Reports in Mechanical Engineering
Vol. 6, No. 2, 2025, pp. 80-94
ISSN: 2683-5894, DOI: https://doi.org/10.31181/rme510

Development of the Design and Control System Programming for a Wheeled Mobile Robot

Bolatova A.B [1], Samenov G.K [1*], Balabekova K.G [1*], Zabiyeva A.B[1*], Gordey K.S [1], Makazhanov E.J [2], Bermukhambetov, V. A [2]
[1] L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
[2] IGD Kazakhstan LLP, Astana, Kazakhstan

Article Info

Article history:

Received December 19, 2025
Revised January 12, 2026
Accepted January 21, 2026

Keywords:
Wheeled Mobile Robot,
Control Unit,
Program Code,
Programming,
Sensors,
Sensor Systems,
Autonomous Robot

ABSTRACT

The developed wheeled mobile robot is equipped with laser devices and multiple sensors, enabling autonomous exploration and data collection in hard-to-reach environments without operator intervention. The main objective of this study is to investigate the design features of wheeled robots integrated with an program code - based control unit, including their ability to recognize obstacles, assess environmental conditions, analyze situations, and make decisions autonomously. The research involved three stages: the design and development of a physical prototype, programming the control system in C++, and planned field testing under real-world conditions. A comprehensive review of scientific literature was conducted to justify the robot's electric-driven wheeled design, followed by optimization of its mechanical and control architecture. Sensor systems were integrated with program code to enable adaptive responses to varying terrains and changing center of gravity during movement. Differential wheel speed distribution was calculated and simulated for complex geographical conditions, ensuring stable and efficient mobility. The program code control program processes diverse environmental data, including temperature, soil and water samples, gas concentrations, radiation levels, terrain inclination, vibrations, and seismic activity. The developed system demonstrates a significant advancement in autonomous robotic exploration, offering a reliable platform for research and monitoring in inaccessible or hazardous environments.

Corresponding Author:

Samenov G.K.
L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
Email: samenov_gk@enu.kz
Balabekova K.G
L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
Email: balabekova_kg@enu.kz
Zabiyeva A.B
L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
Email: zabiyeva_ab@enu.kz

1. Introduction

The use of vehicles and robots in complex geo-mining environments, such as underground workings, pipelines, and industrial debris, is crucial for ensuring worker safety by monitoring parameters including gas levels, humidity, water level and acidity, vibrations, and radiation. Autonomous robots are particularly valuable in challenging terrains, such as mountainous regions with unstable relief, variable geological features, and hard-to-access areas.

The primary objective of this study is to investigate the design features of wheeled robots equipped with an program code - based control unit, focusing on their ability to detect obstacles, respond to external influences, analyze situations, and make autonomous decisions. Based on the collected data, the study aims to develop and test an electric-driven wheeled mobile robot with an program code control system.

This scientific project focuses on three main directions:

1. The design and scientific justification of a wheeled mobile robot with electric drive, suitable for mountainous

regions with unstable terrain and variable geological conditions.

2. The development of a control unit integrated with an program code system using C++ and Python programming on an Arduino microcontroller.
3. The investigation of sensor systems and their integration with AI to enable autonomous decision-making in areas where human presence is limited or impossible.

Special attention is given to programming the program code control unit, which requires highly efficient algorithms to maintain balance both in static positions and during movement, compensating for changes in the center of gravity. The system is also designed to process environmental data, including temperature, water and soil samples, gas concentration, radiation levels, terrain inclination, vibrations, and seismic activity.

2. Literature Review

Previous studies have analyzed kinematic and dynamic models of mobile robots, focusing on the relationships between translational and angular velocities (Kozlowski & Pazderski, 2006; Maalouf et al., 2006). However, many works neglected wheel or track slippage, which significantly affects motion accuracy and control. Accurate modeling of slippage is essential to correlate wheel angular velocity with the actual linear displacement of the platform.

Experimental kinematic models of tracked and wheeled robots have been developed considering the positions of instantaneous centers of rotation (ICR) and their variation with speed and trajectory curvature (Mandow et al., 2007; Moosavian & Kalantari, 2008). Dynamic models for steering systems, including nonholonomic constraints and Coulomb friction-based contact modeling, have also been explored, with nonlinear feedback controllers used for trajectory tracking(Kozłowski & Pazderski, 2004; Yi, Song, et al., 2007). Slip-based control schemes, such as differential drive, are widely applied to wheeled and tracked robots, providing maneuverability and terrain adaptability, though they introduce challenges for energy efficiency, control complexity, and wheel wear (Anousaki & Kyriakopoulos, 2004; Tulekov et al., 2023; Yi et al., 2009; Yi, Zhang, et al., 2007).

Recent approaches integrated slip effects and steering dynamics for planar and spatial motion analysis, considering tangential stress as a function of shear deformation rather than simple friction models (Anousaki & Kyriakopoulos, 2007; Martínez et al., 2005; Yu et al., 2009). Experimental methods, including extended Kalman filters and laser scanning, have been applied to estimate slippage parameters and validate kinematic models (Caracciolo et al., 1999; Martínez et al., 2006; Wong, 2001; Yu et al., 2010). These studies revealed that actual ICR positions vary with speed and trajectory curvature, highlighting the limitations of idealized models and the need for adaptive control strategies (Le et al., 1997; Wong & Chiang, 2001).

Despite these advances, existing models are often computationally intensive or assume idealized conditions, limiting their applicability for real-time autonomous navigation in complex, unstructured environments. This underscores the necessity for an program code -integrated control system capable of processing sensor data and adapting to variable terrain, which is the focus of the present study.

3. Materials and Methods

The research was conducted in several stages, including: (1) scientific justification of the wheeled mobile robot design with an electric drive; (2) development of the optimized mechanical structure; (3) investigation of sensor systems and their integration with with program code; and (4) development of an program code C++ control unit.

The study involves the evaluation, selection, and application of methods and tools for industrial testing and modeling of optimal wheeled mobile robot designs. The ultimate goal is to develop a wheeled mobile robot with AI capabilities, examining its structural features and assessing its ability to operate in challenging environments, such as mountainous regions with unstable terrain, variable geological characteristics, and hard-to-access areas. Based on the obtained results, the design will undergo experimental testing under real-world conditions.

A key aspect of the study is the differential wheel speed distribution, a motion control method that regulates the speed of individual wheels to perform turns and maneuvers. This approach is commonly used in differential-drive robots (e.g., two-wheeled platforms) and four-wheeled robots with independently driven wheels (skid-steering).

