



Instituto Tecnológico de Aeronáutica



# **Projeto, Construção e Testes de Um Robô Omnidirecional do Tipo Ballbot**

Pedro Henrique de Jesus

Orientador: Prof. Dr. Cairo L. Nascimento Jr.

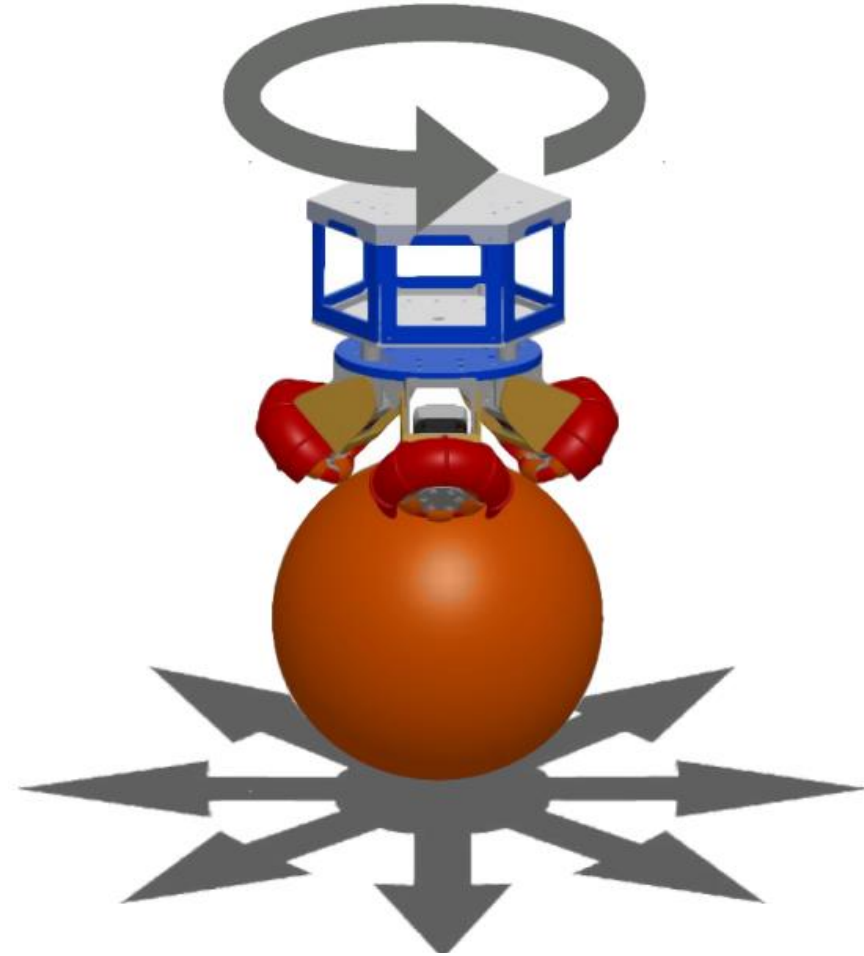
Coorientador: Prof. Dr. Douglas Soares dos Santos

14/12/2021

- Introdução.
- Fundamentação:
  - História do *ballbot*;
  - Modelagem;
  - Controlador LQR;
  - Visão computacional;
  - Navegação por *waypoints*.
- Simulação:
  - Arquitetura de controle;
  - Resultados de simulação 2D;
  - Resultados de simulação 3D.
- Implementação:
  - Projeto e construção.
- Resultados Experimentais:
  - Controle de posição, velocidade e guinada;
  - Navegação autônoma;
  - Rastreamento de objetos.
- Conclusão e trabalhos futuros.

## Motivação

- Categoria recente de robôs omnidirecionais.
- Capacidade de mudar a direção do seu deslocamento em espaços confinados.
- Implementação de navegação autônoma e rastreamento de objetos.



## Objetivo

Construir e testar o robô do tipo *ballbot* com a capacidade de seguir objetos, e também se locomover autonomamente em um ambiente supervisionado por câmeras.

Com esse propósito:

- Simular o modelo não linear do *ballbot* em duas e três dimensões;
- Testar os controladores: posição, velocidade, guinada e velocidade de guinada;
- Implementar um sistema de localização por câmera;
- Desenvolver um sistema de navegação por *waypoints*;
- Implementar um sistema de rastreamento por cor.

