Studies in Computational Intelligence 1093

Ahmad Taher Azar
Anis Koubaa *Editors*

Artificial Intelligence for Robotics and Autonomous Systems Applications

Springer

Studies in Computational Intelligence

Volume 1093

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Indexed by SCOPUS, DBLP, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

Ahmad Taher Azar · Anis Koubaa
Editors

Artificial Intelligence for Robotics and Autonomous Systems Applications

Springer

Editors
Ahmad Taher Azar
College of Computer and Information Sciences
Prince Sultan University
Riyadh, Saudi Arabia

Automated Systems and Soft Computing Lab (ASSCL)
Prince Sultan University
Riyadh, Saudi Arabia

Faculty of Computers and Artificial Intelligence
Benha University
Benha, Egypt

Anis Koubaa
College of Computer and Information Sciences
Prince Sultan University
Riyadh, Saudi Arabia

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-031-28714-5 ISBN 978-3-031-28715-2 (eBook)
https://doi.org/10.1007/978-3-031-28715-2

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Four Wheeled Humanoid Second-Order Cascade Control of Holonomic Trajectories

A. A. Torres-Martínez, E. A. Martínez-García, R. Lavrenov, and E. Magid

Abstract This work develops model-based second-order cascade motion controller of a holonomic humanoid-like wheeled robot. The locomotion structure is comprised of four mecanum wheels radially arranged. The model is given as a function of all wheels contribution adding maneuverability to upper limbs. High-order derivatives are synchronized through numeric derivations and integration, obtained online for consistent performance of inner loops feedback. The controller deploys reference inputs vectors, both global and local to each cascade loop. In this approach, the controller decreases errors in position, velocity and acceleration simultaneously through Newton-based recursive numerical approximations. A main advantage of this approach is the robustness obtained by three recursive feedback cascades: distance, velocity and acceleration. Observers are modeled by combining multi-sensor inputs. The controller showed relative complexity, effectiveness, and robustness. The proposed approach demonstrated good performance, re-routing flexibility and maneuverability through numerical simulations.

Keywords Cascade-control · Holonomy · Wheeled-humanoid · Path-tracking · Control-loop · Mobile-robot

A. A. Torres-Martínez · E. A. Martínez-García (✉)
Institute of Engineering and Technology, Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Mexico
e-mail: edmartin@uacj.mx

R. Lavrenov · E. Magid
Institute of Information Technology and Intelligent Systems, Kazan Federal University, Kazan, Russian Federation
e-mail: magid@it.kfu.ru

E. Magid
HSE Tikhonov Moscow Institute of Electronics and Mathematics, HSE University, Moscow, Russian Federation

A. T. Azar and A. Koubaa (eds.), *Artificial Intelligence for Robotics and Autonomous Systems Applications*, Studies in Computational Intelligence 1093,
https://doi.org/10.1007/978-3-031-28715-2_15

1 Introduction

Wheeled mobile robots are widely used in a number of applications. The performance of a wheeled robot is considerably good in particular on flat and structured floors. For instance, they are faster, more efficient in reaching positions and usually more effective in terms of mechanical energy requirements than walking bipeds. In numerous robotic applications, particularly in cluttered environments, omnidirectional rolling locomotion capabilities result suitable to easily change the robot's body posture to basically move towards any direction without explicit yaw control. Nowadays, deployment of omnidirectional wheels as a means for locomotion in mobile robotics is highly demanding in a number or applications due to their ability to move in any direction and particularly driving in plain structured confined spaces [1, 2]. The use of omniwheels, unlike conventional wheels have fewer kinematic constraints and allow the robot to move in a wide range of mobility. Adding, holonomy, considerable maneuverability. In modern times, the cases of omniwheel-based holonomic robots developed for different applications are considerable numerous and relevant. For instance, the use of personal assistant robots as walking-helper tool, demonstrated capability to provide guidance and dynamic support for impaired walking people [3]. There are also other types of human-robot assistants in cases where mobile robots are purposed to perform interaction socially assistive [4]. In healthcare, robotic systems have been designed with mecanum wheels for providing omnidirectional motion to wheelchairs [5]. Although, manipulators onboard mobile platforms are not relatively new approaches, however mobile manipulators with omnidirectional locomotion provides interesting advantages. Mecanum-wheeled platforms have been performed as holonomic vehicular manipulators moving in industrial working spaces [6]. Park et al. [7] presented a controller for velocity tracking and vibration reduction of a cart-pole inverted pendulum like-model of omnidirectional assistive mobile robot. The robot adopted mecanum wheel rolling with suspension for keeping consistent contact mecanum wheel and ground while transporting heavy goods placed on high locations.

Moreover, instrumented omnidirectional platforms with visually guided servoing devices have been reported [8]. Furthermore, holonomic robotic platforms have been exploited as robotized sporting and training technology to provide assistance and training in racquet sports [9]. A traditional robotic application is exploiting advantages of omniwheel-based mobile robots being deployed as domestic assistants in household environments [10]. An advantageous application of mecanum-wheels used as an omni-directional mobile robot in industrial fields has been critical. For instance, autonomous material indoor transportation [11] as well as robotic platforms of four omnidirectional wheels working in warehouses [12]. The work [13] performed collaborative manipulation by multirobot displacing payloads transported to desired locations in planar obstacle-clustered scenarios maneuvering through narrow pathways for which advocated the use of Mecanum-Wheeled Robots positioning without body-orientation change. The work [14] developed modular reconfiguration by deploying a group of vehicles to perform different mission tasks. Recofnigura-

tion was done at the level of motion planning deploying four-wheel-drive Mecanum vehiclemobile robots. A critical deployment of omniwheel robotics is in a gaining popularity field for omnidirectional humanoids, that is in nursing and rehabilitation [15].

A variety of demands on the use of robots differs considerably on how to deploy them. For instance, in the industry with robots working on-site with highly accurate robotic arms, or mobile platforms moving heavy loads and assisting humans workers in close proximity [16].

This chapter presents the trajectory tracking model of a humanoid robot at the stage of kinematic modeling and simulation. This work approaches a model of two upper limbs of three joints fixed on a trunk that is placed on a Mecanum wheeled platform with four asynchronous rolling drives. This research's main contribution is the development of a three-cascade kinematic trajectory tracking controller. Each cascade is comprised of a different order derivative deduced from the robot's kinematic model. Different observers to complement the control are developed considering a deterministic approach and based on wheels-encoder and an inertial measurement unit. The omniwheels physical arrangement is radially equidistant and tangential rotated with respect to (w.r.t.) the center of reference. This work shows numerical simulation results that allow validating and understanding proposed models and ideas, as well as to refine them before converting them into feasible and operational physical systems.

This chapter organizes the sections as follows. Section 2 briefly discusses similar works. Section 3 deduces motion equations of the robot's arms and its four-wheel four-drive omnidirecional rolling platform. Section 4 defines the sensing model and observers used as online feedback elements. Section 5 describes the three-cascade controller. Finally, Sect. 7 provides conclusion of the work.

2 Related Work

The study of position errors and calibration methods for robot locomotion with omnidirectional wheels has demonstrated to be relevant [17]. The work [18] developed a reference-based control for a mecanum-wheels omnidirectional robot platform, relying on the robot's kinematic model and generate trajectories and optimal constrained navigation. The cost function quantified differences between the robot's path prediction and using a family of parameterized reference trajectories. [19] demonstrated control of a time-varying proportional integral derivative model for trajectory tracking of a mecanum-wheeled robot. It used linearization of a nonlinear kinematic error model and controller's parametric coefficients adjusted by trial-and-error.

Omniwheel-based robot motion is effected by systematic perturbations differently as in conventional wheeled robots. Identifying the sources of pose errors are critical to develop methods for kinematic errors reduction of omni-directional robotic system [20]. The work [21] evaluated a method to correct systematic odometry errors of a humanoid-like three-wheeled omnidirectional mobile robot. Correction was made

by iteratively adjusting effective values with respect to robot's kinematic parameters, matching referenced positions by estimation. The correct functionality of a four Mecanum wheels robot was approached by the Dijkstra's algorithm and tested in [22]. The work [23], proposed an odometry-based kinematic parameters error calibration deploying least squares linear regression for a mobile robot with three omniwheels. Similarly, [24] presented a calibration system to reduce pose errors based on the kinematic formulation of a three-wheeled omnidirectional robot, considering systematic and non-systematic errors compensation. Another three-omniwheel robot was reported in [27], where motion calibration is obtained by optimizing the effective kinematic parameters and the inverse Jacobian elements are minimized through a cost function during path tracking. Previous cited works reported different solutions to calibrate odometric position errors in omni-wheel-based mobile robots. Such works highlight two main approaches, by numerical estimations and by modeling deterministic metrical errors. A main difference with respect to the present context is the focus on tracking control of local Cartesian points within a global trajectory, unlike encoders usage, other pose measurement methods are considered, for instance data fusion of online heterogeneous inertial measurements.

The research [25] reported a radially arranged omnidirectional four-wheeled robot controlled by three proportional-integral-derivative controllers (PID). The PIDs controlled speed, heading, and position during trajectory tracking and using odometry to measure the robot's posture. The research [26] developed a theoretical kinematic basis for accurate motion control of combined mobility configurations based on coefficients for speeds compensation mainly caused by wheels slippage in a four mecanum wheels industrial robot. The work [28] reported a controller based on observer and a high order sliding mode for a multirobot system of three-wheeled omnidirectional platforms. Previous cited works reported motion control approaches of omniwheel robotic platforms that tackled either slippage problems or motion inaccuracies to improve the robot's posture. As a difference with the present work, it assumes that pose observation is already adequate but rather focusing on robustly controlling the robot's path motion along a linear trajectory segment by triple kinematics control of high-order derivatives, simultaneously.

The research reported in [29] introduced a general model for analysis of symmetrical multi-wheel omnidirectional robot. Inclusion of constrained trajectory planning optimization was implemented for free collision navigation. The reported work in [30] introduced a kinematic model for trajectory tracking control of a radially oriented four-wheel omnidirectional robot using odometry as feedback.

