WHEELED ROBOT KINEMATIC SIMULATION

By
Shashank Sinha

Table of Contents

CHAPTER 1

BASIC KINEMATIC SIMULATION OF LAND BASED ROBOTS

Need of a Mathematical Model

- To understand the behaviour of the system and design the system.
- To design suitable controllers, navigation systems and adjust their performances.
- To predict or estimate the system parameters, to illustrate or mimic or simulate the real system etc.

Relating the Robot Velocity and the Ground Frame

For a fundamental land-based mobile robot, three degrees of freedom are considered: two for translation within the plane of motion and one for rotation about the axis perpendicular to that plane. To accurately determine the absolute position of the robot, it is essential to establish a connection between the robot's velocity and a fixed ground inertial frame. **Figure 1** represents a general configuration of the robot WRT an inertial ground frame.

x_b and y_b are the coordinate axes in the robot frame B. x_i and y_i are the coordinate axes in the ground inertial frame.

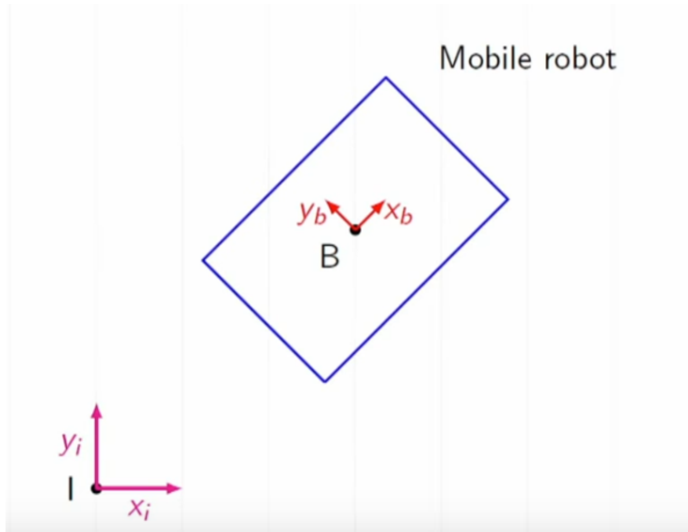

Figure 1: **Robot placed in an Inertial Ground Frame.**

Further, we can take the translational velocities of the robot as u and v along the x_b and y_b axes respectively. r represents the rotational velocity along the z_b axis (coming out of the plane of the paper). Let the center of the robot be at coordinates (x, y) and at an angle ψ with respect to the inertial ground frame I. As can be seen in **Figure 2**.

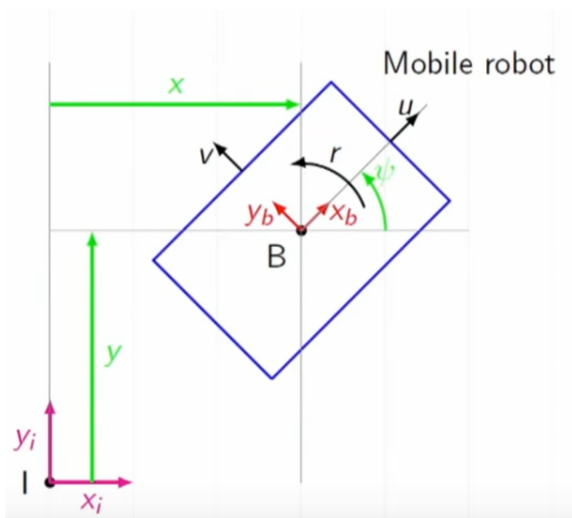

Figure 2: **Velocity of Robot WRT Ground Frame.**

Taking components of the robot velocity u, v and r in the ground frame, we get

$$\dot{y} = vcos(\psi) + usin(\psi)$$

$$\dot{x} = ucos(\psi) - vsin(\psi)$$

$$\dot{\psi} = r$$

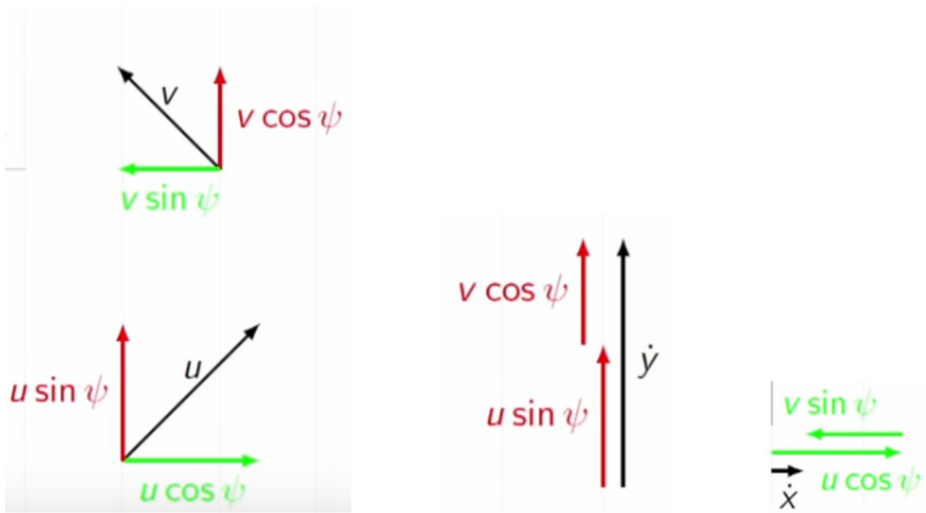

Figure 3: **Components of the Robot Velocity in the Ground Frame.**

We can represent this set of equation in matrix form for a cleaner and simplified representation as,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} ucos(\psi) - vsin(\psi) \\ vcos(\psi) + usin(\psi) \\ r \end{bmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\psi)\boldsymbol{\zeta} \tag{1.1}$$

Where,

η represents the position of the robot in the ground frame and $\dot{\eta}$ represents the velocity WRT ground frame

$J(\psi)$ represents the rotation matrix that relates the robot frame and the ground frame

