Autonomous Systems Assignment on Robot Controllers

Yash Chaudhary
12403014
Technische Hochschule Deggendorf
Cham, Germany
yash.chaudhary@stud.th-deg.de

Aakash Vanmali
12402734
Technische Hochschule Deggendorf
Cham, Germany
aakash.vanmali@stud.th-deg.de

Amin Yazdani
12404412
Technische Hochschule Deggendorf
Cham, Germany
amin.yazdani@stud.th-deg.de

***Abstract*—The focus of this paper is to design and implement various controllers for the TurtleBot 3, both in Gazebo environment and for real world use with a real TurtleBot 3. The controllers designed include a nominal controller, a Stanley controller, and Model Predictive Controllers (MPC) with and without avoidance capabilities. The goal is to evaluate and compare the performance and robustness of these controllers in autonomous navigation tasks. This work contributes to the understanding of control strategies in autonomous systems, offering practical insights into their applications and effectiveness.**

***Index Terms*—TurtleBot3, Gazebo simulation, stanley controller, nominal controller, modal predictive controller**

I. Introduction and Motivation

Technological progress is driven mainly by the major breakthroughs in robotics and autonomous system technologies. These technologies have enabled machines to perform tasks autonomously and without the need for frequent human interaction. A human can create a program for a machine which will allow it to perform all the tasks expected with high precision without the need for the human programmer to ever look at it again. These technologies are finding increasing popularity in the fields like manufacturing, healthcare, transportation, warehousing, assembly, etc. Another key aspect in this revolution is the advancements in robot control, which is a subset of autonomous systems, involving programming machines to perform specific tasks with precision and adaptability. The integration of autonomy and control in robotics not only enhances efficiency and productivity but also opens new possibilities for complex problem-solving and improved quality of life. With this study, the aim is to design different controllers for control of Turtle bot 3 in the Gazebo environment and test the code with the control of a real turtle bot. Several controllers will be used and their results would be compared. The conclusion would be drawn based on the results of each controller and their efficiency based on the simulation and real-world application.

II. Systems

A. *Differential Drive Mobile Robot*

Differential drive mobile robots are a popular class of robots characterized by their simple yet effective locomotion mechanism. These robots are equipped with two independently driven wheels mounted on either side of the robot's body, allowing for a range of movement patterns. The differential drive mechanism operates by varying the relative speeds of the two wheels. When both wheels rotate at the same speed and direction, the robot moves in a straight line. By adjusting the speed and direction of each wheel, the robot can turn left or right, rotate in place, or follow curved paths. This ability to manoeuvre with precision makes differential drive robots highly versatile and suitable for navigating complex environments.

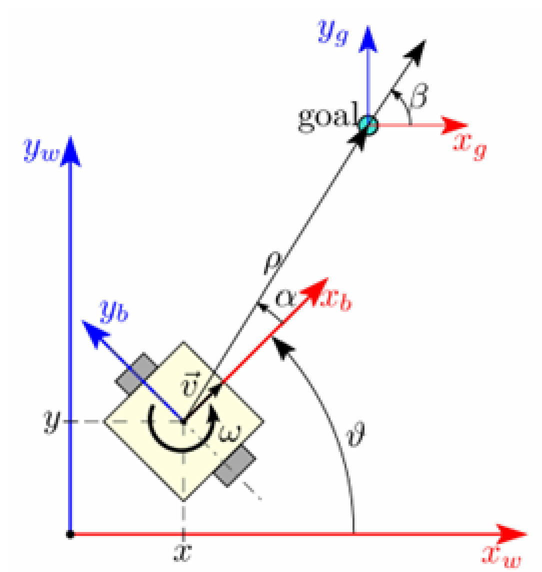

Fig. 1. Differential Drive Mobile Robot Frame

The kinematics of a differential drive mobile robot described in the inertial frame $\{x_b, y_b, \theta\}$ is given by -

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \theta \end{bmatrix} = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

where,
$\dot{x}$ and $\dot{y}$ are the linear velocities in the direction of x_w and y_w of the world frame.
Let α denote the angle between x_b axis of the robot reference frame and the vector connecting the centre of the axle of the

wheels with the final position.
Converting these to polar coordinate frame with its origin at goal position, we get –

$$\rho = \sqrt{\delta x^2 + \delta y^2}$$
$$\alpha = -\theta + atan2(\delta y, \delta x)$$
$$\beta = -\theta - \alpha$$

It can be shown that, $v = k_\rho, \omega = k_\omega \alpha + k_\beta \beta$

Thus, we get,

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho cos(\alpha) \\ k_\rho sin(\alpha) - k_\alpha \alpha - k_\beta \beta \\ -k_\rho sin(\alpha) \end{bmatrix}$$

