

INTEGRATION OF TRAILER AND TRACTOR MOBILE ROBOT CONSIDERING TOWING DYNAMICS

Submitted: 28th February 2025; accepted: 25th April 2025

Ünal Dana, Levent Çetin

DOI: 10.14313/jamris-2026-003

Abstract:

The industrial robots are used in different areas, one of these areas is towing trailer's. This research proposes the use of an omnivheel mobile robot, designed specifically for towing purposes, to enhance the towing capabilities. The study focuses on analyzing, controlling, and testing the dynamics of a trailer under the influence of external towing forces applied by an omnivheel mobile robot. The study includes dynamical analysis, path planning, tracking control, prototype design, tuning of selected actuators, and simulation results of the proposed omnivheel tractor robot. The dynamic equations, the path planning algorithm, the path tracking methods, and the PID tuning of actuators are tested in a MATLAB Simulink Simscape environment using the designed 3D models. The simulation results are used to fine-tune the system parameters and minimize errors. Improved tuning of these parameters leads to better simulation outcomes. The final simulation results show that positional accuracy reaches 4.0 cm RMS precision (peak deviations: 7.6 cm) over a 6.63-meter course, and speed deviations remain below %4.3. Angular control maintains stability by capping errors at 0.117 rad (average: 0.008 rad) and limiting transient overshoot to %17.17 for aggressive maneuvers.

Keywords: omnivheel mobile robot, dynamical analysis, tracking control, path planning, PID control

1. Introduction

The advancements in warehouse robotics over the past decade have been significant [1], driving increased attention towards controlling and evaluating various mobile robot designs for warehouse environments [2, 3]. This research focuses primarily on key areas such as energy efficiency, path planning [4, 5], path tracking [6, 7], dynamic behavior analysis [4,8–10] and mechanical design considerations [2,11]. These topics form the foundation for mobile robotics, specifically for autonomous towing of trailers exploiting in transferring products between warehouse and manufacturing facilities or vice versa.

Previous studies in mobile robotics have examined trailer systems [12–14], which share mechanical similarities with non-holonomic two-wheel mobile robots. A significant portion of the literature focuses on trailer dynamics and control for path tracking, often employing Lyapunov-based approaches to address tracking control for non-holonomic wheeled trailer's [15, 16].

While these methods offer useful insights, they elaborately calculate required torques for actuated wheels by implementing differential drive models concerning only trailers and combined dynamics of tractor and trailer. Nevertheless, in context of mobilization and autonomy of already deployed trailers, these strategies become invalid because of the fact that only towing of unactuated trailers is feasible in their controls. To tow for motion control of the trailer with non-holonomic dynamics, a tractor robot should have fully controlled actuation for planar motion.

Omnivheel robots, known for their maneuverability and holonomic kinematics, have been a significant focus of research. However, their complex kinematic and dynamic analyses present additional challenges [8, 11, 17]. Control strategies for omnivheel robots have been successfully tested, employing a PID control method to adjust wheel accelerations [17], implementing adaptive controllers [9]. This approach calculates required motor torques using inverse dynamic analysis and adjusts the robot's velocity based on position errors. The desired velocity and planned path are provided as input by the user, with the system controlling the robot accordingly. Although this method is effective, the article does not specify the exact path tracking or planning algorithms tested alongside the control system.

Path planning is a critical aspect that enables autonomous movement in mobile robots and requires an accompanying path tracking algorithm [4, 7]. During path planning, a mobile robot typically relies on map data, which may be preexisting or acquired in real-time using sensors. These data are essential for computing an optimal route from the initial position to the destination through path planning algorithms. The resulting path can be represented as a set of discrete poses or as a mathematical function, both of which require a tracking algorithm to follow the planned route [5, 6]. Path tracking algorithms, such as the Pure Pursuit method, determine the next destination based on a lookahead distance. This approach continuously adjusts the robot's trajectory by selecting the closest point along the path that lies within the lookahead distance, ensuring smooth and efficient navigation.

This study addresses the challenge of towing a trailer using an omnivheel mobile robot, offering a unique approach to this common problem in mobile robotics.

While previous research has examined the dynamics of omniwheel robots [8, 11, 17–19] and trailer’s [4, 7, 10, 13, 19–22], this study re-examines them due to changes in the problem’s requirements. It derives the dynamics of towing and the forces involved, providing new insights into their interaction.

This study introduces a method using cubic Bezier functions with pure pursuit tracking. This approach reduces the need for excessive maneuvers and corrects rotational errors, making it suitable for all trailer operations. The control strategy employs dynamic analysis with Lyapunov stability methods. Though commonly used for non-holonomic robots, this method is adapted here for towing applications, with experiments conducted to assess its effectiveness. This research contributes significantly by enhancing warehouse robotics efficiency in towing operations. Additionally, the findings benefit any system utilizing towed trailer’s, offering valuable insights and methodologies for improved performance.

2. Dynamical Analysis

The proposed design of the towed trailer mobile robot consists of two parts: the non-holonomic trailer part and the holonomic omniwheel tractor part. The two parts are connected to each other using a revolute joint, this joint allows the rotations on Z direction and transmits only the applied forces without deflection but not the torque. To achieve the desired pose of the trailer, the required towing forces on the revolute joint must be calculated. Then, the corresponding motor torques or motor speeds of the towing robot must be determined based on the calculated external forces of the trailer.

The following assumptions and problem-based constraints were considered throughout the whole research.

- The trailer has 3 degrees of freedom.
- Slip between the wheels and the ground is neglected.
- The center of mass of the towing robot is at the geometrical center of its body, equidistant from all three wheels, as shown in Figure 1a.
- The center of mass of the trailer is located between the middle of the two wheels, as shown in Figure 1b.

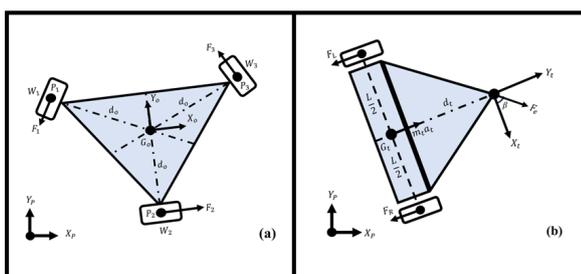


Figure 1. 2D illustration of (a) omniwheel mobile robot and (b) trailer

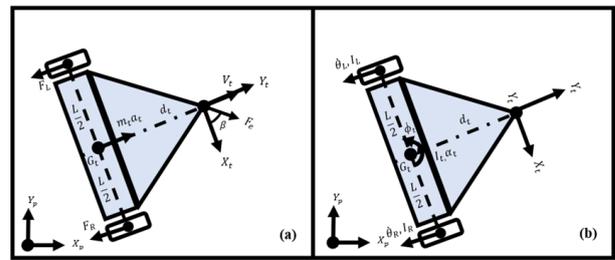


Figure 2. Forces (a), torques (b) on trailer

2.1. Towing Dynamics

To model the dynamics of the trailer’s motion the geometrical and mechanical relationship between the trailer’s wheels and body frame should be known. All of the necessary parameters are presented in Figure 2.

The trailer’s linear velocity V_t is determined as the average of the left and right wheel velocities V_L and V_R , while the angular velocity $\dot{\phi}_t$ depends on the difference between these velocities and the robot’s wheelbase L .

These relationships are essential for determining how the robot’s wheels must move to achieve a desired trajectory. Specifically, the relationship between the linear and angular velocities is given by Equation (1).

$$V_t = \frac{V_L + V_R}{2}, \quad \dot{\phi}_t = \frac{V_R - V_L}{L} \tag{1}$$

Furthermore, the reverse calculation allows to determine the required wheel velocities for a given linear and angular velocity. By rearranging the equations in Equation (1) the following expressions for the left and right wheel velocities are obtained as shown in Equation (2).

$$V_L = \frac{2V_t - \dot{\phi}_t L}{2}, \quad V_R = \frac{2V_t + \dot{\phi}_t L}{2} \tag{2}$$

The velocities of the left and right wheels are related to the angular velocities θ_L and θ_R through the wheel radius r_t , as shown in Equation (3).

$$V_L = \dot{\theta}_L r_t, \quad V_R = \dot{\theta}_R r_t \tag{3}$$

To express the angular velocities in terms of the trailer’s linear velocity V_t and angular velocity $\dot{\phi}_t$, the matrix equation in Equation (4) is derived by using Equation (2) and Equation (3).

$$\begin{bmatrix} \dot{\theta}_L \\ \dot{\theta}_R \end{bmatrix} = \frac{1}{2r_t} \begin{bmatrix} 2 & -L \\ 2 & L \end{bmatrix} \begin{bmatrix} V_t \\ \dot{\phi}_t \end{bmatrix} \tag{4}$$

This equation allows to determine the angular velocities of the left and right wheels based on the trailer’s linear and angular velocities. The relationships for the angular accelerations of the wheels in terms of the linear and angular accelerations of the trailer is expressed in Equation (5) by taking the derivative of Equation (4).

$$\begin{bmatrix} \ddot{\theta}_L \\ \ddot{\theta}_R \end{bmatrix} = \frac{1}{2r_t} \begin{bmatrix} 2 & -L \\ 2 & L \end{bmatrix} \begin{bmatrix} \dot{V}_t \\ \dot{\phi}_t \end{bmatrix} \tag{5}$$

These relationships helps to understand how changes in the trailer's velocity affect the angular accelerations of the wheels and by taking inverse of this equation we can understand the inverse relationship shown on Equation (6).

$$\begin{bmatrix} \dot{V}_t \\ \ddot{\phi}_t \end{bmatrix} = \begin{bmatrix} \frac{r_t}{2} & \frac{r_t}{L} \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_L \\ \ddot{\theta}_R \end{bmatrix} \quad (6)$$

The forces exerted on the wheels are related to the angular accelerations of the wheels through their moments of inertia. Specifically, the forces F_L and F_R are related to the angular accelerations $\ddot{\theta}_L$ and $\ddot{\theta}_R$ by the system of equations shown in Equation (7).

$$F_R = \frac{I_R \ddot{\theta}_R}{r_t}, \quad F_L = \frac{I_L \ddot{\theta}_L}{r_t} \quad (7)$$

Equation (8) represents the balance of forces and torques acting on the trailer, where $\vec{F}_e \times \vec{L}_e$ represents the external torque, and the terms involving \vec{F}_L and \vec{F}_R describe the torque contributions from the wheels. This balance is fundamental for controlling the robot's rotational motion.

$$\vec{F}_e \times \vec{L}_e + \vec{F}_L \times \vec{L}_t + \vec{F}_R \times \vec{L}_t = I_t \ddot{\phi}_t \quad (8)$$

