The 11th International Congress
Motor Vehicles & Motors 2026
Kragujevac, Serbia
October 8th - 9th, 2026

Motor Vehicles & Motors

MVM2026-066
https://doi.org/10.24874/CMVM2026.066
Original scientific paper

**Matija Simanić**[1]
**Ilija Vidović**[2]
**Luka Milić**[3]
**Momir Drakulić**[4]
**Aleksandar Đurić**[5]

# SIMULATION-BASED MOTION ANALYSIS OF A LIGHTWEIGHT TRACKED ROBOT USING WEBOTS

**ABSTRACT:** This paper presents the modelling of a lightweight tracked robot and the simulation of its motion using the Webots software package. The robot model was developed in Webots based on an existing model of the Jaguar robot, while the simulation was performed on a scaled 3D representation of the Military Academy grounds in Belgrade. The motion analysis includes the analysis of various kinematic and dynamic parameters, such as the robot's trajectory, travelled distance, speed and acceleration, orientation angle, and longitudinal motion force. Using the Python programming language, a controller was developed directly in Webots to enable the robot's motion and data collection for motion analysis. A single simulation run was performed using one complex trajectory. The robot travelled approximately 143 m during 97 s of simulation, with an average speed of 1.47 m/s compared with the set speed of 1.5 m/s, corresponding to a relative deviation of approximately 2.0%. The consistency of the developed model was assessed based on the obtained kinematic and dynamic parameters.

**KEYWORDS:** Tracked robot, Webots, Kinematic parameters, Dynamic parameters, Python.

# ANALIZA KRETANJA LAKOG GUSENIČNOG ROBOTA ZASNOVANA NA SIMULACIJI KORIŠĆENJEM WEBOTS PLATFORME

**REZIME**: Ovaj rad predstavlja modeliranje lakog gusenič nog robota i simulaciju njegovog kretanja korišćenjem softverskog paketa Webots. Model robota je razvijen u Webots-u na osnovu postojećeg modela robota Jaguar, dok je simulacija izvršena na skaliranoj 3D reprezentaciji kompleksa Vojne akademije u Beogradu. Analiza kretanja

[1] *Matija Simanić, University of Defence in Belgrade, Military Academy, msimanic4@gmail.com*
[2] *Ilija Vidović, University of Defence in Belgrade, Military Academy, ilijavidovic50@gmail.com*
[3] *Luka Milić, University of Defence in Belgrade, Military Academy, luka.milic777@gmail.com*
[4] *Momir Drakulić, University of Defence in Belgrade, Military Academy, momir.drakulic@va.mod.gov.rs,* https://orcid.org/0000-0002-8367-7281
[5] *Aleksandar Đurić, University of Defence in Belgrade, Military Academy, aleksandar.djrc@gmail.com,* https://orcid.org/0000-0002-2165-528X

obuhvata analizu različitih kinematskih i dinamičkih parametara, kao što su: putanja robota, pređeni put, brzina i ubrzanje, ugaoni položaj i sila uzdužnog kretanja. Koristeći programski jezik Python, u Webots-u je razvijen kontroler koji omogućava kretanje robota i prikupljanje podataka za analizu kretanja. Jedna simulacija je izvršena korišćenjem jedne složene putanje. Robot je prešao približno 143 m tokom 97 s simulacije, sa prosečnom brzinom od 1,47 m/s u poređenju sa podešenom brzinom od 1,5 m/s, što odgovara relativnom odstupanju od približno 2,0%. Konzistentnost razvijenog modela je procenjena na osnovu dobijenih kinematskih i dinamičkih parametara.

**KLJUČNE REČI:** gusenični robot, Webots, kinematski parametri, dinamički parametri, Python

# INTRODUCTION

Bruzzone et al. (2022) state that tracked robots are widely used for operation on uneven and soft terrain, since their specific ground pressure is significantly lower than that of wheeled robots, while they are more mobile than legged robots. In addition, they provide greater stability and better obstacle-crossing capability. Due to these characteristics, they are widely used in military, rescue, and other tasks performed in complex environments.

Simulators are widely used for conducting tests due to their significant advantages over testing with physical prototypes. Some of the main advantages include:

- Lower financial costs,
- Reduced testing time,
- The possibility of repeating tests,
- Rapid modification of simulation parameters,
- Determination of the required characteristics of a real system.

Gunjal (2021) demonstrated that motion simulations can be performed using the Webots software package, which was also used in this work. Other software tools that can be used include Simulink within the MATLAB software package, as described by Haidar (2024), as well as Gazebo, which Rivera et al. (2019) describe as an important software tool in mobile robotics due to its high flexibility and realistic simulation of physical processes.

Webots enables the simulation of both passive and active objects, as well as the use of different types of sensors and control mechanisms. Control algorithms can be implemented using different programming languages, while Python was selected for the purposes of this study. The physics engine enables physical characteristics such as moments, forces, contact between bodies, and rigid-body motion dynamics to be taken into account. Webots also enables the simulation of contact between the ground and the tracked locomotion system.

