"**Construcción de un Robot de Tracción Omnidireccional para la Implementación de Algoritmos de Control Autónomo, Considerando Centros de Masa Desplazable para el Laboratorio de Investigación en Automatización Robótica y Sistemas Inteligentes**"

De La Cruz Vaca, Aida Liliana y Tapia Claudio, Edison Fernando

Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Maestría en Electrónica y Automatización, Mención en Redes Industriales

Artículo científico, previo a la obtención del título de Magíster en Electrónica y Automatización, Mención en Redes Industriales.

Ing. Víctor Hugo, Andaluz Ortiz, Ph.D.

19 de diciembre del 2022

Latacunga

Virtual Training System for the Autonomous Navigation of an Omnidirectional Traction Robot

De La Cruz Aida, Tapia Edison and Andaluz Víctor H.

Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador

{alde2,eftapia2,vhandaluz1}@espe.edu.ec

Abstract. This paper presents the development of a virtual environment for the training of autonomous omnidirectional drive vehicle control. The virtual system considers the virtualization of structured and unstructured environments. Therefore, the virtual environment considers mathematical models of the omnidirectional robot in order to simulate more realistically the behavior and motion constraints of the robot. The integration of the control schemes is considered in the MatLab software, for which a communication between the Unity3D graphic engine and MatLab is considered through the use of DLL libraries. For the validation of the control algorithms on the virtual training environment, the construction of an omnidirectional traction vehicle with Mecanum configuration. In addition, the constructed prototype will be used for the identification and validation of the mathematical models that represent its behavior. Finally, a usability analysis of the developed training system is presented.

Keywords: Omnidirectional robot, Virtual training, autonomous control, dynamic modeling.

1 Introducción

Currently, technological development has allowed advances in the area of robotics to be focused not only on the industrial area [1,2]. The latest developments are oriented to non-industrial applications, for example, mining, agriculture, security, construction, health, among others. Among the most developed robots are land, aerial and aquatic mobile robots [3,4]. Different applications that are usually performed by humans are nowadays developed by robotic platforms, e.g., domestic cleaning, crop spraying, traffic surveillance, among others [5,6].

Considering the different mechanisms of terrestrial mobile robots, it can be described*: i) unicycle robots*, have a mechanical structure of two wheels independently controlled by DC motors [7,8]; *ii) car-like robots*, are based on the Ackerman system with its linear velocity and angle of rotation [9,10]; and *iii) omnidirectional robots*, consisting of wheels with rollers that allow frontal, lateral and angular displacement [11,12]. Due to their mobility, omnidirectional robots have different applications, e.g., inspection of hazardous environments, cargo transportation, among others.

As described above, this work presents the implementation of a virtual training system for the control and autonomous navigation of an omnidirectional traction robot [13]. It considers the digitization of laboratory and industrial environments, with the purpose of executing load transfer tasks through mobile robots [11,14]. Mathematical modeling of an omnidirectional traction robot is determined, in order to be implemented in the virtual environment and in the development of control algorithms[11,15]. The proposed dynamic model considers as input signals two linear velocities and an angular velocity; in addition, it considers the displacement of the center of mass, which is caused by placing a displaced load on the robotic platform. In order to implement different control strategies, MatLab software is considered, which through the use of DLL libraries communicates in real time with Unity3D [16]. A cascade control scheme is proposed consisting of: kinematic controller and an adaptive dynamic compensation controller. To validate the mathematical models and evaluate the proposed control scheme, a Mecanum-type four-wheeled omnidirectional robotic prototype was built. Finally, a usability test was implemented to different engineering users in order to evaluate the developed virtual training system.

2 Training System Methodology

Figure 1 shows the methodology developed for the digitalization of a virtual training system for the autonomous control of an omnidirectional traction robot. The implemented methodology considers four main stages: construction of the mobile robot; mathematical modeling; digitization in the graphics engine; and design of advanced control algorithms.

(i) Construction, this stage considers the design and mechanical and electrical construction of an omnidirectional traction robotic prototype. For the construction of the robotic system, a Mecanum configuration of four wheels controlled by four independent motors coupled with encoder each motor is considered, in order to know the velocity and position of the robotic system. In addition, each motor has a driver that will allow controlling the velocity of the wheels, as well as the direction of rotation, thus separating the control and power stages respectively. A PID controller is implemented for each motor.

(ii) Mathematical Modeling, through the heuristic method, the mathematical models of the omnidirectional robot is determined, in order to represent the behavior and motion restriction of the mobile robot. Mathematical models will be incorporated in the virtual environment and in the control scheme. The dynamic parameters of the mobile robot are determined experimentally.

(iii) 3D Digitalization, the mobile robot and the virtual environment are modeled with CAD tools. Furthermore, elements are considered to simulate disturbances and different types of surfaces that affect the displacement of the omnidirectional robot. Then, using 3DS Max software, it is exported to Unity 3D software.

