DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ROBOTICS

Design of desired trajectories and motion planning of a wheeled robot - Motion Planning of a robotic arm in a simulation environment.

Team

Gkovaris Christos-Grigorios
Spanou Maria

Timetable of the Assignment

Version	Date	Progress
1.0	23/4/2024	Lab Exercise Announcement
2.0	24/5 - 31/5	Implementation of move_jackal.py
2.1	15/5 – 21/5	Implementation of three different trajectories
2.2	24/5 - 31/5	Implementation of trajectories_jackal.py
3.0	24/5 - 31/5	Implementation of move_rrbot_joint.py
3.1	15/5 – 21/5	Implementation of trajectories for the joints
3.2	24/5 - 31/5	Implementation of trajectories_rrbot.py
4.0	15/5 – 1/6	Implementation of Lab Exercise Report

Feature Analysis

The assignment was implemented on a machine (Lenovo Ideapad 3), with the following specifications:

- AMD Ryzen 7 (7730U)
- 8 CPU cores
- 16 Threads
- Boost Clock up to 4.5GHz
- Base Clock 2.0GHz
- CPU Socket FP6
- Intergraded Graphics (Radeon Graphics)

Furthermore, the assignment was implemented on **Windows 11 Home**, using the **Visual Studio**.

2.0 Implementation of move_jackal.py

The development of the 'move_jackal.py' program helps in moving the wheeled robot with an arbitrary desired linear velocity (u_d) and rotational velocity (ω_d). The above velocities are expressed with respect to the body-fixed frame.

The program begins by importing the necessary libraries, including rospy (for controlling ROS), argparse (for processing command-line arguments), and matplotlib.pyplot (for displaying graphs). It then defines the move_jackal function, which takes as parameters the transmission frequency (rate_hz), the linear and angular velocities (linear_velocity and angular_velocity_deg, respectively), and the movement duration (duration). The angular velocities are converted from degrees to radians.

The command that needs to be executed in the terminal for the correct operation of the robot is:

```
python3 move_jackal.py <rate_hz> <linear_velocity> <angular_velocity_deg> <duration>
```

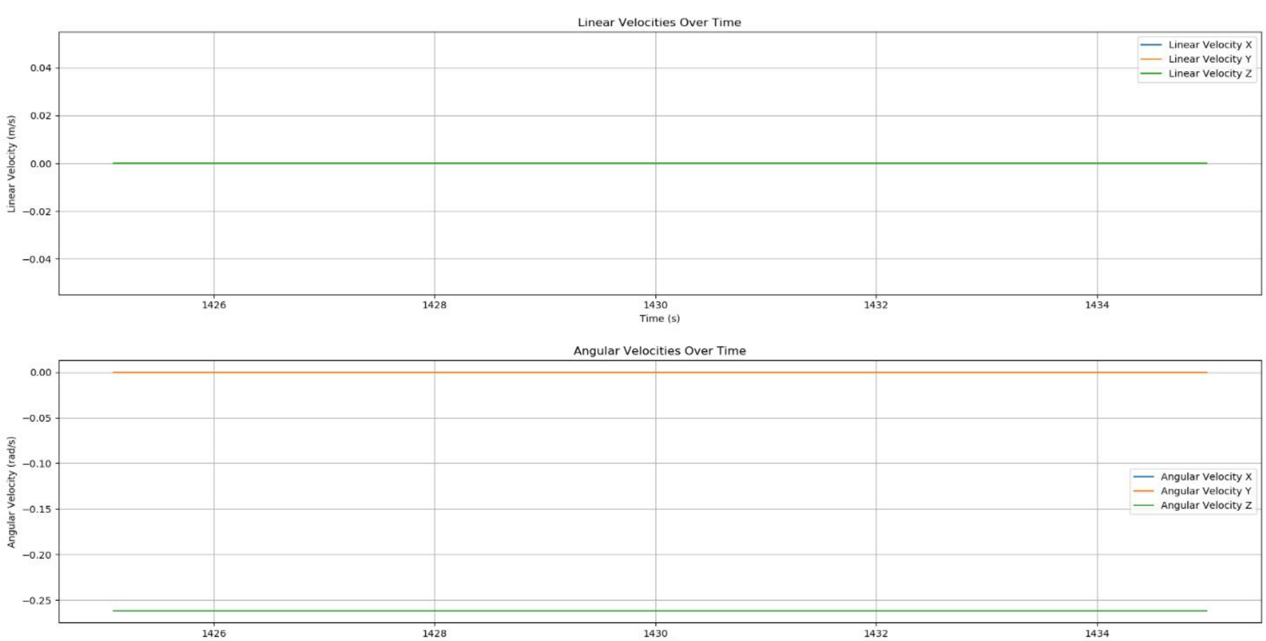
Execution of the above command for:

rate_hz = 10 Hz

linear_velocity = 0 m/sec

angular_velocity_deg = -15°/sec

duration = 10 sec



The results of the screenshot show the linear and angular velocities of the Jackal robot during the execution of the program with the mentioned parameters.

The linear velocities for the X, Y, and Z axes are all zero (0 m/s). This is expected since we set the linear velocity to [0.0, 0.0, 0.0], meaning that the robot does not move linearly in any direction.

The angular velocities on the Z-axis show a constant value of approximately -0.25 rad/s. This value is correct because it corresponds to the conversion of -15°/sec to radians. The conversion is:

$$-15^{o} \cdot \frac{\pi}{180} = -0.2618 \text{ rad/sec}$$

The angular velocities on the X and Y axes are zero, as expected, since no angular velocity was set for these axes.

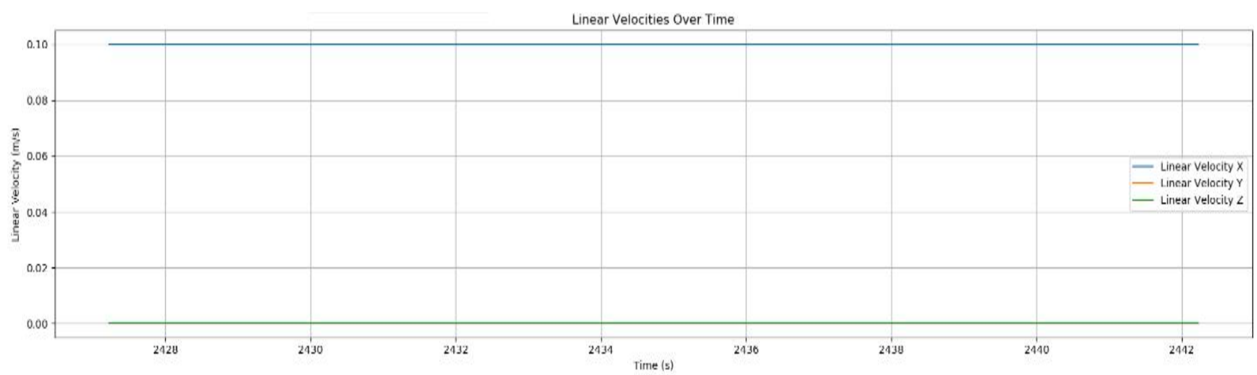
<u>Execution of the above command for:</u>

