

Designing Digital Twin of a Three Wheeled Robot Using Simscape for Trajectory Tracking

Khadr Aymen^{1,*} and Houidi Ajmi²

¹Laboratory of Mechanics of Sousse, National Engineering School of Sousse, Polytechnic School of Sousse, University of Sousse, Sousse, Tunisia

²Laboratory of Mechanics of Sousse, Higher Institute of Applied Sciences and Technology of Sousse, University of Sousse, Sousse, Tunisia

*Corresponding author: khadr.aymen@yahoo.fr

Submitted 12 April 2026; Revised 08 June 2026; Accepted 12 June 2026; Available online 19 June 2026.

Copyright © 2026 The Authors.

Abstract: Due to the increased production costs, one of the areas of interest in the industry is the development of digital twins of autonomous mobile robots in order to predict their behavior using different control algorithms and validate them in a virtual environment. For this purpose, this paper presents the design of a digital twin of a three wheeled mobile robot using the Simscape Multibody and Matlab/Simulink tools to enable precise real-time tracking and virtual testing. The framework initiates by exporting an accurate 3D assembly of the robot from Solidworks to Simscape to replicate the kinematics, geometric constraints and inertial properties of the physical model. The imported model is then enhanced by adding actuator dynamics, sensor blocks and wheel-ground contact friction forces. The usefulness of the proposed model is demonstrated through a closed loop dynamic simulation of tracking a predefined path. Obtained results prove that the developed robot digital twin can be reliably used to predict the dynamic behavior of this type of robots and validate control strategies to be applied. Further, the proposed methodology will help to have a correct digital representation of the robot and to reduce both time and cost of research and development.

Keywords: Digital twin; Dynamic; Mobile robot; Simscape Multibody; Trajectory tracking.

1. INTRODUCTION

Autonomous Mobile Robots (AMRs) are becoming increasingly common in a wide range of applications, including manufacturing, healthcare, and logistics. There are many different types of AMRs. Some of the most common types include wheeled robots, tracked robot, and legged robots. Each type of robot has its own strengths and weaknesses. Wheeled robots are the most common type of autonomous mobile robot. They are relatively easy to control and can move quickly over a variety of surfaces. However, they can be unstable on uneven terrain.

To design and develop effective AMRs, it is essential to have a good understanding of their behavior in the real world. This can be achieved through numerical modelling as a valuable tool for the development and evaluation of AMRs without the need for physical prototypes. This can help to improve the design of robots, to test the performance of different control algorithms for AMRs in a variety of environments including those that are difficult or dangerous to access in the real world and to reduce the cost and time of development.

There are many different modelling approaches that can be used to simulate AMRs. The choice of modelling approach depends on the specific application of the AMR. For example, kinematic modelling is often used for applications that require accurate predictions of the robot's motion in path planning and obstacle avoidance problems. Applications requiring accurate predictions of robot interactions with its environment, control design, and actuator sizing frequently use dynamic modelling, which describes the motion of the robot in terms of physical laws such as Newton's laws of motion. Currently, there are several simulators developed and applied to create virtual 3D models of robots and to simulate their motion. The most used of them being Webots [1], Gazebo [2] and Robotics Toolbox developed by Peter Corke [3]. A common limitation associated with these simulators is that the dynamic model and the implemented control schemes are sometimes unknown. For this purpose, the multibody approach can be used to model the complex systems such robots as a mechanical system composed of rigid and/or deformable bodies connected by mechanical links. Simscape Multibody [4] is considered as a powerful tool frequently used to build high-fidelity digital twins of a wide range of physical systems including cardiovascular modelling [5], wind turbine [6, 7] and three wheeled vehicles [8]. It is also very widespread in the modelling of different categories of robots including AMRs. In fact, Simscape is integrated within the Matlab/Simulink toolbox and it provides a variety of features that make it well-suited for mobile robot modelling, including a library of blocks that can be used to model the kinematics, the dynamics,

the control and the interaction between the robot and the environment. Numerous research works are presented in the literature showing the application of Simscape Multibody in the modelling and simulation of different categories of robots such as manipulator robots [9-12], parallel robots [13-15] and legged robots [16-18].

For autonomous mobile robots, there is not enough research works that have used this software. Moreover, it can be noted that the simulations carried out with developed models essentially aim to analyze their behavior from a kinematic point of view. Such as the work presented in [19] which shows the design and the simulation collision avoidance of a three wheeled robot (Pioneer 3-DX) in the Simscape environment in which its motion is defined by adding actuation to wheels angular velocities. Also, in [20] a model for a three-wheeled omnidirectional robot is developed using Simscape Multibody software and used in a closed-loop control scheme to perform a desired path starting from desired position and calculating the required wheel velocities serving as inputs to the model. Another two wheeled non-holonomic robot (TurtleBotBurger3) presented in [21] is implemented and simulated in Simscape in order to follow different trajectories by adjusting the speeds of its motors. But for the design of control schemes and the sizing of robot actuators, dynamic analysis becomes essential. In this context, this study presents a common approach of numerical modelling based on the integration between the SolidWorks software and Simscape Multibody into Simulink to create a digital twin of a three wheeled mobile robot. The design is performed initially using SolidWorks to model mechanical structure of the robot. Then it is exported by Simscape to the Simulink environment which converts it to a multibody system in the form of a block diagram. Finally, a control scheme is applied to form a full model ready for simulation and tests purposes. The novelty of this work is that focused on the implementation of the dynamic model of the digital twin of the robot in addition to its kinematics in a control scheme design in order to predict its dynamic behavior and the motors torques necessary to perform a desired motion. This is performed also by taking into account the interaction of the robot with its environment by adding a contact model to define the friction between wheels and road. So, it is possible to predict the generated contact forces applied to wheels which have a great influence on the dynamic behavior of the robot.