ζ represents the robot velocity in the robot frame

Running a Basic Simulation

Plotting the Trajectory

Using eq 1.1, we can do a basic kinematic simulation of a mobile robot. We will use MATLAB to run this simulation.

```
%% Simulation Parameters
dt = 0.1;          % Differential units of time
ts = 10;           % Total simulation time
t = 0:dt:ts;

%% Initial Conditions
x0 = 0;
y0 = 0;
psi0 = 0;

eta(:,1) = [x0; y0; psi0];   % Initial pose of the robot

%% Trajectory calculation
for i = 1:length(t)
psi = eta(3,i);
```

```
  % Rotation matrix
  j_psi = [cos(psi) -sin(psi) 0;
           sin(psi) cos(psi) 0;
           0 0 1];

  % Velocity input command
  u = 0.1;
  v = 0.1;
  r = 0;

  zeta = [u;v;r];

  % Pose in Ground frame
  eta_dot(:,i) = j_psi*zeta;
  eta(:,i+1) = eta(:,i) + dt*eta_dot(:,i);
end
%% Plotting trajactory
axis([-2 2 -2 2]);
plot(eta(1,1:length(t)),eta(2,1:length(t)));
xlabel('x');
ylabel('y');
```

We can tell that the simulation is correct by noticing the input velocities of $u = 0.1$, $v = 0.1$ and $r = 0$, the time of the simulation, i.e. 10 seconds which would give us a final position of $(x, y) = (1, 1)$. This can be verified in the graph generated by the simulation(**Figure 4**).

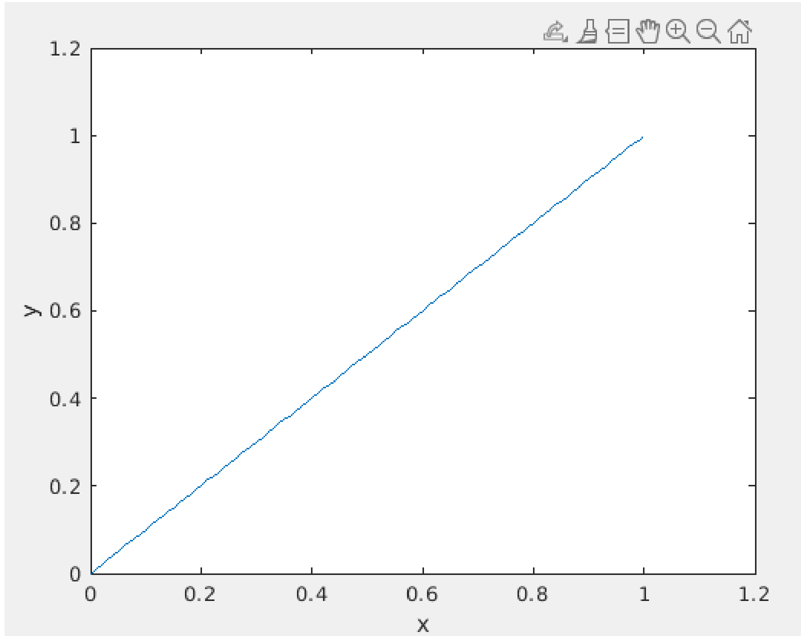

Figure 4: **Trajectory genetated by the above simulation**

Animating the Trajectory

With the above MATLAB code we can plot the trajectories for various velocity inputs. However, it doesn't tell us how the robot moves from one moment to the other. In order to see the robot moving as time passes, we need to include a section of code which animates the trajectory, as follows:

```
%% Animation of motion
function [] = Animate(l,w,eta,t)
% Coordinates of Robot
Rco = [-l/2 -l/2 l/2 l/2 -l/2;
        w/2 -w/2 -w/2 w/2 w/2];
% Animation
figure
for i = 1:length(t)
    psi = eta(3,i);
```

```
        r_psi = [cos(psi) -sin(psi);
                 sin(psi) cos(psi)];
        pose = r_psi*Rco;
        fill(pose(1,:)+eta(1,i),pose(2,:)+eta(2,i),'g');
        hold on, grid on
        axis([-2 3 -2 3]);
        plot(eta(1,1:i),eta(2,1:i));
        xlabel('x');
        ylabel('y');
        pause(0.05);
        hold off
    end
```

The animated trajectory cannot be fully seen in this documentation but here is a screenshot of how it may look (**Figure 5**).

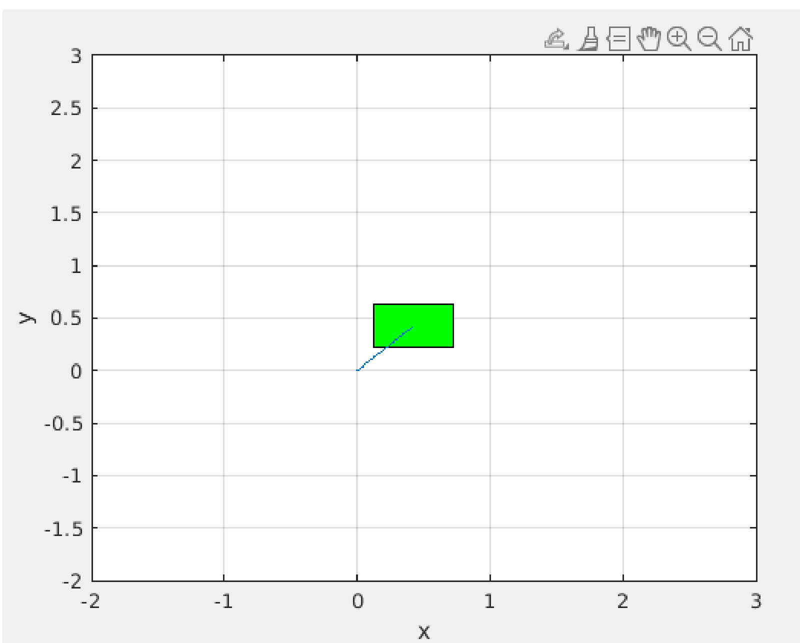

Figure 5: **Intermediate Screenshot of the Animation**

CHAPTER 2

GENERALISED WHEEL KINEMATIC MODEL

Introduction

It can be seen that the motion of the robot in the final animation of the previous chapter looks unnatural, as the direction of motion and orientation of the robot do not have any relation between them. This is because we have not yet included any physical mechanisms in our simulation, which enable the robot motion. In case of land based robots, these mechanisms can be in the form of wheels or legs. And as the title of this report suggests, we are going to explore the kinematics of Wheeled Mobile Robots(WMRs). A basic diagram of a WMR can be seen in the **Figure 6** below, which features a 4-wheel Mecanum drive.