4. Calculations and Modeling

The objective of this section is to develop a comprehensive mathematical and computational model of a wheeled mobile robot. Accurate calculations are required to ensure proper mechanical design, reliable motion control, and prediction of the robot's behavior under various operating conditions. Mathematical modeling enables simulation of robot motion, parameter optimization, and reduction of costs associated with physical prototyping.

The calculations presented in this section aim to:

- establish the relationship between wheel rotation and robot motion;
- analyze the influence of wheel geometry and configuration on speed, maneuverability, and stability;
- predict the dynamic response of the robot to control inputs for trajectory tracking and path planning;
- provide a basis for validating the robot design in simulation environments prior to real-world implementation.

4.1 Kinematic Modeling

The motion of a four-wheeled mobile robot is primarily determined by the rotation of its wheels. Let ω_i denote the angular velocity of the *i*-th wheel ($i = 1, \dots, 4$), and R the effective wheel radius.

The longitudinal component of the velocity of the *i*-th wheel in its local coordinate frame is given by

$$v_{ix} = R\,\omega_i, i = 1,2,3,4.$$

This relation forms the basis of the robot kinematic model. It allows the determination of the robot's translational velocity

$$\mathrm{v} = [v_x, v_y]^T$$

and angular velocity about the vertical axis $\omega_z$in the global coordinate system:

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = J(\theta) \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix},$$

where $J(\theta)$is the Jacobian matrix depending on the wheel arrangement and the robot orientation angle θ.

Modeling velocities in both local and global coordinate frames is essential for the synthesis of motion control algorithms, including trajectory tracking and obstacle avoidance. Without this step, motion planning algorithms cannot accurately predict the robot's path.

4.2 Dynamic Modeling

In addition to kinematics, dynamic modeling accounts for forces and torques acting on the robot. The motion of the platform is influenced by wheel-ground friction, contact forces, and inertial effects.

Using the Newton–Euler formulation, the resultant force F and torque τacting on the robot are expressed as:

$$F = M\,\dot{v}, \tau = I_z\,\dot{\omega}_z,$$

where

M is the robot mass (kg),

I_z is the moment of inertia about the vertical axis (kg·m²),

$\dot{v}$ is the linear acceleration (m/s²), and

$\dot{\omega}_z$ is the angular acceleration (rad/s²).

Dynamic modeling enables estimation of the required motor torque and stability margins, ensuring that the selected actuators can operate under real-world conditions without excessive slip or loss of traction.

4.3 Differential-Drive Velocity Relationships

Consider a four-wheeled robot with the following parameters:

- wheel radius $R = 0.075$ m,
- wheelbase $C = 0.24$ m.

The linear velocity $v$and angular velocity $\omega$of the robot are related to the angular velocities of the left $\omega_L$and right $\omega_R$wheels by:

$$v = \frac{R(\omega_R + \omega_L)}{2},$$

$$\omega = \frac{R(\omega_R - \omega_L)}{C}.$$

For

$\omega_R = 12$ rad/s and

$\omega_L = 10$ rad/s,
the resulting velocities are:

$$v = 0.825 \text{ m/s}, \omega = 0.625 \text{ rad/s}.$$

The turning radius of the robot is defined as:

$$R_t = \frac{v}{\omega} = 1.32 \text{ m}.$$

4.4 Actuator Dynamics and Control

The dynamics of the DC motor driving the wheels are described by:

$$L\frac{di}{dt} + Ri + K_e\omega = U,$$

$$J\frac{d\omega}{dt} + B\omega = K_t i,$$

where
i is the armature current (A),
U is the applied voltage (V),
L is the armature inductance (H),
R is the armature resistance (Ω),
K_e and K_t are the back-EMF and torque constants,
J is the rotor inertia (kg·m²), and B is the viscous friction coefficient.
For numerical simulation, the discrete-time formulation is used:

$$i_{k+1} = i_k + \frac{U - Ri_k - K_e\omega_k}{L}\Delta t,$$

$$\omega_{k+1} = \omega_k + \frac{K_t i_k - B\omega_k}{J}\Delta t.$$

A PID controller is employed for speed regulation:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt},$$

where the control error is defined as:

$$e(t) = \omega_{\text{target}} - \omega(t).$$

With controller gains
$K_p = 0.5$,
$K_i = 0.1$,
$K_d = 0.05$,
simulation results indicate that the motor speed converges to the reference value within 0.2 s.

4.5 Trajectory Modeling

For a robot starting from the origin $(0, 0)$and moving along a circular trajectory, the curvature of the path is defined as:

$$k = \frac{\omega}{v}.$$

The radius of the circular trajectory is therefore:

$$R = \frac{v}{\omega}.$$

The robot position in the global coordinate frame (X_g, Y_g) evolves according to:

$$X(t) = R\sin(\omega t),$$

$$Y(t) = R(1 - \cos(\omega t)),$$

with initial conditions:

$$X(0) = 0, Y(0) = 0, \theta(0) = 0.$$

The heading angle of the robot is given by:

$$\theta(t) = \omega t.$$

4.6 Assumptions and Kinematic Structure

The kinematic model incorporates differential-drive principles and a steering-with-slip analogy. The following assumptions are adopted:

- the robot's center of mass coincides with the geometric center of the chassis;
- wheels located on the same side rotate at identical angular velocities;
- the robot moves on a rigid planar surface with all wheels maintaining continuous contact with the ground.

Fig. 1 illustrates the kinematic model of the four-wheeled mobile robot, showing the relationship between individual wheel angular velocities, turning radius, and platform motion. This model forms the basis for programming the program code control unit and for simulating robot motion under various environmental conditions.

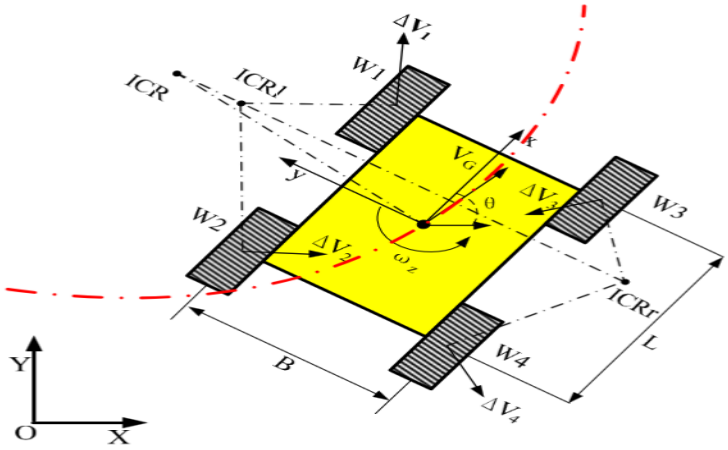

Figure 1: Kinematic Model of the Four-Wheeled Mobile Robot Illustrating the Relationship between Individual Wheel Angular Velocities, Turning Radius, and Resulting Translational and Rotational Motion of the Platform

4.7 Differential Wheel Speed Control of the Mobile Robot

Differential wheel speed control is a commonly used motion control strategy for wheeled mobile robots. This method is based on regulating the angular velocities of individual wheels in order to generate translational motion, turning maneuvers, and changes in trajectory. Differential wheel speed control is widely applied in classical differential-drive robots, such as two-wheeled platforms, as well as in four-wheeled robots with independently driven wheels, including skid-steered systems.

