position of the WMR wheels, from

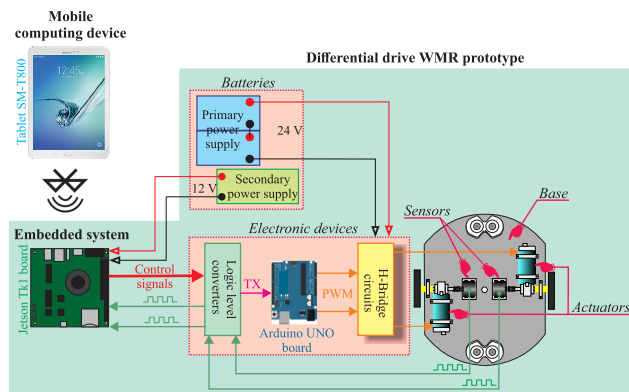


FIGURE 2. Diagram of the technological integration of hardware.

which the angular velocities of the wheels are computed. The resolution of the encoders is 2000 ppr. *d) Batteries*, two are rechargeable of 12 V_{CD} – 12 A/h and connected to integrate a primary power supply of 24 V_{CD}, which supplies energy to the motors, sensors, and electronic devices; while a secondary power supply is obtained with a lithium polymer battery of 12 V_{CD} – 20,000 mAh to supply energy to the embedded system. *e) Electronic devices*, this part includes a bluetooth module for the wireless communication between the embedded system and the mobile device, an Arduino UNO board which receives the duty cycles to generate the PWM signals that activate two H-bridge circuits to drive the DC motors, two logic level converters that receive and electrically treat the signals from the bluetooth module and the encoders, and voltage regulators used to reduce the voltage of the primary power supply to the one required for the different electronic devices.

A graphical representation of the technological integration of hardware is shown in Fig. 2. The software that completes the intuitive and simple platform to carry out the trajectory tracking task with a WMR, that is the trajectory planning app, the trajectory generation algorithm, and the trajectory tracking control algorithm, are introduced separately in next sections.

III. TRAJECTORY PLANNING APP

For WMR, the traditional methods for trajectory planning are road maps, cell decomposition, and artificial potential fields [29]. However, these methods imply algorithms with complex mathematical equations. Thus, by using the sketch approach, an app that allows planning trajectories intuitively in a mobile computing device is proposed in this section.

The operation of the trajectory planning app consists of three stages: 1) Hand-draw a sketch on the screen of the mobile computing device (tablet) to represent the desired trajectory for the WMR. 2) Geometrically scale the coordinates of the hand-drawn sketch to the real measurements of the WMR workspace. 3) Wirelessly send the scaled coordinates

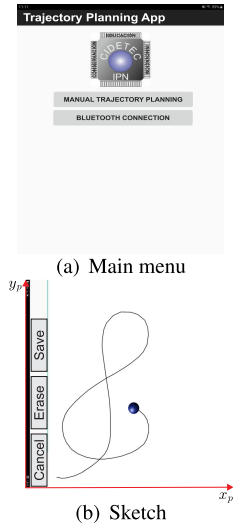


FIGURE 3. App developed.

to the embedded system (Jetson TK1 board). This app was developed in the open source OS Android, since it is the OS most used in mobile devices [30], and has the following buttons:

- *Manual trajectory planning.* This button allows performing stages 1 and 2. The main menu of the app and a hand-drawn sketch are shown in Fig. 3. In the sketch, the circle represents the WMR and x_p, y_p are the references from which the coordinates of the sketch are obtained, which are related to the pixels of the screen of the tablet.
- *Bluetooth connection.* This button pares the tablet with the Jetson TK1 board, so that stage 3 is carried out.

For the stage 2 of the operation of the trajectory planning app, the following own authorship algorithm is executed:

- 1) Since the sketch is a set of consecutive pixels, see Fig. 4(a), take the coordinates of h pixels of the hand-drawn sketch with the `MotionEvent` class to generate the following array:

$$T = \{(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_h, y_h)\}.$$

The `MotionEvent` class provides the methods `getX(int)` and `getY(int)` to query x_0, \dots, x_h and y_0, \dots, y_h , respectively. Also, note that h is the number of columns of T , hence it is obtained with instruction `length`. Both class and instruction are of the OS Android.

- 2) Carry out a geometric scaling of the hand-drawn sketch according to the dimension of the workspace. In this case the workspace has $2 \text{ m} \times 2 \text{ m}$, from which a space of $1.6 \text{ m} \times 1.6 \text{ m}$ is considered because, when executing tracking, the middle point of the axis that join the wheels of the WMR is the one taken into account to achieve the task. This consideration ensures that the WMR performs trajectories inside the workspace. Since T is a two dimension array of two rows and h columns, its rows are decomposed in two arrays, T_x

and T_y , with the values x_j and y_j , respectively, where $j = 0, 1, 2, \dots, h$. From T_x and T_y the smallest values, x_{\min} and y_{\min} , and the largest values, x_{\max} and y_{\max} , are found. With these values, considering the real workspace, and the function transformation method, the scaling factors, ϑ_x and ϑ_y , where found. Therefore, the arrays with the scaled values for the real workspace are computing by using the following expressions:

$$T_{xr} = \vartheta_x T_x, \tag{1}$$

$$T_{yr} = \vartheta_y T_y, \tag{2}$$

with

$$\vartheta_x = \begin{cases} 1 - (l_x - 1.6), & (l_x - 1.6) < 1 \\ \left(\frac{1.6}{l_x}\right), & (l_x - 1.6) > 1, \end{cases}$$

$$\vartheta_y = \begin{cases} 1 - (l_y - 1.6), & (l_y - 1.6) < 1 \\ \left(\frac{1.6}{l_y}\right), & (l_y - 1.6) > 1, \end{cases}$$

where $l_x = x_{\max} - x_{\min}$ and $l_y = y_{\max} - y_{\min}$. Then, T_{xr} and T_{yr} becomes in the first and second row, respectively, of a new two dimension array, T_r . The resulting points are exemplified in Fig. 4(b).

- 3) Compute the distance between the scaled points, whose coordinates are in T_r , with the following expression:

$$d_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2}, \tag{3}$$

until $j = h - 1$. Then, take from T_r only the points with a distance equal or greater than 1 cm between them. The rest of points are eliminated, that is, the length of the T_r columns is modified (see Fig. 5(a)).

- 4) In the modified T_r , after computing the distance with (3), add points between the points separated with more than 2 cm (see Fig. 5(b)). Again, the length of the T_r columns is modified.

Thus, T_r is the information wirelessly sent to the embedded system with the *Bluetooth connection* button, which is used to generate the desired trajectory in the next section. Note that the described algorithm allows the trajectory planning in any size screen, since relies only on the workspace size. Also, it is important to mention that the app does not demands high performance, since the method for the trajectory planing is simple, so that it can be executed in mobile devices with low-performance.

IV. TRAJECTORY GENERATION ALGORITHM

This section proposes an algorithm to generate the desired trajectory for the WMR. Some methods that are commonly used in the trajectory generation are parametric functions, splines, Bézier polynomials, and concatenation of straight-lines [28], [31]. In this paper, a combination of first order Bézier polynomials and concatenation of straight-lines is used. This due to Bézier polynomials ensure to join two points continuously.

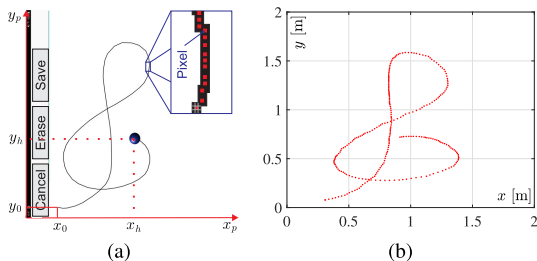


FIGURE 4. Geometric scaling.

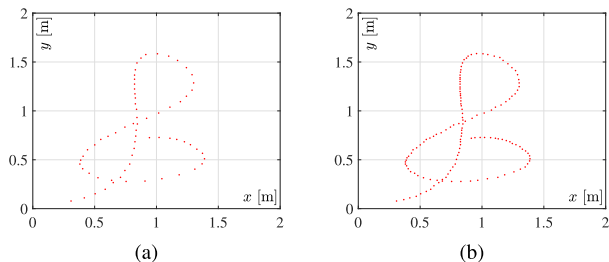


FIGURE 5. Removing and collocating points.

Also, the algorithms based on Bézier polynomials consume less computing time and are the most numerically stable [32].

The algorithm that carries out the trajectory generation is, in general, described as follows:

- 1) Estimate the distances between each of the points of the last T_r resulting in previous section. To this end, redefine $j = 0, 1, 2, \dots, k$, with k being the columns of the last T_r , and compute (3) until $j = k - 1$. These distances are saved in an array, D , and summed to calculate the length, φ , in meters of the desired trajectory that shall be generated. That is:

$$D[j] = d_j, \quad \varphi = \sum_{j=0}^{k-1} D[j]. \quad (4)$$

- 2) Compute the time duration in which the generated trajectory shall be executed by the WMR in experiment when performing tracking. This is achieved with the following relation:

$$t = \frac{\varphi}{v_p}, \quad (5)$$

where v_p is a proposed velocity, which cannot be out of the velocity range of the real WMR.

- 3) Calculate the time in which the Bézier Polynomials shall interpolate between the points in T_r , with the intention of join them later. Then, an array of times is generated with the expression given by

$$t_s[j] = \frac{D[j]}{v_p}. \quad (6)$$

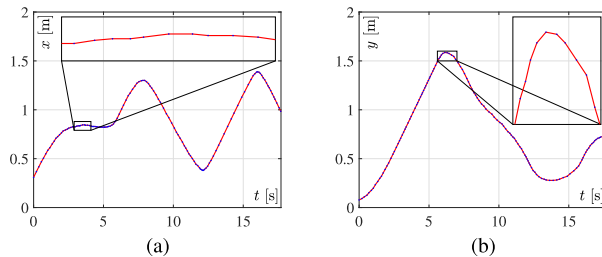


FIGURE 6. Join of points through Bézier polynomials.

- 4) Connect the values of the first row of T_r using the first order Bézier polynomial defined by

$$B(t) = \left(\frac{t - t_i}{t_f - t_i} \right) (P_f - P_i) + P_i, \quad (7)$$

with $P_i = T_r[0, j]$ and $P_f = T_r[0, j + 1]$ being the constant values between which $B(t)$ smoothly interpolates in the interval $[t_i, t_f]$, where t_i is the initial time and t_f the final one, which are given by

$$t_i = \begin{cases} 0, & j = 0 \\ \sum_{j=0}^{j-1} t_s[j], & j \geq 1, \end{cases} \quad t_f = t_i + t_s[j].$$

- 5) Connect the values of the second row of T_r proceeding as in the previous step, that is, using (7). In this case $P_i = T_r[1, j]$ and $P_f = T_r[1, j + 1]$ are redefined.