The work [31] presented a four Mecanum wheels omnidirectional mobile robot for motion planning tasks implementing fault-tolerance on wheels with a fuzzy controller. The work conducted by [32] proposed a neural control algorithm to determine neural network weights adaptation with parametric disturbances as an intelligent control for path motion by a four mecanum wheels mobile robot. Fault tolerant navigation control on a four mecanum-wheel structure was developed by [33], using adaptive control second order dynamics and parametric uncertainty. A controller for an omni-wheeled industrial manipulator was presented by [34], it adaptively combined a fuzzy wavelet neural network, a sliding mode and a fractional-order criterion.

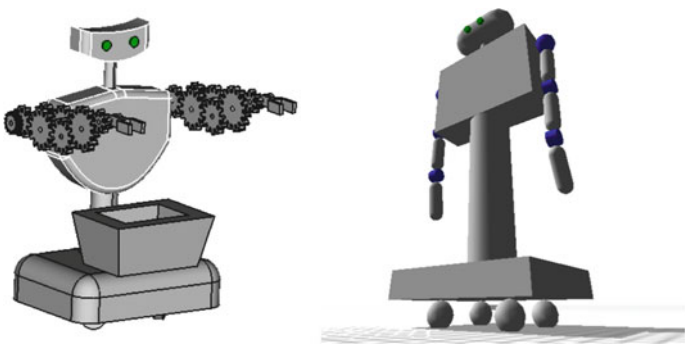
Finally, a path-following control using extended Kalman filtering for sensor fusion was introduced in [35].

Some of previous cited works reported approaches using soft-computing techniques combined with traditional control methods for tracking, either for recovery of disturbances and fault tolerances in tracking motion control. As a difference, in the present research a model-based recursive control is proposed with the particularity of implementing inner multi-cascades combining multiple higher-order inputs. Numerical errors are reduced with respect to a reference model by successive approximations as convergence laws. The focus presented in this research differs from most the cited related work, fundamentally in the class of control's structure and the kind of observers models. For instance, while a traditional PID controller might combine three different order derivatives as a summation of terms into an algebraic expression, the proposed approach exploits each derivative inside another of lower order and faster sampling as different recursive control cycles.

3 Robot Motion Model

This section describes the essential design parts of the proposed robotic structure at the level of simulation model. Additionally, both kinematic models, the onboard manipulators and the four mecanum wheels and the omnidirectional locomotive structure are illustrated.

Figure 1a depicts the humanoid CAD concept of the proposed robotic platform. Figure 1b shows a basic figure created in C/C++ language as a resource for numerical simulations, which deploy the Object Dynamic Engine (ODE) libraries to create simulated animations.



(a) CAD structure. (b) Simulation model.

Fig. 1 Mecanum four-wheeled humanoid structure. **a** a CAD mo del. **b** a simulation model from the physics engine ODE

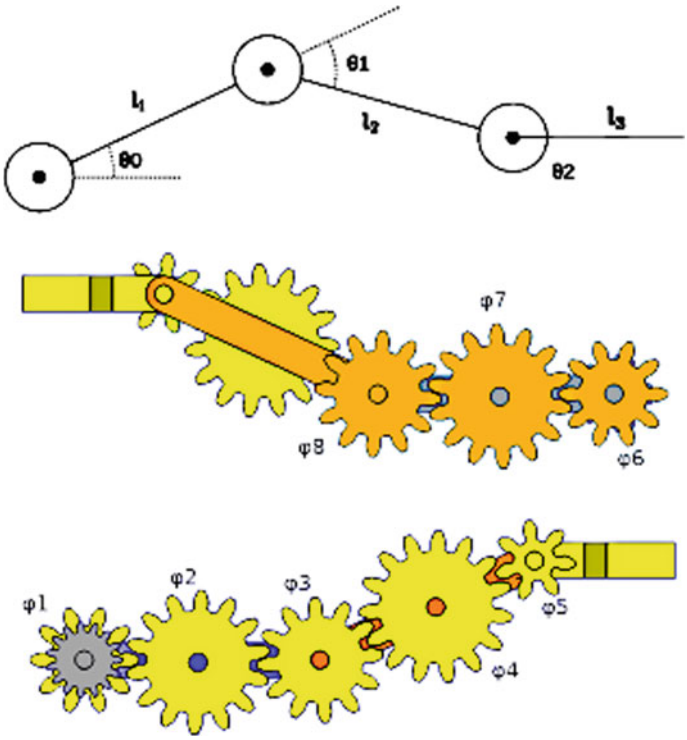


Fig. 2 Onboard arms basic mechanism. Joints and links kinematic parameters (above). Side of the elbow mechanism (middle). Side of the wrist mechanism and shoulder gray-color gear (below)

The four mecanum wheels are located symmetric radially arranged and equidistant beneath the chassis structure. Each wheel is independently driven both rotary directions. This work provides the emphasis on the omnidirectional locomotion controller, since motion over the plane ground has impacts on the manipulators position, adding robot's translation and orientation is given in models separately along the manuscript. Figure 2 illustrates a basic conceptual design purposed to help describing joints' functional form. The limbs purpose in this manuscript is to illustrate general interaction in general scenarios with manipulable objects.

Therefore, the onboard arms may be modeled for multiple degrees of freedom. However, in this manuscript the manipulators have been established symmetrically planar with three rotary joints: shoulder (θ_0), elbow (θ_2) and wrist (θ_2), all turning in pitch (see Fig. 2). Additionally, the robot's orientation is assumed to be the arms' yaw motion (θ_t). The onboard arm's side view is shown in Fig. 2(below), where the gray-color gear is the actuating device φ_0 that rotates a shoulder. The arm's joint ϕ_{l_1} describes angular displacements for link l_1. Figure 2(middle) shows an antagonistic arm's side view where the orange-color joint mechanism for θ_1 (elbow) is depicted. The mechanism device for θ_1 has asynchronous motion from θ_0 and θ_2. Additionally,

Fig. 2(middle) shows yellow-color gearing system to depict how the wrist motoin is obtained and transmitted from the actuating gear φ_2 towards φ_5. The wrist rotary angle is the joint θ_2 that rotates the gripper's elevation angle.

Hence, without lose of generality, the Cartesian position is a system of equations that are established for now in two-dimension, $z = 0$, $\dot{z} = 0$ and $\ddot{z} = 0$. Let z be the depth dimension no treated in this section. Subsequently, a third Cartesian component may be stated when the robot's yaw is defined as it impacts the arms pose, given in the next sections.

From the depiction of Fig. 2(above), the following arms position x_a, y_A expressions are deduced to describe the motion in sagittal plane (pitch),

$$x_a = l_1 \cdot \cos(\theta_0) + l_2 \cdot \cos(\theta_0 + \theta_1) + l_3 \cdot \cos(\theta_0 + \theta_1 + \theta_2), \tag{1}$$

and

$$y_a = l_1 \cdot \sin(\theta_0) + l_2 \cdot \sin(\theta_0 + \theta_1) + l_3 \cdot \sin(\theta_0 + \theta_1 + \theta_2). \tag{2}$$

where the functional forms of actuating joints are described in the following Definition 1,

Definition 1 (*Joints functional forms*) Assuming gears angles and teeth numbers by φ_i and n_j, respectively, let φ_0 be the actuating joint,

$$\theta_0 = \varphi_0. \tag{3}$$

Let φ_6 be an actuating gear that transmits rotation to gear $\varphi_6 \equiv \theta_1$ for link l_2,

$$\varphi_8 = \left(\frac{n_6 n_7}{n_7 n_8}\right) \cdot \varphi_6 = \frac{n_6}{n_8} \cdot \varphi_6, \tag{4}$$

Therefore, $\theta_1 \doteq \varphi_8$. Let φ_1 transmit motion to $\varphi_5 \equiv \theta_2$ for l_3 rotation by

$$\varphi_5 = \left(\frac{n_1 n_2 n_3 n_4}{n_2 n_3 n_4 n_5}\right) \cdot \varphi_1 = \frac{n_1}{n_5} \cdot \varphi_1, \tag{5}$$

therefore $\theta_2 \doteq \varphi_5$.

Previous statements conduct to the following Proposition 1.

Proposition 1 (Arm's kinematic law) *The kinematic control including the gears mechanical advantages n_6/n_8 and n_1/n_5, reaches the reference angular positions $\theta_{0,1,2}$, while varying joints angles $\varphi_{0,8,5}$ by*

$$\begin{pmatrix} x_{a_{t+1}} - x_{a_t} \\ y_{a_{t+1}} - y_{a_t} \end{pmatrix} = \begin{pmatrix} -l_1 c_0 & -\frac{n_6}{n_8}(l_1 c_0 + l_2 c_{01}) & -\frac{n_1}{n_5}(l_1 c_0 + l_2 c_{01} + l_3 c_{012}) \\ l_1 s_0 & \frac{n_6}{n_8}(l_1 s_0 + l_2 s_{01}) & \frac{n_1}{n_5}(l_1 s_0 + l_2 s_{01} + l_3 s_{012}) \end{pmatrix} \begin{pmatrix} \theta_0 - \varphi_0 \\ \theta_1 - \varphi_8 \\ \theta_2 - \varphi_5 \end{pmatrix} \tag{6}$$

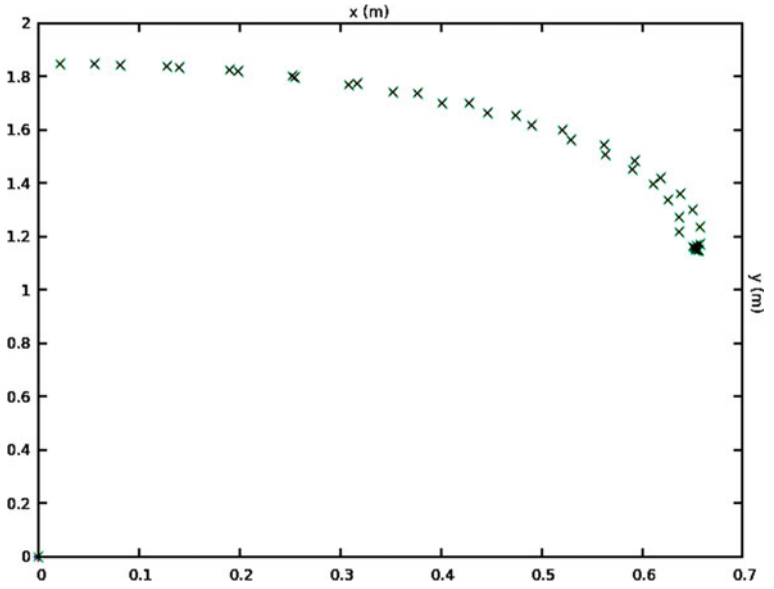


Fig. 3 Onboard arms local Cartesian motion simulation for an arbitrary trajectory

Hence, the law (6) *is satisfied when* $\lim_{\varphi_i \to \theta_i}(x, y)_{t+1} - (x, y)_t = 0$. *Being* x, y *a Cartesian position, and* $(\theta_i - \varphi_i)$ *an instantaneous joints error.*

It follows that validating previous kinematic expressions (1) and (2), Fig. 3 shows a numerical simulation for Cartesian position along an arbitrary trajectory.