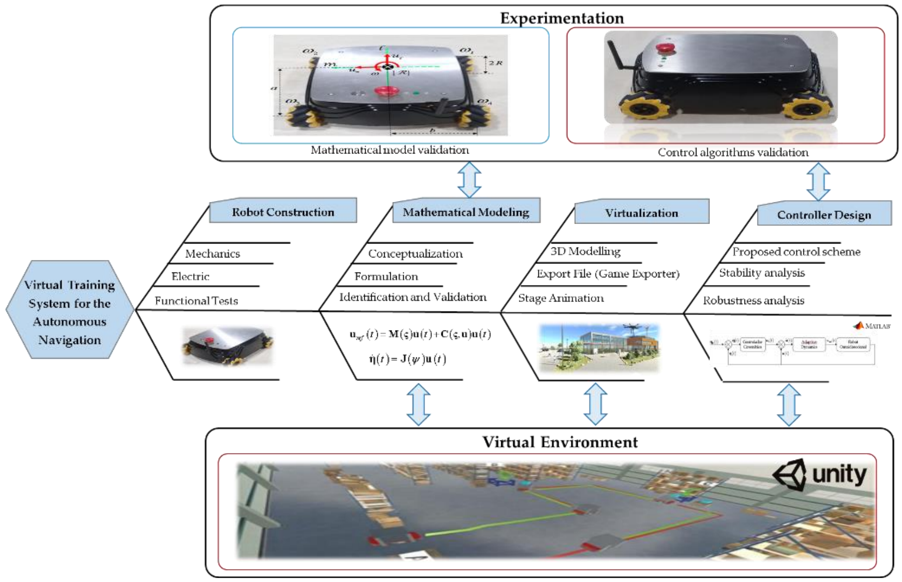


Fig. 1. Methodology for the virtual training system

(iv) Control Scheme Design, the main objective of the virtual environment is to be a theoretical-practical tool in the teaching-learning process, specifically in robotics for the design and evaluation of advanced control algorithms. In this work, a cascade adaptive control scheme. Finally, for the evaluation of the training system, simulation and experimental tests are considered*: (a) Simulation*, bilateral communication between MatLab and Unity3D software through the use of DLL libraries; and *(b) Experimentation,* experimental tests are considered for the identification and validation of the mathematical models and to evaluate the considering control algorithms in this paper. For both simulation and experimental tests, a sampling period $T_o = 100\ [\mathrm{ms}]$ is considered.

3 Mobile Robot with Omnidirectional Traction

This item, describes the mathematical models of the platform with omnidirectional traction, considering a Mecanum configuration. The mathematical models representing the behavior of the mobile platform are considered in the control scheme and in the 3D virtual environment.

3.1 Kinematic Modeling

The mobile platform with omnidirectional traction considered in this work has a mecanum configuration, as shown in Fig., 2. Where $\boldsymbol{\eta}(t)$ defines the position and orientation of the control point with respect to $\{\mathcal{R}\}$.

The kinematic model of the mobile platform with omnidirectional traction considered in this work is considering four velocities represented with respect to the moving reference system $\mathcal{R}(\ell, m, n)$. The movement of the platform is defined by three linear velocities u_l, u_m and u_n and an angular velocity ω rotating about the vertical axis of the moving reference system $\mathcal{R}(\ell, m, n)$.

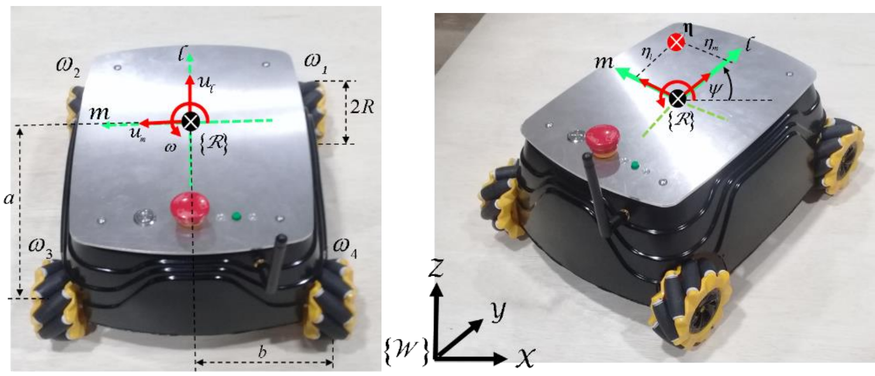


Fig. 2. Mobile robot with omnidirectional traction

Therefore, the motion of the mobile platform is defined as,

$$\begin{cases} \dot{\eta}_x = u_f \cos(\psi) - u_l \sin(\psi) - \omega\eta_l \sin(\psi) - \omega\eta_m \cos(\psi) \\ \dot{\eta}_y = u_f \sin(\psi) + u_l \cos(\psi) + \omega\eta_l \cos(\psi) - \omega\eta_m \sin(\psi) \\ \dot{\eta}_\psi = \omega \end{cases} \tag{1}$$

The (1) can be described as:

$$\dot{\boldsymbol{\eta}}(t) = \mathbf{J}(\psi)\mathbf{u}(t) \tag{2}$$

where, $\dot{\boldsymbol{\eta}}(t) \in R^n$ represents the velocity vector of the control point on $\{\mathcal{R}\}$; $\mathbf{J}(\psi) \in R^{mxn}$ with $m = n = 4$ is the Jacobian matrix of the mobile platform; and $\mathbf{u}(t) \in \Re^n$ is the maneuverability vector of the platform . The matrix $\mathbf{J}(\psi)$ is of full rankwith $|\mathbf{J}(\psi)| = 1$; therefore, there exists the inverse matrix of $\mathbf{J}(\psi)$. The maneuverability vector $\mathbf{u}(t)$ can be defined as,

$$\mathbf{u}(t) = \mathbf{J}^{-1}(\psi)\dot{\boldsymbol{\eta}}(t) \tag{3}$$

being $\mathbf{J}^{-1}(\psi) = \frac{1}{|\mathbf{J}(\psi)|}\mathbf{J}^T(\psi)$ where $\mathbf{J}^T(\psi)$ is the matrix transpose of $\mathbf{J}(\psi)$.