The aim of this work is to develop a tracked robot model based on the real Jaguar robot and to create a simulation environment representing the Military Academy grounds, in which complex robot motion can be simulated. Finally, the kinematic and dynamic parameters obtained from the simulation are analyzed and the consistency of the developed robot model is assesed. The main contribution of this work is the development and application of a Webots-based model that combines a modified tracked robot representation with a scaled 3D terrain for the analysis of complex robot motion. The study investigates whether the developed model can adequately reproduce the expected kinematic and dynamic behavior of the tracked robot under the considered simulation conditions.

## Kinematic model

Hellström (2011) states that the motion of tracked robots is based on the principle of differential steering. Separate drive mechanisms are present on the left and right tracks, and therefore the direction of motion is controlled by changing the relative speeds of the left and right tracks. Thus, if both tracks have the same speed and direction of motion, the robot moves in a straight line, while if the tracks have different speeds, the robot follows a circular trajectory. Bruzzone et al. (2022) also describe a steering condition in which the tracks have the same speed but opposite directions of motion, causing the robot to rotate about its own vertical axis. This type of steering was also used to control the robot during the simulation.

Kozłowski and Pazderski (2004) describe skid-steer steering as a more realistic approach to modeling tracked robot motion. It is based on the principle of differential steering, with lateral slipping occurring due to the interaction between the robot and the ground, as described by Bruzzone et al., (2022)

In the present simulation, differential steering is used as the control principle, while lateral track slip is not explicitly defined through a separate skid-steering model. Instead, the interaction between the tracked locomotion system and the terrain is handled implicitly by the Webots physics engine through the Track and Physics nodes. Therefore, the

kinematic equations presented in this section represent the idealized differential-steering model, while the simulated dynamic response additionally reflects the physical interaction implemented in Webots.

The position of a mobile robot is defined using a coordinate system. Thus, the robot can be represented by the coordinates of its center and its orientation angle with respect to the coordinate system. The state vector is commonly defined in the following form:

$$q = [x, y, \theta] \tag{1}$$

Figure 1 presents the kinematic model of a tracked robot with differential steering. The linear velocities of the left and right tracks, denoted as $v_{left}$ and $v_{right}$, respectively, are shown, as well as the angular velocity of the robot about its vertical axis, denoted by $\omega$. The position and orientation of the robot are defined with respect to the coordinate system ($x_w$,$y_w$), while the angle $\vartheta$ represents the robot orientation with respect to the coordinate system, i.e., its direction of motion. The track width, denoted by $w$, can also be observed.

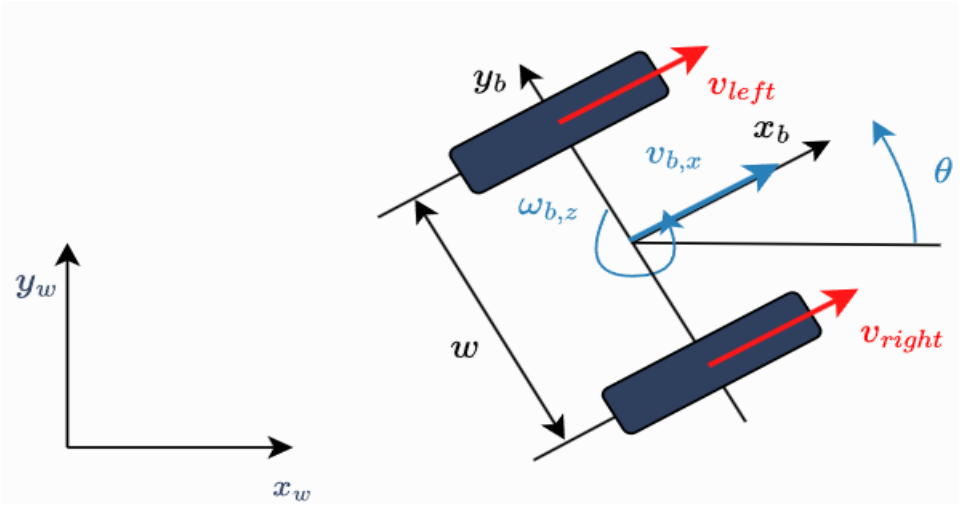



***Figure 1*** Kinematics of a differentially controlled tracked robot *(ROS 2 Control maintainers, n.d.)*

According to Muždeka and Rakić (2008), the linear velocity of the robot center is calculated as the average of the left and right track velocities, while the difference between the left and right track velocities divided by the track width represents the angular velocity of the robot. This can be expressed by the following equations.

$$v = \frac{v_{left} + v_{right}}{2} \tag{2}$$

$$\omega = \frac{v_{left} - v_{right}}{w} \tag{3}$$

Based on these equations, the equation for the robot's turning radius can be derived as follows:

$$R = \frac{w}{2}\frac{v_{left} + v_{right}}{v_{left} - v_{right}} \tag{4}$$

This equation confirms that when the track velocities are approximately equal, the robot tends to move in a straight line, while an increasing difference between the track velocities results in a circular trajectory with a smaller turning radius.

