Modeling Path Trajectories and Object Avoidance for a Differential Drivetrain Robot

Aaron Omadutt

May 3, 2024

1 Introduction

Invented in 1827 by Frenchman Onésiphore Pecqueur, the automobile differential is a mechanical device that has remained critical in the development of wheeled vehicles and their ability to maneuver around curves. A differential is a mechanism typically configured using a series of gears with the goal of controlling the speeds of a vehicle's rear and/or front wheels. This functionality is essential, particularly during turns, where it ensures that the outside wheel rotates faster than the inside one to cover a greater distance in the same amount of time, maintaining stability and control.

In the field of automotive robotics, modelling path trajectories and object avoidance are crucial in ensuring a robot's ability to efficiently fulfill its assigned tasks. Efficient path trajectories enable robots to navigate from one point to another with minimal energy consumption and travel time, while effective object avoidance strategies ensure safe operation and prevent damage to the robot or its surroundings. These methodologies lay the foundation for reliable performance and seamless integration into various automotive applications.

This paper utilizes the works of Lynch et al. "Modern Robotics: Mechanics, Planning, and Control" [2], Szczepanski et al. "Energy Efficient Local Path Planning Algorithm Based on Predictive Artificial Potential Field" [4] and Knights et al. "Obstacles Avoidance Algorithm for Mobile Robots, Using the Potential Fields Method"[1] to model the path trajectories for a differential drivetrain robot around objects, as well as model optimized routes for the robot to navigate. The primary mathematical techniques discussed and employed throughout the paper encompass kinematics, providing mathematical descriptions of robot motion, and differential equations, for modeling dynamics and responses to external forces.

The purpose of this model is to enable precise path-following capabilities for robots. This would prove useful in various scenarios, such as navigating predefined routes or avoiding obstacles. Understanding the dynamics of movement specific to this drivetrain is relevant for assessing its performance compared to other wheeled or tracked systems in different scenarios, thereby highlighting their respective strengths and limitations.

2 Model and Methods

2.1 Mathematical Model of a Differential Drive Robot

By employing the figures and equations established in Matlab's MathWorks differential drive kinematics library[3] derived from Lynch et al's study [2], the Mathematical Model for the general movement of a differential drive robot can be defined by its primary kinematic equations.

Kinematics, in robotics, is the understanding of how a robot's joints are connected and how they move in relation to each other. It involves using geometry to study the motion of the various parts that make up a robot. This includes considering how each part can rotate or translate independently.

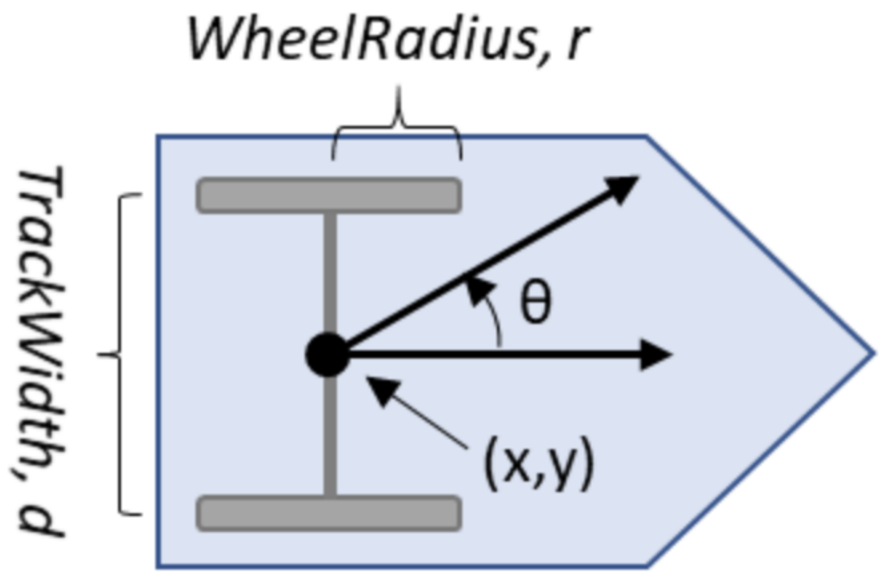

Figure 1: Differential Drive Kinematic Diagram, image reproduced from Matlab's Differential Drive Library [3]

The kinematic equations used to describe motion of a robot were modeled using Figure 1. which describes the vehicles general movements. This specific model approximates movements for a robot with a single fixed axle and wheels separated by a specified track width. The following equations build upon the layout of Figure 1. laying out the mathematical groundwork that describe the kinematic motion of the robot.