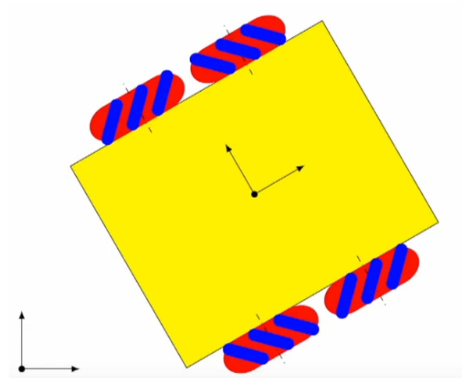

Figure 6: **A 4-wheel Mecanum drive robot.**

Mecanum wheels are a type of omnidirectional wheel which enable a vehicle to move in any direction with a combination of translational and rotational movements. These wheels have passive rollers(shown by blue stripes on the red wheels) mounted at an angle of 45-degree to the axis of the wheel. These rollers allow the wheel to slide laterally and rotate when driven in different directions.

Modeling a Single Mecanum Wheel

Relating the Wheel Angular Velocity and the Wheel Velocity

If we remove the body of the robot and just focus on any one of the wheels, let's say top right for the time being, we will get a diagram that looks like **Figure 7**.

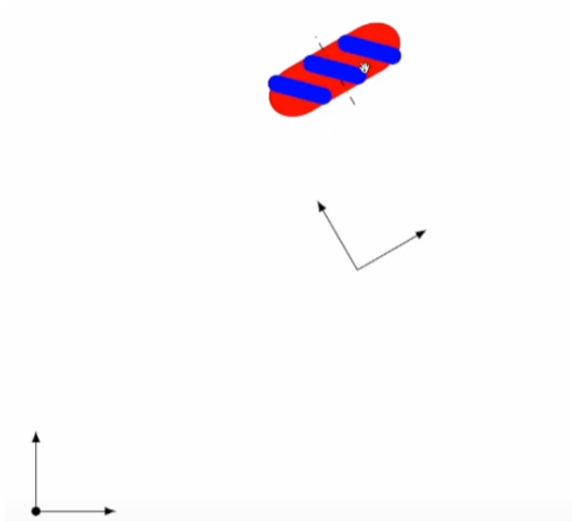

Figure 7: **Single Wheel with Body and Ground Coordinates**

Further focusing on just the robot body coordinates and the wheel, we get the following diagram (**Figure 8**).

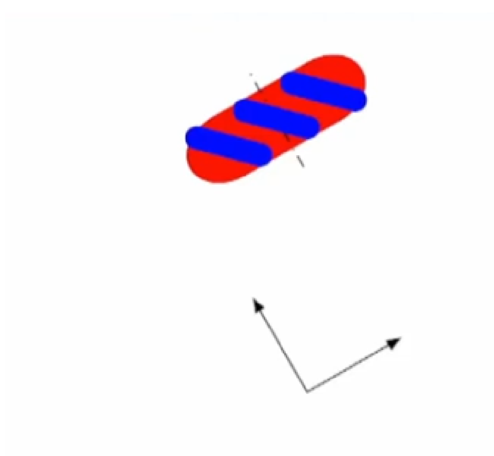

Figure 8: **Single Wheel with the Body Coordinates**

To simplify our calculations, let us consider just one roller. If the wheel rotates with angular velocity ω_i, and the radius of the wheel is a_i. We get the drive velocity, v_{drive} as $\omega_i a_i$. Due to this drive velocity the passive rollers will also rotate and impart a lateral velocity v_{slide} to the wheel. See **Figure 9**.

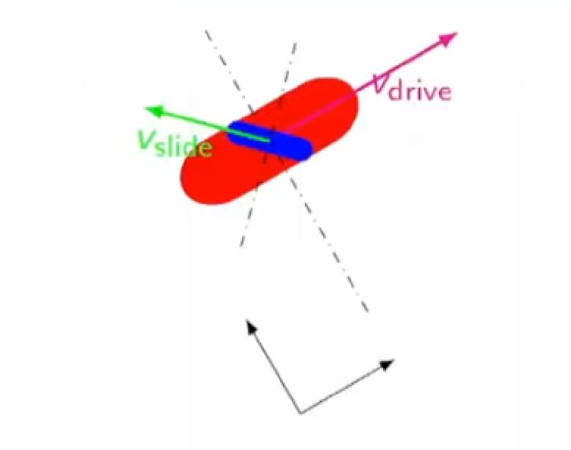

Figure 9: v_{slide} **and** v_{drive}

Considering $\dot{\beta}_i$ as the angular velocity of the passive roller and ρ_i as the radius, we get v_{slide} as $\dot{\beta}_i\rho_i$. Taking c_i as the center of the wheel, and y_{ci} and x_{ci} as the wheel coordinate axes we can make the following diagram (**Figure 10**).

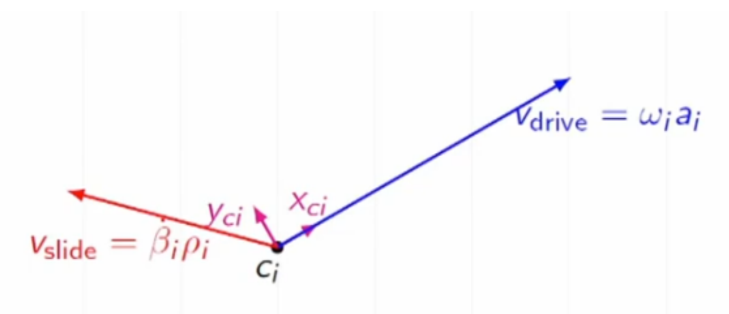

Figure 10: **Velocity of Wheel in the Wheel Frame**

Let the angle between the passive roller velocity v_{slide} and y_{ci} axis be ϕ_i. Resolving v_{slide} into components along the wheel coordinate axes we get $-\dot{\beta}_i\rho_i sin(\phi_i)$ along x_{ci} and $\dot{\beta}_i\rho_i cos(\phi_i)$ along y_{ci} as shown in **Figure 11**.