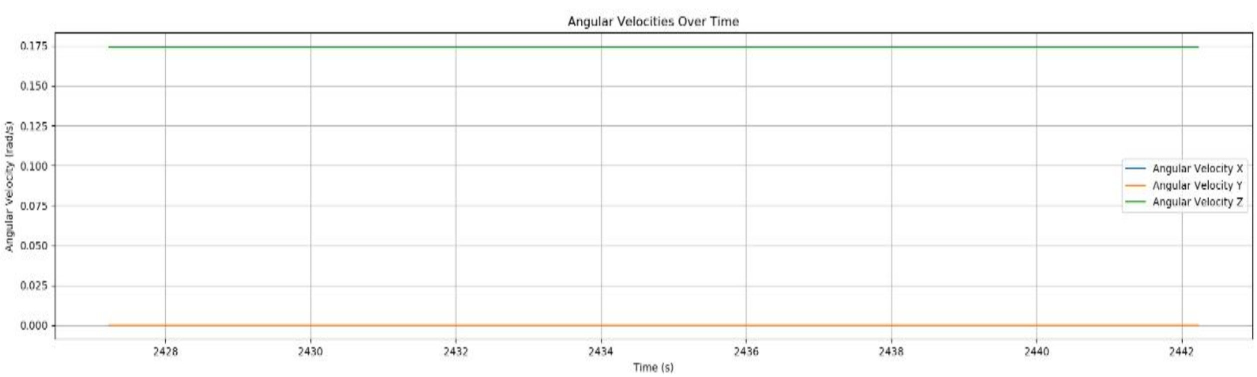
rate_hz = 10 Hz

linear_velocity = 0.1 m/sec

angular_velocity_deg = 10°/sec

duration = 15 sec





2.1 Implementation of three different trajectories

To develop the solution, we will use the method of linear functions with parabolic segments, taking into account the parameters given in Table 2 and the registration number (AM=5351) for determining the time parameters. The initial conditions will be as follows:

Initial Position : (x_0,y_0),

Final Position : (x_f,y_f),

Initial Orientation : θ_0,

Final Orientation : θ_f,

Maximum Linear Velocity : 0.2m/sec

Maximum Angular Velocity : 30°/sec

The time parameters for the three trajectories we are required to calculate are defined as follows:

Total time for each path : t_f

Acceleration/Deceleration time : $t_b = 0.1 \cdot t_f$

Rotation of the Robot:

The first trajectory will describe the rotation of the robot so that it "looks" at a given final position (x_f,y_f). The following will apply::

For the final angular position (θ_f), the values are derived from Table 2, so the following applies:

$$\theta_f = \tan^{-1}\left(\frac{y_f}{x_f}\right) = \tan^{-1}(3/5) = 0.54 \text{ rad}$$

For the maximum angular velocity ($\dot{\theta}_{max}$), which is provided in the statement, the following holds:

$$\dot{\theta}_{max} = 30°/\text{sec} = 30 \cdot \frac{\pi}{180} = \frac{\pi}{6} \text{ rad/sec}$$

The mathematical expression of the trajectory, will be as follows:

$$\theta_{(t)} = \begin{Bmatrix} \theta_0 + \frac{\ddot{\theta}}{2}t^2 \quad ,0 \le t \le t_b \\ \theta_0 + \frac{\ddot{\theta}}{2}t_b^2 + \ddot{\theta}t_b(t - t_b) \quad ,t_b < t \le t_f - t_b \\ \theta_f - \frac{\ddot{\theta}}{2}\left(t_f - t\right)^2,\ t_f - t_b < t \le t_f \end{Bmatrix}$$

At the time $t = t_b$, the following holds:

$$\dot{\theta}_{coast} = \frac{\theta_h - \theta_b}{t_h - t_b} = \ddot{\theta} \cdot t_b,$$

$$\theta_h = \frac{\theta_f + \theta_0}{2},$$

$$t_h = \frac{t_f}{2},$$

$$\boldsymbol{\theta_b = \theta_0 + \frac{1}{2} \cdot \ddot{\theta}} \cdot t_b^2$$

Solving for t_b , we obtain:

$$t_b = \frac{t_f}{2} - \frac{\sqrt{(\ddot{\theta} \cdot 2 \cdot t_f^2 - 4 \cdot \ddot{\theta} \cdot (\theta_f - \theta_0))}}{2\ddot{\theta}}$$

Thus, it follows that:

$$\theta_{\left(\frac{t_f}{2}\right)} = \frac{\theta_f + \theta_0}{2} = \theta_0 + \frac{\ddot{\theta}}{2} t_b^2 + \ddot{\theta} t_b \left(\frac{t_f}{2} - t_b\right) \quad =>$$

$$\theta_0 + \theta_f = 2\theta_0 + \ddot{\theta} t_b^2 + \ddot{\theta} t_b t_f - 2\ddot{\theta} t_b^2 =>$$

$$\ddot{\theta} = \frac{\theta_f - \theta_0}{t_b(t_f - t_b)} \ (1)$$

I choose $t_b = 0.1 \cdot t_f$ and $\theta_f = 0.54$ rad. From (1), it holds that:

$$\ddot{\theta} = \frac{\theta_f - \theta_0}{t_b(t_f - t_b)} = \frac{0.54}{0.09 t_f^2} = \frac{6}{t_f^2} \ (2)$$

Having the maximum angular velocity $\dot{\theta}_{max} = \frac{\pi}{6}$ rad/sec, it holds that:

$$\dot{\theta}_{\max} = \ddot{\theta}_{max} \cdot t_b \leftrightarrow \ddot{\theta}_{max} = \frac{\dot{\theta}_{\max}}{0.1 \cdot t_f} \leftrightarrow \ddot{\theta}_{max} = \frac{\frac{\pi}{6}}{0.1 \cdot t_f} \leftrightarrow \ddot{\theta}_{max} = \frac{\pi}{0.6 \cdot t_f} \text{ rad}/sec^2 \ (3)$$

However, the following must also hold simultaneously:

$$\left|\ddot{\theta}\right| \leq \left|\ddot{\theta}_{\max.}\right| \xRightarrow{(2),(3)} \frac{6}{t_f^2} \leq \frac{\pi}{0.6 t_f} \leftrightarrow \frac{t_f^2}{6} \geq \frac{0.6 t_f}{\pi} \leftrightarrow t_f \cdot \pi \geq 3.6 \leftrightarrow t_f \geq 1.2 \ sec$$

For $t_f = 1.2$ sec and $t_b = 0.12$ sec, from relation (1), it holds that:

$$\ddot{\theta} \cong \frac{0.54}{0.1296} = 4.16 \text{ rad/sec}^2$$

For $\dot{\theta}_{coast}$, the following applies:

$$\dot{\theta}_{coast} = \frac{6}{1.2^2} \cdot 0.12 = 0.5 \leq 0.52 \text{ rad/sec, ΙΣΧΎΕΙ}$$

Therefore, the trajectory becomes:

$$\theta_{(t)} = \begin{cases} \theta_0 + \frac{\ddot{\theta}}{2} t^2 & ,0 \le t \le t_b \\ \theta_0 + \frac{\ddot{\theta}}{2} t_b^2 + \ddot{\theta} t_b (t - t_b) & ,t_b < t \le t_f - t_b \\ \theta_f - \frac{\ddot{\theta}}{2} (t_f - t)^2 & ,\ t_f - t_b < t \le t_f \end{cases}$$

$$\theta_{(t)} = \begin{cases} \frac{4.16}{2} t^2 & ,0 \le t \le 0.12 \\ \frac{4.16}{2} 0.12^2 + 4.16 \cdot 0.12 (t - 0.12) & ,0.12 < t \le 1.08 \\ 0.54 - \frac{4.16}{2} (1.2 - t)^2 & ,1.08 < t \le 1.2 \end{cases}$$

$$\theta_{(t)} = \begin{cases} 2.08 t^2 & ,0 \le t \le 0.12 \\ 2.08 \cdot 0.12^2 + 4.16 \cdot 0.12 (t - 0.12) & ,0.12 < t \le 1.08 \\ 0.54 - 2.08 (1.2 - t)^2 & ,\ 1.08 < t \le 1.2 \end{cases}$$

$$\theta_{(t)} = \begin{cases} 2.08 t^2 & ,0 \le t \le 0.12 \\ 0.029 + 0.49 (t - 0.12) & ,0.12 < t \le 1.08 \\ 0.54 - 0.81 (1.2 - t)^2 & ,\ 1.08 < t \le 1.2 \end{cases}$$

$$\dot{\theta}_{(t)} = \begin{cases} 4.16 t & ,0 \le t \le 0.12 \\ 0.49 & ,0.12 < t \le 1.08 \\ 1.62 t + 1.94 & ,\ 1.08 < t \le 1.2 \end{cases}$$

$$\ddot{\theta}_{(t)} = \begin{cases} 4.16 & ,0 \le t \le 0.12 \\ 0 & ,0.12 < t \le 1.08 \\ 1.62 & ,\ 1.08 < t \le 1.2 \end{cases}$$