## Modelling of the tracked robot and simulation environment in Webots

As previously mentioned, the aim of this work was to develop a tracked robot model in the Webots software package based on the real Jaguar robot, perform a simulation of complex robot motion, and subsequently analyse the obtained results. The modelling and simulation were performed using Webots R2025a with the Open Dynamics Engine (ODE) physics engine. The robot modelling process began with the selection of the existing SurveyorSrv1 tracked robot model, shown in Figure 2.

The existing upper structure with sensors and cameras was first removed from this robot. This modification, as well as all subsequent changes, was performed within the PROTO node, which is used to modify and define the characteristics of the robot. In order to obtain the final appearance of the robot, its physical, geometric, and structural

characteristics were modified, including the track width, wheel diameter, chassis dimensions, robot colour and mass, as well as the addition of a camera mount with a camera positioned at the same height as on the real robot. Table 1 represents robot parameters that are modified in PROTO node.

***Table 1*** *Robot parameters*

| | |
|---|---|
| Robot mass [kg] | 14 |
| Distance between tracks [m] | 0.370 |
| Track wheel radius [m] | 0.075 |
| Track width [m] | 0.059 |
| Distance between front/rear track wheels [m] | 0.495 |

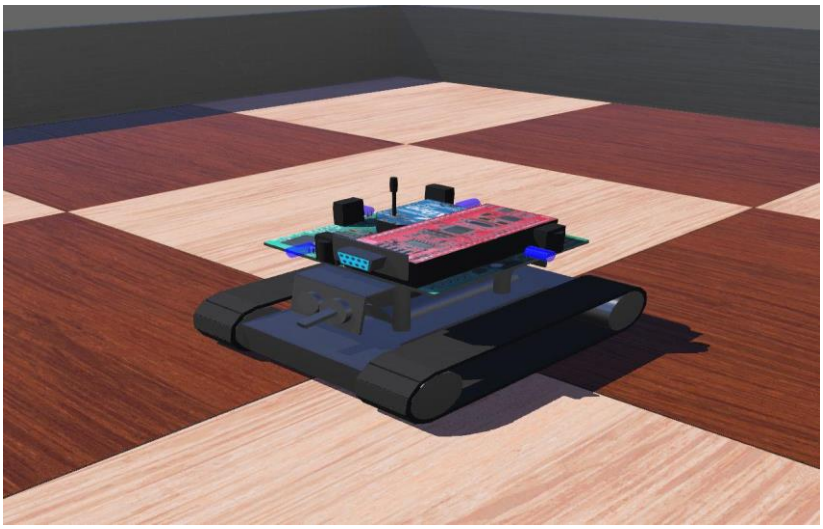


***Figure 2*** Tracked robot SurveyorSrv1

This approach resulted in a model that closely corresponds to the real Jaguar robot. The final appearance of the tracked robot modelled in Webots is shown in **Error! Reference source not found.**, while **Error! Reference source not found.** presents the real Jaguar robot.

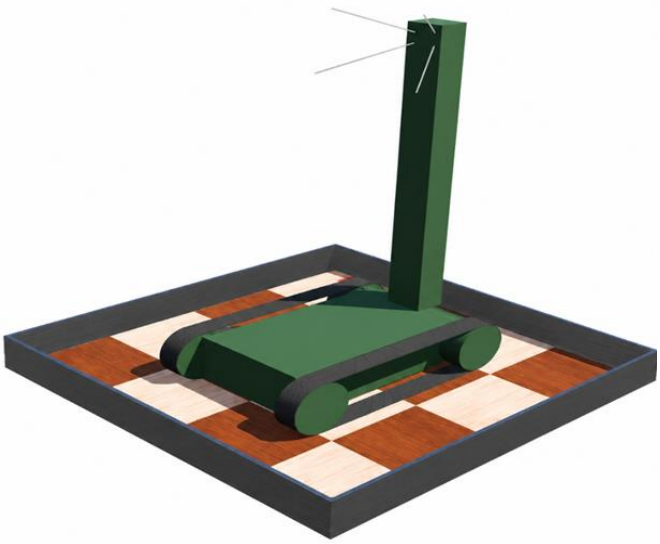


***Figure 3*** Model of the tracked robot in Webots



***Figure 4*** Model of the real tracked robot Jaguar

To achieve realistic contact between the ground and the tracked locomotion system, the Track node was used. It enables the definition of the track shape and trajectory, the addition of drive elements, animation of the tracked mechanism, and simulation of its interaction with the ground. LinearMotor elements were used to control the motion, while TrackWheel elements were used to define the drive and idler wheels. According to Cyberbotics Ltd., n.d. (2026, May 26), the animatedGeometry parameter defines the appearance and geometry of the animated objects, while geometriesCount defines the number of animated objects that move along the track trajectory. In this work, these parameters were used for the visual representation of the tracks.

The boundingObject element was used to define the collision geometry of the chassis and, more importantly, the tracked locomotion system. An important feature of boundingObject is that it does not have to exactly follow the geometry of the object whose collision it represents, but can instead approximate its physical behaviour. In this work,

this feature was used to define the collision object of the tracked locomotion system, which was represented by two cylinders and one box element. The cylinders follow the geometry of the wheels, while the box provides contact with the ground along the entire length of the tracked system.