The state of the vehicle, it's current position and orientation, can be defined as the following three-element vector:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

where the variables are defined as the following:

$$\begin{aligned} x &= \text{x-component of global position (meters)} \\ y &= \text{y-component of global position (meters)} \\ \theta &= \text{ vehicle heading (orientation of the vehicle) (radians)} \end{aligned} \tag{1}$$

The units for this model are not constrained to meters; they can be applied to any other similar metric, as long as the entire system, including both kinematics and environmental parameters, is defined using the same units.

The speeds of the wheels can be defined using the rate of change in the state vector as well as the robots wheel radius and track width:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ -\frac{r}{d} & \frac{r}{d} \end{bmatrix} \begin{bmatrix} \dot{\phi}_L \\ \dot{\phi}_R \end{bmatrix} \tag{2}$$

Where the parameters follow the same format as Figure 1. and are defined as follows:

$$\begin{aligned} r &= \text{wheel radius} \\ d &= \text{track width} \\ \dot{\phi}_L &= \text{left wheel speed} \\ \dot{\phi}_R &= \text{right wheel speed} \end{aligned} \tag{3}$$

The generalized format, where the inputs to the system is given, can be defined as:

$$\begin{aligned} v \text{ (vehicle speed)} &= \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L) \\ w \text{ (angular turning velocity)} &= \frac{r}{d}(\dot{\phi}_R - \dot{\phi}_L) \end{aligned} \tag{4}$$

Using the generalized format, the wheel speed state vector can be rewritten as a system of simpler ODEs:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{5}$$

2.2 Mathematical Model for Potential Field Path Planning

In addition to the mathematical model governing the robot's movements, integrating a guidance and control system alongside a trajectory path planning function is essential for effective navigation within specific spaces.

The robot's guidance and control system should facilitate accurate trajectory following, which can be predetermined or computed in real-time. Concurrently, trajectory planning serves the fundamental purpose of enabling the robot to assess its surroundings and execute a series of calculations to determine the feasibility of reaching its intended location.

This process involves representing the robot as a point within the space and the environment as a potential field. The potential field is an array of vectors, each representing forces within the space. Each vector is defined by its magnitude and direction.

Two types of potential fields contribute to the overall environment mapping: attractive and repulsive. An attractive potential field facilitates navigation to desired locations, while a repulsive field generates barriers around obstacles to prevent collisions.

Utilizing the mathematical models of potential fields derived in both Knights et al. [1] and Szczepanski et al. [4], the attractive and repulsive potential fields can begin being modeled by calculating the distance (in meters) between the robot and its goal ($\mathrm{D_G}$) and concurrently its obstacles ($\mathrm{D_O}$). Additionally, the angle (θ) between the robot and its goal is also necessary to ensure the robots orientation is correct.

$$\begin{aligned} D_G &= \sqrt{(x_G - x)^2 + (y_G - y)^2} \\ D_O &= \sqrt{(x_O - x)^2 + (y_O - y)^2} \\ \theta &= \tan^{-1}\left(\frac{y_G - y}{x_G - x}\right) \end{aligned} \tag{6}$$

Additionally, we can define both the attractive and repulsive potential fields using the following functions denoted as U_{att} and U_{rep} for the attractive and repulsive fields, respectively. Additionally, there are two other parameters that influence how the potential fields will interact with the robot as it navigates its path, d^* and Q^*.

d^* defines the distance at which the repulsive force starts to influence the robot's movement. If the distance to an obstacle is less than d^*, the repulsive force becomes significant. Additionally, Q^* is a constant value that scales the repulsive force, determining how quickly the repulsive force increases as the robot gets closer to an obstacle.

Additionally, ζ and η are the parameters that control the attractive and repulsive force to the goal and away from objects respectively.

For $D_{\mathrm{O}} \leq d^*$, the attractive potential field is defined as :

$$U_{\text{att}} = \zeta \left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{\text{goal}} \\ y_{\text{goal}} \end{bmatrix} \right\| \tag{7}$$

Where the $\| \begin{bmatrix} x \\ y \end{bmatrix} \|$ notation represents the norm of the vector.

Similarly, for $D_\mathrm{O} > d^*$:

$$U_\mathrm{att} = \| \frac{d^*}{| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_\mathrm{goal} \\ y_\mathrm{goal} \end{bmatrix}} \| \zeta \| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_\mathrm{goal} \\ y_\mathrm{goal} \end{bmatrix} \| \tag{8}$$

The repulsive potential field can be defined as the following function:

$$U_\mathrm{rep} = (\eta \, (\frac{1}{Q^*} - \frac{1}{D_O})(\frac{1}{D_O}))(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_\mathrm{object} \\ y_\mathrm{object} \end{bmatrix}) \tag{9}$$

It is important to note that since the attractive, U_att, and repulsive, U_rep, potential fields are forces, they have the units of Newtons in this particular model.