Linear motion of the Robot:

The second trajectory will describe the linear motion of the robot, allowing it to reach the given final position (x_f,y_f). The following will hold:

For the initial and final positions, the values are derived from Table 2, so it follows that:

Initial Position : (0,0) Final Position : (5,3)

For the maximum linear velocity, which is given in the problem statement, the following applies:

V_{max} = 0.2 m/sec

The velocity can be broken down into two components, $\dot{x}$ and $\dot{y}$, which are:

$\dot{x}$ = cos(θ)·V = $\frac{5}{\sqrt{34}} \cdot V$ $\dot{y}$ = sin(θ)·V = $\frac{3}{\sqrt{34}} \cdot V$

The maximum values of the above components, using V_{max}, will be as follows:

$\dot{x}_{max}$ = cos(θ) · V_{max} = 0.171 m/sec $\dot{y}_{max}$ = sin(θ) · V_{max} = 0.103 m/sec

The mathematical formulation of the trajectory, will be as follows:

$$x_{(t)} = \begin{Bmatrix} x_0 + \frac{\ddot{x}}{2}t^2 \quad ,0 \le t \le t_b \\ x_0 + \frac{\ddot{x}}{2}t_b^2 + \ddot{x}t_b(t - t_b) \quad , t_b < t \le t_f - t_b \\ x_f - \frac{\ddot{x}}{2}\left(t_f - t\right)^2 , \ t_f < t \le t_f \end{Bmatrix}$$

$$y_{(t)} = \begin{Bmatrix} y_0 + \frac{\ddot{y}}{2}t^2 \quad ,0 \le t \le t_b \\ y_0 + \frac{\ddot{y}}{2}t_b^2 + \ddot{y}t_b(t - t_b) \quad , t_b < t \le t_f - t_b \\ y_f - \frac{\ddot{y}}{2}\left(t_f - t\right)^2 , \ t_f - t_b < t \le t_f \end{Bmatrix}$$

At the time t = $\frac{t_f}{2}$, the following holds:

$$x_{\left(\frac{t_f}{2}\right)} = \frac{x_f + x_0}{2} = x_0 + \frac{\ddot{x}}{2} t_b^2 + \ddot{x} t_b \left(\frac{t_f}{2} - t_b\right) \quad =>$$

$$x_0 + x_f = 2x_0 + \ddot{x} t_b^2 + \ddot{x} t_b t_f - 2\ddot{x} t_b^2 =>$$

$$\ddot{x} = \frac{x_f - x_0}{t_b(t_f - t_b)} \ (1)$$

I choose t_b = 0.1 · t_f and x_f = 5 m. From (1), it follows that:

$$\ddot{x} = \frac{x_f - x_0}{t_b(t_f - t_b)} = \frac{5}{0.09 t_f^2} = \frac{55.5}{t_f^2} \ (2)$$

Similarly, I choose t_b = 0.1 ˙ t_f and y_f = 0.54 rad. From (1), it follows that:

$$\ddot{y} = \frac{y_f - y_0}{t_b(t_f - t_b)} = \frac{3}{0.09 t_f^2} = \frac{33.3}{t_f^2} \ (3)$$

Given the maximum speed $\dot{x}_{max}$= 0.171 m/sec, it follows that:

$$\dot{x}_{max} = \ddot{x}_{\max.} t_b \leftrightarrow \quad \ddot{x}_{\max.} = \frac{0.171}{0.1 t_f} \leftrightarrow \ddot{x}_{\max.} = \frac{1.71}{t_f} \ (4)$$

However, the following must also hold:

$$|\ddot{x}| \leq |\ddot{x}_{\max.}| \xRightarrow{(2),(4)} \frac{55.5}{t_f^2} \leq \frac{1.71}{t_f} \leftrightarrow \frac{t_f^2}{55.5} \geq \frac{t_f}{1.71} \leftrightarrow t_f \geq \frac{55.5}{1.71} = 32.45 \ (5)$$

Given the maximum speed $\dot{y}_{max}$= 0.103 m/sec, it follows that:

$$\dot{y}_{max} = \ddot{y}_{\max.} t_b \leftrightarrow \quad \ddot{y}_{\max.} = \frac{0.103}{0.1 t_f} \leftrightarrow \ddot{y}_{\max.} = \frac{1.03}{t_f} \ (6)$$

However, the following must also hold:

$$|\ddot{y}| \leq |\ddot{y}_{\max.}| \xRightarrow{(3),(6)} \frac{33.3}{t_f^2} \leq \frac{1.03}{t_f} \leftrightarrow \frac{t_f^2}{33.3} \geq \frac{t_f}{1.03} \leftrightarrow t_f \geq \frac{33.3}{1.03} = 32.33 \ (7)$$

To satisfy the relationships (5) and (7) simultaneously, we choose t_f = 32 sec, which means t_b = 3.2 sec. Therefore, from the relationships (2) and (3), the following will hold:

$$\ddot{x} = \frac{x_f - x_0}{t_b(t_f - t_b)} = \frac{5}{0.09t_f^2} = \frac{55.5}{1024} = 0.05\ \mathrm{m/sec^2}$$

$$\dot{x}_{coast} = \ddot{x}t_b = 0.05 \cdot 3.2 = 0.16\ m/sec \leq 0.171\ \text{m/sec, IΣΧ'ΥΕΙ}$$

$$\ddot{x}_{\min} = \frac{4|x_f - x_0|}{t_f^2} = 4 \cdot \frac{5}{32^2} = 0.02\ \mathrm{m/sec^2}$$

$$(4):\ \ddot{x}_{\max.} = \frac{1.71}{32} = 0.05\ \mathrm{m/sec^2}$$

$$\ddot{y} = \frac{y_f - x_0}{t_b(t_f - t_b)} = \frac{3}{0.09t_f^2} = \frac{3.33}{1024} = 0.03\ \mathrm{m/sec^2}$$

$$\dot{y}_{coast} = \ddot{y}t_b = 0.03 \cdot 3.2 = 0.096\ m/sec \leq 0.103\ \text{m/sec, IΣΧ'ΥΕΙ}$$

$$\ddot{y}_{\min} = \frac{4|y_f - x_0|}{t_f^2} = 4 \cdot \frac{3}{1024} = 0.012\ \mathrm{m/sec^2}$$

$$(4):\ \ddot{y}_{\max.} = \frac{1.03}{32} = 0.032\ \mathrm{m/sec^2}$$

Indeed, after the above calculations, the following results are obtained:

$$\ddot{x}_{\min} \le \ddot{x} \le \ddot{x}_{max}$$

$$\ddot{y}_{\min} \le \ddot{y} \le \ddot{y}_{max}$$

and

$$\dot{x}_{coast} = 0.16 \le \dot{x}_{max} = 0.171 \text{ m/sec}$$

$$\dot{y}_{coast} = 0.096 \le \dot{y}_{max} = 0.103 \text{ m/sec}$$

Therefore, the trajectories become:

$$x_{(t)} = \begin{Bmatrix} \frac{0.05}{2}t^2 & ,0 \le t \le 3.2 \\ \frac{0.05}{2}(3.2)^2 + 0.05 \cdot 3.2(t-3.2) & ,3.2 < t \le 28.8 \\ 5 - \frac{0.05}{2}(32-t)^2 & ,\ 28.8 < t \le 32 \end{Bmatrix}$$

$$x_{(t)} = \begin{Bmatrix} 0.025t^2 & ,0 \le t \le 3.2 \\ 0.16t - 0.256 & ,3.2 < t \le 28.8 \\ 5 - 0.025(1024 - 64t + t^2) & ,\ 28.8 < t \le 32 \end{Bmatrix}$$

$$x_{(t)} = \begin{Bmatrix} 0.025t^2 & ,0 \le t \le 3.2 \\ 0.16t - 0.256 & ,3.2 < t \le 28.8 \\ -0.025t^2 + 1.6t - 20.6 & ,\ 28.8 < t \le 32 \end{Bmatrix}$$

$$\dot{x}_{(t)} = \begin{Bmatrix} 0.05t & ,0 \le t \le 3.2 \\ 0.16 & ,3.2 < t \le 28.8 \\ -0.05t + 1.6 & ,\ 28.8 < t \le 32 \end{Bmatrix}$$

$$\ddot{x}_{(t)} = \begin{Bmatrix} 0.05 & ,0 \le t \le 3.2 \\ 0 & ,3.2 < t \le 28.8 \\ -0.05 & ,\ 28.8 < t \le 32 \end{Bmatrix}$$

$$y_{(t)} = \begin{Bmatrix} \frac{0.03}{2}t^2 \quad ,0 \le t \le 3.2 \\ \frac{0.03}{2}(3.2)^2 + 0.03 \cdot 3.2(t-3.2) \quad ,3.2 < t \le 28.8 \\ 3 - \frac{0.03}{2}(32-t)^2 \, , \; 28.8 < t \le 32 \end{Bmatrix}$$

$$y_{(t)} = \begin{Bmatrix} 0.015t^2 \quad ,0 \le t \le 3.2 \\ 0.096t - 0.1248 \quad ,3.2 < t \le 28.8 \\ 3 - 0.015(1024 - 64t + t^2) \, , \; 28.8 < t \le 32 \end{Bmatrix}$$

$$y_{(t)} = \begin{Bmatrix} 0.015t^2 \quad ,0 \le t \le 3.2 \\ 0.096t - 0.1248 \quad ,3.2 < t \le 28.8 \\ -0.015t^2 + 0.96t - 12.36 \, , \; 28.8 < t \le 32 \end{Bmatrix}$$

$$\dot{y}_{(t)} = \begin{Bmatrix} 0.03t \quad ,0 \le t \le 3.2 \\ 0.096 \quad ,3.2 < t \le 28.8 \\ -0.03t + 0.96 \, , \; 28.8 < t \le 32 \end{Bmatrix}$$

$$\ddot{y}_{(t)} = \begin{Bmatrix} 0.03 \quad ,0 \le t \le 3.2 \\ 0 \quad ,3.2 < t \le 28.8 \\ -0.03 \, , \; 28.8 < t \le 32 \end{Bmatrix}$$

Final rotation of the Robot:

The third trajectory will describe the final rotation of the robot after it has already reached the final position (x_f,y_f). The following will hold:

For the initial and final angular positions, the values are derived from Table 2, thus:

$$\theta_0 = 0.54 \text{ rad} \qquad \theta_f = \text{round}(\frac{5351}{3500}) = 1.5 \text{ rad}$$

For the maximum angular velocity ($\dot{\theta}_{max}$), which is given in the statement, the following holds:

$$\dot{\theta}_{max} = 30°/sec = 30 \cdot \frac{\pi}{180} = \frac{\pi}{6} \text{ rad/sec}$$

The mathematical formulation of the trajectory, will be as follows:

$$\theta_{(t)} = \begin{Bmatrix} \theta_0 + \frac{\ddot{\theta}}{2}t^2 \quad ,0 \le t \le t_b \\ \theta_0 + \frac{\ddot{\theta}}{2}t_b^2 + \ddot{\theta}t_b(t - t_b) \quad ,t_b < t \le t_f - t_b \\ \theta_f - \frac{\ddot{\theta}}{2}\left(t_f - t\right)^2,\ t_f - t_b < t \le t_f \end{Bmatrix}$$

For the moment in time $t = t_b$, it holds:

$$\dot{\theta}_{coast} = \frac{\theta_h - \theta_b}{t_h - t_b} = \ddot{\theta} \cdot t_b,$$

$$\theta_h = \frac{\theta_f + \theta_0}{2},$$

$$t_h = \frac{t_f}{2},$$

$$\boldsymbol{\theta_b = \theta_0 + \frac{1}{2} \cdot \ddot{\theta}} \cdot t_b^2$$

Solving for t_b, results in the following:

$$t_b = \frac{t_f}{2} - \frac{\sqrt{(\ddot{\theta}2 \cdot t_f^2 - 4 \cdot \ddot{\theta} \cdot (\theta_f - \theta_0))}}{2\ddot{\theta}}$$

Thus, it holds that:

$$\theta_{\left(\frac{t_f}{2}\right)} = \frac{\theta_f + \theta_0}{2} = \theta_0 + \frac{\ddot{\theta}}{2} t_b^2 + \ddot{\theta} t_b \left(\frac{t_f}{2} - t_b\right) \quad =>$$

$$\theta_0 + \theta_f = 2\theta_0 + \ddot{\theta} t_b^2 + \ddot{\theta} t_b t_f - 2\ddot{\theta} t_b^2 =>$$

$$\ddot{\theta} = \frac{\theta_f - \theta_0}{t_b(t_f - t_b)} \ (1)$$

I choose t_b = 0.1 · t_f and θ_f = 0.54 rad. From (1), it follows that:

$$\ddot{\theta} = \frac{\theta_f - \theta_0}{t_b(t_f - t_b)} = \frac{0.96}{0.09 t_f^2} = \frac{10.6}{t_f^2} \quad (2)$$

With the maximum angular velocity $\dot{\theta}_{max} = \frac{\pi}{6}$ rad/sec, it holds that:

$$\dot{\theta}_{\max} = \ddot{\theta}_{max} \cdot t_b \leftrightarrow \ddot{\theta}_{max} = \frac{\dot{\theta}_{\max}}{0.1 \cdot t_f} \leftrightarrow \ddot{\theta}_{max} = \frac{\frac{\pi}{6}}{0.1 \cdot t_f} \leftrightarrow \ddot{\theta}_{max} = \frac{\pi}{0.6 \cdot t_f} \text{ rad}/sec^2 \ (3)$$

However, it must simultaneously satisfy the following:

$$\left|\ddot{\theta}\right| \le \left|\ddot{\theta}_{\max.}\right| \xRightarrow{(2),(3)} \frac{10.6}{t_f^2} \le \frac{\pi}{0.6 t_f} \leftrightarrow \frac{t_f^2}{10.6} \ge \frac{0.6 t_f}{\pi} \leftrightarrow t_f \cdot \pi \ge 6.36 \leftrightarrow t_f \ge 2.02\ sec$$