This paper is organized as follows: Section 2 describes the process of building the digital twin of the three wheeled mobile robot. Section 3 presents the scheme of control implemented in order to perform a desired trajectory. In Section 4, dynamic simulations are carried out to verify the correctness of the digital twin model, comparing its outputs with desired ones. Then, they are used to analyze the dynamic behavior of the robot and predict the driven wheels torques controlled to track a desired motion. Finally, Section 5 is devoted to conclusions and perspectives.

2. DIGITAL TWIN MODEL IMPLEMENTATION

2.1 3D CAD Model of the Mobile Robot Prototype

The adopted methodology of designing a digital twin is started with the creation of a CAD model of the mobile robot using Solidworks [22] presented as the mostly used software for design and simulation of mechanical systems. The robot subject of this study is a three-wheeled differentially driven robot defined as multibody system and composed by five rigid bodies (see Figure 1). For more realistic simulation, the ground is modelled in the form of a thin plate as described in Figure 2. It will then be used to model the contact between the robot's wheels and the ground.

Based on the form and the defined material of each body, the inertial parameters can be obtained (see Table 1); mass, coordinates of center of mass with respect to the local reference frame and inertia tensor measured with respect to a frame with origin at the center of mass and axes parallel to the local reference frame. The resulting 3D CAD model of the robot described in Figure 3 is structured as an assembly of the previous rigid bodies interconnected through revolute joints by adding the different constraints.

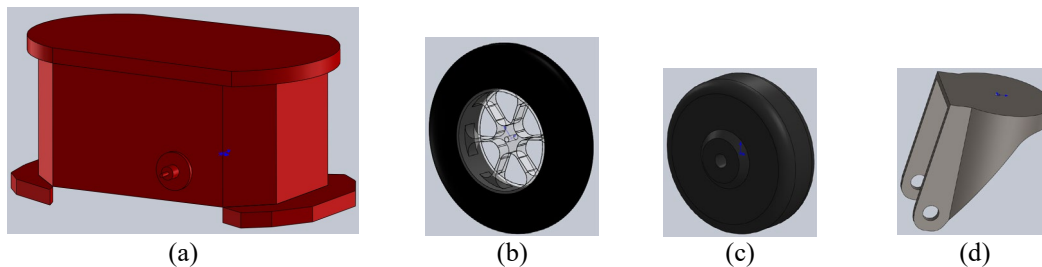


Figure 1. Components of the mobile robot and the ground part: (a) Chassis (a); (b) Two differential driven wheels; (c) Rear castor wheel; (d) For balancing and the castor wheel clamp.

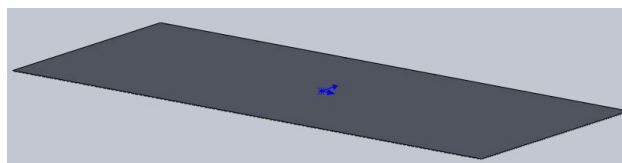


Figure 2. Ground part.

Table 1. Inertial parameters of robot bodies.

Body	Mass (kg)	Coordinates of center of mass (m)	Inertia tensor (kg.m ²)
Chassis	4.65	$\begin{bmatrix} -0.05 \\ 0 \\ 0.06 \end{bmatrix}$	$\begin{bmatrix} 65 & 0 & -15 \\ 0 & 120 & 0 \\ -15 & 0 & 107 \end{bmatrix} 10^{-3}$
Driven Wheel	0.65	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 3 \end{bmatrix} 10^{-3}$
Castor Wheel	0.08	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 24 & 0 & 0 \\ 0 & 24 & 0 \\ 0 & 0 & 42 \end{bmatrix} 10^{-6}$
Castor Wheel Clamp	0.07	$\begin{bmatrix} 0 \\ -0.006 \\ 0.014 \end{bmatrix}$	$\begin{bmatrix} 36 & 0 & 0 \\ 0 & 39 & -11 \\ 0 & -11 & 24 \end{bmatrix} 10^{-6}$

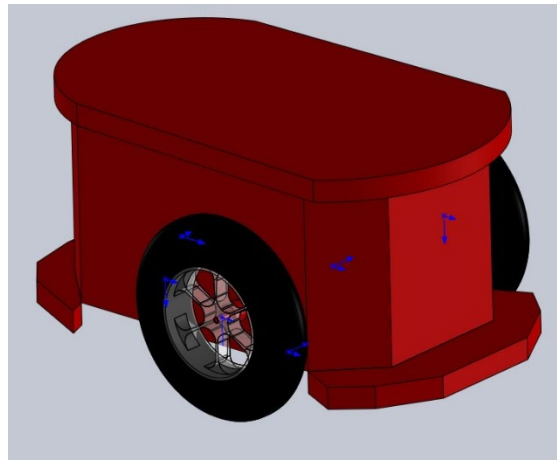


Figure 3. 3D CAD model of the three wheeled mobile robot.

2.2 Simscape Multibody Block Diagram of the Mobile Robot

In the next step, the 3D CAD model of the robot is exported from Solidworks by using the Simscape Multibody Link (Plug-In provided by Mathworks and installed on Solidworks) that generates an XML file of the assembly and an IGES file of each body. After that the XML file is imported into Matlab/Simulink via the command “smimport” which generates the blocks representing bodies, joints, and constraints between bodies. The initial imported Simscape Multibody model of the robot is presented in Figure 4.