Moreover, from the system of nonlinear equations modeling position (1) and (2) and hereafter assuming that joints $\theta_j(\varphi_k)$ are functions in terms of gears rotations. Thus, first-order derivative w.r.t. time is algebraically deduced and Cartesian velocities are described by

$$\begin{pmatrix} \dot{x}_a \\ \dot{y}_a \end{pmatrix} = l_1 \begin{pmatrix} -s_0 \\ c_0 \end{pmatrix} \dot{\theta}_0 + l_2 \begin{pmatrix} -s_{01} \\ c_{01} \end{pmatrix} \sum_{i=0}^{1} \dot{\theta}_i + l_3 \begin{pmatrix} -s_{012} \\ c_{012} \end{pmatrix} \sum_{i=1}^{2} \dot{\theta}_i. \tag{7}$$

It follows the second-order derivative which describe the arms Cartesian accelerations, where the serial links' Jacobian is assumed a non stationary matrix $\mathbf{J}_t \in \mathbb{R}^{2\times 3}$, such that

$$\begin{pmatrix} \ddot{x}_a \\ \ddot{y}_a \end{pmatrix} = \mathbf{J}_t \cdot \begin{pmatrix} \ddot{\theta}_0 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + \dot{\mathbf{J}}_t \cdot \begin{pmatrix} \dot{\boldsymbol{\theta}}_0 \\ \dot{\boldsymbol{\theta}}_1 \\ \dot{\boldsymbol{\theta}}_2 \end{pmatrix} \tag{8}$$

The ultimate purpose of this section is to merely establish kinematic models as a basis for the following sections. However, the essential focus of this research is the locomotive control of the mobile holonomic structure. At this point, the two-

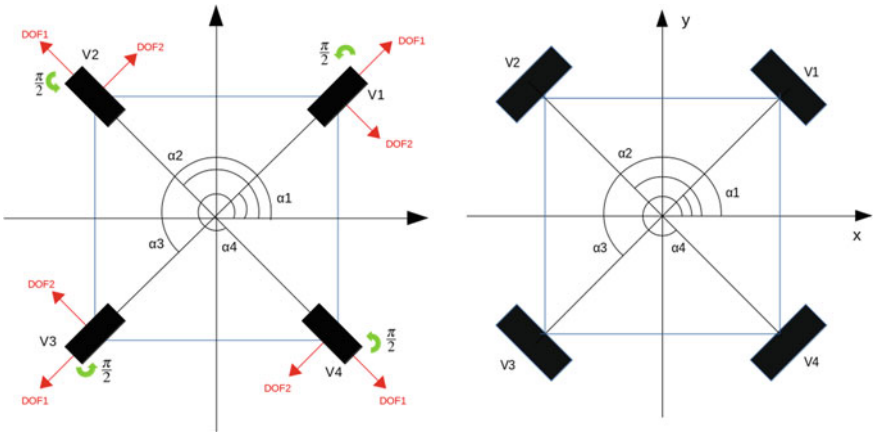


Fig. 4 4W4D holonomic kinematics. Mecanum wheels location without twist (left). Wheels positions and twisted $\pm\frac{\pi}{2}$ w.r.t. its center (right)

dimension manipulators can therefore exploit the native holonomic mobility such as position and rotation as to provide three-dimension spatial manipulator's trajectories. Additionally, omnidirectional mobility as a complement to the arms, allows arms' degrees of freedom complexity reduction.

Let us establish the following mobility kinematic constraints depicted in Fig. 4.

Therefore, without loss of generality let us state the following Proposition 2,

Proposition 2 (Holonomic motion model) *Let* $\mathbf{u}_t$ *be the robot state vector in its Cartesian form with components* (x, y) *orientation, such that* $\mathbf{u}_t \in \mathbb{R}^2$, $u = (x, y)^\top$. *Hence, the forward kinematics is*

$$\dot{\mathbf{u}} = r\mathbf{K} \cdot \dot{\boldsymbol{\Phi}}, \tag{9}$$

where, $\mathbf{K}$ *is a stationary kinematic control matrix containing geometrical parameters and* r *is the wheels radius. Let* $\boldsymbol{\Phi}_t = (\phi_1, \phi_2, \phi_3, \phi_4)^\top$ *be the four wheels angular velocity vector. Likewise, the backward kinematics where the constraints matrix is a non square system*

$$\dot{\boldsymbol{\Phi}} = \frac{1}{r} \cdot \mathbf{K}^+ \cdot \dot{\mathbf{u}} = \frac{1}{r} \cdot \mathbf{K}^\top (\mathbf{K} \cdot \mathbf{K}^\top)^{-1} \cdot \dot{\mathbf{u}}. \tag{10}$$

Therefore, according to geometry of Fig. 4 and the general models of previous Proposition 2, the following algebraic deduction arises, Cartesian speeds $\dot{x}$ and $\dot{y}$ in holonomic motion are obtained from wheels tangential velocities V_k expressed as,

$$\dot{x} = V_1 \cdot \cos\left(\alpha_1 - \frac{\pi}{2}\right) + V_2 \cdot \cos\left(\alpha_2 - \frac{\pi}{2}\right) + V_3 \cdot \cos\left(\alpha_3 - \frac{\pi}{2}\right) + V_4 \cdot \cos\left(\alpha_4 - \frac{\pi}{2}\right), \tag{11}$$

as well as

$$\dot{y} = V_1 \cdot \sin\left(\alpha_1 - \frac{\pi}{2}\right) + V_2 \cdot \sin\left(\alpha_2 - \frac{\pi}{2}\right) + V_3 \cdot \sin\left(\alpha_3 - \frac{\pi}{2}\right) + V_4 \cdot \sin\left(\alpha_4 - \frac{\pi}{2}\right). \tag{12}$$

Moreover, let V_k be the wheels tangential velocities described in terms of the angular speeds $\dot{\phi}_k$, such that the following equality is stated,

$$V_k = r \cdot \dot{\phi}_k \tag{13}$$

From where, the stationary non square kinematic control matrix $\mathbf{K}$ is provided by Definition 2,

Definition 2 (*4W4D holonomic kinematic matrix*) Each wheel with angle α_k w.r.t. the robot's geometric center, thus

$$\mathbf{K} = \begin{pmatrix} \cos(\alpha_1 - \frac{\pi}{2}) & \cos(\alpha_2 - \frac{\pi}{2}) & \cos(\alpha_3 - \frac{\pi}{2}) & \cos(\alpha_4 - \frac{\pi}{2}) \\ \sin(\alpha_1 - \frac{\pi}{2}) & \sin(\alpha_2 - \frac{\pi}{2}) & \sin(\alpha_3 - \frac{\pi}{2}) & \sin(\alpha_4 - \frac{\pi}{2}) \end{pmatrix}. \tag{14}$$

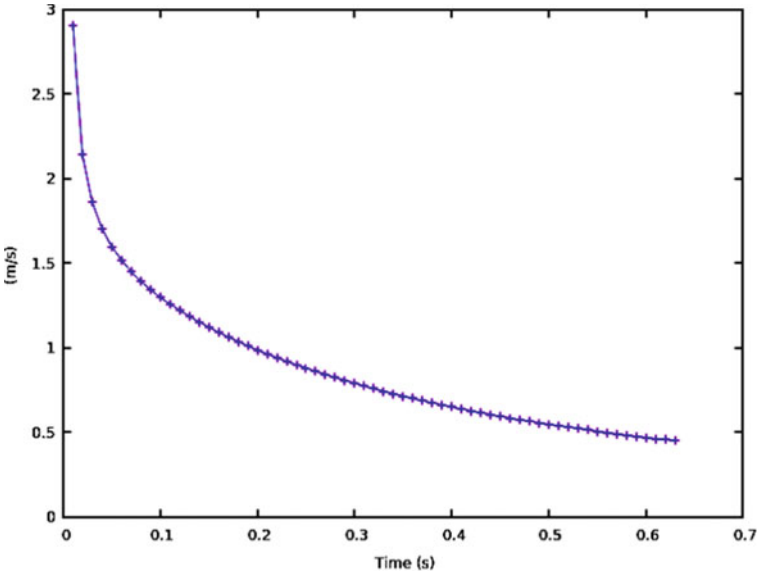
It follows that the speed holonomic model in as a function of wheels angular speeds and matrix $\mathbf{K}$ is

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = r \cdot \mathbf{K} \cdot \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{pmatrix}. \tag{15}$$

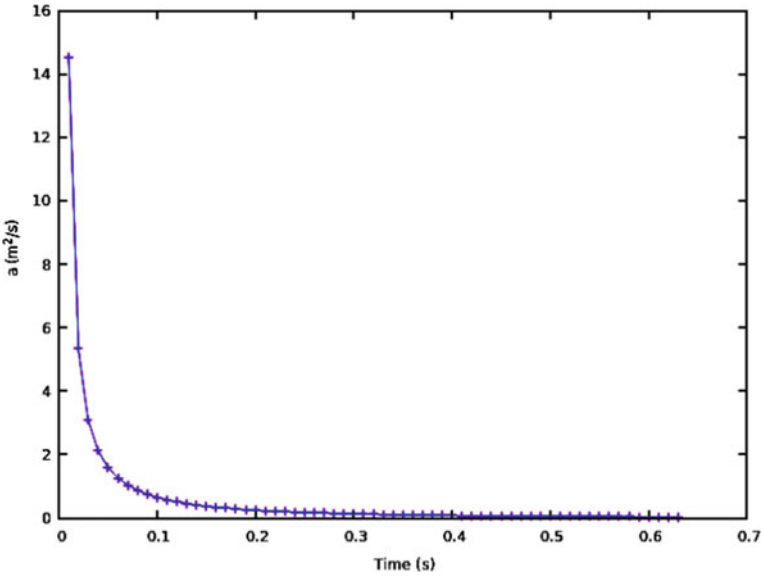
Similarly, from previous model higher-order derivatives are deduced for subsequent treatment for the sake of controller cascades building. Thus, the second-order kinematic model is

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = r \cdot \begin{pmatrix} \cos(\alpha_1 - \frac{\pi}{2}) & \cos(\alpha_2 - \frac{\pi}{2}) & \cos(\alpha_3 - \frac{\pi}{2}) & \cos(\alpha_4 - \frac{\pi}{2}) \\ \sin(\alpha_1 - \frac{\pi}{2}) & \sin(\alpha_2 - \frac{\pi}{2}) & \sin(\alpha_3 - \frac{\pi}{2}) & \sin(\alpha_4 - \frac{\pi}{2}) \end{pmatrix} \cdot \begin{pmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \ddot{\phi}_3 \\ \ddot{\phi}_4 \end{pmatrix}. \tag{16}$$

Likewise, a third-order derivative is provided by model



(a) First order derivative performance.