For t_f = 2.02 sec and t_b = 0.202 sec, from relation (1), it holds that:

$$\ddot{\theta} \cong \frac{10.6}{4.1} = 2.58 \text{ rad/sec}^2$$

Therefore, the trajectory becomes:

$$\theta_{(t)} = \begin{cases} \theta_0 + \frac{\ddot{\theta}}{2}t^2 & ,0 \le t \le t_b \\ \theta_0 + \frac{\ddot{\theta}}{2}t_b^2 + \ddot{\theta}t_b(t - t_b) & ,t_b < t \le t_f - t_b \\ \theta_f - \frac{\ddot{\theta}}{2}\left(t_f - t\right)^2 & ,\ t_f - t_b < t \le t_f \end{cases}$$

$$\theta_{(t)} = \begin{cases} 0.54 + \frac{2.58}{2}t^2 & ,0 \le t \le 0.202 \\ 0.54 + \frac{2.58}{2}0.202^2 + 2.58 \cdot 0.202(t - 0.202) & ,0.202 < t \le 1.818 \\ 1.5 - \frac{2.58}{2}(2.02 - t)^2 & ,1.818 < t \le 2.02 \end{cases}$$

$$\theta_{(t)} = \begin{cases} 0.54 + 1.29t^2 & ,0 \le t \le 0.202 \\ 0.54 + 0.052 + 0.52t - 0.1 & ,0.202 < t \le 1.818 \\ 1.5 - 1.29(4.08 - 4.04t + t^2) & ,\ 1.818 < t \le 2.02 \end{cases}$$

$$\theta_{(t)} = \begin{cases} 1.29t^2 + 0.54 & ,0 \le t \le 0.202 \\ 0.52t + 0.49 & ,0.12 < t \le 1.08 \\ -1.29t^2 + 5.2t - 3.76 & ,\ 1.818 < t \le 2.02 \end{cases}$$

$$\dot{\theta}_{(t)} = \begin{cases} 2.58t & ,0 \le t \le 0.202 \\ 0.52 & ,0.12 < t \le 1.08 \\ -2.58t + 5.2 & ,\ 1.818 < t \le 2.02 \end{cases}$$

$$\ddot{\theta}_{(t)} = \begin{cases} 2.58 & ,0 \le t \le 0.202 \\ 0 & ,\ 0.12 < t \le 1.08 \\ -2.58 & ,\ 1.818 < t \le 2.02 \end{cases}$$

2.2 Implementation of trajectories_jackal.py

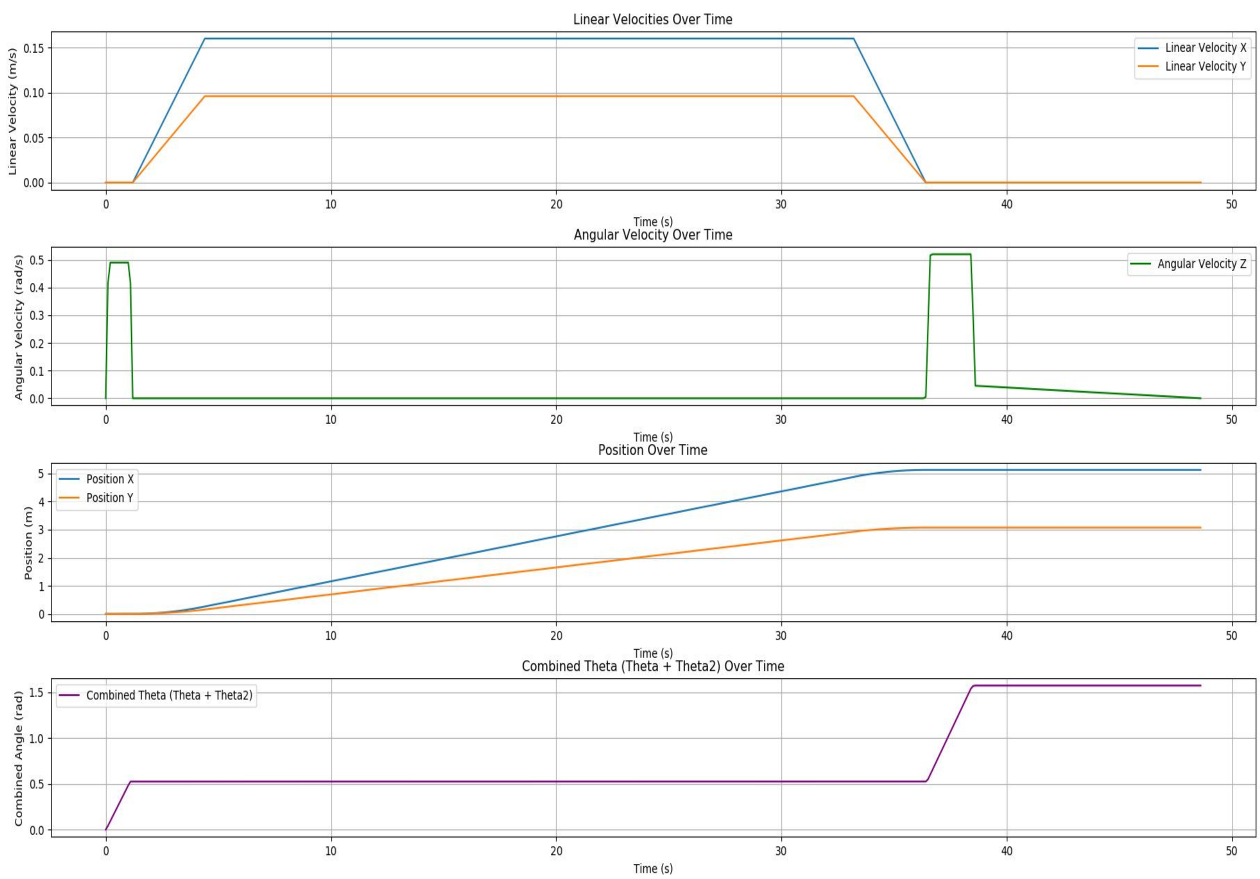
he development of the program trajectories_jackal.py assists in the movement of the wheeled robot and the recording of velocity and position data.

This program utilizes libraries such as rospy (for controlling ROS), std_srvs.srv (for calling empty services), matplotlib.pyplot (for displaying graphs), and geometry_msgs.msg (for velocity messages).

The program begins by importing the necessary libraries and then stores velocity and position data in lists. It also initializes the angles and positions xxx, yyy to store the trajectories. Following this, the function reset_gazebo_simulation_time is defined, which waits for the Gazebo simulation reset service to become available and then calls it to reset the simulation. There are various other functions that handle the movement and data recording of the robot.

The command that needs to be executed in the terminal for the correct operation of the robot is:

```
python3 trajectories_jackal.py
```



The first diagram illustrates the linear velocity along the X and Y axes over time. The velocity along the X axis (represented by the blue line) increases sharply at the beginning, reaching approximately 0.15 m/s by the 5th second and remains constant until the 40th second, where it decreases sharply to zero. Similarly, the velocity along the Y axis (represented by the orange line) also increases initially but reaches a value of 0.1 m/s, maintaining this steady value until the 40th second, where it also drops to zero.

The second diagram depicts the angular velocity along the Z axis (represented by the green line). The angular velocity shows a sharp increase at first, reaching approximately 0.5 rad/s by the 2nd second, and then suddenly drops to zero. It remains zero until the 35th second, at which point there is another spike to the same value, before ultimately returning to zero.

The third diagram shows the position along the X and Y axes over time. The position along the X axis (represented by the blue line) steadily increases from zero to approximately 5 meters by the 40th second and remains constant until the end of the measurement. The position along the Y axis (represented by the orange line) also increases steadily but at a lower rate, reaching about 3 meters in the same timeframe, and remains constant until the end.