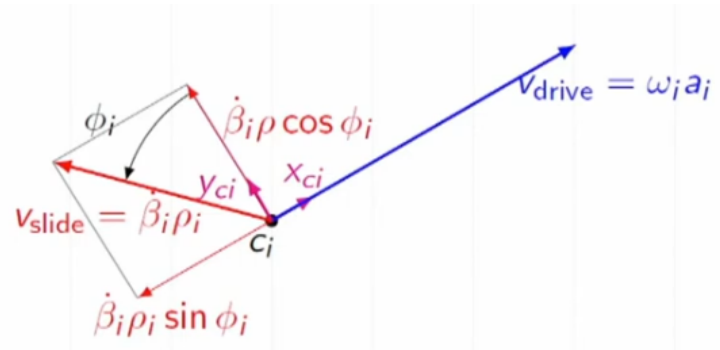

Figure 11: **Resolving v_{slide} into components**

From the above figure we can conclude the following:

$$\dot{y}_{ci} = \dot{\beta}_i \rho_i cos(\phi_i)$$

$$\implies \dot{\beta}_i \rho_i = \frac{\dot{y}_{ci}}{cos(\phi_i)} \tag{2.1}$$

$$\dot{x}_{ci} = \omega_i a_i - \dot{\beta}_i \rho_i sin(\phi_i)$$

Substituting the value of $\dot{\beta}_i \rho_i$ from eq 2.1,

$$\dot{x}_{ci} = \omega_i a_i - \dot{y}_{ci} tan(\phi_i)$$

$$\implies \omega_i a_i = \dot{x}_{ci} + \dot{y}_{ci} tan(\phi_i)$$

$$\implies \omega_i = \frac{1}{a_i}(\dot{x}_{ci} + \dot{y}_{ci} tan(\phi_i)) \tag{2.2}$$

Representing eq 2.2 in vector form,

$$\omega_i = \frac{1}{a_i} \begin{bmatrix} 1 & tan(\phi_i) \end{bmatrix} \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix} \tag{2.3}$$

The above equation represents the relation between wheel angular velocity and the wheel velocity in the wheel frame.

Relating the Wheel Velocity with the Robot Velocity

Until now we looked at the velocity of the wheel in the wheel frame. Let us now take a step back from the wheel frame and see things from the robot frame. Remember **Figure 8**, we will rotate that figure a little bit and take a generalised case in which the wheel is at some angle θ_{Bi} WRT the body frame B. This will result in the following diagram (**Figure12**):

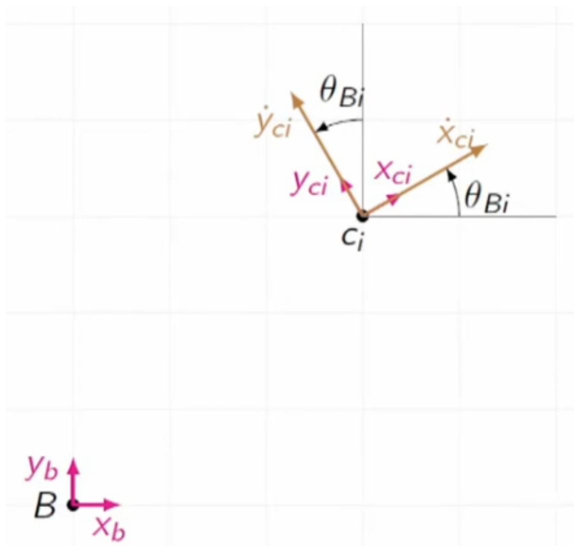

Figure 12

We know that velocity of the wheel in the wheel frame can be written as:

$$^{ci}v_{ci} = \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix}$$

From the above diagram we can see that the wheel velocity in the robot frame can be written as a rotation transformation about an axis perpendicular to the plane by an angle θ_{Bi} as follows:

$$^{B}v_{ci} = {}^{B}_{ci}R(\theta_{Bi})\, {}^{ci}v_{ci}$$

$$\implies {}^{B}v_{ci} = \begin{bmatrix} cos(\theta_{Bi}) & -sin(\theta_{Bi}) \\ sin(\theta_{Bi}) & cos(\theta_{Bi}) \end{bmatrix} \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix} \tag{2.4}$$

We can arrive at another equation for ${}^{B}v_{ci}$ by approaching it from the vantage point of vehicle velocity. In doing so we will be able to relate the wheel velocity and the vehicle velocity. In the diagram below (**Figure 13**), the velocity of the body frame is shown.

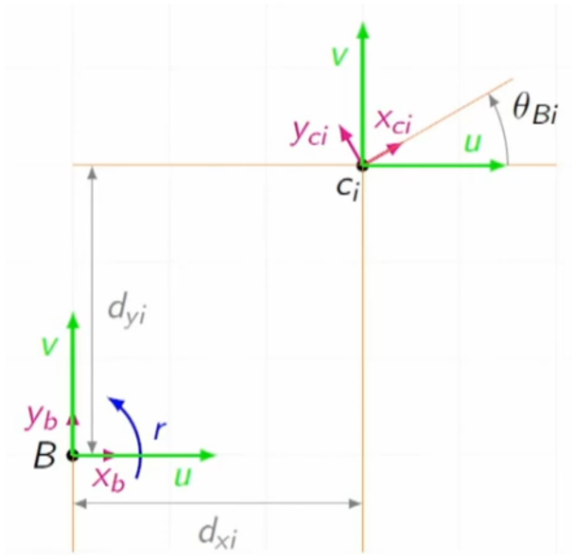

Figure 13

Let the wheel coordinates WRT body frame be (d_{xi}, d_{yi}), we can then write:

$$\implies {}^{B}v_{ci} = \begin{bmatrix} u - rd_{yi} \\ v + rd_{xi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_{yi} \\ 0 & 1 & d_{xi} \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{2.5}$$

As can be seen in **Figure 14**.

