Kinematic Control of a differential Drive Robot (Go-to-goal)

Martin Oghogho Jr

June 22, 2021

1 Introduction

Moving a robot is one of the basic methods of robot control. A go-to goal problem is simply having control of a robot and telling it to move from a particular point to the desired point. We want to be able to control our mobile robot such that it can move from one configuration to the desired point. Various methods have been developed to avoid obstacles. However, in this project we would focus more on the basics, that is on controlling the robot without any obstacles. The main aim is to tell our mobile robot to go where we want it to go.

For us to have control of our robot, a design process has to be considered. Firstly, we need to determine the physical specifications of our system, then understand how our system work by developing a mathematical model and finally developing a controller such that our robot goes to the desired position. A simple visual representation of the Pioneer 3DX is shown in Figure 1 below.

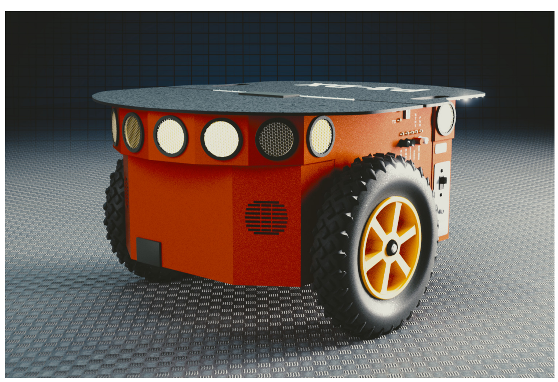

Figure 1: Pioneer 3DX Mobile Robot

2 Robot Description and Mathematical Model

Before determining the mathematical model of our system, we need to have some physical specifications as shown in Table 1. The implementation of this robot would be done in CoppeliaSim Software.

Component	value	unit
Weight	9	Kg
Max. Speed	1.2	mm
Wheel Dia.	1.95	mm
Wheel Dist (L)	3.81	mm

Table 1: Specifications of Pioneer 3dx

2.1 Kinematic model

In this section, a mathematical model of our system will be developed. Since we want to have control of the robot wheels, more emphasis will be laid on the kinematic model than the dynamic model of our system. A kinematic model is a model that is concerned with the mathematics of motion without taking into account the forces affecting it [1]. Now consider our differential drive mobile robot in 2D space at some point (x_r, y_r) as shown in Figure 2. If we rotate the robot by some angle θ, the robot will have a position and an orientation given as (x_r, y_r, θ_r) when expressed with respect to an inertial frame in a Cartesian coordinate. The mathematical expression of our robot based on its position and orientation in space is expressed as a transformation matrix given as follows.

$$\begin{aligned} \dot{q}_I &= R(\theta)\dot{q}_R \\ &= R(\theta) \cdot \begin{bmatrix} \dot{x}_r & \dot{y}_r & \dot{\theta}_r \end{bmatrix}^T \end{aligned} \tag{1}$$

where,

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Since we want to control the robot wheels independently, we need to understand how the robot moves in space. A forward motion will require the two wheels to rotate at the same time and vice-versa for a backward motion. The robot will rotate about its axis if one wheel is driving forward and the other is moving in the reverse direction at the same rate. However, the motion of the wheels is characterized by two constraints known as holonomic constraints. Firstly, there is no lateral motion, that is, the wheels can only move forward or backward, but not sideways. Therefore, velocity in the lateral axis is zero:

$$\dot{y}_r = 0 \tag{3}$$

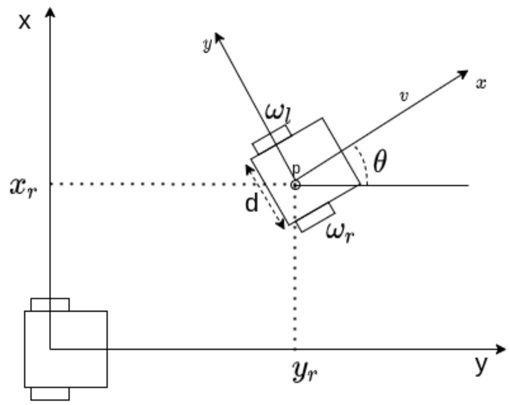

Figure 2: Simple motion representation of a mobile robot

From (1), if we take the inverse of the rotation matrix, the velocity of the inertial frames gives:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{4}$$

The other constraint is the pure rolling constraint which represents the fact that each wheels makes a contact with the ground at a point P as shown in Figure 3 [2].

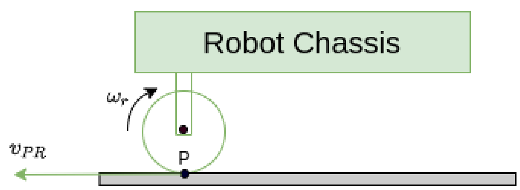

Figure 3: Pure rolling motion constraint

Since we are independently considering the speed of each wheel, thus we want to control the angular velocities (ω_R and ω_L) of both right and left wheels respectively. If one wheel spins and the other doesn't, then the robot will around the second wheel in a circular form. Then the linear speed and angular velocity of that wheel (taking the right wheel as an example) will be:

$$\begin{aligned} v_1 &= r\omega_R \\ \omega_1 &= \frac{r}{d}\omega_R \end{aligned} \tag{5}$$

