TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 69, Issue Special I, February, 2026

DYNAMIC MODELING AND SIMULATION OF AN AUTONOMOUS MOBILE ROBOT FOR INDUSTRIAL APPLICATIONS USING MATLAB/SIMULINK

Mihai CRENGANIS, Radu BREAZ, Gabriel RACZ, Claudiu-Damian PETRU

Abstract: *This paper presents the development and the simulations of a virtual dynamic model for a mobile robot that can be used in industrial applications. The dynamic model was developed using the MATLAB/Simulink platform and the Simscape library. The main goal is to build a "digital twin" that allows the analysis of the dynamic behavior of the robot under various operating conditions. The dynamic model takes into account friction forces, component masses, inertia and various motion configurations. The model includes a modular approach that integrates kinematic actuator blocks, transposed Jacobian matrices and detailed modeling of resistive forces. The analyzed configuration includes locomotion based on conventional wheels and differential traction but reaching the target points through different types of motion, to highlight the differences in maneuverability and energy efficiency.*

Key words: *Mobile Robot, Matlab-Simulink, Dynamic model*

1. INTRODUCTION

In recent decades, mobile robotics has experienced an accelerated development, being integrated into more and more industrial fields due to the ability of mobile robots to perform complex tasks in an autonomous, flexible and efficient manner. These robots are used in a wide range of applications, from internal material transport in smart factories to autonomous inspections in hazardous or inaccessible environments [1], [2].

To achieve high performance, it is essential that mobile robotic systems are designed not only at the hardware level, but also with a strong focus on modeling and simulation of their dynamic behavior. Dynamic models allow understanding the interaction between mechanical components, propulsion systems and the operating environment, thus providing a valuable framework for optimizing motion, control and energy consumption [3].

A modern and increasingly used approach in this regard is the concept of digital twin, which consists of a virtual model that faithfully reflects the behavior of a real physical system. It allows testing and validating control strategies, anticipating failures and optimizing processes, without the need for expensive physical prototypes or tests in real hazardous environments [4], [5].

This paper proposes the development and analysis of a virtual dynamic model for a mobile robot intended for industrial applications. The model was developed using the MATLAB/Simulink platform and the Simscape library, thus providing an integrated environment for advanced simulations that include mechanical, electrical and control components. Unlike simplified approaches, the model presented in this paper takes into account a significant number of real physical factors: component masses, moments of inertia, friction forces (static and dynamic), as well as the effects generated by various motion configurations and applied loads [6].

An essential element of this work is the modular architecture of the proposed dynamic model, which facilitates its adaptation and extension according to the application requirements. The model includes dedicated blocks for kinetic drives, transformations through transposed Jacobian matrices, as well as a detailed representation of the resistive forces at the wheel-ground contact points – essential aspects for a realistic simulation of the robot's maneuverability [7], [8].

The main goal of this model is to build a functional digital twin that allows the analysis of the robot's dynamic behavior in various operational scenarios. Although the mechanical configuration remains the same – based on conventional wheels and differential traction – several types of trajectories and travel modes are analyzed, in order to highlight their impact on the overall performance, especially in terms of maneuverability and energy efficiency [9].

Through the simulations performed, the work contributes to the detailed understanding of the relationship between the robot's constructive parameters, the dynamic operating conditions and the obtained behavior, thus providing a solid basis for the design and optimization of industrial mobile robots from the perspective of both dynamic performance and energy consumption.

2. MATHEMATICAL MODEL OF A DIFFERENTIAL DRIVE MOBILE ROBOT

This section presents the complete mathematical model of a differential drive mobile robot with four powered wheels. The model includes the kinematic equations, the dynamic motion equations, the DC motor model, as well as the friction forces that affect the robot's behavior during both translational and rotational motion. Additionally, the torque required at each drive wheel is derived for cases where the robot follows a predefined velocity profile. ·

2.1. The Robot Kinematics

The robot's position and orientation in a 2D plane are described by the variables *x*, *y*, and *θ*.