In addition, $\mathbf{W} = [\omega_1\ \omega_2\ \omega_3\ \omega_4]^T$ represents the angular velocity of each wheel, defined as,

$$\mathbf{W}(t) = \boldsymbol{\Gamma}(a,b)\ \mathbf{u}(t) \tag{4}$$

Considering the mechanical characteristics of the $\mathbf{\Gamma}$ platform it is defined as

$$\mathbf{\Gamma}(a,b)=\frac{1}{R}\begin{bmatrix}1 & 1 & (a+b)\\ 1 & -1 & -(a+b)\\ 1 & 1 & -(a+b)\\ 1 & -1 & (a+b)\end{bmatrix} \qquad (5)$$

3.2 Dynamic Modeling

The dynamic of the mobile platform was developed based on the Euler-Lagrange formulation, for which it is considered that the power energy is equal to zero, since it has no displacement in the $\mathcal{Z}$ axis with respect to $\{\mathcal{R}\}$. Therefore, the kinetic energy of the robot is defined as,

$$\mathcal{L} = E_C = \dot{\boldsymbol{\eta}}^T \mathbf{M}_{R1}\dot{\boldsymbol{\eta}} + \mathbf{W}^T \mathbf{I}_1\ \mathbf{W} \qquad (6)$$

where $M_{R1}=\frac{1}{2}diag\{m_R, m_R, I_R\}$ with m_R and I_R defined the mass and inertia of the robot, respectively. Also, $I_1=\frac{1}{2}diag\{I_W, I_W, I_W, I_W\}$ where I_W is the inertia of the wheels. Thus, the dynamic of the platform is defined as

$$\bar{\mathbf{M}}\dot{\boldsymbol{\eta}} + \bar{\mathbf{C}}\boldsymbol{\eta} = \mathbf{E}^T \boldsymbol{\tau}_i \qquad (8)$$

Now, considering that the moving platform is driven by DC motors, it is possible to define

$$\boldsymbol{\tau}_i = \frac{k_{pa}}{R_{pa}}\left(v_i - k_{pb}\mathbf{W}_i\right), \quad \text{with } i=1,2,3,4 \qquad (9)$$

where, v_i represents the input voltage to each motor; k_{pa}, R_{pa}, k_{pb} electrical constants of the motor. In addition, one PD controller per motor is considered,

$$\mathbf{v}_v = \mathbf{K_P}\left(\mathbf{u_{ref}} - \mathbf{u}\right) - \mathbf{u}\mathbf{K_D} \qquad (10)$$

where $\mathbf{K_P} > 0$ and $\mathbf{K_D} > 0$ weigh control errors. Through (8) - (10) the mathematical model with velocity reference signals is obtained.

$$\begin{bmatrix}u_{fref}\\ u_{lref}\\ \omega_{ref}\end{bmatrix}=\begin{bmatrix}\varsigma_1 & \varsigma_2 & \varsigma_3\\ \varsigma_2 & \varsigma_4 & -\varsigma_5\\ -\varsigma_6 & \varsigma_7 & \varsigma_8\end{bmatrix}\begin{bmatrix}\dot{u}_f\\ \dot{u}_l\\ \dot{\omega}\end{bmatrix}+\begin{bmatrix}\omega\varsigma_9+\varsigma_{10} & -\omega\varsigma_{11} & -2\omega\varsigma_{12}\\ -\omega\varsigma_{13} & -\omega\varsigma_9+\varsigma_{14} & -2\omega\varsigma_{15}\\ \omega\varsigma_{12} & \omega\varsigma_{15} & \varsigma_{16}\end{bmatrix}\begin{bmatrix}u_f\\ u_l\\ \omega\end{bmatrix},$$

$$\mathbf{u_{ref}}(t) = \mathbf{M}(\varsigma)\dot{\mathbf{u}}(t) + \mathbf{C}(\varsigma,\mathbf{u})\mathbf{u}(t), \qquad (11)$$

where $\varsigma = \begin{bmatrix}\varsigma_1 & \varsigma_2 & \dots & \varsigma_j\end{bmatrix}^T \in R^j$ with $j=16$ represent the dynamic parameters of the mobile platform.