The steps 4 and 5 for the sketch points in Fig. 5(b) are represented in Fig. 6. The information obtained with the trajectory generation algorithm is required by the trajectory tracking control algorithm, which shall be described in next section.

V. TRAJECTORY TRACKING CONTROL ALGORITHM

This section proposes the control algorithm to achieve the trajectory tracking task. This proposal is carried out departing from the WMR dynamic model. Also, the stability analysis of the control algorithm as well as the program developed to execute it in the embedded system are introduced. Before that, some remarks on the dynamic model of the WMR are recalled.

In order to render the paper self contained and for the ease of reference of the reader, the following result taken from the control literature is presented, which is useful for the purposes here pursued:

Theorem 1: [Theorem 4.18 in [33], p. 172] Let $D \subset \mathcal{R}$ be a domain that contains the origin and $V : [0, \infty) \times D \rightarrow \mathcal{R}$ be a continuously differentiable function such that

$$\alpha_1(\|x\|) \leq V(t, x) \leq \alpha_2(\|x\|), \quad (8)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -W_3(x), \quad \forall \|x\| \geq \mu > 0, \quad (9)$$

$\forall t \geq 0$ and $\forall x \in D$, where α_1 and α_2 are class \mathcal{K} and $W_3(x)$ is a continuous positive definite function. Take $r > 0$ such that $B_r \subset D$ and suppose that

$$\mu < \alpha_2^{-1}(\alpha_1(r)). \quad (10)$$

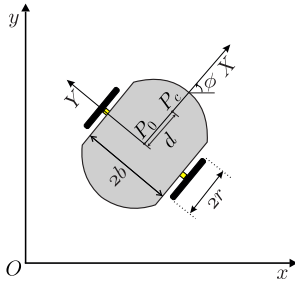


FIGURE 7. WMR diagram.

Then, there exists a class \mathcal{KL} function β and for every initial state $x(t_0)$, satisfying $\|x(t)\| \leq \alpha_1^{-1}(\alpha_1(r))$, there is $T \geq 0$ (dependent on $x(t_0)$ and μ) such that the solution of $\dot{x} = f(t, x)$ satisfies

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0), \quad \forall t_0 \leq t < t_0 + T, \quad (11)$$

$$\|x(t)\| \leq \alpha_1^{-1}(\alpha_2(\mu)), \quad \forall t_0 \geq \quad \forall t_0 + T. \quad (12)$$

Moreover, if $D = \mathcal{R}^n$ and α_1 belongs to class \mathcal{K}_∞ , then (11) and (12) hold for any initial state $x(t_0)$, with no restriction on how large μ is.

Roughly speaking, Theorem 1 states that x converges into a domain whose radius is bounded by $\alpha_1^{-1}(\alpha_2(\mu))$ which is known as the ultimate bound. Also, some useful properties of the Euclidean and the spectral norms are as follows. Given some $n \times n$ matrices F and G , with the former symmetric, and some $y, x \in \mathcal{K}^n$, then [34], p. 176, [35], p. 26:

$$\lambda_{\min}(F)\|y\|^2 \leq y^T F y \leq \lambda_{\max}(F)\|y\|^2, \quad \forall y \in \mathcal{R}^n,$$

$$\pm y^T G x \leq |y^T G x| \leq \|y\| \|G\| \|x\|, \quad \forall y, x \in \mathcal{R}^n.$$

A. DYNAMIC MODEL OF THE WMR

The Fig. 7 shows the differential drive WMR. There, $2b$ is the width of the mobile robot and r is the radius of each wheel. The xy world coordinate system is O and P_0 is the XY coordinate system fixed to the mobile robot. Also, the origin of the coordinate system P_0 is placed at the middle of the axis which is common to the right and left driving wheels. The center of mass of the mobile robot is P_c , which is on the X -axis, and the distance from P_0 to P_c is d . For the latter description, m_c and m_w are, respectively, the mass of the body and the mass of the wheel with a motor. Whereas I_c, I_w , and I_m are the moment of inertia of the body about the vertical axis through P_c , the moment of inertia of the wheel with a motor about the wheel axis, and the moment of inertia of the wheel with a motor about the wheel diameter, respectively.

The configuration of the mobile robot can be described by the five generalized coordinates $q = [x, y, \phi, \theta_r, \theta_l]^T$ where (x, y) are the coordinates of P_0 , ϕ is the heading angle of the mobile robot, and θ_r, θ_l are the angles of the right and left driving wheels. It is assumed that the wheels roll and do not slip. The dynamic model of this mobile robot is given by [27]:

$$\dot{q} = S(q)v(t), \quad (13)$$

$$\bar{M}(q)\dot{v} + \bar{C}(q, \dot{q})v + \bar{g}(q) = \bar{B}(q)\tau, \quad (14)$$

where

$$S(q) = \begin{bmatrix} \frac{r}{2} \cos \phi & \frac{r}{2} \cos \phi \\ \frac{r}{2} \sin \phi & \frac{r}{2} \sin \phi \\ \frac{r}{2b} & -\frac{r}{2b} \\ 0 & 1 \end{bmatrix},$$

$$M(q) = \begin{bmatrix} \frac{r^2}{4b^2}(mb^2 + I) + I_w & \frac{r^2}{4b^2}(mb^2 - I) \\ \frac{r^2}{4b^2}(mb^2 - I) & \frac{r^2}{4b^2}(mb^2 + I) + I_w \end{bmatrix},$$

$$\bar{C}(q, \dot{q}) = \begin{bmatrix} 0 & \frac{r^2}{2b} m_c d \dot{\phi} \\ -\frac{r^2}{2b} m_c d \dot{\phi} & 0 \end{bmatrix},$$

$$\bar{B}(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \bar{g}(q) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (15)$$

$\tau = [\tau_r, \tau_l]^T$ consists of torques τ_r and τ_l applied at the right and the left driving wheels. $v = [v_1, v_2]^T$ represents angular velocities of the right and the left driving wheels, v_1 and v_2 respectively. Whereas $m = m_c + 2m_w$ and $I = m_c d^2 + 2m_w b^2 + I_c + 2I_m$. Finally, defining $v = \text{sign}(v_1 + v_2)\sqrt{\dot{x}^2 + \dot{y}^2}$ and $w = \dot{\phi}$, the following is obtained from (13):

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}, \quad (16)$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = A \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad A = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix}, \quad (17)$$

$$\text{sign}(v_1 + v_2) = \begin{cases} +1, & v_1 + v_2 \geq 0 \\ -1, & v_1 + v_2 < 0 \end{cases}, \quad (18)$$

where A is a nonsingular matrix.

B. PROPOSAL OF THE CONTROL ALGORITHM AND STABILITY ANALYSIS

Proposition 1: Consider dynamics (14), (16), (17) and (18) in closed loop with controller:

$$\tau = K_p \tilde{v} + K_i \int_0^t \tilde{v}(s) ds, \quad (19)$$

$$v_d = A^{-1} \begin{bmatrix} v_d \\ w_d \end{bmatrix}, \quad \tilde{v} = v_d - v = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \end{bmatrix}, \quad (20)$$

$$w_d = -k_{p\phi} \phi_e - k_{d\phi} \frac{d\phi_e}{dt} + k_{d\phi} w, \quad \phi_e = \phi - \phi^*, \quad (21)$$

$$v_d = k_{pv} (\chi^* - \chi)^T \hat{\phi}, \quad \hat{\phi} = [\cos(\phi), \sin(\phi)]^T,$$

where ϕ^* and ϕ_e are defined in Fig. 8, $\chi^* = [x^*(t), y^*(t)]^T$ and $\chi = [x, y]^T$ represent the time varying desired trajectory and the actual robot position, respectively. If $\chi^* - \chi = 0$, then ϕ_e is defined as the internal angle between vectors $\dot{\chi}^* = [\dot{x}^*(t), \dot{y}^*(t)]^T$ and $\hat{\phi}$ where $\hat{\phi}$ stands for the unitary vector

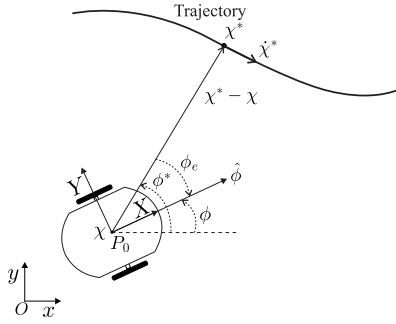


FIGURE 8. Trajectory tracking task representation.

depicted in Fig. 8. Moreover:

$$\begin{aligned} (\chi^* - \chi)^T \hat{\phi} &= (x^*(t) - x) \cos(\phi) + (y^*(t) - y) \sin(\phi), \\ &= \|\chi^* - \chi\| \cos(\phi_e), \end{aligned}$$

because $(\chi^* - \chi)^T \hat{\phi}$ represents the standard scalar product of vector $(\chi^* - \chi)$ and the unitary vector $\hat{\phi}$, see Fig. 8. $K_p = \text{diag}\{k_{p1}, k_{p2}\}$, $K_i = \text{diag}\{k_{i1}, k_{i2}\}$, and k_{pv} represent positive constants. The state

$$\left(\phi_e, \tilde{v}_1, \tilde{v}_2, \int_0^t \tilde{v}_1(s) ds, \int_0^t \tilde{v}_2(s) ds, (\chi^* - \chi)^T \right)$$

has an ultimate bound which can be rendered arbitrarily small by a suitable choice of the positive controller gains $k_{p\phi}$, $k_{d\phi}$, k_{p1} , k_{p2} , k_{i1} , k_{i2} , k_{pv} , provided that WMR is not required to move too fast.

Remark 1: Controller in Proposition 1 is composed by a proportional-derivative (PD) trajectory tracking controller (21) which computes the WMR desired angular velocity w_d . This controller is driven by the orientation angle error ϕ_e , i.e. if $\phi_e \rightarrow 0$ then robot orientation points toward the desired point on the desired trajectory. The WRM desired translational velocity v_d is proportional to distance existing between the WMR actual position on the xy plane and the desired point on the desired trajectory. However, only the component of such a distance in the direction of the actual WMR orientation is considered. The expression in (19) computes the actual torques to be applied at the robot wheels in order to force both the actual WMR angular velocity w and the actual WMR translational velocity v to reach their desired values w_d and v_d , respectively. These torques are computed using two proportional-integral (PI) velocity controllers driven by the wheels velocity errors. According to (20) if these errors tend to zero, i.e. if $\tilde{v} \rightarrow 0$, then $w \rightarrow w_d$ and $v \rightarrow v_d$. This is the rationale behind controller in Proposition 1.