Analysis of the forces along the y -axis, which are influenced by the wheel forces and the trailer's acceleration is given by Equation (9).

$$F_{ey} - F_L - F_R = m_t \dot{V}_t \quad (9)$$

This equation shows that the net force along the y -axis is the sum of the forces from the left and right wheels, minus the external force F_{ey} , which is proportional to the robot's change in velocity. The force F_{ex} acting along the x -axis can be derived from the moments of inertia of the robot and the geometry of the system. It is expressed in Equation (10).

$$F_{ex} = \frac{F_L L}{2d_t} - \frac{F_R L}{2d_t} - \frac{I_t \ddot{\phi}_t}{d_t} \quad (10)$$

Further, the force F_{ex} can also be described in terms of the angular accelerations of the left and right wheels, as shown in Equation (11).

$$F_{ex} = \frac{I_L \ddot{\theta}_L L}{2d_t r_t} - \frac{I_R \ddot{\theta}_R L}{2d_t r_t} + \frac{I_t \ddot{\theta}_L r_t - I_t \ddot{\theta}_R r_t}{d_t L} \quad (11)$$

This shows how the wheel inertia and angular accelerations influence the external force acting on the trailer. The forces along the y -axis is as shown in Equation (12).

$$F_{ey} - F_L - F_R = m_t \dot{V}_t \quad (12)$$

This equation shows that the net force along the y -axis depends on the forces from both wheels and the external force. The total force along the y -axis can also be expressed as in Equation (13).

$$F_{ey} = F_L + F_R + m_t \dot{V}_t \quad (13)$$

This shows that the total force along the y -axis is the sum of the forces from the left and right wheels, plus the contribution from the trailer's change in velocity. The force F_{ey} can be expressed in terms of the angular accelerations of the left and right wheels, as shown in Equation (14).

$$F_{ey} = \frac{I_L \ddot{\theta}_L}{r_t} + \frac{I_R \ddot{\theta}_R}{r_t} + m_t \frac{\dot{V}_R + \dot{V}_L}{2} \quad (14)$$

The force F_{ey} can be written as in Equation (15).

$$F_{ey} = \frac{I_L \ddot{\theta}_L}{r_t} + m_t \frac{\ddot{\theta}_L r_t}{2} + \frac{I_R \ddot{\theta}_R}{r_t} + m_t \frac{\ddot{\theta}_R r_t}{2} \quad (15)$$

This equation gives a more detailed relationship between the forces and the angular accelerations of the left and right wheels. The final equation of model is a matrix representation of the external forces along the x - and y -axes in terms of the angular accelerations of the left and right wheels. This matrix form is useful for numerical computation and simulation of the trailer's dynamics. The matrix equation is given in Equation (16).

$$\begin{bmatrix} F_{ex} \\ F_{ey} \end{bmatrix} = \begin{bmatrix} \frac{I_L L}{2d_t r_t} + \frac{I_t r_t}{d_t L} & -\frac{I_R L}{2d_t r_t} - \frac{I_t r_t}{d_t L} \\ \frac{I_L}{r_t} + m_t \frac{r_t}{2} & \frac{I_R}{r_t} + m_t \frac{r_t}{2} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_L \\ \ddot{\theta}_R \end{bmatrix} \quad (16)$$

2.2. Omniwheel Robot Dynamics

The designed omniwheel mobile robot, used for towing the trailer, calculates the external forces affecting the trailer to determine the needed equivalent motor angular velocities. In this context, the geometrical relationships between the motor vectors resulting from the mechanical connections to the mobile robot body play a crucial role. These relationships are fundamental to calculating the necessary velocities. All required geometrical and mechanical parameters are specified in the conceptual design in Figure 3.

The velocity vector of the omniwheel robot's center of mass, denoted by \dot{p}_o , is given by the following equation.

$$\dot{p}_o = [V_{x_o} \quad V_{y_o} \quad \dot{\phi}_o]^T, \quad \dot{q}_o = [\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^T \quad (17)$$

Equation (17) defines the velocity vector \dot{p}_o of the omniwheel robot's center of mass, where V_{x_o} and V_{y_o}

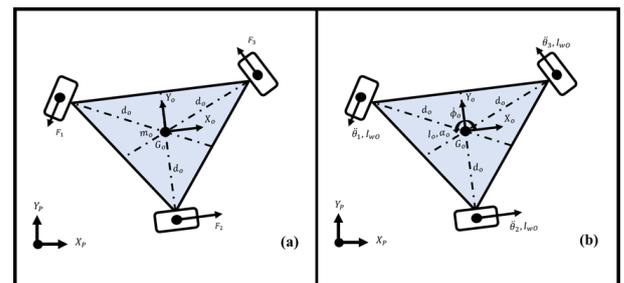


Figure 3. Forces (a), torques (b) on omniwheel mobile robot

represent the linear velocities along the X_o and Y_o axes, and $\dot{\phi}_o$ is the angular velocity of the omnivheel robot.

To calculate the linear velocity along the X_o direction V_{x_o} , we use the following equation.

$$V_{x_o} = \frac{2V_2 - V_1 - V_3}{3} \quad (18)$$

Equation (18) combines the contributions of the individual motor velocities V_1 , V_2 , and V_3 , where the motors are arranged in a triangular configuration. The equation shows how the omnivheel robot's X_o direction velocity is influenced by these motor velocities.

Similarly, the linear velocity along the Y_o direction V_{y_o} is computed as:

$$V_{y_o} = \frac{\sqrt{3}V_3 - \sqrt{3}V_1}{3} \quad (19)$$

Equation (19) shows that the Y_o direction velocity V_{y_o} depends on the motor velocities V_1 and V_3 , with a factor of $\sqrt{3}$ from the geometry of the robot's wheels.

$$\dot{\phi}_o = \frac{V_1 + V_2 + V_3}{3d_o} \quad (20)$$

Equation (20) provides the angular velocity at the robot's center of mass based on the sum of the motor velocities and the distance d_o from the center of mass to the wheels.

The individual motor velocities V_1 , V_2 , and V_3 are related to the motor angular velocities $\dot{\theta}_1$, $\dot{\theta}_2$, and $\dot{\theta}_3$ by the following relations.

$$V_1 = \dot{\theta}_1 r_o, \quad V_2 = \dot{\theta}_2 r_o, \quad V_3 = \dot{\theta}_3 r_o \quad (21)$$

In equation (21), r_o represents the radius of the omnivheels, and each omnivheel's velocity is proportional to its angular velocity.

Rearranging the system of equations for linear and angular velocities to write the matrix equation that transforms the motor angular velocities into the linear velocities of the omnivheel robot's center of mass.

$$\begin{bmatrix} V_{x_o} \\ V_{y_o} \\ \dot{\phi}_o \end{bmatrix} = r_o \begin{bmatrix} -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ \frac{1}{3d_o} & \frac{1}{3d_o} & \frac{1}{3d_o} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (22)$$

Equation (22) expresses the relationship between the motor angular velocities and the linear velocities of the robot's center of mass in matrix form. This matrix allows for the direct calculation of the robot's velocities from the motor angular velocities.

The inverse of this relationship is found to calculate the motor angular velocities $\dot{\theta}_1$, $\dot{\theta}_2$, and $\dot{\theta}_3$ from the linear velocities V_{x_o} , V_{y_o} , and $\dot{\phi}_o$.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{r_o} \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & d_o \\ 1 & 0 & d_o \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & d_o \end{bmatrix} \begin{bmatrix} V_{x_o} \\ V_{y_o} \\ \dot{\phi}_o \end{bmatrix} \quad (23)$$

Equation (23) provides the inverse matrix that calculates the motor angular velocities from the omnivheel robot's linear velocities and angular velocity.

The total torque T_{z_o} acting on the omnivheel robot due to the forces exerted by the motors can be expressed as:

$$T_{z_o} = F_1 d_o + F_2 d_o + F_3 d_o + I_o \dot{\omega}_o \quad (24)$$

Equation (24) describes the total torque acting on the robot. It includes the contributions from the individual motor forces F_1 , F_2 , and F_3 , as well as the rotational inertia term $I_o \dot{\omega}_o$.

$$F_{x_o} = \frac{2F_2 - F_3 - F_1}{3} + m_o a_x \quad (25)$$

$$F_{y_o} = \frac{\sqrt{3}F_3 - \sqrt{3}F_1}{3} + m_o a_y \quad (26)$$

Equations (25) and (26) calculate the forces along the X_o and Y_o axes.

The forces exerted by each motor can be calculated as:

$$F_1 = \frac{I_{w_o} \ddot{\theta}_1}{r_o}, \quad F_2 = \frac{I_{w_o} \ddot{\theta}_2}{r_o}, \quad F_3 = \frac{I_{w_o} \ddot{\theta}_3}{r_o} \quad (27)$$

In equation (27), the forces generated by the individual motors are related to the motor angular accelerations $\ddot{\theta}_1$, $\ddot{\theta}_2$, and $\ddot{\theta}_3$, and depend on the moment of inertia I_{w_o} and the radius r_o .

To express the forces F_{x_o} and F_{y_o} in terms of the motor angular accelerations, we use the following equations.

$$F_{x_o} = A_{11} \ddot{\theta}_1 + A_{12} \ddot{\theta}_2 + A_{13} \ddot{\theta}_3 \quad (28)$$

$$F_{y_o} = A_{21} \ddot{\theta}_1 + A_{22} \ddot{\theta}_2 + A_{23} \ddot{\theta}_3 \quad (29)$$

$$T_{z_o} = A_{31} \ddot{\theta}_1 + A_{32} \ddot{\theta}_2 + A_{33} \ddot{\theta}_3 \quad (30)$$

Equations (28) (29) and (30) are expressed with coefficients to shorten the equations.

These relationships can be written in matrix form:

$$\begin{bmatrix} F_{x_o} \\ F_{y_o} \\ T_{z_o} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} \quad (31)$$

Equation (31) expresses the forces and torques in terms of the motor angular accelerations in matrix form.

The coefficients A_{ij} are defined as follows:

$$A_{11} = A_{13} = -\frac{I_{w_o}}{3r_o} - m_o \frac{r_o}{3} \quad (32)$$

$$A_{12} = \frac{2I_{w_o}}{3r_o} + m_o \frac{2r_o}{3} \quad (33)$$

$$A_{21} = -A_{23} = -\frac{\sqrt{3}I_{w_o}}{3r_o} - m_o \frac{\sqrt{3}r_o}{3} \quad (34)$$

$$A_{22} = 0 \quad (35)$$

$$A_{31} = A_{32} = A_{33} = \frac{I_{w_o} d_o}{r_o} + I_o \frac{r_o}{3d_o} \quad (36)$$

Equations (32) through (36) define the coefficients A_{ij} . The angular accelerations of the motors can be expressed as a function of the forces and torques.