The fourth diagram presents the combined angle Theta (the sum of two angles) over time. This angle increases sharply from zero to 0.5 rad at the beginning and remains constant until the 35th second, where it sharply increases to 1.0 rad, and finally stabilizes at 1.5 rad until the end of the measurement.

3.0 Implementation of move_rrbot_joint.py

The program begins with the import of necessary libraries, such as sys (used for accessing command line arguments and exiting the program), rospy (a Python library for ROS that enables communication with other nodes in the ROS network), numpy (used for numerical operations, specifically for creating arrays of joint angles), matplotlib.pyplot (used for creating graphical representations of joint positions as a function of time), and std_msgs.msg (contains the Float64 message type, which is used for publishing the positions of the joints).

Then, the program initializes three lists to store data: time_data, q1_data, and q2_data. These are used to record the time and positions of the two joints during the program's execution. Next, a ROS node named "robot_move2" is initialized, which is anonymous to allow multiple instances of nodes with the same name to run without conflict.

The command that needs to be executed in the terminal for the proper functioning of the arm is:

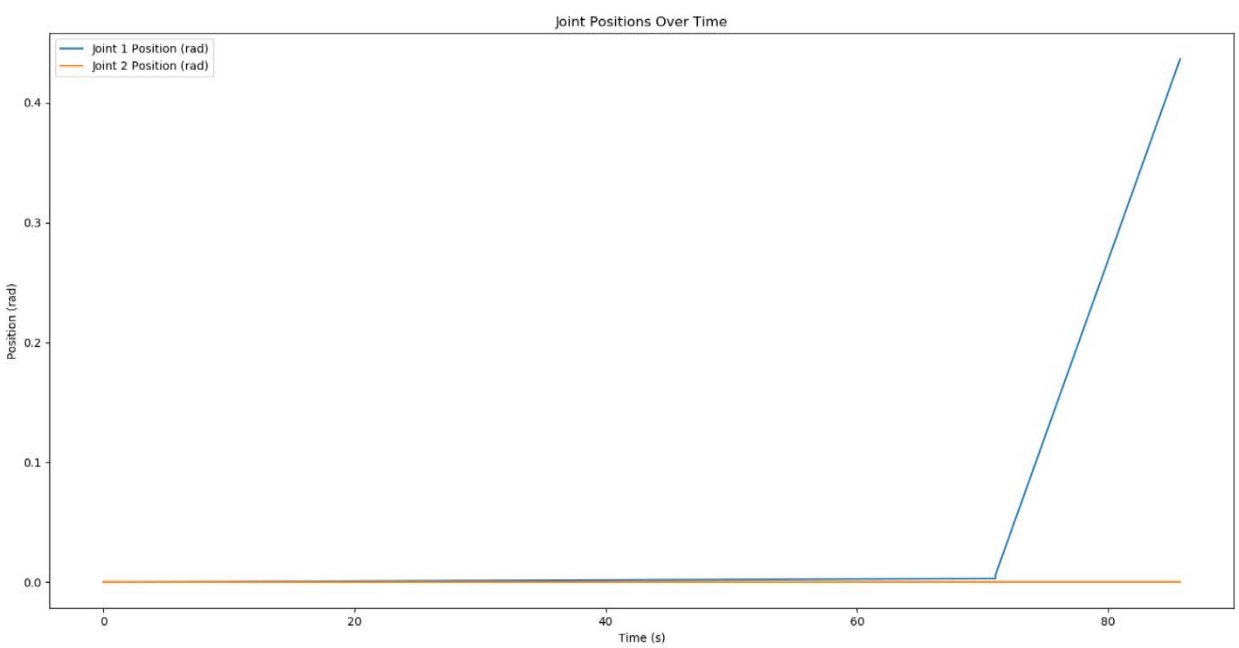
```
python3 move_rrbot_joint.py <joint_type> <start_angle_degrees> <end_angle_degrees>
```

Execution of the above command for:

joint_type = q1

start_angle_degrees = 0°

end_angle_degrees = 25°



The graph shows the positions of the two joints of the robotic arm as a function of time. The horizontal axis (X) represents time in seconds, while the vertical axis (Y) represents the positions of the joints in radians (rad).

The blue curve represents the position of Joint 1 (Joint 1 Position) in radians. We observe that the position of Joint 1 remains constant at 0 for a long period and then increases sharply after about 80 seconds, reaching approximately 0.45 rad. This sudden increase indicates that Joint 1 started moving after a long period of inertia.

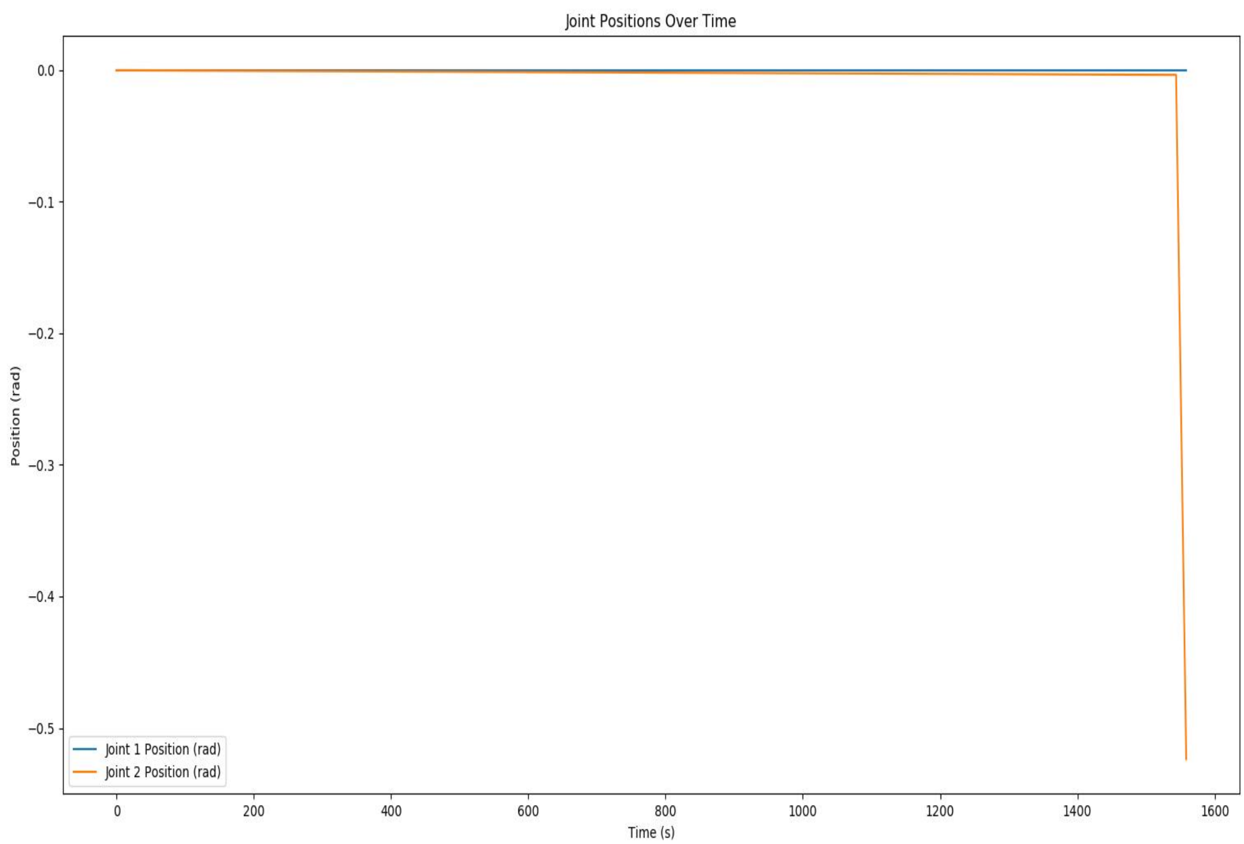
The orange curve represents the position of Joint 2 (Joint 2 Position) in radians. We notice that the position of Joint 2 remains constant at 0 throughout the duration depicted in the graph. This indicates that Joint 2 did not move during the execution of the program and remained in a stable position.