Proof: Replacing (19) in (14) and adding and subtracting some convenient terms it is found:

$$\bar{M}(q) \dot{\tilde{v}} + \bar{C}(q, \dot{q}) \tilde{v} + K_p \tilde{v} + K_i z = \bar{M}(q) \dot{v}_d + \bar{C}(q, \dot{q}) v_d, \quad (22)$$

$$\dot{z} = \tilde{v}. \quad (23)$$

Expression (23) is included in order to render easier the variable manipulation in the algebraic procedure that follows.

Note that, according to (17) and (20), the following is obtained:

$$\tilde{v} = A^{-1} \begin{bmatrix} \tilde{v} \\ \tilde{w} \end{bmatrix}, \quad \tilde{v} = v_d - v, \quad \tilde{w} = w_d - w. \quad (24)$$

Using $\dot{\phi} = w$ it can be written:

$$\dot{\phi} = w_d - \tilde{w}, \quad (25)$$

$$\dot{\phi}^* = w - \frac{d\phi_e}{dt}. \quad (26)$$

Replacing (26) in w_d given in (21) and replacing w_d in (25), it is obtained:

$$\dot{\phi}_e = -\tilde{w} - k_{p\phi} \phi_e + (k_{d\phi} - 1) \dot{\phi}^*. \quad (27)$$

Finally, the closed loop dynamics is given by (22), (23), (27) and

$$\frac{d}{dt} (\chi^* - \chi) = \begin{bmatrix} \dot{x}^*(t) - v \cos \phi \\ \dot{y}^*(t) - v \sin \phi \end{bmatrix}.$$

Now, consider the following continuously differentiable function:

$$\begin{aligned} V(\phi_e, \tilde{v}, z, \chi^* - \chi) &= \frac{1}{2} \phi_e^2 + \frac{1}{2} \tilde{v}^T \bar{M}(q) \tilde{v} + z^T \bar{M}(q) \tilde{v} \\ &\quad + \frac{1}{2} z^T K_i z + \frac{1}{2} z^T K_p z \\ &\quad + \frac{1}{2} (\chi^* - \chi)^T (\chi^* - \chi). \end{aligned} \quad (28)$$

Note that (28) can be bounded as follows:

$$\begin{aligned} V &\geq \frac{1}{2} |\phi_e|^2 + \frac{1}{2} \lambda_{\min} \{\bar{M}(q)\} \|\tilde{v}\|^2 \\ &\quad - \lambda_{\max} \{\bar{M}(q)\} \|z\| \|\tilde{v}\| + \frac{1}{2} \lambda_{\min} \{K_i\} \|z\|^2 \\ &\quad + \frac{1}{2} \lambda_{\min} \{K_p\} \|z\|^2 + \frac{1}{2} (\chi^* - \chi)^T (\chi^* - \chi), \\ &= \frac{1}{2} |\phi_e|^2 + \frac{1}{2} \begin{bmatrix} \|\tilde{v}\| \\ \|z\| \end{bmatrix}^T Q_1 \begin{bmatrix} \|\tilde{v}\| \\ \|z\| \end{bmatrix} \\ &\quad + \frac{1}{2} (\chi^* - \chi)^T (\chi^* - \chi), \\ &\geq \frac{1}{2} |\phi_e|^2 + \frac{1}{2} \lambda_{\min} \{Q_1\} \left[\|\tilde{v}\|^2 + \|z\|^2 \right] \\ &\quad + \frac{1}{2} \|\chi^* - \chi\|^2 \\ &\geq \frac{1}{2} \min \{1, \lambda_{\min} \{Q_1\}\} \|\phi_e, \tilde{v}_1, \tilde{v}_2, z_1, z_2, \tilde{\chi}_1, \tilde{\chi}_2\|^2, \\ Q_1 &= \begin{bmatrix} \lambda_{\min} \{\bar{M}(q)\} & -\lambda_{\max} \{\bar{M}(q)\} \\ -\lambda_{\max} \{\bar{M}(q)\} & \lambda_{\min} \{K_i\} + \lambda_{\min} \{K_p\} \end{bmatrix}, \\ \tilde{\chi} &= \chi^* - \chi = [\tilde{\chi}_1, \tilde{\chi}_2]^T. \end{aligned}$$

Hence, it is concluded that:

$$V \geq \alpha_1 (\|\xi\|) = c_1 \|\xi\|^2, \quad \forall \xi \in R^7, \quad (29)$$

with $\xi = [\phi_e, \tilde{v}_1, \tilde{v}_2, z_1, z_2, \tilde{\chi}_1, \tilde{\chi}_2]^T$ and $c_1 = \frac{1}{2} \min \{1, \lambda_{\min} \{Q_1\}\}$.

On the other hand, it is possible to upper bound (28) as:

$$\begin{aligned}
 V &\leq \frac{1}{2} |\phi|^2 + \frac{1}{2} \lambda_{\max} \{ \bar{M}(q) \} \|\tilde{v}\|^2 \\
 &\quad + \lambda_{\max} \{ \bar{M}(q) \} \|z\| \|\tilde{v}\| + \frac{1}{2} \lambda_{\max} \{ K_p \} \|z\|^2 \\
 &\quad + \frac{1}{2} \lambda_{\max} \{ K_i \} \|z\|^2 + \frac{1}{2} (\chi^* - \chi)^T (\chi^* - \chi), \\
 &= \frac{1}{2} |\phi_e|^2 + \frac{1}{2} \left[\frac{\|\tilde{v}\|}{\|z\|} \right]^T \\
 &\quad \times \underbrace{\begin{bmatrix} \lambda_{\max} \{ \bar{M}(q) \} & \lambda_{\max} \{ \bar{M}(q) \} \\ \lambda_{\max} \{ \bar{M}(q) \} & \lambda_{\max} \{ K_i \} + \lambda_{\max} \{ K_p \} \end{bmatrix}}_{Q_2} \\
 &\quad \times \left[\frac{\|\tilde{v}\|}{\|z\|} \right] + \frac{1}{2} (\chi^* - \chi)^T (\chi^* - \chi), \\
 &\leq \frac{1}{2} |\phi_e|^2 + \frac{1}{2} \lambda_{\max} \{ Q_2 \} \left[\|\tilde{v}\|^2 + \|z\|^2 \right] \\
 &\quad + \frac{1}{2} (\chi^* - \chi)^T (\chi^* - \chi) \\
 &\leq \frac{1}{2} \max \{ 1, \lambda_{\min} \{ Q_2 \} \} \|\xi\|^2. \tag{30}
 \end{aligned}$$

According to (29) and (30), it is had that

$$\alpha_1 (\|\xi\|) \leq V(\xi) \leq c_2 \|\xi\|^2 = \alpha_2 (\|\xi\|), \tag{31}$$

$$c_2 = \frac{1}{2} \max \{ 1, \lambda_{\min} \{ Q_2 \} \}. \tag{32}$$

Note that α_1 and α_2 are class \mathcal{K}_∞ functions.

It is straightforward to verify that the time derivative of V along the trajectories of the closed loop dynamics (22), (23), (27) is given as:

$$\begin{aligned}
 \dot{V} &= -\tilde{w}\phi_e - k_{p\phi}\phi_e^2 + \phi_e(k_{d\phi} - 1)\dot{\phi}^* - \frac{1}{2}\tilde{v}^T K_p \tilde{v} \\
 &\quad - \frac{1}{2}\tilde{v}^T [K_p - 2\bar{M}(q)] \tilde{v} + z^T \bar{C}(q, \dot{q})^T \tilde{v} \\
 &\quad - z^T K_i z + (\tilde{v} + z)^T [\bar{M}(q) \dot{v}_d + \bar{C}(q, \dot{q}) v_d] \\
 &\quad + (\chi^* - \chi)^T \dot{\chi}^* - (\chi^* - \chi)^T \hat{\phi} v.
 \end{aligned}$$

It can be bounded the following term as:

$$\begin{aligned}
 &-\frac{1}{2} z^T K_i z + z^T [\bar{M}(q) \dot{v}_d + \bar{C}(q, \dot{q}) v_d] \\
 &\leq -\frac{1}{2} \lambda_{\min} \{ K_i \} \|z\|^2 \\
 &\quad + \|z\| \left[\lambda_{\max} \{ \bar{M}(q) \} \|\dot{v}_d\| + \frac{r^2}{2b} m_{cd} |\dot{\phi}| \|v_d\| \right] \\
 &\leq -\frac{1}{2} \lambda_{\min} \{ K_i \} \|z\|^2 + \|z\| s_3 \leq 0, \text{ if } \frac{2s_3}{\lambda_{\min} \{ K_i \}} \\
 &\leq \|z\| \leq \|\xi\|, \tag{33} \\
 s_3 &= \|A^{-1}\| \left[\lambda_{\max} \{ \bar{M}(q) \} \right. \\
 &\quad \times \sqrt{\max \{ k_{p\phi}, k_{d\phi} \}^2 \left(|\dot{\phi}_e| + |\ddot{\phi}_e| + |\ddot{\phi}|^2 + \dot{v}_d^2 \right)} \\
 &\quad \left. + \frac{r^2}{2b} m_{cd} |\dot{\phi}| \right]
 \end{aligned}$$

$$\times \sqrt{\max \{ k_{p\phi}, k_{d\phi} \}^2 \left(|\phi_e| + |\dot{\phi}_e| + |\dot{\phi}|^2 + v_d^2 \right)}.$$

In a similar way, it can be obtained

$$-\frac{1}{2} \tilde{v}^T K_p \tilde{v} + \tilde{v}^T [\bar{M}(q) \dot{v}_d + \bar{C}(q, \dot{q}) v_d] \leq 0, \tag{34}$$

if $\frac{2s_3}{\lambda_{\min} \{ K_p \}} \leq \|\tilde{v}\| \leq \|\xi\|$. Therefore, it can be bounded \dot{V} as:

$$\begin{aligned}
 \dot{V} &\leq -k_{p\phi} |\phi_e|^2 - \frac{1}{2} \tilde{v}^T \underbrace{[K_p - 2\bar{M}(q)]}_{K_{PM}} \tilde{v} - \frac{1}{2} z^T K_i z \\
 &\quad - \tilde{w}\phi_e + (k_{d\phi} - 1) \phi_e \dot{\phi}^* - z^T \bar{C}^T(q, \dot{q}) \tilde{v} \\
 &\quad + (\chi^* - \chi)^T \dot{\chi}^* - (\chi^* - \chi)^T \hat{\phi} (v_d - \tilde{v}),
 \end{aligned}$$

where $\tilde{v} = v_d - v$. Matrix K_{PM} is positive definite (and, hence, all of its eigenvalues are positive) if it is strictly diagonally dominant and its diagonal entries are positive [34], i.e.:

$$\begin{aligned}
 k_{p1} - 2(|\bar{M}_{11}| + |\bar{M}_{12}|) &> 0, \\
 k_{p1} - 2(|\bar{M}_{21}| + |\bar{M}_{22}|) &> 0.
 \end{aligned}$$