In order to enable contact and motion of the robot, as well as to take other physical laws into account, the Physics node was included.

Table 2 represents specified values in Webots nedeed for simulation.

***Table 2*** *Specified values in Webots*

| | |
|---|---|
| Robot speed [m/s] | 1.5 |
| Friction coefficient | 0.8 |
| Rolling resistance coefficient | 0.05 |

No experimental measurements from the real Jaguar robot were used for calibration or quantitative validation of the simulation model. The model parameters were defined based on the available physical and geometric characteristics of the real robot, while the consistency of the developed model was assessed through the internal consistency of the simulated kinematic and dynamic parameters.

The scaled representation of the Military Academy grounds was created using the Blosm add-on within the Blender software package. It was used to obtain a 3D model of the terrain and its real configuration. Additional modifications of the terrain were then performed in Blender to facilitate its use in the simulation. The prepared terrain was exported in the .obj format and imported into Webots, where the Solid and Mesh nodes were used to display and utilize the complex terrain within the simulation. Here, the boundingObject element was also required to provide the reaction of the ground to the tracked locomotion system and thus enable its motion. **Error! Reference source not found.** shows the terrain imported into Webots.

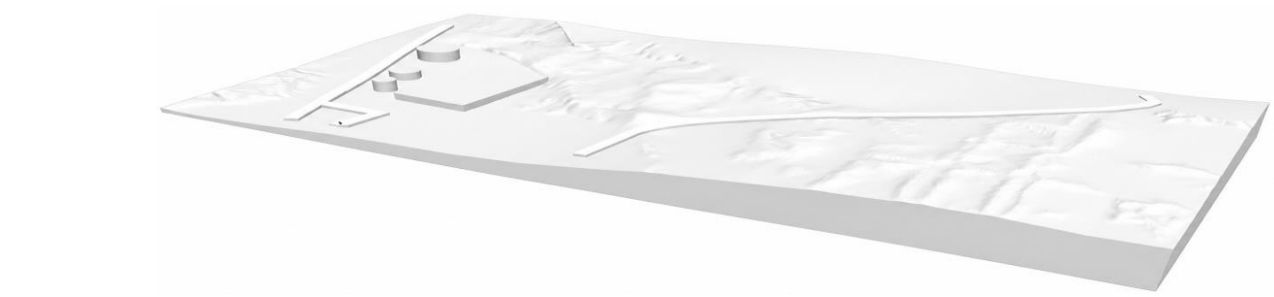


***Figure 5*** Display of scaled-down terrain of the Military academy

## Simulation setup and data collection

Robot motion control was implemented using the Python programming language and the Webots Python API. The API provides access to nodes, sensors, actuators, and other simulation elements through the controller module, enabling robot motion and control within the simulation. Motion control was achieved by changing the speeds of the left and right tracks using the LinearMotor element. The robot was controlled manually using keyboard commands:

- W – forward motion,
- S – backward motion
- D – turning right,
- A – turning left.

Since the robot was controlled manually using keyboard commands, the exact trajectory cannot be reproduced with the same precision in repeated simulations. Therefore, the presented results are representative of the considered simulation scenario rather than a statistically repeatable experimental dataset. Automated trajectory generation and predefined control inputs would enable more objective comparison and improved repeatability in future studies.

In addition, the control code was used to collect the data required for motion analysis. Knowing this it is clear that a single simulation was run on a complex trajectory.

The following parameters were recorded:

- time,
- lateral displacement,
- longitudinal displacement,
- horizontal displacement,

- travelled distance,
- speed,
- acceleration,
- orientation angle,
- terrain inclination angle,
- acceleration-induced force,
- slope-induced force,
- rolling resistance force,
- total longitudinal force acting on the robot.

The simulation was performed using a single complex motion on the imported terrain. The complex motion consisted of straight-line motion, turns, downhill motion, and uphill motion. To facilitate the interpretation and verification of the validity of the obtained simulation results, Figure 6 presents the robot's trajectory during the simulation. This trajectory enables the obtained diagrams to be related to specific sections of the robot's motion, including straight-line motion, turning, uphill, and downhill sections.