It contains an environment module formed by a world frame node (1), a mechanism configuration node (2) to define the gravity field and a solver configuration node (3) used to set the parameters of the solver for the simulation. The blocks number (4) and number (5) perform the transformation of the robot and the ground coordinate systems respectively to the defined position and initial orientation in the global coordinate system. Block number (6) represents a 6-DOF (degrees of freedom) joint which allows the free motion of the robot in the space. Blocks number (7) and number (8) contain the 3D models of the chassis and the ground bodies developed in Solidworks. Blocks number (9) and number (10) represent revolute joints which perform the rotation of the left and right drive wheels. The 3D models of these wheels are developed using Solidworks and are represented by blocks number (12) and number (13). Block number (11) represents a revolute joint between the chassis and the castor wheel clamp that is presented by block number (14) and its design was done using Solidworks. Finally, block number (15) represents a revolute joint allowing the rotation of the rear castor wheel with respect to the castor wheel clamp and the block number (16) contain the 3D model of the castor wheel body developed using Solidworks.

The 3D visualization of the three wheeled mobile robot generating by Simscape Multibody in Mechanics Explorers is depicted in Figure 5 showing the attached frames and the center of masses of bodies.

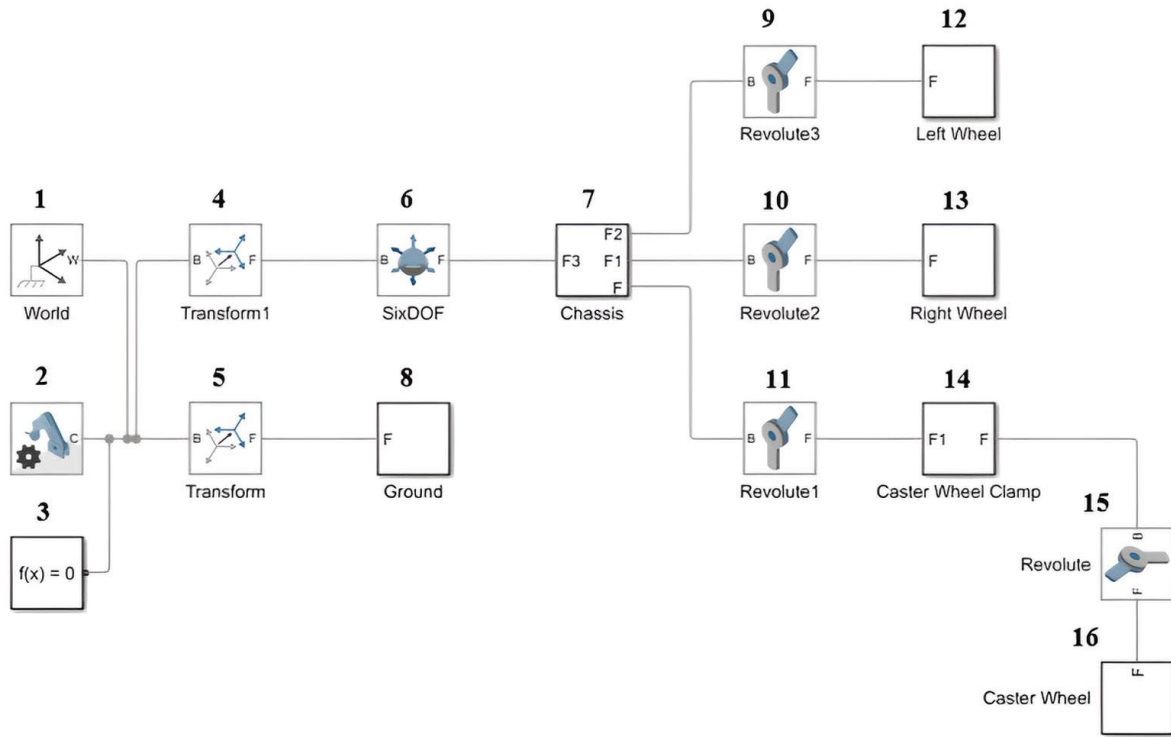


Figure 4. Initial Simscape Multibody block diagram of the robot.

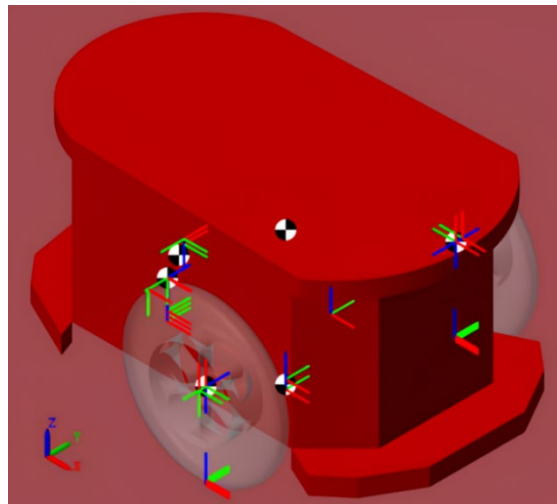


Figure 5. 3D visualization of the robot.

2.3 Physical Model of the Contact

The contact of the wheels and the ground is an important aspect for the study of the dynamic behavior of the robot. For this purpose, a physical model block diagram is added from the Simscape Multibody “Contact Forces Library” in order to calculate the interaction between wheels of the robot and ground. In fact, without this model, during simulation the robot falls through the ground due to the gravitational force. From this library, a 3D block “Sphere to plane Force” (see Figure 6) is applied to simulate how surfaces of the wheels of the robot interact with ground in the real world. This block implements a contact force between a sphere and a plane. In our model, three blocks are added to the initial Simscape Multibody block diagram of the robot as shown in Figure 7. The three wheels of the robot act as sphere, and the ground surface is considered as a plane. Consequently, the robot can move on the ground and not fall through it.

