Manufacturing Science and Education 2025

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 68, Issue Special II, Month July, 2025

TRAJECTORY SIMULATION AND ENERGY CONSUMPTION EVALUATION FOR MOBILE ROBOTS EQUIPPED WITH MECANUM WHEELS USING MATLAB

Iosif Adrian MAROȘAN, Claudia Emilia GÎRJOB, Mihai CRENGANIȘ, Cristina-Maria BIRIȘ

Abstract: *This scientific paper presents the development of a MATLAB script designed for simulating the trajectory of a mobile robot equipped with Mecanum wheels and estimating the energy consumption associated with its movement along the generated trajectory. The simulation considers both direct and inverse kinematics, which are essential for creating the robot's dynamic model. This dynamic model is used to calculate the trajectory and estimate the energy consumption for the robot's movement along it.*
The MATLAB script includes an interactive graphical interface that allows the user to add points along the robot's trajectory by specifying coordinates (x, y) and the rotation angle (θ). Based on these inputs, the user can generate and visualize the full trajectory of the robot. Furthermore, the interface allows the selection of three different motion modes for the robot, namely: linear motion (piecewise), interpolated motion (spline), and combined motion (Holonomic). These various motion modes are implemented in the script to provide flexibility and allow the user to observe the robot's behaviour under different motion conditions. This paper focuses on simulating two distinct types of trajectories, each with its own characteristics, and simulating them using the three available motion modes. Following these simulations, energy consumption will be analysed for each trajectory, considering the dynamic factors of the robot's movement. A comparative analysis of energy consumption across the three motion types applied to the same trajectories will also be conducted to evaluate the energy efficiency of each approach. This comparative analysis will help identify the most efficient solutions for minimizing energy consumption in practical applications of mobile robots with Mecanum wheels. The results obtained will contribute to a deeper understanding of the energy behaviour of mobile robots, providing valuable insights for the design and optimization of autonomous systems with applications in logistics, transportation, and industrial automation.
Keywords: *Omnidirectional Mobile Robots, Mecanum Wheels, MATLAB simulation, Energy-Efficient Motion Planning.*

1. INTRODUCTION

Mobile robots with Mecanum wheels are prized for their omnidirectional mobility and tight-space maneuverability. By using passive rollers set at specific angles, they can move in any direction without reorienting, but this incurs complex kinematics and higher energy use. Precise simulation frameworks are therefore needed to predict both trajectories and energy consumption under varied conditions. For example, modeling trajectories with polynomials and tuning their coefficients and waypoints via evolutionary algorithms has yielded substantial energy savings [1]. Complementary experimental validations on four-wheeled platforms have confirmed that energy consumption models can achieve prediction accuracies exceeding 95%, underscoring the reliability of these methodologies [2].

Simultaneously, holistic energy models combining kinematic, dynamic and electrical factors—alongside motor behavior, friction losses and control parameters—offer a complete picture of onboard energy flow. One such model has accurately predicted power consumption across diverse operating regimes[3]. Similarly, other investigations have emphasized the importance of partitioning energy consumption into contributions from motion, control, sensing, and communication systems to identify key areas for energy optimization [4].

Motion planning advances have been key to boosting energy efficiency: for example, integrating MPC with velocity-obstacle methods into simulations enables real-time obstacle avoidance while cutting energy use [5]. Moreover, studies focusing on motion control and obstacle avoidance have shown that adapting control algorithms to dynamic environments can further reduce unnecessary energy consumption while ensuring safe navigation [6].

In addition to trajectory and energy optimization, the issue of mechanical vibrations induced by the Mecanum wheel design has been addressed in recent work. Reducing these vibrations not only enhances user comfort in applications such as mobility aids but also contributes to energy savings by minimizing losses due to mechanical oscillations [7]. Furthermore, recent developments in trajectory planning optimization have employed computationally efficient methods to generate near-optimal paths that balance energy consumption, motion smoothness, and safety constraints [8].

2. LITERATURE REVIEW

In this section, a comprehensive review of the state-of-the-art in Mecanum-wheeled mobile robotics is presented. The Table 1 summarizes key details of each study including title, publication year, type of work, and domain and the subsequent discussion integrates these findings to illustrate the logical progression of research in this area.

Table 1

Summary of Selected Studies in Mecanum-Wheeled Mobile Robotics.

Title	Year	Type of Work	Domain
Simulink Based Dynamic model for mecanum drive Autonomous Mobile platforms considering friction forces [9]	2024	Research Article	Dynamic modeling; Mecanum-wheeled mobile robots
Fuzzy Logic-Based Driving Decision for an Omnidirectional Mobile Robot Using a Simulink Dynamic Model [10]	2024	Research Article	Fuzzy control; driving decisions; dynamic modeling
Monitoring the Current Provided by a Hall Sensor Integrated in a Drive Wheel Module of a Mobile Robot [11]	2023	Research Article	Current monitoring; sensors; mobile robots
FPGA-Based Mechatronic Design and Real-Time Fuzzy Control with Computational Intelligence Optimization for Omni-Mecanum-Wheeled Autonomous Vehicles [12]	2019	Research Article	Mechatronic design; real-time fuzzy control; autonomous vehicles with Mecanum wheels
Energy Consumption Analysis of the Selected Navigation Algorithms for Wheeled Mobile Robots [13]	2023	Research Article	Energy consumption analysis; navigation algorithms; mobile robots
Energy Utilization Prediction Techniques for Heterogeneous Mobile Robots: A Review [14]	2024	Review	Energy prediction; mobile robots