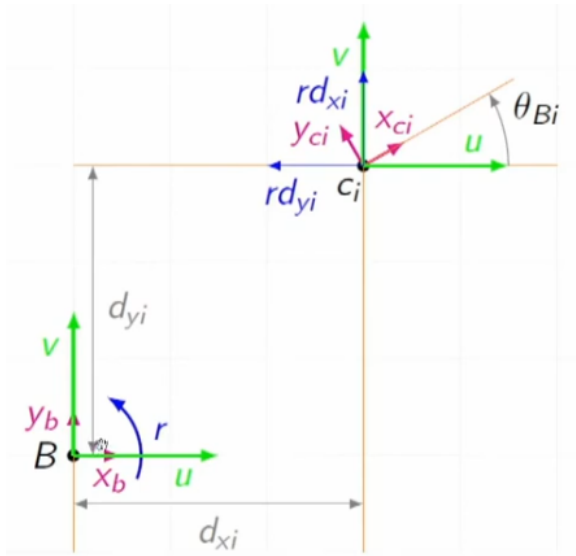

Figure 14: **Wheel Velocity in the Body Frame**

We now have two equations for ${}^{B}v_{ci}$, eq 2.4 and 2.5. We can equate them to get the following result:

$$\begin{bmatrix} cos(\theta_{Bi}) & -sin(\theta_{Bi}) \\ sin(\theta_{Bi}) & cos(\theta_{Bi}) \end{bmatrix} \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_{yi} \\ 0 & 1 & d_{xi} \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

$$\implies \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix} = \begin{bmatrix} cos(\theta_{Bi}) & sin(\theta_{Bi}) \\ -sin(\theta_{Bi}) & cos(\theta_{Bi}) \end{bmatrix} \begin{bmatrix} 1 & 0 & -d_{yi} \\ 0 & 1 & d_{xi} \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{2.6}$$

With this we now have a relation between the wheel velocity and the robot velocity.

Relating the Wheel Angular Velocity with the Robot Velocity: The Generalised Wheel Kinematic Model

In the initial section we derived eq 2.3, showing the relation between the wheel angular velocity and the wheel velocity in the wheel frame as:

$$\omega_i = \frac{1}{a_i}\begin{bmatrix} 1 & tan(\phi_i) \end{bmatrix}\begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix}$$

And in the last section we derived eq 2.6, showing the relation between the wheel velocity and the robot velocity as:

$$\begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix} = \begin{bmatrix} cos(\theta_{Bi}) & sin(\theta_{Bi}) \\ -sin(\theta_{Bi}) & cos(\theta_{Bi}) \end{bmatrix}\begin{bmatrix} 1 & 0 & -d_{yi} \\ 0 & 1 & d_{xi} \end{bmatrix}\begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

We can now substitute the value of $\begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix}$ from eq 2.6 into eq 2.3 to get the following result:

$$\implies \omega_i = \frac{1}{a_i}\begin{bmatrix} 1 & tan(\phi_i) \end{bmatrix}\begin{bmatrix} cos(\theta_{Bi}) & sin(\theta_{Bi}) \\ -sin(\theta_{Bi}) & cos(\theta_{Bi}) \end{bmatrix}\begin{bmatrix} 1 & 0 & -d_{yi} \\ 0 & 1 & d_{xi} \end{bmatrix}\begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{2.7}$$

Where,

ω_i represents the angular velocity of the i^{th} wheel

a_i represents the radius of the i^th wheel

θ_{Bi} represents the angle between the vehicle frame (B) and the wheel frame (c_i)

d_{xi} and d_{yi} are the position coordinates of c_i with reference to frame B

ϕ_i is the angle between the passive roller axis and x_{ci} axis

u, v and r represent the forward, lateral and angular velocities of the mobile robot with respect to the frame B.

This equation relates the angular velocity of a single wheel with the robot velocity. In order to get a complete picture we must relate the angular velocities of all the wheels with the robot velocity. Once we have done that, we will get an equation that looks like:

$$\boldsymbol{\omega} = \boldsymbol{M}\boldsymbol{\zeta} \tag{2.8}$$

$$\implies \boldsymbol{\zeta} = \boldsymbol{W}\boldsymbol{\omega} \tag{2.9}$$

Where,

$\boldsymbol{\zeta}$ represents the Input Velocity Vector

$\boldsymbol{W}$ represents the Wheel Configuration Matrix and,

$\boldsymbol{\omega}$ represents the Wheel Angular Velocity Vector

Using this Generalised Model we can calculate the different input velocities for different angular velocities of wheels with different wheel configurations. We will explore this with a few examples in the next chapter.

CHAPTER 3

SIMULATING DIFFERENT WHEEL CONFIGURATIONS

Introduction

In the last chapter we arrived at an equation relating the wheel angular velocities and the robot velocities. Building upon that, in this chapter we will go through some examples of how different wheel configurations can be modelled using eq 2.9.

Two Wheel Differential Drive

In a two-wheel differential drive system the robot is propelled by two wheels with a differential mechanism. This setup is straightforward yet highly effective for achieving basic movement and maneuverability. The robot is equipped with two main wheels, one on each side, that provide propulsion and movement. These wheels are usually driven by individual motors. A differential mechanism connects the two wheels, allowing them to rotate at different speeds. This enables the robot to perform turning maneuvers by varying the speed of each wheel independently. Along with the two powered wheels there can be a third passive wheel so as to ensure balance. A basic skeleton of such a robot can be seen in **Figure 15**.

Calculating the Wheel Configuration Matrix

In order to calculate the wheel configuration matrix, we must first figure out all the relevant parameters of the vehicle.

Let the radius of both wheels be a. For simplicity, let the wheel axis coincide with the y_{ci} axis making $d_{x1} = d_{x2} = 0$ and $\theta_{B1} = \theta_{B2} = 0$. If the wheels are placed at a distance $2d$ apart then $d_{y1} = d$ and $d_{y2} = -d$. There are no passive rollers here so $\phi_1 = \phi_2 = 0$.

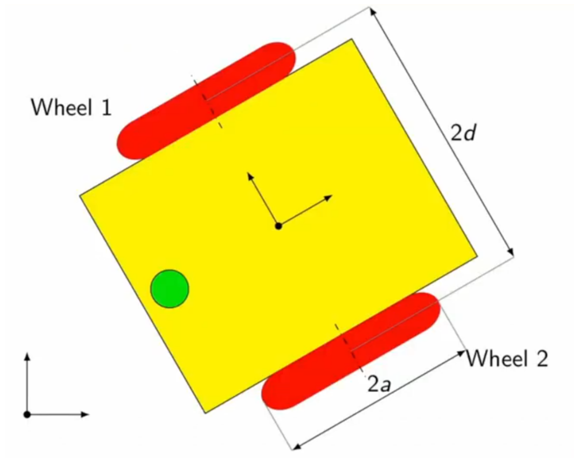

Figure 15: **A Two Wheel Differential Drive**

Now that we have all the parameters, we can substitute them into the generalised wheel model(eq 2.9) as follows:

$$\omega_i = \frac{1}{a_i}\begin{bmatrix}1 & tan(\phi_i)\end{bmatrix}\begin{bmatrix}cos(\theta_{Bi}) & sin(\theta_{Bi})\\ -sin(\theta_{Bi}) & cos(\theta_{Bi})\end{bmatrix}\begin{bmatrix}1 & 0 & -d_{yi}\\ 0 & 1 & d_{xi}\end{bmatrix}\begin{bmatrix}u\\ v\\ r\end{bmatrix}$$

$$\implies \omega_1 = \frac{1}{a}\begin{bmatrix}1 & tan(0)\end{bmatrix}\begin{bmatrix}cos(0) & sin(0)\\ -sin(0) & cos(0)\end{bmatrix}\begin{bmatrix}1 & 0 & -d\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ r\end{bmatrix}$$

$$\omega_2 = \frac{1}{a}\begin{bmatrix}1 & tan(0)\end{bmatrix}\begin{bmatrix}cos(0) & sin(0)\\ -sin(0) & cos(0)\end{bmatrix}\begin{bmatrix}1 & 0 & d\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ r\end{bmatrix}$$

$$\implies \omega_1 = \frac{1}{a}\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -d\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ r\end{bmatrix} = \frac{1}{a}(u - rd) \tag{3.1}$$

$$\omega_2 = \frac{1}{a}\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & d\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ r\end{bmatrix} = \frac{1}{a}(u + rd) \tag{3.2}$$

Combining eq 3.1 and 3.2 we can write a vector equation as:

$$\begin{bmatrix}\omega_1\\ \omega_2\end{bmatrix} = \begin{bmatrix}\frac{1}{a}(u - rd)\\ \frac{1}{a}(u + rd)\end{bmatrix}$$

$$\implies \begin{bmatrix}\omega_1\\ \omega_2\end{bmatrix} = \begin{bmatrix}\frac{1}{a} & -\frac{d}{a}\\ \frac{1}{a} & \frac{d}{a}\end{bmatrix}\begin{bmatrix}u\\ r\end{bmatrix} \tag{3.3}$$

The above eq 3.3 is of the form $\boldsymbol{\omega} = \boldsymbol{M}\boldsymbol{\zeta}$ where,

$$\boldsymbol{M} = \begin{bmatrix}\frac{1}{a} & -\frac{d}{a}\\ \frac{1}{a} & \frac{d}{a}\end{bmatrix}$$

In order to reach the final form of eq 2.9, we need to calculate $\boldsymbol{W} = \boldsymbol{M}^{-1}$. Calculating the inverse of a 2x2 non singular matrix is fairly simple but not all wheel configurations allow for such simple calculations as we will see ahead. However for the time being we can calculate $\boldsymbol{W}$ and get the final equation as:

$$\begin{bmatrix}u\\ r\end{bmatrix} = \begin{bmatrix}\frac{a}{2} & \frac{a}{2}\\ -\frac{a}{2d} & \frac{a}{2d}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\end{bmatrix}$$

$$\Longrightarrow \begin{bmatrix} u \\ v \\ r \end{bmatrix} = \boldsymbol{\zeta} = \begin{bmatrix} \frac{a}{2} & \frac{a}{2} \\ 0 & 0 \\ -\frac{a}{2d} & \frac{a}{2d} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \boldsymbol{W}\boldsymbol{\omega} \tag{3.4}$$

Simulating a Two Wheel Differential Drive

Now that we have derived a kinematic wheel model of a two wheel differential drive, we can run a simulation based on the same and see it in action. We already have the code to animate a mobile robot if we have the input velocity matrix from **chapter 1**. All we need to do now is calculate the velocities from the input angular velocities of the wheels using eq 3.4, instead of directly giving the input velocities.

We can do this by defining a function that takes in the angular velocities of the wheels and gives the resulting robot velocity as follows:

```
function [eta] = TwoWheelDD(eta,Vehicle_Parameters,omega,t,dt)
    for i = 1:length(t)
    psi = eta(3,i);
    j_psi = [cos(psi) -sin(psi) 0;
             sin(psi) cos(psi) 0;
             0 0 1];

    % Wheel configuration matrix
    a = Vehicle_Parameters(1);       % Wheel radius
    d = Vehicle_Parameters(2)/2;     % Distance between the wheels

    W = [a/2 a/2;
         0 0;
        -a/(2*d) a/(2*d)];
```

```
    % Velocity input command
    zeta(:,i) = W*omega;

    % Pose in Ground frame
    eta_dot(:,i) = j_psi*zeta(:,i);
    eta(:,i+1) = eta(:,i) + dt*eta_dot(:,i);
end
```

Calling the function would look like this:

```
%% Simulation Parameters
dt = 0.1;
ts = 10;
t = 0:dt:ts;

%% Physical parameters of the vehicle
a = 0.3;     % Wheel radius
w = 0.4;     % Width of the vehicle
l = 0.6;     % Length of the vehicle

%% Initial Conditions
x0 = 0;
y0 = 0;
psi0 = 0;
eta(:,1) = [x0; y0; psi0];
```

```
%% Trajectory Calculation
% Two Wheel Differential Drive
    omegaL = 0.5;    % Angular velocity of the left wheel
    omegaR = 0.5;    % Angular velocity of the right wheel
    omega_DD = [omegaL;omegaR];

    eta = TwoWheelDD(eta,[a;w],omega_DD,t,dt);

% Animation of motion
Animate(l,w,eta,t);
```

By changing the angular velocities of the wheel i.e. omegaDD, we can make the robot follow various trajectories. In the code provided above, since both the wheel angular velocities are equal, the robot will simply follow a straight line path as can be seen in the **Figure 16** below.