The contact model implemented in the block “Sphere to plane Force” is based on the Kelvin-Voigt model [23, 24] in which the contact between bodies is considered deformable and defined using a visco-elastic approach. According to this model, the normal contact force is calculated as the sum of a linear elastic force component and a damping force component as follows:

$$F_N = k\delta + d\dot{\delta} \tag{1}$$

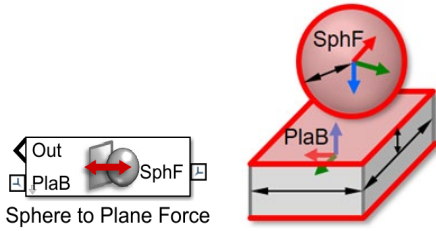


Figure 6. 3D block “Sphere to plane Force”.

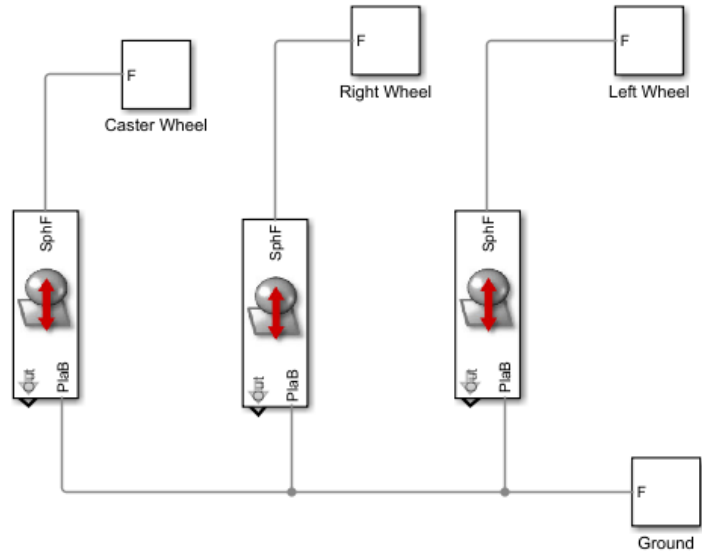


Figure 7. Adding the “Sphere to plane Force” block to the multibody block diagram of the robot.

where $k = 150000$ N/m is the stiffness of contact and $d = 20000$ Ns/m is the damping coefficient [25]. δ is the interpenetration depth between bodies in contact (m) and $\dot{\delta}$ is the interpenetration velocity (m/s).

For the frictional contact force, the stick–slip continuous friction law is used in the “Sphere to plane Force” block to calculate it as a function of sliding velocity, v based on the exponential (Stribeck-type) model [26, 27]. This model allows friction to be continuously reduced with increasing relative speed for a certain speed regime:

$$F_f = \mu(v)F_N \operatorname{sgn}(v) \tag{2}$$

$$\mu(v) = \mu_k + (\mu_s - \mu_k)e^{-\left(\frac{|v|}{v_s}\right)} \tag{3}$$

where $\mu(v)$ is the velocity-dependent friction coefficient, $\mu_s = 0.5$ is the static friction coefficient, $\mu_k = 0.3$ is kinetic friction coefficient and $v_s = 0.01$ m/s is the Stribeck velocity.

3. DYNAMIC CONTROL SYSTEM DESIGN

The effectiveness of the digital twin of our robot developed in Simscape Multibody is illustrated through the simulation of its dynamic response when following a desired path. The last one can be easily obtained using a simplified kinematic model of a three wheeled mobile robot as described in [28 - 30]. The trajectory-tracking problem is posed as follows: firstly, the desired angular velocities of the two driven wheels are adjusted to follow a predefined path according to the kinematic model. Then, it will be used in feedback control loop with a simple PI controller that will provide as output the required motor torques (τ_r and τ_l) to control the velocity of each wheel as mentioned in the control simulation schematic diagram (see Figure 8).

The controlled driving torque are calculated as follows:

$$\tau_r = K_{pr}e_r + K_{ir} \int e_r dt \tag{4}$$

$$\tau_l = K_{pl}e_l + K_{il} \int e_l dt \tag{5}$$

where K_p and K_i are the PI controller parameters and e is the error between the desired angular velocity and the measured one. The subscripts r and l denote the right and left wheels respectively. The gains of the PI controller are obtained manually using trial and error method [31]. In the initial Simscape Multibody block diagram of the robot depicted in Figure 4, the joints relative to the driven wheels (Revolute 2 and Revolute 3) are not actuated neither in position nor in torque. Since we are interested in dynamic simulations the torque is defined as provided by input and the position is automatically computed. Figure 9 shows the modified actuation of the described joints by adding a feedback control loop using PI controller to calculate the controlled motor torque applied in each wheel. The added Simulink-Physical Signal (PS) and PS-Simulink Converters blocks are used to manage the flow and conversion between a unit-aware PS and a standard Simulink signal, which is usually unitless. The desired angular velocities of the two driven wheels are generated using functions implemented in the Embedded MATLAB functions blocks.

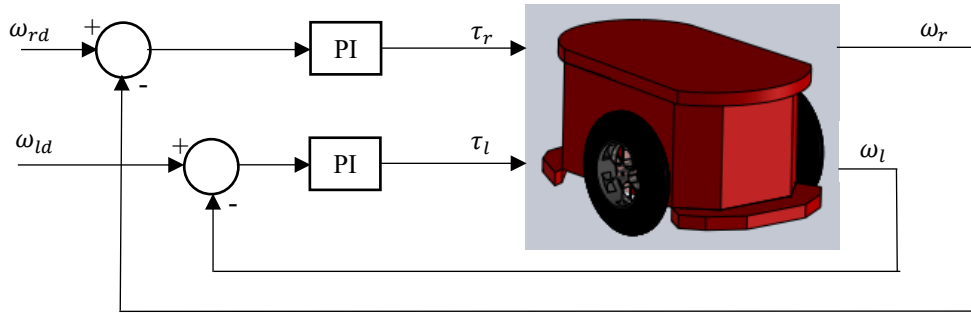


Figure 8. Control simulation schematic diagram.