For instance, one study developed a dynamic model in Simulink for autonomous mobile platforms by integrating a 3D CAD model and incorporating friction forces. This approach established a robust foundation for understanding the complex dynamic behavior of omnidirectional systems [9]. Building on this foundational work, another investigation introduced a fuzzy logic-based driving decision system that leveraged the established dynamic model to enable intelligent control in complex, real-world environments, thereby enhancing the robot's adaptability and performance [10]. Complementing these modeling approaches, further research proposed a real-time current monitoring method using a Hall sensor integrated within a drive wheel module, which provided critical insights into the performance and efficiency of the actuation systems by enabling precise, real-time measurement of electrical consumption [11]. In addition, another study advanced the field by presenting an FPGA-based mechatronic design that combined real-time fuzzy control with computational intelligence optimization. This innovative approach significantly improved the performance and reliability of autonomous vehicles equipped with Mecanum wheels, particularly under dynamic and unpredictable conditions [12]. On the energy analysis front, a comparative study examined energy consumption associated with various navigation algorithms for wheeled mobile robots. This work provided valuable quantitative data that can inform the development of energy-efficient navigation strategies, emphasizing the importance of algorithm selection in reducing overall power consumption [13]. Finally, a comprehensive review synthesized a range of energy utilization prediction techniques for heterogeneous mobile robots, offering an in-depth overview of the complexity and accuracy of existing energy models. This review highlighted both the strengths and limitations of current approaches, setting the stage for future enhancements in energy prediction methodologies [14].

In summary, recent advances in dynamic modeling, intelligent control, and energy analysis have markedly improved Mecanum-wheeled robots. Simulink-based dynamic models coupled with fuzzy logic capture omnidirectional behavior and enhance control, while real-time monitoring and FPGA-based mechatronics boost performance and reliability. Comparative energy studies highlight the need for efficient navigation and accurate consumption forecasting. Building on this foundation, our MATLAB-based study will deliver an integrated simulation that predicts both trajectories and energy use under varied conditions, guiding the design of more efficient, robust Mecanum-wheeled systems.

3. KINEMATIC AND DYNAMIC MODELING OF A FOUR-MECANUM-WHEELED MOBILE ROBOT

This chapter presents a comprehensive model of the motion and dynamics of a mobile robot with four Mecanum wheels. The model describes how the robot moves in any direction and how it interacts with external forces, including friction. This solid theoretical framework underpins future simulations, control strategies, and energy consumption analysis, contributing to the achievement of efficient and reliable robotic navigation.

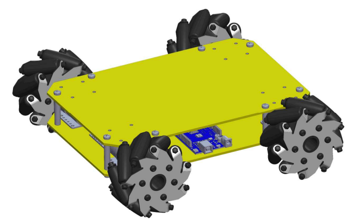

Fig. 1. Four-mecanum-wheeled mobile robot

In Figure 1, a detailed CAD model of a mobile robot equipped with Mecanum wheels is presented, which offers omnidirectional traction capabilities. This model serves as a starting point for an in-depth analysis of the robot, and in the following sections the equations that define the kinematic and dynamic behavior of the mobile platform will be presented.

3.1 Four-Wheel Mecanum Kinematics

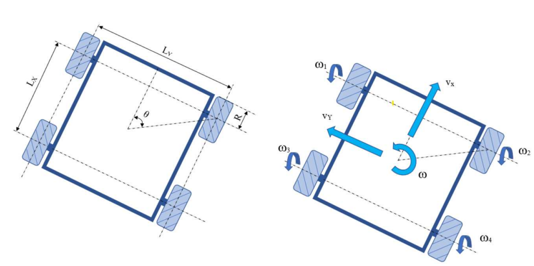

Fig. 2. Schematic diagram of the four-wheel Mecanum mobile robot [15].

The kinematic solution for a mobile robot with four Mecanum wheels is adopted from the work presented in [15] and is available in the Mobile Robotics Simulation Toolbox [16] within the MATLAB software package (The MathWorks Inc., Natick, MA, USA). Figure 2 illustrates the schematic diagram of the mobile robot with four Mecanum wheels. According [15] and [9], the kinematic model takes as inputs the wheel speeds—denoted as ω_1 through ω_4 (in rad/s)—and produces as outputs the linear velocities v_x and v_y (in m/s), together with the angular velocity ω (in rad/s), thus providing a complete description of the robot's motion.

To describe this relationship mathematically, both the direct and inverse kinematic equations can be used. The direct kinematic equations for the robot's movement are expressed in Equation (1).