(b) Second order derivative performance.

Fig. 5 General higher order derivatives for 4W4D holonomic model. Velocity (above). Acceleration (below)

$$\begin{pmatrix} \dddot{x} \\ \dddot{y} \end{pmatrix} = r \cdot \begin{pmatrix} \cos(\alpha_1 - \frac{\pi}{2}) & \cos(\alpha_2 - \frac{\pi}{2}) & \cos(\alpha_3 - \frac{\pi}{2}) & \cos(\alpha_4 - \frac{\pi}{2}) \\ \sin(\alpha_1 - \frac{\pi}{2}) & \sin(\alpha_2 - \frac{\pi}{2}) & \sin(\alpha_3 - \frac{\pi}{2}) & \sin(\alpha_4 - \frac{\pi}{2}) \end{pmatrix} \cdot \begin{pmatrix} \dddot{\phi}_1 \\ \dddot{\phi}_2 \\ \dddot{\phi}_3 \\ \dddot{\phi}_4 \end{pmatrix}. \tag{17}$$

The fact that matrix **K** is stationary keeps simplistic the linear derivative expressions. For this type of four-wheel holonomic platforms, their kinematic models produce the following behavior curves shown in Fig. 5.

4 Observer Models

This section establishes the main sensing models that are assumed deterministic and in the cascade controller as elements of feedback for observing the robot's model state. It is worth saying that perturbation models and noisy sensor measurements and calibration methods are out of the scope of this manuscript's interest.

Thus, let us assume a pulse shaft encoder fixed for each wheel. Hence, let $\hat{\phi}_{\varepsilon_k}$ be a measurement of the angular position of the k^{th}-wheel,

$$\hat{\phi}_{\varepsilon_{k_t}}(\eta) = \frac{2\pi}{R}\eta_t, \tag{18}$$

where η_t is the instantaneous number of pulses detected while wheel is rotating. Let R be defined as the encoder angular resolution. Furthermore, the angular velocity encoder-based observation is given by the backward high-precision first-order derivative,

$$\dot{\hat{\phi}}_{\varepsilon}(\eta, t) = \frac{3\hat{\phi}_t - 4\hat{\phi}_{t-1} + \hat{\phi}_{t-2}}{(t_k - t_{k-1})(t_{k-1} - t_{k-2})}, \tag{19}$$

with three previous measurements of angle $\hat{\theta}_{\varepsilon}$ and time t_k. Hence, the kth wheel's tangential velocity is obtained by

$$\upsilon_k = \frac{\pi r}{R\Delta t}\left(3\eta_t - 4\eta_{t-1} + \eta_{t-2}\right), \tag{20}$$

where r is the wheel's radius and considering varying time loops, let $\Delta t \doteq (t_k - t_{k-1})(t_{k-1} - t_{k-2})$. Without loss of generality, let us substitute previous statements in Proposition 3 to describe Cartesian speeds observation, such that

Proposition 3 (Encoder-based velocity observer) *For simplicity, let us define the constants* $\beta_k \doteq \alpha_k - \frac{\pi}{2}$ *as constant angles for wheels orientation. Therefore, the encoder-based velocity observers* $\hat{\dot{x}}, \hat{\dot{y}}$ *are modeled by*

$$\hat{\dot{x}}_k = \sum_{k=1}^{4} \upsilon_k \sin(\beta_k) \tag{21}$$

and

$$\hat{\dot{y}}_k = \sum_{k=1}^{4} \upsilon_k \cos(\beta_k). \tag{22}$$

Moreover, the four wheels tangential speeds contribute to exert yaw motion w.r.t. the center of the robot. Thus, since by using encoders the wheels linear displacements can be inferred, then an encoder-based yaw observation $\hat{\theta}_\varepsilon$ is possible,

$$\hat{\theta}_\varepsilon = \frac{\pi r}{2L} \sum_{k=1}^{4} \eta_k, \tag{23}$$

where L is the robot's distance between any wheel and its center of rotation. Thus, the robot's angular velocity observer based only on the encoders measurements is

$$\hat{\dot{\theta}}_\varepsilon = \frac{\pi r}{4RL\Delta t} \sum_{k=1}^{4} \left(3\eta_{k_t} - 4\eta_{k_{t-1}} + \eta_{k_{t-2}}\right). \tag{24}$$

In addition, in order to decrease time-based derivative order of gyroscope's raw measurements $\dot{\theta}_g$, let us integrate sequence of raw measurements according to the Newton-Cotes integration approach as Definition 3,

Definition 3 (*Online sensor data integration*) The robot's yaw observation is inferred by time integration of raw angular velocity measurements, such that

$$\hat{\theta}_g = \int_{t_0}^{t_N} \hat{\dot{\theta}}_g \mathrm{d}t = \frac{t_N - t_0}{2N} \left(\hat{\dot{\theta}}_{g_0} + 2 \sum_{k=1}^{N-1} \hat{\dot{\theta}}_{g_k} + \hat{\dot{\theta}}_{g_N} \right), \tag{25}$$

with N available measurements in time segment $t_N - t_0$,

Furthermore, for the robot's yaw motion let us assume a fusion of encoders and inertial measurements about angle θ_t (inclinometer [rad]) and angular velocity ω_g (gyroscope [rad/s]), such that a complete robot's yaw rate observer is provided by Proposition 4.

Proposition 4 (Yaw deterministic observers) *The robot's angular velocity is an averaged model of three sensing models, encoder* $\hat{\theta}_\varepsilon$*, inclinometer* $\hat{\theta}_t$ *and gyroscope* $\hat{\theta}_g$ *such that*

$$\hat{\theta}_t = \frac{1}{3}\hat{\theta}_\varepsilon + \frac{1}{3}\hat{\theta}_t + \frac{1}{3}\int_{t_1}^{t_2} \hat{\dot{\theta}}_g dt, \tag{26}$$

where $\hat{\theta}_\varepsilon$ *is substituted by* (23)*. It follows that its first-order derivative,*

$$\hat{\omega}_t = \frac{1}{3}\hat{\dot{\theta}}_\varepsilon + \frac{1}{3}\frac{d\hat{\theta}_t}{dt} + \frac{1}{3}\hat{\dot{\theta}}_g, \tag{27}$$

where $\hat{\dot{\theta}}_\varepsilon$ *is substituted by* (24)*.*

5 Omnidirectional Cascade Controller

The relevant topic of this manuscript is a second-order cascade controller. It implies three nested control cycles. The highest frequency is the acceleration loop within the velocity cycle, and both in turn inside the loop for position, being the latter the slowest. The global cycle of position uses global references, while the rest of the inner cycles work on predictions as local references. The proposed cascade controller considers three feedback simultaneously and reduces the errors by recursive calculations and successive approximations with different sampling frequencies.

By stating the equation provided in Proposition 2 in the form of differential equation

$$\frac{d\mathbf{u}}{dt} = r \cdot \mathbf{K} \cdot \frac{d\boldsymbol{\Phi}}{dt} \tag{28}$$

and solving according to the following expression, where both differentials dt are reduced, and by integrating w.r.t. time establishing the limits in both sides of equation:

$$\int_{\mathbf{u}_1}^{\mathbf{u}_2} d\mathbf{u}dt = r \cdot \mathbf{K} \int_{\boldsymbol{\Phi}_1}^{\boldsymbol{\Phi}_2} d\boldsymbol{\Phi}dt, \tag{29}$$

resulting in the following equality:

$$\mathbf{u}_2 - \mathbf{u}_1 = r \cdot \mathbf{K} \cdot (\boldsymbol{\Phi}_2 - \boldsymbol{\Phi}_1). \tag{30}$$

Therefore, considering the Moore-Penrose approach to obtain the pseudoinverse of the non square stationary kinematic matrix $\mathbf{K}$ and by solving and algebraically arranging using continuous-time notation, for and expression in terms of a recursive backward solution,

$$\boldsymbol{\Phi}_{t+1} = \boldsymbol{\Phi}_t + \frac{1}{r} \cdot \mathbf{K}^T(\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\mathbf{u}_{ref} - \hat{\mathbf{u}}_t), \tag{31}$$

where, the prediction vector $\mathbf{u}_{t+1}$ is re-formulated as the global reference $\mathbf{u}_{ref}$ or the goal the robot is desirable to reach. Likewise, for the forward kinematic solution $\mathbf{u}_{t+1}$ is

$$\mathbf{u}_{t+1} = \mathbf{u}_t + r \cdot \mathbf{K} \cdot (\boldsymbol{\Phi}_{t+1} - \hat{\boldsymbol{\Phi}}_t). \tag{32}$$

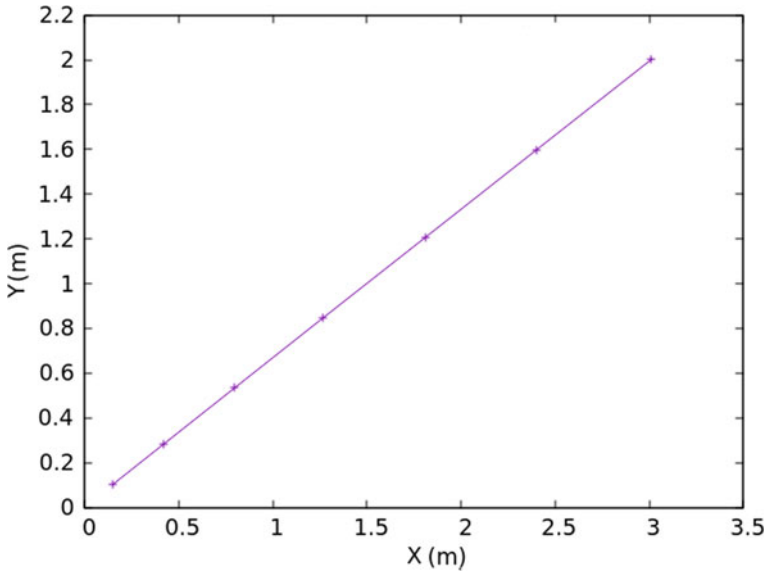


Fig. 6 A global cascade recursive simulation for Cartesian position

Therefore, the first global controller cascade is formed by means of the pair of recursive expressions (31) and (32). Proposition 5 highlights the global cascade by.