Note that:

$$(\chi^* - \chi)^T \hat{\phi} \tilde{v} = \|\chi^* - \chi\| \cos(\phi_e) \frac{r}{2} (\tilde{v}_1 + \tilde{v}_2),$$

and, hence:

$$\begin{aligned}
 -(\chi^* - \chi)^T \hat{\phi} (v_d - \tilde{v}) &= -k_{pv} \left[(\chi^* - \chi)^T \hat{\phi} \right]^2 \\
 &\quad + (\chi^* - \chi)^T \hat{\phi} \tilde{v}, \\
 &= -k_{pv} \left[\|\chi^* - \chi\| \cos(\phi_e) \right]^2 \\
 &\quad + \|\chi^* - \chi\| \cos(\phi_e) \\
 &\quad \times \frac{r}{2} (\tilde{v}_1 + \tilde{v}_2).
 \end{aligned}$$

Taking into account this, considering (15), (24) and using matrix notation, it can be written:

$$\begin{aligned}
 \dot{V} &\leq - \begin{bmatrix} |\phi_e| \\ |\tilde{v}_1| \\ |\tilde{v}_2| \\ |z_1| \\ |z_2| \end{bmatrix}^T Q_0 \begin{bmatrix} |\phi_e| \\ |\tilde{v}_1| \\ |\tilde{v}_2| \\ |z_1| \\ |z_2| \end{bmatrix} \\
 &\quad - \begin{bmatrix} |\tilde{v}_1| \\ |\tilde{v}_2| \\ \|\chi^* - \chi\| |\cos(\phi_e)| \end{bmatrix}^T Q_1 \\
 &\quad \times \begin{bmatrix} |\tilde{v}_1| \\ |\tilde{v}_2| \\ \|\chi^* - \chi\| |\cos(\phi_e)| \end{bmatrix} + |\phi_e| |k_{d\phi} - 1| |\dot{\phi}^*| \\
 &\quad - \frac{1}{2} k_{p\phi} \phi_e^2 - \frac{1}{2} k_{pv} \|\chi^* - \chi\|^2 \cos^2(\phi_e) \\
 &\quad + \|\dot{\chi}^*\| \|\chi^* - \chi\|,
 \end{aligned}$$

where

$$Q_0 = \begin{bmatrix} \frac{1}{2}k_{p\phi} & -\frac{r}{4b} & -\frac{r}{4b} & 0 & 0 \\ -\frac{r}{4b} & q_{md} & 0 & 0 & q_{di} \\ -\frac{r}{4b} & 0 & q_{md} & q_{di} & 0 \\ 0 & 0 & q_{di} & \frac{1}{2}k_{i1} & 0 \\ 0 & q_{di} & 0 & 0 & \frac{1}{2}k_{i2} \end{bmatrix},$$

$$Q_1 = \begin{bmatrix} q_{md} & 0 & -\frac{r}{4} \\ 0 & q_{md} & -\frac{r}{4} \\ -\frac{r}{4} & -\frac{r}{4} & \frac{1}{2}k_{pv} \end{bmatrix},$$

with

$$q_{md} = \frac{1}{4}\lambda_{\min}\{K_{PM}\}, \quad q_{di} = -\frac{r}{4b}m_{cd}|\dot{\phi}|$$

Hence, there always exists a large enough $k_{pv} > 0$ such that

$$-\frac{1}{2}k_{p\phi}\phi_e^2 - \frac{1}{2}k_{pv}\|\chi^* - \chi\|^2 \cos^2(\phi_e) + \|\dot{\chi}^*\| \|\chi^* - \chi\| \leq -\frac{1}{2}k_{pv1}\|\chi^* - \chi\|^2 \cos^2(\phi_e),$$

if

$$-\frac{1}{2}k_{p\phi}\phi_e^2 - \frac{1}{2}k_{pv2}\|\chi^* - \chi\|^2 \cos^2(\phi_e) + \|\dot{\chi}^*\| \|\chi^* - \chi\| < 0, \quad (35)$$

where

$$k_{pv} = k_{pv1} + k_{pv2}, \quad k_{pv1} > 0, \quad k_{pv2} > 0.$$

Since ϕ_e and $\cos^2(\phi_e)$ are not zero simultaneously, then (35) is always true provided that $k_{p\phi}$ and k_{pv2} are large enough and the product $\|\dot{\chi}^*\| \|\chi^* - \chi\|$ is not too large.

Remark 2: If both $\|\dot{\chi}^*\|$ and $\|\chi^* - \chi\|$ are large in practice, one possible situation is that the desired point χ^* is far from the WMR actual position χ and that distance between these points is increasing very fast. Hence, it is likely that the trajectory tracking task be interrupted if $\cos^2(\phi_e) = 0$. This explains why this condition might be a source of instability. Note however, that the trajectory tracking task might be restored when $\cos^2(\phi_e)$ becomes far from zero provided that the product $k_{pv2}\|\chi^* - \chi\|$ is large enough. This also means that WMR might lag the desired position, but distance between χ^* and χ remains within a bound if $\dot{\chi}^*$ also remains within a bound.

On the other hand, if:

$$-\frac{1}{2}k_{p\phi}\phi_e^2 + \|\dot{\chi}^*\| \|\chi^* - \chi\| < 0,$$

the trajectory tracking task is not interrupted despite $\cos^2(\phi_e) = 0$ for a while.

Furthermore, note that matrix Q_1 is positive definite if $\det Q > 0$, which is always satisfied by large enough constants $k_{pv} > 0$ and $\lambda_{\min}\{K_{PM}\} > 0$. Hence, it can be written:

$$\dot{V} \leq - \begin{bmatrix} |\phi_e| \\ |\tilde{v}_1| \\ |\tilde{v}_2| \\ |z_1| \\ |z_2| \end{bmatrix}^T Q_0 \begin{bmatrix} |\phi_e| \\ |\tilde{v}_1| \\ |\tilde{v}_2| \\ |z_1| \\ |z_2| \end{bmatrix} - \frac{1}{2}k_{pv1}\|\chi^* - \chi\|^2 \times \cos^2(\phi_e) + \|\xi\| |k_{d\phi} - 1| |\dot{\phi}^*|.$$

Matrix Q_0 is positive definite if all of its five leading principal minors are positive. This can be always be satisfied by using large enough controller gains $k_{p\phi}, k_{p1}, k_{p2}, k_{i1}, k_{i2}$. This means that Q_0 can be rendered positive definite and, hence, $\lambda_{\min}\{Q_0\} > 0$. Then, it is obtained:

$$\dot{V} \leq -c_3 \|\xi\|^2 + |k_{d\phi} - 1| |\dot{\phi}^*| \|\xi\|,$$

where

$$c_3 = \min \left\{ \lambda_{\min}\{Q_0\}, \frac{k_{pv1}}{2} \cos^2(\phi_e) \right\}.$$

Finally, taking into account (33), (34), and using some constant $0 < \theta < 1$, it can be written:

$$\dot{V} \leq -(1 - \theta) c_3 \|\xi\|^2 - \theta c_3 \|\xi\|^2 + |k_{d\phi} - 1| |\dot{\phi}^*| \|\xi\|, \leq -(1 - \theta) c_3 \|\xi\|^2 = -W_3(\xi), \quad \forall \|\xi\| \geq \mu > 0, \quad (36)$$

with $\mu = \max \left\{ \frac{|k_{d\phi} - 1| |\dot{\phi}^*|}{\theta c_3}, \frac{2s_3}{\lambda_{\min}\{K_p\}}, \frac{2s_3}{\lambda_{\min}\{K_i\}} \right\}$ and ξ satisfying (35).

We stress that $W_3(\xi)$ is a positive definite function provided that $|\dot{\phi}|$ is constrained to remain within a given upper bound and $\cos^2(\phi_e)$ is far from zero, i.e. according to (31) and (36) conditions (8) and (9) are satisfied locally. Further, (10) can be expressed as $\mu < \sqrt{\frac{\alpha_1(r)}{c_2}}$ for some $r > 0$. Thus, Theorem 1 ensures that for any initial condition satisfying $\|\xi(t_0)\| \leq \sqrt{\frac{\alpha_1(r)}{c_2}}$, the closed loop state ξ has an ultimate bound which satisfies (11), (12) and it is given as $\alpha_1^{-1}(c_2\mu^2)$. Note that, if we assume that ξ remains within a domain $S = \{\xi \in \mathbb{R}^5 : \|\xi\| < r\}$ and we allow only slow movements, i.e. $|\dot{x}^*(t)|, |\dot{y}^*(t)|, |\dot{x}^*(t)|, |\dot{y}^*(t)|$ are small, then μ is bounded because $\phi_e, \dot{\phi}_e, \ddot{\phi}_e, \dot{\phi}, \ddot{\phi}$ and $\dot{\phi}^*$ are bounded. Moreover, this also ensures that \dot{v}_d and v_d , in definition of s_3 , are bounded. Hence, $\mu > 0$ can be rendered arbitrarily small by choosing a large $\lambda_{\min}(Q_0)$, i.e. large values for $k_{p\phi}, k_{p1}, k_{p2}, k_{i1}, k_{i2}$ and large values for $k_{pv}, \lambda_{\min}(K_p)$ and $\lambda_{\min}(K_i)$ such that $\lambda_{\min}(K_p) > \max\{k_{p\phi}^2, k_{d\phi}^2, k_{pv}\}$ and $\lambda_{\min}(K_i) > \max\{k_{p\phi}^2, k_{d\phi}^2, k_{pv}\}$, provided that $\cos(\phi_e)$ is far from zero, i.e. ϕ_e is close to zero. Moreover, according to (29), (30), (32), it is also required that ratios k_{p2}/k_{p1} and k_{i2}/k_{i1} be kept constant as K_p y K_i are increased. Thus, the ultimate bound $\alpha_1^{-1}(c_2\mu^2)$ can be rendered arbitrarily small. Moreover, this and $\mu < \sqrt{\frac{\alpha_1(r)}{c_2}}$ as well as $\|\xi(t_0)\| \leq \sqrt{\frac{\alpha_1(r)}{c_2}}$ ensure that given any finite $r > 0$ there always exists a

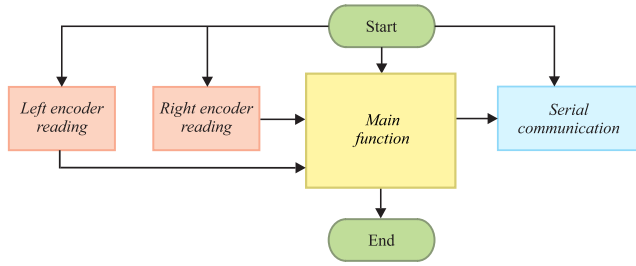


FIGURE 9. General flowchart.

small enough initial condition such that this result stands. This completes the proof of Proposition 1.

