AIN SHAMS UNIVERSITY-

FACULTY OF ENGINEERING

[MCT443s]

Design of Autonomous systems

Major Task: Furuta Pendulum

Team 9

|  |  |  |
| --- | --- | --- |
| Name | ID | Section |
| Musa Gamal El-Shenawy | 2100614 | 3 MCT |
| Mahmoud Salama Mohamed | 2100961 | 3 MCT |
| Abdelkarim Wael Abdelkarim | 2101733 | 3 MCT |
| Abdullah Mahmoud Abdelkarim | 2100658 | 3 MCT |
| Moaz Gamal ALsayed | 2101231 | 3 MCT |
| Muhannad Nasir El-Din Aly | 2001556 | 3 MCT |

GitHub Repository : <https://github.com/Musa2004-me/Mecanum-Wheel-Robot>

Submitted to: Dr. [Omar Mahmoud Mohamed Shehata](https://eng.asu.edu.eg/staff/omar.shehata)

Table of Contents

[Major Task 1](#_Toc217346947)

[Introduction 4](#_Toc217346948)

[Objectives 5](#_Toc217346949)

[Robot Design and Modeling 5](#_Toc217346950)

[Perception System Development 5](#_Toc217346951)

[System Integration and Validation 5](#_Toc217346952)

[Robot Platform Description (Mecanum Robot) 6](#_Toc217346953)

[Mecanum Wheels 7](#_Toc217346954)

[KINEMATIC 8](#_Toc217346955)

[Kinematic Model of the Mecanum Robot 8](#_Toc217346956)

[Justification for Using Mecanum Drive 8](#_Toc217346957)

[The configuration parameters and system velocities are defined as follows: 9](#_Toc217346958)

[The Relation between Motions and the Translation MATRIX 14](#_Toc217346959)

[FOUR MECANUM OMNIDIRECTIONAL SOLUTION 14](#_Toc217346960)

[CAD Design 17](#_Toc217346961)

[Project Package(mec) 19](#_Toc217346962)

[1) Model Folder Description 19](#_Toc217346963)

[2) Meshes Folder Description 19](#_Toc217346964)

[3) Worlds Folder Description 19](#_Toc217346965)

[4) Rviz Folder Description 19](#_Toc217346966)

[5) Config Folder Descreption 20](#_Toc217346967)

[6) New\_launch Folder Description 20](#_Toc217346968)

[7) Scripts Folder Description 21](#_Toc217346969)

[1- driver\_node.py 21](#_Toc217346970)

[2- lane\_detection\_node.py 21](#_Toc217346971)

[3- obstacle\_Detection\_Node.py 22](#_Toc217346972)

[4- mecanum\_motion.py 22](#_Toc217346973)

[5- imu.py 23](#_Toc217346974)

[23](#_Toc217346975)

[6- odom\_node.py 23](#_Toc217346976)

[7- scenario\_01.py 24](#_Toc217346977)

[8- scenario\_02.py 24](#_Toc217346978)

[REFERENCES 25](#_Toc217346979)

# Table of Figuers

[Figure 1 Motions of Omnidirectional platform 6](#_Toc217347232)

[Figure 2 Mecanum wheel design 7](#_Toc217347233)

[Figure 3 Robot motion according to the direction and angular speed of the wheels 7](#_Toc217347234)

[Figure 4 Wheels Configuration and Posture definition 9](#_Toc217347235)

[Figure 5 wheel 1 in the robot coordinate 9](#_Toc217347236)

[Figure 6 wheel 1 motion principle 9](#_Toc217347237)

[Figure 7 Robot CAD Model without Covering 17](#_Toc217347238)

[Figure 8 Robot Model with covering CAD Model 17](#_Toc217347239)

[Figure 9 Track 01CAD Model 18](#_Toc217347240)

[Figure 10 Track 02 CAD Model 18](#_Toc217347241)

[Figure 11 mec package folders 19](#_Toc217347242)

[Figure 12 urdf model 19](#_Toc217347243)

[Figure 13 High-Level Flow (Launch Sequence) 20](#_Toc217347244)

[Figure 14 flowchart of driver\_node 21](#_Toc217347245)

[Figure 15 flowchart of lane\_detection\_node 21](#_Toc217347246)

[Figure 16 flowchart ofobstacle\_detection\_node 22](#_Toc217347247)

[Figure 17 flowchart of mecanum\_motion 22](#_Toc217347248)

[Figure 18 flowchart of imu\_node 23](#_Toc217347249)

[Figure 19 flowchart of odom\_node 23](#_Toc217347250)

[Figure 20 flowchart of scenario\_01 24](#_Toc217347251)

[Figure 21 flowchart of scenario\_02 24](#_Toc217347252)

# Introduction

Autonomous wheeled mobile robots play a critical role in modern intelligent transportation systems, warehouse automation, and service robotics. A key capability of such robots is the ability to navigate structured environments while maintaining lane alignment and avoiding obstacles in a safe and efficient manner. These tasks require the integration of perception, control, and decision-making algorithms within a reliable robotic platform.