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} F_{x_o} \\ F_{y_o} \\ T_{z_o} \end{bmatrix} \quad (37)$$

Equation (37) is the inverse of Equation (31). This inverse equation calculates $\ddot{\theta}_1$, $\ddot{\theta}_2$, and $\ddot{\theta}_3$ based on the equivalent forces and torques on the robot. The coefficients B_{ij} are defined as follows:

$$B_{11} = B_{31} = -2B_{21} = -\frac{1}{2}(I_{w_o}r_o + m_o r_o) \quad (38)$$

$$B_{22} = 0 \quad (39)$$

$$B_{12} = -B_{32} = \frac{-3}{2(\sqrt{3}I_{w_o}r_o + \sqrt{3}m_o r_o)} \quad (40)$$

$$B_{13} = B_{12} = B_{11} = \frac{d_o r_o}{3I_{w_o}d_o^2 + I_o r_o^2} \quad (41)$$

Equations (38) through (41) define the coefficients of B_{ij} .

2.3. Mobile Robot Integration

The integration of two systems, the trailer and the omniwheel mobile robot, can be modeled using the dynamic analysis of both systems. When these systems are integrated as shown in the Figure 4, the external forces acting on the trailer, denoted F_{ex} and F_{ey} , are equal to the tractor's resulting tracking forces F_{x_o} and F_{y_o} , as expressed in Equation (28) and Equation (29).

By multiplying the Jacobian from the dynamic models of both the trailer and the omniwheel mobile robot, we can establish the relationship between the wheel accelerations of the omniwheel mobile robot and the trailer wheel velocities. The resulting equation is provided below, which models this relationship.

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = K_o K_t \begin{bmatrix} \dot{\theta}_L \\ \dot{\theta}_R \end{bmatrix} \quad (42)$$

Equation (42) defines how the wheel accelerations of the omniwheel mobile robot are related to the trailer wheel velocities through the product of the Jacobian matrices K_o and K_t .

The matrix K_t , representing the trailer's dynamic model, is given by:

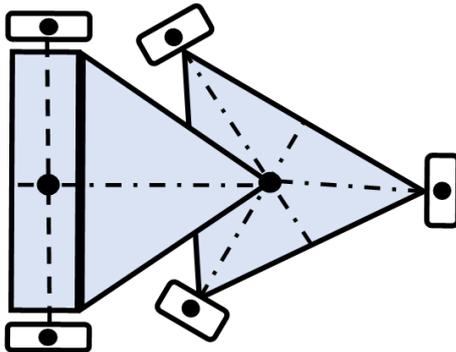


Figure 4. Mobile robot integrated trailer representation

$$K_t = \begin{bmatrix} \frac{I_{wt}L}{2d_t r_t} + \frac{I_t r_t}{d_t L} & -\frac{I_{wt}L}{2d_t r_t} - \frac{I_t r_t}{d_t L} \\ \frac{I_{wt}}{r_t} + m_t \frac{r_t}{L} & \frac{I_{wt}}{r_t} + m_t \frac{r_t}{L} \\ 0 & 0 \end{bmatrix} \quad (43)$$

Equation (43) provides the matrix K_t , which takes into account the physical properties of the trailer, including moments of inertia I_{wt} and I_t , the wheel radius r_t , and the distance L between the wheels. This matrix provides the relationship between the trailer's wheel accelerations and its dynamics.

The matrix K_o , which models the dynamic properties of the omniwheel mobile robot, is defined as:

$$K_o = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \quad (44)$$

Equation (44) shows the matrix K_o , where the elements B_{ij} represent coefficients that account for the robot's geometry and dynamics, including the effects of forces and torques on the wheel accelerations. This matrix is essential in transforming the trailer's wheel velocities into the mobile robot's motor accelerations.

3. Control Method

As it is shown in dynamical analysis. The control task for motion control of trailer realized by applying the towing forces which are generated via the omniwheel tractor robot. Therefore, control requirements are given as: path planning between way points concerning non-holonomic trailer kinematics, derivation of required forces using its differential kinematics based control strategy and actuator level control of omniwheel robot motion for controlled towing.

3.1. Path Planning

Path planning in trailer-tractor robots are essential in order to achieve autonomous movement [4, 5, 20]. This autonomous movement can be planned in different ways, they can aim choosing the most efficient, feasible or wanted set of movements for this reason the path planning of mobile robots is a complicated and important task. Path planning of mobile robots can be done using both mathematical functions and algorithms depending on the path planning strategy. A path planning strategy for a mobile robot is often determined by considering a planned scenario or a set of movements, the planned scenario includes the work environment and all of the constraints of the mobile robot. An example for this is if a mobile robot is in a closed and blocked environment, the movement of the mobile robot is unconstrained and full autonomy of the mobile robot is desired from the path planner, the most common path planning strategies uses local and global maps [23, 24] to determine the most efficient movement sets by help of many different algorithms [25–28]. These path planning strategies will aim to be the most beneficent for that certain hypothetical scenarios. In this research a path planning method is required in order to test the path tracking success of the trailer tractor robot.

Due to this need, a hypothetical scenario and all of the defined constraints of the trailer tractor robot are identified and based on this the designed path planning is simulated and tested in MATLAB Simulink.

In an ideal environment without obstacles, the simplest method to achieve the desired pose is to move along the shortest linear path. However, this straight path is not suitable for the given problem because it would result in rotations around the center of mass of the trailer. These rotations around the center of mass of the trailer is harder to apply because of the non-holonomic structure of the trailer. To solve this, a cubic Bézier curve (with $n = 3$) is employed to generate a path that avoids these rotations. The general form of the Bézier curve [29] is given by Equation (45).

$$f(t) = \sum_{i=0}^n b_i B_i^n(t), \quad 0 \leq t \leq 1 \quad (45)$$

This equation expresses the Bézier curve as a weighted sum of basis functions $B_i^n(t)$, where b_i are the control points, and t is the parameter that varies between 0 and 1. This sum produces a curve that interpolates between the control points.

The basis functions $B_i(t)$ for the Bézier curve are defined in Equation (46).

$$B_i(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n \quad (46)$$

These functions determine the influence of each control point on the final curve. The coefficients are computed using binomial expansions, ensuring that the curve smoothly interpolates between the given control points.

Algorithm 1 Path Generation Procedure

```

1: procedure PATHGENERATED( $P_f, P_i$ )
2:    $P_0 \leftarrow [P_i(1); P_i(2)]$ 
3:    $P_3 \leftarrow [P_f(1); P_f(2)]$ 
4:    $L \leftarrow \text{norm}(P_3 - P_0)$ 
5:    $P_1 \leftarrow (P_0 + 0.5L)[\cos(P_i(3)); \sin(P_i(3))]$ 
6:    $P_2 \leftarrow P_3 - 0.5L[\cos(P_f(3)); \sin(P_f(3))]$ 
7:    $N \leftarrow 50$ 
8:    $t \leftarrow \text{linspace}(0, 1, N)$ 
9:    $\text{path\_generated} \leftarrow \text{zeros}(2, N)$ 
10:  for  $i = 1$  to  $N$  do
11:     $\text{path\_generated}(:, i) \leftarrow (1 - t(i))^3 P_0 +$ 
     $3(1 - t(i))^2 t(i) P_1 + 3(1 - t(i)) t(i)^2 P_2 + t(i)^3 P_3$ 
12:  end for
13: end procedure

```

For cubic Bézier curves, where $n = 3$, the curve is determined by four control points. The path planning algorithm utilizes two control points P_1 and P_2 along with the initial and desired poses of the trailer. Control point P_1 is positioned towards the front of the trailer's initial pose, while control point P_2 is placed in the opposite direction, towards the front of the trailer's desired pose as shown on the Figure 5.

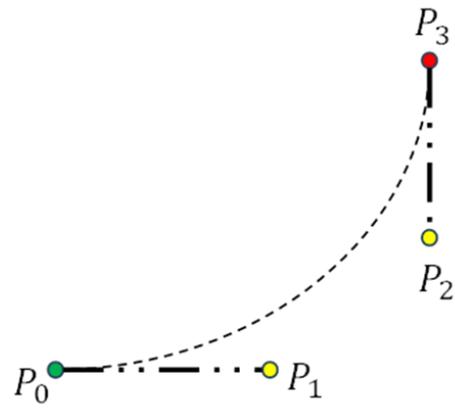


Figure 5. Cubic Bézier curve

The cubic Bézier curve ensures that the trailer follows a smooth path without excessive rotation around its center of mass, providing an efficient solution for path planning.

This approach avoids the complexities of more advanced algorithms while still achieving the desired motion. The generated path is calculated using the path generation procedure, described in Algorithm 1.

3.2. Path Tracking

Path tracking is the process that allows autonomous systems to travel along a desired trajectory. The process typically consists of several sets of rules; hence, these are referred to as algorithms.

The path tracking algorithms are responsible for determining the next waypoint on the planned path and calculating the desired velocities to reach the waypoint. Many different path tracking algorithms exist, but one of the most suitable algorithms for the trailer tractor system is the pure pursuit algorithm [6].

The pure pursuit algorithm, implemented in MATLAB Simulink, uses a specific distance called the lookahead distance to decide the next waypoint, as shown in Figure 6. The waypoints are generated from the path planner by dividing the generated path into a number of waypoints and listing them from start to finish. Each waypoint consists of the position and orientation of the vehicle in task space. The next waypoint is the farthest point within the radius of the lookahead distance.

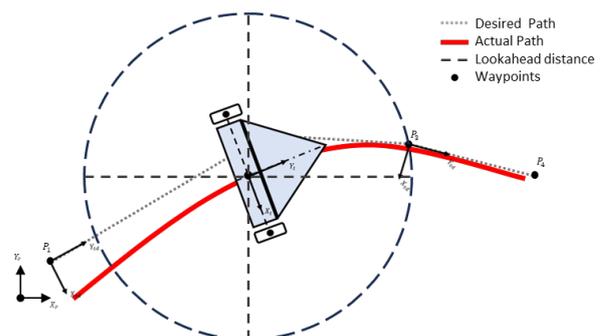


Figure 6. Pure pursuit algorithm

The waypoints and the actual pose of the vehicle are used to calculate the steering angle. The algorithm can generate a constant linear velocity, and the angular velocity can be bounded to the desired limits.