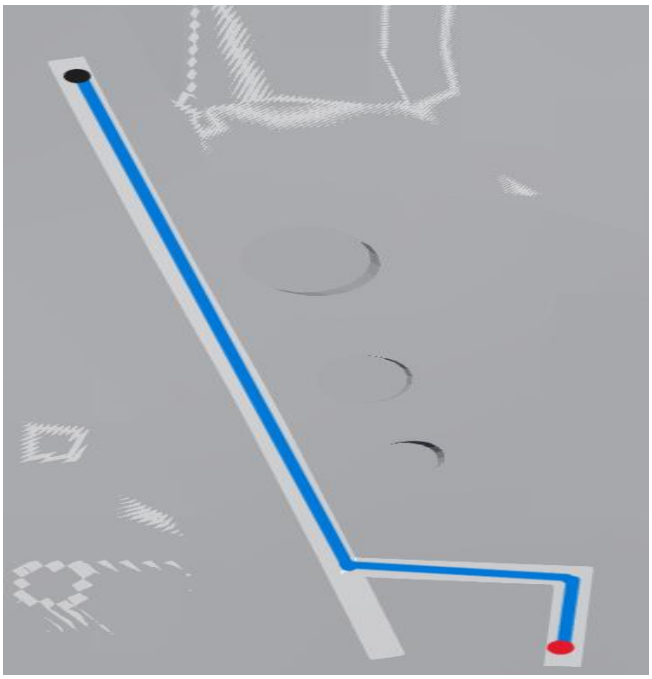


***Figure 6*** Robot trajectory during the simulation

## Results and Discussion

At the beginning of the simulation, the robot control code began recording the data described in the previous section. The simulation results were obtained directly from the control code and used to generate graphs describing the robot's motion. The recorded data include the robot's trajectory, travelled distance, speed, acceleration, orientation angle, and longitudinal force. These parameters are analysed to evaluate the robot's motion during the simulation and the consistency of the developed model. Particular attention is given to the relationship between the robot's motion, changes in terrain configuration, and the analysed kinematic and dynamic parameters.

## *Robot trajectory*

**Error! Reference source not found.** and **Error! Reference source not found.** present the 3D trajectory of the robot and its trajectory in the XY plane, respectively. Based on **Error! Reference source not found.**, changes in height with respect to the coordinate system can be observed, corresponding to the uphill and downhill sections that the robot traversed during the simulation. **Error! Reference source not found.** enables a more straightforward analysis of changes in the direction of motion. The diagrams clearly show the straight-line sections of the trajectory, as well as the points at which changes in the direction of motion occurred.

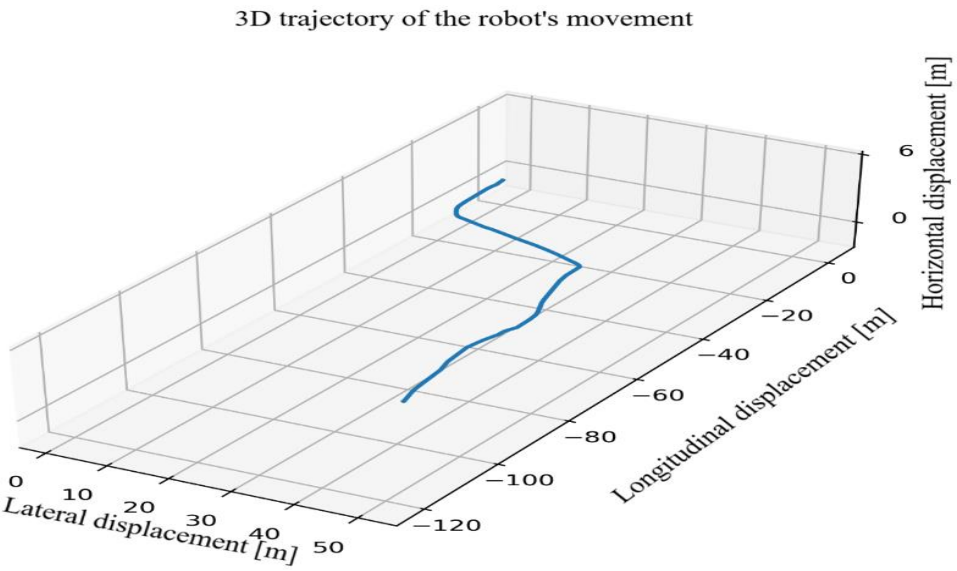



***Figure 7*** 3D trajectory of the robot's movement

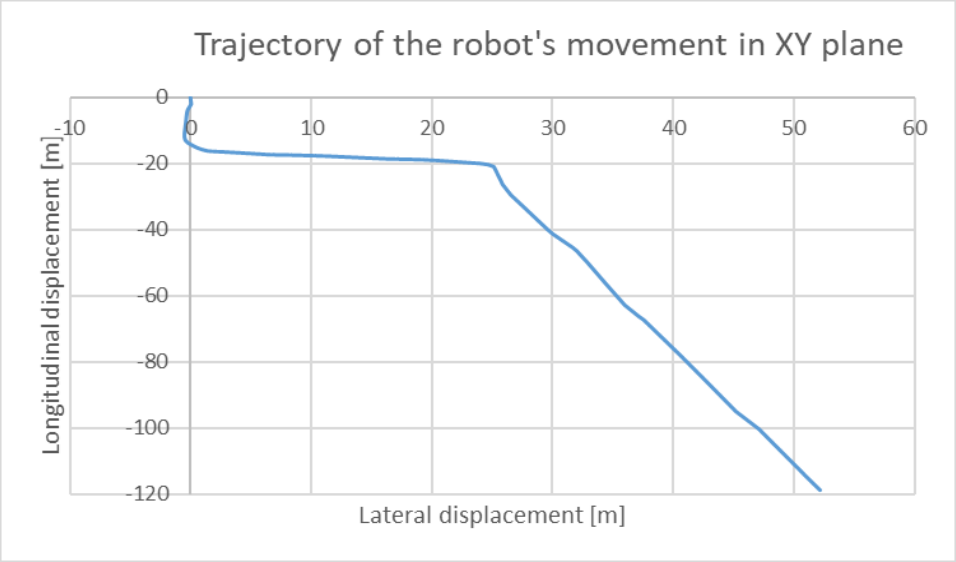



***Figure 8*** Trajectory of the robot's movement in XY plane

## *Travelled distance*

Figure 9 shows the total travelled distance as a function of time. The diagram shows that the robot travelled approximately 143 m during 97 s of simulation. The average speed of the robot can be determined by dividing the total travelled distance by the simulation time. The obtained average speed was 1.47 m/s compared with the set speed of 1.50 m/s, corresponding to a relative deviation of approximately 2.0%. The diagram also shows that the robot moved continuously throughout the simulation, without stopping.