4 Control Scheme

The scheme for adaptive autonomous control for a robot with omnidirectional drive is presented in this item. The proposed control scheme for solving the motion control problem of an omnidirectional robot control is shown in Fig., 3. The control scheme is based on the mathematical models of the mobile platform:

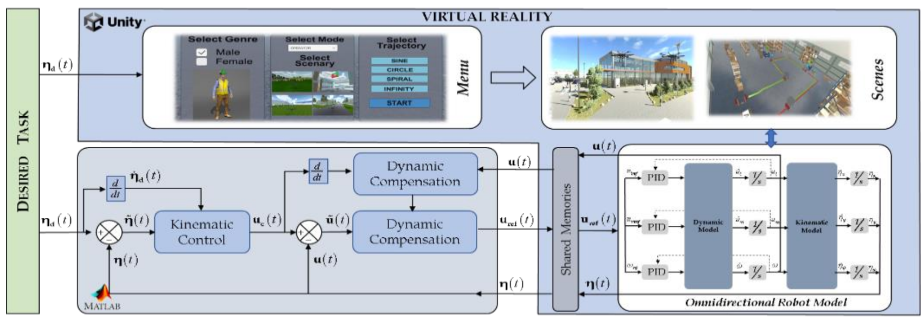


Fig. 3. Control scheme in the virtual training system

4.1 Kinematic Controller, the controller is based on the kinematic of the omnidirectional platform. Therefore, considering (3) we have,

$$\mathbf{u_c} = \mathbf{J}^{-1}\left(\dot{\boldsymbol{\eta}}_\mathbf{d} + \boldsymbol{\Gamma}\tanh\left(\alpha\left(\boldsymbol{\eta}_\mathbf{d} - \boldsymbol{\eta}\right)\right)\right) \qquad (12)$$

where, $\mathbf{u_c} = [u_{lc} \quad u_{mc} \quad \omega_c] \in R^3$ is the calculated velocities; $\mathbf{J}^{-1}$ is the inverse matrix; $\boldsymbol{\Gamma} > 0 \in R^{3x3}$ weight control errors; and $\alpha \in R^{+}$ that defines the saturation slope of the control errors.

4.2 Adaptive Dynamic Controller, the output of the dynamic controller in each sampling period considers the adaptation of dynamic parameters of the mobile platform. The dynamic parameters of the robot may vary as a function of the load carried by the robot or the surface on which it performs the task. Hence, the dynamic model (12) can be expressed as:

$$\begin{bmatrix} u_{fref} \\ u_{lref} \\ \omega_{ref} \end{bmatrix} = \underbrace{\begin{bmatrix} \varsigma_1 & \varsigma_2 & \varsigma_3 \\ \varsigma_2 & \varsigma_4 & -\varsigma_5 \\ -\varsigma_6 & \varsigma_7 & \varsigma_8 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \dot{u}_f \\ \dot{u}_l \\ \dot{\omega} \end{bmatrix} + \underbrace{\begin{bmatrix} \omega u_f & u_f & -\omega u_l & -2\omega^2 & 0 & 0 & 0 & 0 \\ -\omega u_l & 0 & 0 & 0 & -\omega u_f & u_l & -2\omega^2 & 0 \\ 0 & 0 & 0 & \omega u_f & 0 & 0 & \omega u_l & \omega \end{bmatrix} \begin{bmatrix} \varsigma_9 \\ \varsigma_{10} \\ \vdots \\ \varsigma_{16} \end{bmatrix}}_{\chi}$$

$$\mathbf{u_{ref}}(t) = \mathbf{M}(\varsigma)\dot{\mathbf{u}}(t) + \boldsymbol{\chi}(\varsigma, \mathbf{u}) \qquad (13)$$

Based on dynamic model (13) we propose the following control law,

$$\mathbf{u_{ref}} = \mathbf{M}\boldsymbol{\sigma} + \boldsymbol{\chi} \qquad (14)$$

where, $\boldsymbol{\sigma} \in R^3$ defined as: $\boldsymbol{\sigma} = \dot{\mathbf{u}}_\mathbf{c} + \boldsymbol{\kappa}\left(\mathbf{u}_\mathbf{c} - \mathbf{u}\right)$ with $\boldsymbol{\kappa} = diag\left(\kappa_{ul}, \kappa_{um}, \kappa_{\omega}\right) > 0 \in R^{3x3}$ weight velocities control errors; and the velocities control errors are defined as $\tilde{\mathbf{u}} = \mathbf{u}_\mathbf{c} - \mathbf{u}$. Now, rewrite (14):

$$\mathbf{u}_\mathbf{ref}\left(t\right) = \boldsymbol{\Omega}\left(\boldsymbol{\sigma}, \mathbf{u}\right)\varsigma\left(t\right) \tag{15}$$

where,

$$\boldsymbol{\Omega} = \begin{bmatrix} \dot{u}_f & \dot{u}_l & \dot{\omega} & 0 & 0 & 0 & 0 & 0 & \omega u_f & u_f & -\omega u_l & -2\omega^2 & 0 & 0 & 0 & 0 \\ 0 & \dot{u}_f & 0 & \dot{u}_l & -\dot{\omega} & 0 & 0 & 0 & -\omega u_l & 0 & 0 & 0 & -\omega u_f & u_l & -2\omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\dot{u}_f & \dot{u}_l & \dot{\omega} & 0 & 0 & 0 & \omega u_f & 0 & 0 & \omega u_l & \omega \end{bmatrix}; \boldsymbol{\varsigma} = \begin{bmatrix} \varsigma_1 & \varsigma_2 & \cdots & \varsigma_j \end{bmatrix}^T \in R^j$$