The robot's motion is characterized by its linear velocity *v* and angular velocity *ω*. The linear velocities of the left and right wheels are denoted by *vl* and *vr*. *R* represents the radius of the wheels and *L* is the distance between the left and right wheels (the wheelbase).

The kinematic equations are expressed below:

$$\frac{dx}{dt} = v \cdot \cos(\theta) \quad (1)$$

$$\frac{dy}{dt} = v \cdot \sin(\theta) \quad (2)$$

$$\frac{d\theta}{dt} = \omega \quad (3)$$

$$v = \frac{(v_r + v_l)}{2} \quad (4)$$

$$\omega = \frac{(v_r - v_l)}{L} \quad (5)$$

2.2. The Equations of Motion

To describe the robot's dynamics, we consider the total mass *m*, the moment of inertia *Iz*, the torques *τl* and *τr* applied to the left and right wheels respectively, and the wheel radius *R*. For moving forward (a translational motion):

$$m \cdot \frac{dv}{dt} = \frac{\tau_l + \tau_r}{R} - F_{fric}, roll - F_{fric}, viscous \quad (6)$$

For the rotational motion:

$$I_z \cdot \frac{d\omega}{dt} = \left(\frac{L}{2R}\right) \cdot (\tau_r - \tau_l) - \left(\frac{L}{2}\right) \cdot F_{fric}, slide - \tau_{fric}, viscous \quad (7)$$

2.3. The Friction Forces

The rolling resistance has the following expression:

$$F_{fric}, roll = \mu_r \cdot m \cdot g \quad (8)$$

For the lateral sliding friction one can express:

$$F_{fric}, slide = \mu_s \cdot m \cdot g \cdot sgn(\omega) \quad (9)$$

Viscous friction can be expressed as follows:

$$F_{fric}, viscous = \mu_v \cdot v \quad (10)$$

$$\tau_{fric}, viscous = \mu_v \cdot \omega \quad (11)$$

2.4. The DC Motor Model

The torque produced by the motor is related to the current as follows:

$$\tau = K_t \cdot i \quad (12)$$

The voltage applied to the motor terminals is given by:

$$V = R_a \cdot i + L_a \cdot \frac{di}{dt} + K_e \cdot \omega \quad (13)$$

Where τ is the motor torque, *i* is the current, *V* is the applied voltage, *Ra* is the armature resistance, *La* is the armature inductance, *Kt* is the torque constant, *Ke* is the back EMF constant, and ω is the angular velocity of the motor.

2.5. Complete System of Equations

$$m \cdot \frac{dv}{dt} = \frac{\tau_l + \tau_r}{R} - \mu_r \cdot m \cdot g - \mu_v \cdot v \quad (14)$$

$$I_z \cdot \frac{d\omega}{dt} = \left(\frac{L}{2R}\right) \cdot (\tau_r - \tau_l) - \left(\frac{L}{2}\right) \cdot \mu_s \cdot m \cdot g \cdot sgn(\omega) - \mu_v \cdot \omega \quad (15)$$

2.6. Wheel Torque Computation for Imposed Velocity Profile

When the robot follows a predefined linear velocity *v*(t) and angular velocity *ω*(t), the required torques at the left and right wheels can be computed by first determining the necessary translational force and rotational torque.