The success of trajectory tracking depends on the following factors:

- The number of waypoints in the path and the lookahead distance are important. Within the lookahead distance radius, there must always be at least one waypoint.
- The vehicle's desired linear velocity and angular velocity should be proportional to the lookahead distance. A lower lookahead distance causes the vehicle to oscillate around the planned path, while a higher lookahead distance causes the vehicle to converge to the planned path over a longer time. The parameters should be tuned with a constant distance between each waypoint for the algorithm to work regardless of the planned path.
- The algorithm always has a slight error due to the lookahead distance; therefore, the tuning of parameters is done by considering oscillations on the planned path.
- The pure pursuit algorithm does not stop the vehicle at the last waypoint; therefore, a stopping algorithm is needed.

Algorithm 2 Pure Pursuit Path Tracking Algorithm

```

1: procedure PUREPURSUIT( $P, P_v, \theta_v, L$ )
2:    $d_{min} \leftarrow Max$ 
3:    $P_{lh} \leftarrow None$ 
4:   for each  $P_i$  in  $P$  do
5:      $d \leftarrow Dist(P_v, P_i)$ 
6:     if  $d \geq L$  and  $d < d_{min}$  then
7:        $d_{min} \leftarrow d$ 
8:        $P_{lh} \leftarrow P_i$ 
9:     end if
10:  end for
11:   $dx \leftarrow P_{lh}(1) - P_v(1)$ 
12:   $dy \leftarrow P_{lh}(2) - P_v(2)$ 
13:   $\alpha \leftarrow atan2(dy, dx) - \theta_v$ 
14:   $k \leftarrow \frac{2 \cdot \sin(\alpha)}{L}$ 
15:   $\delta \leftarrow atan(k)$ 
16:  return  $\delta$ 
17: end procedure
    
```

3.3. Dynamic Tracking Control

The problem definition states that one of the objectives of this study is to control the trailer pose p_c and velocity $V_t, \dot{\phi}_t$ in the task space. This means that trailer dynamics need to be controlled with a tracking control method. The dynamic tracking control method should maintain the velocity at the referenced value while tracking continues, by calculating the forces or torques that need to be applied.

The dynamic tracking control equations for the trailer are given by:

$$F_x = -I_t \dot{w}_c - K_b w_c, \tag{47}$$

$$F_y = m_t \dot{V}_c + K_a V_c, \tag{48}$$

The dynamic tracking control based on the Lyapunov function [15, 16] works well for the non-holonomic wheeled mobile robot. As shown above Towing a trailer is a non-holonomic motion control problem which results in nonlinear mathematical model. Hence, Lyapunov control design method results in deployable control laws compatible with kinematics, it is adopted in our problem as a complementary part of motion planning [15, 16]. In Figure 8, the simulation setup for dynamic tracking control is shown.

Where \dot{V}_c is the trailer's acceleration, K_a is the tangential velocity control coefficient, V_c is the trailer's desired tangential velocity, \dot{w}_c is the trailer's angular acceleration, w_c is the trailer's desired angular velocity, and K_b is the angular velocity control coefficient. The Equations (47) and (48) describe the forces that need to be applied to maintain the desired velocity

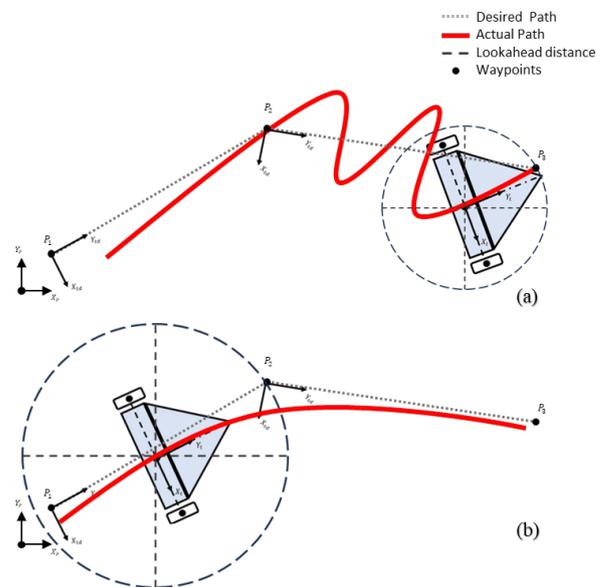


Figure 7. Pure pursuit algorithm example for (a) low, (b) high lookahead distance

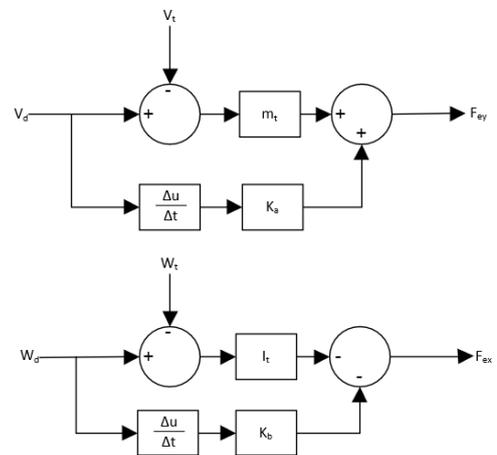


Figure 8. Dynamic tracking control

and orientation of the trailer. The control coefficients K_a and K_b are tuned to ensure that the path tracking is performed successfully. The method is based on Lyapunov stability, which guarantees that the system will remain stable while following the desired path.

3.4. Motor Tuning

The actuator is the selected motor Pololu 37Dx73L, 12 V 131:1 gear motor with an internal 64 CPR two-channel encoder. This motor has a gearbox in front of it and has an encoder at the back, which causes the inaccuracies in the encoder readings to be less effective on the motor torque or position.

This motor is a gray box system model, where system models are called gray box when the theoretical structure is known, but the parameters are unknown. Gray box model systems are theoretical white box models because, with proper data, these system parameters can be identified, and this process is called system identification.

The system identification starts with data acquisition. To collect data from the three DC gear motors, a setup containing an NI myDAQ card, Arduino Mega, and L298N motor driver is made. With this setup, different PWM values are given to the motor, and their encoder data are collected from one channel. With the help of a myDAQ card used as a switch and data collector, the start of the motor is coordinated with the given input voltage. The encoder data is collected with a sampling rate of 50 kHz. All of the collected samples are gathered for 5 seconds. The PWM voltage is given in the first second and cut off at the exact third second. All of the encoder data collected is stored in an Excel file. This collected data must be processed to be used in system identification.

The theoretical structure of this system is most similar to the transfer function of a basic armature-controlled DC motor. The armature-controlled DC motor transfer function has the input as voltage and the output as angular velocity. System identification data collected from encoder channels need to be converted to motor angular velocities, and the input PWM should be converted to the equivalent voltages. These conversions are made in MATLAB using MATLAB scripts. The conversion for equivalent voltage of PWM is simple: multiplying the 12 V with the duty percentage works. The encoder data collected provides the rotation information, with each rising pulse counted, and 16 counts equals one rotation. With the time data collected, we can find the RPM values by dividing the total rotation counted over seconds. Disturbed encoder data is cleaned and filtered with a low-pass filter. The change of motor angular velocity and input voltage over time is given in Figure 9.

These data conversions are made for every five input voltage duty percentages, and encoder data are acquired from all three motors. When the conversions are done, the data are used in the MATLAB system identification toolbox. The transfer function is obtained using an iterative curve-fitting method.

The System Identification Toolbox in MATLAB requires the user to know the number of poles and

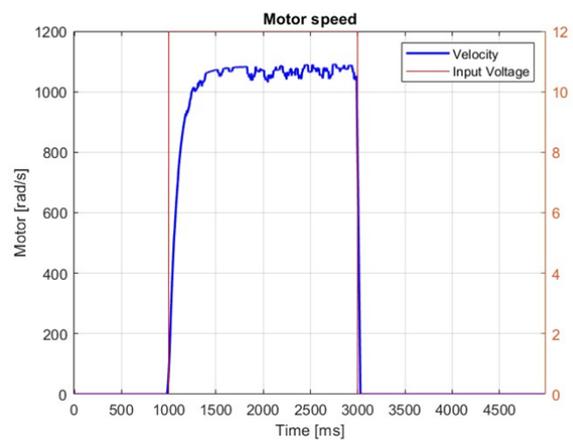


Figure 9. Input voltage and output velocity against time

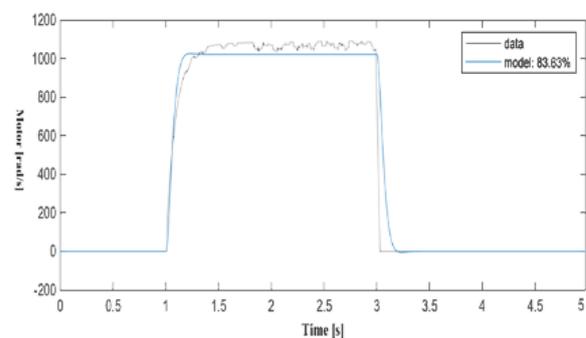


Figure 10. Curve fitting of the transfer function

zeros of the transfer function. The required pieces of information are known since the system is a gray box model, the system has two poles and no zeros because it works similarly to basic armature-controlled DC motors.

Figure 10 shows the curve fitting of the transfer function. The created transfer function's system response is %83.63 accurate according to the processed encoder data. The motor transfer function is known, which means we can use other MATLAB tools to find the optimal PID coefficients. Finding the PID coefficients can be done with another toolbox called PID Tuner. While tuning the PID, the control effort should be saturated. This saturation is necessary to apply these PID values in real-life applications and to see the same results in simulation. The control effort is the voltage input of this system, which means it can only be between 12 V and -12 V. The output can only be between 1050 rad/s and -1050 rad/s. These saturation values come from the specs of the components used. The sample time of the simulation must be the same as the sample time of the used hardware, and since the hardware is digital, a fixed-step solver should be used.

The motor transfer function is controlled using the tuned PID parameters in Figure 11-12. The control system reaches a steady state in 0.5 seconds. Steady-state error and overshoot of the system response are zero.

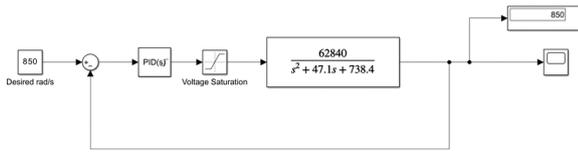


Figure 11. Tested motor transfer function and PID parameters

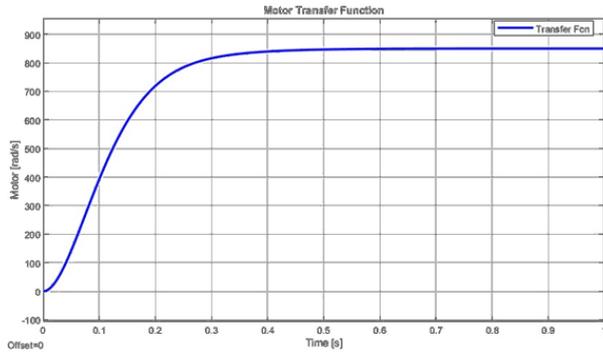


Figure 12. PID controlled motor transfer function

$$G(s) = \frac{62840}{s^2 + 47.1s + 738.4} \quad (49)$$

$$u(s) = P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \quad (50)$$

This method is applied to all three motors, and different PID parameters are tuned. Table 1 shows the PID parameters tuned for all three motors. Equation (50) describes the effect of PID control parameters on the control effort.