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{R}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -2/(L_x+L_y) & 2/(L_x+L_y) & -2/(L_x+L_y) & -2/(L_x+L_y) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -(L_x+L_y)/2 \\ 1 & 1 & (L_x+L_y)/2 \\ 1 & 1 & -(L_x+L_y)/2 \\ 1 & -1 & (L_x+L_y)/2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{2}$$

The inverse kinematic equations are presented in Equation (2).

$$J = \begin{bmatrix} \frac{r}{4} & \frac{-r}{4} & \frac{-r}{2\cdot(L_x+L_y)} \\ \frac{r}{4} & \frac{r}{4} & \frac{r}{2\cdot(L_x+L_y)} \\ \frac{r}{4} & \frac{r}{4} & \frac{-r}{2\cdot(L_x+L_y)} \\ \frac{r}{4} & \frac{-r}{4} & \frac{r}{2\cdot(L_x+L_y)} \end{bmatrix} \tag{3}$$

$$J^t = \begin{bmatrix} \frac{r}{4} & \frac{r}{4} & \frac{r}{4} & \frac{r}{4} \\ \frac{-r}{4} & \frac{r}{4} & \frac{r}{4} & \frac{-r}{4} \\ \frac{-r}{2\cdot(L_x+L_y)} & \frac{r}{2\cdot(L_x+L_y)} & \frac{r}{2\cdot(L_x+L_y)} & \frac{-r}{2\cdot(L_x+L_y)} \end{bmatrix} \tag{4}$$

The Jacobian matrix is key to kinematic and dynamic analysis, relating joint velocities to end-effector motion and mapping applied forces to movements. Its transpose converts operational-space forces into joint torques, allowing resistance and inertial effects of the robot body to be expressed as wheel torques. This enables accurate dynamic simulation and optimized control of the mobile platform. By leveraging this matrix, control algorithms can precisely modulate wheel speeds to execute complex maneuvers such as lateral strafing and pivoting with minimal error. Incorporating real-time environmental feedback into the Jacobian framework further enhances adaptability and stability in dynamic conditions. The Mecanum-specific Jacobian is given in Equations (3) and (4).

3.2 Dynamic Equations of the Mobile Robot with Four Mecanum Wheels

The dynamics and kinematics of the four-Mecanum-wheeled mobile robot are described using a series of mathematical models. The process begins by converting the torque applied at each wheel into force using

$$F_i = \frac{\tau_i}{r} \tag{5}$$

where τ_i is the torque at wheel i (for $i = 1, \ldots, 4$) and r is the wheel radius. These individual forces are then mapped into the net force components in the robot's body frame and the resultant moment via the mapping matrix:

$$\begin{bmatrix} F_x \\ F_y \\ T_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{d} & -\frac{1}{d} & \frac{1}{d} & \frac{1}{d} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{6}$$

with

$$d = L_x + L_y$$

where L_x and L_y are the distances from the robot's center to the wheels along the X and Y axes, respectively.

The dynamics in the body frame are then governed by the following equations. The acceleration along the x-axis is given by:

$$\dot{v}_x = \frac{F_x}{m} + \omega v_y \tag{7}$$

and along the y-axis by:

$$\dot{v}_y = \frac{F_y}{m} - \omega v_x \tag{8}$$

where m is the mass of the robot and ω is its angular velocity. The angular acceleration is expressed as:

$$\dot{\omega} = \frac{T_z}{I_z} \tag{9}$$

with I_z representing the moment of inertia about the vertical axis.

To transform these velocities from the body frame to the global frame, the kinematic equations are used:

$$\dot{X} = v_x \cos\theta - v_y \sin\theta \tag{10}$$

$$\dot{Y} = v_x \sin\theta + v_y \cos\theta \tag{11}$$

and the rate of change of orientation is given by:

$$\dot{\theta} = \omega \tag{12}$$

Finally, the complete state derivative vector is assembled as:

$$dx = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{\omega} \end{bmatrix} \tag{13}$$

In summary, Equation (5) converts the wheel torques into forces; Equation (6) maps these forces to obtain the net force components and moment in the body frame; Equations (7)–(9) describe the dynamic evolution of the robot's linear and angular velocities; and Equations (10)–(12) transform these velocities into the global frame, with Equation (13) summarizing the complete state derivative.

This integrated model forms the basis for simulating the robot's dynamics and for designing effective control strategies in MATLAB.

4. SIMULATION IN THE MATLAB ENVIRONMENT

Simulation in MATLAB provides a flexible and robust environment for testing and validating trajectory planning algorithms. In this study, a dedicated script was developed to generate the robot's trajectories, several modes of motion (linear, interpolated, and holonomic) were implemented, and an interactive graphical interface was created to facilitate the configuration of waypoints, the setting of simulation parameters, and the visualization of results.

4.1 Development of the MATLAB script for generating trajectories

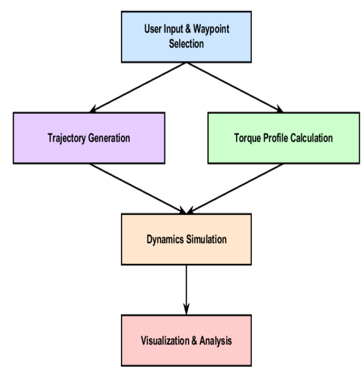

Fig. 3. Block Diagram of the MATLAB Script.