Proposition 5 (Feedback position cascade) *Given the inverse kinematic motion with observation in the workspace* $\hat{\mathbf{u}}_t$

$$\boldsymbol{\Phi}_{t+1} = \boldsymbol{\Phi}_t + \frac{1}{r} \cdot \mathbf{K}^T (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\mathbf{u}_{ref} - \hat{\mathbf{u}}_t), \tag{33}$$

and direct kinematic motion with observation in the control variables space $\hat{\boldsymbol{\Phi}}_t$,

$$\mathbf{u}_{t+1} = \mathbf{u}_t + r \cdot \mathbf{K} \cdot (\boldsymbol{\Phi}_{t+1} - \hat{\boldsymbol{\Phi}}_t). \tag{34}$$

Proposition 5 is validated through numerical simulations that are shown in Fig. 6 the automatic Cartesian segments and feedback position errors decreasing.

Without loss of generality and following the same approach as Proposition 5, let us represent a second controller cascade to control velocity. Thus, the following equation expresses the second-order derivative kinematic expression given in (16) as a differential equation,

$$\frac{d\dot{\mathbf{u}}}{dt} = r \cdot \mathbf{K} \cdot \frac{d\dot{\boldsymbol{\Phi}}}{dt} \tag{35}$$

and solving definite integrals,

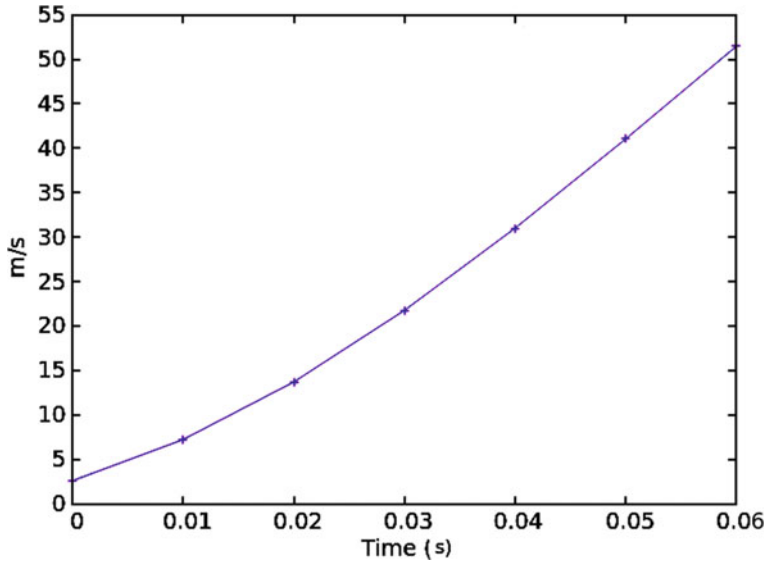


Fig. 7 Numerical simulation for second cascade inner recursive in terms of velocities

$$\int_{\dot{\mathbf{u}}_1}^{\dot{\mathbf{u}}_2} d\dot{\mathbf{u}}dt = r \cdot \mathbf{K} \int_{\dot{\boldsymbol{\Phi}}_1}^{\dot{\boldsymbol{\Phi}}_2} d\dot{\boldsymbol{\Phi}}dt, \tag{36}$$

and similarly obtaining the following first-order equality,

$$\dot{\mathbf{u}}_2 - \dot{\mathbf{u}}_1 = r \cdot \mathbf{K} \cdot (\dot{\boldsymbol{\Phi}}_2 - \dot{\boldsymbol{\Phi}}_1). \tag{37}$$

It follows the Proposition 6 establishing the second cascade controlling the first-order derivatives.

Proposition 6 (Feedback velocity cascade) *The backwards kinematic recursive function with in-loop velocity observers and prediction* $\dot{\mathbf{u}}_{t+1}$ *used as local reference* $\dot{\mathbf{u}}_{ref}$ *is given by*

$$\dot{\boldsymbol{\Phi}}_{t+1} = \dot{\boldsymbol{\Phi}}_t + \frac{1}{r} \cdot \mathbf{K}^T \cdot (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\dot{\mathbf{u}}_{ref} - \hat{\dot{\mathbf{u}}}_t), \tag{38}$$

likewise the forward speeds kinematic model,

$$\dot{\mathbf{u}}_{t+1} = \dot{\mathbf{u}}_t + r \cdot \mathbf{K} \cdot (\dot{\boldsymbol{\Phi}}_{t+1} - \hat{\dot{\boldsymbol{\Phi}}}_t). \tag{39}$$

Proposition 6 is validated by simulation Fig. 7.

Similarly, the third-order model from Eq. (17),

$$\dddot{\mathbf{u}} = \mathbf{K} \cdot \dddot{\boldsymbol{\Phi}} \tag{40}$$

and stated as a differential equation,

$$\frac{d\ddot{\mathbf{u}}}{dt} = r \cdot \mathbf{K} \cdot \frac{d\ddot{\boldsymbol{\Phi}}}{dt}, \tag{41}$$

which is solved by definite integral in both sides of equality

$$\int_{\ddot{\mathbf{u}}_1}^{\ddot{\mathbf{u}}_2} d\ddot{\mathbf{u}} dt = r \cdot \mathbf{K} \int_{\ddot{\boldsymbol{\Phi}}_1}^{\ddot{\boldsymbol{\Phi}}_2} d\ddot{\boldsymbol{\Phi}} dt, \tag{42}$$

thus, a consistent second-order derivative (acceleration) equality is obtained,

$$\ddot{\mathbf{u}}_2 - \ddot{\mathbf{u}}_1 = r \cdot \mathbf{K} \cdot (\ddot{\boldsymbol{\Phi}}_2 - \ddot{\boldsymbol{\Phi}}_1). \tag{43}$$

Therefore, the following Proposition 7 is provided and notation rearranged for a third recursive inner control loop in terms of accelerations.

Proposition 7 (Feedback acceleration cascade) *The backwards kinematic recursive function with in-loop acceleration observers and prediction* $\ddot{\boldsymbol{\Phi}}_{t+1}$ *used as local reference* $\ddot{\mathbf{u}}_{ref}$ *is given by*

$$\ddot{\boldsymbol{\Phi}}_{t+1} = \ddot{\boldsymbol{\Phi}}_t + \frac{1}{r} \cdot \mathbf{K}^T \cdot (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\ddot{\mathbf{u}}_{ref} - \hat{\ddot{\mathbf{u}}}_t). \tag{44}$$

Additionally, the forward acceleration kinematic model is

$$\ddot{\mathbf{u}}_{t+1} = \ddot{\mathbf{u}}_t + r \cdot \mathbf{K} \cdot (\ddot{\boldsymbol{\Phi}}_{t+1} - \hat{\ddot{\boldsymbol{\Phi}}}_t), \tag{45}$$

Proposition 7 is validated through numerical simulation of Fig. 8 showing an arbitrary performance.

At this point is worth highlighting a general convergence criteria for recursive control loops. The cycles end up by satisfying feedback error numerical precision $\varepsilon_{u,\Phi}$, such that.

Definition 4 (*Convergence criteria*) When the feedback error numerically meets a local general reference according to the limit

$$\lim_{\Delta\Phi \to 0} \left(\boldsymbol{\Phi}^{ref} - \hat{\boldsymbol{\Phi}} \right) = 0,$$

where the feedback error is $\mathbf{e}_\Phi = (\boldsymbol{\Phi}^{ref} - \hat{\boldsymbol{\Phi}})$ that numerically will nearly approach zero. Thus, given the criterion $\mathbf{e}_\Phi < \varepsilon_\Phi$ is a

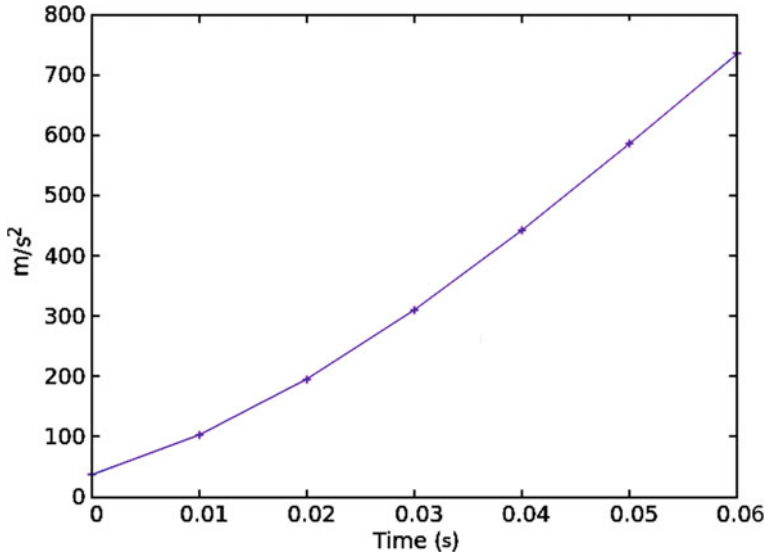


Fig. 8 Numerical simulation for third cascade inner recursive in terms of accelerations

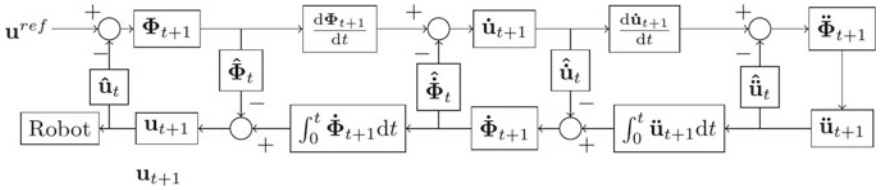


Fig. 9 Second-order cascade controller block diagram

$$\left\| \frac{\boldsymbol{\Phi}_{ref} - \hat{\boldsymbol{\Phi}}_t}{\boldsymbol{\Phi}_{ref}} \right\| < \varepsilon. \tag{46}$$

Although, previous Definition 4 was described as a criterion for Φ, it has general applicability being subjected to condition any other control variable in process.