Finally, the mathematical model describing the motion of the robot, the guidance and control system and the trajectory path planning functions can now be combined which will allow us to model the desired and actual trajectories of the robots movements around obstacles.

Several parameters will be analyzed, including chassis size and wheel radius, to determine their effects on the robot's movements around obstacles.

The utilization of MATLAB will serve as the primary computational tool for simulating the model with varying parameters. Various potential fields will be generated to assess the robot's movement across diverse environments, followed by an analysis of the robot's path based on its specified parametric inputs

3 Results

Using MATLAB, an attractive potential field was constructed to serve as the endpoint/ target goal for the robot to navigate towards, while four circular objects were created an placed within the enviroment, each possessing a repulsive potential field so the robot would navigate around them. The code for this portion of the results section was adapted from Szczepanski, R. et al.[4]. To review, the parameters being utilized were:

$$\begin{aligned} r &= \text{wheel radius (meters).} \\ d &= \text{track width (meters)} \\ \zeta &= \text{controls the attractive force towards the goal. (Newton/meters)} \\ \eta &= \text{controls the repulsive force from obstacles. (Newton/meters)} \\ d^* &= \text{defines the distance at which the repulsive force (meters)} \\ &\quad \text{starts to influence the robot's movement.} \\ Q^* &= \text{constant value that scales the repulsive force. (unit-less)} \end{aligned} \tag{10}$$

Three different models were simulated, each with the same attractive and repulsive potential parameters, but differing wheel radii and track widths. The values for the repulsive and attractive potential field were:

$$\begin{aligned} \zeta &= 1.1547 \text{ (Newton/meters)} \\ \eta &= 0.0732 \text{ (Newton/meters)} \\ d^* &= 0.3 \text{ (meters)} \\ Q^* &= 0.75 \end{aligned} \tag{11}$$

The configuration for each robot were as such:

$$\begin{aligned} \text{blue robot} &: r = 5 \text{ meters}, d = 5 \text{ meters} \\ \text{green robot} &: r = 10 \text{ meters}, d = 2 \text{ meters} \\ \text{red robot} &: r = 10 \text{ meters}, d = 10 \text{ meters} \end{aligned} \tag{12}$$

The path trajectories for the three different models can be seen in Figure 2. alongside how many time steps it took for each respective robot to reach its desired endpoint. The time steps in this model represent how many loop iterations the robot took before it reached its goal.

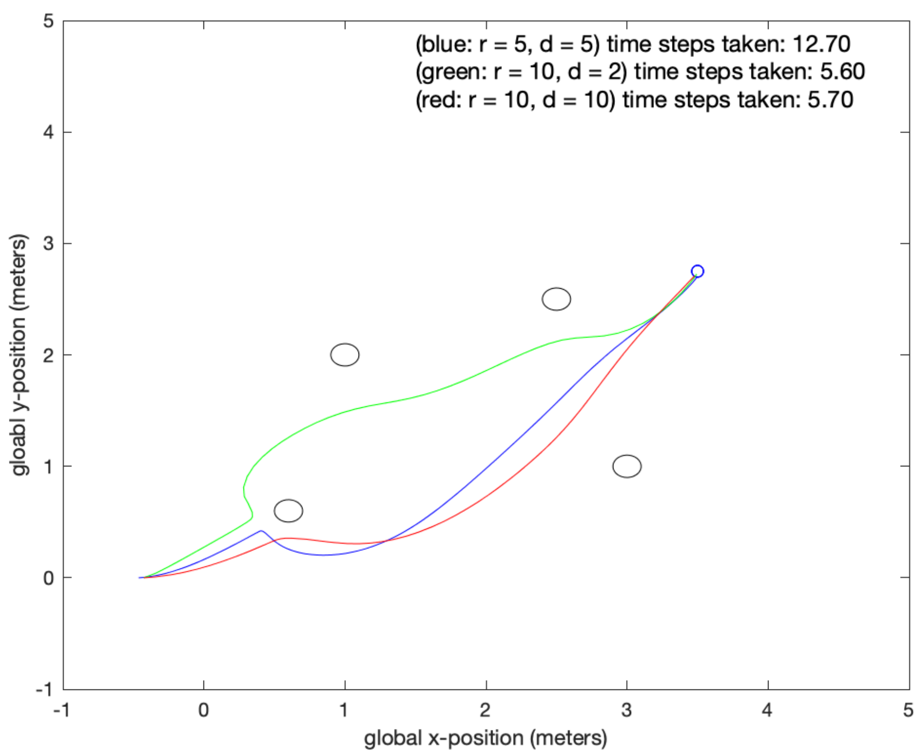

Figure 2: Overlay of three different model configurations and paths where the parameters are blue: r=5, d=5, green: r=10, d=2, and red: r=10, d=10