The total force for a translational movement has the expression:

$$F_{total(t)} = m \cdot \frac{dv}{dt} + \mu_r \cdot m \cdot g + \mu_v \cdot v(t) \quad (16)$$

Total torque for a rotational movement is:

$$\tau_{total(t)} = I_z \cdot \frac{d\omega}{dt} + \left(\frac{L}{2}\right) \cdot \mu_s \cdot m \cdot g \cdot sgn\big(\omega(t)\big) + \mu_v \cdot \omega(t) \quad (17)$$

Individual traction forces:

$$F_{r(t)} = \frac{F_{total(t)}}{2} + \frac{\tau_{total(t)}}{L} \quad (18)$$

$$F_{l(t)} = \frac{F_{total(t)}}{2} - \frac{\tau_{total(t)}}{L} \quad (19)$$

Resulting the wheel torques:

$$\tau_r^{res(t)} = F_{r(t)} \cdot R \quad (20)$$

$$\tau_l^{res(t)} = F_{l(t)} \cdot R \quad (21)$$

These equations allow for determining the torque at each drive wheel as a function of time, based on the desired motion profile and considering both inertia and friction effects.

3. IMPLEMENTATIONS IN MATLAB/SIMULINK

The dynamic mathematical model presented above in the paper can be directly replicated in a MATLAB/Simulink simulation using the block diagram. In order to be able to run the equations in real time, the most suitable block is the "Embedded Matlab function" in Simulink.

In Figure 1, the block diagram corresponding to the dynamic model for the mobile robot with differential traction is presented.

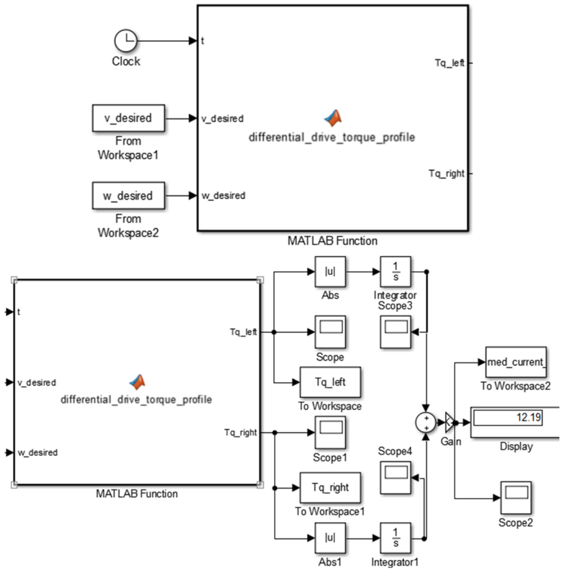

Fig. 1. Simulink block diagram used to implement the dynamic model equations.

The Simulink block diagram simulates the electric power consumption of a mobile robot with differential locomotion, based on imposed speed profiles.

At the input, the calculation block contains three data:

- a Clock block that generates the current time t;
- a v_desired signal (the desired linear velocity) from the Workspace;
- a w_desired signal (the desired angular velocity) from the Workspace.

These signals are input data into the MATLAB Function block, which, based on the robot dynamics equations, calculates the necessary torques at the left and right wheels, called Tq_left and Tq_right. The block contains MATLAB code that takes into account all the robot parameters in the dynamic model, such as the total mass, the wheel diameters, the width of the robot and its length.

At the output, the two torques are taken and:

- are visualized in Scope blocks;
- are saved in Workspace for later analysis;
- are passed through Abs blocks to obtain the absolute value of the torque (regardless of the direction of rotation);
- are integrated in time with 1/s blocks to obtain the energy consumed (integrated torque).

The results of the two integrators are added, and the sum is displayed numerically in a Display block and sent to Workspace. Subsequently, the transformation from the total torque developed to the current consumed is made based on the characteristics of the 37D DC electric motor. The resulting value is an approximation of the current consumed by the robot for the given trajectory.

The scheme is very useful because it allows the analysis of the efficiency of the robot's movement and because based on it, the energy costs associated with a certain speed profile can be evaluated.

The simulation was carried out using the data provided by the manufacturer for the Pololu 37D DC motor. The characteristics of this motor are presented below in Table 1.