C. PROGRAM DEVELOPED TO EXECUTE THE CONTROL ALGORITHM IN THE EMBEDDED SYSTEM

In order to implement the trajectory tracking control algorithm (19) in the embedded system (Jetson TK1), a program was developed in Python. This programming language was chosen because is powerful and easy to learn [36], compared with other languages such as C or Java. The developed program has four functions: a) *Right encoder reading*, b) *left encoder reading*, c) *main*, and d) *serial communication*, which are represented in Fig. 9. The functions a and b obtain from the encoders the information related with the position of the wheels of the WMR prototype, the derivative of this position is computing by using Euler method to estimate v_1 and v_2 , respectively. In the c function the algorithm to generate the desired trajectory, introduced in Section IV, is executed. Then, the trajectory tracking control algorithm (19) is computed. Since the experimental instrumentation of τ requires an electrical conditioning, c function also transforms the torques τ_r and τ_l given by (19) in continuous voltages through the relation:

$$\begin{pmatrix} u_r \\ u_l \end{pmatrix} = \frac{R_a}{k_m} \begin{pmatrix} \tau_r \\ \tau_l \end{pmatrix},$$

where u_r and u_l are the voltages for the right and left DC motors of the WMR prototype, respectively, R_a is the armature resistance, and k_m is the torque constant of the DC motors. Note that u_r and u_l can take values in the intervals $[-u_{r\max}, +u_{r\max}]$ and $[-u_{l\max}, +u_{l\max}]$, with $u_{r\max}$ and $u_{l\max}$ being the maximum voltages of the right and left DC motors, respectively. For purposes of implementing u_r and u_l via a PWM, that is, obtaining the switched voltages u_{rc} and u_{lc} , the corresponding average voltages u_{rav} and u_{lav} are obtained by using

$$\begin{pmatrix} u_{rav} \\ u_{lav} \end{pmatrix} = \begin{pmatrix} \left| \frac{u_r}{u_{r\max}} \right| \\ \left| \frac{u_l}{u_{l\max}} \right| \end{pmatrix}, \quad (37)$$

where these voltages can take values in the interval $[0, 1]$ and are associated with the duty cycles of the PWM signals. In order to determine the directions, dir_r and dir_l , of u_r and

u_l , respectively, and consequently the direction of the DC motors, the following conditions are considered:

$$dir_r = \begin{cases} 0 & \text{for } u_r \geq 0, \\ 1 & \text{for } u_r < 0, \end{cases} \quad (38)$$

$$dir_l = \begin{cases} 0 & \text{for } u_l \geq 0, \\ 1 & \text{for } u_l < 0. \end{cases} \quad (39)$$

Lastly, the *communication serial* function send values obtained from (37) to the Arduino UNO board through RS-232 protocol. This latter computes (38) and (39).

Is important to mention that the four described functions are programmed in the Jetson TK1 board as subprocess. This is aided with the *multiprocessing* package by Python, which allows process-based parallelism and hence leveraging multiple processors. Thus, the functions are simultaneously executed in the four cores of the Jetson TK1 board processor.

With the app described in Section III and the computing program here described, the intuitive planning, generation, and tracking of trajectory for a WMR are accomplished, as can be corroborated in next section.

VI. EXPERIMENTAL TESTS

Experimental results obtained from the technological integration are shown in this section for two different sketches. Before that, a briefly description of the gain tuning of the proposed trajectory tracking control algorithm (19) is given.

A. GAIN TUNING OF THE CONTROL ALGORITHM

For tuning the gains of the trajectory tracking control algorithm (19), an automatic procedure based on parallel computing was used [37]. This procedure consists mainly in four steps: i) By taking into account the tuning rule and the number of control gains, a finite set of numerical values are generated and grouped in arrays through a Matlab script. In this case, large values for $k_{p\phi}$, $k_{d\phi}$, k_{pv} , k_{p1} , k_{p2} , k_{i1} , k_{i2} with $k_{p1}/k_{p2} = 1$ and $k_{i1}/k_{i2} = 1$ were considered. Hence, a large number of combinations to select such gains is obtained. ii) With these combinations, numerical simulations of the system in closed-loop are simultaneously performed through Matlab Parallel Computing Toolbox. iii) From the several obtained system responses, the ones achieving the control objective with a tracking error equal or less than 0.1 [m] are selected, i.e. $(\chi^* - \chi) \leq 0.1$. iv) Lastly, the gain combination that delivers a control response with the smallest error is identified.

B. EXPERIMENTAL RESULTS

For the experimental tests of the intuitive and simple platform herein presented to wirelessly plan, generate, and track trajectories with a WMR, through a technological integration of hardware and software, the sketches in Figs. 3 and 10 were considered.

For both sketches, the trajectory planning was performed with the app, as described in Section III, by choosing $v_p = 0.3$ m/s. After planning the desired trajectory for the WMR and sending it wirelessly to the embedded system, the

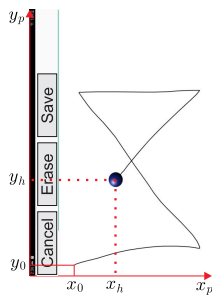


FIGURE 10. Sketch 2.

trajectory generation algorithm (introduced in Section IV) automatically generated the desired trajectory for the WMR. Subsequently, the trajectory tracking control algorithm, proposed in Section V, carried out the trajectory tracking task as shown in Figs. 11 and 12. The physical parameters of the WMR prototype and control gains required for the execution of the trajectory tracking algorithm are given below.

The radius of each wheel and the half of the width of the WMR were measured directly in the prototype, obtaining:

$$r = 0.075 \text{ m}, \quad b = 0.19 \text{ m},$$

whereas the control gains used were:

$$\begin{aligned} k_p\phi &= 11, & k_d\phi &= 0.1, & k_{p1} &= 0.2, & k_{p2} &= 0.2, \\ k_{i1} &= 0.1, & k_{i2} &= 0.1, & k_{pv} &= 6. \end{aligned}$$

These gains were delivered by the procedure described in the previous section.

With the intention of showing the robustness of the proposed control algorithm, consequently the robustness of the platform, in the experiments, the desired trajectory was initialized at $(x^*, y^*) = (0.3, 0.07)$ [m] and $\phi^* = 0.22$ [rad/s] for the sketch in Fig. 3, whereas $(x^*, y^*) = (0.07, 0.08)$ [m] and $\phi^* = 0.77$ [rad/s] where considered for the sketch in Fig. 10. For both sketches, the initial position and orientation of the WMR prototype were set at $(x, y) = (0, 0)$ and $\phi = 0$. While executing the experiments, the angular velocities of the wheels v_1 and v_2 , estimated with the sensors described in Section II, and the kinematic model (16) of the WMR were used to obtain the real-time estimation of (x, y, ϕ) by computer integration.

The corresponding experimental results for the sketches shown in the Figs. 3 and 10 are presented in Figs. 11 and 12, respectively. There, it can be seen that: *i)* The desired trajectory χ^* has been well planned and generated through the technological integration. See for example Figs. 11(c), 11(d) and Figs. 6(a), 6(b) in Section IV. *ii)* The proposed control algorithm successfully executes the trajectory tracking task, i.e. $\chi \rightarrow \chi^*$, even when the WMR is far from the initial position of the desired trajectory. This can be corroborated with Figs. 11(b) and 12(b), where the WMR prototype located far from the initial desired position reaches such a position and tracks the desired trajectory with an error around zero.

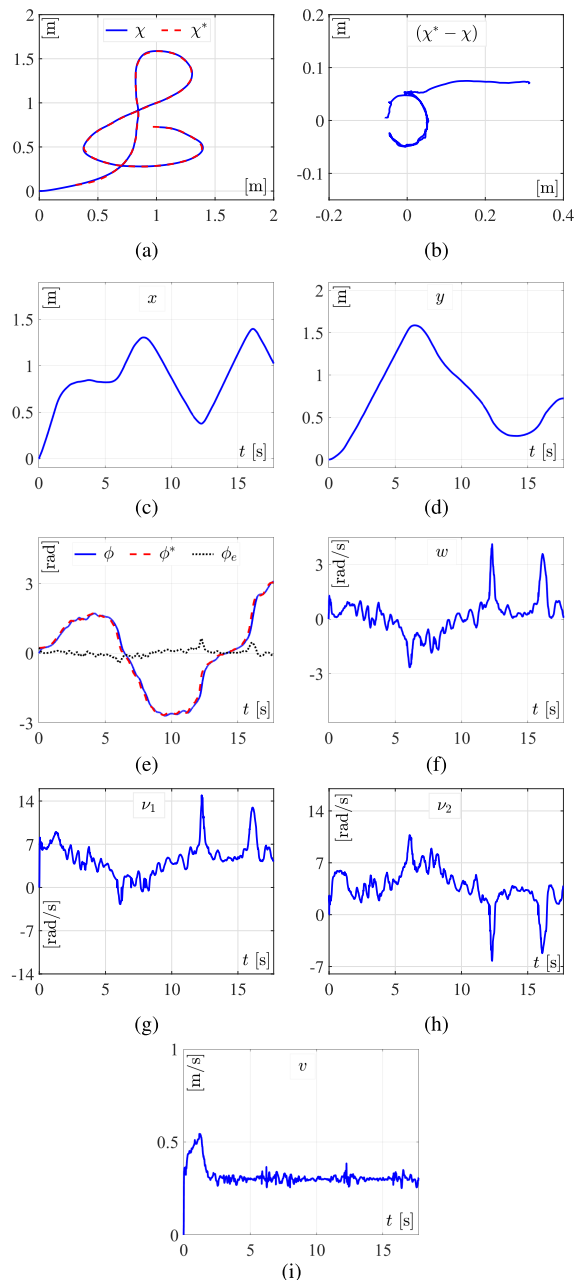


FIGURE 11. Experimental results for the sketch in Fig. 3.