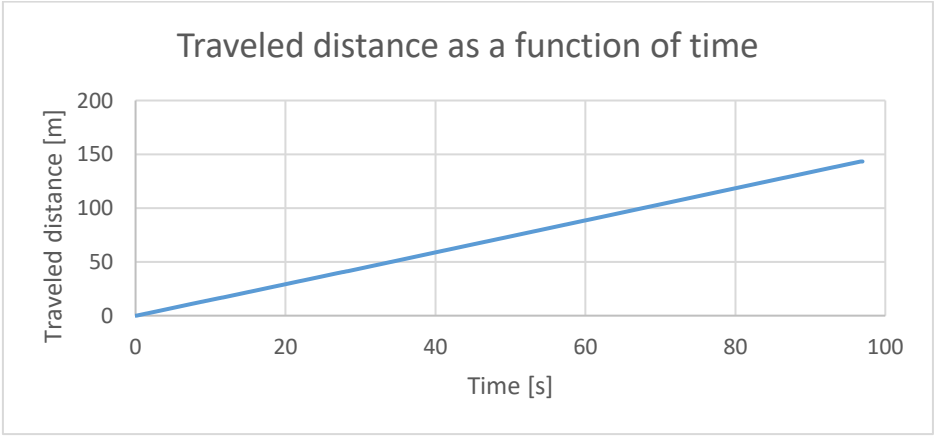



***Figure 9*** Travelled distance as a function of time

## *Robot speed and acceleration*

The analysis of the robot's speed and acceleration is one of the most important aspects of the motion analysis, as these parameters enable the evaluation of motion dynamics, control stability, and the robot's response to changes in terrain configuration.

**Error! Reference source not found.** shows that the robot maintained an almost constant speed throughout the simulation, with small oscillations. These oscillations may occur during changes in the direction of motion, when encountering slopes or terrain irregularities, but they can also be significantly attributed to the numerical method used to calculate the speed. The speed was calculated based on the distance travelled during each simulation step. The simulation time step was 64 ms, which is a very small-time interval. Since the robot does not travel exactly the same distance during every simulation step, variations in the calculated speed occur, including occasional decreases to values below 1 m/s. These variations do not represent an error, but rather reflect the numerical method used for calculating the speed.

On the other hand, **Error! Reference source not found.** shows the robot's acceleration during the simulation. It can be observed that the acceleration is almost constantly equal to zero, which is consistent with the nearly constant speed of the robot. Oscillations also occur for the same reason as in the speed diagram. Positive acceleration values indicate an increase in speed, while negative values indicate a decrease in speed. From Table 3 it can be seen that maximum acceleration and deceleration have almost exact values, which indicates a nearly symmetrical response of the robot during acceleration and deceleration.

***Table 3*** *Acceleration and deceleration maximum values*

| | |
|---|---|
| Maximum acceleration [m/s$^2$] | 8.133 |
| Maximum deceleration [m/s$^2$] | 8.066 |