In four-wheeled skid-steered configurations, steering is achieved by imposing different angular velocities on the left and right wheel pairs. Straight-line motion occurs when all wheels rotate at equal speeds, while turning is produced by creating a speed difference between the wheel pairs on opposite sides of the robot. This approach eliminates the need for mechanical steering mechanisms and provides a compact and robust solution for mobile robot navigation.

4.8 Kinematic Scheme of Steering with Slip and Differential Drive

As shown in Fig. 2, the kinematic scheme is based on differential wheel speed control with steering-slip assumptions. The following assumptions are adopted for the development of the kinematic model:

- the robot's center of mass coincides with the geometric center of the chassis;
- wheels located on the same side of the robot rotate at identical angular velocities;
- the robot moves on a rigid planar surface, with all four wheels maintaining continuous contact with the ground.

Under these assumptions, the kinematic scheme establishes the relationship between wheel angular velocities,

turning radius, and overall robot motion. The model serves as the basis for simulating the robot's motion dynamics and for implementing the program code control unit. It enables accurate prediction of robot behavior during turning maneuvers, trajectory changes, and interactions with complex or uneven terrain.

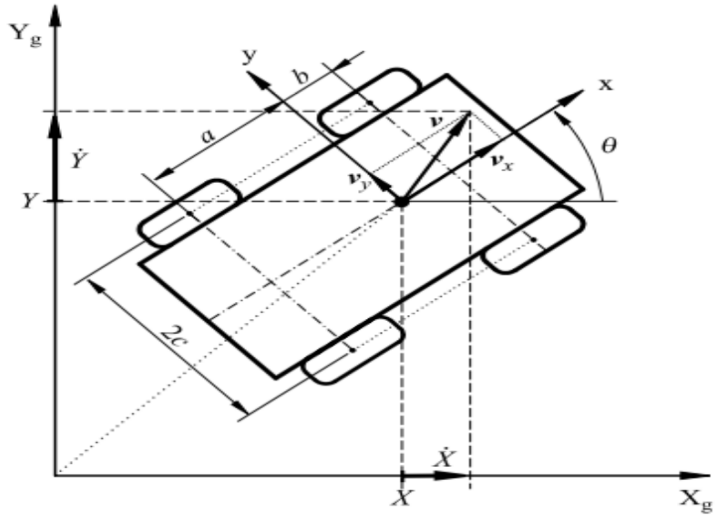

Figure 2: Kinematic Scheme of a Four-Wheeled Skid-Steered Mobile Robot Showing the Assumed Coordinate Systems, Wheel Arrangement, and motion constraints Used in the Kinematic Analysis

4.9 Kinematic Model of a Skid-Steered Mobile Robot

The skid-steered robot is assumed to move on a flat planar surface. A global coordinate system

$$I(X_g, Y_g, Z_g)$$

and a body-fixed local coordinate system $B(x, y, z)$, attached to the robot's center of mass, are defined. The position of the center of mass is described by (X, Y, Z), where $Z = \text{const}$.

The linear velocity in the local frame is

$$\mathrm{v} = (v_x, v_y, 0)^T,$$

and the angular velocity vector is

$$\omega = (0,0,\omega)^T.$$

The generalized coordinates are defined as

$$\mathrm{q} = (X, Y, \Theta)^T,$$

with the corresponding generalized velocities

$$\dot{\mathrm{q}} = (\dot{X}, \dot{Y}, \dot{\Theta})^T.$$

The transformation between local and global velocities is given by:

$$\begin{bmatrix}\dot{X}\\ \dot{Y}\end{bmatrix} = \begin{bmatrix}\cos\Theta & -\sin\Theta\\ \sin\Theta & \cos\Theta\end{bmatrix}\begin{bmatrix}v_x\\ v_y\end{bmatrix}.$$

Since the motion is planar, it follows that $\dot{\Theta} = \omega$.

Let each wheel rotate with angular velocity $\omega_i(t)$, $i = 1, \ldots, 4$. Wheel thickness is neglected, and each wheel is assumed to contact the ground at a single point P_i.

In skid-steered robots, lateral wheel velocity components are generally nonzero, enabling changes in orientation through controlled lateral slip. In the present model, longitudinal wheel slip is neglected, yielding:

$$v_{ix} = R\,\omega_i$$

The geometric and physical parameters of the developed mobile robot used in the modeling are summarized in Table 1.

Table 1: Geometric and Physical Parameters of the Developed Four-Wheeled Mobile Robot used in Kinematic and Dynamic Modeling

Parameter	Symbol	Value
Robot mass (kg)	M	15
Robot width (m)	B	0.25
Robot length (m)	L	0.35
Wheelbase (m)	C	0.24
Wheel radius (m)	R	0.075
Wheel width (m)	B_w	0.05

Fig. 3 presents the preliminary three-dimensional CAD model of the robot chassis, illustrating the overall geometry and structural layout.

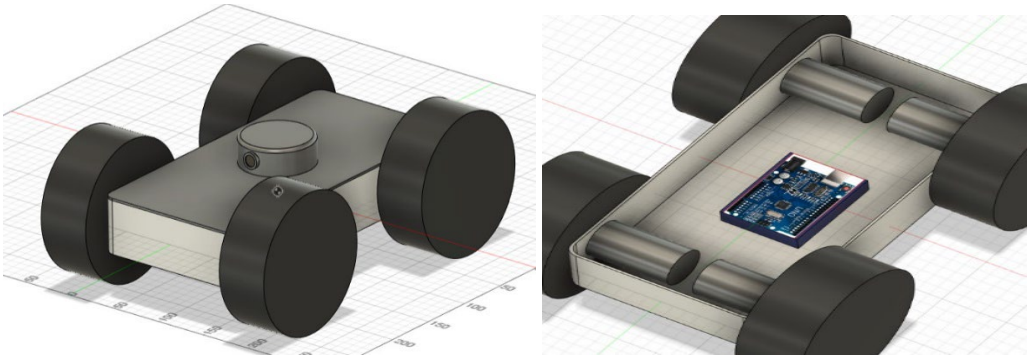

Figure 3: Preliminary Three-Dimensional CAD Model of the Robot Chassis Illustrating the Overall Geometry and Structural Layout

Fig. 4 shows the structural design of the mobile robot with the placement of the motor, gearbox, and control unit inside the chassis.