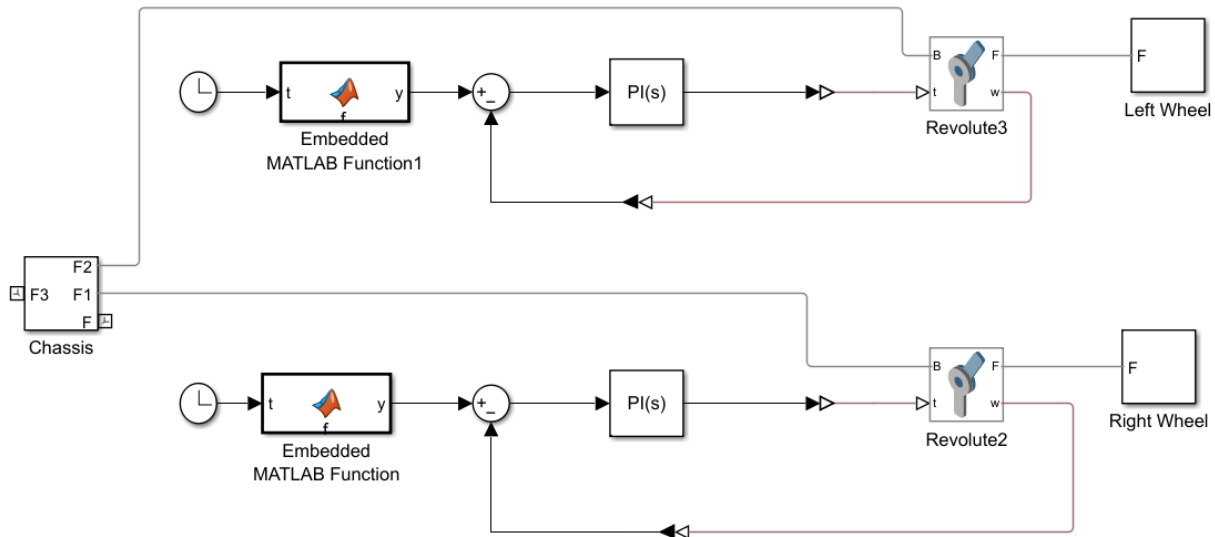


Figure 9. Torque actuation of the driven wheels' joints.

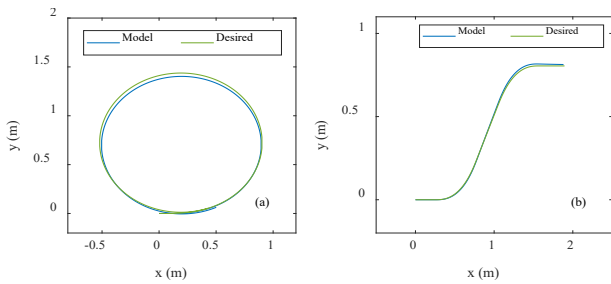


Figure 10. Trajectory tracking: (a) Circular; (b) Avoidance obstacle.

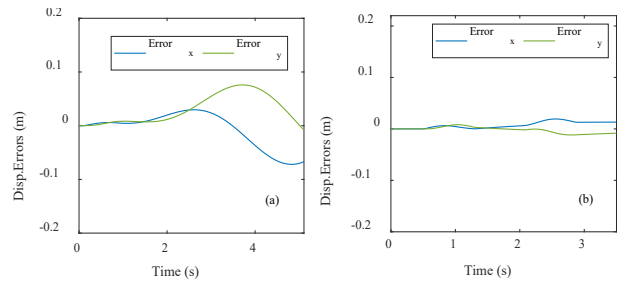


Figure 11. Displacement errors in x and y coordinates: (a) Circular; (b) Avoidance obstacle.

4. SIMULATION OF TRAJECTORY TRACKING

As described in Section 3, based on the kinematic model, two reference trajectories were selected by adjusting the desired angular velocities of the two driven wheels: The first one is a circular trajectory in which the robot will start from the initial position (0,0,0) with initial speed 0.5 m/s during a 0.1 s. Then it enters in a circular profile with a desired radius of 0.7 m and a desired speed of 1 m/s. The second one is an avoidance obstacle trajectory in which the robot moves from its initial position (0,0,0) in straight line with initial speed 0.5 m/s during 0.1 s then it will turn to the left in the first time and to the right in the second time. Finally, the robot returns to move in straight line with the initial speed 0.5 m/s.

All simulation results obtained for both paths are shown in this section. As seen in Figure 10, the robot has successfully followed both reference trajectories with slight displacement errors in x and y coordinates as shown in Figure 11. Also, the correctness of the robot motion is verified by the curves representing the orientation angle of the robot. It is clear from Figure 12(a) that the yaw angle continues to increase gradually until reaching the value (2π) which corresponds to the circular profile closure by the robot. For the avoidance obstacle trajectory, regarding Figure 12(b), the yaw angle will start and end with a zero value with an intermediate phase for which it takes a constant value.