In this project, an autonomous mobile robot is designed and simulated using the Robot Operating System 2 (ROS 2 Humble) and the Gazebo Classic simulation environment. The robot is required to perform lane-keeping over a fixed distance and execute obstacle avoidance maneuvers involving lane transitions. While the project description targets a differential drive platform, this work adopts a Mecanum-wheel robot, which provides holonomic motion and enables independent control of forward, lateral, and rotational movements.

The use of a Mecanum drive introduces additional flexibility during navigation, particularly during lane changes and obstacle avoidance, where lateral motion can be achieved without large heading changes. The project focuses on developing perception algorithms using camera and range sensors, implementing feedback control strategies for lane keeping, and designing a decision-making logic to manage obstacle detection and lane transitions. All components are integrated and validated in simulation, demonstrating a complete autonomous navigation system.

# Objectives

The main objective of this project is to design, implement, and validate an autonomous Mecanum-wheel mobile robot capable of lane keeping and obstacle avoidance in a simulated environment, as follows:

## Robot Design and Modeling

* + Design a Mecanum-wheel mobile robot that fits within the provided track constraints.
  + Develop a kinematic model suitable for holonomic motion control.

## Perception System Development

* + Implement camera-based lane detection to estimate lane position and lateral error.
  + Utilize range sensing (LiDAR) for reliable obstacle detection.
  + Read sensor data directly from simulated sensors, in compliance with project requirements.

## System Integration and Validation

* + Integrate perception, control, and decision-making modules within ROS 2.
  + Validate system performance through simulation in Gazebo.
  + Analyze the robot’s behavior during both lane keeping and obstacle avoidance phases.

# Robot Platform Description (Mecanum Robot)

* The robot is based on a four-wheel Mecanum drive configuration, enabling holonomic motion (independent control of longitudinal, lateral, and rotational velocities).
* Each wheel is equipped with rollers angled at 45°, allowing omnidirectional movement without changing robot orientation.
* Compared to a differential drive robot, the Mecanum platform provides greater maneuverability, especially during lane transitions and obstacle avoidance.

We used four Mecanum wheels in our project. The wheel topology was the same as figure 2. The direction and the velocity of the diagonal wheels were set independently. Using the same speed in each wheel at the same time during the operation led us to get eight directions for the robot’s motion without changing its orientation. By changing the velocities of the diagonal wheels we achieved a motion between 0° to 360°. For example, to accomplish a transversal motion to the right, the right wheels were rotated against each other inwardly, while the left wheels were rotated against each other outwardly (See Figure 4). By using the same technique we achieved all eight different motions shown in Figure 4.

A set of square objects with arrows

AI-generated content may be incorrect.

Figure 1 Motions of Omnidirectional platform

## Mecanum Wheels

In this design, similar to the Omni wheel, There are a series of free moving rollers attached to the hub but with an 45° of angle about the hub's circumference but still the overall side profile of the wheel is circular.

Omnidirectional motion can be reached by mounting four Mecanum wheels on the corners of a four-sided base. Because of the angled rollers, the mechanical design is much more difficult, but due to the smoother transfer of contact surfaces a higher loads can be supported.

A drawing of a fan

AI-generated content may be incorrect.

A diagram of a mechanical scheme

AI-generated content may be incorrect.

Figure 2 Mecanum wheel design

Figure 3 Robot motion according to the direction and angular speed of the wheels

# KINEMATIC

### Kinematic Model of the Mecanum Robot

* The robot motion is described using Mecanum wheel kinematics, mapping wheel angular velocities to robot velocities:
  + Longitudinal velocity
  + Lateral velocity
  + Angular velocity
* Forward and inverse kinematic equations were implemented to convert:
  + Desired robot velocities → wheel speeds
  + Wheel feedback → robot motion estimation

### Justification for Using Mecanum Drive

* Although the project specification targets a differential drive robot, a Mecanum robot was selected because:
  + It supports lateral (sideways) motion, which simplifies lane changing.
  + It reduces the need for large heading changes during obstacle avoidance.
  + It allows smoother and faster transitions between lanes while maintaining stability.
* The control strategy was designed to respect the same task objectives (lane keeping and obstacle avoidance) defined in the project description.

### A diagram of a square with arrows and lines AI-generated content may be incorrect.The configuration parameters and system velocities are defined as follows:

A paper with text and images

AI-generated content may be incorrect.

A diagram of a mathematical equation

AI-generated content may be incorrect.