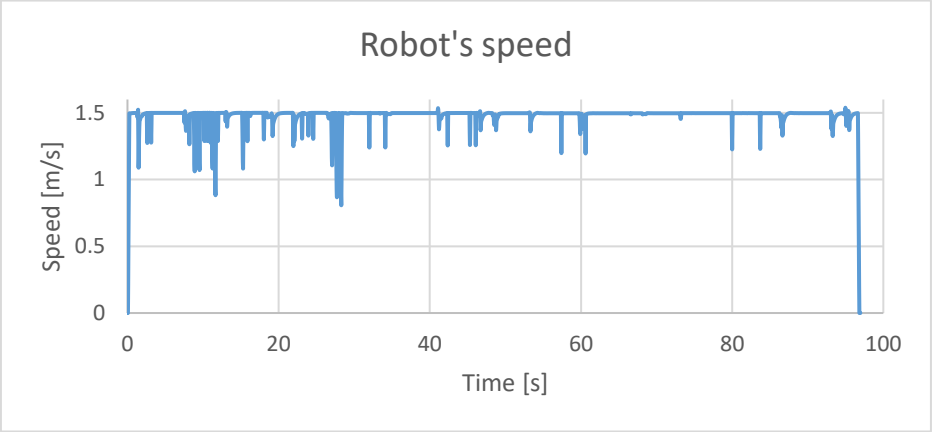



***Figure 10*** Robot speed during the simulation

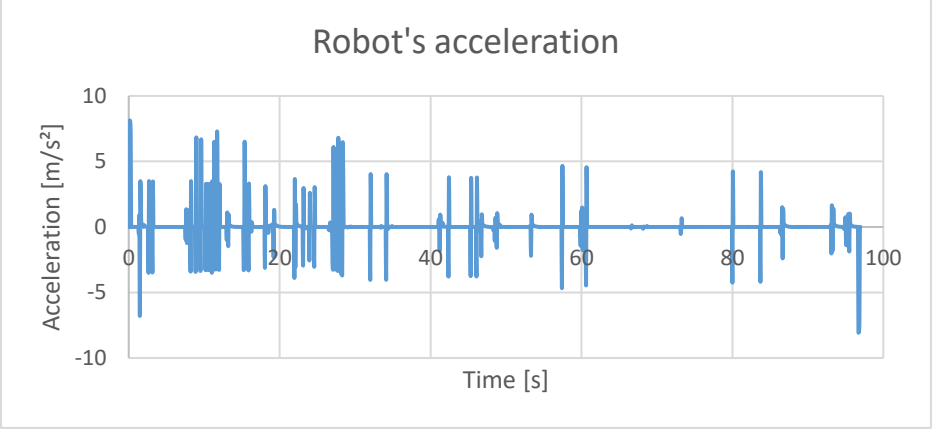



***Figure 11*** Robot acceleration during the simulation

## *Orientation angle*

Figure 12 shows the variation of the robot's orientation angle during the simulation. It enables the analysis of changes in the robot's direction of motion, identification of straight-line motion and turns, as well as verification of the robot's motion trajectory.

Analysis of the diagram confirms that the robot did not maintain a constant direction of motion, but performed multiple changes in direction. Rapid changes in the orientation angle indicate changes in the robot's direction of motion, while approximately constant angle values correspond to periods of straight-line motion.

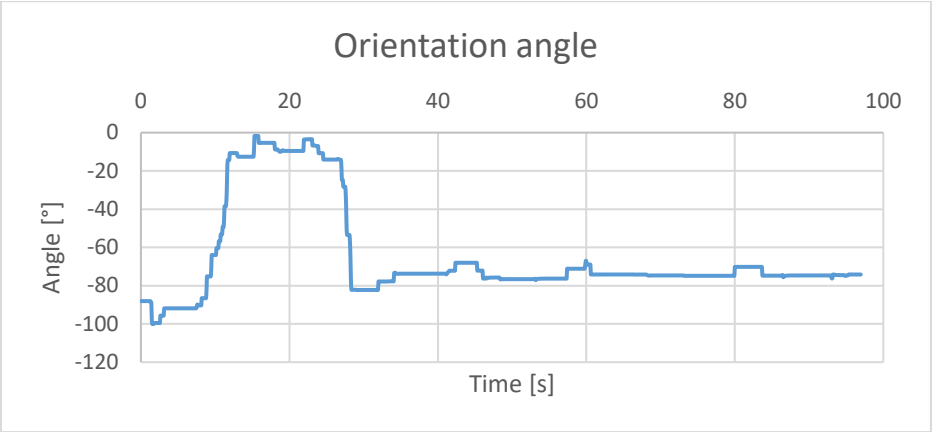



***Figure 12*** Orientation angle of the robot during the simulation

## *Longitudinal driving force of the robot*

Longitudinal force represents one of the most important dynamic parameters of robot motion, as it directly describes the force required to overcome rolling resistance, terrain inclination, and changes in speed.

The total longitudinal force acting on the robot was calculated as follows:

$$F = F_a + F_g + F_k \tag{5}$$

where:
$F_a$: Acceleration force;

$$F_a = ma \tag{6}$$

$m$: Robot mass;
$a$: Acceleration;
$F_g$: Slope force;

$$F_g = mg \sin \alpha \tag{7}$$

$g$: Acceleration due to gravity;
$sin\alpha$: Sine of the angle of inclination;
$F_k$: Rolling resistance force;

$$F_k = \mu mg \tag{8}$$

$\mu$: Rolling resistance coefficient.

Accordingly, Figure 13 does not represent the directly measured traction force of the motor or the force acting on the tracks, but rather the resultant longitudinal force that the robot needs to overcome in order to maintain its motion.

As can be observed from Figure 13 the longitudinal force remains almost constant throughout most of the simulation. This corresponds to the nearly constant speed of the robot, for which the acceleration force is approximately equal to zero. In this case, the main components contributing to the longitudinal force are the slope force and rolling resistance force. The rolling resistance force remains constant because the parameters determining it, which can be seen in equation *(8)*, remain unchanged throughout the simulation. Therefore, its variation with changes in the normal force due to terrain inclination was not explicitly considered. This represents a simplifying assumption of the adopted longitudinal force model.

At certain points, the total longitudinal force takes negative values. This can be explained by the robot decelerating or moving downhill, in which case the component of slope force acts in the direction of motion and reduces the force required to overcome rolling resistance. The oscillations in the total longitudinal force are caused by variations in the

acceleration force, which result from the numerical method used to calculate the speed and, consequently, the acceleration.

The analysis of the longitudinal force diagram shows that the robot operated in a stable motion regime during most of the simulation, with the required longitudinal force remaining approximately constant.

