

# Universidade Federal do Rio Grande do Norte

Programa de Pós-graduação em Engenharia Elétrica e de Computação  
EEC1507 - Sistemas Robóticos Autônomos

## RELATÓRIO

1º PROJETO DE SISTEMAS ROBÓTICOS AUTÔNOMOS - META 3

**Alunos:**

Edel Mary Quinn de Oliveira Figueiredo

Luís Gabriel Pereira Condados

Samigo Ricardo de Oliveira Silva

**Professor orientador:** Pablo Javier Alsina

Natal-RN  
2020

# Universidade Federal do Rio Grande do Norte

Programa de Pós-graduação em Engenharia Elétrica e de Computação  
EEC1507 - Sistemas Robóticos Autônomos

## RELATÓRIO

Relatório apresentado à disciplina de EEC1507- Sistemas Robóticos Autônomos, correspondente a 1º unidade do semestre 2020.2, sob orientação do **Prof. Pablo Javier Alsina**.

Alunos:

Edel Mary Quinn de Oliveira Figueiredo  
Luís Gabriel Pereira Condados  
Samigo Ricardo de Oliveira Silva

Natal-RN  
2020

# Sumário

<b>Sumário</b> . . . . .	<b>3</b>	
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>1</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> . . . . .	<b>1</b>
2.1	Modelo Cinemático do Robô . . . . .	1
2.2	Polinômio de Terceiro Grau . . . . .	3
2.2.1	Primeiro Caso Especial . . . . .	4
2.2.2	Segundo Caso Especial . . . . .	5
2.2.3	Terceiro Caso Especial . . . . .	5
2.2.4	Caso Geral . . . . .	5
2.3	GERAÇÃO DE CAMINHO . . . . .	6
2.4	GERAÇÃO DE TRAJETÓRIA . . . . .	6
2.4.1	Perfil de Velocidade . . . . .	6
2.4.2	Geração de Trajetória com Polinômio de Terceiro Grau . . . . .	8
2.5	CONTROLE ESTABILIZANTE . . . . .	8
2.6	CONTROLE DE CAMINHO . . . . .	10
2.7	CONTROLE DE TRAJETÓRIA . . . . .	12
2.7.1	Controle de Trajetória por Linearização da Realimentação Dinâmica . . . . .	12
<b>3</b>	<b>METODOLOGIA</b> . . . . .	<b>13</b>
<b>4</b>	<b>RESULTADOS E CONCLUSÕES</b> . . . . .	<b>15</b>
4.1	CONTROLE ESTABILIZANTE . . . . .	15
4.2	CONTROLE SEGUIDOR DE CAMINHO . . . . .	19
4.3	CONTROLE DE TRAJETÓRIA . . . . .	23
	<b>REFERÊNCIAS</b> . . . . .	<b>27</b>
<b>5</b>	<b>ANEXO 1</b> . . . . .	<b>28</b>

# 1 INTRODUÇÃO

O controle cinemático de um robô pode apresentar problemas complexos de resolução. Alguns robôs são naturalmente instáveis e podem possuir restrições no que diz respeito à velocidade do robô. Neste caso, os controladores podem ser aplicados de acordo com um objetivo específico: controladores estabilizantes, seguidores de trajetória e seguidores de caminho. Foi implementado neste trabalho um gerador de caminho baseado em polinômios interpoladores de 3º grau, baseado no trabalho de (8) para robô móvel com acionamento diferencial e restrição não-holonômica, o mesmo interpolador foi utilizado na geração de trajetória.

## 2 REFERENCIAL TEÓRICO

### 2.1 MODELO CINEMÁTICO DO ROBÔ

Para a construção da modelagem do robô com rodas de dois graus de liberdade, conhecida também na literatura como roda padrão, temos que levar em consideração a restrição de rolamento puro, onde todo o movimento da roda tem que ser acompanhado pela rotação correspondente da roda, enquanto que para a restrição de derrapagem lateral todo o movimento deve ser restrito ao plano da roda, ou seja, a roda não pode ser movimentada em direção ao eixo. Com a escolha do robô móvel com acionamento diferencial, o uso da relação entre a velocidade das rodas pode ser obtido pelo giro do robô com o raio  $r$  também pode ser obtido por análise simples a partir da Figura 1:

Analizando a natureza do movimento circular na Figura 1 podemos observar as seguintes relações,

$$\omega \left( r - \frac{b}{2} \right) = v_e \quad (1)$$

$$\omega \left( r + \frac{b}{2} \right) = v_d \quad (2)$$

Somando  $v_e$  e  $v_d$  temos:

$$v = \frac{(\omega_d + \omega_e).r_w}{2} \quad (3)$$

E subtraindo  $v_e$  e  $v_d$  podemos chegar em:

$$\omega = \frac{(\omega_d - \omega_e).r_w}{b} \quad (4)$$

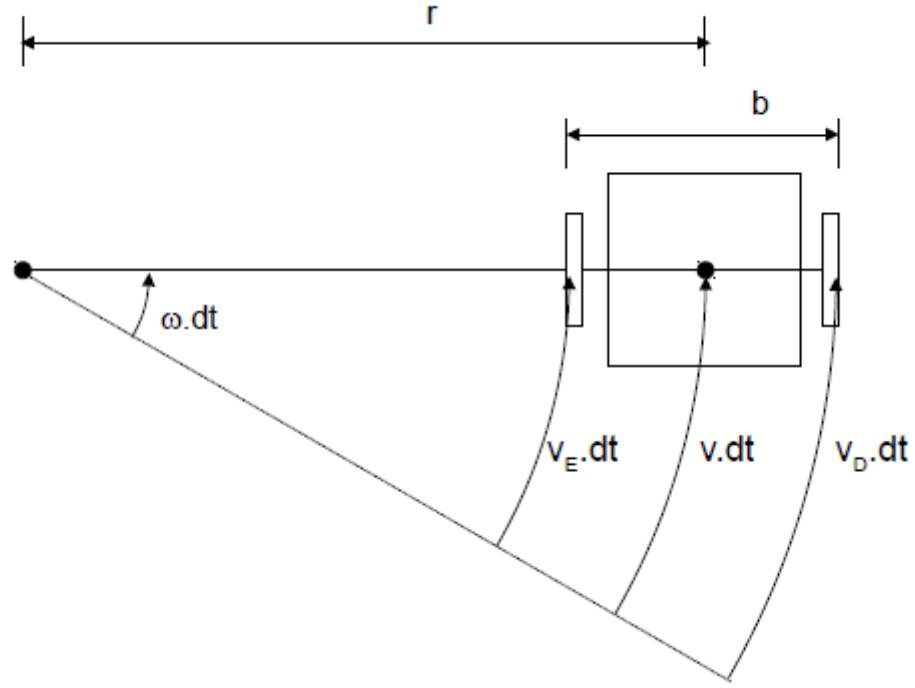


Figura 1 – Movimentos Infinitesimais para o modelo de robô utilizado.

Podemos também relacionar as velocidades das rodas com o raio de giro instantâneo do robô

$$\frac{\omega_e}{\omega_d} = \frac{r - \frac{b}{2}}{r + \frac{b}{2}} \quad (5)$$

Por fim temos a **função de cinemática direta do robô**, considerando o espaço de configuração:  $[x \ y \ \theta]^T$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{(\omega_d + \omega_e) \cdot r_w}{2} \cdot \cos \theta \\ \frac{(\omega_d + \omega_e) \cdot r_w}{2} \cdot \sin \theta \\ \frac{(\omega_d - \omega_e) \cdot r_w}{b} \end{bmatrix} \quad (6)$$

Podemos também achar a matriz de transformação que relaciona as velocidades das rodas com a velocidade linear e angular, fazendo isso, podemos chegar ao seguinte resultado:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} (r_d/2) & (r_e/2) \\ (r_d/b) & -(r_e/2) \end{bmatrix} \cdot \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} \quad (7)$$

E a relação inversa:

$$\begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} = \begin{bmatrix} (1/r_d) & (b/2r_d) \\ (1/r_e) & -(b/2r_e) \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (8)$$

## 2.2 POLINÔMIO DE TERCEIRO GRAU

Uma polinômio de terceiro grau parametrizado com relação a  $\lambda$  possuí as seguintes equações:

$$x(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 \quad (9)$$

$$y(\lambda) = b_0 + b_1\lambda + b_2\lambda^2 + b_3\lambda^3 \quad (10)$$

$$\theta(\lambda) = \tan^{-1} \left( \frac{dy/d\lambda}{dx/d\lambda} \right) = \tan^{-1} \frac{b_1 + 2b_2\lambda + 3b_3\lambda^2}{a_1 + 2a_2\lambda + 3a_3\lambda^2} \quad (11)$$

$\lambda$  varia de 0 à 1, do ponto inicial até o ponto final, ou seja  $\lambda = 0$  para  $(x_i, y_i, \theta_i)$  e  $\lambda = 1$  para  $(x_f, y_f, \theta_f)$ .

Condições de contorno:

$$(x(0), y(0), \theta(0)) = (x_i, y_i, \theta_i)$$

$$(x(1), y(1), \theta(1)) = (x_f, y_f, \theta_f)$$

Das restrições acima temos que,

$$\begin{cases} x(0) = a_0 \\ y(0) = b_0 \\ \alpha(0) = \tan \theta(0) = \frac{b_1}{a_1} \\ x(1) = a_0 + a_1 + a_2 + a_3 \\ y(1) = b_0 + b_1 + b_2 + b_3 \\ \alpha(1) = \tan \theta(1) = \frac{b_1 + 2b_2 + 3b_3}{a_1 + 2a_2 + 3a_3} \end{cases}$$

O sistema acima pode ser simplificado para,

$$\begin{cases} \alpha_i a_1 - b_1 = 0 \\ a_1 + a_2 + a_3 = \Delta x \\ b_1 + b_2 + b_3 = \Delta y \\ \alpha_f a_1 + 2\alpha_f a_2 + 3\alpha_f a_3 - b_1 - 2b_2 - 3b_3 = 0 \end{cases}$$

onde  $\alpha_i = \tan \theta(0)$ ,  $\alpha_f = \tan \theta(1)$ ,  $\Delta x = x_f - x_i$  e  $\Delta y = y_f - y_i$ .

Reescrevendo estas equações para o formato matricial e realizando um pivotamento parcial, chega-se a:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3d_f & 2d_f & d_f \\ 0 & 0 & 1 & 0 & 0 & -d_i \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ a_3 \\ a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} \Delta y \\ 3\Delta y \\ 0 \\ \Delta x \end{bmatrix}$$

Devido ao sistema possuir 8 incógnitas e apenas 6 restrições, sendo assim o sistema têm 2 variáveis que podem assumir qualquer valor. Escolhe-se as variáveis  $a_1$  e  $a_2$  como variáveis arbitrárias, pois o cálculo dos demais a partir destes não resulta em divisões, o que pode causar singularidade.