Figure 4 Wheels Configuration and Posture definition

A white text with black text

AI-generated content may be incorrect.

Figure 5 wheel 1 in the robot coordinate

A diagram of a mathematical equation

AI-generated content may be incorrect.

Figure 6 wheel 1 motion principle

A black and white image of a mathematical equation

AI-generated content may be incorrect.So, we can calculate the velocity of the wheel i and the tangential velocity of the free roller attached to the wheel touching the floor:

A math equations on a white background

AI-generated content may be incorrect.considering the equations (eq.1) , the velocity of the wheel 𝑖 in the frame 𝑆𝑖𝑃𝑖𝐸𝑖 , can be derived by:

A black and orange math symbols

AI-generated content may be incorrect.The transformation matrix from velocities of the ith wheel to its center:

The velocity of the wheel’s center translated to the XROYR coordinate system can be achieved by equation 7.

A black text on a white background

AI-generated content may be incorrect.

Then, the transformation matrix from the ith wheel’s center to the robot coordinate’s system can be obtained from equation 5.

A black and white text

AI-generated content may be incorrect.

A mathematical equation with black text

AI-generated content may be incorrect.Since the robot’s motion is planar, we also have:

A white background with black and white clouds

AI-generated content may be incorrect.Where:

A mathematical equation with letters and numbers

AI-generated content may be incorrect.From (eq.3) and (eq.5), the inverse kinematic model can be obtained:

*As r\_i ≠ 0 , 0 < |γ\_i| < π/2 , det(^P\_i T\_R) ≠ 0, det(^w\_i T\_(P\_i)) ≠ 0*

hence, by merging equations 4 and 6 the robot’s base velocity (at point O) related to the rotational velocity of the ith wheel can be obtained from eq. 9.

A black text on a white background

AI-generated content may be incorrect.

According to eq.3 and eq.4 there is a relationship between variables in each robot’s wheels frames and its center. And with the inverse kinematic, the velocity of the system can be obtained by implementing 𝑣ir the linear velocity and ωi the rotational speed of wheel ith in eq.10 and the contrary in eq.11.

A group of math equations

AI-generated content may be incorrect.

A group of math equations

AI-generated content may be incorrect.

A math equations and formulas

AI-generated content may be incorrect.Considering the fact that 𝑙𝑖𝑥 = 𝑙𝑖cos𝑎𝑖 and 𝑙𝑖𝑦 = 𝑙𝑖𝑠𝑖𝑛𝑎𝑖 , and assuming that the wheels are in a same size, the transformation matrix is:

A group of math equations

AI-generated content may be incorrect.Since there is a relation between independent variables 𝑣𝑖𝑟 and 𝜔𝑖 in each joint and the systems angular and linear velocity, assuming that there is no wheel slipping on the ground, the system inverse kinematic can be obtained by eq.14.

A group of math equations

AI-generated content may be incorrect.eq.15 shows the Jacobian matrix for the system’s inverse kinematic:

And for the forward kinematic according to the eq.10, we have:

A white background with black dots

AI-generated content may be incorrect.

## The Relation between Motions and the Translation MATRIX

Analyzing the motion of a four Mecanum wheeled robot brings out the following conclusion: According to the inverse kinematic, there is a relationship between velocities in each joint and the robot’s center velocity, thus, the velocity of the robot’s center is reflected by and obtained from an individual wheels velocity. According to the robot kinematic, inverse kinematics can be achieved when the rank of the system is less than the rank of the Jacobian matrix for each wheel of the robot that reduces the degree of freedom of the robot’s joints. Hence in a four Omni-differential design, the kinematic works with following conditions:

* R Jacobian full column rank, i.e. if rank (R) = 3, the robot performs a better movement.
* The rank of the Jacobian matrix column dissatisfaction, i.e. if the rank (R)

## FOUR MECANUM OMNIDIRECTIONAL SOLUTION

Typical Mecanum four system shown in Figure 2; the parameters of this configuration are shown in table 1. In this configuration wheels sizes are the same.

A table with numbers and symbols

AI-generated content may be incorrect.

By replacing the parameters of Table 1 in matrix (eq. 15) and considering eq.14 we have come up with:

A group of math equations

AI-generated content may be incorrect.

A math equations with numbers and symbols

AI-generated content may be incorrect.According to equations (10) and (11) for Forward and Inverse kinematics there is:

Longitudinal Velocity:



Transversal Velocity:

Angular velocity:

The resultant velocity and its direction in the stationery coordinate axis (x, y, z) can be achieved by the following equations (eq. 25, 26):

A white background with black dots

AI-generated content may be incorrect.