Therefore, given Propositions 5, 6 and 7, which establish the different recursive control loops for each order of derivative involved, Fig. 9 shows cascades coupling forming the controller.

Therefore, Fig. 9 is summarized in Table 1 depicting only the coupling order of solutions provided in Propositions 5–7.

Essentially, the coupling element between steps 1 and 2 of Table 1 is a first order derivative of wheels angular velocities w.r.t. time. Likewise, the following list briefly describes the observers or sensing models interconnecting every step in the controller

Table 1 Cascade-coupled recursive controllers

Steps	Equation
1	$\boldsymbol{\Phi}_{t+1} = \boldsymbol{\Phi}_t + \frac{1}{r} \cdot \mathbf{K}^T \left(\mathbf{K} \cdot \mathbf{K}^T\right)^{-1} \cdot (\mathbf{u}_{ref} - \hat{\mathbf{u}}_t)$
2	$\dot{\mathbf{u}}_{t+1} = \dot{\mathbf{u}}_t + r \cdot \mathbf{K} \cdot (\dot{\boldsymbol{\Phi}}_{t+1} - \hat{\dot{\boldsymbol{\Phi}}}_t)$
3	$\ddot{\boldsymbol{\Phi}}_{t+1} = \ddot{\boldsymbol{\Phi}}_t + \frac{1}{r} \cdot \mathbf{K}^T (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\ddot{\mathbf{u}}_{t+1} - \hat{\ddot{\mathbf{u}}}_t)$
4	$\ddot{\mathbf{u}}_{t+1} = \ddot{\mathbf{u}}_t + r \cdot \mathbf{K} \cdot (\ddot{\boldsymbol{\Phi}}_{t+1} - \hat{\ddot{\boldsymbol{\Phi}}}_t)$
5	$\dot{\boldsymbol{\Phi}}_{t+1} = \dot{\boldsymbol{\Phi}}_t + \frac{1}{r} \cdot \mathbf{K}^T (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\dot{\mathbf{u}}_{t+1} - \hat{\dot{\mathbf{u}}}_t)$
6	$\mathbf{u}_{t+1} = \mathbf{u}_t + r \cdot \mathbf{K} \cdot (\boldsymbol{\Phi}_{t+1} - \hat{\boldsymbol{\Phi}}_t)$

cascades. Basically, it is about variational operations with fundamentals in numerical derivation and integration to transform sensor data into consistent physical units.

1. Feedback from step 1 to step 2:

$$\dot{\boldsymbol{\Phi}}_{t+1} = \frac{d}{dt}\boldsymbol{\Phi}_{t+1} = \frac{3\hat{\boldsymbol{\Phi}}_t - 4\hat{\boldsymbol{\Phi}}_{t-1} + \hat{\boldsymbol{\Phi}}_{t-2}}{\Delta t},$$

2. Feedback from step 2 to step 3:

$$\ddot{\mathbf{u}}_{t+1} = \frac{d}{dt}\dot{\mathbf{u}}_{t+1} = \frac{3\hat{\dot{\mathbf{u}}}_t - 4\hat{\dot{\mathbf{u}}}_{t-1} + \hat{\dot{\mathbf{u}}}_{t-2}}{\Delta t},$$

3. Feedback from step 4 to step 5:

$$\dot{\mathbf{u}}_{t+1} = \int_{t_0}^{t} \ddot{\mathbf{u}}_{t+1} dt = \frac{t_n - t_0}{2 \cdot n_t} \cdot (\hat{\ddot{\mathbf{u}}}_0 + 2\sum_{i=1}^{n-1} \hat{\ddot{\mathbf{u}}}_i + \hat{\ddot{\mathbf{u}}}_k),$$

4. Feedback from step 5 to step 6:

$$\boldsymbol{\Phi}_{t+1} = \int_{t_0}^{t} \dot{\boldsymbol{\Phi}}_{t+1} dt = \frac{t_n - t_0}{2 \cdot n_t} \cdot (\hat{\dot{\boldsymbol{\Phi}}}_0 + 2\sum_{i=1}^{n-1} \hat{\dot{\boldsymbol{\Phi}}}_i + \hat{\dot{\boldsymbol{\Phi}}}_k),$$

Additionally, the following listing Algorithm 1 is the controller in pseudocode notation.

Data: $\epsilon, \mathbf{K}, \mathbf{u}_{ref}, \mathbf{u}_t, \hat{\mathbf{u}}_t, \dot{\mathbf{u}}_t, \hat{\dot{\mathbf{u}}}_t, \ddot{\mathbf{u}}_t, \hat{\ddot{\mathbf{u}}}_t, \boldsymbol{\Phi}_t, \hat{\boldsymbol{\Phi}}_t, \hat{\dot{\boldsymbol{\Phi}}}_t, \ddot{\boldsymbol{\Phi}}_t, \hat{\ddot{\boldsymbol{\Phi}}}_t$
$\mathbf{u}_{ref} = (x_i, y_i)^T$;
while $(\mathbf{u}_{ref} - \mathbf{u}_t) < \epsilon_u$ **do**
  $\boldsymbol{\Phi}_{t+1} = \boldsymbol{\Phi}_t + \frac{1}{r} \cdot \mathbf{K}^T (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\mathbf{u}_{ref} - \hat{\mathbf{u}}_t)$;
  $\frac{d}{dt}\boldsymbol{\Phi}_{t+1} = \frac{3\hat{\boldsymbol{\Phi}}_t - 4\hat{\boldsymbol{\Phi}}_{t-1} + \hat{\boldsymbol{\Phi}}_{t-2}}{\Delta t}$;
  while $(\dot{\boldsymbol{\Phi}}_{t+1} - \hat{\dot{\boldsymbol{\Phi}}}_t) < \varepsilon_{\dot{\Phi}}$ **do**
    $\dot{\mathbf{u}}_{t+1} = \dot{\mathbf{u}}_t + r \cdot \mathbf{K} \cdot (\dot{\boldsymbol{\Phi}}_{t+1} - \hat{\dot{\boldsymbol{\Phi}}}_t)$;
    $\frac{d}{dt}\dot{\mathbf{u}}_{t+1} = \frac{3\hat{\dot{\mathbf{u}}}_t - 4\hat{\dot{\mathbf{u}}}_{t-1} + \hat{\dot{\mathbf{u}}}_{t-2}}{\Delta t}$;
    while $(\ddot{\mathbf{u}}_{t+1} - \hat{\ddot{\mathbf{u}}}_t) < \varepsilon_{\ddot{u}}$ **do**
      $\ddot{\boldsymbol{\Phi}}_{t+1} = \ddot{\boldsymbol{\Phi}}_t + \frac{1}{r} \cdot \mathbf{K}^T \cdot (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\ddot{\mathbf{u}}_{t+1} - \hat{\ddot{\mathbf{u}}}_t)$;
      $\ddot{\mathbf{u}}_{t+1} = \ddot{\mathbf{u}}_t + r \cdot \mathbf{K} \cdot (\ddot{\boldsymbol{\Phi}}_{t+1} - \hat{\ddot{\boldsymbol{\Phi}}}_t)$;
      $\int_a^b \ddot{\mathbf{u}}_{t+1} dt = \frac{b-a}{2 \cdot n} \cdot (\hat{\ddot{\mathbf{u}}}_0 + 2 \cdot \sum_{j=1}^{j_{n-1}} \hat{\ddot{\mathbf{u}}}_{k_j} + \hat{\ddot{\mathbf{u}}}_{k_n})$;
    end
    $\dot{\boldsymbol{\Phi}}_{t+1} = \dot{\boldsymbol{\Phi}}_t + \frac{1}{r} \cdot \mathbf{K}^T \cdot (\mathbf{K} \cdot \mathbf{K}^T)^{-1} \cdot (\dot{\mathbf{u}}_{t+1} - \hat{\dot{\mathbf{u}}}_t)$;
    $\int_a^b \dot{\boldsymbol{\Phi}}_{t+1} dt = \frac{b-a}{2 \cdot n} \cdot (\hat{\dot{\boldsymbol{\Phi}}}_0 + 2 \cdot \sum_{j=1}^{j_{n-1}} \hat{\dot{\boldsymbol{\Phi}}}_{k_j} + \hat{\dot{\boldsymbol{\Phi}}}_{k_n})$;
  end
  $\mathbf{u}_{t+1} = \mathbf{u}_t + r \cdot \mathbf{K} \cdot (\boldsymbol{\Phi}_{t+1} - \hat{\boldsymbol{\Phi}}_t)$;
end

Algorithm 1: Second-order three cascade controller pseudocode

The following Figures of Sect. 6 show the numerical simulation results under a controlled scheme. The robot navigate to different Cartesian positions and within trajectory segments the cascade controller is capable to control position, then controls the velocity exerted within such segment of distance, and similarly the acceleration is controlled within a small such velocity-window that is being controlled.

6 Results Analysis and Discussion

The three cascade controllers required couplings between them through numerical derivations and integrations overtime. In this case, backwards high precision derivatives and Newton-Cotes integration were used. Although, the traditional PID also deploys three derivative orders, the use of them is by far different in their implementation. The proposed cascade model worked considerably stable, reliable and numerically precise.