The first step in realizing the simulation consisted of writing a script that allows for the definition of a custom trajectory based on waypoints entered by the user; Figure 3 shows a logical flow of the main stages in a robot simulation system, organized in the form of a block diagram where each block represents a key stage and the arrows indicate the order and manner in which information propagates between stages. The first block in the diagram is the blue block that represents the initial stage, where the user defines the input parameters and sets the waypoints through which the robot must pass—these data are fundamental for all subsequent calculations and determine the trajectory the robot will follow. The purple block on the left receives the waypoint information and is responsible for generating a continuous trajectory (e.g., through linear interpolation or spline), the result of which consists of the coordinates $x(t), y(t)$ and, optionally, the robot's angle $\theta(t)$, as a function of time. In parallel with the trajectory generation, the green block on the right calculates the required wheel torque profile, considering acceleration, and other dynamic constraints; within this block, methods such as sinusoidal modulation, load-dependent torque adjustments, or other control models may be applied. In the current version of the simulator no explicit mechanical-friction terms are included. The fourth, orange block represents the dynamic simulation, in which the data from the previous blocks (trajectory and torque) are integrated to execute a simulation of the robot's physical behavior by considering masses, moments of inertia, forces, and torques, resulting in the evolution over time of the robot's state (position, velocity, acceleration, energy consumption, etc.). The simulated actuators are brushed DC motors driving the mecanum wheels through a fixed 47:1 gear train. Their dynamics are modelled in the simplest quasi-steady form: back-EMF plus winding resistance; inductance, rotor inertia, gearbox damping and saturation are neglected, so current is assumed to follow the commanded voltage instantaneously and generated torque is $\tau = K_t \cdot I$. In other words, the script captures the electrical steady-state and the rigid-body robot motion, but it does not integrate a full electromechanical state-space model of the motors. The final pink block represents the visualization and analysis phase of the data obtained from the simulation, allowing the user to evaluate performance, identify potential issues, and make optimization decisions. The MATLAB script is based on several essential elements, including a data structure for points, in which the waypoints are stored in a matrix that contains the coordinates (X, Y) and, optionally, a rotation angle θ associated with each point (the first being considered as the Start and marked in green, the last as the Final and marked in red, and the intermediate ones marked in blue). It also allocates time for each point by dividing the total time interval $[0, T]$ into segments corresponding to the number of points (resulting in a time vector associated with each waypoint and, for a finer simulation, generating an additional set of time samples, for example 500 samples). Furthermore, the coordinates are interpolated based on the time vectors and the coordinates build the trajectory using different methods (linear, spline, etc.), a trajectory that is subsequently used to calculate speeds and accelerations, which are essential in the robot's dynamics, and to compute the motion parameters where numerical differentiation of the interpolated coordinates yields the values (v_x, v_y, a_x, a_y) that are later used in the calculation of torque (if energy consumption is also modeled) or in validating speed limits. Thus, the MATLAB script ensures the generation of a coherent trajectory between the selected points, being sufficiently flexible to adapt to different planning requirements.

4.2 Implementation of Different Motion Modes: Linear, interpolated (Spline), and Holonomic

To cover a broader range of applications and robot behaviors, three motion modes have been implemented: linear motion, which uses linear interpolation between waypoints to create a trajectory composed of straight-line segments connecting each waypoint with the orientation angle θ also interpolated linearly; spline-interpolated motion, is used to connect the X and Y coordinates (and, optionally, the angle) with smooth spline curves, ensuring smoother transitions in speeds and accelerations at the waypoints and making it suitable for scenarios requiring fluid and continuous movements (such as manipulation applications or mobile robots with comfort requirements); and holonomic motion, which adds an additional component such as a sinusoidal oscillation in X and Y to simulate a situation where rotation is decoupled from translation, allowing the robot to orient itself independently of its direction of movement, a feature that is particularly useful for robots with Mecanum or Omni Wheel drives that can move in any direction without first changing orientation. The desired motion mode can be selected via a drop-down menu or a user-defined variable, and in the simulation, each mode calculates the coordinates $(x(t), y(t))$ and orientation $\theta(t)$ differently, directly influencing the robot's dynamics and energy consumption.

4.3 Interactive Graphical Interface for Trajectory Planning

To simplify the process of selecting points and configuring the simulation, an intuitive graphical user interface (GUI) was developed in MATLAB. The interface is divided into two main panels: the left panel, shown in Figure 4, which provides a graphical area (axes) where the user can click to add points.

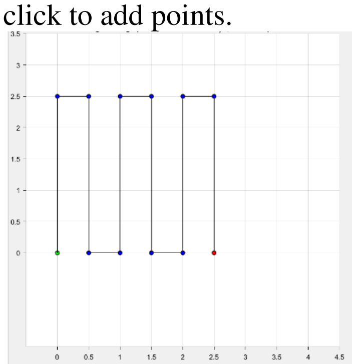

Fig. 4. Interactive Graphical Area for Waypoint Selection.

The right panel, shown in Figure 5, which contains the configuration controls (a list box with waypoints, a field for total simulation time, a motion type selector, a simulation button, etc.).

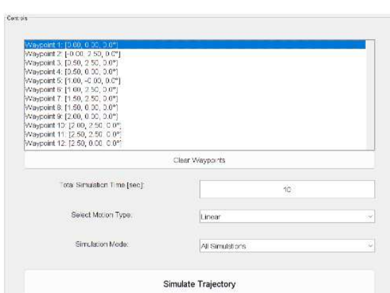

Fig. 5. Control Panel for Waypoint Configuration and Simulation Settings.