Since p is the midpoint of the two wheels, therefore the wheel move half the speed as shown in (6)

$$v_x = \frac{r\omega_R}{2} \tag{6}$$

The same effect will be shown for the left wheel, however, the left wheel moves in a counterclockwise direction, thus it'll move in the negative direction only for the left wheel

$$\omega_2 = -\frac{r}{d}\omega_L \tag{7}$$

As explained above, the velocity in the lateral axis is zero ($v_y = 0$). Therefore, the linear and angular velocities of both wheels is obtained as follows:

$$\begin{aligned} v = v_x = v_1 + v_2 = \frac{r(\omega_R + \omega_L)}{2}; v_y = 0 \\ \omega = \omega_1 + \omega_2 = \frac{r(\omega_R - \omega_L)}{d} \end{aligned} \tag{8}$$

where r is the radius of the wheels and d is the axial distance between the wheels, $d = 2b$. Therefore, considering 1 and 8 we can derive our equation of motion with respect to the inertial frame as such:

$$\dot{q}_i = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{9}$$

A basic understanding has been established with respect to the motion of the robot. In this regard, we want the robot to move to a particular point in space as shown in Figure 4. Therefore, the position of the goal is described as shown:

$$\begin{aligned} x_g = x_r + r_{gr} \cdot \cos\theta \\ y_g = y_r + r_{gr} \cdot \sin\theta \end{aligned} \tag{10}$$

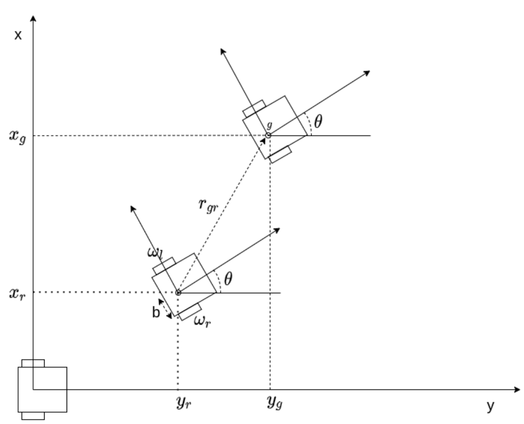

Figure 4: Movement of the robot to a final point in space

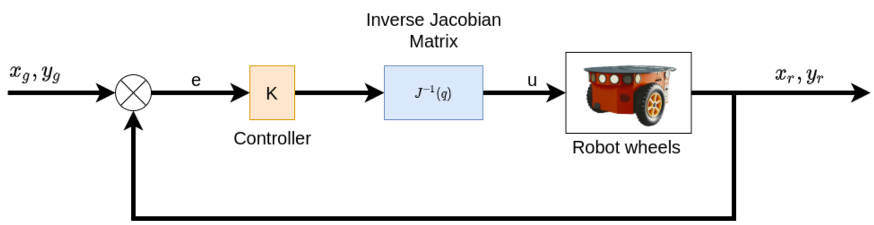

Figure 5: Movement of the robot to a final point in space

Given the block diagram shown in Figure 5, we want to compute a control law that will drive the wheels to the desired position. Therefore we need to compute the forward and inverse kinematics of our robot. The forward Kinematics can be computed from 9 as follows:

Substituting the parameters from 8 into 9, we get:

$$\begin{bmatrix} \dot{x}_p^i \\ \dot{y}_p^i \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r(\omega_R+\omega_L)}{2} \\ 0 \\ \frac{r(\omega_R-\omega_L)}{d} \end{bmatrix} \tag{11}$$

Evaluating 11 above, we obtain our forward Kinematics as shown below:

$$K_{fwd} = J(q) \cdot u \tag{12}$$

$$K_{fwd} = \frac{R}{2} \begin{bmatrix} \cos\theta + \frac{1}{b} \cdot \sin\theta & \cos\theta - \frac{1}{b} \cdot \sin\theta \\ \sin\theta - \frac{1}{b} \cdot \cos\theta & \sin\theta + \frac{1}{b} \cdot \cos\theta \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \tag{13}$$

where $J(q)$ is the Jacobian matrix of our system and u is the input to the robot wheels The two-dimensional representation shown above is enough to compute the angular velocities of the left and right wheels. From the block diagram, we are more concerned with the inverse kinematics of the system. Thus, it can be represented by taking the matrix inverse of $J(q)$ as shown:

$$J^{-1}(q) = \frac{1}{R} \begin{bmatrix} \cos\theta + b \cdot \sin\theta & \sin\theta - b \cdot \cos\theta \\ \cos\theta - b \cdot \sin\theta & \sin\theta + b \cdot \cos\theta \end{bmatrix} \tag{14}$$

Therefore, our control law is derived from the block diagram as shown:

$$u = J^{-1}(q) \cdot K \cdot e \tag{15}$$

$$u = J^{-1}(q) \cdot K(x_g - x_r) \tag{16}$$

We've been able to mathematically derive the input necessary to drive our robot to a desired destination. The code for this example is self-explanatory and is accessible on my GitHub page.

References

[1] "Kinematics," Jun 2021. [Online]. Available: https://en.wikipedia.org/wiki/Kinematics

[2] A. Awatef and B. H. Mouna, "Dynamic modeling and inverse dynamic control of mobile robot," in *2017 International Conference on Green Energy Conversion Systems (GECS)*, 2017, pp. 1–5.