Table 1

Parameter	**Symbol / Unit**	**Value**
Nominal voltage	VDC	12 V
Nominal current	A	0.66 A
Nominal torque	Nm	0.22 Nm
Nominal speed	rpm	180 rpm
Stall torque	Nm	2.06 Nm
Maximum torque	Nm	2.06 Nm
No-load speed	rpm	200 rpm
Nominal output power	W	4.0 W
Maximum output power	W	10.0 W
Torque constant (estimated)	Nm/A	~0.32 Nm/A
Terminal resistance (est.)	Ω	~2.18 Ω
Terminal inductance	mH	—
Starting (stall) current	A	5.5 A
No-load current	A	0.2 A
Demagnetization current	A	>5.5 A (est.)
Rotor inertia	g·cm²	—
Weight of motor	kg	0.19 kg

The two tests on which the tests were performed are presented in Figures 2 and 3 below.

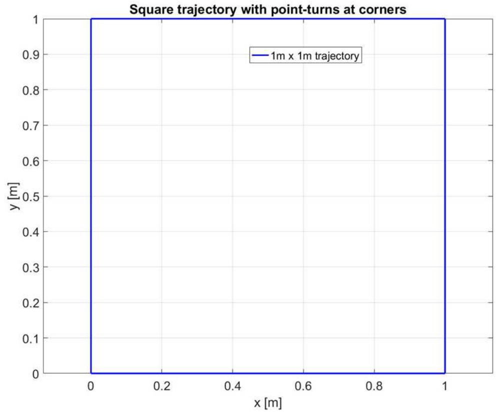

Fig. 2. Squared 1x1 m trajectory with sharp corners

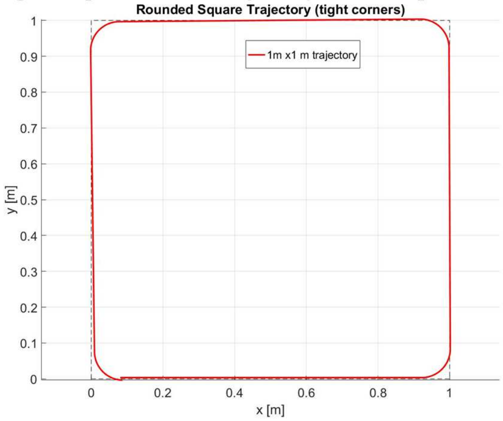

Fig. 3. Squared 1x1 m trajectory with rounded tight corners

The results obtained from the simulation are as follows, Figure 4.

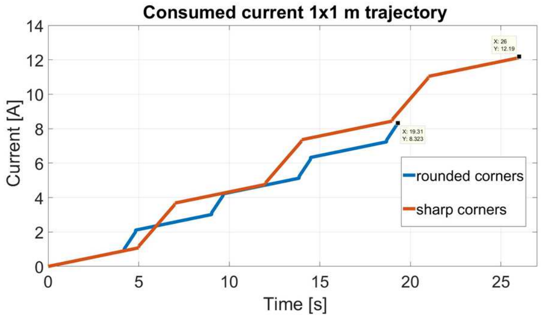

Fig. 4. Consumed current after simulations

The graph shown shows the evolution of the current consumed by the mobile robot that follows a 1x1 meter square trajectory in two different modes. A first mode would be that with 90 degree fixed point rotations and another is with corner curve-type traversing. The blue line corresponds to a trajectory with rounded corners. In this case, the robot turns smoothly without stopping, and the orange line corresponds to a trajectory with sharp corners. In this case, the robot stops and rotates at a fixed point. It is observed that in the case of sharp corners, the current increases faster and reaches a higher value by the end of the trajectory. At the end of the simulated time, the trajectory with rounded corners consumed approximately 8.32 [A], and the one with corner stops approximately 12.19 [A]. The difference is explained by the fact that rotations on the spot require more torque and therefore more current from the motors. Therefore, continuous, smooth motion is more energy efficient.

4. COMPLEX TRAJECTORY SIMULATION

The simulations that were carried out aimed to compare two types of trajectories. The first trajectory was with sharp turns Figure 5.