Resolvendo o sistema com  $a_1$  e  $a_2$  arbitrárias tem-se:

$$\begin{cases} b_3 = 3\alpha_f \Delta x - 2\Delta y - \alpha_f a_2 - (2\alpha_f - \alpha_i) a_1 \\ b_2 = 3\Delta y - 3\alpha_f \Delta x + \alpha_f a_2 - 2(\alpha_i - \alpha_f) a_1 \\ b_1 = \alpha_i a_1 \\ a_3 = \Delta x - a_2 - a_1 \end{cases} \quad (12)$$

As equações em 12 representam o caso geral para o polinômio de terceiro grau, embora respeite a restrição não-holônoma, possuindo singularidades para  $\theta = \pm\pi/2$ , estes casos especiais são tratados como em (8), bem como as otimizações para escolher os coeficientes livres.

Para trabalhar com uma folga numérica em torno da singularidade define-se aqui uma variável  $\delta$  para representar uma pequena variação.

### 2.2.1 PRIMEIRO CASO ESPECIAL

Se  $\theta_i \in [(\pi/2 - \delta), (\pi/2 + \delta)]$  e  $\theta_f \in [(\pi/2 - \delta), (\pi/2 + \delta)]$

$$\begin{cases} b_1 = \Delta y \text{ (coef. livre)} \\ b_2 = 0 \text{ (coef. livre)} \\ a_0 = x_i \\ a_1 = 0 \\ a_2 = 3\Delta x \\ a_3 = -2\Delta x \\ b_0 = y_i \\ b_3 = \Delta y - b_1 - b_2 \end{cases} \quad (13)$$

### 2.2.2 SEGUNDO CASO ESPECIAL

Apenas  $\theta_i \in [(\pi/2 - \delta), (\pi/2 + \delta)]$

$$\begin{cases} a_3 = -\Delta x/2 \text{ (coef. livre)} \\ b_3 = \text{qualquer (coef. livre)} \\ a_0 = x_i \\ a_1 = 0 \\ a_2 = \Delta x - a_3 \\ b_0 = y_i \\ b_1 = 2(\Delta y - \alpha_f \Delta x) - \alpha_f a_3 + b_3 \\ b_2 = (2\alpha_f \Delta x - \Delta y) + \alpha_f a_3 - 2b_3 \end{cases} \quad (14)$$

### 2.2.3 TERCEIRO CASO ESPECIAL

Se  $\theta_f \in [(\pi/2 - \delta), (\pi/2 + \delta)]$

$$\begin{cases} a_1 = 3\Delta x/2 \text{ (coef. livre)} \\ b_2 = \text{qualquer (coef. livre)} \\ a_0 = x_i \\ a_2 = 3\Delta x - 2a_1 \\ a_3 = a_1 - 2\Delta x \\ b_0 = y_i \\ b_1 = \alpha_i a_1 \\ b_3 = \Delta y - \alpha_i a_1 - b_2 \end{cases} \quad (15)$$

### 2.2.4 CASO GERAL

$$\begin{cases} a_1 = \Delta x \text{ (coef. livre)} \\ a_2 = 0 \text{ (coef. livre)} \\ a_0 = x_i \\ a_3 = \Delta x - a_1 - a_2 \\ b_0 = y_i \\ b_1 = \alpha_i a_1 \\ b_2 = 3(\Delta y - \alpha_f \Delta x) + 2(\alpha_f - \alpha_i) + \alpha_f a_2 \\ b_3 = 3\alpha_f \Delta x - 2\Delta y - (2\alpha_f - \alpha_i)a_1 - \alpha_f a_2 \end{cases} \quad (16)$$

## 2.3 GERAÇÃO DE CAMINHO

Neste trabalho foi utilizado um gerador de caminho baseado na utilização do polinômio de grau 3, como apresentado anteriormente, ou seja, dados uma configuração inicial e final  $((x_i, y_i, \theta_i)$  e  $(x_f, y_f, \theta_f)$ ) calcula-se o polinômio que passa por estes pontos para  $\lambda$  in  $[0, 1]$ , com os parâmetros do polinômio calculados, gera-se o caminho variando  $\lambda$  de zero(0) a um(1) de forma discreta, para se ter os pontos  $(x, y)$  que irão compor o caminho desejado.

## 2.4 GERAÇÃO DE TRAJETÓRIA

A trajetória além de possuir um caminho definido, também possui um perfil de velocidade definido, fazendo com que a trajetória gere referências de posição e velocidade em função do tempo, esta é a diferente entre uma trajetória e um caminho. Neste trabalho foi utilizado um perfil de velocidade cossenoidal e foi realizado dois tipos de caminhos paramétricos, as subseções seguintes tratarão de explicar cada uma destas.

### 2.4.1 PERFIL DE VELOCIDADE

O perfil de velocidade utilizado foi o perfil cossenoidal, pois além ser derivável no tempo, é um perfil suave e de simples manipulação analítica. Podemos ver na imagem a seguir o perfil de velocidade em função do tempo, bem como sua primeira derivada (aceleração).

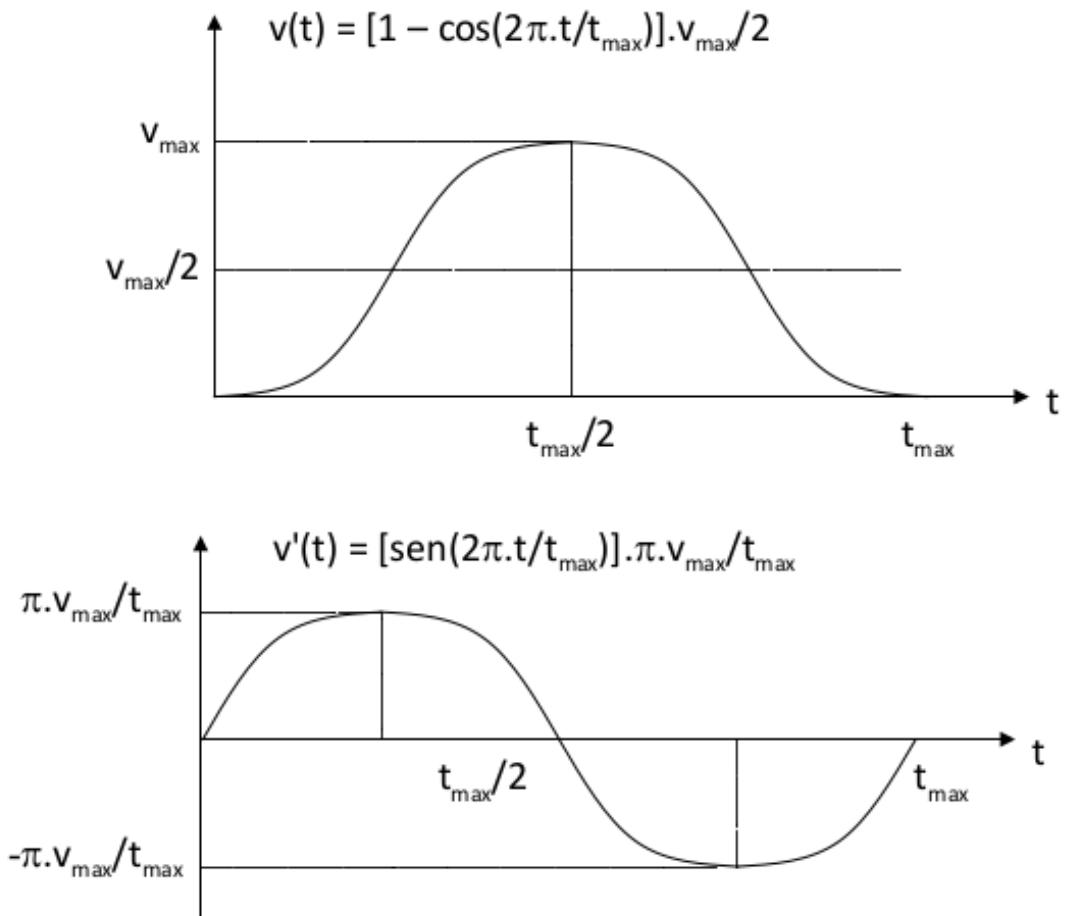


Figura 2 – Perfil de Velocidade Cosenoidal

Perfil de velocidade cossenoidal:

$$v(t) = [1 - \cos(2\pi t/t_{max})] \frac{v_{max}}{2} \quad (17)$$

Aceleração,

$$\dot{v}(t) = [\sin(2\pi t/t_{max})] \frac{\pi v_{max}}{t_{max}} \quad (18)$$

Relação entre  $t_{max}$  e  $v_{max}$ :

$$\begin{aligned} L &= \int_0^{t_{max}} v(t) dt = \int_0^{t_{max}} [1 - \cos(2\pi t/t_{max})] \frac{v_{max}}{2} dt \\ L &= \frac{v_{max} t_{max}}{2} \end{aligned}$$

Portando,

$$v_{max} = \frac{2L}{t_{max}} \quad (19)$$

#### 2.4.2 GERAÇÃO DE TRAJETÓRIA COM POLINÔMIO DE TERCEIRO GRAU

Reparametrizando o polinômio de terceiro grau de  $\lambda(s)$  para  $\lambda(t)$  utilizando o perfil de velocidade cossenoideal:

$$dl = \sqrt{(dx)^2 + (dy)^2}$$

$$dl = \sqrt{(Dx)^2 + (Dy)^2} \cdot d\lambda$$

Portando,

$$\frac{dl}{d\lambda} = \frac{1}{\sqrt{(Dx)^2 + (Dy)^2}} \quad (20)$$

A velocidade linear pode ser escrita como,

$$v(t) = \frac{dl}{dt} = \frac{dl}{d\lambda} \frac{d\lambda}{dt} = \sqrt{(Dx)^2 + (Dy)^2} \frac{d\lambda}{dt}$$

Assim temos que  $dl/dt$ :

$$\frac{d\lambda}{dt} = \frac{v(t)}{\sqrt{(Dx)^2 + (Dy)^2}} \quad (21)$$

O expressão acima pode ser integrada, caso tenha-se o perfil de velocidade, de modo a reparametrizar a trajetória, ou seja,  $\lambda(t)$ .

#### 2.5 CONTROLE ESTABILIZANTE

Para ser mais exato, foi utilizado neste trabalho um controle estabilizante por referência variável, baseado no trabalho de VIEIRA(9). O controlador baseia-se em técnicas de desacoplamento, ou seja, separar em dois controles independentes e faz uso de técnicas de controle linear monovariável.