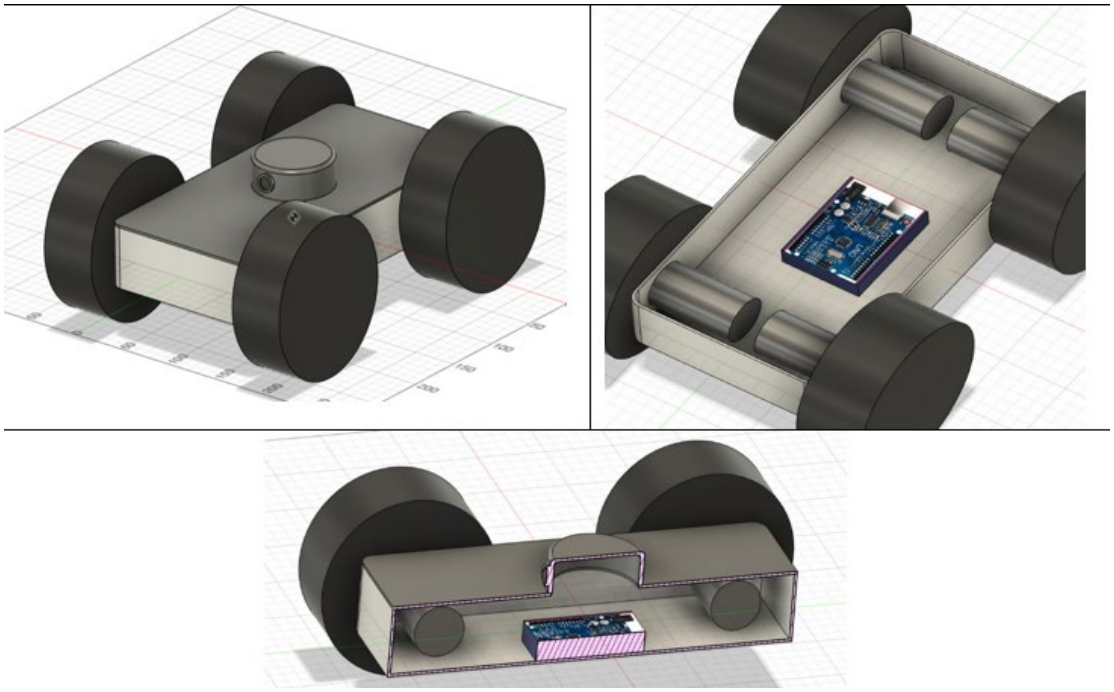

Figure 4: Structural Design of the Mobile Robot Showing the Placement of the Electric Motor, Gearbox, and Control Unit within the Chassis

Fig. 5 shows the external view of the assembled four-wheeled mobile robot prototype.

Figure 5: External View of the Assembled Four-Wheeled Mobile Robot Prototype

The electrical circuit for connecting two DC motors to an Arduino Uno is based on the use of an external motor driver, which provides proper interfacing between the low-voltage control signals of the microcontroller and the higher current load required by the motors. The Arduino sends logical control signals to the driver inputs—typically PWM signals for speed regulation and digital signals for direction control—while the motor power supply is delivered to a dedicated power input of the driver from an external source.

Each motor is connected to its corresponding pair of output terminals on the driver, and the ground line of the power supply must be connected to the Arduino's ground to ensure correct signal referencing. Structurally, the circuit consists of three key subsystems: the control loop (Arduino → driver inputs), the power loop (external power supply → driver → motors), and the common reference loop (GND), which interconnects all components to maintain stable system operation.

4.10 Preliminary Performance Evaluation of the Mobile Robot

To strengthen the experimental validation of the proposed kinematic and dynamic models, a set of preliminary performance evaluations was conducted in simulation and hardware-in-the-loop conditions. The objective of these experiments was not full-scale benchmarking, but rather an initial verification of the robot's response to speed commands, turning behavior, and obstacle avoidance capabilities.

4.11 Speed Response Analysis

The speed response of the mobile robot was evaluated using the DC motor model implemented in Octave and the Arduino-based PID controller. A step input corresponding to the target angular velocity was applied. As shown in Figures 9 - 12, the motor speed converges smoothly to the desired value within approximately 0.2 s, with no oscillatory behavior observed.

This response time confirms the adequacy of the selected PID gains ($K_p = 0.5$, $K_i = 0.1$, $K_d = 0.05$) for basic speed regulation tasks and demonstrates that the control system can ensure stable longitudinal motion under nominal operating conditions.

4.12 Turning Behavior and Maneuverability

Turning performance was analyzed using differential wheel speed control. By applying distinct angular velocities to the left and right wheel sets ($\omega_L = 10$ rad/s, $\omega_R = 12$ rad/s), the robot followed a circular trajectory with a calculated turning radius of $R_t = 1.32$ m.

The simulated trajectory closely matches the analytical kinematic model defined by

$$R = \frac{v}{\omega},$$

confirming the consistency between theoretical predictions and simulated motion. The smooth evolution of the heading angle $\theta(t) = \omega t$ indicates stable yaw dynamics without abrupt deviations, which is essential for trajectory tracking and path planning.

4.13 Preliminary Obstacle Avoidance Demonstration

Basic obstacle avoidance behavior was demonstrated in simulation using sensor feedback and differential speed

adjustment. When an obstacle was detected in the robot's forward path, wheel speeds were asymmetrically modified, inducing a controlled turn that allowed the robot to bypass the obstacle and resume its original trajectory.

Although this experiment was performed at a preliminary level and without advanced environment mapping, it demonstrates the feasibility of integrating the proposed kinematic model with sensor-driven control logic. These results provide an initial proof-of-concept for more advanced AI-based navigation and autonomous decision-making algorithms.

Fig. 6 illustrates the electrical circuit diagram used to connect two DC motors to the Arduino Uno via a motor driver.