The results were systematically obtained by using kinematic equations that were similar to those achieved from the experimental results. The results show that the platform performs full omnidirectional motions. This shows that by using Mecanum wheels in the platform the robot can achieve any direction between 𝟎 ° to 𝟑𝟔𝟎 °without changing its orientation.

# CAD Design

A black rectangular object with wheels and a small figure

AI-generated content may be incorrect.

Figure 7 Robot CAD Model without Covering

A red rectangular object with wheels

AI-generated content may be incorrect.

Figure 8 Robot Model with covering CAD Model

A road with a straight road

AI-generated content may be incorrect.

Figure 9 Track 01CAD Model

A green box on a road

AI-generated content may be incorrect.

Figure 10 Track 02 CAD Model

# A screenshot of a computer AI-generated content may be incorrect.Project Package(mec)

Figure 11 mec package folders

* The mec package is a complete ROS2 setup for a mecanum-wheeled robot, including models, meshes, controllers, launch files, scripts, RViz configuration, and simulation worlds for testing motion, perception, and scenarios.

## Model Folder Description

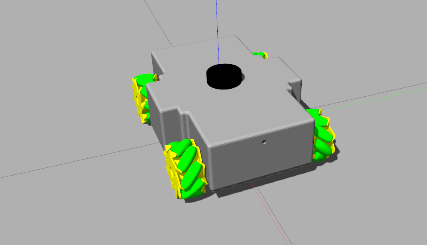
* The model folder contains all URDF/Xacro files responsible for defining the robot structure, sensors, control interfaces, and simulation environments.  
   The modeling approach is modular, allowing easy maintenance, scalability, and reuse.

Figure 12 urdf model

## Meshes Folder Description

* The meshes folder contains all 3D STL models used to represent the robot and the simulation environment in Gazebo. These models are used only for visualization and collision representation and do not include control or logic.

## Worlds Folder Description

* Contains Gazebo world files for robot simulation, including tracks, obstacles, and empty environments.

## Rviz Folder Description

* Holds RViz configuration files to visualize the robot, sensors, and environment in ROS2.

## Config Folder Descreption

* This contains the control settings for the robot, specifically the mecanum wheels:
* controller\_manager: Manages all controllers, sets the update rate, and broadcasts joint states.
* Wheel velocity controllers: Each wheel has a ForwardCommandController to control its velocity.
* In short, this file connects the motors to the simulation and defines how wheel velocities are controlled.

## A diagram of a machine AI-generated content may be incorrect.New\_launch Folder Description

* Contains ROS2 launch files to start the robot, its drivers, perception nodes, and simulation environments.
* There are 3 files
  + 1. driver\_Track01.launch.py
    2. driver\_Track02.launch.py
    3. perception.launch.py

A diagram of a robot launch flow

AI-generated content may be incorrect.

Figure 13 High-Level Flow (Launch Sequence)

## Scripts Folder Description

* Includes Python scripts for robot control, sensor processing, lane detection, obstacle avoidance, and scenario management.

### A diagram of a process AI-generated content may be incorrect.A screenshot of a computer program AI-generated content may be incorrect.driver\_node.py

Figure 14 flowchart of driver\_node

### A diagram of a computer program AI-generated content may be incorrect.lane\_detection\_node.py

Figure 15 flowchart of lane\_detection\_node

### obstacle\_Detection\_Node.py

A diagram of a program

AI-generated content may be incorrect.

Figure 16 flowchart ofobstacle\_detection\_node

### A diagram of a flowchart AI-generated content may be incorrect.mecanum\_motion.py

Figure 17 flowchart of mecanum\_motion

### A diagram of a computer program AI-generated content may be incorrect.imu.py

### 

Figure 18 flowchart of imu\_node

### A diagram of a computer program AI-generated content may be incorrect.odom\_node.py

Figure 19 flowchart of odom\_node

### A diagram of a computer program AI-generated content may be incorrect.scenario\_01.py

Figure 20 flowchart of scenario\_01

### A screenshot of a computer AI-generated content may be incorrect.scenario\_02.py

Figure 21 flowchart of scenario\_02

# REFERENCES

[1] O. Diegel, A. Badve, G. Bright, J. Potgieter, and S. Tlale, “Improved Mecanum Wheel Design for Omni-directional Robots,” no. November, pp. 27–29, 2002.

[2] I. Doroftei, V. Grosu, and V. Spinu, Omnidirectional Mobile Robot - Design and Implementation, Bioinspiration and Robotics Walking and Climbing Robots, no. September. I-Tech, 2007.

[3] R. P. A. van Haendel, “Design of an omnidirectional universal mobile platform,” National University of Singapore, 2005.

[4] T. A. Baede, “Motion control of an omnidirectional mobile robot,” 2006.

[5] X. Li and A. Zell, “Motion control of an omnidirectional mobile robot,” 2006