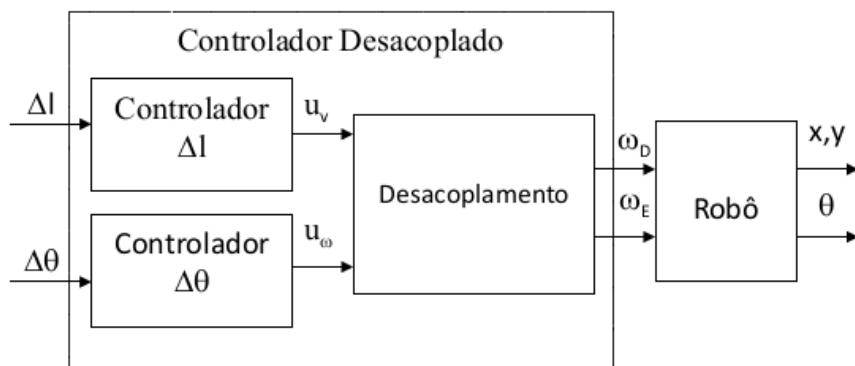


Figura 3 – Controladores Desacoplados para  $l(t)$  e  $\theta(t)$

Sendo  $u_v$  a entrada virtual associada a  $\Delta l$ (ou  $v(t)$ ) e  $u_\omega$  a entrada virtual associada à  $\Delta\theta$  (ou  $\omega(t)$ ).

O erro  $\Delta l$  é formulado usando-se referencial variável, como em (9), para sempre ser tratado como um caso trivial. Os casos especiais são:

1. Caso esteja na posição desejada, girar até a orientação desejada
2. Quando na orientação desejada, andar em frente para atingir a posição desejada na direção da orientação atual.

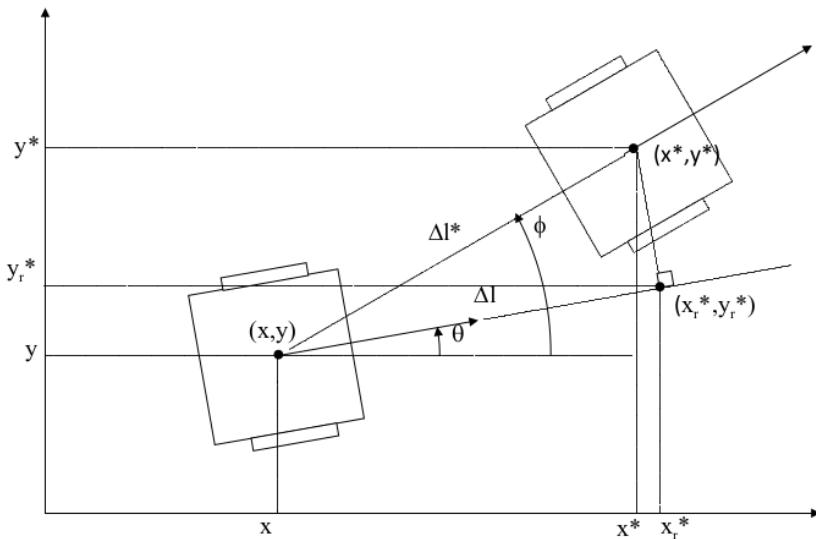


Figura 4 – Ilustração da formulação dos erros linear e angular com referencial variável.

Os erros de orientação e de deslocamento que servirão de entrada para o controlador estabilizante será:

$$\Delta\theta = \theta^* - \theta \quad (22)$$

$$\Delta l = \Delta l^* \cdot \cos(\Delta\theta) \quad (23)$$

Sendo,

$$\Delta x = x^* - x$$

$$\Delta y = y^* - y$$

$$\Delta\theta^* = \tan^{-1}(\Delta y / \Delta x)$$

$$\Delta l^* = \sqrt{\Delta x^2 + \Delta y^2}$$

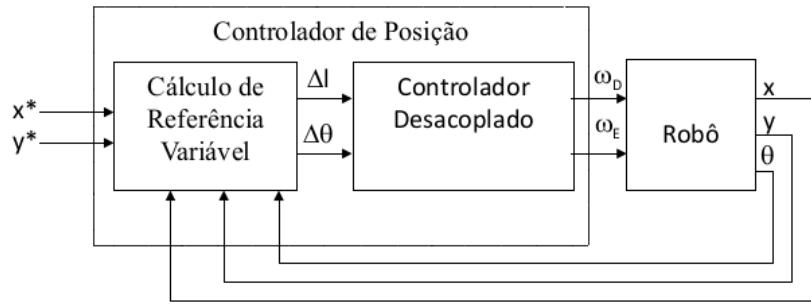


Figura 5 – Diagrama do sistema de controle de posição por referência variável.

## 2.6 CONTROLE DE CAMINHO

Baseado em um referencial Serret-Frenet (**S-F**), onde é localizado no ponto de projeção ortonormal, o vetor tangente, o vetor normal e o vetor binomial, do robô sobre o caminho onde o referencial **S-F** move-se, ou seja, a posição do robô tem que convergir para o alvo virtual  $p^*(s)$ , que está se movendo junto com o referencial **S-F**. Para Minimizar os erros o controlador usa as seguintes equações:

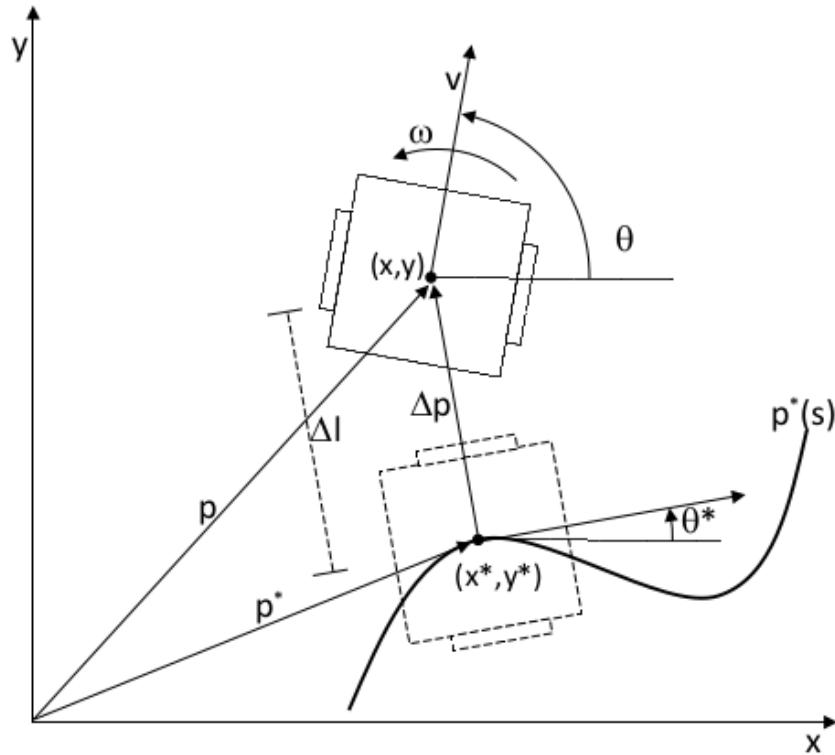


Figura 6 – Descrição do robô em relação ao caminho: baseada em um referencial Serret-Frenet.

$$\Delta l = \sqrt{(x - x^*)^2 + (y - y^*)^2} \quad (24)$$

$$\Delta\theta = \theta - \theta^* \quad (25)$$

Onde o caminho tem como característica uma curvatura inversa ao raio de giro,  $\kappa(s)$ . Logo  $\theta^*$  tem que satisfazer a seguinte equação:

$$\frac{d\theta^*}{dt} = \kappa(s) \cdot \frac{ds}{dt} \quad (26)$$

O alvo virtual só existe se  $k(s)$  possuir um limite superior a  $p^*(s)$ , não podendo ter curvas abruptas e o robô não pode se afastar muito do caminho, pois a parametrização Serret-Frenet é local, logo o modelo cinemático em relação ao referencial **S-F** é:

$$\begin{aligned} \frac{d(\Delta l)}{dt} &= v \sin(\Delta\theta) \\ \frac{d(s)}{dt} &= \frac{v \cos(\Delta\theta)}{1 - \kappa(s)\Delta l} \\ \frac{d(\Delta\theta)}{dt} &= \omega - \frac{\kappa(s)v \cos(\Delta\theta)}{1 - \kappa(s)\Delta l} = u \end{aligned}$$

Para a lei de controle de Samson a expressão dos erros de seguimento de caminho são:

$$\begin{aligned} \frac{\Delta l}{dt} &= v \sin(\Delta\theta) \\ \frac{\Delta\theta}{dt} &= u \end{aligned}$$

Onde  $u$  é uma nova estrada de controle para o erro angular. A lei de controle cinemático de **Samson** para o seguimento de caminho são:

$$v = \text{constante} \quad (27)$$

$$u = - \left( K_\theta \Delta\theta + \frac{K_l \cdot \Delta l \cdot v \cdot \sin(\Delta\theta)}{\Delta\theta} \right) \quad (28)$$

$$\omega = u + \frac{\kappa(s) \cdot v \cdot \cos(\Delta\theta)}{1 - \kappa(s)\Delta l} \quad (29)$$

Onde  $K_\theta$  e  $K_l > 0$ .

Com esta lei de controle acaba-se minimizado a função de Lyapunov para:

$$v(\Delta l, \Delta\theta) = \frac{K_l \cdot \Delta l^2 + \Delta\theta^2}{2}$$

## 2.7 CONTROLE DE TRAJETÓRIA

O objetivo deste tipo de controlador é fazer o robô seguir uma determinada trajetória minimizando o erro entre a referência e a trajetória que o robô está localizado. A referência a ser seguida é apenas a posição  $(x^*(t), y^*(t))$ , dessa forma se encontra de forma implícita a orientação desejada a ser seguida.