Additionally, the differential drive robot's path trajectory was modelled against a predetermined route, to visually grasp how much the robot deviates from its desired path. The code to generate this particular model was adapted from Matlab's "DifferentialDrive" library[3] which is based on a study by Lynch et al. [2]."

To model this, a desired path was created in MATLAB using a series of way-points represented as vectors, seen in Figure 3. It is important to note that this model was generated using a track-width of 0.2 and wheel radius of 0.05. The units for this particular model are ambiguous and can be any consistent unit as long as it matches the global coordinates.

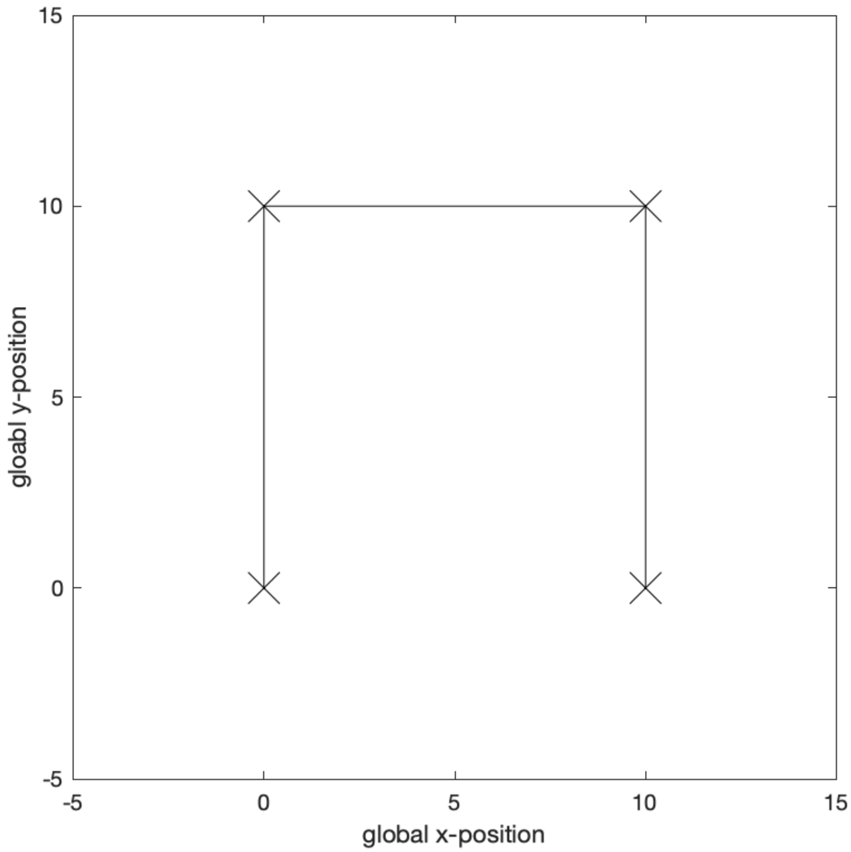

Figure 3: Desired Waypoint Path Trajectory

These way-points were then fed into a pure pursuit function, which calculated a point ahead of the vehicle on the path known as the pursuit point and steered the vehicle towards it. This approach allowed the robot to track the path, making real-time steering adjustments as needed.

Within Figure 4. we can see how the robot attempts to follow the desired path but is constrained by its kinematics in how it moves.