with $j = 16$. Due to the uncertainties at the time of calculating the control actions caused by the parameter values used and the dynamic effects not considered in the modeling. In the case of any uncertainty in the unicycle robot parameters, the control law:

$$\mathbf{u}_\mathbf{ref} = \boldsymbol{\Omega}\hat{\boldsymbol{\varsigma}} = \boldsymbol{\Omega}\boldsymbol{\varsigma} + \boldsymbol{\Omega}\tilde{\boldsymbol{\varsigma}} = \mathbf{M}\boldsymbol{\sigma} + \boldsymbol{\chi} + \boldsymbol{\Omega}\tilde{\boldsymbol{\varsigma}} \tag{16}$$

where, ς y $\hat{\varsigma}$ are the actual dynamic parameters and estimated parameters of the robot; hence, the errors is defined as $\tilde{\varsigma} = \hat{\varsigma} - \varsigma$.

4.3 Stability Analysis

By considering (13) in a compact form it follows $\mathbf{M}\dot{\mathbf{u}} + \boldsymbol{\chi} = \mathbf{M}\boldsymbol{\sigma} + \boldsymbol{\chi} + \boldsymbol{\Omega}\tilde{\boldsymbol{\varsigma}}$ and is equivalent to $\mathbf{M}\left(\boldsymbol{\sigma} - \dot{\mathbf{u}}\right) = -\boldsymbol{\Omega}\tilde{\boldsymbol{\varsigma}}$. In which the following are $\boldsymbol{\sigma} - \dot{\mathbf{u}} = \dot{\tilde{\mathbf{u}}} + \boldsymbol{\kappa}\tilde{\mathbf{u}}$; hence $\mathbf{M}\left(\dot{\tilde{\mathbf{u}}} + \boldsymbol{\kappa}\tilde{\mathbf{u}}\right) = -\boldsymbol{\Omega}\tilde{\boldsymbol{\varsigma}}$ same that represents the equation of the controller's errors. Then propose a Lyapunov candidate function $V\left(\tilde{\mathbf{u}}, \tilde{\boldsymbol{\varsigma}}\right) = \frac{1}{2}\left(\tilde{\mathbf{u}}^\mathrm{T}\mathbf{M}\tilde{\mathbf{u}} + \tilde{\boldsymbol{\varsigma}}^\mathrm{T}\Upsilon\tilde{\boldsymbol{\varsigma}}\right)$ and its partial derivative $\dot{V}\left(\tilde{\mathbf{u}}, \tilde{\boldsymbol{\varsigma}}\right) = -\tilde{\mathbf{u}}^\mathrm{T}\mathbf{M}\boldsymbol{\kappa}\tilde{\mathbf{u}} - \tilde{\mathbf{u}}^\mathrm{T}\boldsymbol{\Omega}\tilde{\boldsymbol{\varsigma}} + \tilde{\boldsymbol{\varsigma}}^\mathrm{T}\Upsilon\dot{\tilde{\boldsymbol{\varsigma}}}$, where $\Upsilon \in R^{16x16}$ is a positive definite diagonal matrix; the matrix $\mathbf{M} > 0$ must be greater than zero and the vector of the real dynamic constants is considered constant with the errors $\dot{\tilde{\varsigma}} = \dot{\hat{\varsigma}}$.

By means of the law for updating parameters $\dot{\hat{\varsigma}} = \Upsilon^{-1}\boldsymbol{\Omega}\tilde{\mathbf{u}}$ to substitute in the derivative of the Lyapunov candidate function $\dot{V} = -\tilde{\mathbf{u}}^\mathrm{T}\mathbf{M}\boldsymbol{\kappa}\tilde{\mathbf{u}} \leq 0$, The same that allows to verify the stability with the control errors as bounded signals. For integration, $V\left(t\right) - V\left(0\right) = -\int_0^T \tilde{\mathbf{u}}^\mathrm{T}\mathbf{M}\boldsymbol{\kappa}\tilde{\mathbf{u}}dt$ and without considering $V\left(t\right)$ is obtained $V\left(0\right) \geq \int_0^T \tilde{\mathbf{u}}^\mathrm{T}\mathbf{M}\boldsymbol{\kappa}\tilde{\mathbf{u}}dt$. With $\mathbf{M}\boldsymbol{\kappa}$ defined positive and symmetrical $\lambda_{\min}\left(\mathbf{M}\boldsymbol{\kappa}\right)\left\|\tilde{\mathbf{u}}\right\|^2 \leq \tilde{\mathbf{u}}^T\mathbf{M}\boldsymbol{\kappa}\tilde{\mathbf{u}} \leq \lambda_{\max}\left(\mathbf{M}\boldsymbol{\kappa}\right)\left\|\tilde{\mathbf{u}}\right\|^2$, where $\lambda_{\min}\left(.\right)$, $\lambda_{\max}\left(.\right)$ are the minimum and maximum eigenvalues of the matrix. Finally, there is (17) after the conditions analyzed in [17] considering the constants $\alpha_1 = \chi(\mathbf{M}\boldsymbol{\kappa})$ and $\mu_r = \chi(\boldsymbol{\Gamma})$.