Os referenciais de velocidade no espaço de configuração podem ser mapeados por referências equivalentes da velocidade linear e angular:

$$v^* = \pm \sqrt{(\dot{x}^*)^2 + (\dot{y}^*)^2} \quad (30)$$

$$\omega^* = \frac{\dot{x}^* \ddot{y}^* - \ddot{x}^* \dot{y}^*}{\dot{x}^{*2} + \dot{y}^{*2}} \quad (31)$$

### 2.7.1 CONTROLE DE TRAJETÓRIA POR LINEARIZAÇÃO DA REALIMENTAÇÃO DINÂMICA

Envolvendo uma realimentação PD e compensação do modelo não-linear o DFL (Dynamic Feedback Linearization)(Novel, et al 1995), é acrescentado um estado adicional ao sistema, derivando o modelo cinemático  $\dot{\mathbf{q}} = {}^q \mathbf{T}_v \cdot \mathbf{V}$ :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \ddot{x} = \dot{v} \cos \theta - v \omega \sin \theta \\ \ddot{y} = \dot{v} \sin \theta + v \omega \cos \theta \end{cases}$$

E colocando na forma matricial teremos:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & -v \sin \theta \\ \sin \theta & v \cos \theta \end{bmatrix} \cdot \begin{bmatrix} \dot{v} \\ \omega \end{bmatrix} \quad (32)$$



Figura 7 – Caption

Sendo assim este tipo de controlador requer a trajetória desejada junto com suas derivadas de primeira e segunda ordem, os valores de  $[x \ y \ \theta]$   $\mathbf{T}$ ,  $\dot{x}$  e  $\dot{y}$ , além da determinação  $v$  através de  $\dot{v}$  e  $\omega$  e  $\dot{v}$  determina como necessários para impor a trajetória desejada. Para a realimentação das acelerações de comando com  $K_{px}, K_{py} > 0$ , onde são os ganhos de  $P$  e  $K_{dx}, K_{dy} > 0$  que serão os ganhos de  $D$  para a seguinte equação:

$$\begin{cases} \ddot{x}_c = \ddot{x}^* + K_{dx}(\dot{x}^* - \dot{x}) + K_{px}(x^* - x) \\ \ddot{y}_c = \ddot{y}^* + K_{dy}(\dot{y}^* - \dot{y}) + K_{py}(y^* - y) \end{cases} \quad (33)$$

A compensação do modelo não linear:

$$\begin{bmatrix} \dot{v}_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{v} & \frac{\cos \theta}{v} \end{bmatrix} \cdot \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \end{bmatrix} \quad (34)$$

E a integração para a velocidade linear de comando é:

$$\int_0^t \dot{v}_c(t) dt \quad (35)$$

Para este tipo de controlador a trajetória deve ser duplamente derivável, o estado inicial do integrador em  $v_c$  de ser sempre diferente de 0(zero), a trajetória desejada deve ser persistente de uma maneira que evite as singularidades e haja uma possível modificação para contornar a divisão por zero quando  $v \rightarrow 0$ , substituindo o  $v$  por  $(v + \delta)$ ,  $\delta > 0$  e  $\delta \ll$  que a velocidade nominal.

### 3 METODOLOGIA

A simulação foi realizada no ambiente *CoppeliaSim*(2) em configuração cliente/servidor, sendo o servidor responsável pela simulação da física em tempo real e o cliente em C++(6) responsável por calcular o polinômio de terceiro grau que servirá de caminho para o robô e a trajetória, para os controles seguidor de caminho e seguidor de trajetória, respectivamente, bem como realizar a rotina de controle propriamente dita, para cada um dos três controladores. A comunicação foi feita utilizando a biblioteca para Cpp disponibilizada pelo *CoppeliaSim* (B0-based remote API, C++ (2)), gráficos (gerados com o uso da biblioteca *matplotlib-cpp*(7)) também serão gerados e exibidos pelo cliente, em tempo de execução. Links para o código fonte bem como os demais recursos utilizados encontram-se em anexo 1.

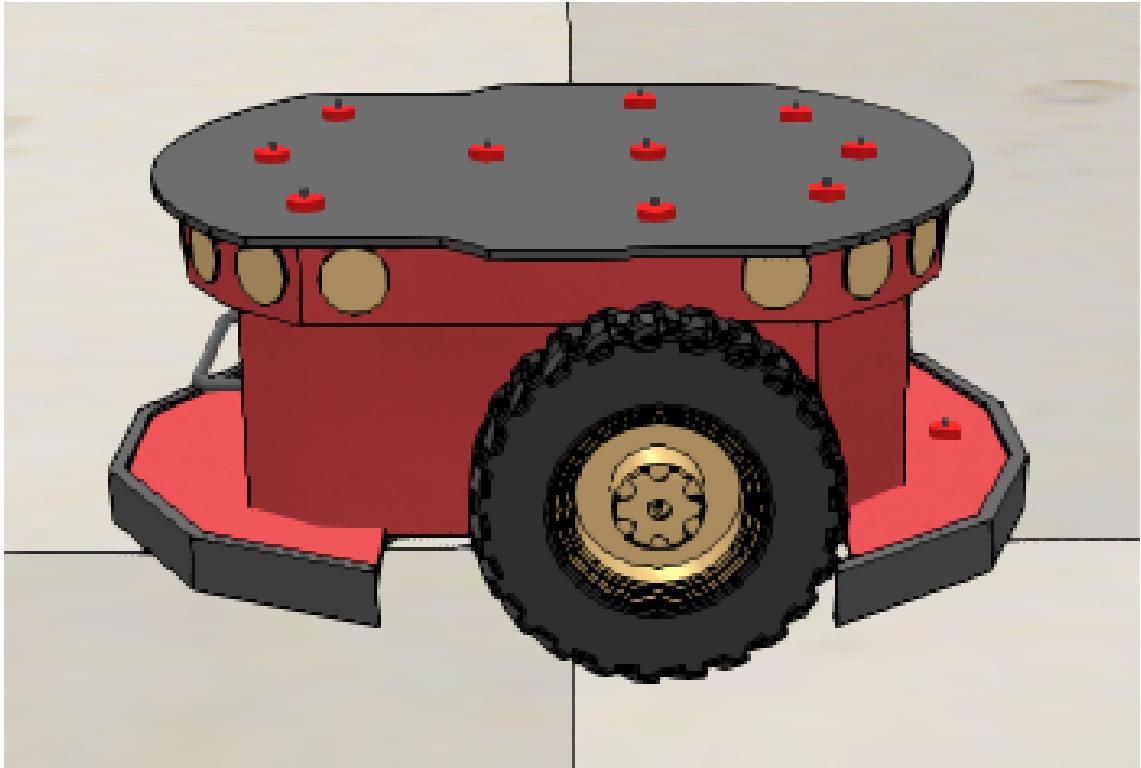


Figura 8 – Modelo simulado do robô Pioneer.

O robô utilizado possui rodas com rádio igual a  $0.09751m$  ( $r_w$ ) e a distância entre os centros das mesmas é de  $0.331m$  ( $b$ ), podemos ver na figura 8 o modelo do robô no ambiente simulado.

Sabendo as dimensões do robô, mais exatamente, o raio das rodas e o comprimento do eixo das rodas, podemos calcular as relações conforme eq. (7) (8) e teremos:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 0.049 & 0.049 \\ 0.295 & -0.295 \end{bmatrix} \cdot \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix}$$

$$\begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} = \begin{bmatrix} 10.255 & 1.697 \\ 10.255 & -1.697 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}$$

O cenário criado no ambiente é bem simples, contem apenas 2 objetos que não vêm no cenário padrão do simulador, o robô e um objeto circular na cor vermelha, como podemos ver na figura 9, o robô na estado inicial representa a configuração inicial (posição e orientação) e a marcação em vermelho é a configuração final, esse padrão foi utilizado na implementação dos algoritmos de controlos apresentados neste trabalho, as configurações destes dois objetos no mundo servirá como entrada para o gerador de caminho que será executado pelo programa cliente, o programa cliente coleta a informação de posição e orientação destes dois objetos e utiliza como entrada para a função que calculará os coeficientes da curva, além de realizar leituras periódicas (a cada passo de amostragem do

controlador) da configuração atual do robô e do target (objeto em vermelho) por meio da B0based remote API, C++.

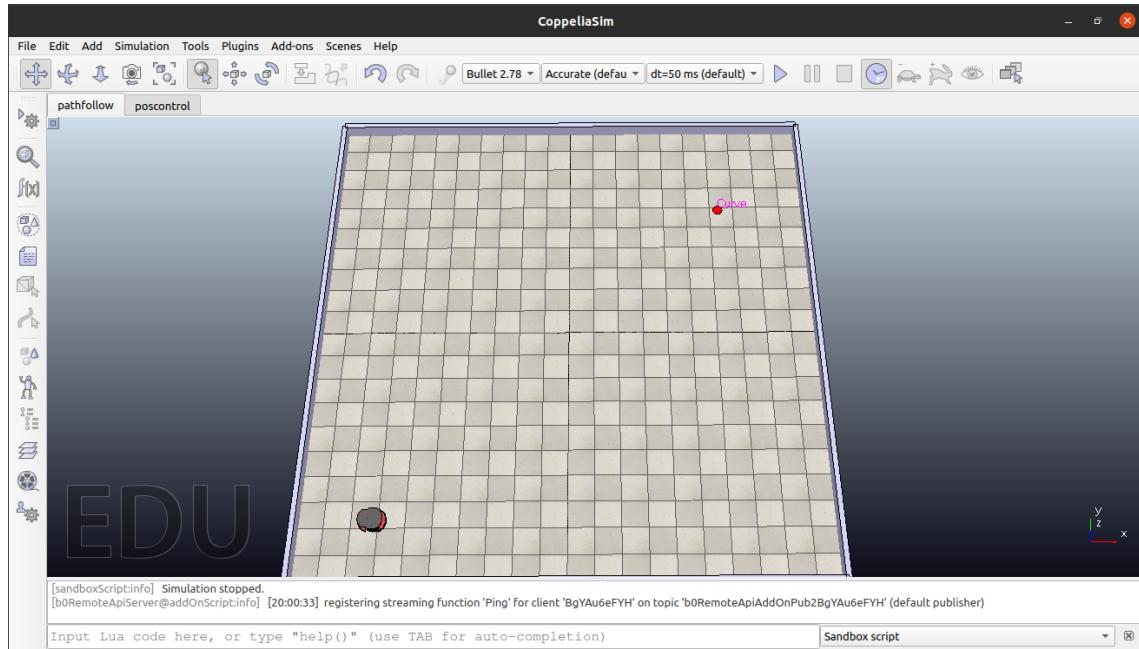


Figura 9 – Cenário no simulador CoppeliaSim.

Foi criado três códigos, três ambientes para demonstração, uma para cada tipo de controlador, todas seguem um mesmo padrão, onde é utilizado duas threads, uma responsável por plotar em tempo real as informações relevantes, como a informação dos erros lineares e angulares, posição do robô, entre outras, e uma thread responsável por realizar a leitura das medições necessárias para o controlador bem como executar a rotina de controle propriamente dita, dessa forma é possível plotar sem interferir no tempo de amostragem dos controladores.