After experiments, an usability test of the technological integration of hardware and software –herein introduced– was carried out with high school and college students. All of them were visitors to the Mechatronics Laboratory at CIDETEC-IPN. These students were separate in groups of fifteen. The test consisted on three steps: *1)* Only mention to the groups that the WMR prototype can perform any trajectory drawn by hand in the tablet and a practical example was shown. Then, the students were allowed to use the technological platform. *2)* Explain to students the mathematical and programming process involved in the technological platform to perform the planning, generation, and tracking of trajectories. Also, it was described how the planning,

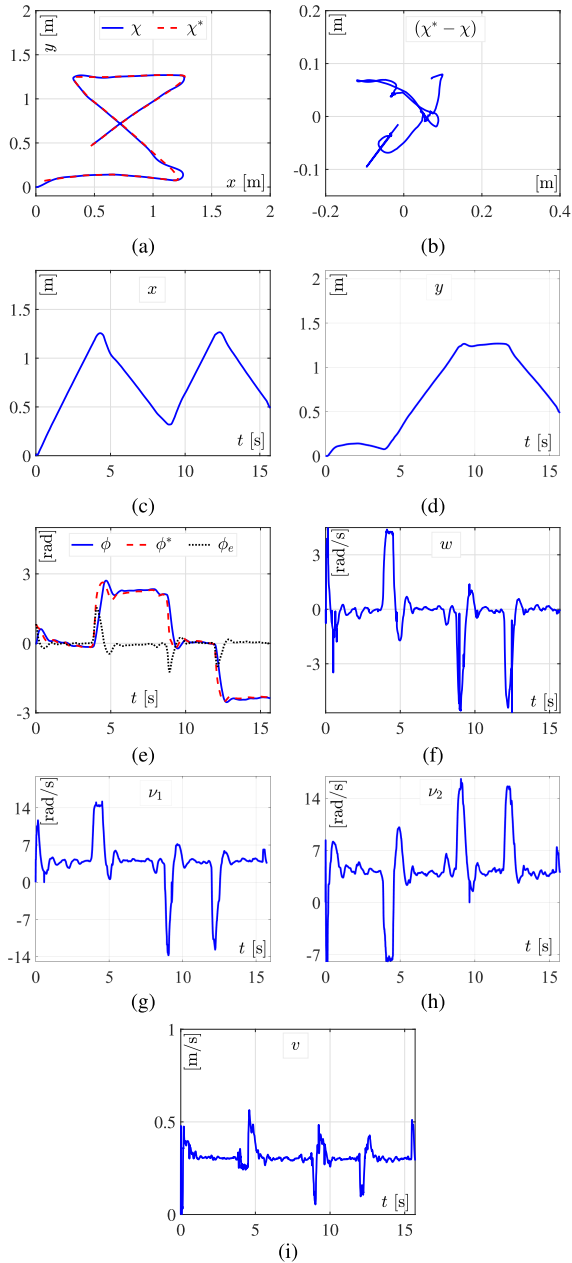


FIGURE 12. Experimental results for the sketch in Fig. 10.

generation, and tracking of trajectories can be achieved with a PC and specialized software. 3) Listen to opinions from the students about their experience when using the technological integration. Conclusion of these opinions was that all the students were amazed when realized of all the mathematical and programming functions contained in the tablet and embedded system. Also, they were overwhelmed with all the mathematics involved. Thus, all students expressed that with the tablet and the developed app and algorithms is better and easier to control the WMR prototype. This compared with the programming of the planning, generation, and control algorithms for the tracking task with a PC and specialized software, every time a new experiment is performed by the WMR

prototype. This usability test corroborates that the introduced technological integration of hardware and software delivers an intuitive and simple platform to plan, generate, and track trajectories with a WMR.

It is important to remark that, to the authors' knowledge, there is not a technological platform reported in literature (see [8]–[19]) that can achieve the trajectory tracking task with a WMR through a mobile computing device. Thus, a proper comparison between the proposed technological integration and one reported in literature cannot be carried out. Nevertheless, in order to evaluate only the performance of the trajectory tracking control algorithm herein proposed, a comparative assessment between this proposed control and two controls taken from literature is presented in Appendix.

VII. CONCLUSION

This paper has presented a technological integration of hardware and software, which delivers an intuitive and simple platform to perform the wireless planning, generation, and tracking of trajectories with a WMR prototype. For the trajectory planning, an app that exploits the hand-drawn sketch approach was developed by authors. Whereas a trajectory generation algorithm based on combination of Bézier polynomials and concatenation of straight-lines was proposed. Lastly, a trajectory tracking control algorithm was proposed to accomplish the tracking task, which was obtained departing from the dynamic model of the differential drive WMR. The technological platform here presented has three advantages: *i)* The trajectory planning app, by means of the sketch approach, and the wireless connection between the tablet and the Jetson TK1 board allow an intuitive manual planning of a desired trajectory for the WMR prototype in the real workspace. *ii)* The Bézier polynomials and concatenation of straight-lines allow to smoothly interconnect the points of the geometrically scaled coordinates of a sketch representing the desired trajectory. Therefore, a continuous desired trajectory is generated for the WMR prototype with less computing time consumption. *iii)* The proposed trajectory tracking control algorithm is simple and easy to experimentally implement. Thus, this paper has presented an useful technological integration of hardware and software, delivering a platform that could be introduced in industry activities. This mainly because of two reasons: *a)* it includes a simple and robust control algorithm that ensures to achieve the trajectory tracking task. *b)* it can be easily operated through the mobile computing device.

APPENDIX COMPARATIVE ASSESSMENT OF THE PROPOSED CONTROL

A comparative assessment of the performance of the trajectory tracking control algorithm proposed in Section V is given here. This control algorithm is compared with two tracking controls taken from literature. One of them was designed from the kinematic model of the WMR and reported in [27]. The other one is based on the dynamic model of the WMR and

was introduced in [38]. These two controls operate differently to the proposed control algorithm (19). Their objective is to achieve that a WMR tracks a desired trajectory imposed by a reference WMR, which is defined as follows:

$$\begin{aligned} \dot{x}^* &= v_r \cos \phi^*, \\ \dot{y}^* &= v_r \sin \phi^*, \\ \dot{\phi}^* &= \omega_r, \end{aligned} \quad (40)$$

where x^* , y^* , and ϕ^* are the desired trajectory and v_r and ω_r are the reference inputs. Also, the tracking control reported in [27] refers to the linear and angular velocities of the WMR, defined as:

$$\begin{aligned} v &= v_r \cos e_3 + K_1 e_1, \\ w &= \omega_r + v_r K_2 e_2 + K_3 \sin e_3, \end{aligned} \quad (41)$$

where v and w stand as defined in (17), but with v_1 and v_2 non saturated, K_1 , K_2 , K_3 are positive constants, and the tracking error is given by:

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} K_2 \cos \phi & K_2 \sin \phi & 0 \\ -K_2 \sin \phi & K_2 \cos \phi & \frac{1}{K_3} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^* - x \\ y^* - y \\ \phi^* - \phi \end{pmatrix}, \quad (42)$$

whereas the tracking control in [38] represents torque defined as follows:

$$\tau_a = M(q) \left(\dot{v}_a^* - K_{pa} \tilde{v}_a - K_{ia} \int_0^t \tilde{v}_a ds \right) + C(q, \dot{q}) v_a, \quad (43)$$

with $M(q)$ and $C(q, \dot{q})$ as defined in (15),

$$\begin{aligned} \tilde{v}_a &= v_a - v_a^* = \begin{pmatrix} v_{1a} - v_{1a}^* \\ v_{2a} - v_{2a}^* \end{pmatrix}, \\ K_{pa} &= \text{diag} \{k_{pv_a}, k_{p\omega}\}, \\ K_{ia} &= \text{diag} \{k_{iv}, k_{i\omega}\}, \\ \begin{pmatrix} v_{1a}^* \\ v_{2a}^* \end{pmatrix} &= \begin{pmatrix} \frac{1}{r} & \frac{l}{r} \\ \frac{1}{r} & -\frac{l}{r} \end{pmatrix} \begin{pmatrix} v_a \\ w_a \end{pmatrix}, \\ v_a &= v_r \cos e_{3a} + K_{1a} e_{1a}, \\ w_a &= \omega_r + v_r K_{2a} e_{2a} + K_{3a} \sin e_{3a}, \end{aligned}$$

K_{pa} , K_{ia} , K_{1a} , K_{2a} , K_{3a} being positive constants and its corresponding tracking error being:

$$\begin{pmatrix} e_{1a} \\ e_{2a} \\ e_{3a} \end{pmatrix} = \begin{pmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^* - x \\ y^* - y \\ \phi^* - \phi \end{pmatrix}. \quad (44)$$

Thus, the trajectory planning app and the trajectory generation algorithm presented in Sections III and IV, respectively, cannot be directly applied for the tracking controls (41) and (43).

Since the redesign of the trajectory planning app and the trajectory generation algorithm for other controls is beyond of the scope of this study, for the comparative assessment

the trajectory planning and trajectory generation were carried out directly in the embedded system of the WMR prototype described in Section II. The desired trajectory used to test the tracking controls (41) and (43) is defined in Table 1. There, in each interval of the time, t , where applies, v_r and ω_r , are determined by the following polynomials:

$$v_r(t) = \psi(t, t_i, t_f) (\bar{v}_f - \bar{v}_i) + \bar{v}_i, \quad (45)$$

$$\omega_r(t) = \psi(t, t_i, t_f) (\bar{\omega}_f - \bar{\omega}_i) + \bar{\omega}_i, \quad (46)$$

where

$$\psi(t, t_i, t_f) = \left(\frac{t - t_i}{t_f - t_i} \right)^3 \left[r_1 - r_2 \left(\frac{t - t_i}{t_f - t_i} \right) + r_3 \left(\frac{t - t_i}{t_f - t_i} \right)^2 \right], \quad (47)$$

and \bar{v}_i , \bar{v}_f are constant linear velocities between which v_r smoothly interpolates in the time interval $[t_i, t_f]$, with t_i being the initial time and t_f the final one. Similarly, $\bar{\omega}_i$ and $\bar{\omega}_f$ are constant angular velocities between which ω_r smoothly interpolates in the time interval $[t_i, t_f]$. Whereas the coefficients r_1 , r_2 , and r_3 were chosen as in [39]:

$$r_1 = 10, \quad r_2 = 15, \quad r_3 = 6.$$

It is important to mention that, in the times t_i and t_f , (45) and (46) satisfy: $[v_r(t_i), v_r(t_f)] = [\bar{v}_i, \bar{v}_f]$ and $[\omega_r(t_i), \omega_r(t_f)] = [\bar{\omega}_i, \bar{\omega}_f]$.

To test the tracking control algorithm (19) herein proposed, the desired trajectory translated to the plane XY is required. Hence, such a trajectory is generated by introducing v_r and ω_r of the Table 1 in (40). The resulting desired trajectory is shown in Fig. 13(a).

For the real-time implementation of the proposed tracking controller (19), only the following WMR parameters are required:

$$r = 0.075 \text{ m}, \quad b = 0.19 \text{ m},$$

which are the same needed for the control (41); whereas the control (43) needs the following WMR parameters:

$$\begin{aligned} r &= 0.0755 \text{ m}, \quad b = 0.19 \text{ m}, \\ d &= 0.01 \text{ m}, \quad m_c = 19.7 \text{ kg}, \\ m_w &= 1.732 \text{ kg}, \quad m = 23.164 \text{ kg}, \\ I_w &= 0.004 \text{ kg} \cdot \text{m}^2, \quad I_m = 0.001 \text{ kg} \cdot \text{m}^2, \\ I_c &= 0.135 \text{ kg} \cdot \text{m}^2, \quad I = 0.265 \text{ kg} \cdot \text{m}^2. \end{aligned}$$

In the experiments, same initial conditions were used for the three controls, that is:

$$x(0) = -0.2, \quad y(0) = -0.05, \quad \phi(0) = 0.$$

The gains for the control (41) were chosen as:

$$K_1 = 0.2, \quad K_2 = 10, \quad K_3 = 7.$$

While, the gains for the control (43) were selected as:

$$K_{1a} = 0.2, \quad K_{2a} = 10, \quad K_{3a} = 7,$$

TABLE 1. Reference velocities, v_r and ω_r .