$$\dot{V} \leq -\alpha_1\left\|\tilde{\mathbf{v}}\right\|^2 - \mu_r\left\|\tilde{\varsigma}\right\|^2 + \tau, \;\; \tau = \mu_r\left\|\tilde{\varsigma}\right\|\left\|\varsigma\right\|. \tag{17}$$

In the analysis of control errors, considering the position error is determined by proposing a candidate Lyapunov function of quadratic errors $\mathbf{V}\left(\tilde{\boldsymbol{\eta}}\right) = \frac{1}{2}\tilde{\boldsymbol{\eta}}^T\tilde{\boldsymbol{\eta}}$ and its respective derivative $\mathbf{V}\left(\tilde{\boldsymbol{\eta}}\right) = \tilde{\boldsymbol{\eta}}^T\dot{\tilde{\boldsymbol{\eta}}}$. By considering velocities errors $\dot{\tilde{\mathbf{\eta}}}(t) = \dot{\mathbf{\eta}}_\mathbf{d} - \dot{\mathbf{\eta}}$ and equalize with the control law (12) and the kinematic model results in a negative definite function $\dot{\mathbf{V}}(\tilde{\mathbf{\eta}}) = -\tilde{\mathbf{\eta}}^T \mathbf{\Gamma} \tanh(\alpha \tilde{\mathbf{\eta}}(t))$. Therefore, in order to guarantee the stability of the control law, the condition must be fulfilled that $\mathbf{\Gamma} > 0$, to ensure that $\tilde{\mathbf{\eta}}(t) \in R^3 \to 0$ when $t \to \infty$.

5 Experimental Results

This item presents the most relevant results of the implemented system. This item considers the construction of a prototype; the digitalization environment; the implemented control scheme; and the results of the usability test.

5.1 Robot Omnidireccional

This work presents the construction of a robot with omnidirectional traction, developed in the Master's program in Electronics, mention Industrial Networks. Fig., 4 shows the robotic prototype built.

Fig. 4. Omnidirectional robotic prototype

From the electrical point of view, the omnidirectional robot is mainly composed of four main stages, see Fig., 5: *i) Power system,* it consists of a LIPO battery, a bank of current protections, and a DC/DC converter module that allows powering all the devices that are part of the robotic platform, i.e., actuators, sensors, control system and communication modules; *ii) Motor driver*, it is responsible for generating the voltage corresponding to the DC motors through H-Bridges, in addition, a PID controller is considered for each motor; *iii) Actuators*, consists of four DC motors with encoder, each motor supporting a current of 2A; *iv)Control system,* is composed of a control unit where the closed-loop control algorithms are implemented; and finally *v)Communication,* manages the wireless communication between the robot and an external computer.

On the other hand, the identification of the dynamic parameters that represents the behavior of the omnidirectional robot was performed with the robotic prototype built. The method used for the identification is through general identification, or black box, where the objective is to establish the input-output relationship of the system, without making physical interpretations on the composition of the mathematical model, see Fig. 6 (a)

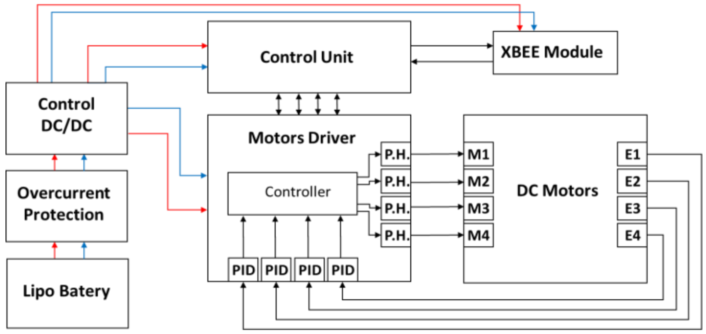
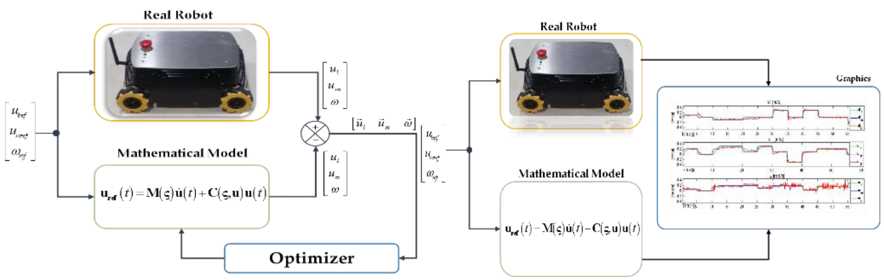


Fig. 5. Omnidirectional robot electrical diagram



(a) Dynamic parameters identification (b) Dynamic parameters validation

Fig. 6. Dynamic parameters identification and validation

The identification of the dynamic parameters of the mobile platform, was developed based on the minimization of an objective function: $\min f(\varsigma)$ where, $\varsigma \in \mathbb{R}^{16}$ are the parameters of the system to be identified. The identification of the parameters is carried out through the optimization algorithm, when the value of $f(\varsigma)$ reaches a minimum the search delivers the vector of parameters ς that satisfy the minimization of the function. Fig. 6(b) presents the validation results of the dynamic model of the mobile robot.

The kinematic and dynamic models obtained from the constructed prototype are implemented in the digitalization environment. Furthermore, perturbations in the robot output are considered according to the type of surface.