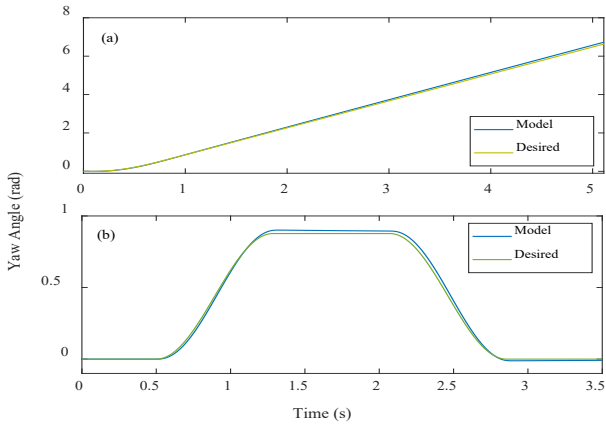


Figure 12. Orientation angle of the robot: (a) Circular; (b) Avoidance obstacle.

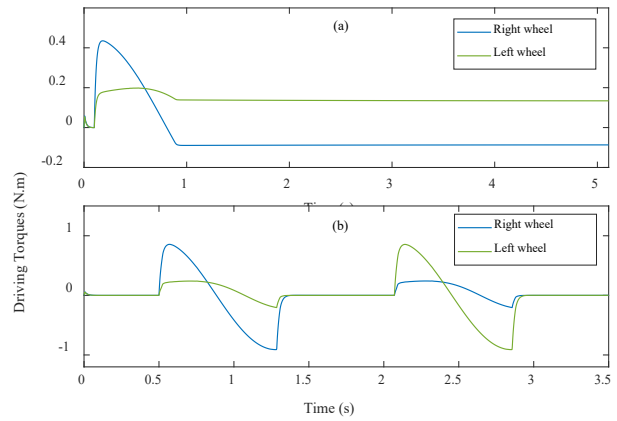


Figure 13. Controlled driving torques: (a) Circular; (b) Avoidance obstacle.

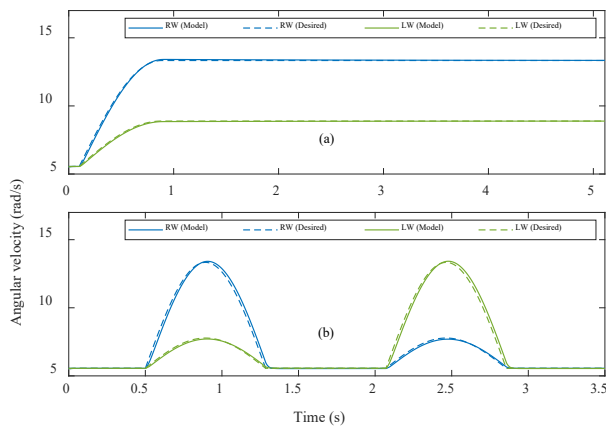


Figure 14. Wheels angular velocities: (a) Circular; (b) Avoidance obstacle (RW: Right wheel, LW: Left wheel).

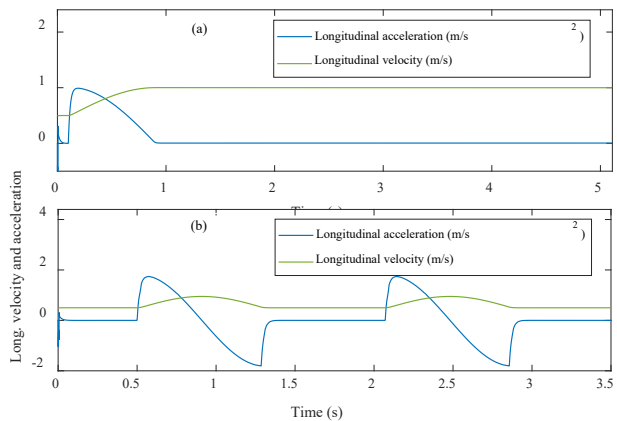


Figure 15. Longitudinal velocity and acceleration: (a) Circular; (b) Avoidance obstacle.

On other hand, it can be noted that the PI controller has correctly controlled the driving torques depicted in Figure 13 which are associated with regulating the angular speeds of the wheels to match the desired ones in order to execute the reference paths as shown in Figure 14. The resulting longitudinal velocity and acceleration of the robot are plotted in Figure 15. It is clear for the circular path that the longitudinal speed converges towards the desired one (1 m/s). For the avoidance trajectory case, the longitudinal velocity increases from 0.5 m/s to 1 m/s twice due to the change in the robot direction to the left in the first time and to the right in the second time. In other hand we can identify the phases of acceleration and deceleration carried out by the robot when turning to the right or to the left. These phases correspond well to the curves of driving torques obtained.

By examining Figure 13(a), we can note that after the acceleration, a negative driving torque for the right wheel and a positive one for the left wheel are obtained despite the speed being constant. This can be explained by the fact that in a circular path the centrifugal forces push the robot outwards so to counteract them the robot requires increased inner-wheel torque or decreased outer-wheel torque to stay on track and maintain the desired radius. For this purpose, in our case (circular trajectory to the left) the controller must adjust the driving torques as follows: the outer driving torque corresponding to the right wheel is decreased and takes a negative value and the inner driving torque corresponding to the left wheel increased and takes a positive value.

To predict the robot's performance, stability and interaction with its environment, ground reaction forces are determined based on the contact model introduced in Simscape Multibody. As shown in Figure 16, the obtained normal forces acting on driven wheels are equal when the robot move straight ahead. But in case of turning to the left, the normal force is decreased for the left wheel and increased for the right one due to the centrifugal acceleration and vice versa for a right turn. This corresponds to the dynamics of road vehicles and it proves the correctness of the robot's dynamic behavior.

The previously observed errors in displacement and orientation are particularly evident in the presence of the slipping phenomenon resulting from the friction forces generated at the contact surface between wheels and ground. These forces are presented with respect to the contact frames in Figure 17 and Figure 18 for both studied trajectories.