## 4 RESULTADOS E CONCLUSÕES

### 4.1 CONTROLE ESTABILIZANTE

Resultados obtidos para o controle de posição com condição de parada, erro linear  $\leq 0.01m$ , controle estabilizante de posição, com as seguintes configurações:

Ganhos dos controladores *PID* linear e angular:

$$\begin{cases} K_{P_{linear}} = 0.1 \\ K_{I_{linear}} = 0.01 \\ K_{D_{linear}} = 0.0 \\ K_{P_{angular}} = 0.5 \\ K_{I_{angular}} = 0.15 \\ K_{D_{angular}} = 0.0 \end{cases}$$

Configurações inicial e final(posição inicial e posição final):

$$\begin{cases} x_i = -4.0\text{m} \\ y_i = 2.3\text{m} \\ \theta_i = 0.0\text{rad} \\ x_f = 3.0\text{m} \\ y_f = -3.0\text{m} \\ \theta_f = 0.0\text{rad} \end{cases}$$

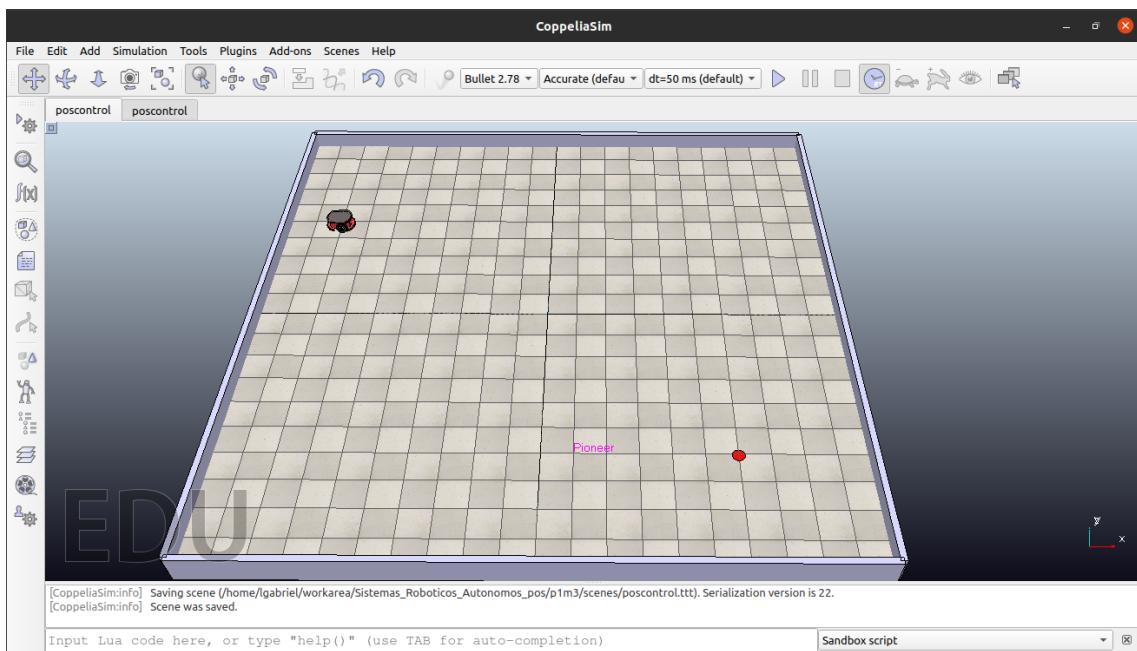


Figura 10 – Estado inicial da simulação para o teste do controle estabilizante.

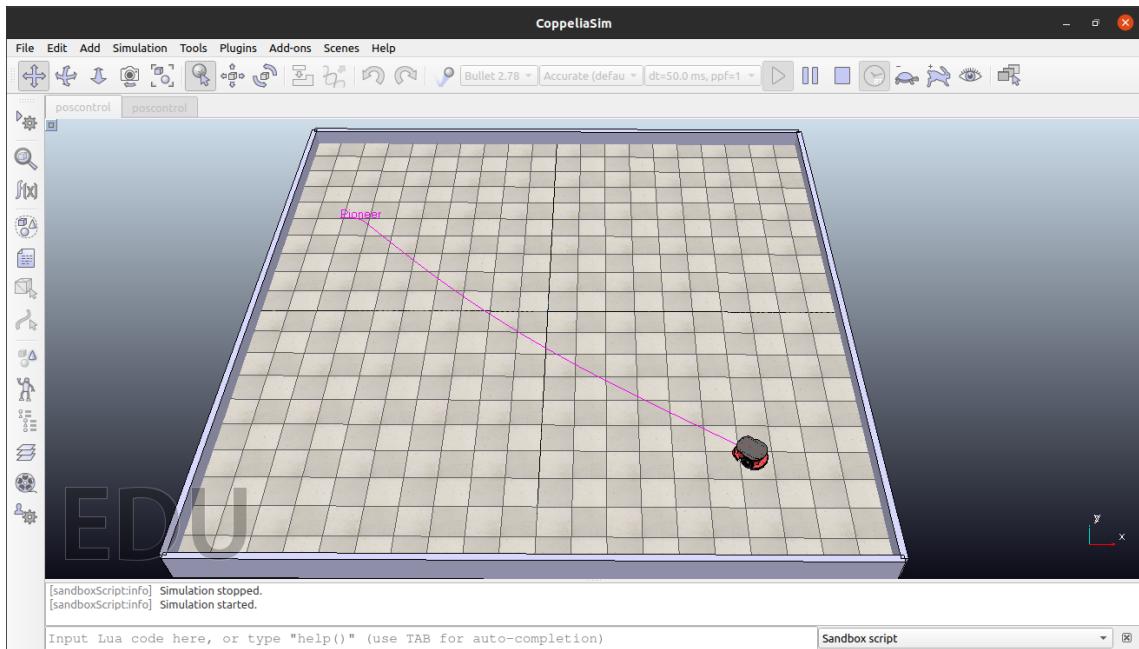


Figura 11 – Estado final da simulação para o teste do controle estabilizante.

A curva em rosa presente na figura 11 é o caminho percorrido pelo robô, o mesmo caminho pode ser visto na figura a seguir, no plot de  $x$  vs  $y$ .

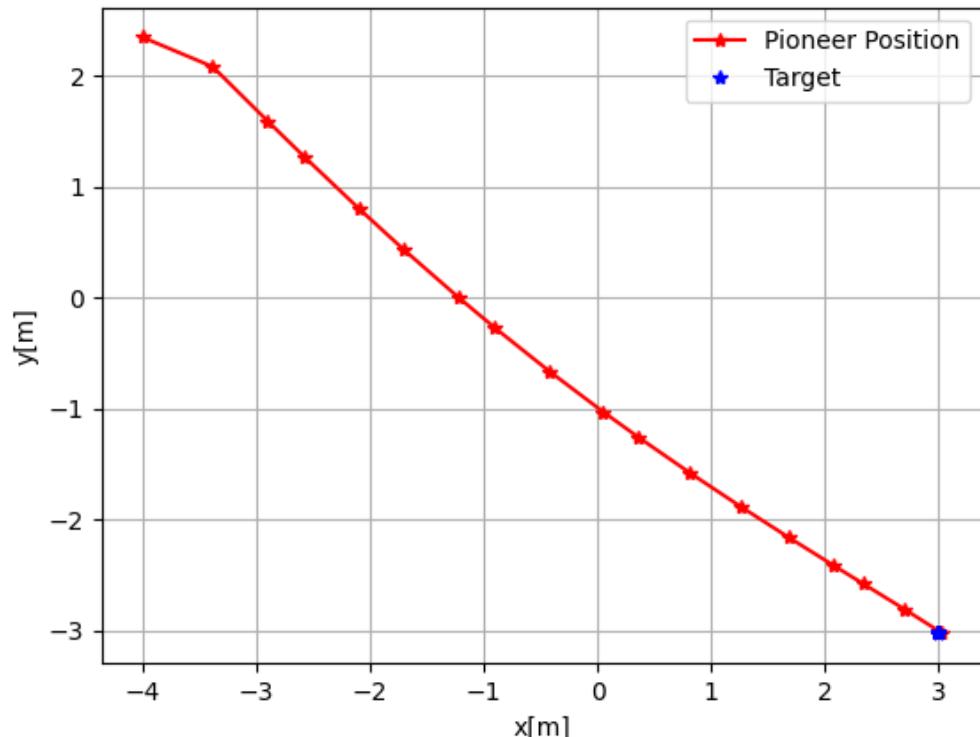


Figura 12 – Gráfico dos instantes da posição do robô no mundo com relação ao referencial global.

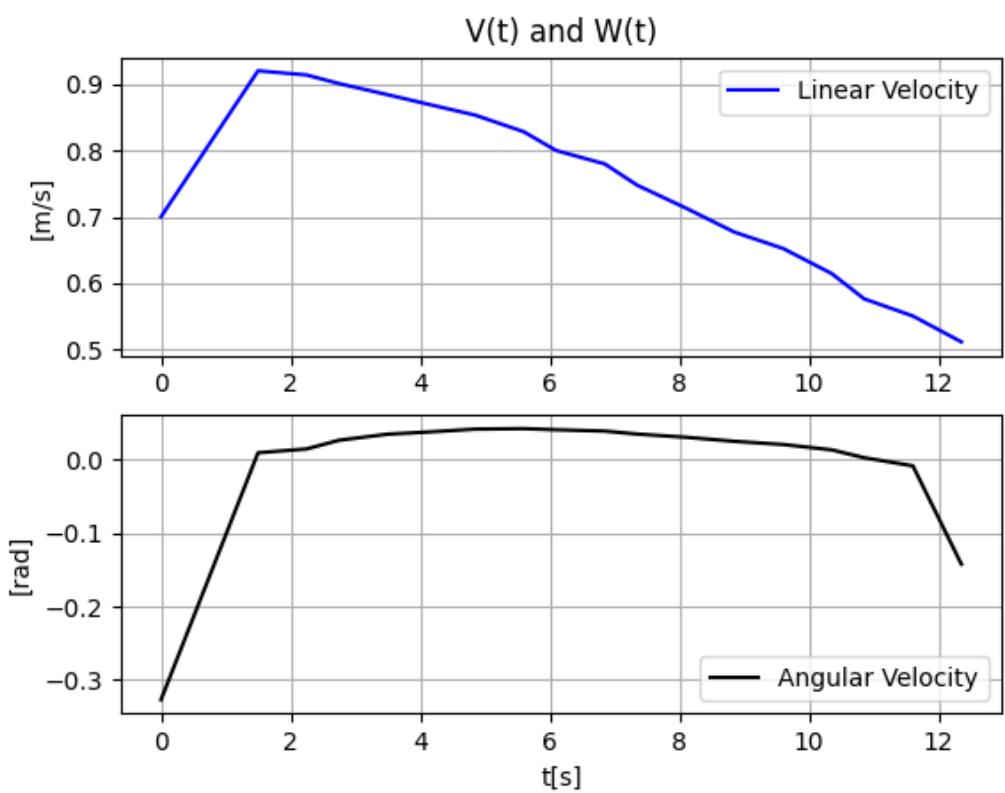


Figura 13 – Gráfico da evolução no tempo das velocidades linear e angular do robô.