Within the interface, point addition and management are carried out as follows: a left-click adds a waypoint at the coordinates selected on the graph (with the first point becoming the starting point, marked in green, the last point becoming the endpoint, marked in red, and the intermediate points marked in blue), Shift Click allows the specification of a rotation angle for that point, and a right-click recenters the axes on the selected area for easier navigation on an extended map. In the control panel, the user can enter the total simulation time, select the motion type (linear, spline, or holonomic) and the simulation mode (a single simulation or all modes in parallel), and by pressing the Simulate Trajectory button, the interface calculates and displays the resulting trajectories, along with additional information such as current consumption or comparative graphs between modes. Visual feedback elements include a list box that displays the coordinates and angles of each waypoint for quick verification, a Clear Waypoints button that deletes all points and resets the interface, as well as, optionally, warning or confirmation messages when points are deleted. Through this interface, the user gains complete control over the trajectory and simulation parameters without the need to directly modify the source code, and the graphical elements such as distinct colors for points, drop-down menus, and buttons—contribute to an intuitive and efficient user experience.

5. ANALYSIS OF THE RESULTS OBTAINED FROM THE SIMULATIONS

This chapter presents a detailed analysis of the simulation results, highlighting the robot's performance, the energy consumption associated with different motion modes, and the validation of the proposed model. By examining both the animations and static representations, an integrated view is provided of how the interpolation methods and control strategy affect the robot's dynamic behavior and energy efficiency. This comprehensive discussion is essential for optimizing trajectories and control parameters in high-performance industrial applications.

5.1 Interactive Graphical Interface for Trajectory Planning

The robot's performance was evaluated by studying the trajectories generated for different motion modes. Below, each image is discussed in detail: Figure 6 shows an animation where the robot follows a linear trajectory. Notice that the waypoints are connected by straight segments, and abrupt changes in direction occur at each waypoint. This mode is effective for tasks requiring rapid execution, though it may introduce significant dynamic fluctuations.

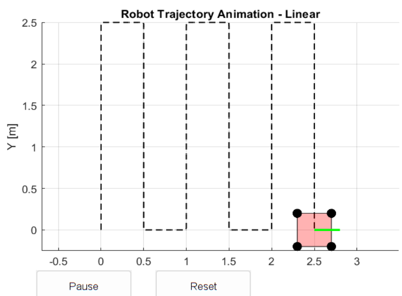

Fig. 6. Animation – Linear Trajectory.

Figure 7 provides a static representation of the linear trajectory. The image clearly highlights the straight segments and the abrupt direction changes at the waypoints.

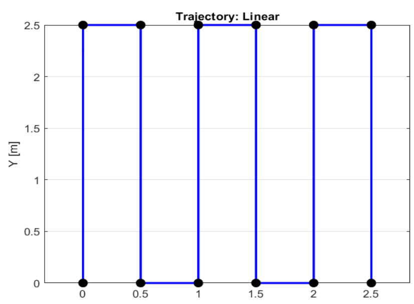

Fig. 7. Static Image – Linear Trajectory

Figure 8 shows an animation of a trajectory obtained by spline interpolation. Smooth transitions between waypoints result in a fluid motion, reducing dynamic oscillations.

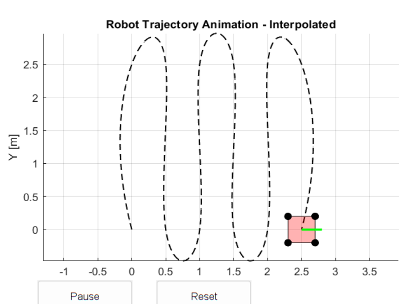

Fig. 8. Animation – Interpolated (Spline) Trajectory

Figure 9 displays the interpolated trajectory in a static form. The smooth curves between waypoints ensure gradual transitions in speed and acceleration.

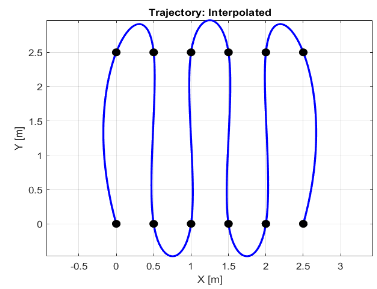

Fig. 9. Static Image – Interpolated Trajectory

Figure 10 presents an animation of the holonomic trajectory. In this mode, the decoupling of rotation from translation allows the robot to move in any direction while maintaining an independent orientation, which is crucial for complex maneuvers.

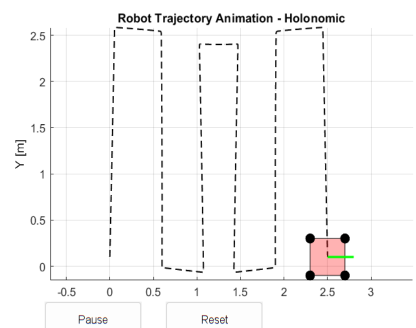

Fig. 10. Animation – Holonomic Trajectory.

Figure 11 shows the holonomic trajectory in a static image. This representation emphasizes the flexibility of the motion and the capability to maintain an independent orientation during movement.

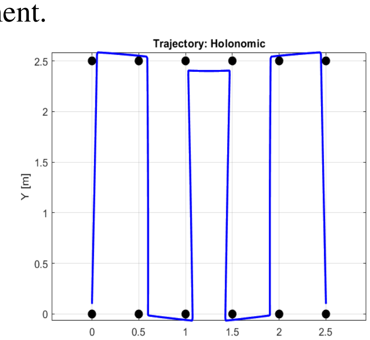

Fig. 11. Static Image – Holonomic Trajectory

Figure 12 illustrates a comprehensive visual comparison of the trajectories generated under the linear, interpolated, and holonomic motion modes. This figure not only showcases the distinct geometric patterns produced by each control strategy but also facilitates a holistic assessment of their performance. In particular, the linear mode is characterized by abrupt transitions between waypoints, the interpolated (spline) mode exhibits smooth, continuous curves that ease acceleration and deceleration, and the holonomic mode displays complex motion dynamics with decoupled rotation and translation.