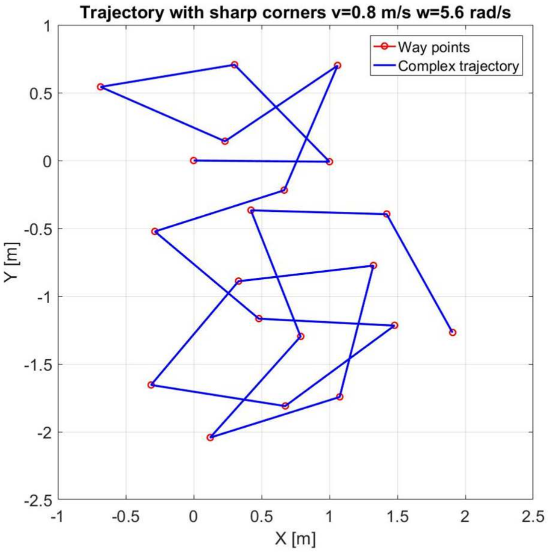

Fig. 5. Imposed robot trajectory with sharp turn

In this case the robot stops at each waypoint and rotates in place before continuing its movement. The second is the continuous turns (smooth turn) Figure 6. In this case the robot follows a curved and continuous trajectory without stops. In the case of the trajectory with sharp turns, the possible linear velocity was v=0.8 m/s, and the angular velocity w=5.6 rad/s. This leads to a fast movement, however there are frequent stops and accelerations.

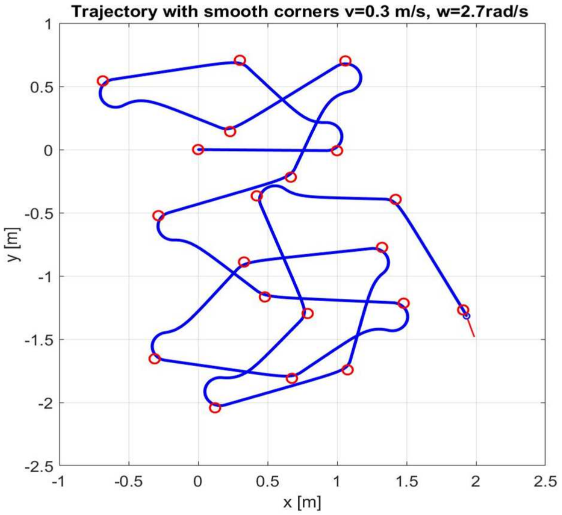

Fig. 6. Imposed robot trajectory smooth turn

In contrast, for the trajectory with smooth turns, the robot moved with a maximum linear velocity possible to respect the trajectory v=0.3 m/s and an angular velocity w= 2.7 rad/s, resulting in a more fluid movement and without mechanical shocks.

5. CONCLUSIONS

In Figure 7, the measured current is shown for both trajectories. For the smooth-turn case (blue), the execution time is shorter - about 65.6 s - compared with the sharp-turn case (red), about 86.3 s. The peak current is slightly higher for the smooth turn (~97.9 A) than for the sharp turn (~92.3 A).

From an energy standpoint, total consumption is given by the time integral of the current (∫I·dt). Based on the nearly linear ramps in the plot, the approximate totals are:

- sharp turns: ~92.3·86.3 ≈ 3983 A·s ≈ 1.106 Ah;
- smooth turns: ~97.9·65.6 ≈ 3210 A·s ≈ 0.892 Ah.

Therefore, although the smooth-turn trajectory exhibits a slightly higher peak current, it results in both a shorter travel time and a lower total consumption (and, at the same voltage, lower energy). Thus, the smooth-turn trajectory is advantageous in terms of execution time and overall energy, while also providing gentler motion for the mechanical components.

Overall, the results confirm that the digital twin allows a precise assessment of motion strategies, supporting decisions that balance energy efficiency, execution time, and mechanical wear according to industrial priorities.