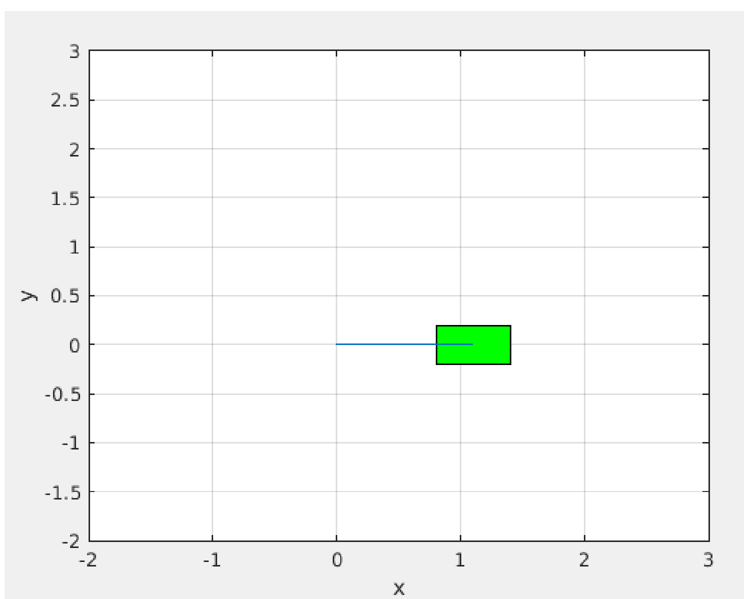

Figure 16: **A Two Wheel Differential Drive moving straight**

Four Wheel Mecanum Drive

We are already familiar with what a Mecanum wheel drive is, form the previous chapter. Now let us try and figure out the wheel kinematic model for such a wheel configuration.

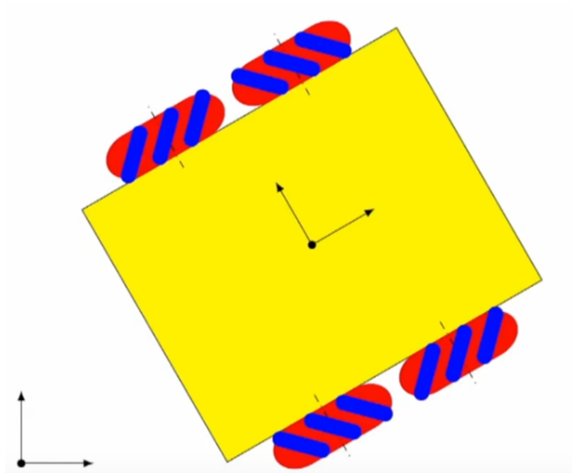

Figure 17: **A 4-wheel Mecanum drive robot.**

Calculating the Wheel Configuration Matrix

The process will be the same as the one we followed in the previous section for two wheel differential drive. We begin by determining all the relevant vehicle parameters. First let us number the wheel so that there is no confusion later on. The wheels on the left side of the vehicle are numbered 1(front) and 2(rear), and the wheels on the right side of the vehicle are numbered 3(front) and 4(rear).

Starting with the wheel radii, let it be a for all the wheels. Let the distance between the wheels on the opposite side of the vehicle be $2d$ which results in $d_{y1} = d_{y3} = d$ and $d_{y2} = d_{y4} = -d$. Let the distance between the wheels on the same side of the vehicles be $2l$ which gives us $d_{x1} = d_{x4} = l$ and $d_{x2} = d_{x3} = -l$. Since all the wheels are aligned in the same direction as the vehicle, $\theta_{Bi} = 0$. And lastly for the angle of the passive roller let the magnitude be $\pi/4$. As can be seen in the **Figure 17**, the rollers of the front wheel have a positive angle where as the rear wheels have a negative angle, so we get $\phi_1 = \phi_3 = \pi/4$ and $\phi_2 = \phi_4 = -\pi/4$.

With all the parameters determined, we can now move on to the next step which is putting these values in the eq 2.9:

$$\omega_i = \frac{1}{a_i}\begin{bmatrix}1 & tan(\phi_i)\end{bmatrix}\begin{bmatrix}cos(\theta_{Bi}) & sin(\theta_{Bi}) \\ -sin(\theta_{Bi}) & cos(\theta_{Bi})\end{bmatrix}\begin{bmatrix}1 & 0 & -d_{yi} \\ 0 & 1 & d_{xi}\end{bmatrix}\begin{bmatrix}u \\ v \\ r\end{bmatrix}$$

$$\implies \omega_1 = \frac{1}{a}\begin{bmatrix}1 & tan(\pi/4)\end{bmatrix}\begin{bmatrix}cos(0) & sin(0) \\ -sin(0) & cos(0)\end{bmatrix}\begin{bmatrix}1 & 0 & -d \\ 0 & 1 & l\end{bmatrix}\begin{bmatrix}u \\ v \\ r\end{bmatrix}$$

$$\omega_2 = \frac{1}{a}\begin{bmatrix}1 & tan(-\pi/4)\end{bmatrix}\begin{bmatrix}cos(0) & sin(0) \\ -sin(0) & cos(0)\end{bmatrix}\begin{bmatrix}1 & 0 & -d \\ 0 & 1 & -l\end{bmatrix}\begin{bmatrix}u \\ v \\ r\end{bmatrix}$$

$$\omega_3 = \frac{1}{a}\begin{bmatrix}1 & tan(\pi/4)\end{bmatrix}\begin{bmatrix}cos(0) & sin(0) \\ -sin(0) & cos(0)\end{bmatrix}\begin{bmatrix}1 & 0 & d \\ 0 & 1 & -l\end{bmatrix}\begin{bmatrix}u \\ v \\ r\end{bmatrix}$$

$$\omega_4 = \frac{1}{a}\begin{bmatrix}1 & tan(-\pi/4)\end{bmatrix}\begin{bmatrix}cos(0) & sin(0) \\ -sin(0) & cos(0)\end{bmatrix}\begin{bmatrix}1 & 0 & d \\ 0 & 1 & l\end{bmatrix}\begin{bmatrix}u \\ v \\ r\end{bmatrix}$$

$$\implies \omega_1 = \frac{1}{a}\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -d\\0 & 1 & l\end{bmatrix}\begin{bmatrix}u\\v\\r\end{bmatrix} = \frac{1}{a}[u+v-r(d-l)]$$

$$\omega_2 = \frac{1}{a}\begin{bmatrix}1 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -d\\0 & 1 & -l\end{bmatrix}\begin{bmatrix}u\\v\\r\end{bmatrix} = \frac{1}{a}[u-v-r(d-l)]$$

$$\omega_3 = \frac{1}{a}\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & d\\0 & 1 & -l\end{bmatrix}\begin{bmatrix}u\\v\\r\end{bmatrix} = \frac{1}{a}[u+v+r(d-l)]$$

$$\omega_4 = \frac{1}{a}\begin{bmatrix}1 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & d\\0 & 1 & l\end{bmatrix}\begin{bmatrix}u\\v\\r\end{bmatrix} = \frac{1}{a}[u-v+r(d-l)]$$