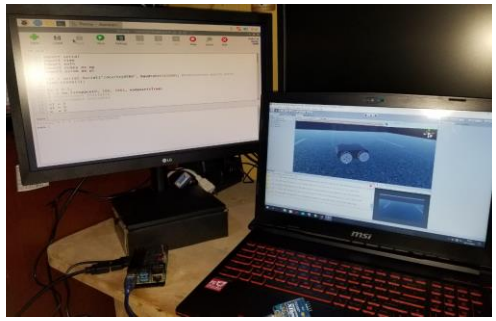


Fig. 7. Virtual training system – Unity3D graphics engine

5.2 Control Scheme Implementation

Relevant results are presented in this subsection. Each test was run both in the virtual environment and experimentally. Figure 8 presents the stroboscopic movement of the omnidirectional traction robot based on experimental data. Figure 9 presents the control errors evolution $\tilde{\boldsymbol{\eta}} \in R^3$, which are $\left|\tilde{\boldsymbol{\eta}}(t)\right| < 0,2\ [\mathrm{m}]$. Finally, Fig., 10 presents the control actions for the omnidirectional robot.

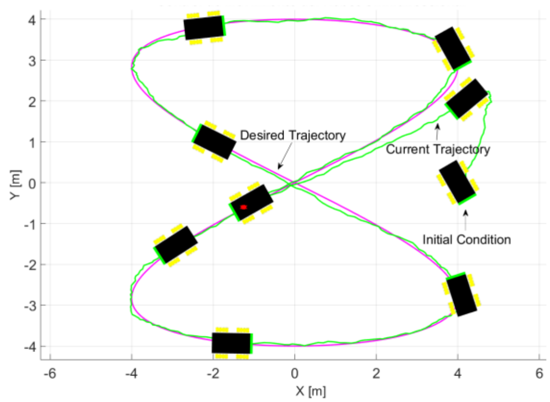


Fig. 8. Stroboscopic motion of the omnidirectional robot based on experimental data

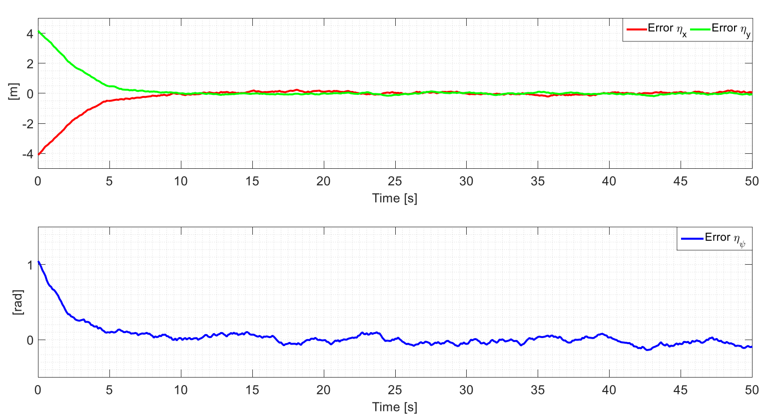


Fig. 9. Control errors evolution

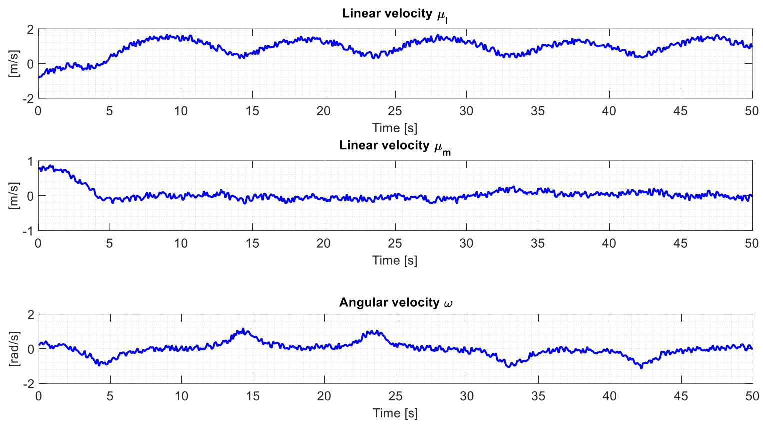


Fig. 10. Velocity commands of the mobile robot

Finally, through the usability test (SUS [17], [18]) a weighting of 78.5% was obtained from a group of fifteen students, so it can be concluded that the training system developed is good.

6 Analysis and Conclusions

This work presented the development of a digitalization system for the autonomous control of omnidirectional traction robots. The system considers a digitized environment in Unity3D, in which the mathematical models of the omnidirectional robot are implemented. The mathematical models obtained were validated with a built robotic prototype, which consisted of four Mecanum type wheels. In order to evaluate the training system developed, a cascade control scheme was implemented. Finally, from the usability tests performed on students, a SUS percentage of 78.5% was obtained, which is within the acceptable margin.

Acknowledgment

This article shows the results of the Master's Degree Program in "Maestría en Electrónica y Automatización con Mención en Redes Industriales" of the Postgraduate Centre of the Universidad de las Fuerzas Armadas ESPE. Finally, the authors would like to thank the ARSI research group for their advice in the development of the degree work.