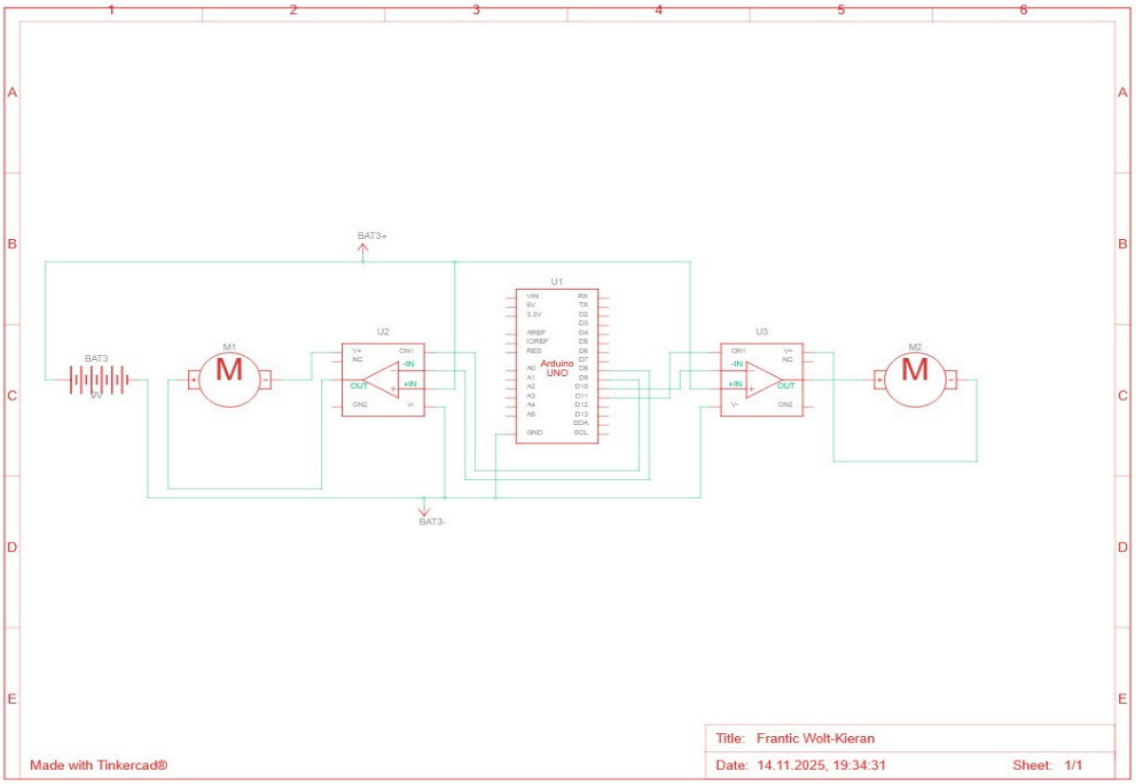

Figure 6: Electrical Circuit Diagram for Connecting two DC Motors to the Arduino Uno Microcontroller via a Motor Driver

The motor requires significant torque, especially considering that testing is planned under challenging geological conditions; therefore, it is connected through a motor driver (Duan et al., 2014). Before testing the motor, programming and simulation are performed in Tinkercad.

The steering motion of the mobile robot is controlled by a set of sensors that provide precise evaluation of both the operator's intended actions and the vehicle's actual behavior. The primary element is the steering wheel angle sensor, which measures the magnitude and rate of change of the wheel angle, enabling the electronic system to determine the desired direction of motion. This is complemented by wheel angle sensors installed in the steering mechanisms, which register the actual position of the steered wheels, a critical factor for active steering systems (Akhmetov et al., 2025).

Angular velocity sensors (gyroscopes) and lateral acceleration sensors play a key role in analyzing the robot's dynamics. Gyroscopes measure yaw rate, i.e., rotation around the vertical axis, allowing assessment of deviation from the intended trajectory. Lateral acceleration sensors detect side forces experienced during cornering, enabling evaluation of stability and identifying conditions that may lead to skidding or understeer. The combined data from these sensors form the basis of electronic stability control systems (ESP/ESC) (Balabekova et al., 2023).

Wheel speed sensors, used in ABS systems, monitor wheel slip during braking and maneuvering. Their readings are essential for the proper operation of anti-lock braking systems, traction control systems, and electronic brake-force distribution systems, which actively intervene in vehicle control during turns. Longitudinal acceleration sensors and body or suspension position sensors are also significant, as they allow for consideration of mass redistribution and

body roll under challenging driving conditions, directly affecting wheel-road adhesion.

Fig. 7 presents the connection and simulation diagram of the robot control system implemented in the Tinkercad environment.

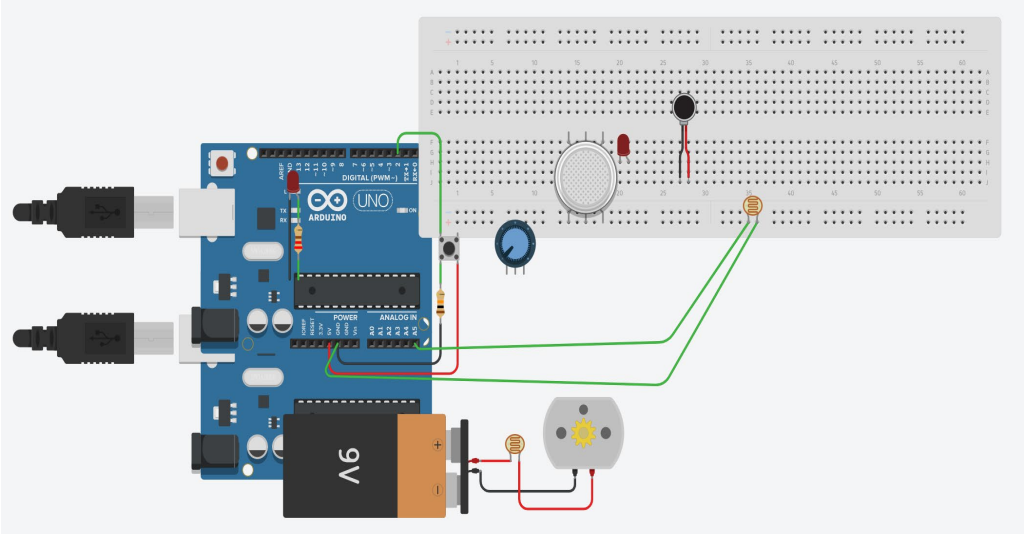

Figure 7: Connection and Simulation Diagram of the Mobile Robot Control System Implemented in the Tinkercad Environment

The next critical step in the design of the mobile robot is programming the control unit and integrating program code agents. A portion of the C++ code is presented, with comments describing the manipulation processes using the Arduino Uno.