Combining the above four equations into vector form we get:

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix} = \begin{bmatrix}\frac{1}{a}[u+v-r(d-l)]\\\frac{1}{a}[u-v-r(d-l)]\\\frac{1}{a}[u+v+r(d-l)]\\\frac{1}{a}[u-v+r(d-l)]\end{bmatrix}$$

$$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix} = \frac{1}{a}\begin{bmatrix}1 & 1 & -d+l\\1 & -1 & -d+l\\1 & 1 & d-l\\1 & -1 & d-l\end{bmatrix}\begin{bmatrix}u\\v\\r\end{bmatrix} \tag{3.5}$$

From eq 3.5 we can conclude that

$$\boldsymbol{M} = \frac{1}{a}\begin{bmatrix} 1 & 1 & -d+l \\ 1 & -1 & -d+l \\ 1 & 1 & d-l \\ 1 & -1 & d-l \end{bmatrix}$$

In order to find $\boldsymbol{W}$, we need $\boldsymbol{M^{-1}}$ but $\boldsymbol{M}$ is not a square matrix, we cannot have a normal inverse. We did not encounter this issue in the previous case where we found $\boldsymbol{M}$ as a simple 2x2 non-singular square matrix. Here we need to use some math trickery in order to find $\boldsymbol{W}$. Since we do not have access to a normal inverse we are going to settle for the next closest thing, which is a pseudo-inverse. We will not get into how exactly the pseudo-inverse is calculated here, rather just use the result of the operation, which gives us:

$$\boldsymbol{W} = \frac{a}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \frac{-1}{d-l} & \frac{-1}{d-l} & \frac{1}{d-l} & \frac{1}{d-l} \end{bmatrix}$$

Using this value of $\boldsymbol{W}$, we get the final equation as follows:

$$\begin{bmatrix} u \\ v \\ r \end{bmatrix} = \boldsymbol{\zeta} = \frac{a}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \frac{-1}{d-l} & \frac{-1}{d-l} & \frac{1}{d-l} & \frac{1}{d-l} \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \boldsymbol{W\omega} \tag{3.6}$$

Simulating a Four Wheel Mecanum Drive

Just as we did in the case of two wheel differential drive, we will use the above eq 3.6 to simulate the four wheel mecanum drive by defining a function that takes in the angular velocities of the four wheels and calculates the robot velocities. The function goes as follows:

```
function [eta] = Mecanum(eta,Vehicle_parameters,omega,t,dt)
    for i = 1:length(t)
    psi = eta(3,i);
    J_psi = [cos(psi) -sin(psi) 0;
             sin(psi) cos(psi) 0;
             0 0 1];

    %% Wheel configuration matrix
    a = Vehicle_parameters(1);      % Radius of the wheels

    % Coordinates of the wheels
    lw = Vehicle_parameters(2)/2;
    dw = Vehicle_parameters(3)/2;

    W = (a/4)*[1 1 1 1;
         1 -1 1 -1;
         -1/(dw-lw) -1/(dw-lw) 1/(dw-lw) 1/(dw-lw)];

    %% Velocity input command
    zeta(:,i) = W*omega;
```

```
    %% Pose in Ground frame
    eta_dot(:,i) = J_psi*zeta(:,i);
    eta(:,i+1) = eta(:,i) + dt*eta_dot(:,i);
end
```

Calling the function would look like:

```
%% Simulation Parameters
dt = 0.1;
ts = 10;
t = 0:dt:ts;

%% Physical parameters of the vehicle
a = 0.3;     % Radius of the wheels
w = 0.4;     % Distance between the wheels on the same side
l = 0.6;     % Distance between the wheels on opposite sides

%% Initial Conditions
x0 = 0;
y0 = 0;
psi0 = 0;

eta(:,1) = [x0; y0; psi0];

%% Trajectory Calculation
% Mecanum Wheel Drive
    omega1 = 1.5;
    omega2 = -1.5;
```

```
    omega3 = 1.5;
    omega4 = -1.5;

    omega_M = [omega1;omega2;omega3;omega4];

    eta = Mecanum(eta,[a;l;w],omega_M,t,dt);

% Animation of motion
Animate(l,w,eta,t);
```

By changing the angular velocities of the wheels i.e. omegaM, we can make the robot follow various trajectories. In the code provided above, the input angular velocities are such that the robot will move laterally along the y axis in a straight line path as it can be seen in the **Figure 17** below.

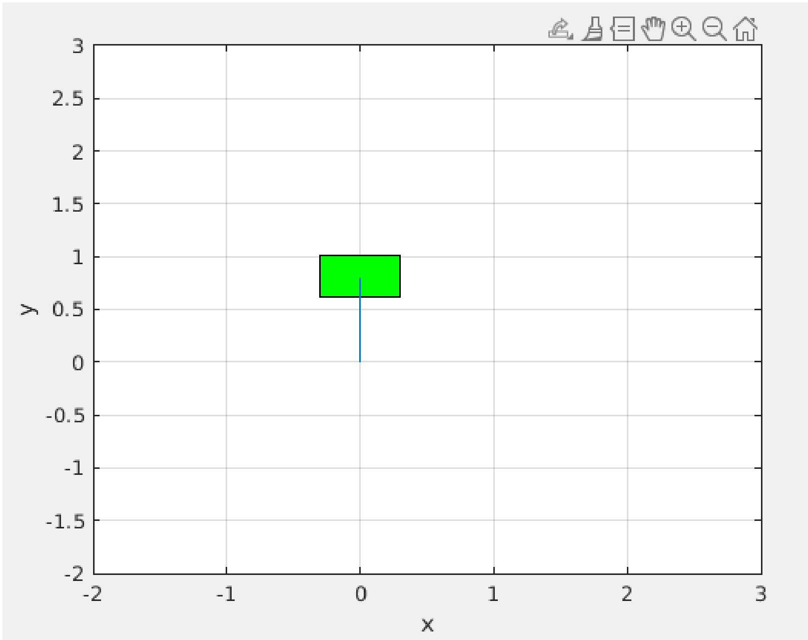

Figure 18: **Mecanum Wheel Drive motion corresponding to the code**