By juxtaposing these trajectories, the figure allows for an integrated evaluation of precision, smoothness, and motion complexity, thereby highlighting the trade-offs and advantages inherent to each control approach.

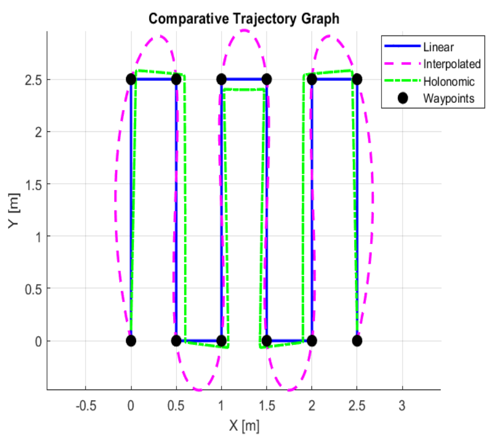

Fig. 12. Trajectory Comparisons

5.2 Comparison of Energy Consumption for Each Motion Mode

This section analyzes the energy consumption profiles, represented by the current curves, for each motion mode. The following figures illustrate the evolution of current during simulation for each mode:

Figure 13 shows the current profile for the linear trajectory. Notice the current peaks associated with the abrupt directional changes at the waypoints.

Figure 14 illustrates the current profile for the interpolated trajectory. Due to the smooth acceleration and deceleration transitions, the energy consumption is more uniform with moderate fluctuations.

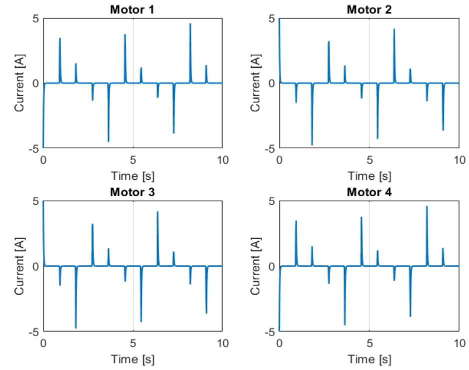

Fig. 13. Current Profile – Linear Trajectory.

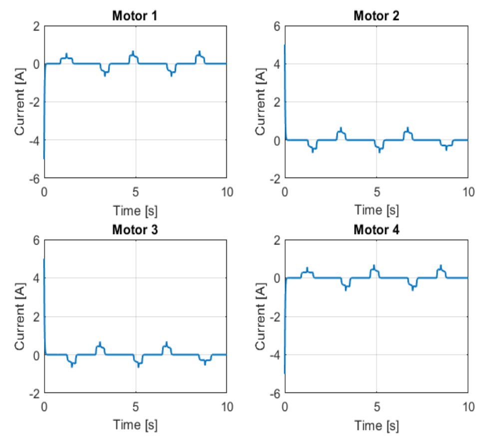

Fig. 14. Current Profile – Interpolated Trajectory.

Figure 15 presents the current profile for the holonomic trajectory. Variations in energy consumption are observed due to the decoupled rotation and lateral movements, resulting in temporary increases in current drawing during complex maneuvers.

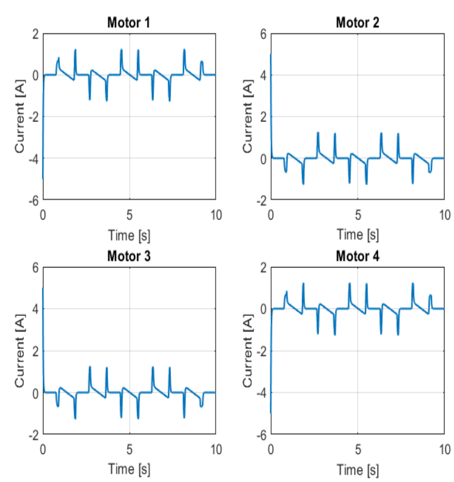

Fig. 15. Current Profile – Holonomic Trajectory.

5.3 Interpretation of the Results and Model Validation

We analyze energy consumption in two steps: first, we tabulate average and peak current demands for each motion mode to quantify energy profiles; then we compare these profiles across system components. This dual approach both validates the simulation model and reveals how dynamic conditions and control strategies influence energy use. Table 2 offers a quantitative benchmark, while accompanying graphs show temporal variations in current draw and pinpoint opportunities for further optimization.

Table 2

Energy Consumption Values for Different Motion Modes.

Motion Mode	Average Current (A)	Maximum Current (A)
Linear	0.54	4.58
Interpolated	0.29	5
Holonomic	0.38	4.5

Figure 16 provides a comparative analysis of the currents measured at each motor. This figure details how the current variations are influenced by directional changes and dynamic transitions, highlighting significant differences between the motion modes.

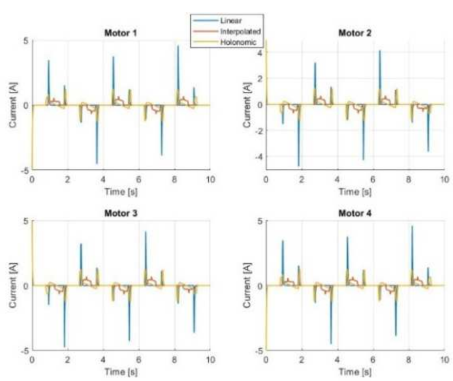

Fig. 16. Comparison of Motor Currents.