Table 1. Tuned PID Parameters

	P	I	D	N
Motor 1	0.231	0.358	0.007	15796.2
Motor 2	0.158	0.493	0.011	5750.9
Motor 3	0.166	0.535	0.012	6962.8

4. Simulation Procedure and Results

The proposed mobile robot is drawn in a 3D CAD program, and the mobile robot is exported to MATLAB Simulink using Simscape. Simscape is a Simulink add-on that can generate mechanical and electromechanical models inside a simulation world, working together with MATLAB. This tool allows this research to be tested in an ideal and known simulation setting. Thus, the allowed degrees of freedom of the world, the friction coefficient, and many other variables can be adjusted. Additionally, simulation results or any variable can be measured, viewed, and logged.

Before simulating the full towing robot, the trailer dynamics, dynamic tracking control, path planning, and path tracking are tested using a simplified Simscape model displayed on Figure 13-14 that includes

only the trailer portion of the mechanical system shown on Figure 19. This approach allows for a more manageable analysis by dividing the problem into two parts, facilitating the evaluation of the control methods and algorithms effectiveness [30].

The path generator subsystem shown in Figure 15 uses the pseudocode in Algorithm 1 and creates the path of the cubic Bézier function. The path is divided into waypoints and a lookahead distance is calculated. The waypoints are then used in the path tracking algorithm to calculate the desired velocities. The desired output velocities are then used in the dynamic tracking control subsystem shown in Figure 16. This subsystem uses the control equations and produces the desired external towing forces.

The Simscape Multibody Link Plugin exports the trailer’s geometrical and physical attributes to MATLAB Simulink from SolidWorks assembly and saves

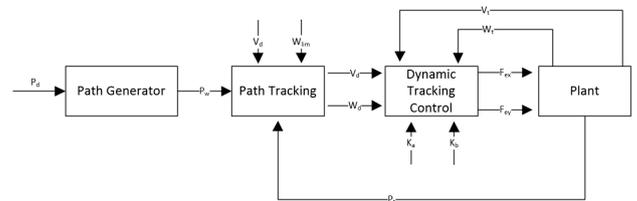


Figure 13. Simulation block diagram of trailer control

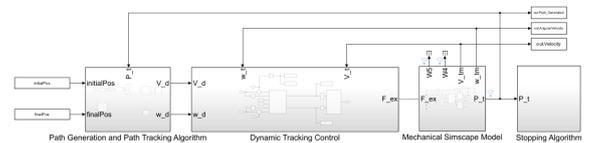


Figure 14. Trailer control simulation in MATLAB Simulink

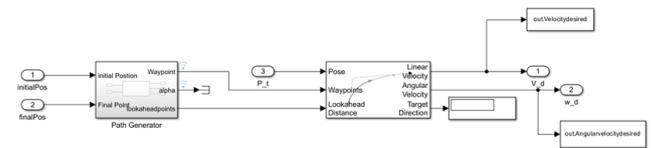


Figure 15. Subsystem of path generation and path tracking algorithm

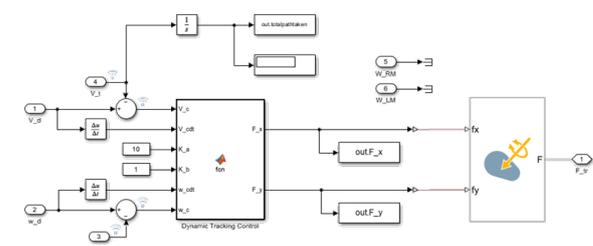


Figure 16. Dynamic tracking control subsystem

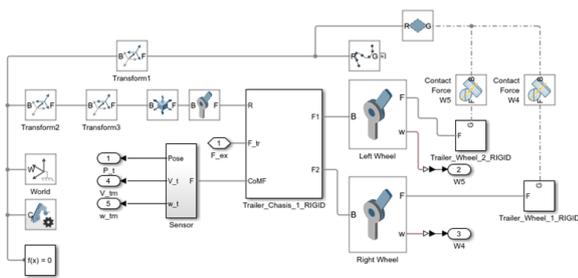


Figure 17. Subsystem of the trailer mechanical Simscape model

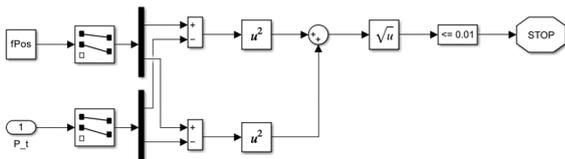


Figure 18. Stopping algorithm of the trailer

attributes in a MATLAB file, Simscape schematic in a Simulink file, and model appearance in STL or STEP format. After the export, some variables can be changed to adjust for the assumptions made for the convenience of the analysis in the first chapter. The Simscape mechanical model is created in an empty world when first exported. The movement plane, main body constraints with the world frame, wheel contact parameters, body frames, and sensors are added to the subsystem of the mechanical Simscape model shown in Figure 17. The generated path is shown in simulation Figures 23-24 with a spline. The frame created on the towing point is used for applying external towing forces, and the frame on the center of mass is used for position and velocity measurements.

The pure pursuit algorithm is insufficient when the mobile robot needs to stop. Therefore, the current and final positions of the trailer are used to create a stopping algorithm. The distance between the center of mass of the trailer and the final waypoint is calculated.

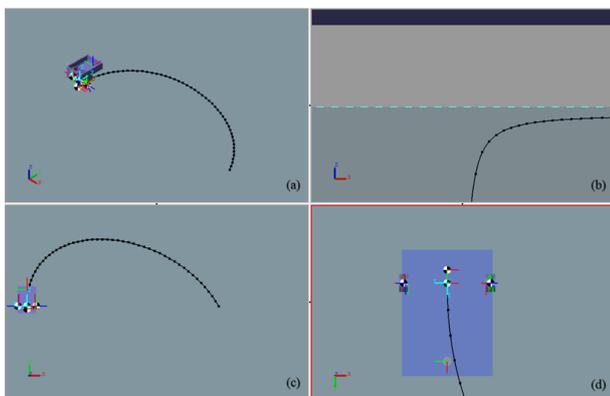


Figure 19. Simulation start, the trailer is at the beginning of the path (a) isometric view, (b) front view, (c) top view, (d) bottom view

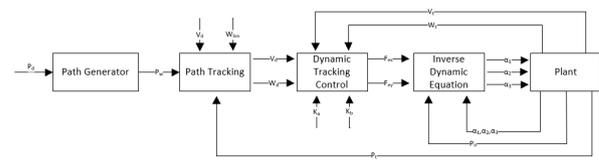


Figure 20. Simulation Block Diagram of the tractor robot

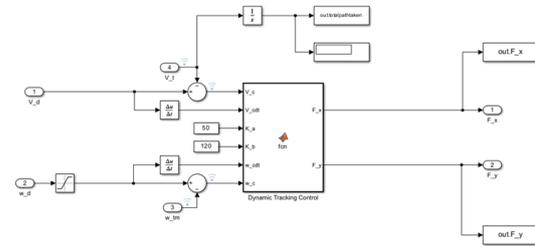


Figure 21. Subsystem of dynamic tracking control of trailer towed by omniwheel mobile robot

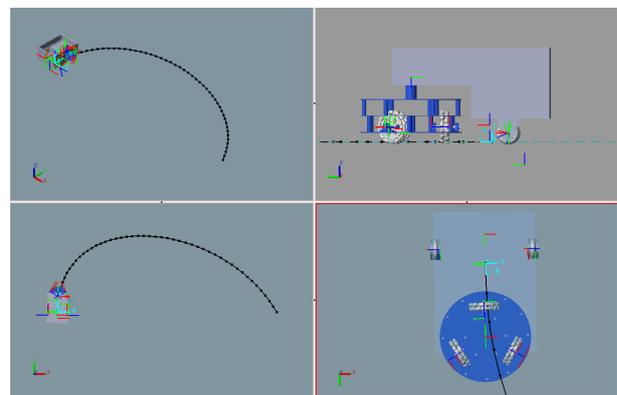


Figure 22. Simulation start, the mobile robot and trailer are at the beginning of the path (a) isometric view, (b) left view, (c) top view, (d) bottom view

When the trailer is within a radius of 0.01 m of the final position, the trailer stops moving and the simulation ends. Figure 19 illustrates the simulation settings from isometric, front, top, and bottom views.

After the trailer control simulation approves the success of the control methods and algorithms, the omniwheel mobile robot is added to the Simscape mechanical model, and the external towing forces are replaced with the inverse dynamics of the omniwheel mobile robot as shown in Figure 20. The stopping, path tracking, and path planning algorithms remain the same; therefore, the implementation of the mechanical model and inverse dynamic analysis is simpler.

Figure 21 illustrates the same method of dynamic tracking control used in Figure 16, but the overall dynamics of the system has changed, so the coefficients K_a and K_b are adjusted again.

The simulation is shown on Figure 22. The analysis of the simulations are done by investigating the graphs generated from the simulation data. The dynamic

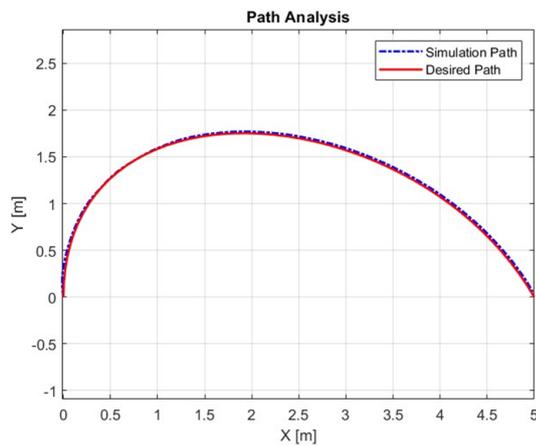


Figure 23. Bézier Curve path analysis with tuned parameters

tracking control parameters, path tracking parameters, implementation of selected motors to the simulation and PID tuning of the motors are done by analyzing the results. Analysis of the success of path tracking requires the simulation path to be generated in different forms such as square, circle and archimedean curved paths rather than using the path generation algorithm to generate Bézier curved paths.