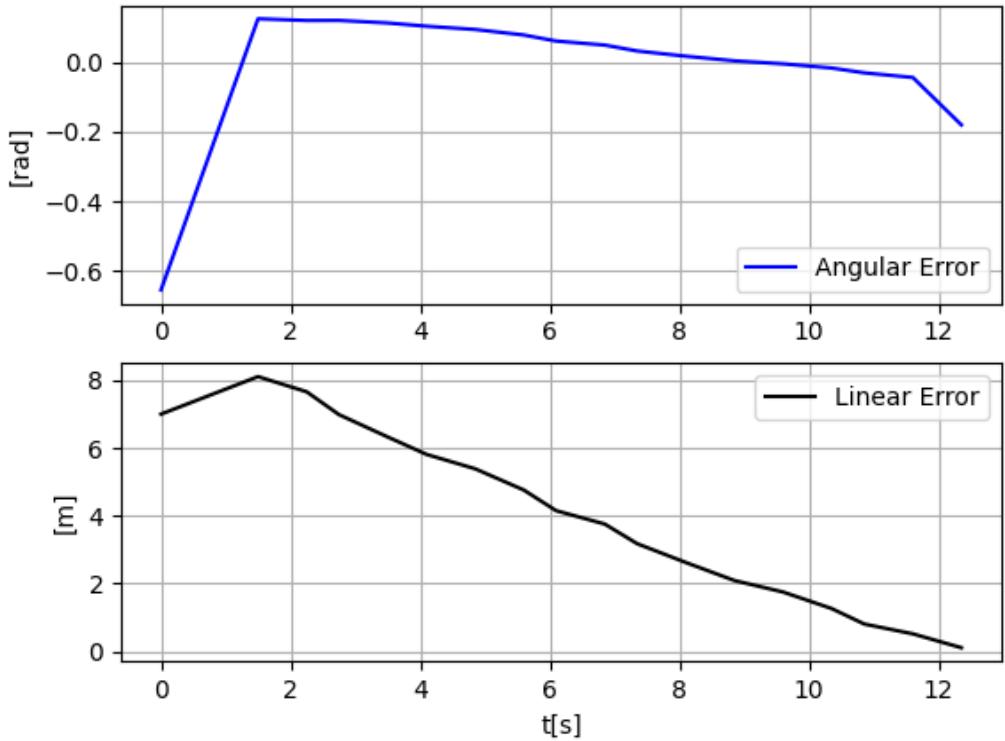


Figura 14 – Gráfico dos erros linear e angular do robô no tempo.

Podemos verificar pelos gráficos presentes nas figuras 13 e 14 que o robô levou um pouco mais de 12s para chegar ao alvo, além disso é possível observar que a velocidade linear decresce conforme o erro linear diminui e o erro angular diminui conforme o robô se alinha com a reta que vai do robô até o alvo.

## 4.2 CONTROLE SEGUIDOR DE CAMINHO

Os resultados apresentados aqui foram obtidos com as seguintes configurações:

Configuração inicial e final,

$$\begin{cases} v = 0.3 \text{ m/s} \\ x_i = -3.6 \text{ m} \\ y_i = -3.9 \text{ m} \\ \theta_i = 0 \text{ rad} \\ x_f = 3.2 \text{ m} \\ y_f = 2.9 \text{ m} \\ \theta_f = \frac{\pi}{4} \text{ rad} \end{cases}$$

Coeficientes do polinômio interpolador:

$$\begin{cases} a_0 = -3.6004 \\ a_1 = 6.7754 \\ a_2 = 0.0000 \\ a_3 = 0.0000 \\ b_0 = -3.8750 \\ b_1 = -0.0002 \\ b_2 = 13.6248 \\ b_3 = -6.8247 \end{cases}$$

Ganhos do controlador:

$$\begin{cases} K_l = 0.1 \\ K_\theta = 1.0 \end{cases}$$

Condição de parada: distância até o ponto final  $\leq 0$ .

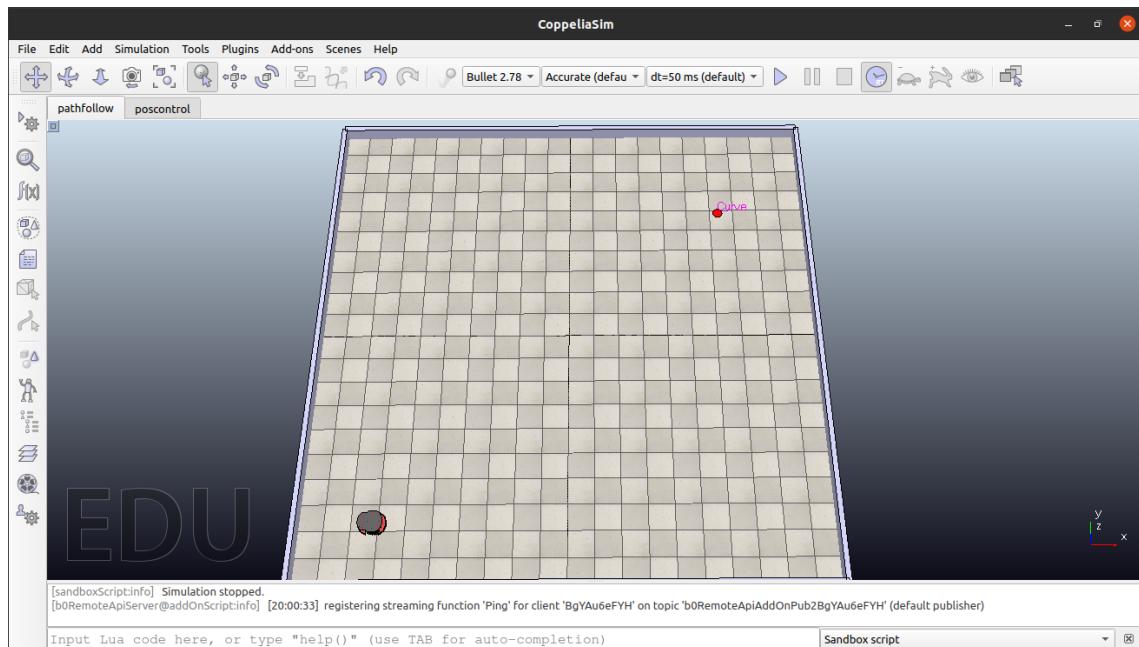


Figura 15 – Estado inicial da simulação para o teste do seguidor de caminho.

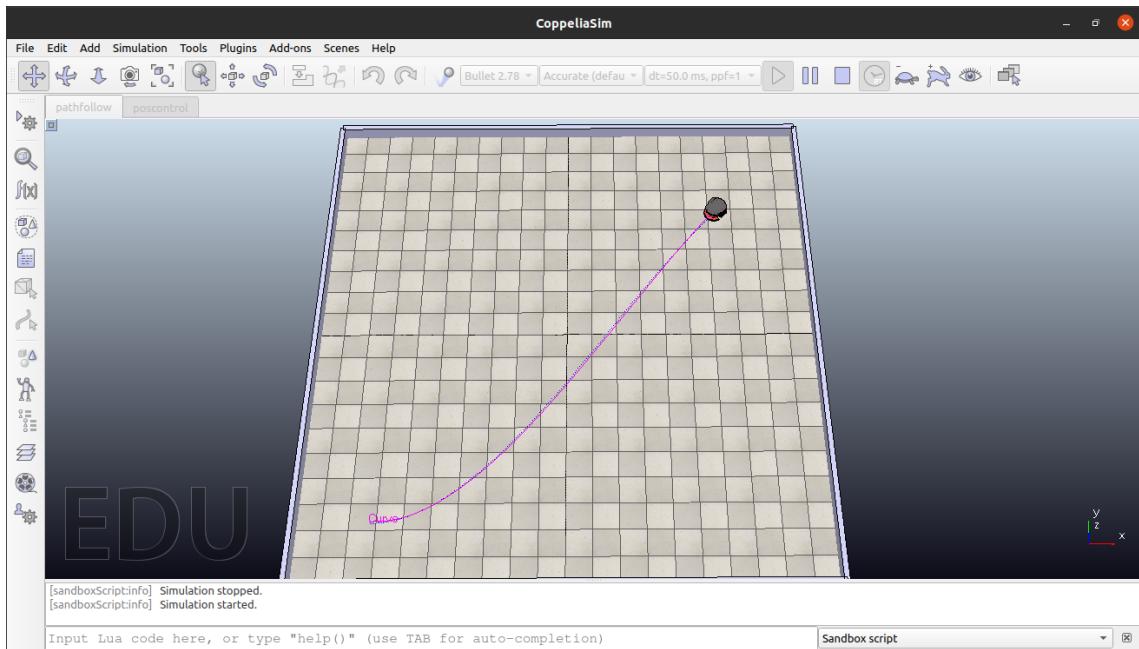


Figura 16 – Estado final da simulação para o teste do seguidor de caminho.

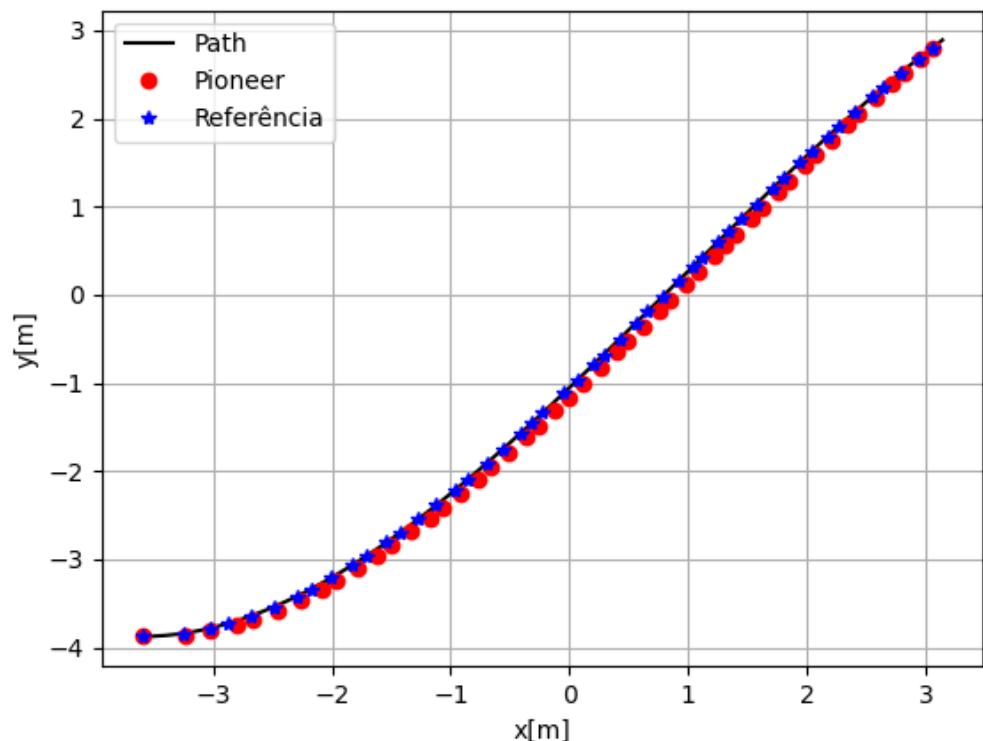


Figura 17 – Gráfico do plano x-y, posição do robô em vermelho, a marcação em azul é a referência num instante ou seja o referencial S-F e o caminho que está sendo percorrido em preto.