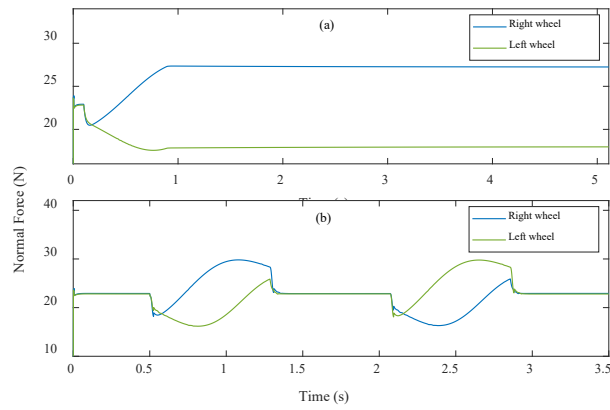


Figure 16. Normal force acting on driving wheels: (a) Circular; (b) Avoidance obstacle.

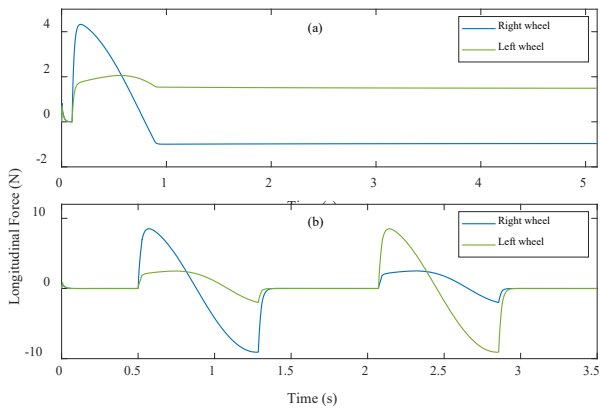


Figure 17. Longitudinal force acting on driving wheels: (a) Circular; (b) Avoidance obstacle.

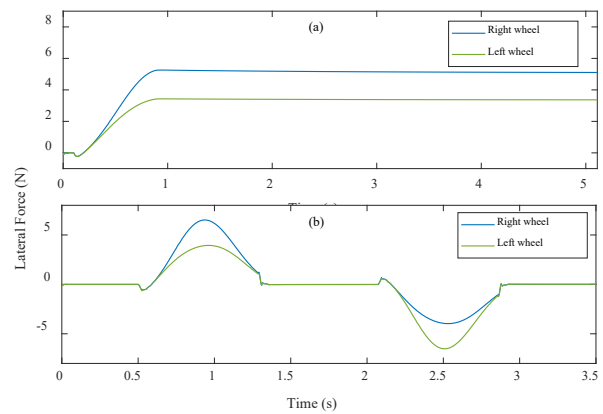


Figure 18. Lateral force acting on driving wheels: (a) Circular; (b) Avoidance obstacle.

It can be seen that the right-side wheel needs more longitudinal and lateral forces than the left wheel to make a left turn and inversely for a right turn. This difference occurs during acceleration or braking as shown in Figure 15, leading to both longitudinal slip along the direction of motion and lateral slip which is perpendicular to the direction of motion. In addition, longitudinal forces have the same shape of driving torque (see Figure 13) and longitudinal acceleration (see Figure 15) by taking positive values in acceleration and negative values when decelerate.

In this study, the developed digital twin for the mobile robot offers significant improvements for more efficient simulations. It allows better visualization of the robot's motion in the created virtual reality environment and a high-performance analysis of its kinematic and dynamic behavior. Thus, facilitating the development and the test of control strategies and reduces the complexity and number of steps in the process. This helps to improve overall performance in terms of cost and time.

5. CONCLUSION

This paper shows a methodology of designing a digital twin of a three wheeled robot using Simscape Multibody and Matlab/Simulink tools to simulate two reference trajectories. The reliability of simulation results is enhanced by a comparison with desired values obtained using a simple kinematic model. Therefore, the digital twin demonstrates feasible effectiveness to predict the dynamic behavior of the robot and the interaction with its environment. The current work focuses on a comprehensive numerical analysis. So, an experimental validation is a significant development to be pursued in subsequent studies. Future studies will focus on the application of other advanced controllers to demonstrate the usefulness of the developed model. In a next investigation, we proceed to do a sensitive analysis, firstly, by considering the geometric and inertial parameters of the robot and secondly, by varying the wheel/ground contact model and parameters.

FUNDING

The authors receive no financial support for the research, authorship, and publication of this article.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