4.1. Simulation Results for Tuned Dynamic Tracking Control Parameters

The tuning of the dynamic tracking control parameters (K_a, K_b) are done by analyzing the settling time, rise time, overshoot and steady state error data gained from simulations. The parameter values are kept when a desired result is reached. In this case the control parameters are $K_a = 200$ and $K_b = 300$.

Figure 23 demonstrates the path taken by trailer when the tuned dynamic tracking control parameters are used. The path taken by the trailer remained correct. The trailer control simulation has an RMS error of 29.7 mm, and the tractor robots simulation has a 3.23 mm RMS error. Although the error seems lower in the tractor robot simulation, the path taken by the trailer is closer to the desired path in the trailer control simulation. The lookahead distance in the tracking algorithm of both simulations was the same; therefore, the difference must be due to the dynamic tracking control coefficients. The reason for the error on waypoints 49 and 50 is that in both of the simulations, the trailer starts to shift its orientation irrelevant to the path planning when the lookahead distance radius cannot detect any more waypoints ahead. The path tracking algorithm needs to be adjusted so that the last point can be detected. The solution to the path tracking algorithm is to adjust the lookahead distance according to the stopping radius in the stopping algorithm.

The dynamic tracking control method successfully controls the trailer with the desired velocities in the trailer control simulation shown in Figures 24 and 25. The tractor robot simulation is controlled with the same dynamic tracking control method, but the

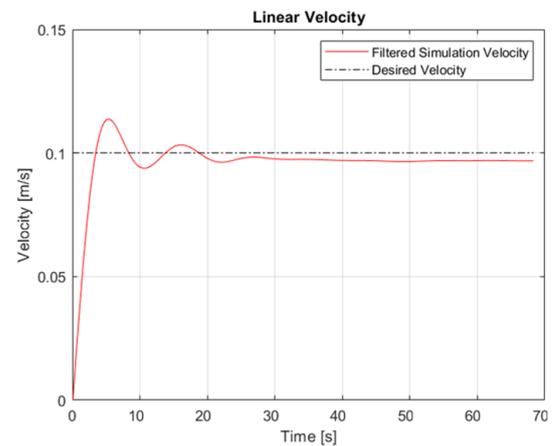


Figure 24. Linear velocity of the towed trailer

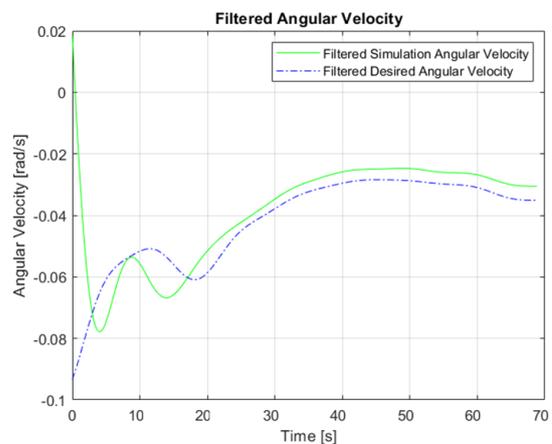


Figure 25. Angular velocity of towed trailer

results have high steady-state errors. This steady-state error can result from the hand-tuned dynamic tracking control parameters. The settling time of the trailer control simulation is faster than the tractor robot simulation. The trailer control simulation's linear velocity controls are slow and robust with no overshoot, while the tractor robot simulation's linear velocity control is fast and aggressive and has overshoots. The simulated angular velocities of both simulations have difficulty settling, and the desired angular velocity is always changing. The rapid change of the desired value causes the simulated values to have errors. The tractor robot requires a faster response to rapid changes, or another solution is to reduce the rate of rapid changes. Therefore, better tuning of the dynamic tracking control coefficients is needed to achieve faster settling times.

Figures 24-25 shows the measured and desired velocities of the trailer when simulation is started with the tuned dynamic tracking control parameters.

The use of cubic Bézier curves for path generation and the implementation of the path tracking algorithm both functions effectively. The resulting simulation path aligns closely with the desired trajectory and the trailer successfully reaches its target pose with minimal discrepancies. Both simulations achieve the same

desired pose by employing identical path planning and tracking algorithms. However, in the trailer control simulation, the trailer reaches the desired pose more quickly compared to the tractor robot's simulation. This difference arises from variations in the trailer's average speed; specifically, the slower average speed observed in the tractor robot is attributed to steady-state errors and overshoots within the control mechanism. As shown in Figure 23, at the coordinate [1, 1.6], the simulation of the tractor robot reveals that the trailer deviates from its intended path by 0.05 meters. This deviation persists until the target point is finally reached.

After tuning the dynamic tracking control coefficients, the simulations are repeated, and the desired linear velocity is reached with a 3.96% steady-state error, as shown in Figure 24, while the error occurring in the desired angular velocity is decreased (Figure 25). The tuned dynamic tracking control coefficients are used in other paths, and the trajectory control is analyzed.

4.2. Simulation Results for Different Paths

The success of the path tracking is further analyzed by generating non-optimal paths for path tracking in Simulink. The tuned dynamic tracking control parameters are used and the effect of lookahead distance is analyzed on a square path. The lookahead distance is initially 0.25 m, for the square path analysis two lookahead distances are examined, 0.75 m and 1.25 m.

Figures 26-28 are the result of simulations when the lookahead distance is 0.75 m.

The square path analysis in Figures 26-31 shows the importance of the lookahead distance. Errors on the track appear when a sharp turn is ahead and a higher lookahead distance is needed to track the path. The desired angular velocities change with the lookahead distance in both of the lookahead distances.

The desired velocities are achieved in Figures 27 and 30, therefore, the path error only happens because the desired angular velocities are incorrect. Figures 29-31 are the result of simulations when the lookahead distance is 1.25 m and the trailer is following a square path.

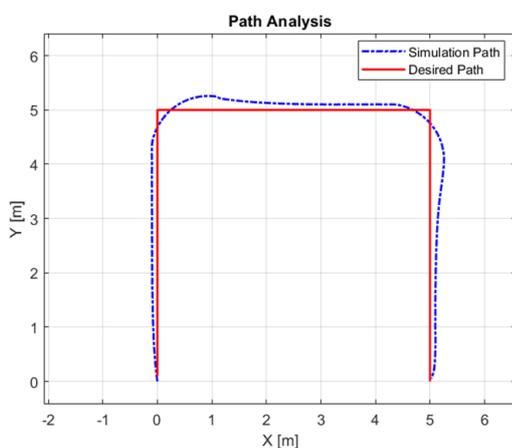


Figure 26. Square path analysis ($D=0.75$ m)

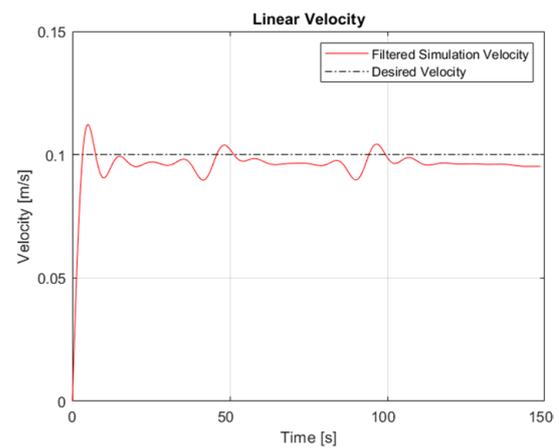


Figure 27. Linear velocity of towed trailer on square path

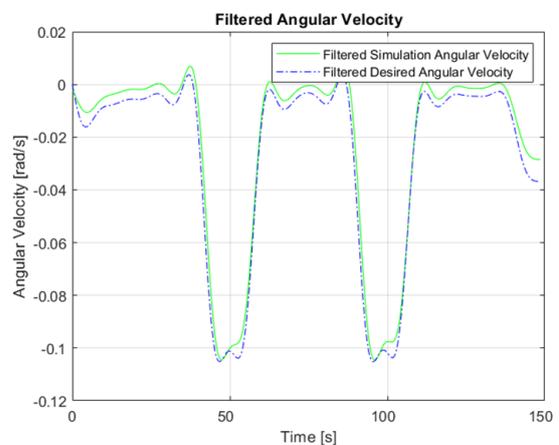


Figure 28. Angular velocity of towed trailer on square path

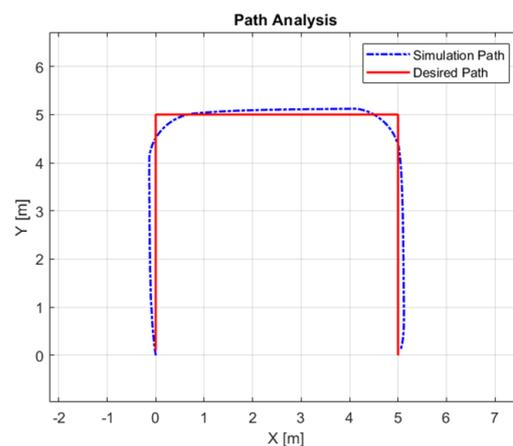


Figure 29. Square path analysis ($D=1.25$ m)

Circular path analysis uses the same tuned parameters and analyzes the behavior of the tracking algorithm in circular motion. Figure 32 shows the circular path taken by the towed trailer in simulation.

The circular path in Figure 32 is tracked with the same parameters as the Bézier curve path planning simulations. The circular path should have a fixed

desired angular velocity, but the small oscillations in Figure 34 are due to the low resolution of the path. The desired linear velocity is reached within an acceptable margin of error.

The circular path and the archimedean curve path are analyzed with the tuned dynamic tracking control parameters ($K_a = 200$ and $K_b = 300$.) and the initial lookahead distance (0.25 m).

The Archimedean spiral path in Figure 35 is tracked with an error at the beginning of the path. The curve at the beginning of the path is sharp, so the maximum value of the angular velocity is desired in Figure 37. It is observed that the limit angular velocity of 0.1 rad/s is not enough to take the turn without errors. Even with a larger curved turn, the trailer finds its track and reaches the desired velocity as shown in Figure 36.