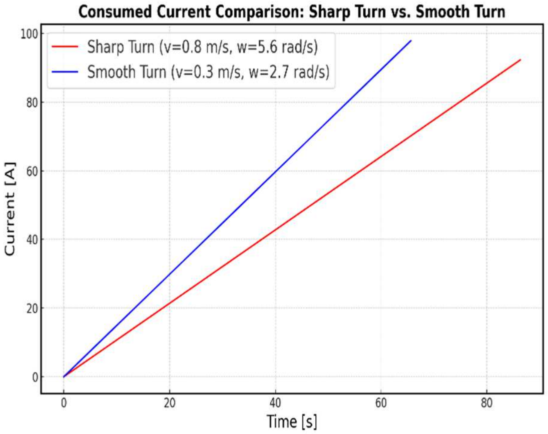

Fig. 7. Imposed robot trajectory with sharp turn

6. REFERENCES

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, Wiley, 2006.

[2] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd ed., MIT Press, 2011.

[3] B. Siciliano and O. Khatib (Eds.), Springer Handbook of Robotics, Springer, 2016.

[4] M. Grieves and J. Vickers, “Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems,” in Transdisciplinary Perspectives on Complex Systems, Springer, 2017.

[5] P. Glaessgen and D. Stargel, “The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles,” AIAA Infotech@Aerospace Conference, 2012.

[6] The MathWorks Inc., Simscape User’s Guide, MATLAB R2023a Documentation.

[7] A. De Luca and R. Mattone, “Actuator Failure Detection and Isolation Using Generalized Momenta,” in Proceedings of the IEEE International Conference on Robotics and Automation, 2003.

[8] C. Canudas-de-Wit, B. Siciliano, and G. Bastin, Theory of Robot Control, Springer, 1996.

[9] A. Kelly, Mobile Robotics: Mathematics, Models, and Methods, Cambridge University Press, 2013.

Modelare dinamică și simulare a unui robot mobil autonom pentru aplicații industriale utilizând MATLAB/Simulink

Această lucrare prezintă dezvoltarea și simulările unui model dinamic virtual pentru un robot mobil, ce poate fi utilizat în aplicații industriale. Modelul dinamic a fost realizat utilizând platforma MATLAB/Simulink și biblioteca Simscape. Obiectivul principal este construirea unui „geamăn digital” care să permită analiza comportamentului dinamic al robotului în diverse condiții de operare. Modelul dinamic ia în considerare forțele de frecare, masele componentelor, inerția și diferite configurații de mișcare. Modelul adoptă o abordare modulară, care integrează acționare cinematica, și o modelare detaliată a forțelor rezistente. Configurația analizată include locomoția pe baza roților convenționale și a tracțiunii diferențiale, dar atingerea punctelor țintă se realizează prin diferite tipuri de deplasare, pentru a evidenția diferențele în manevrabilitate și eficiență energetică.

Mihai CRENGANIȘ, Assoc. Prof., PhD, Lucian Blaga University of Sibiu, Faculty of Engineering, Machines and Industrial Equipment Dep., Emil Cioran 4, Sibiu, Romania, mihai.crenganis@ulbsibiu.ro, +40745 311 868.

Radu BREAZ, Prof., PhD, Lucian Blaga University of Sibiu, Faculty of Engineering, Machines and Industrial Equipment Dep., Emil Cioran 4, Sibiu, Romania, radu.breaz@ulbsibiu.ro.

Gabriel RACZ, Prof., PhD, Lucian Blaga University of Sibiu, Faculty of Engineering, Machines and Industrial Equipment Dep., Emil Cioran 4, Sibiu, Romania, gabriel.racz@ulbsibiu.ro.

Claudiu-Damian PETRU, PhD student, Lucian Blaga University of Sibiu, Faculty of Engineering, Machines and Industrial Equipment Dep., Emil Cioran 4, Sibiu, Romania, claudiu.petru@ulbsibiu.ro.