	$v_r(t)$	$\omega_r(t)$
$0 \leq t \leq 0.4$	$\bar{v}_i(t_i=0) = 0.1;$ $\bar{v}_f(t_f=0.4) = 0.4;$	$\omega_r = 0;$
$0.4 \leq t \leq 3$	$v_r = 0.4;$	$\omega_r = 0;$
$3 \leq t \leq 3.4$	$v_r(t)$	$\omega_r(t)$
	$\bar{v}_i(t_i=3) = 0.4;$ $\bar{v}_f(t_f=3.4) = 0.3;$	$\bar{\omega}_i(t_i=3) = 0;$ $\bar{\omega}_f(t_f=3.4) = 1.5;$
$3.4 \leq t \leq 5.09$	$v_r = 0.3;$	$\omega_r = 1.5;$
$5.09 \leq t \leq 5.49$	$v_r(t)$	$\omega_r(t)$
	$\bar{v}_i(t_i=5.09) = 0.3;$ $\bar{v}_f(t_f=5.49) = 0.4;$	$\bar{\omega}_i(t_i=5.09) = 1.5;$ $\bar{\omega}_f(t_f=5.49) = 0;$
$5.49 \leq t \leq 7.7$	$v_r = 0.4;$	$\omega_r = 0;$
$7.7 \leq t \leq 8.1$	$v_r(t)$	$\omega_r(t)$
	$\bar{v}_i(t_i=7.7) = 0.4;$ $\bar{v}_f(t_f=8.1) = 0.3;$	$\bar{\omega}_i(t_i=7.7) = 0;$ $\bar{\omega}_f(t_f=8.1) = -1.5;$
$8.1 \leq t \leq 9.79$	$v_r = 0.3;$	$\omega_r = -1.5;$
$9.79 \leq t \leq 10.19$	$v_r(t)$	$\omega_r(t)$
	$\bar{v}_i(t_i=9.79) = 0.3;$ $\bar{v}_f(t_f=10.19) = 0.4;$	$\bar{\omega}_i(t_i=9.79) = -1.5;$ $\bar{\omega}_f(t_f=10.19) = 0;$
$10.19 \leq t \leq 12.4$	$v_r = 0.4;$	$\omega_r = 0;$
$12.4 \leq t \leq 12.8$	$v_r(t)$	$\omega_r(t)$
	$\bar{v}_i(t_i=12.4) = 0.4;$ $\bar{v}_f(t_f=12.8) = 0.3;$	$\bar{\omega}_i(t_i=12.4) = 0;$ $\bar{\omega}_f(t_f=12.8) = 1.5;$
$12.8 \leq t \leq 15.05$	$v_r = 0.3;$	$\omega_r = 1.5;$
$15.05 \leq t \leq 15.45$	$v_r(t)$	$\omega_r(t)$
	$\bar{v}_i(t_i=15.05) = 0.3;$ $\bar{v}_f(t_f=15.45) = 0.4;$	$\bar{\omega}_i(t_i=15.05) = 1.5;$ $\bar{\omega}_f(t_f=15.45) = 0;$
$15.45 \leq t \leq 18.8$	$v_r = 0.4;$	$\omega_r = 0.$

$$k_{pv_a} = 2, \quad k_{p\omega} = 2, \quad k_{i\nu} = 50, \quad k_{i\omega} = 50.$$

Lastly, the gains for the proposed control (19) were set as follows:

$$k_{p\phi} = 11, \quad k_{d\phi} = 0.1, \quad k_{p1} = 0.1, \quad k_{p2} = 0.1, \\ k_{i1} = 0.1, \quad k_{i2} = 0.1, \quad k_{pv} = 4.$$

The results obtained from the three controls are shown in Fig. 13. To differentiate the variables of the three controls, subscripts *PC*, *C1*, and *C2* were used for the proposed control (19), control (41), and control (43), respectively, where was needed. Because of the difference in the operation of the three controls, a comparison is carried out but only considering their performance when tracking the desired trajectory with respect to the plane *xy*. In Fig. 13(a), a similar trajectory tracking performance is observed for the three controls. However, there is a main difference on the beginning of the experiment. Note that, the proposed control (19) tracks the desired trajectory from the beginning and the other controls do not. In terms of the trajectory tracking error, the proposed control has minor error than the control (41), but has a larger error than the control (43). Although, it can be thought that control (43) has better performance than the

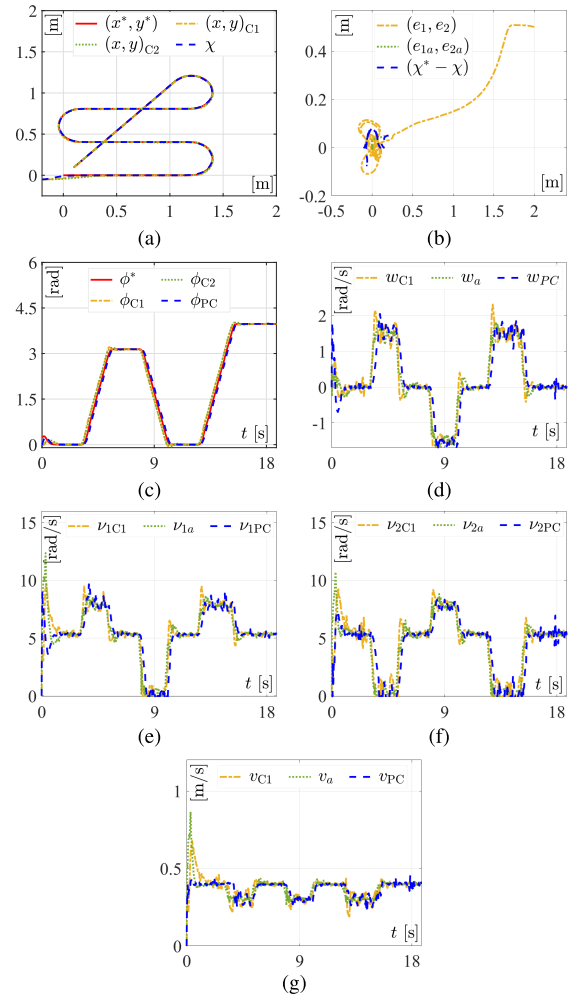


FIGURE 13. Experimental results of the proposed control compared with two from literature.

proposed control (19), it is important to note that the proposed control does not require physical parameters of the WMR prototype associated with masses and inertias. Therefore, it can be stated that the proposed control is also robust under uncertain parameters. Another important characteristic of the proposed control, observed in Fig. 13, is that the maximum velocities ν_1 , ν_2 , and ν are the smallest when abrupt changes appear in the desired trajectory. This implies that the WMR prototype moves smoothly along all the desired trajectory, when performing the tracking task. This is a very important behavior to pursuit industrial applications.

Finally, it is worth mentioning that a performance comparison between the Jetson TK1 board and the DS1104 board from dSPACE, when executing complex trajectory tracking controls, was reported in [40]. There, it was concluded that the Jetson TK1 board has a performance similar to the DS1104 board.

REFERENCES

- [1] Y. F. Zheng and J. Y. S. Luh, "Optimal load distribution for two industrial robots handling a single object," *J. Dyn. Syst., Meas., Control*, vol. 111, no. 2, pp. 232–237, Jun. 1989.