4.3. Simulation Results for Implementation of PID Controlled DC Motors

The three motors selected for the towing robot are integrated in the simulation in this section. In the simulation the estimated transfer function for the three motors are used to simulate the electromechanical

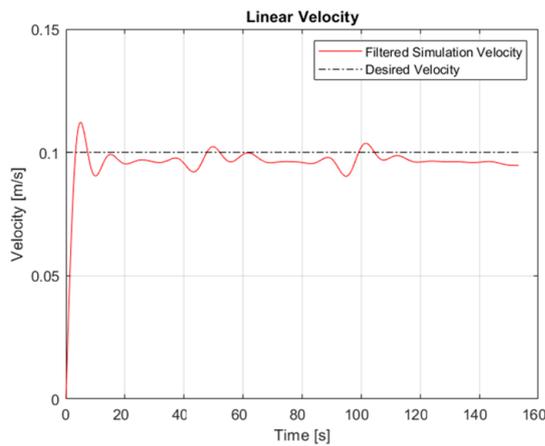


Figure 30. Linear velocity of towed trailer on square path

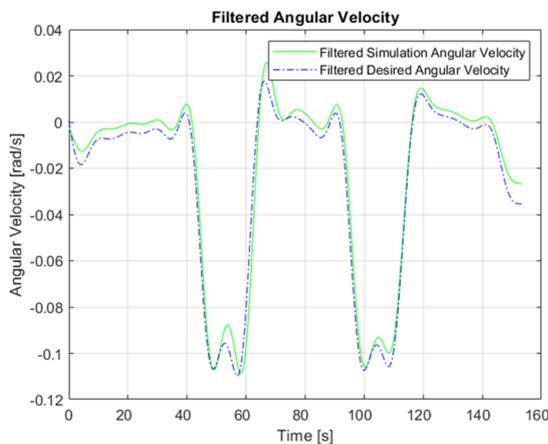


Figure 31. Angular velocity of towed trailer on square path

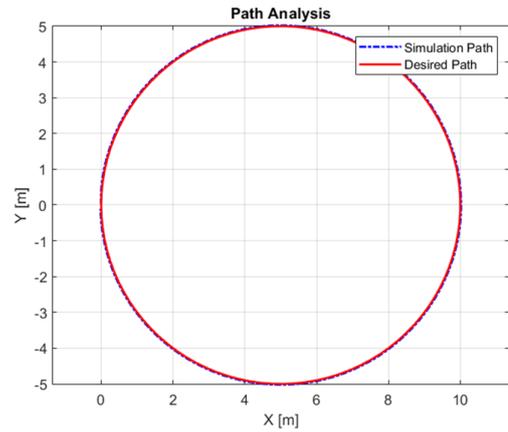


Figure 32. Circular path analysis

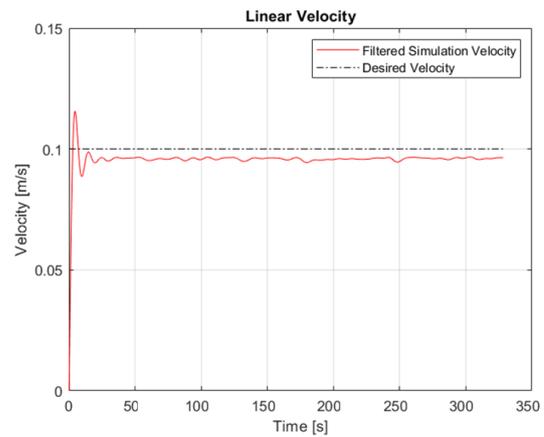


Figure 33. Linear velocity of towed trailer on circular path

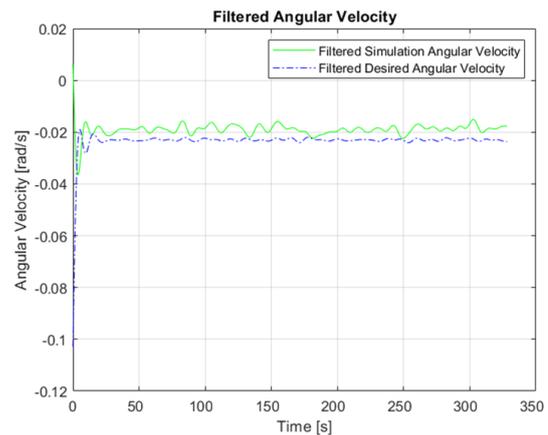


Figure 34. Angular velocity of towed trailer on circular path

effect motors. In order to simulate the electromechanical limits of the motor, stall torques are defined using saturation blocks. PID control of the motors can now be implemented in the simulation with the found PID parameters.

The implemented on Simulink subsystem is shown in Figure 39.

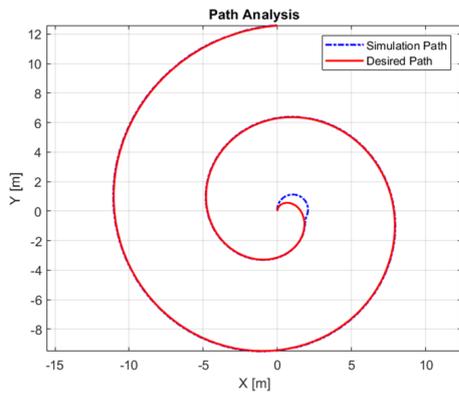


Figure 35. Archimedean spiral path analysis

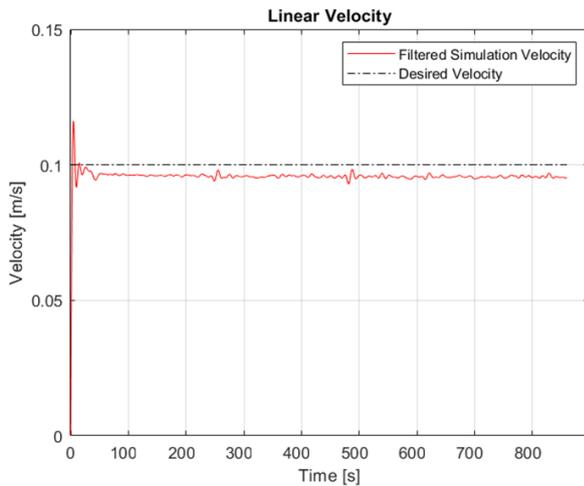


Figure 36. Linear velocity of towed trailer on Archimedean spiral path

The characteristics of the selected dc motors and the PID control parameters determined with tests are integrated into the simulation to apply a feasible simulation setting in real life application, as shown in Figures 38-39. The integration of the motors and PID control in the simulation caused the trailer to have small oscillations around the desired path shown in Figure 40. The linear and angular velocities of the trailer in Figures 41-42 are closer to the desired inputs when the PID control is applied.

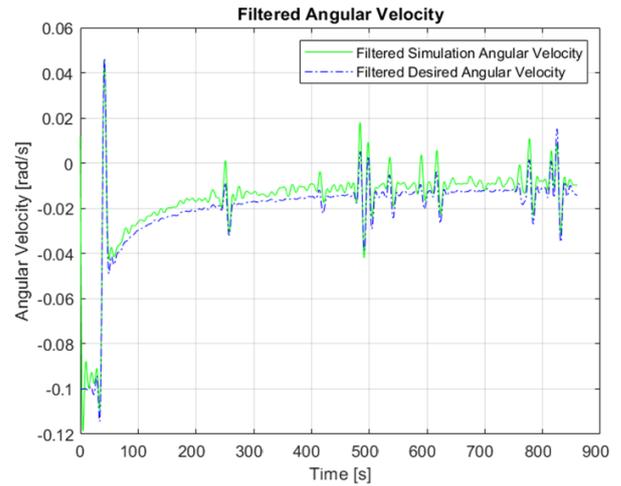


Figure 37. Simulated vs desired angular velocity of towed trailer on Archimedean spiral path

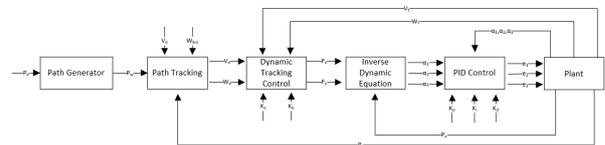


Figure 38. Simulation block diagram including PID control

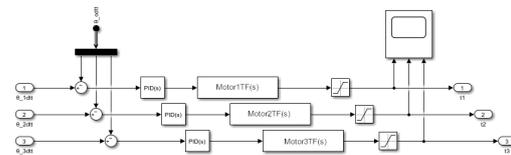


Figure 39. Simulating the selected motors

4.4. Comparison of Results

The performance results of the simulations are presented in Table 2 for each step of different tuning and implemented control system for tractor trailer mobile robot. The performance criteria for the simulation results are; steady state linear velocity error ($V_{ess},\%$) to analyze the closeness of the final linear velocity to the desired value the in order to tune the

Table 2. Performance Comparison of Non-Tuned, Tuned, and Motor Integrated PID Tuned Controllers

Metric	Non-Tuned	Tuned	Motor Integrated PID Tuned
Steady State Linear Velocity Error ($V_{ess},\%$)	13.29	6.26	4.32
Linear Velocity Overshoot (%)	8.68	17.48	17.17
Steady State Time (T_{ss},s)	N/A	N/A	61.9
Maximum Angular Velocity Error ($\phi_{emax},rad/s$)	0.104	0.092	0.117
Average Angular Velocity Error ($\phi_{eavg},rad/s$)	0.010	0.006	0.008
Execution Time (s)	74.7720	68.7573	67.2488
Tracking Error (Max, m)	0.0777	0.0755	0.0756
Tracking Error (RMS, m)	0.0425	0.0408	0.0404
Tracked Path Length (m)	6.620	6.620	6.630
Planned Path Length (m)	6.627	6.627	6.627

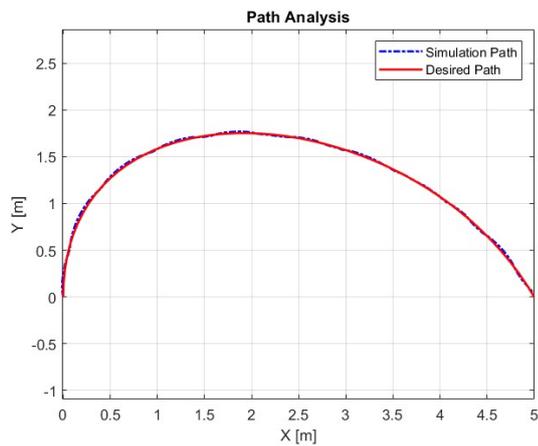


Figure 40. Bézier curve path analysis with implementation of PID controlled dc motors

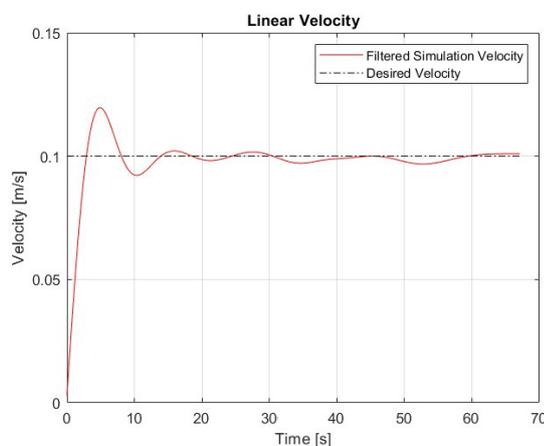


Figure 41. Linear velocity of towed trailer on Bézier curve path with implementation of PID controlled dc motors

controllers, linear velocity overshoot (%) to analyze the initial maximum linear velocity error of the mobile robot for the determination of success of the tuned