Fig. 8 shows a fragment of the C++ program code implemented in the Arduino development environment for motor control.

Arduino Uno

sketch_jun27a.ino

```
  digitalWrite(dirPin, HIGH);   //Place dirPin to HIGH so we spin CW
  Serial.begin(9600);           //Start serial port
  sensor.begin();
  PCICR |= (1 << PCIE0);    //enable PCMSK0 scan so we can use interrupts
  PCMSK0 |= (1 << PCINT0);  //Set pin "D8" trigger an interrupt on "any" state change.
}



void loop() {
  if (loop_starts)           //We reset angle when the magnet is detected on D8
  {
    angle = 0;
    loop_starts = false;
  }

  digitalWrite(stepPin, HIGH);                  //Make one step
  delayMicroseconds(Value);                     //Small delay
  digitalWrite(stepPin, LOW);                   //Make another step
  delayMicroseconds(Value);                     //Add another delay

  VL53L0X_RangingMeasurementData_t measure;
  sensor.rangingTest(&measure, false); // pass in 'true' to get debug data printout!
  if (measure.RangeStatus != 4)
  {
    int r = measure.RangeMilliMeter;
    if (r > maxdist) //Limit the dsitance to maximum set distance above
      r = maxdist;
    Serial.print(angle);          //Print the values to serial port
    Serial.print(",");
    Serial.print(r);
    Serial.println(",");

    angle = angle + angle_step;   //Increase angle value by the angle/loop value set above (in this case 2.4ï¿½ each loop)
  }
  else
```

Figure 8: Fragment of the C++ Program Code for Motor Control Implemented in the Arduino Development Environment

C++ Programming for Arduino Motor Control

The Arduino Uno was programmed in C++ to control a DC motor via the TA6586 driver using a joystick. The connections are as follows:

Joystick: VRy (Y-axis) connected to A0, used to control motor speed and direction.

TA6586 Driver:

- IN1 connected to pin 9 (PWM-capable)
- IN2 connected to pin 10 (PWM-capable)
- OUT1 and OUT2 connected to the motor

VCC to 5V Arduino, GND to Arduino GND

Motor power supplied separately if required (TA6586 can operate at 3-14 V).

When the joystick is at the midpoint, analogRead returns approximately 512. Moving the joystick upward commands forward motion, while downward commands reverse motion. The motor speed is proportional to the deviation from the center position.

```
const int JOY_Y_PIN = A0; // Y-ось джойстика const int MOTOR_IN1 = 9; // IN1 TA6586 (PWM для forward) const int MOTOR_IN2 = 10; // IN2 TA6586 (PWM для reverse);
void setup () {
pinMode(MOTOR_IN1, OUTPUT);
pinMode(MOTOR_IN2, OUTPUT);
analogWrite(MOTOR_IN1, 0); // Initial stop
analogWrite(MOTOR_IN2, 0);
Serial.begin(9600); // Optional debugging
}
void loop () {
  int yValue = analogRead(JOY_Y_PIN); // Read joystick Y-axis (0-1023)
  // Map to -255 to 255 speed range
  int speed = map (yValue, 0, 1023, -255, 255);
  // Dead zone to prevent drift
  if (abs(speed) < 20) {
    speed = 0;
  }
if (speed > 0) {
// Forward: PWM on IN1, LOW on IN2
analogWrite(MOTOR_IN1, speed);
analogWrite(MOTOR_IN2, 0);
Serial.println("Forward: " + String(speed));
} else if (speed < 0) {
// Reverse: LOW on IN1, PWM on IN2
analogWrite(MOTOR_IN1, 0);
analogWrite(MOTOR_IN2, abs(speed));
Serial.println("Reverse: " + String(abs(speed)));
} else {
// Stop: LOW on both pins
analogWrite(MOTOR_IN1, 0);
analogWrite(MOTOR_IN2, 0);
Serial.println("Stop");
}
delay (100); // Delay for stability
```

5. Results and Discussion

The preliminary performance results confirm that the developed kinematic and dynamic models are suitable for predicting the robot's motion behavior at basic operational levels. The observed speed response, turning stability, and obstacle avoidance capability demonstrate that the platform can reliably execute fundamental maneuvers.

While these experiments are limited in scope, they establish a solid experimental foundation for future work, including real-world testing under complex terrain conditions, quantitative performance metrics, and full integration of AI-based navigation systems.

Table 2: Preliminary Performance Metrics of the Mobile Robot

Metric	Symbol/ Definition	Value	Evaluation method
Speed Response Time	t_r - time to reach 95% of target speed	0.2 s	Octave simulation with PID speed control
Steady-State Speed Error	$e_{ss} = \omega_{target} - \omega_{steady}$	< 3%	Motor dynamic model
Linear Velocity	$v = \frac{R(\omega_R + \omega_L)}{2}$	0.825 m/s	Analytical calculation
Angular Velocity	$\omega = \frac{R(\omega_R - \omega_L)}{C}$	0.625 rad/s	Analytical calculation
Turning Radius	$R_t = \frac{v}{\omega}$	1.32 m	Kinematic model and simulation
Yaw Stability	Absence of oscillations in $\omega(t)$	Stable	Time-domain response analysis
Slip-Induced Deviation	Lateral deviation during turning	Negligible (preliminary)	Skid-steering kinematic simulation
Obstacle Avoidance Capability	Successful trajectory deviation and recovery	Demonstrated	Sensor-triggered differential speed control

5.1 DC Motor Dynamic Model

A dynamic model of the DC motor was implemented in Octave to evaluate its performance. The motor parameters used in the simulation are summarized as follows:

The dynamic behavior of the DC motor I

$R = 2\ \Omega$, (armature resistance)

$L = 0.5$ H, (armature inductance)

$K_e = 0.02$V\cdotps\cdotprad^{-1} (back-EMF constant)

$K_t = 0.02$ N\cdotpm\cdotpA^{-1} (torque constant)

$J = 0.01$ kg\cdotpm^2 (rotor moment of inertia)

B=0.001N\cdotpm\ cdots (viscous friction coefficient)

$U = 2.5$ V, (applied voltage)

The dynamic behavior of the DC motor is described by the following system of differential equations, governing the armature current $i(t)$and the angular velocity $\omega(t)$:

$$\frac{di(t)}{dt} = \frac{U - Ri(t) - K_e\omega(t)}{L},$$

$$\frac{d\omega(t)}{dt} = \frac{K_t i(t) - B\omega(t)}{J}.$$

For numerical simulation, a discrete-time approximation was employed using a fixed time step

$$\Delta t = 0.01 \text{ s},$$

over a total simulation interval of 10 s. The iterative computation scheme is given by:

$$i_{k+1} = i_k + \frac{U - Ri_k - K_e\omega_k}{L}\Delta t,$$

$$\omega_{k+1} = \omega_k + \frac{K_t i_k - B\omega_k}{J}\Delta t,$$

where $i_k$and $\omega_k$denote the current and angular velocity at the k-th time step, respectively.