Execution of the above command for:

joint_type = q2

start_angle_degrees = 0°

end_angle_degrees = -30°



The graph shows the positions of the two joints of the robotic arm as a function of time. The horizontal axis (X) represents time in seconds, while the vertical axis (Y) represents the positions of the joints in radians (rad).

The blue curve represents the position of Joint 1 (Joint 1 Position) in radians. We observe that the position of Joint 1 remains constant at 0 for the entire duration.

The orange curve represents the position of Joint 2 (Joint 2 Position) in radians. We notice that the position of Joint 2 also remains constant at 0 for a long time. At the end of the graph, the position of Joint 2 decreases sharply, reaching approximately -0.5 rad. This sudden change indicates that Joint 2 also started moving after a long period of inertia.

Overall, the graph indicates that the joints remained inactive for most of the time and moved abruptly at the end of the observation.

3.1 Implementation of trajectories for the Joints

For the development of the solution, we will use the cubic polynomial method, taking into account the parameters provided in Table 4 and the registration number (AM=5351) for determining the time parameters. Therefore, the initial conditions will be as follows:

Initial Joint Angles : ($q_{1,0}$, $q_{2,0}$),

Final Joint Angles : ($q_{1,f}$, $q_{2,f}$),

Maximum Angular Velocity of Joint 1 : 9°/sec

Maximum Angular Velocity of Joint 2 : 11°/sec

The time parameters for the two trajectories that we are required to calculate are defined as follows:

Total time of each path : t_f

<u>First Joint:</u>

The first trajectory will describe the movement of the first joint of the robot, with the following conditions:

For the initial and final velocities, the following applies:

$$\dot{q}_1(0) = 0 \text{ rad/sec} \qquad \dot{q}_1(t_f) = 0 \text{ rad/sec}$$

For the initial and final angular positions, the values are derived from Table 4, thus:

$$q_{1,0} = 0 \text{ rad} \qquad q_{1,f} = \text{round}(\tfrac{5351}{80}) \cdot \tfrac{\pi}{180} = 0.37\pi \text{ rad}$$

For the maximum angular velocity ($\dot{q}_{1max}$), which is given in the problem statement, the following applies:

$$\dot{q}_{1max} = 9 \cdot \frac{\pi}{180} = \frac{\pi}{20} \text{ rad/sec}$$

We assure that $q_{1(t)}$ is a third-degree polynomial:

$$q_{1\,(t)} = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\dot{q}_{1(t)} = a_1 + 2a_2 t + 3a_3 t^2$$

$$\ddot{q}_{1(t)} = 2a_2 + 6a_3 t$$

We find the coefficients α_i using the initial and final conditions (velocities and angles):

$$a_0 = q_{1,0} = 0 \quad , \qquad a_1 = 0 \qquad (1)$$

$$\alpha_2 = \frac{3(q_{1,f} - q_{1,0})}{t_f^2} \leftrightarrow a_2 = 3\,\frac{0.37 \cdot \pi}{t_f^2} \leftrightarrow a_2 = \frac{1.11 \cdot \pi}{t_f^2} \qquad (2)$$

$$\alpha_3 = -\frac{2(q_{1,f} - q_{1,0})}{t_f^3} \leftrightarrow a_3 = -2\,\frac{0.37 \cdot \pi}{t_f^3} \leftrightarrow a_3 = \frac{-0.74 \cdot \pi}{t_f^3} \qquad (3)$$

To follow the trajectory, it must hold that at the midpoint (t=$\frac{t_F}{2}$), the condition $\dot{q}_{1(t)} \leq \dot{q}_{1,max}$ must be satisfied:

$$|\dot{q}_{1\left(\frac{t_f}{2}\right)}| \leq |\frac{\pi}{20}| \leftrightarrow |\frac{2a_2t_f}{2} + 3a_3\left(\frac{t_f}{2}\right)^2| \leq |\frac{\pi}{20}|$$

$$\leftrightarrow \left|\frac{1.11\cdot\pi}{t_f} + \frac{3}{4}\left(-\frac{0.74\cdot\pi}{t_f}\right)\right| \leq |\frac{\pi}{20}| \leftrightarrow |\frac{1.11\cdot\pi}{t_f} - \frac{0.555\cdot\pi}{t_f}| \leq |\frac{\pi}{20}| \leftrightarrow$$

$$t_f \geq 11.1 \quad (4)$$

Second Joint:

The second trajectory will describe the motion of the robot's second joint, and the following will apply:

For the initial and final velocities, it holds that:

$$\dot{q}_2(0) = 0 \text{ rad/sec} \qquad \dot{q}_2(t_f) = 0 \text{ rad/sec}$$

For the initial and final angular positions, the values come from Table 4, so it holds that:

$$q_{2,0} = 0 \text{ rad} \qquad q_{2,f} = \text{round}(\frac{5351}{120}) \cdot \frac{\pi}{180} = -\frac{\pi}{4} \text{ rad}$$

For the maximum angular velocity ($\dot{q}_{2max}$), which is given in the statement, it holds that:

$$\dot{q}_{2max} = 11 \cdot \frac{\pi}{180} = \frac{\pi}{16.3} \text{ rad/sec}$$

Assuming that $q_{1(t)}$ is a third-degree polynomial:

$$q_{2\ (t)} = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\dot{q}_{2(t)} = a_1 + 2a_2 t + 3a_3 t^2$$

$$\ddot{q}_{2(t)} = 2a_2 + 6a_3 t$$

We find the coefficients α_i using the initial and final conditions (velocities and angles):

$$a_0 = q_{2,0} = 0 \quad , \qquad a_1 = 0 \qquad (5)$$

$$\alpha_2 = \frac{3(q_{2,f} - q_{2,0})}{t_f^2} \leftrightarrow a_2 = 3\frac{-\frac{\pi}{4}}{t_f^2} \leftrightarrow a_2 = -\frac{3\pi}{4t_f^2} \qquad (6)$$

$$\alpha_3 = -\frac{2(q_{2,f} - q_{2,0})}{t_f^3} \leftrightarrow a_3 = -2\frac{-\frac{\pi}{4}}{t_f^3} \leftrightarrow a_3 = \frac{\pi}{2t_f^3} \qquad (7)$$