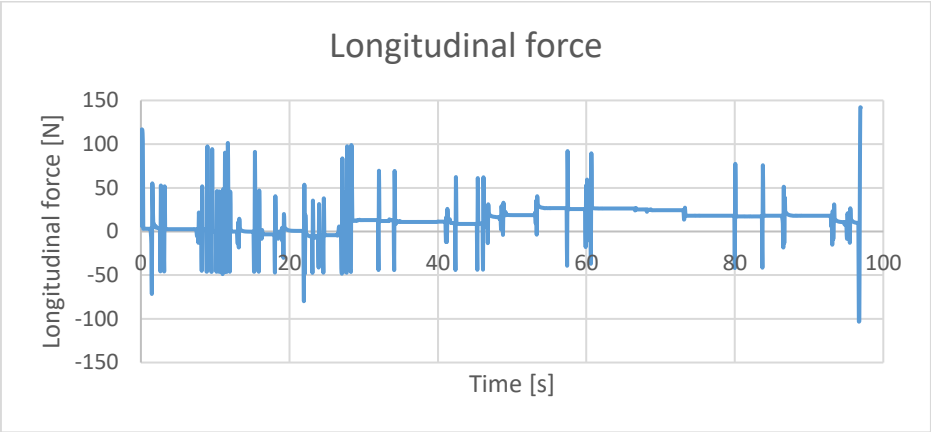



***Figure 13*** Longitudinal driving force of the robot

## CONCLUSIONS

Within this work, the modelling of a lightweight tracked robot, the development of the simulation environment, and the collection and analysis of simulation data were successfully performed. The developed model enabled the analysis of the robot's trajectory, speed, acceleration, orientation angle, and longitudinal force, providing an overall insight into its motion under the considered simulation conditions.

The robot travelled approximately 143 m during 97 s of simulation, with an average speed of 1.47 m/s compared with the set speed of 1.50 m/s. The relative deviation of approximately 2.0% indicates good agreement between the prescribed and achieved motion. In addition, the analysed changes in trajectory, orientation angle, acceleration, and longitudinal force were mutually consistent and logically interconnected, supporting the consistency of the developed simulation model. The similar magnitudes of the maximum acceleration and deceleration values also indicate a nearly symmetrical dynamic response of the robot during changes in speed.

Based on the obtained results, it can be concluded that the developed model successfully reproduces the considered robot motion regime and provides a suitable basis for analysing the dynamic behaviour of a tracked robot in a simulated environment. The simulation approach also enables investigation of robot motion on more complex terrains, rapid modification of simulation parameters, and testing of different motion scenarios and robot configurations.

The main limitations of the presented approach are the use of manual robot control, the analysis of a single simulation run, and the absence of experimental measurements from the real Jaguar robot for quantitative validation. Therefore, future work should focus on implementing automated trajectory generation and autonomous control, performing repeated simulations under different conditions, and comparing the simulation results with experimental measurements obtained from the real robot. Such an approach would enable a more comprehensive assessment of the accuracy and applicability of the developed simulation model.

## ACKNOWLEDGMENTS

This work was supported by the Ministry of Defense, Serbia, under the project „Modelling of Automotive Vehicle Assemblies for the Optimization of Dynamic Parameters”, No. VA-TT/1/26-28.

## REFERENCES

Bruzzone, L., Nodehi, S. E., & Fanghella, P. (2022). Tracked Locomotion Systems for Ground Mobile Robots: A Review. *Machines*, 10(8), 648. https://doi.org/10.3390/machines10080648

Cyberbotics Ltd. (n.d.). (2026, May 26) Webots documentation. Cyberbotics. https://cyberbotics.com/doc/guide/index

Gunjal, S. R. (2021). *Development of virtual simulation for evaluation of robotic assistive environment* [Master's thesis]. The University of Texas at Arlington.

Haidar, F. (2024). Integrated vehicle dynamics modeling, path tracking, and simulation: A MATLAB implementation approach. *Engineering Perspective*, 1(1), 7–16. https://doi.org/10.29228/eng.pers.75106

Hellström, T. (2011). Kinematics Equations for Differential Drive and Articulated Steering. Umea University. https://www8.cs.umu.se/kurser/5DV122/HT13/material/Hellstrom-ForwardKinematics.pdf

Kozłowski, K., & Pazderski, D. (2004). Modeling and control of a 4-wheel skid-steering mobile robot. *International Journal of Applied Mathematics and Computer Science*, 14(4), 477–496.

Muždeka, S., & Rakić, S. (2008). Konstrukcija borbenih vozila. Centar za vojnonaučnu dokumentaciju, informacije i bibliotekarstvo.

Rivera, Z. B., De Simone, M. C., & Guida, D. (2019). Unmanned ground vehicle modelling in gazebo/ros-based environments. *Machines*, 7(2), 42. https://doi.org/10.3390/machines7020042

ROS 2 Control maintainers. (n.d.). Wheeled Mobile Robot Kinematics. ROS 2 Control Documentation. Retrieved May 25, 2026, from https://control.ros.org/rolling/doc/ros2_controllers/doc/mobile_robot_kinematics.html