This formulation ensures consistency with the continuous-time dynamic model and enables efficient numerical integration using standard simulation tools such as Octave.

The results are presented in Fig. 9 and 12.

The simulation demonstrates the expected dynamic behavior of the motor, with a smooth rise in current followed by stabilization, and a corresponding increase in shaft speed to its steady-state value. These results validate the motor parameters and provide a reliable basis for further integration with the Arduino-based control system and joystick interface.

Fig. 9 illustrates the simulated transient response of the DC motor current as a function of time.

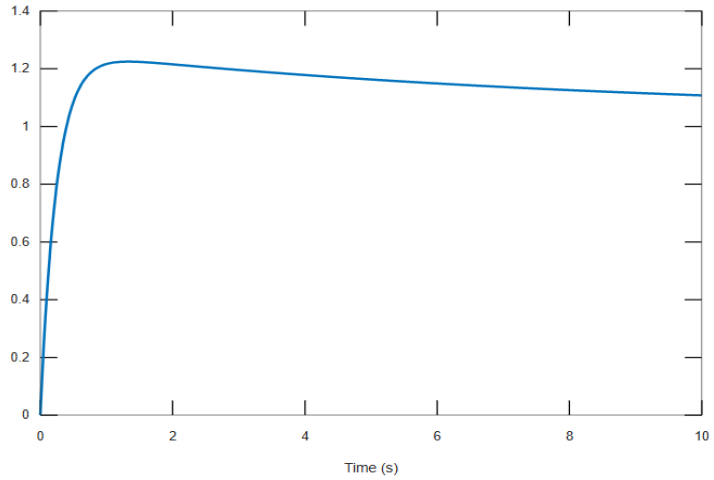

Figure 9: Simulated Transient Response of DC Motor Current as a Function of Time Obtained Using Octave

The mathematical model of the DC motor used for simulation is shown in Fig. 10. Fig. 11 presents the numerical simulation results of the motor current obtained from the implemented dynamic model.

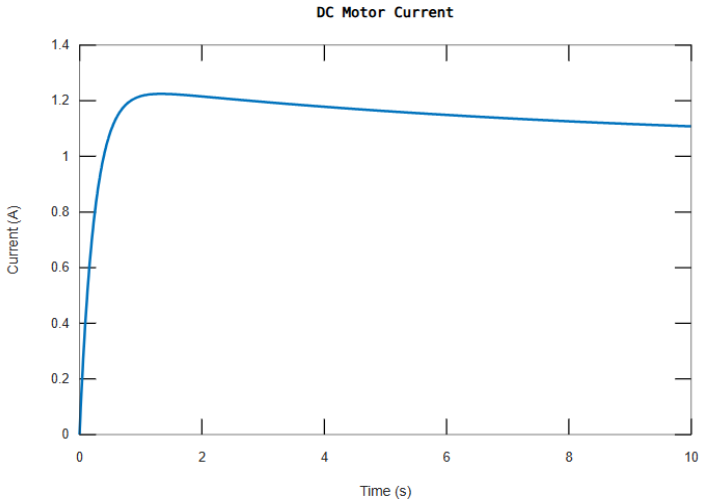

Figure 10: Mathematical Model of the DC Motor Used for Dynamic Simulation and Control Analysis

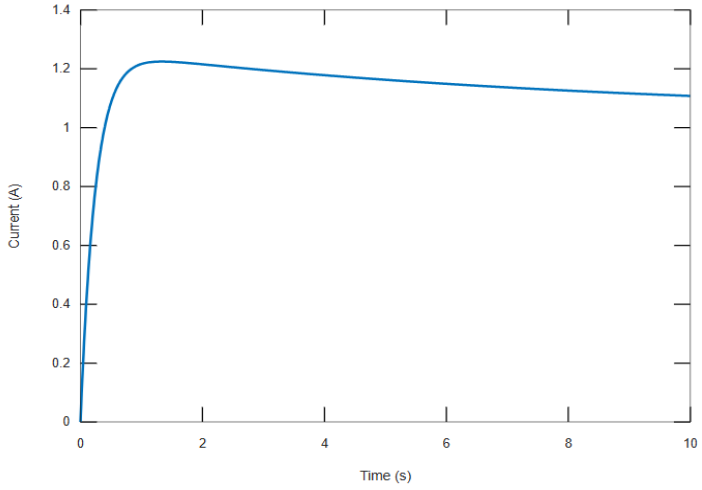

Figure 11: Time-Domain Response of the DC Motor Current Obtained from Numerical Simulation

Fig. 12 shows the simulated angular velocity of the DC motor shaft expressed as rotational speed over time.

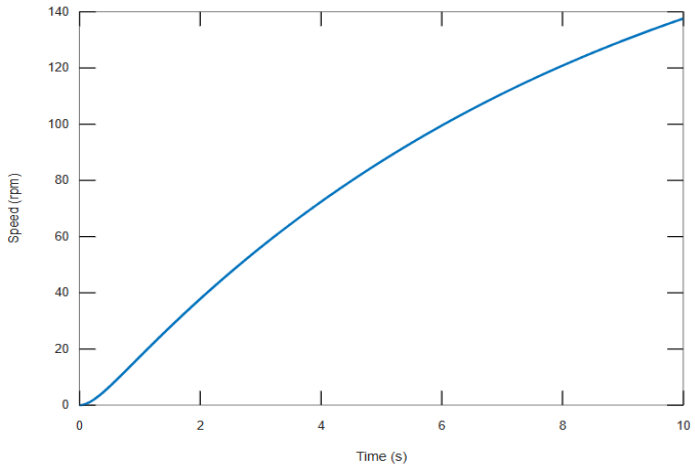

Figure 12: Simulated Angular Velocity of the DC Motor Shaft Expressed as Rotational Speed Over Time

6. Conclusions

A comprehensive review of the scientific literature on mobile robot design was conducted. The physical assembly of a four-wheeled mobile robot model was completed, and the motion and manipulation code based on a DC motor was programmed. Following the construction of the physical model, further research is planned, including 3D modeling of the electrically driven mobile robot in SolidWorks, simulations, state diagrams, and calculations for maintaining equilibrium in three-dimensional space under complex geological conditions.