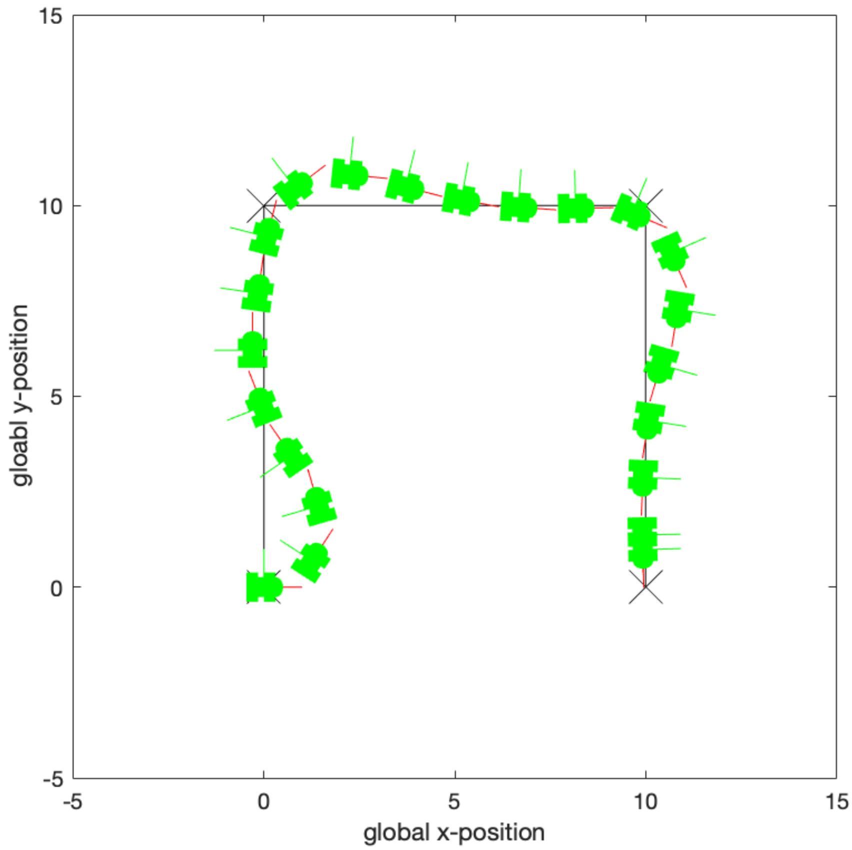

Figure 4: Desired Waypoint Path Trajectory

4 Discussion/Conclusion

From the tests conducted, it is generally observed that a smaller wheel radius tends to decrease the amount of time needed for the robot to reach its desired path. The smaller wheel radius results in more wheel rotations needed to travel to the endpoint, which in turn results in the robot taking a longer time to reach said location. This is seen in Figure 2 where the difference in time steps taken between the blue and red paths is apparent. The blue robot, with a smaller wheel radius of 5 meters, takes 14.20 seconds to reach the endpoint, while the red robot, with a wheel radius of 10 meters, reaches the endpoint in 6.80 time steps.

Additionally, when the track width is either sufficiently small or large enough, it can influence the robot's path around an object, as seen by the differing robot trajectories in Figure 2. Furthermore, a smaller track width allows the robot to make sharper turns when avoiding obstacles. This is evident in the comparison of path trajectories between the red and blue robots, where the blue robot with a smaller track width navigates around the objects in sharper turns.

In the pure pursuit model in which the robot navigates along a series of way-points, we observe that initially, when the robot begins perpendicular to the path, it deviates significantly before realigning with the path itself. However, once aligned, the robot maneuvers quite closely to the desired path. This behavior could be further refined within the model by incorporating a function that can independently increase or decrease one wheel speed at the intersection of the waypoint edges, allowing the robot to rotate in place. This capability would enable the robot to follow a set of linear paths with minimal deviation.

Overall, the implementations of these models are useful for observing how a differential drivetrain robot behaves when trying to reach a desired endpoint or follow a specified path. However, these models are very rudimentary in how they model the robot's path trajectories. The robot's movement towards the endpoint in this specific model is linear, as it moves in a straight line towards the goal and only deviates if it encounters an obstacle. However, by implementing a more sophisticated path planning feature into the model, the robot's path trajectories could be further optimized, reducing the number of time steps needed to reach the desired endpoint.

5 Acknowledgements

I would like to thank my peer reviewers, Ethan J. and Kenneth C., for their feedback on this project. Some changes I made based on conversations with them were the consolidation of my results into a single figure to model different path trajectories based on the differing parameters. Another change was the placement of my repulsive field obstacles to generate more interesting paths. I would also like to thank Professor J. Goldwyn for continual advice and support throughout the project.

References

[1] Vesna Antoska Knights, Zoran Gacovski, and Stojce Deskovski. Obstacles avoidance algorithm for mobile robots, using the potential fields method. *Universal Journal of Electrical and Electronic Engineering*, 5:75–84, 12 2017.

[2] Kevin M Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control.* Cambridge Univeristy Press, 2017.

[3] MathWorks. Differentialdrivekinematics, Accessed: 2024.

[4] Rafal Szczepanski et al. Energy efficient local path planning algorithm based on predictive artificial potential field. *IEEE Access*, 10:39729–39742, 2022.