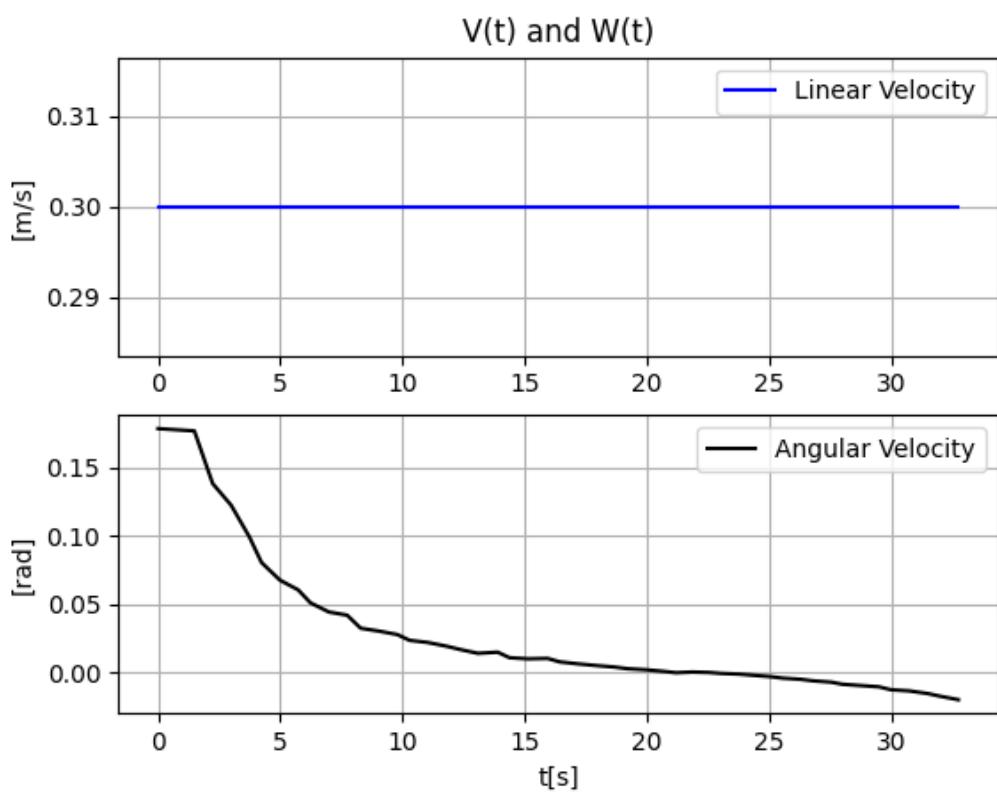


Figura 18 – Gráfico de velocidade linear e angular em função do tempo.

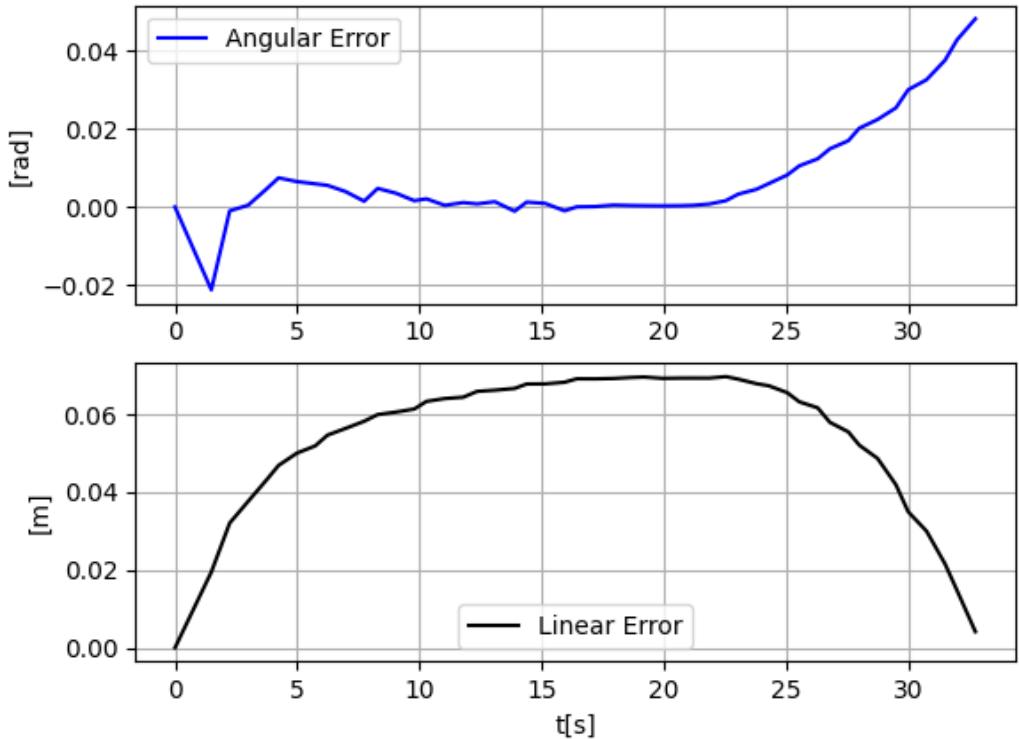


Figura 19 – Gráfico da evolução dos erros linear e angular no tempo.

Podemos observar nos gráficos das figuras 18 e 19, o tempo gasto para completar o caminho foi de aproximadamente 30s e que em todo o percurso os erros linear e angular mantiveram-se próximo de zero, porém é notável a instabilidade quando o robô chega na ultima referência, fazendo os erros aumentarem, isso não foi muito evidente aqui pois a condição de parada levou em conta uma margem de erro até a referência final.

### 4.3 CONTROLE DE TRAJETÓRIA

Os resultados apresentados aqui são provenientes da simulação com o uso do controle de trajetória, usando o perfil de velocidade cosenoidal e o polinômio de terceiro grau, como os apresentados na seção de referencial teórico. A trajetória gerada para a configuração da simulação possuí as seguintes características:

Ganhos do controlador:

$$\begin{cases} K_{px} = K_{py} = 0.8 \\ K_{dx} = K_{dy} = 0.8 \end{cases}$$

Polinômio interpolador:

$$\begin{cases} a_0 = -3.3753 \\ a_1 = 6.3753 \\ a_2 = 0.0000 \\ a_3 = 0.0000 \\ b_0 = -2.4250 \\ b_1 = -0.0001 \\ b_2 = 11.8500 \\ b_3 = -7.9000 \end{cases}$$

Para o perfil de velocidade foi fixado a velocidade máxima,  $V_{max} = 0.5\text{m/s}$ ) e o tempo máximo foi calculado como em eq.19, portanto foi calculado o comprimento de percurso a ser realizado, para isso foi calculado a integral do caminho gerado, de forma numérica para  $\lambda$  de 0 até 1:

$$v = 0.5 \text{ m/s}$$

$$L = 7,6625 \text{ m}$$

$$t_{max} = 30,8492 \text{ s}$$

As imagens a seguir foram obtidas da simulação com esse controlador nessas configurações.

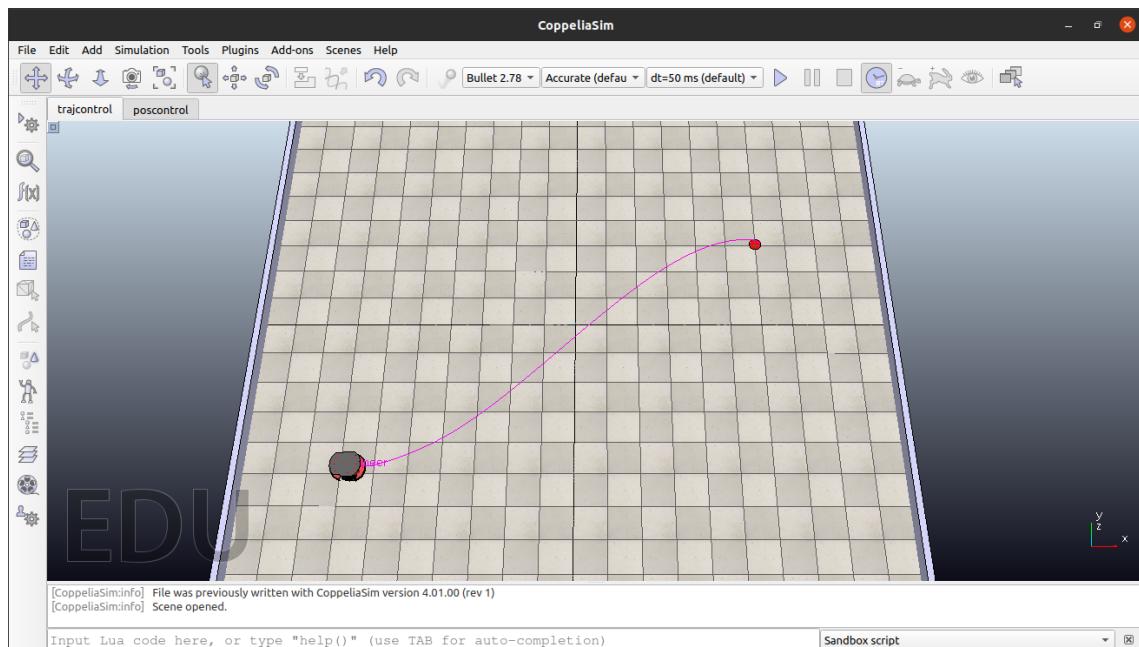


Figura 20 – Estado inicial da simulação.