References

1. Sánchez, H.; Martínez, L.S.; González, J.D. Educational Robotics as a Teaching Tool in Higher Education Institutions: A Bibliographical Analysis. *J. Phys.: Conf. Ser.* **2019**, *1391*, 012128, doi:10.1088/1742-6596/1391/1/012128.
2. Ruspini, E. The Robot and Us: An "Antidisciplinary" Perspective on the Scientific and Social Impacts of Robotics. *Journal of Tourism Futures* **2019**, *5*, 297–298, doi:10.1108/JTF-09-2019-089.

3. Sharma, A.; Sinha, A. Approach of Automation Manufacturing and Optimization in a Robotic Industry. *International Journal of Robotics and Automation* **2021**, *7*, 18–35.
4. Andaluz, V.; Varela Aldás, J.; Chicaiza, F.; Quevedo, W.; Ruales Martínez, M. Teleoperation of a Mobile Manipulator with Feedback Forces for Evasion of Obstacles. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao* **2019**, *2019*, 291–304.
5. Pal, S.; Gupta, S.; Das, N.; Ghosh, K. Evolution of Simultaneous Localization and Mapping Framework for Autonomous Robotics—A Comprehensive Review. *Journal of Autonomous Vehicles and Systems* **2022**, *2*, doi:10.1115/1.4055161.
6. Carvajal, C.P.; Proaño, L.; Pérez, J.A.; Pérez, S.; Ortiz, J.S.; Andaluz, V.H. Robotic Applications in Virtual Environments for Children with Autism. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2017**, *10325 LNCS*, 175–187.
7. Zeini, M.; Pirmoradian, M. Design and Construction of a Unicycle Robot Controlled by Its Center of Gravity. *Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering* **2021**, *13*, 59–73.
8. Sun, Z.; Xia, Y.; Dai, L.; Campoy, P. Tracking of Unicycle Robots Using Event-Based MPC With Adaptive Prediction Horizon. *IEEE/ASME Transactions on Mechatronics* **2020**, *25*, 739–749, doi:10.1109/TMECH.2019.2962099.
9. Chen, X.; Huang, Z.; Sun, Y.; Zhong, Y.; Gu, R.; Bai, L. Online On-Road Motion Planning Based on Hybrid Potential Field Model for Car-Like Robot. *J Intell Robot Syst* **2022**, *105*, 7, doi:10.1007/s10846-022-01620-5.
10. Chen, H.; Yang, H.; Wang, X.; Zhang, T. Formation Control for Car-like Mobile Robots Using Front-Wheel Driving and Steering. *International Journal of Advanced Robotic Systems* **2018**, *15*, 1729881418778228, doi:10.1177/1729881418778228.
11. Ortiz, J.S.; Molina, M.F.; Andaluz, V.H.; Varela, J.; Morales, V. Coordinated Control of a Omnidirectional Double Mobile Manipulator. In Proceedings of the IT Convergence and Security 2017; Kim, K.J., Kim, H., Baek, N., Eds.; Springer: Singapore, 2018; pp. 278–286.
12. Yunardi, R.T.; Arifianto, D.; Bachtiar, F.; Prananingrum, J.I. Holonomic Implementation of Three Wheels Omnidirectional Mobile Robot Using DC Motors. *Journal of Robotics and Control (JRC)* **2021**, *2*, 65–71, doi:10.18196/jrc.2254.
13. Saenz, A.; Santibañez, V.; Bugarin, E.; Dzul, A.; Ríos, H.; Villalobos-Chin, J. Velocity Control of an Omnidirectional Wheeled Mobile Robot Using Computed Voltage Control with Visual Feedback: Experimental Results. *Int. J. Control Autom. Syst.* **2021**, *19*, 1089–1102, doi:10.1007/s12555-019-1057-6.
14. Andaluz, V.H.; Pérez, J.A.; Carvajal, C.P.; Ortiz, J.S. Virtual Environment for Teaching and Learning Robotics Applied to Industrial Processes. In Proceedings of the Augmented Reality, Virtual Reality, and Computer Graphics; De Paolis, L.T., Bourdot, P., Eds.; Springer International Publishing: Cham, 2019; pp. 442–455.
15. Wu, M.; Dai, S.-L.; Yang, C. Mixed Reality Enhanced User Interactive Path Planning for Omnidirectional Mobile Robot. *Applied Sciences* **2020**, *10*, 1135, doi:10.3390/app10031135.
16. Andaluz, V.H.; Chicaiza, F.A.; Gallardo, C.; Quevedo, W.X.; Varela, J.; Sánchez, J.S.; Arteaga, O. Unity3D-MatLab Simulator in Real Time for Robotics Applications. In *Augmented Reality, Virtual Reality, and Computer Graphics*; De Paolis, L.T., Mongelli, A., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2016; Vol. 9768, pp. 246–263 ISBN 978-3-319-40620-6.
17. Sauro, J.; Lewis, J.R. When Designing Usability Questionnaires, Does It Hurt to Be Positive? In Proceedings of the Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11; ACM Press: Vancouver, BC, Canada, 2011; p. 2215.
18. Salvendy, G. *Handbook of Human Factors and Ergonomics*; John Wiley & Sons, 2012; ISBN 978-1-118-12908-1.