The control unit has been programmed using Arduino, and preliminary testing of the integrated program code system has been initiated. Scientific research on sensor systems and their integration with program code was carried out to enable autonomous operation in areas without human presence, where the robot can make decisions independently.

Special attention is given to programming the program code-based control unit, which requires the development of a highly efficient algorithm capable of rapid and precise maintenance of equilibrium, both in static and dynamic states. This is particularly important due to shifts in the center of gravity during movement. The software also ensures recognition, selection, and processing of external data, including temperature, water and soil samples, gas concentration, radiation levels, slope, vibrations, and seismic activity.

Additionally, calculations were performed to determine the optimal wheel speed distribution, torque requirements, and dynamic responses under varying terrain conditions, ensuring stability and maneuverability of the mobile robot in complex environments. These calculations provide the theoretical foundation for designing a robust program code driven control system capable of operating safely and effectively in challenging geological conditions.

References

Akhmetov, K., Bolatova, A., Makazhanov, Y., & Samenov, G. (2025). Experimental study of brushless motors and MPU6050 using a PID controller. *Bulletin of L.N. Gumilyov Eurasian National University Technical Science and Technology Series*, *152*(3), 99–109. https://doi.org/10.32523/2616-7263-2025-152-3-99-109

Anousaki, G., & Kyriakopoulos, K. J. (2004). A dead-reckoning scheme for skid-steered vehicles in outdoor environments. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004* (Vol. 1, pp. 580–585). IEEE. https://doi.org/10.1109/ROBOT.2004.1307211

Anousaki, G. C., & Kyriakopoulos, K. J. (2007). Simultaneous localization and map building of skid-steered robots. *IEEE Robotics & Automation Magazine*, *14*(1), 79–89. https://doi.org/10.1109/MRA.2007.339625

Balabekova, K., Zabieva, A., & Orazalina, A. (2023). Research of railway gauge modification systems. *Bulletin of L.N. Gumilyov Eurasian National University Technical Science and Technology Series*, *144*(3), 143–153. https://doi.org/10.32523/2616-7263-2023-144-3-143-153

Caracciolo, L., De Luca, A., & Iannitti, S. (1999). Trajectory tracking control of a four-wheel differentially driven

mobile robot. In *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)* (Vol. 4, pp. 2632–2638). IEEE. https://doi.org/10.1109/ROBOT.1999.773994

Duan, Z., Cai, Z., & Min, H. (2014). Robust dead reckoning system for mobile robots based on particle filter and raw range scan. *Sensors*, *14*(9), 16532–16562. https://doi.org/10.3390/s140916532

Kozlowski, K., & Pazderski, D. (2006). Practical stabilization of a skid-steering mobile robot-A kinematic-based approach. In *2006 IEEE International Conference on Mechatronics* (pp. 519–524). IEEE. https://doi.org/10.1109/ICMECH.2006.252581

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.

Le, A. T., Rye, D. C., & Durrant-Whyte, H. F. (1997). Estimation of track-soil interactions for autonomous tracked vehicles. In *Proceedings of International conference on robotics and automation* (Vol. 2, pp. 1388–1393). IEEE. https://doi.org/10.1109/ROBOT.1997.614331

Maalouf, E., Saad, M., & Saliah, H. (2006). A higher level path tracking controller for a four-wheel differentially steered mobile robot. *Robotics and Autonomous Systems*, *54*(1), 23–33. https://doi.org/10.1016/j.robot.2005.10.001

Mandow, A., Martinez, J. L., Morales, J., Blanco, J. L., Garcia-Cerezo, A., & Gonzalez, J. (2007). Experimental kinematics for wheeled skid-steer mobile robots. In *2007 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1222–1227). IEEE. https://doi.org/10.1109/IROS.2007.4399139

Martínez, J. L., González, J., Morales, J., Mandow, A., & García-Cerezo, A. J. (2006). Mobile robot motion estimation by 2D scan matching with genetic and iterative closest point algorithms. *Journal of Field Robotics*, *23*(1), 21–34. https://doi.org/10.1002/rob.20104

Martínez, J. L., Mandow, A., Morales, J., Pedraza, S., & Garcia-Cerezo, A. (2005). Approximating kinematics for tracked mobile robots. *The International Journal of Robotics Research*, *24*(10), 867–878. https://doi.org/10.1177/0278364905058239

Moosavian, S. A. A., & Kalantari, A. (2008). Experimental slip estimation for exact kinematics modeling and control of a tracked mobile robot. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 95–100). IEEE. https://doi.org/10.1109/IROS.2008.4650798

Tulekov, A., Togizbayeva, B., Kenesbek, I., Kenesbek, A., & Zabiyeva, A. (2023). The Use of Composite Materials in the Production of Tower Cranes. https://doi.org/10.12700/APH.20.9.2023.9.16

Wong, J. (2001). *The theory of ground vehicles* (3rd ed.). New York : John Wiley.

Wong, J., & Chiang, C. (2001). A general theory for skid steering of tracked vehicles on firm ground. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *215*(3), 343–355. https://doi.org/10.1243/0954407011525683

Yi, J., Song, D., Zhang, J., & Goodwin, Z. (2007). Adaptive trajectory tracking control of skid-steered mobile robots. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (pp. 2605–2610). IEEE. https://doi.org/10.1109/ROBOT.2007.363858

Yi, J., Wang, H., Zhang, J., Song, D., Jayasuriya, S., & Liu, J. (2009). Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation. *IEEE transactions on robotics*, *25*(5), 1087–1097. https://doi.org/10.1109/TRO.2009.2026506

Yi, J., Zhang, J., Song, D., & Jayasuriya, S. (2007). IMU-based localization and slip estimation for skid-steered mobile robots. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2845–2850). IEEE. https://doi.org/10.1109/IROS.2007.4399477

Yu, W., Chuy, O., Collins, E. G., & Hollis, P. (2009). Dynamic modeling of a skid-steered wheeled vehicle with experimental verification. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4212–4219). IEEE. https://doi.org/10.1109/IROS.2009.5354381

Yu, W., Chuy, O. Y., Collins, E. G., & Hollis, P. (2010). Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle. *IEEE transactions on robotics*, *26*(2), 340–353. https://doi.org/10.1109/TRO.2010.2042540