Figure 17 provides a comprehensive overview of the current profiles for the three distinct motion modes. It clearly demonstrates how the dynamic characteristics of each trajectory such as acceleration, deceleration, and rotational movements directly influence overall energy consumption.

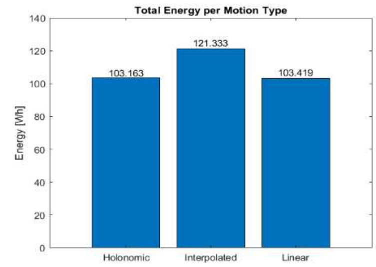

Fig. 17. Overall energy Comparison.

The combined analysis of the energy consumption table and the comparative current profiles leads to the following observations: The Interpolated mode consumes roughly 18 Wh more than the other modes because its spline path (20.53 m) is about 3 m longer and therefore takes longer to cover; at a motor speed of 100 rpm, the energy scales almost linearly with path length (time) as long as the instantaneous power stays nearly constant. Linear and Holonomic have almost identical lengths (~17.5 m), which is why they end up with practically identical energies (~103 Wh). The 0.26 Wh difference comes from the small lateral deviations added in Holonomic, but their impact on total consumption is minimal at this low speed. In short, at a motor speed of 100 rpm the dominant factor in energy use is the actual path length; the control profile affects the energy only if it changes the distance significantly (or if it drives the motor into peak-current operation more often, which is not the case here).

6. CONCLUSION

Incorporating kinematic and dynamic modeling resulted in a robust mathematical model capable of precisely describing the robot's movements from the direct and inverse kinematic equations to the transformation of forces via the Jacobian matrix, thereby providing theoretical framework for simulations and enabling an in-depth understanding of the dynamic behavior of the Mecanum-wheeled robot. The implementation of three motion modes linear motion, spline interpolation, and holonomic motion demonstrated the system’s versatility in addressing various operational requirements, while a comparative analysis of current profiles, used as an indicator of energy consumption, highlighted that linear motion exhibits abrupt transitions between waypoints leading to significant current spikes and high energy consumption during sudden directional changes; spline interpolation ensures smooth and continuous transitions between points, resulting in a more uniform energy profile with moderate current fluctuations that translate into enhanced energy efficiency; and holonomic motion, by decoupling rotation from translation to facilitate maneuverability in complex environments, generates temporary variations in current that underscore the need for adaptive control strategies to optimize consumption.

Our next steps are to embed advanced control strategies directly into the simulator, and then test them on a physical prototype to calibrate every parameter and to validate the model's outputs. This workflow will ensure that the final solutions combine top-tier Mecanum maneuverability with optimal energy consumption, precisely matching the demands of autonomous platforms in manufacturing, logistics, and transportation.

7. REFERENCES

[1]Xie, L., Stol, K., Xu, W., *"Energy-Optimal Motion Trajectory of an Omni-Directional Mecanum-Wheeled Robot via Polynomial Functions," Robotica*, vol. 38, no. 8, pp. 1400–1414, 2020, doi: 10.1017/S026357471900153X.

[2]Xie, L., Herberger, W., Xu, W., Stol, K. A., *"Experimental validation of energy consumption model for the four-wheeled omnidirectional Mecanum robots for energy-optimal motion control," 2016 IEEE 14th Int. Work. Adv. Motion Control. AMC 2016*, pp. 565–572, 2016, doi: 10.1109/AMC.2016.7496410.

[3]Hou, L., Zhou, F., Kim, K., Zhang, L., *"Practical model for energy consumption analysis of omnidirectional mobile robot," Sensors*, vol. 21, no. 5, pp. 1–18, 2021, doi: 10.3390/s21051800.

[4]Zhang, L., Kim, J., Sun, J., *"Energy modeling and experimental validation of four-wheel mecanum mobile robots for energy-optimal motion control," Symmetry (Basel).*, vol. 11, no. 11, 2019, doi: 10.3390/sym11111372.

[5]Azizi, M. R., Rastegarpanah, A., Stolkin, R., *"Motion planning and control of an omnidirectional mobile robot in dynamic environments," Robotics*, vol. 10, no. 1, 2021, doi: 10.3390/robotics10010048.

[6]Alwan, H. M., Nikolavich, V. A., Shbani, A., Vladmirovna, K. O., *"Motion Control and Obstacle Avoidance of Mobile Robot with Mecanum Wheels," Int. J. Technol.*, vol. 16, no. 1, pp. 82–96, 2025, doi: 10.14716/ijtech.v16i1.7254.

[7]Yadav, P. S., Mohanta, J. C., Agrawal, V., Ahmed, M. F., *"Modeling and performance evaluation of mecanum wheel platform with reduced vibration and a robust path planning," J. Brazilian Soc. Mech. Sci. Eng.*, vol. 46, no. 5, pp. 1–25, 2024, doi: 10.1007/s40430-024-04847-5.

[8]Almasri, E., Uyguroğlu, M. K., *"Modeling and trajectory planning optimization for the symmetrical multiwheeled omnidirectional mobile robot," Symmetry (Basel).*, vol. 13, no. 6, 2021, doi: 10.3390/sym13061033.