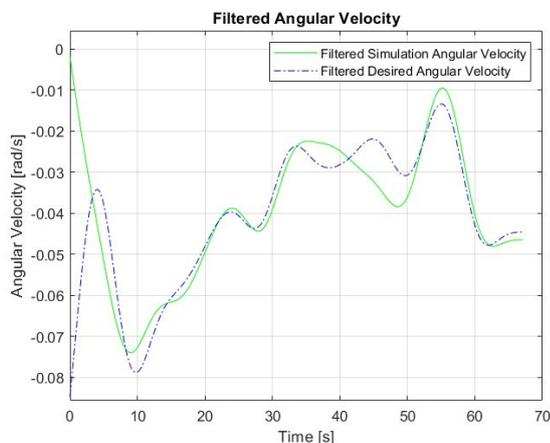


Figure 42. Angular velocity of towed trailer on Bézier curve path with implementation of PID controlled dc motors

controllers, steady state time ($T_{ss,s}$) in order to determine the time that the controller takes to be in the %5 range of the desired output, maximum and average angular velocity error ($\dot{\phi}_{\max}$, $\dot{\phi}_{\text{avg}}$, rad/s), execution time (s), tracking error (Max, RMS, m) and tracked and planned path length (m) to decide if the tracking control algorithms and path planning are effective. Each performance criteria has a purpose and effects the tuning of control parameter. All of the results in the Table 2 are simulated using the same path planning and path tracking algorithms to achieve the same pose from the same initial conditions and with equal conditions. The difference between non-tuned, tuned and motor integrated PID tuned results is, non-tuned results refer to the results when there is no PID controller, dynamic tracking parameters K_a is 50 and K_b is 120 and they are not tuned for integrated trailer tractor mobile robot, tuned results are from the Figures 23-25 when dynamic tracking parameters K_a is 200 and K_b is 300 without PID control and motor integration, the motor integrated PID tuned results are from the Figures 38-42 when the motors transfer function and the tuned PID controller is added to the control system, working together with the tuned dynamic tracking. According to these results comments about success of the methods can be made.

In order to compare the success of the simulated control method, results regarding control performance collected from references that uses different methods for path tracking and control of tractor-trailer mobile robots [21, 31, 32]. The methods such as adaptive control, backwards motion control, LQR (Linear Quadratic Regulator) algorithm, model predictive control and PID control are used to control the trajectory successfully. These methods are just as effective as the dynamic tracking control and PID control used in this study. To make this claim, the simulation and test results of these methods shown in different figures are compared and analyzed.

5. Conclusion

The simulation results show the trailer controlled by external forces on the joint successfully tracks the generated path. The simulation results indicate that the designed path planning algorithm for trailer dynamics is working as intended. Path planning can now be developed when the trailer detects obstacles and the path planning is adjusted according to the map data. The pure pursuit tracking algorithm works well with trailer dynamics. The algorithm should have a built-in stopper when the last waypoint is the next target destination stopper should activate. When the trailer starts to move away from the last waypoint, it means that the trailer has passed by, the stopper algorithm should activate and stop the trailer by force.

The trailer's dynamic tracking control is performed effectively and as for tractor robot, the control system is fast and accurate. Steady state error and settling time of the tractor robot can be improved by the better tuning of the dynamic tracking, PID and path

tracking control parameters. Oscillations of the simulated angular velocities can be weakened by further tuning of the dynamic tracking controller coefficients or by increasing the resolution of the path generation. The overall trajectory control of the omniwheel tractor robot performs successfully.

AUTHORS

Ünal Dana* – Izmir Kâtip Celebi University-Balatçık Mah., Izmir, 35620 Türkiye, e-mail: unal.dana@ikc.edu.tr, www.ikc.edu.tr.

Levent Çetin – Izmir Kâtip Celebi University-Balatçık Mah., Izmir, 35620 Türkiye, e-mail: levent.cetin@ikc.edu.tr, www.ikc.edu.tr.

*Corresponding author

ACKNOWLEDGEMENTS

This work was supported by Izmir Kâtip Çelebi University (project no:2024-TYL-FEBE-0008)

References

- [1] R. Raj and A. Kos “A comprehensive study of mobile robot: History, developments, applications, and future research perspectives” *Applied Sciences* vol. 12, no. 14 2022 doi:10.3390/app12146951.
- [2] R. J. Barros, J. L. Silva Filho, J. V. Neto, and T. P. Nascimento “An open-design warehouse mobile robot” *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE) 2020* pp. 1–6.
- [3] D. Jain and M. Y. Sharma. “Adoption of next generation robotics: A case study on amazon” 2017.
- [4] P. M. Kolahi and M. Nazemizadeh “Nonlinear dynamic modeling of tractor-trailer mobile robots with consideration of wheels inertia and their optimal point-to point path planning” *Meccanica* vol. 58, no. 1 2023 pp. 245–253 doi:10.1007/s11012-022-01578-6.
- [5] G. Tian, F. Zhou, and B. Song “Bézier curve based smooth path planning for mobile robot” *Journal of Information and Computational Science* vol. 8 2011.
- [6] R. C. Coulter. “Implementation of the pure pursuit path tracking algorithm” 1992.
- [7] M. Sampei, T. Tamura, T. Itoh, and M. Nakamichi “Path tracking control of trailer-like mobile robot” *Proceedings IROS'91: IEEE/RSJ International Workshop on Intelligent Robots and Systems* 1991 pp. 193–198.
- [8] Z. Hendzel and M. Rykała “Modelling of dynamics of a wheeled mobile robot with mecanum wheels with the use of lagrange equations of the second kind” *International Journal of Applied Mechanics and Engineering* vol. 22, no. 1 2017 pp. 81–99 doi:10.1515/ijame-2017-0005.
- [9] R. D. Ahmad Abu Hatab “Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework” *Advances in Robotics & Automation* vol. 02, no. 02 2013 doi:10.4172/2168-9695.1000107.
- [10] S. M. Yeşiloğlu, H. Temeltaş, and O. Kaynak “Dynamical modeling of two-wheeled cart: Cooperating manipulator approach” *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* vol. 3 2005 pp. 2008–2012 doi:10.1109/icsmc.2005.1571443.
- [11] C. A. Cáceres, J. M. Rosário, and D. Amaya “Design, simulation, and control of an omnidirectional mobile robot” *International Review of Mechanical Engineering* vol. 12, no. 4 2018 pp. 382–389 doi:10.15866/ireme.v12i4.13974.
- [12] J. David and P. Manivannan “Control of truck-trailer mobile robots: a survey” *Intelligent Service Robotics* vol. 7, no. 4 2014 pp. 245–258.
- [13] W. Chung, M. Park, K. Yoo, J. I. Roh, and J. Choi “Backward-motion control of a mobile robot with n passive off-hooked trailers” *Journal of Mechanical Science and Technology* vol. 25, no. 11 2011 pp. 2895–2905 doi:10.1007/s12206-011-0909-7.
- [14] F. Lamiroux and J.-P. Laumond “A practical approach to feedback control for a mobile robot with trailer” *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)* vol. 4 1998 pp. 3291–3296.
- [15] R. Fareh, M. Saad, S. Khadraoui, and T. Rabie “Lyapunov-based tracking control for nonholonomic wheeled mobile robot” *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* vol. 10, no. 8 2016 pp. 1021–1026.
- [16] S. Tzafestas *Introduction to mobile robot control* Elsevier 2014.
- [17] K. Watanabe, Y. Shiraishi, J. Tang, and S. G. Tzafestas “Autonomous control for an omnidirectional mobile robot with feedback control system” *Proceedings of the 1999 IEEE International Conference on Robotics and Automation* 1999.
- [18] A. Khadr “Multibody dynamics modeling and simulation of a three-wheeled mobile robot using a robotic approach” *International Journal of Intelligent Unmanned Systems* 2024.
- [19] G. Campion, G. Bastin, and B. Dandrea-Novel “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots” *IEEE Transactions on Robotics and Automation* vol. 12, no. 1 1996 pp. 47–62 doi:10.1109/70.481750.
- [20] A. B. Robat, K. Arezoo, K. Alipour, and B. Tarvirdizadeh “Dynamics modeling and

- path following controller of tractor-trailer-wheeled robots considering wheels slip." *ISA transactions* 2024.
- [21] N. T. Binh, N. A. Tung, D. P. Nam, and N. H. Quang "An adaptive backstepping trajectory tracking control of a tractor trailer wheeled mobile robot" *International Journal of Control, Automation and Systems* vol. 17 2019 pp. 465–473.
- [22] Z. Jin, C. Wang, D. Liang, S. Wang, and Z. Ding "Fixed-time consensus for multiple tractor-trailer vehicles with dynamics control: A distributed internal model approach" *IEEE Transactions on Intelligent Vehicles* vol. 9 2024 pp. 656–669.
- [23] J. Zhang, S. Singh, et al. "Loam: Lidar odometry and mapping in real-time." *Robotics: Science and systems (RSS)* 2014 pp. 1–9.
- [24] P. Marin-Plaza, A. Hussein, D. Martin, and A. d. l. Escalera "Global and local path planning study in a ros-based research platform for autonomous vehicles" *Journal of Advanced Transportation* vol. 2018, no. 1 2018 Art. no. 6392697.
- [25] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang "Path planning techniques for mobile robots: Review and prospect" *Expert Systems with Applications* vol. 227 2023 Art. no. 120254.
- [26] P. Raja and S. Pugazhenthii "Optimal path planning of mobile robots: A review" *International journal of physical sciences* vol. 7, no. 9 2012 pp. 1314–1320.
- [27] Z. Tang and H. Ma "An overview of path planning algorithms" *IOP conference series: earth and environmental science* vol. 804, no. 2 2021 Art. no. 022024.
- [28] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao "Review of autonomous path planning algorithms for mobile robots" *Drones* vol. 7, no. 3 2023 211.
- [29] S. R. Buss *Bézier Curves* Cambridge University Press 2003 pp. 155–199.
- [30] Ü. Dana and L. Çetin. "Omni-wheel tractor robot". Master's thesis *Izmir Katip Celebi University* İzmir, Turkey 2024.
- [31] G. Lei and Y. Zheng "Research on cooperative trajectory planning algorithm based on tractor-trailer wheeled robot" *IEEE Access* vol. 10 2021 pp. 64209–64221.
- [32] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys "Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trailer system" *IEEE/ASME Transactions on Mechatronics* vol. 20, no. 1 2014 pp. 447–456.