This will drive the robot to $(\rho \quad \alpha \quad \beta) \rightarrow (0 \quad 0 \quad 0)$
Here k_ρ, k_β and k_α are constants which are tuned in the control to provide the correct parameters for the robot control. A recommended range for these is –
$k = (k_\rho, k_\alpha, k_\beta = (3, 8, -1.5)$
Local exponential stability -
$k_\rho > 0, k_\beta < 0, k_\alpha - k_\rho > 0$

B. Car Like Mobile Robot

Car-like mobile robots, also known as Ackermann-steered robots, are inspired by the mechanics of conventional automobiles. These robots are characterized by their steering mechanism, which is designed to mimic the steering system of a car. This type of mobile robot is particularly useful in applications where maneuvering and navigation in large, dynamic environments are required, such as urban settings, warehouses, and agricultural fields. Car-like robots use an Ackermann steering geometry, where the front wheels are steered while the rear wheels are driven. This configuration allows the robot to execute smooth and precise turns, maintaining stability and control even at higher speeds. The steering mechanism ensures that the wheels follow a common center point, reducing slip and enabling efficient path tracking.

For car like mobile robot, to keep wheels aligned to path, the steering angles is considered equal to the heading error for all the iterations of the same, we take -

$$\theta_e = \theta - \theta_p$$

where,
θ_e = Heading error
θ = Heading of robot
θ_p = Heading of path

To eliminate crosstrack error, a proportional control is added whose gain is scaled inversely to forward velocity. This is posed as inverse tan, to map the range of steering angles. e_{fa} increases as the wheels are steered further towards the path. Therefore we get –

$$\delta(t) = \theta_e(t) + arctan(\frac{ke_fa(t)}{v(t)}$$

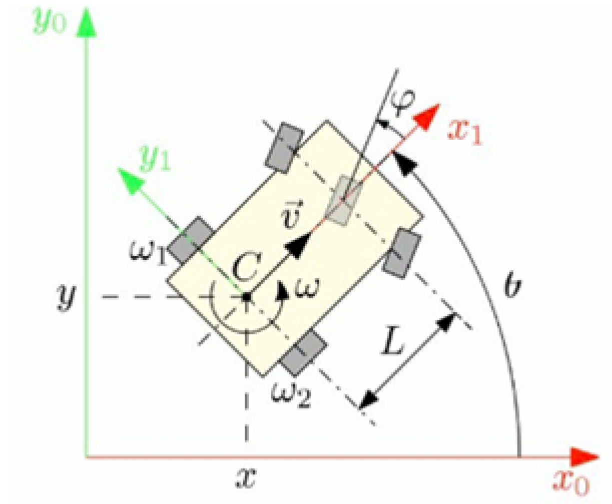

Fig. 2. Car Like Mobile Robot Frame

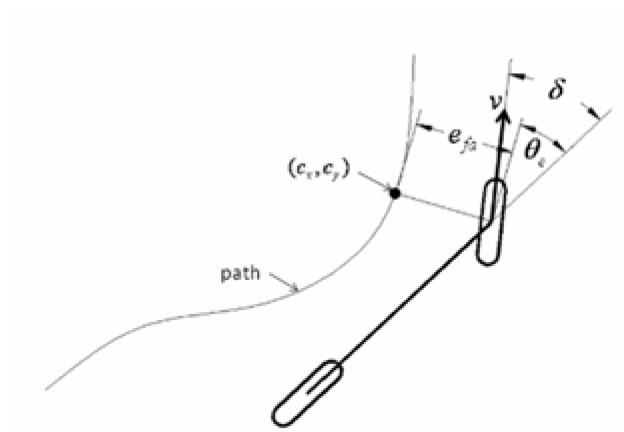

Fig. 3. Derived robot frame

where, k is the positive gain.

Thus, we have,
$x_f = x + Lcos(\theta)$
$y_f = y + Lsin(\theta)$

where,
$x_f and y_f$ = desired position of the vehicle
L = distance moved
x y and θ = current position and orientation

We can simplify crosstrack error as perpendicular distance and thus we get –

$$e_{fa}(t) = (y_{ref}(t) - y(t))cos\theta(t) - (x_{ref}(t) - x(t))sin\theta(t)$$
$$dist2goal = \sqrt{(y_{ref}(t) - y(t))^2 + (x_{ref}(t) - x(t))^2}$$
$$target_{point} = min \quad dist2goal$$

III. Appendix

A. Appendix A

- Yash Chaudhary - Differential Drive Mobile Robot
- Aakash Vanmali - Car like Mobile Robot
- Amin Yazdani - Introduction and Motivation

References

[1] Kevin M. Lynch and Frank C. Park. 2017. Modern Robotics: Mechanics, Planning, and Control (1st. ed.). Cambridge University Press, USA.

[2] Mark W. Spong, Seth Hutchinson, Mathukumalli Vidyasagar. 2005. Robot Modeling and Control (2nd. ed.)

[3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press.