[9]Crenganiș, M., Gîrjob, C. E., Rusu, D. M., Bârsan, A., *"Simulink based dynamic model for mecanum drive autonomous mobile platforms considering friction forces,"* vol. 67, no. Ii, pp. 297–308, 2024

[10] Crenganiș, M. *et al.*, *"Fuzzy Logic-Based Driving Decision for an Omnidirectional Mobile Robot Using a Simulink Dynamic Model," Appl. Sci.*, vol. 14, no. 7, 2024, doi: 10.3390/app14073058.

[11] Constantin, G. *et al.*, *"Monitoring the Current Provided by a Hall Sensor Integrated in a Drive Wheel Module of a Mobile Robot," Machines*, vol. 11, no. 3, 2023, doi: 10.3390/machines11030385.

[12] Huang, H. C., Tao, C. W., Chuang, C. C., Xu, J. J., *"FPGA-based mechatronic design and real-time fuzzy control with computational intelligence optimization for omni-mecanum-wheeled autonomous vehicles," Electron.*, vol. 8, no. 11, 2019, doi: 10.3390/electronics8111328.

[13] Rapalski, A., Dudzik, S., *"Energy Consumption Analysis of the Selected Navigation Algorithms for Wheeled Mobile*

Robots," Energies, vol. 16, no. 3, 2023, doi: 10.3390/en16031532.

[14] Góra, K., Granosik, G., Cybulski, B., *"Energy Utilization Prediction Techniques for Heterogeneous Mobile Robots: A Review," Energies* , vol. 17, no. 13, 2024, doi: 10.3390/en17133256.

[15] Racz, S.-G., *et al.*, *"Mobile Robots—AHP-Based Actuation Solution Selection and Comparison between Mecanum Wheel Drive and Differential Drive with Regard to Dynamic Loads," Machines*, vol. 10, no. 10, p. 886, Oct. 2022, doi: 10.3390/machines10100886.

[16] "MathWorks Student Competitions Team. Mobile Robotics Simulation Toolbox, 2025 GitHub. Retrieved March2025,https://github.com/mathworks-robotics/mobile-robotics-simulation-toolbox".

Simularea traiectoriei și evaluarea consumului energetic pentru roboți mobili echipați cu roți Mecanum folosind MATLAB

Acest articol științific prezintă dezvoltarea unui script MATLAB conceput pentru simularea traiectoriei unui robot mobil echipat cu roți Mecanum și estimarea consumului de energie asociat mișcării sale de-a lungul traiectoriei generate. Simularea ia în considerare atât cinematica directă, cât și cea inversă, care sunt esențiale pentru crearea modelului dinamic al robotului. Acest model dinamic este folosit pentru calcularea traiectoriei și estimarea consumului de energie pentru mișcarea robotului de-a lungul acesteia. Scriptul MATLAB include o interfață grafică interactivă care permite utilizatorului să adauge puncte de-a lungul traiectoriei robotului prin specificarea coordonatelor (x, y) și a unghiului de rotație (θ). Pe baza acestor informații, utilizatorul poate genera și vizualiza traiectoria completă a robotului. Mai mult, interfața permite selectarea a trei moduri diferite de mișcare pentru robot, și anume: mișcare liniară (pe segmente), mișcare interpolată (spline) și mișcare combinată (holonomică). Aceste moduri de mișcare sunt implementate în script pentru a oferi flexibilitate și pentru a permite utilizatorului să observe comportamentul robotului în condiții diferite de mișcare. Articolul se concentrează pe simularea a două tipuri distincte de traiectorii, fiecare cu propriile caracteristici, și pe simularea lor folosind cele trei moduri de mișcare disponibile. După aceste simulări, consumul de energie va fi analizat pentru fiecare traiectorie, ținând cont de factorii dinamici ai mișcării robotului. De asemenea, va fi realizată o analiză comparativă a consumului de energie între cele trei tipuri de mișcare aplicate acelorași traiectorii, pentru a evalua eficiența energetică a fiecărei abordări. Această analiză comparativă va ajuta la identificarea celor mai eficiente soluții pentru minimizarea consumului de energie în aplicațiile practice ale roboților mobili cu roți Mecanum. Rezultatele obținute vor contribui la o înțelegere mai profundă a comportamentului energetic al roboților mobili, oferind perspective valoroase pentru proiectarea și optimizarea sistemelor autonome cu aplicații în logistică, transport și automatizare industrială.

Iosif-Adrian MAROȘAN, Assistant professor, Lucian Blaga University of Sibiu, Machines and Industrial Equipment dep, adrian.marosan@ulbsibiu.ro, Str. Emil Cioran no. 4, Sibiu.

Claudia-Emilia GÎRJOB, Associate professor, Lucian Blaga University of Sibiu, Machines and Industrial Equipment dep, claudia.girjob@ulbsibiu.ro, Str. Emil Cioran no. 4, Sibiu. Autor correspondent.

Mihai CRENGANIȘ, Assistant professor, Lucian Blaga University of Sibiu, Machines and Industrial Equipment dep, mihai.crenganis@ulbsibiu.ro, Str. Emil Cioran no. 4, Sibiu.

Cristina-Maria BIRIȘ, Associate professor, Lucian Blaga University of Sibiu, Machines and Industrial Equipment dep, cristina.biris@ulbsibiu.ro, Str. Emil Cioran no. 4, Sibiu.