A main fundamental in the proposed method is that three loops are nested. The slowest loop establishes a current metric error distance toward a global vector reference. Then, the second and third nested control loops establish local reference velocity and acceleration, both corresponding to the actual incremental distance. The three loops are conditioned to recursively reduce errors up to a numerical precision value by means of successive approximations.

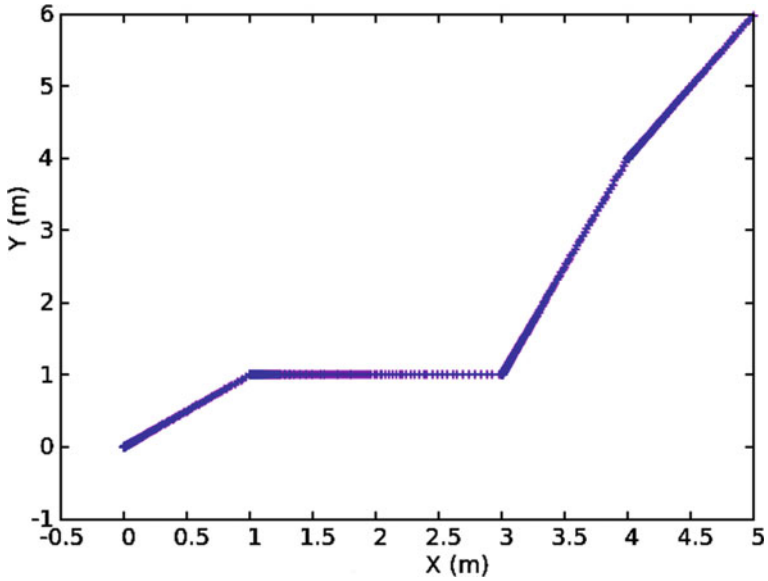


Fig. 10 Controlled Cartesian position along a trajectory compounded of four global points

For instance, Fig. 10 shows a nonlinear Cartesian trajectory comprised of four main linear local segments with different slopes each.

The first cascade loop basically controls the local displacements in between couples of global points. Figure 11 shows four peaks that represent the frontier of each global point reached. The controller locally starts metric displacement after getting each goal point, and approaches the next one by means of successive approximations producing nonlinear rates of motion.

The second inner looped cascade controls the robot's Cartesian velocities w.r.t. time as shown in Fig. 12. At each goal reached the first-order derivative shows local minimal or maximal with magnitudes depending on the speeds due to the loop reference values (local predictions as references). where the last Trajectory's point reached is the global control's reference. In this case, the velocity references are local values to be fitted, or also known as the predicted values for the next local loop calculation at $t + 1$, such as $\dot{\mathbf{u}}_{t+1}$ or $\dot{\boldsymbol{\Phi}}_{t+1}$. As the velocity control loop is inside the metric displacement loop, the velocity is controlled by a set of loops, only for a segment of velocity, the one that is being calculated in the current displacement's loop.

The third inner control loop that manages the second-order derivative produces the results shown in Fig. 13. This control loop is the fastest iterative process, which exerts the higher sampling frequency. In this case, the acceleration references are local values to be fitted, or the predicted values for the next local loop calculation, such as $\ddot{\mathbf{u}}_{t+1}$ or $\ddot{\boldsymbol{\Phi}}_{t+1}$. As the acceleration control loop is inside the velocity loop, the

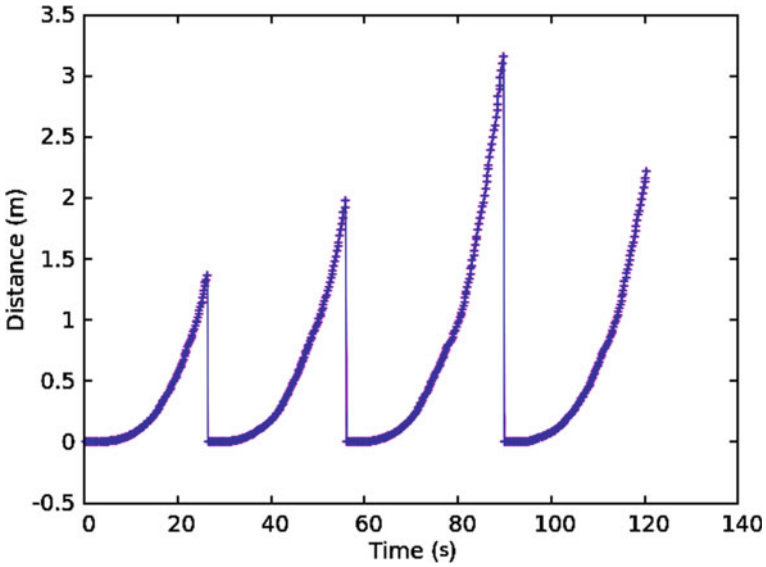


Fig. 11 Controlled displacement performance

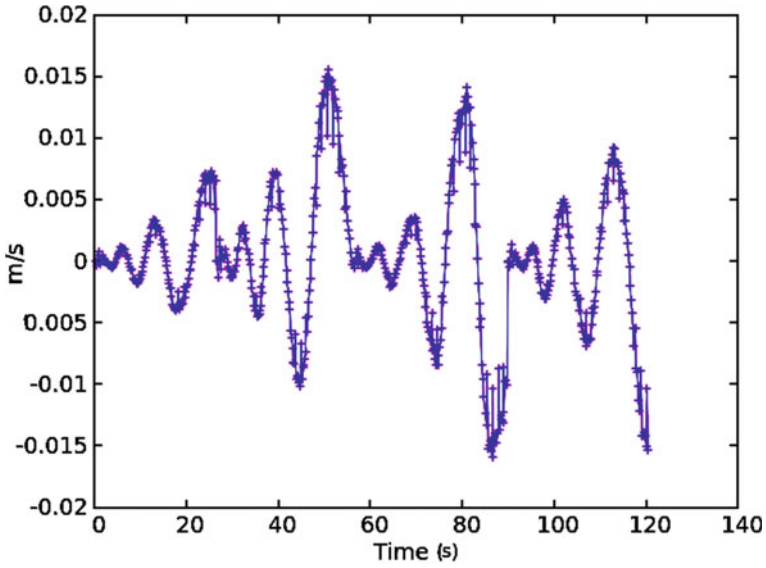


Fig. 12 Controlled linear velocity

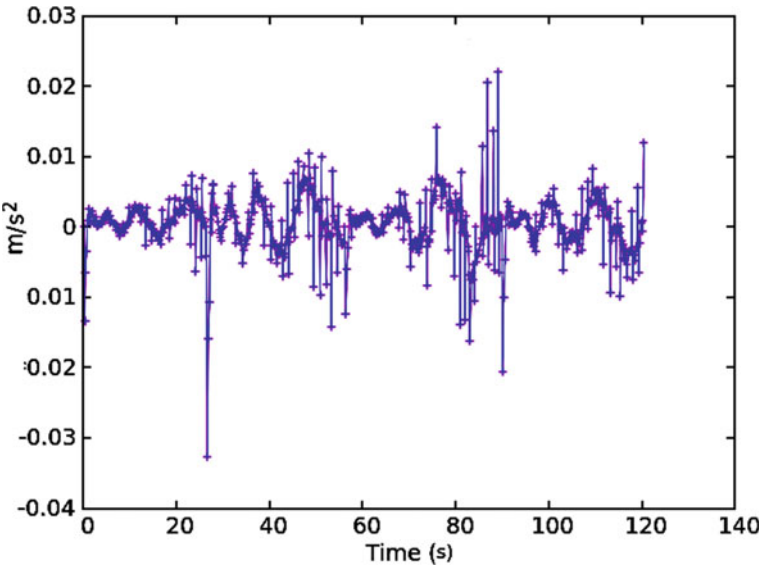


Fig. 13 Controlled acceleration

acceleration is controlled by a set of loops, only for a segment of velocity, the one that is being calculated in the current velocity's loop.

Finally, the observers that provided feedback were stated as a couple of single expression representing a feasible model of sensor fusion (summarized by Proposition 4). The robot's angular motion (angle and yaw velocity) combined wheels motion and inertial movements into a compounded observer model. The in-loop transition between numerical derivatives worked highly reliably. The multiple inner control cascades approach resulted numerically accurate, working online considerably fast. Although, this type of cascade controller has the advantage that input, reference and state vectors and matrices can easily be augmented without any alteration to the algorithm, but if compared with PID controller in terms of speed, the latter is faster due to less computational complexity.

7 Conclusions

The natural kinematic constraints of a genuinely omnidirectional platform always produce straight paths. It is an advantage because it facilitates its displacement. In strictly necessary situations, to generate deliberately curved or even discontinuous displacements, an omnidirectional platform traverses it through linearization of infinitesimal segments.

The proposed cascade control model was derived directly from the physics of the robotic mechanism. In such an approach, the kinematic model was obtained as an independent cascade and established as a proportional control type with a constant gain represented by the robot's geometric parameters. The gain or convergence factor resulted in a non-square stationary matrix (MIMO). Unlike a PID controller, the inverse analytic solution was formulated to obtain a system of linear differential equations. In its solution, definite integration produced a recursive controller, which converged to a solution by successive approximations of the feedback error.

The strategy proposed in this work focuses on connecting all the higher order derivatives of the system in nested forms (cascades), unlike a PID controller which is described as a linear summation of all derivative orders. Likewise, a cascade approach does not need gain adjustment.

The lowest order derivative was organized in the outer cascade. Being the loop with the slowest control cycle frequency and containing the global control references (desired positions). Further, the intermediate cascade is a derivative with the next higher order, and is governed by a local speed reference. That is, this cascade controls the speed during the displacement segment that has projected the cycle of the external cascade. Finally, the acceleration cascade cycle is the fastest loop and controls the portions of acceleration during a small interval of displacement along the trajectory towards the next Cartesian goal.

The proposed research evidenced a good performance, showing controlled limits of disturbances due to the three controllers acting over a same portion of motion. The controller was robust and the precision error ε allowed to adjust the accuracy of the robot goal closeness.

Acknowledgements The corresponding author acknowledges the support of Laboratorio de Robótica. The third and fourth authors acknowledge the support of the Kazan Federal University Strategic Academic Leadership Program ('PRIORITY-2030').