## História do *Ballbot*



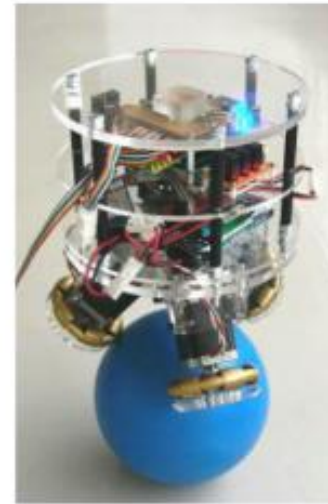
**a**



**b**



**c**



**d**



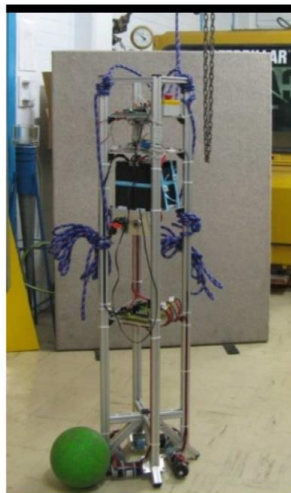
**e**

**2005**

**2009**

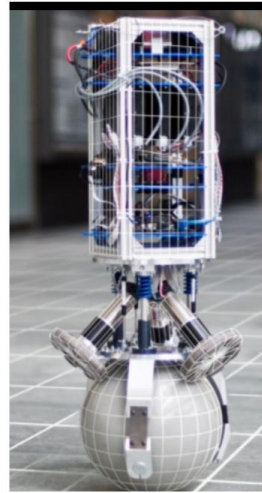
a) Havasi (2005), b) Endo e Nakamura (2005), c) Lauwers *et al.* (2005), d) Kumaga e Ochiai (2009) e e) Yamamoto (2009)

## História do *Ballbot*



**a**

**2009**



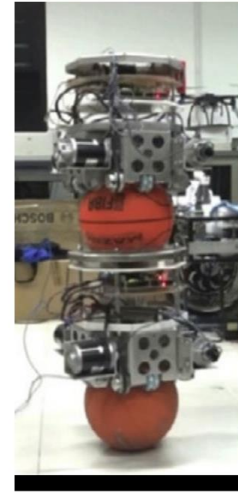
**b**

**2010**



**c**

**2012**



**d**

**2014**



**e**



**f**

**2015**

a) Fong e Uppill (2009), b) Fankhauser e Gwerder (2010), c) Tsai *et al.* (2012), d) Sukvichai e Parnichkun (2014), e) Blonk (2014) e f) Yang *et al.* (2015).

## Modelagem planar

onde:

$m_c$  = massa do corpo [kg]

$I_c$  = momento de inércia do corpo [ $\text{kgm}^2$ ]

$L$  = comprimento do centro da roda até o corpo [m]

$\theta_x$  = ângulo do corpo (Arfagem) [ $^\circ$ ]

$\varphi_x$  = ângulo da roda [ $^\circ$ ]