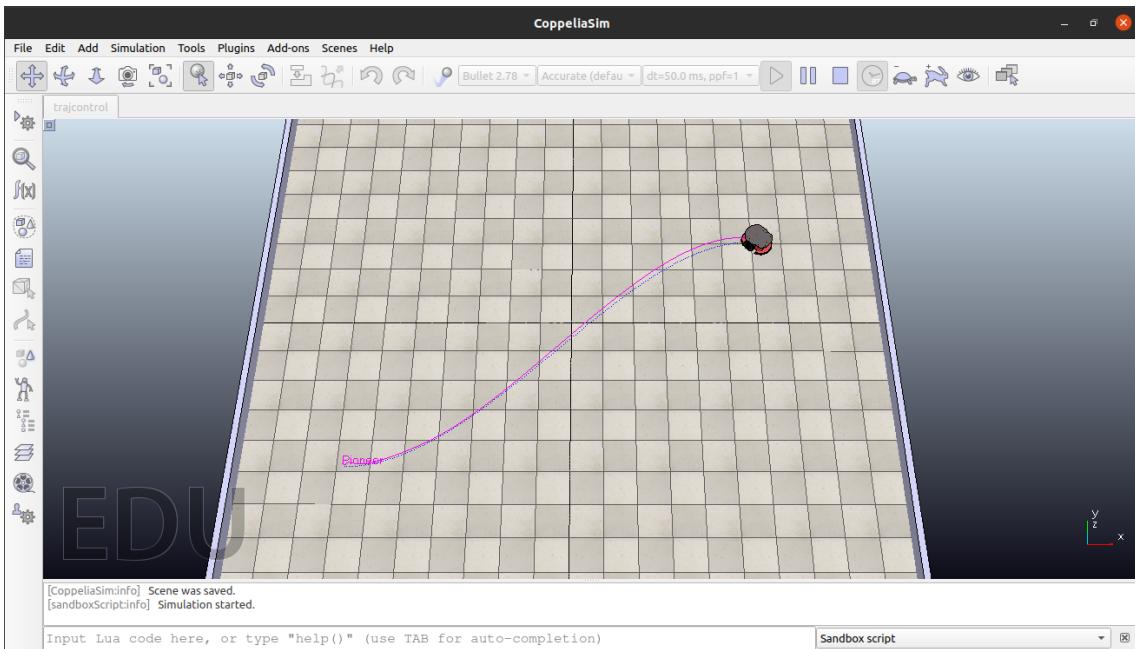


Figura 21 – Estado final da simulação com o controlador de trajetória.

A configuração inicial, apresentada na figura 20 e 21, têm o robô na configuração ( $x_i = -3.37m$ ,  $y_i = -2.43m$ ,  $\theta_i = 0\text{rad}$ ) e configuração final ( $x_f = 3.0m$ ,  $y_f = 1.53m$ ,  $\theta_f = 0\text{rad}$ ).

Configurações inicial e final:

$$\begin{cases} x_i = -3.37m \\ y_i = -2.43m \\ \theta_i = 0\text{rad} \\ x_f = 3.0m \\ y_f = 1.53m \\ \theta_f = 0\text{rad} \end{cases}$$

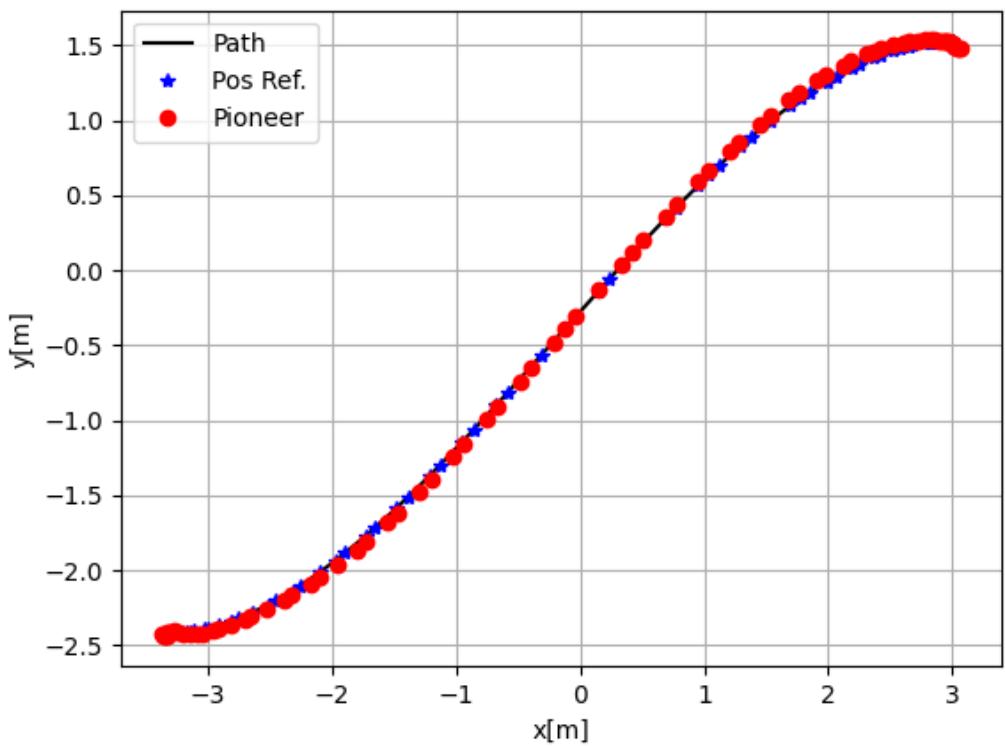


Figura 22 – Gráfico da posição do robô com relação ao referencial global.

No gráfico da figura 22 os pontos em vermelho representam a posição do robô, em azul é a referência  $(x^*,y^*)$  e em preto é o polinômio que interpola a configuração final com a inicial.

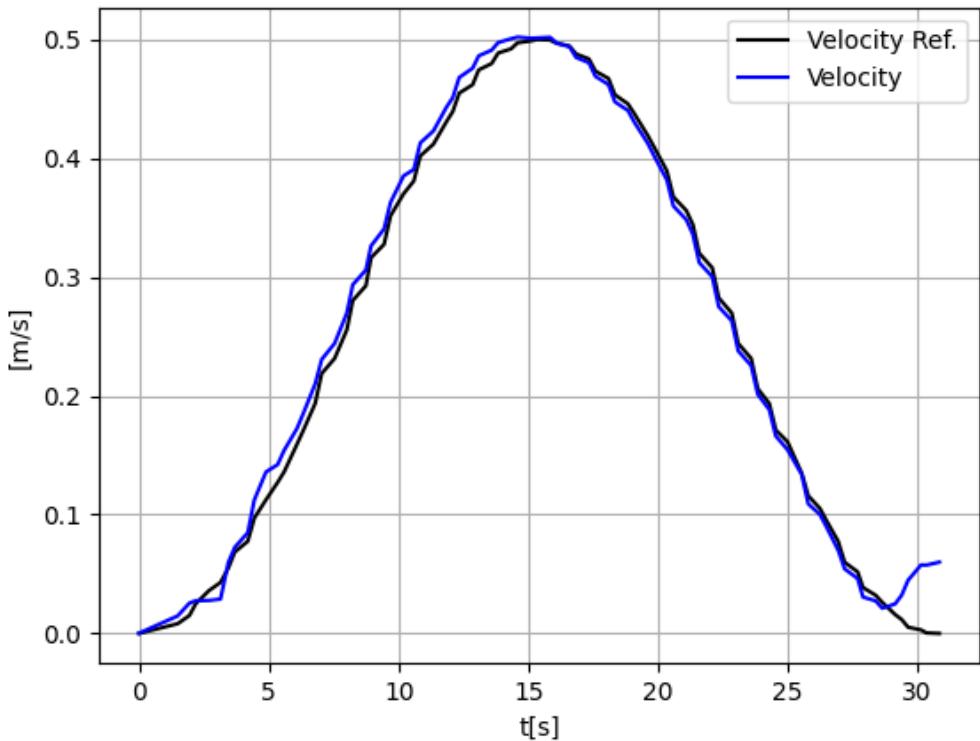


Figura 23 – Gráfico da velocidade no tempo.

O controle de trajetória obteve bons resultados, podemos notar pelos gráficos das figuras 23 e 22 que tanto o perfil de velocidade quanto o caminho foram seguidos com boa precisão. Nota-se instabilidade no final da trajetória, pois o controlador é instável nesse ponto, isso pode ser mais facilmente observável no gráfico da figura 23, uma forma de diminuir este efeito é desligar o controlador um pouco antes de se chegar ao fim da trajetória, é possível fazer isso adotando-se uma margem (um raio) com relação ao ponto final, como sendo o fim da trajetória.

## Referências

- 1 Python Org. Disponível em: <<https://www.python.org/>>. Acesso em: 22 set. 2020.
- 2 E. Rohmer, S. P. N. Singh, M. Freese, "CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2013. [www.coppeliarobotics.com](http://www.coppeliarobotics.com) 13
- 3 Matplotlib: Visualization with Python. Disponível em: <<https://matplotlib.org/>>. Acesso em: 22 set. 2020.
- 4 The NumPy Org. Disponível em: <<https://numpy.org/>>. Acesso em: 22 set. 2020.

- 5 The programming language Lua. Disponível em: <https://www.lua.org/>. Acesso em: 29 set. 2020.
- 6 The C++ programming language. Disponível em: <http://www.cplusplus.org/>. Acesso em: 19 out. 2020. 13
- 7 Matplotlibcpp. Disponível em: <https://github.com/lava/matplotlib-cpp>. Acesso em: 19 out. 2020. 13
- 8 PEDROSA, Diogo Pinheiro Fernandes. Sistema de navegação para robôs móveis autônomos. 2001. 85 f. Dissertação (Mestrado em Automação e Sistemas; Engenharia de Computação; Telecomunicações) - Universidade Federal do Rio Grande do Norte, Natal, 2001. Disponível em: <https://repositorio.ufrn.br/handle/123456789/15417> 1, 4
- 9 VIEIRA, Frederico Carvalho. Controle dinâmico de robôs móveis com acionamento diferencial. 2006. 106 f. Dissertação (Mestrado em Automação e Sistemas; Engenharia de Computação; Telecomunicações) - Universidade Federal do Rio Grande do Norte, Natal, 2006. Disponível em: <https://repositorio.ufrn.br/handle/123456789/15414>. Acesso em: 19 out. 2020. 8, 9

## 5 ANEXO 1

**Código fonte:** <[https://github.com/Gabriellgpc/Sistemas\\_Roboticos\\_Autonomos\\_pos/tree/master/p1m3](https://github.com/Gabriellgpc/Sistemas_Roboticos_Autonomos_pos/tree/master/p1m3)>

**Vídeo de demonstração:** <<https://www.youtube.com/watch?v=KIsikoNiXtg&t=6s>>