- [1] Cyberbotics, Webots software. <https://www.cyberbotics.com>, 2026 (accessed 11.04.2026).
- [2] Gazebo, Gazebo software. <https://www.gazebosim.org>, 2026 (accessed 11.04.2026).
- [3] P. Corke, Robotics Toolbox. <https://www.petercorke.com/toolboxes/robotics-toolbox/>, 2026 (accessed 11.04.2026).
- [4] MathWorks, Simscape Multibody. <https://www.mathworks.com/products/simscape-multibody.html>, 2026 (accessed 11.04.2026).
- [5] J. F. De Canete, P. Del Saz-Orozco, D. Moreno-Boza and E. Duran-Venegas, Object-oriented modeling and simulation of the closed loop cardiovascular system by using SIMSCAPE, *Computers in Biology and Medicine*, 43(4), 2013, 323-333.
- [6] E. Jamila and S. Abdelmjid, Physical modeling of a hybrid wind turbine-solar panel system using simscape, *International Journal of Engineering-Transactions B: Applications*, 27(11), 2014, 1767-1776.
- [7] E. Jamila and S. Abdelmjid, Comparative study of the performance of static synchronous compensator, series compensator and compensator/battery integrated to a fixed wind turbine, *International Journal of Engineering-Transactions A*, 29(4), 2016, 581-589.
- [8] M. Waseem, M. Suhaib and A. F. Sherwani, Modelling and analysis of gradient effect on the dynamic performance of three-wheeled vehicle system using Simscape, *SN Applied Sciences*, 1(3), 2019, 225.
- [9] B. Giovanni and S. Teresa, Designing digital twins of robots using Simscape multibody, *Robotics*, 13(4), 2024, 62.
- [10] Z. Cheng, L. Songxiao and Z. Zhuo, Modeling and simulation of industrial robot arms using Simscape Multibody, *Facta Universitatis, Series: Mechanical Engineering*, 23(2), 2025, 265-286.
- [11] L. N. Truc and N. T. Lam, Quasi-physical modeling of robot IRB 120 using Simscape Multibody for dynamic and control simulation, *Turkish Journal of Electrical Engineering and Computer Sciences*, 28(4), 2020, 1949-1964.
- [12] A. S. Reddy, V. V. M. J. S. Chembuly and V. V. S. K. Rao, Collision free inverse kinematics of redundant manipulator for agricultural applications through optimization techniques, *International Journal of Engineering*, 35(7), 2022, 1343-1354.
- [13] M. Zafer, R. A. Mohammad, N. Vahab and R. Mohammad, A comparative study of two inverse dynamic models of 6 degree-of-freedom rotary Stewart-Gough parallel robot, *International Journal of Dynamics and Control*, 10, 2021, 377-390.
- [14] O. John, P. Nicolás, F. A. Oscar and C. Juan, Analysis of 3 RPS robotic platform motion in Simscape and Matlab GUI environment, *International Journal of Applied Engineering Research*, 12(8), 2017, 1460-1468.
- [15] P. Noskievič and D. Walica, Design and realisation of the simulation model of the Stewart platform using the Matlab-Simulink and the Simscape Multibody library, *21th International Carpathian Control Conference*, High Tatras, Slovakia, 2020, 1-5.
- [16] R. Singh and T. K. Bera, Walking model of Jansen mechanism-based quadruped robot and application to obstacle avoidance, *Arabian Journal for Science and Engineering*, 45, 2020, 653-664.
- [17] A. A. Ammar, A. M. Auday and H. J. Basil, Control of eight-leg walking robot using fuzzy technique based on SimScape Multibody toolbox, *IOP Conference Series: Materials Science and Engineering*, 745, 2019, 012015.
- [18] M. Zulkifli, I Addie, M. A. G. Nor and A. A. H. Azrul, Modeling and analysis of bipedal wheel-legged robot with sprawling legs mechanism, *9th IEEE International Conference on Mechatronics Engineering*, Kuala Lumpur, Malaysia, 2024, 139-144.
- [19] H. A. Biswas, S. W. Nawawi, A. A. Ahmed and M. A. Jibrin, Modeling and simulation of autonomous mobile robot system for collision avoidance in VR environment using Simscape-Multibody, *Journal of Electrical Engineering (ELEKTRIKA)*, 21(2), 2022, 33-44.
- [20] M. B. Emara, A. W. Youssef, M. Mashaly, J. Kiefer, L. A. Shihata and E. Azab, Digital twinning for closed-loop control of a three-wheeled omnidirectional mobile robot, *55th CIRP Conference on Manufacturing Systems- Procedia CIRP*, 107, 2022, 1245-1250.
- [21] K. Khnissi, C. B. Jabeur and H. Seddik, 3D simulator for navigation of a mobile robot using Simscape-Simulink, *IEEE International Conference on Control, Automation and Diagnosis*, Grenoble, France, 2019, 1-6.
- [22] Dassault systems, Solidworks software, <https://www.solidworks.com>, 2026 (accessed 11.04.2026).
- [23] E. Corral, R. G. Moreno, J. Meneses, M. J. G. García and C. Castejón, Spatial algorithms for geometric contact detection in multibody system dynamics, *Mathematics*, 9(12), 2021, 1359.
- [24] M. Machado, P. Moreira, P. Flores and H. M. Lankarani, Compliant contact force models in multibody dynamics: evolution of the hertz contact theory, *Mechanism and Machine Theory*, 53, 2012, 99-121.
- [25] H. D. Choi, H. Kang, K. H. Hyun, S. Kim and Y. K. Kwak, Force distribution estimation of wheeled mobile robot: Application to friction coefficients estimation, *IFAC Proceedings Volumes*, 41(2), 2008, 10451-10455.
- [26] L. Marton and B. Lantos, Control of mechanical systems with Stribeck friction and backlash, *Systems & Control Letters*, 58(2), 2009, 141-147.
- [27] S. Hu and Y. Liu, Disc brake vibration model based on Stribeck effect and its characteristics under different braking conditions, *Mathematical Problems in Engineering*, 2017(1), 2017, 6023809.
- [28] A. Khadr, Multibody dynamics modeling and simulation of a three-wheeled mobile robot using a robotic approach, *International Journal of Intelligent Unmanned Systems*, 12(3), 2024, 270-289.
- [29] J. Cerkala and A. Jadlovska, Nonholonomic mobile robot with differential chassis mathematical modelling and implementation in Simulink with friction in dynamics, *Acta Electrotechnica et Informatica*, 15(3), 2015, 3-8.
- [30] R. Dhaouadi and A. A. Hatab, Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework, *Advances in Robotics & Automation*, 2(2), 2013, 1-7.

- [31] G. M. Van Der Zalm, *Tuning of PID-type controllers: literature overview*. DCT Rapporten 2004.054, Technische Universiteit Eindhoven, 2004.