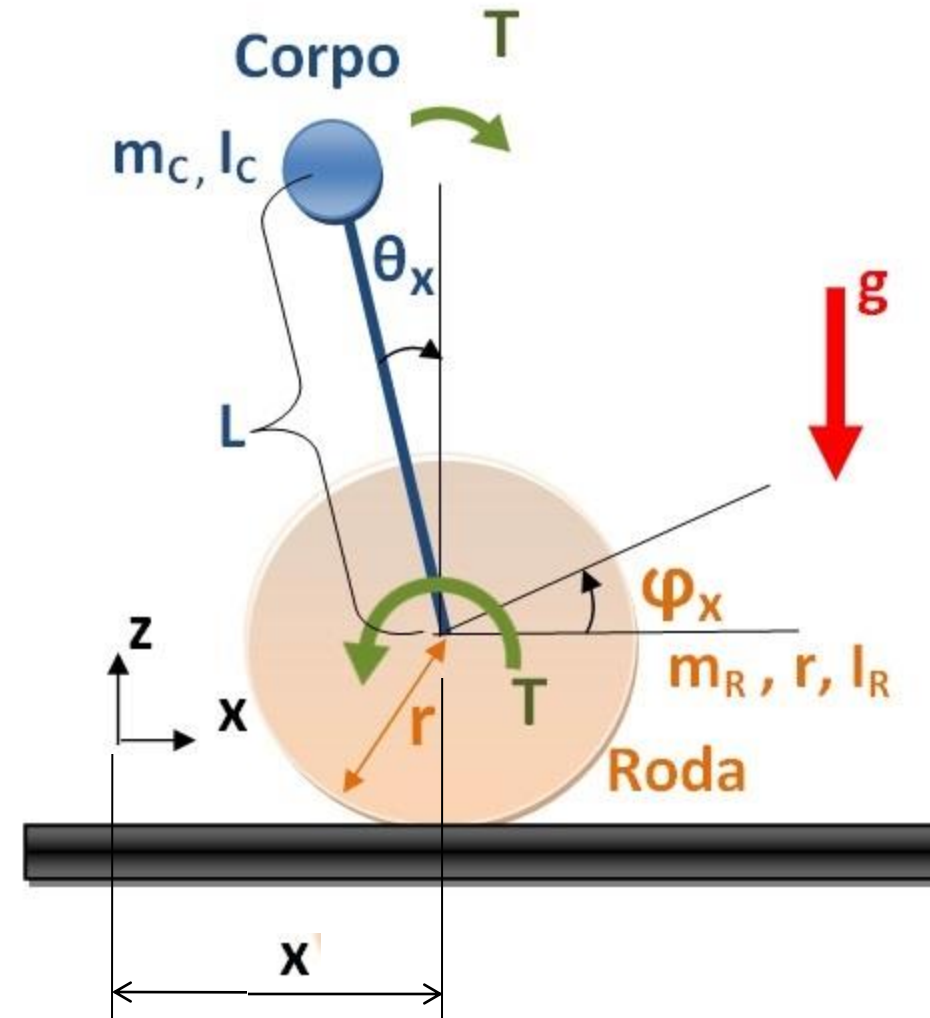
$m_R$  = massa da roda [kg]

$I_R$  = momento de inércia da roda [ $\text{kgm}^2$ ]

$r$  = raio da roda [m]

$g$  = aceleração da gravidade [ $\text{m/s}^2$ ]

$x$  = posição do robô



## Modelo não linear na forma de espaço de estados

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{m_c g L \sin(x_1) + (-I_r - (m_r + m_c) r^2 - m_c r L \cos(x_1)) u + m_c r L \sin(x_1) x_2^2}{I_c + m_c L^2 + m_c r L \cos(x_1)} \\ x_4 \\ -r u \end{bmatrix}$$

onde:

$$u = \ddot{\varphi}_x, x_1 = \theta_x, x_2 = \dot{\theta}_x, x_3 = x, x_4 = \dot{x}$$



## Modelo planar linearizado na origem do espaço de estados

$$\underbrace{\begin{bmatrix} \overline{\dot{\Delta x_1}}(t) \\ \overline{\dot{\Delta x_2}}(t) \\ \overline{\dot{\Delta x_3}}(t) \\ \overline{\dot{\Delta x_4}}(t) \end{bmatrix}}_{\dot{\overline{\Delta \mathbf{x}}}(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m_c g L}{I_c M_c L^2 + M_c r L} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}(t)} \underbrace{\begin{bmatrix} \overline{\Delta x_1}(t) \\ \overline{\Delta x_2}(t) \\ \overline{\Delta x_3}(t) \\ \overline{\Delta x_4}(t) \end{bmatrix}}_{\overline{\Delta \mathbf{x}}(t)} + \underbrace{\begin{bmatrix} 0 \\ \frac{-(I_c + L m_c r + r^2(m_c + m_r))}{I_c + L^2 m_c + L m_c r} \\ 0 \\ -r \end{bmatrix}}_{\mathbf{B}(t)} \overline{\Delta u}(t)$$

onde:

$$\overline{\Delta \mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_{nom}(t), \overline{\Delta \mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_{nom}(t)$$