To follow the trajectory, it must hold that at the midpoint (t=$\frac{t_F}{2}$), the condition $\dot{q}_{2(t)} \leq \dot{q}_{2,max}$ must be satisfied:

$$|\dot{q}_{2\left(\frac{t_f}{2}\right)}| \leq |\frac{\pi}{16.3}| \leftrightarrow |\frac{2a_2 t_f}{2} + 3a_3\left(\frac{t_f}{2}\right)^2| \leq |\frac{\pi}{16.3}|$$

$$\leftrightarrow |\frac{-2 \cdot 3\pi}{8t_f} + \frac{3\pi}{8t_f}| \leq |\frac{\pi}{16.3}| \leftrightarrow |\frac{-3\pi}{8t_f}| \leq |\frac{\pi}{16.3}| \leftrightarrow$$

$$t_f \geq \frac{48.9\pi}{8\pi} \leftrightarrow t_f \geq 6.11 \text{ sec } (8)$$

Therefore, to satisfy relationships (4) and (8), the following must hold:

t_f = 11 sec

Finally, from the above relationships, the following will arise:

$$(2): a_2 =- \frac{1.11 \cdot \pi}{121} = 0.03$$

$$(6): a_2 =- \frac{3 \cdot \pi}{484} = 0.02$$

$$(3): a_3 = \frac{-0.74 \cdot \pi}{1331} =- 0.002$$

$$(7): a_3 = \frac{\pi}{2662} = 0.001$$

Therefore, the trajectories of the joints are formulated as follows:

$$q_{1(t)} = 0.03t^2 - 0.002t^3$$

$$\dot{q}_{1(t)} = 0.06t - 0.006t^2$$

$$\ddot{q}_{1(t)} = 0.06 - 0.012t$$

$$q_{2(t)} =- 0.02t^2 + 0.001t^3$$

$$\dot{q}_{2(t)} =- 0.04t + 0.003t^2$$

$$\ddot{q}_{2(t)} =- 0.04 + 0.006t$$

Still, with the selection of t_f =6 sec, we can verify that the values of the angular velocities of the joints are less than their respective maximum values:

$$\dot{q}_{1(t_f)} = 0.06 \cdot 11 - 0.006 \cdot 121 = 0.66 - 0.726 \leftrightarrow$$

$$\dot{q}_{1(t_f)} =- 0.066 \frac{rad}{sec} \cong 0 \leq \dot{q}_{1,\max.} = \frac{\pi}{20} \text{ rad/sec}$$

$$\dot{q}_{2(t_f)} =- 0.04 \cdot 11 - 0.003 \cdot 121 =- 0.44 + 0.363 \leftrightarrow$$

$$\dot{q}_{2(t_f)} = 0.077 \frac{rad}{sec} \cong 0 \leq \dot{q}_{2,\max.} = \frac{\pi}{16.3} rad/sec$$

3.2 Implementation of trajectories_rrbot.py

The development of the program ‘trajectories_rrbot.py’ aids in the movement of the robotic arm's joints and the recording of position and velocity data. This program uses libraries such as rospy (for controlling ROS), std_srvs.srv (for calling empty services), matplotlib.pyplot (for displaying graphs), and std_msgs.msg (for position messages).

The program begins by importing the necessary libraries and then stores time, position, and velocity data in lists. These lists include time_data, joint1_positions, joint2_positions, joint1_velocities, and joint2_velocities.

Next is the definition of the reset_gazebo_simulation_time function, which waits for the Gazebo simulation reset service to become available and then calls it to reset the simulation. This function ensures that the simulation starts from the beginning each time the program is executed.

The calculate_velocities function calculates the velocities of the joints from the position and time data. It uses the difference in position divided by the difference in time to compute the velocity of each joint.

The function plot_data is used to visualize the data. Initially, the velocities of the joints are calculated using the calculate_velocities function. Then, a graph is created with two subplots: the first shows the positions of the joints as a function of time, and the second shows the velocities of the joints as a function of time. This graph aids in the visual analysis of the joints' movement.

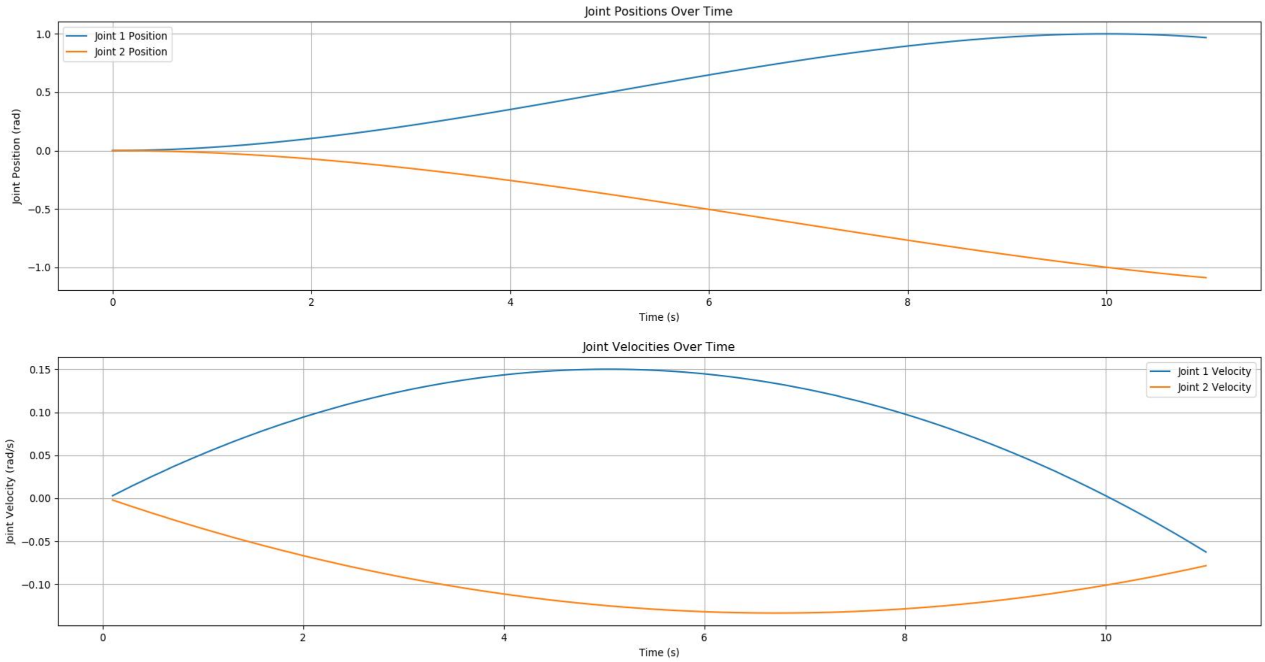
In the main function, a new ROS node named "robot_trajectory2" is initialized. Then, the reset_gazebo_simulation_time function is called to reset the Gazebo simulation.

The movement of the joints lasts for 6 seconds. The function calculates the desired trajectories for the joints using polynomial equations and publishes the positions to the corresponding joints. The time and position data are stored in the lists time_data, joint1_positions, and joint2_positions.

Finally, when the movement is completed, the plot_data function is called to visualize the data. This process allows for the analysis of the robotic arm's joint movement over time.

The command that needs to be executed in the terminal for the proper functioning of the arm is:

```
python3 trajectories_rrbot.py
```



The first diagram illustrates the positions of the two joints of the robotic arm as a function of time. The horizontal axis (X) represents time in seconds, while the vertical axis (Y) represents the positions of the joints in radians (rad).

The blue curve represents the position of joint 1 (Joint 1 Position). We observe that the position of joint 1 gradually increases from 0 to approximately 1 rad over the course of 11 seconds, indicating a continuous movement of the joint towards the desired position.

The orange curve represents the position of joint 2 (Joint 2 Position). We see that the position of joint 2 gradually decreases from 0 to approximately -1 rad during the 11 seconds, showing that the joint steadily moves towards the desired negative position.

The second diagram depicts the velocities of the two joints of the robotic arm as a function of time. The horizontal axis (X) represents time in seconds, while the vertical axis (Y) represents the velocities of the joints in radians per second (rad/s).

The blue curve represents the velocity of joint 1 (Joint 1 Velocity). We observe that the velocity of joint 1 increases initially, reaching its maximum value of approximately 0.15 rad/s, and then gradually decreases as it approaches the end of the movement.

The orange curve represents the velocity of joint 2 (Joint 2 Velocity). We observe that the velocity of joint 2 decreases initially, reaching its maximum negative value of approximately -0.13 rad/s, and then gradually increases as it approaches the end of the movement.

Overall, the graphs show the continuous movement and changes in velocity of the robot arm's joints during the execution of the program, allowing for a visual analysis of the control system's performance.