- [2] W.-D. Jeng, J. T. Lin, and U.-P. Wen, "Algorithms for sequencing robot activities in a robot-centered parallel-processor workcell," *Comput. Oper. Res.*, vol. 20, no. 2, pp. 185–197, Feb. 1993.
- [3] Y. P. Aneja and H. Kamoun, "Scheduling of parts and robot activities in a two machine robotic cell," *Comput. Oper. Res.*, vol. 26, no. 4, pp. 297–312, Apr. 1999.
- [4] S. Wang, C. Zhang, C. Liu, D. Li, and H. Tang, "Cloud-assisted interaction and negotiation of industrial robots for the smart factory," *Comput. Electr. Eng.*, vol. 63, pp. 66–78, Oct. 2017.
- [5] Y. Liu and Y. Zhang, "Toward welding robot with human knowledge: A remotely-controlled approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 769–774, Apr. 2015.
- [6] A. Macchelli and C. Melchiorri, "A real-time control system for industrial robots and control applications based on real-time Linux," *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 55–60, 2002.
- [7] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. V. Mil, J. van Egmond, R. Burger, M. Morariu, J. Ju, X. Gerrmann, R. Ensing, J. V. Frankenhuyzen, and M. Wisse, "Team delft's robot winner of the Amazon picking challenge 2016," in *RoboCup: Robot World Cup XX*, S. Behnke, R. Sheh, S. Sariel, and D. D. Lee, Eds. Cham, Switzerland: Springer, 2017, pp. 613–624.
- [8] R. Chatterjee and F. Matsuno, "Learning obstacle avoidance reflex behavior for autonomous navigation from hand-drawn trajectories," in *Proc. IEEE Int. Conf. Ind. Technol.*, Goa, India, Jan. 2000, pp. 58–63.
- [9] M. Skubic, P. Matsakis, B. Forrester, and G. Chronis, "Extracting navigation states from a hand-drawn map," in *Proc. ICRA. IEEE Int. Conf. Robot. Autom.*, Seoul, South Korea, May 2001, pp. 259–264.
- [10] M. Skubic, C. Bailey, and G. Chronis, "A sketch interface for mobile robots," in *Proc. IEEE Int. Conf. Syst., Man Cybern. Conf. Theme-Syst. Secur. Assurance (SMC)*, Washington, DC, USA, Oct. 2003, pp. 919–924.
- [11] G. Chronis and M. Skubic, "Sketch-based navigation for mobile robots," in *Proc. 12th IEEE Int. Conf. Fuzzy Syst. (FUZZ)*, St Louis, MO, USA, May 2003, pp. 284–289.
- [12] G. Chronis and M. Skubic, "Robot navigation using qualitative landmark states from sketched route maps," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, USA, Apr./May 2004, pp. 1530–1535.
- [13] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz, "Using a qualitative sketch to control a team of robots," *Auto. Robots*, vol. 22, no. 4, pp. 399–410, 2007.
- [14] D. Sakamoto, K. Honda, M. Inami, and T. Igarashi, "Sketch and run: A stroke-based interface for home robots," in *Proc. 27th SIGCHI Conf. Hum. Factors Comput. Syst.*, vols. 1–4. Boston, MA, USA, Apr. 2009, pp. 197–200.
- [15] D. Shah, J. Schneider, and M. Campbell, "A sketch interface for robust and natural robot control," *Proc. IEEE*, vol. 100, no. 3, pp. 604–622, Mar. 2012.
- [16] S. Mook Jung, D. Choi, K. Kwon, and J. Wook Jeon, "A sketch-based interface for remote robot control on an online video screen," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2013, pp. 428–429.
- [17] F. Boniardi, B. Behzadian, W. Burgard, and G. D. Tipaldi, "Robot navigation in hand-drawn sketched maps," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Lincoln, U.K., Sep. 2015, pp. 1–6.
- [18] F. Boniardi, A. Valada, W. Burgard, and G. D. Tipaldi, "Autonomous indoor robot navigation using a sketch interface for drawing maps and routes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 2896–2901.
- [19] M. Mielle, M. Magnusson, and A. J. Lilienthal, "Using sketch-maps for robot navigation: Interpretation and matching," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Lausanne, Switzerland, Oct. 2016, pp. 252–257.
- [20] M. Negulescu and T. Inamura, "Exploring sketching for robot collaboration," in *Proc. 6th Int. Conf. Hum.-Robot Interact. (HRI)*, Lausanne, Switzerland, 2011, pp. 211–212.
- [21] B. Samakrob, S. Auephanitayakul, and N. Theera-Umpon, "Extracting route information from hand-drawn map using fuzzy vector," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Brisbane, QLD, Australia, Jun. 2012, pp. 1–7.
- [22] C. Marquez-Sanchez, R. Silva-Ortigoza, M. Antonio-Cruz, C. Y. Sosa-Cervantes, and V. M. Hernandez-Guzman, "A review of embedded systems used in WMR for the trajectory tracking task," in *Proc. Int. Conf. Mechatronics, Electron. Automot. Eng. (ICMEAE)*, Cuernavaca, Mexico, Nov. 2015, pp. 145–150.
- [23] J. R. G. Sanchez, S. T. Mosqueda, R. S. Ortigoza, M. A. Cruz, G. S. Ortigoza, and J. de J. Rubio, "Assessment of an average tracking controller that considers all the subsystems involved in a WMR: Implementation via PWM or sigma-delta modulation," *IEEE Latin Amer. Trans.*, vol. 14, no. 3, pp. 1093–1102, Mar. 2016.
- [24] A. S. Al-Araji, "Development of kinematic path-tracking controller design for real mobile robot via back-stepping slice genetic robust algorithm technique," *Arabian J. Sci. Eng.*, vol. 39, no. 12, pp. 8825–8835, Dec. 2014.
- [25] S. G. Tzafestas, K. M. Deliparaschos, and G. P. Moustiris, "Fuzzy logic path tracking control for autonomous non-holonomic mobile robots: Design of system on a chip," *Robot. Auto. Syst.*, vol. 58, no. 8, pp. 1017–1027, Aug. 2010.
- [26] R. D'Andrea, "Guest Editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 4, pp. 638–639, 2012.
- [27] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 609–615, Oct. 2000.
- [28] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, U.K.: Springer-Verlag, 2009.
- [29] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019.
- [30] Statcounter. *Operating System Market Share Worldwide*. Accessed: Jul. 14, 2020. [Online]. Available: <https://gs.statcounter.com/os-market-share>
- [31] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, p. 3170, 2018.
- [32] M. Paluszny, H. Prautzsch, and W. Boehm, *Bézier and BSpline Techniques*. Baden-Wurtemberg, German: Springer-Verlag, 2002.
- [33] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [34] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 1993.
- [35] R. Kelly, V. Santibáñez, and A. Loria, *Control of Robot Manipulators in Joint Space*. London, U.K.: Springer, 2005.
- [36] J. M. Perkel, "Pick up python," *Nature*, vol. 518, no. 7537, pp. 125–126, 2015.
- [37] M. A. Cruz, R. S. Ortigoza, C. M. Sánchez, V. M. H. Guzmán, J. S. Gutiérrez, and J. C. H. Lozada, "Parallel computing as a tool for tuning the gains of automatic control laws," *IEEE Latin Amer. Trans.*, vol. 15, no. 6, pp. 1189–1196, Jun. 2017.
- [38] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics," in *Proc. 34th IEEE Conf. Decis. Control*, vol. 4, New Orleans, LA, USA, 1995, pp. 3805–3810.
- [39] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Berlin, Germany: Springer-Verlag, 2008.
- [40] C. M. Sánchez, R. S. Ortigoza, J. R. G. Sánchez, V. M. H. Guzmán, M. A. Cruz, M. M. Aranda, and G. S. Ortigoza, "An embedded hardware for implementation of a tracking control in WMRs," *IEEE Latin Amer. Trans.*, vol. 16, no. 7, pp. 1835–1842, Jul. 2018.



CELSO MÁRQUEZ-SÁNCHEZ received the B.S. degree in computer engineering from the Universidad Autónoma del Estado de México, Mexico City, Mexico, in 2008, and the M.S. degree in computing technology and the Ph.D. degree in engineering of robotic and mechatronic systems from CIDETEC-IPN, Mexico City, in 2014 and 2018, respectively.

He is currently a Professor and a Researcher with the Universidad Politécnica de Texcoco.

He has published articles in journals indexed in JCR. He has also presented papers in international conferences. His research interests include experimental implementation of automatic control, mobile robots, mechatronic systems, embedded systems, and free software. He received the Best Academic Performance Award in cycle of schooling from 2013 to 2014, the Distinction to Polytechnic Merit: Presea Lázaro Cárdenas in Physical and Mathematical Sciences and Engineering Area at the master's level in 2015, and the Best Postgraduate Thesis Award at the master's level.



VICTOR MANUEL HERNÁNDEZ-GUZMÁN (Member, IEEE) was born in Querétaro, Mexico. He received the B.S. degree in electrical engineering from the Instituto Tecnológico de Querétaro, Querétaro, in 1988, the M.S. degree in electrical engineering from the Instituto Tecnológico de la Laguna, Torreón, Mexico, in 1991, and the Ph.D. degree in electrical engineering from CINVESTAV-IPN, Mexico City, Mexico, in 2003.

He has been a Professor with the Universidad Autónoma de Querétaro, Querétaro, since 1995, where he teaches classical and nonlinear control. He has published over 45 articles in refereed journals. He has coauthored the book *Automatic Control: Design Theory, Prototype Construction, Modeling, Identification and Experimental Tests* (Spanish) (Mexico City, Mexico: Colección CIDETEC-IPN, 2013) and the book *Automatic Control with Experiments* (Cham, Switzerland: Springer, 2019). His research interests include control of mechatronic systems, mobile robots, electro mechanical systems, prototype construction to teach classical and nonlinear control techniques, and control of different classes of AC electric motors when actuating on complex nonlinear mechanical loads.



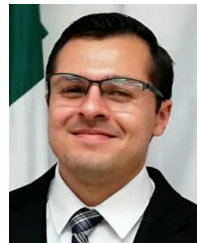
RAMÓN SILVA-ORTIGOZA (Member, IEEE) received the B.S. degree in electronics from Benemérita Universidad Autónoma de Puebla, Puebla, Mexico, in 1999, and the M.S. and Ph.D. degrees in electrical engineering (mechatronics) from the Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), Mexico City, Mexico, in 2002 and 2006, respectively.

He has been a Researcher with the Department of Mechatronics and Renewable Energy, Centro de Innovación y Desarrollo Tecnológico en Cómputo-IPN (CIDETEC-IPN), since 2006. He was also with SNI-CONACYT, Mexico. He has coauthored the book *Control Design Techniques in Power Electronics Devices* (London, U.K.: Springer-Verlag, 2006), the book *Automatic Control: Design Theory, Prototype Construction, Modeling, Identification and Experimental Tests* (Spanish) (Mexico City, Mexico: CIDETEC-IPN, 2013), and the book *Automatic Control with Experiments* (Cham, Switzerland: Springer, 2019). He has published over 60 articles in JCR indexed journals and three chapters in international books. He has presented over 40 papers in international conferences. He was an Advisor of over 25 postgraduate students and two B.S. students. He was a Leader in more than ten research projects. He has collaborated in nine additional research projects. His research work has been cited over 1000 times. His research interests include mechatronic control systems, mobile robotics, control in power electronics, and development of educational technology. He has been a referee in several awards of research and engineering, Mexico, the National Program of Quality Postgraduate, and research projects of CONACYT. Six of his students received the Presea Lázaro Cárdenas Award in 2012, 2015, 2016, 2018, 2019, and 2020, and the most important prize from the Instituto Politécnico Nacional, Mexico. Two of these students received The Best Master Thesis Award from the Instituto Politécnico Nacional, Mexico, in 2015 and 2017. He was an Editor of the book *Mechatronics* (Spanish) (Mexico City, Mexico: CIDETEC-IPN, 2010). He served as a Reviewer for several JCR indexed journals.



MAYRA ANTONIO-CRUZ received the B.S. degree in mechatronics from the Instituto Tecnológico de Reynosa, Mexico, in 2012, and the M.S. degree in computing technology and the Ph.D. degree in engineering of robotic and mechatronic systems from the Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional (CIDETEC-IPN), Mexico City, Mexico, in 2015 and 2018, respectively.

She is currently a Professor and a Researcher with the Sección de Estudios de Posgrado e Investigación de la Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas de IPN. She has published articles in journals indexed in JCR and presented papers in international conferences. Her current research interests include automatic control applications for underactuated mechanical systems, mobile robots, power electronic systems using information and communication technologies, and development of educational technology. She received the Best Academic Performance Award twice for the master's and Ph.D. levels, in 2015 and 2018, respectively, the Distinction to Polytechnic Merit: Presea Lázaro Cárdenas in Physical and Mathematical Sciences and Engineering Area at the master's level in 2016, and the Interdisciplinary Area at the Ph.D. level in 2019.



CARLOS ALEJANDRO MERLO-ZAPATA received the B.S. degree in mechatronics from the Instituto Tecnológico de Reynosa, Mexico, in 2012, and the master's degree in computing technology and the Ph.D. degree in engineering of robotic and mechatronic systems from the Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional (CIDETEC-IPN), in 2013 and 2019, respectively.

He is currently a Professor and a Researcher with the Instituto Tecnológico de Iztapalapa. He has published articles in journals indexed in JCR and presented papers in international conferences. His research interests include control applications for underactuated mechanical systems, such as inverted pendulums and unmanned aerial vehicles, design, programming, and instrumentation of robotic and mechatronic prototypes.

• • •