References

1. Mutalib, M. A. A., & Azlan, N. Z. (2020). Prototype development of mecanum wheels mobile robot: A review. *Applied Research and Smart Technology, 1*(2), 71–82, ARSTech. https://doi.org/10.23917/arstech.v1i2.39.
2. Yadav P. S., Agrawal V., Mohanta J. C., & Ahmed F. (2022) A theoretical review of mobile robot locomotion based on mecanum wheels. *Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy, 9*(2), Evergreen.
3. Palacín, J., Clotet, E., Martínez, D., Martínez, D., & Moreno, J. (2019). Extending the application of an assistant personal Robot as a Walk-Helper Tool. *Robotics, 8*(27), MDPI. https://doi.org/10.3390/robotics8020027.
4. Cooper S., Di Fava A., Vivas C., Marchionni L., & Ferro F. (2020). ARI: The social assistive robot and companion. In *29th IEEE International Conferences on Robot and Human Interactive Communication*, Naples Italy, August 31–September 4. https://doi.org/10.1109/RO-MAN47096.2020.9223470.

5. Li, Y., Dai, S., Zheng, Y., Tian, F., & Yan, X. (2018). Modeling and kinematics simulation of a mecanum wheel platform in RecurDyn. *Journal of Robotics Hindawi*. https://doi.org/10.1155/2018/9373580
6. Rohrig, C., Hes, D., & Kunemund, F. (2017). Motion controller design for a mecanum wheeled mobile manipulator. In *2017 IEEE Conferences on Control Technology and Applications*, USA (pp. 444–449), August 27–30. https://doi.org/10.1109/ccta.2017.8062502.
7. Park J., Koh D., Kim J., & Kim C. (2021). Vibration reduction control of omnidirectional mobile robot with lift mechanism. In *21st International Conferences on Control, Automation and Systems*. https://doi.org/10.23919/ICCAS52745.2021.9649932.
8. Belmonte, Á., Ramón, J. L., Pomares, J., Garcia, G. J., & Jara, C. A. (2019). Optimal image-based guidance of mobile manipulators using direct visual servoing. *Electronics, 8*(374). https://doi.org/10.3390/electronics8040374.
9. Yang, F., Shi, Z., Ye, S., Qian, J., Wang, W., & Xuan D. (2022). VaRSM: Versatile autonomous racquet sports machine. In *ACM/IEEE 13th International Conferences on Cyber-Physical Systems*, Milano Italy, May 4–6. https://doi.org/10.1109/ICCPS54341.2022.00025.
10. Eirale A., Martini M., Tagliavini L., Gandini D., Chiaberge M., & Quaglia G. (2022). Marvin: an innovative omni-directional robotic assistant for domestic environments. arXiv:2112.05597, https://doi.org/10.48550/arXiv.2112.05597.
11. Qian J., Zi B., Wang D., Ma Y., & Zhang D. (2017). The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system. *Sensors, 17* (2073). https://doi.org/10.3390/s17092073.
12. Zalevsky, A., Osipov, O., & Meshcheryakov, R. (2017). Tracking of warehouses robots based on the omnidirectional wheels. In *International Conferences on Interactive Collaborative Robotics* (pp. 268–274). Springer. https://doi.org/10.1007/978-3-319-66471-2_29.
13. Rauniyar A., Upreti H. C., Mishra A., & Sethuramalingam P. (2021). MeWBots: Mecanum-Wheeled robots for collaborative manipulation in an obstacle-clustered environment without communication. *J. of Intelligent & Robotic Systems, 102*(1). https://doi.org/10.1007/s10846-021-01359-5.
14. Zhou, J., Wang, J., He, J., Gao, J., Yang, A., & Hu, S. (2022). A reconfigurable modular vehicle control strategy based on an improved artificial potential field. *Electronics, 11*(16), 2539. https://doi.org/10.3390/electronics11162539
15. Tanioka, T. (2019). Nursing and rehabilitative care of the elderly using humanoid robot. *The Journal of Medical Investigation, 66*,. https://doi.org/10.2152/jmi.66.19
16. Shepherd, S., & Buchstab, A. (2014). KUKA Robots On-Site. In W. McGee and M. Ponce de Leon M (Eds.), *Robotic Fabrication in Architecture, Art and Design* (pp. 373–380). Cham: Springer. https://doi.org/10.1007/978-3-319-04663-1_26.
17. Taheri, H., & Zhao, C. X. (2020). Omnidirectional mobile robots, mechanisms and navigation approaches. *Mechanism and Machine Theory, 153*(103958), Elsevier. https://doi.org/10.1016/j.mechmachtheory.2020.103958.
18. Slimane Tich Tich, A., Inel, F., & Carbone, G. (2022). Realization and control of a mecanum wheeled robot based on a kinematic model. In V. Niola, A. Gasparetto, G. Quaglia & G. Carbone (Eds.), *Advances in Italian Mechanism Science, IFToMM Italy, Mechanisms and Machine Science* (Vol. 122). Cham: Springer. https://doi.org/10.1007/978-3-031-10776-4_77.
19. Thai, N. H., Ly, T. T. K., & Dzung, L. Q. (2022). Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller. *Bulletin of Electrical Engineering and Informatics, 11*(4), 1902–1910. https://doi.org/10.11591/eei.v11i4.3712
20. Han K., Kim H., & Lee J. S. (2010). The sources of position errors of omni-directional mobile robot with mecanum wheel. In *IEEE International Conferences on Systems, Man and Cybernetics*, October 10–13, Istanbul, Turkey (pp. 581–586). https://doi.org/10.1109/ICSMC.2010.5642009.
21. Palacín J., Rubies E., & Clotet E. (2022). Systematic odometry error evaluation and correction in a human-sized three-wheeled omnidirectional mobile robot using flower-shaped calibration trajectories. *Applied Sciences, 12*(5), 2606, MDPI. https://doi.org/10.3390/app12052606.

22. Cavacece, M., Lanni, C., & Figliolini, G. (2022). Mechatronic design and experimentation of a mecanum four wheeled mobile robot. In: V. Niola, A. Gasparetto, G. Quaglia & G. Carbone G. (Eds.) *Advances in Italian Mechanism Science. IFToMM Italy 2022. Mechanisms and Machine Science* (Vol. 122). Cham: Springer. https://doi.org/10.1007/978-3-031-10776-4_93.
23. Lin, P., Liu, D., Yang, D., Zou, Q., Du, Y., & Cong, M. (2019). Calibration for odometry of omnidirectional mobile robots based on kinematic correction. In *IEEE 14th International Conferences on Computer Science & Education*, August 19–21, Toronto, Canada (pp. 139–144). https://doi.org/10.1109/iccse.2019.8845402.
24. Maddahi, Y., Maddahi, A., & Sepehri, N. (2013). Calibration of omnidirectional wheeled mobile robots: Method and experiments. In *Robotica* (Vol. 31, pp. 969–980). Cambridge University Press. https://doi.org/10.1017/S0263574713000210.
25. Ma'arif, I. A., Raharja, N. M., Supangkat, G., Arofiati, F., Sekhar, R., & Rijalusalam, D.U. (2021). PID-based with odometry for trajectory tracking control on four-wheel omnidirectional Covid-19 aromatherapy robot. *Emerging Science Journal, 5*. SI "COVID-19: Emerging Research". https://doi.org/10.28991/esj-2021-SPER-13.
26. Li, Y., Ge, S., Dai, S., Zhao, L., Yan, X., Zheng, Y., & Shi, Y. (2020). Kinematic modeling of a combined system of multiple mecanum-wheeled robots with velocity compensation. *Sensors, 20*(75), MDPI. https://doi.org/10.3390/s20010075.
27. Savaee E., & Hanzaki A. R. (2021). A new algorithm for calibration of an omni-directional wheeled mobile robot based on effective kinematic parameters estimation. *Journal of Intelligent & Robotic Systems, 101*(28), Springer. https://doi.org/10.1007/s10846-020-01296-9.
28. Khoygani, M. R. R., Ghasemi, R., & Ghayoomi, P. (2021). Robust observer-based control of nonlinear multi-omnidirectional wheeled robot systems via high order sliding-mode consensus protocol. *International Journal of Automation and Computing, 18*, 787–801, Springer, https://doi.org/10.1007/s11633-020-1254-z.
29. Almasri, E., & Uyguroğlu, M. K. (2021). Modeling and trajectory planning optimization for the symmetrical multiwheeled omnidirectional mobile robot. *Symmetry, 13*(1033), MDPI. https://doi.org/10.3390/sym13061033.
30. Rijalusalam, D.U., & Iswanto, I. (2021). Implementation kinematics modeling and odometry of four omni wheel mobile robot on the trajectory planning and motion control based microcontroller. *Journal of Robotics and Control, 2*(5). https://doi.org/10.18196/jrc.25121.
31. Alshorman, A. M., Alshorman, O., Irfan, M., Glowacz, A., Muhammad, F., & Caesarendra, W. (2020). Fuzzy-Based fault-tolerant control for omnidirectional mobile robot. *Machines, 8*(3), 55, MDPI. https://doi.org/10.3390/machines8030055.
32. Szeremeta, M., & Szuster, M. (2022). Neural tracking control of a four-wheeled mobile robot with mecanum wheels. *Applied Science, 2022*(12), 5322, MDPI. https://doi.org/10.3390/app12115322.
33. Vlantis, P., Bechlioulis, C. P., Karras, G., Fourlas, G., & Kyriakopoulos, K. J. (2016). Fault tolerant control for omni-directional mobile platforms with 4 mecanum wheels. In *IEEE International Conferences on Robotics and Automation* (pp. 2394–2400). https://doi.org/10.1109/icra.2016.7487389.
34. Wu, X., & Huang, Y. (2021). Adaptive fractional-order non-singular terminal sliding mode control based on fuzzy wavelet neural networks for omnidirectional mobile robot manipulator. *ISA Transactions*. https://doi.org/10.1016/j.isatra.2021.03.035
35. Pizá, R., Carbonell, V., Casanova, Á., Cuenca, J. J., & Salt L. (2022). Nonuniform dual-rate extended kalman-filter-based sensor fusion for path-following control of a holonomic mobile robot with four mecanum wheels. *Applied Science, 2022*(12), 3560, MDPI. https://doi.org/10.3390/app12073560.