$$\mathbf{x}_{nom}(t) = \mathbf{0}, \mathbf{u}_{nom}(t) = \mathbf{0}$$

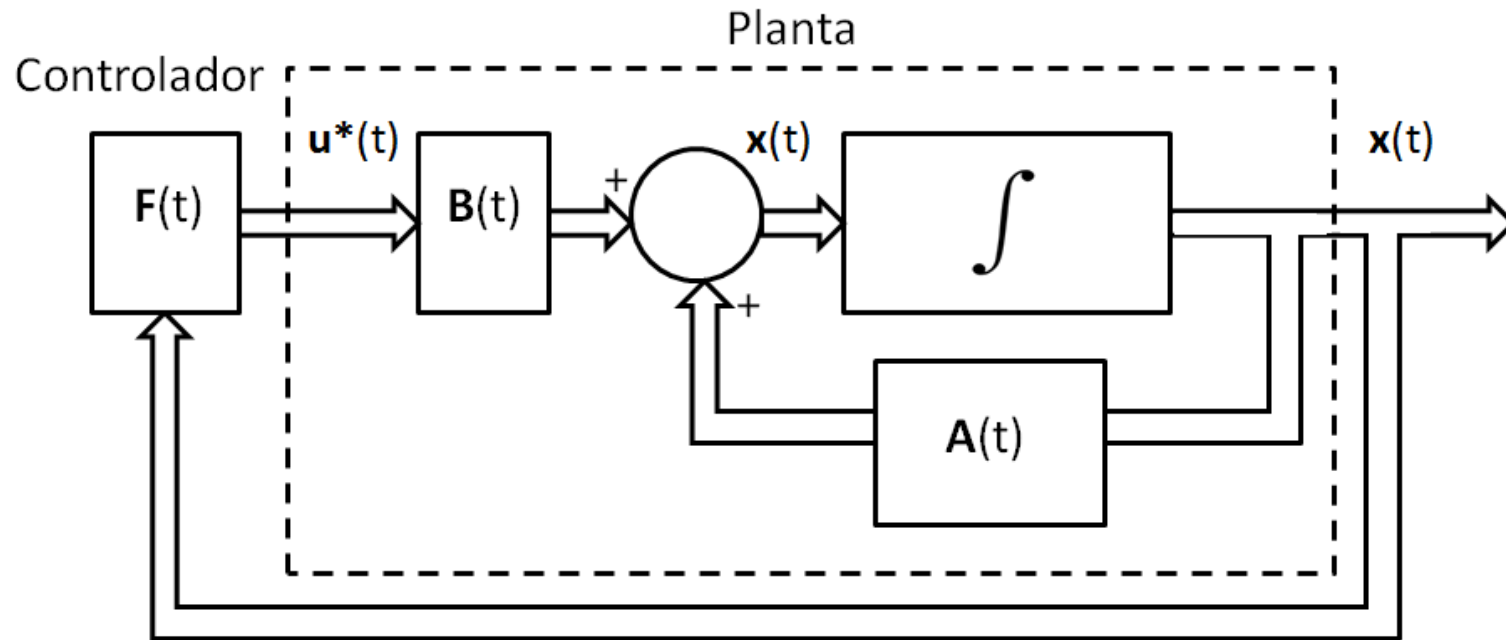
## Modelo planar linearizado na origem do espaço de estados

$$\begin{bmatrix} \dot{\overline{\Delta x_1}}(t) \\ \dot{\overline{\Delta x_2}}(t) \\ \dot{\overline{\Delta x_3}}(t) \\ \dot{\overline{\Delta x_4}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 24.84 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \overline{\Delta x_1}(t) \\ \overline{\Delta x_2}(t) \\ \overline{\Delta x_3}(t) \\ \overline{\Delta x_4}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -0.73 \\ 0 \\ -0.12 \end{bmatrix} \overline{\Delta u}(t)$$

Parâmetro	Descrição	Valor	Unidade
$g$	Aceleração da gravidade	9,81	$m/s^2$
$m_c$	Massa do corpo	1,6	kg
$m_r$	Massa da roda	0,5	kg
$L$	Comprimento da roda até o CG do corpo	0,2	m
$r$	Raio da roda	0,12	m
$I_c$	Momento de inércia do corpo	0,024	$kgm^2$
$I_r$	Momento de inércia da roda	0,0033	$kgm^2$

## Controlador LQR (*Linear- Quadratic Regulator*)

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$$



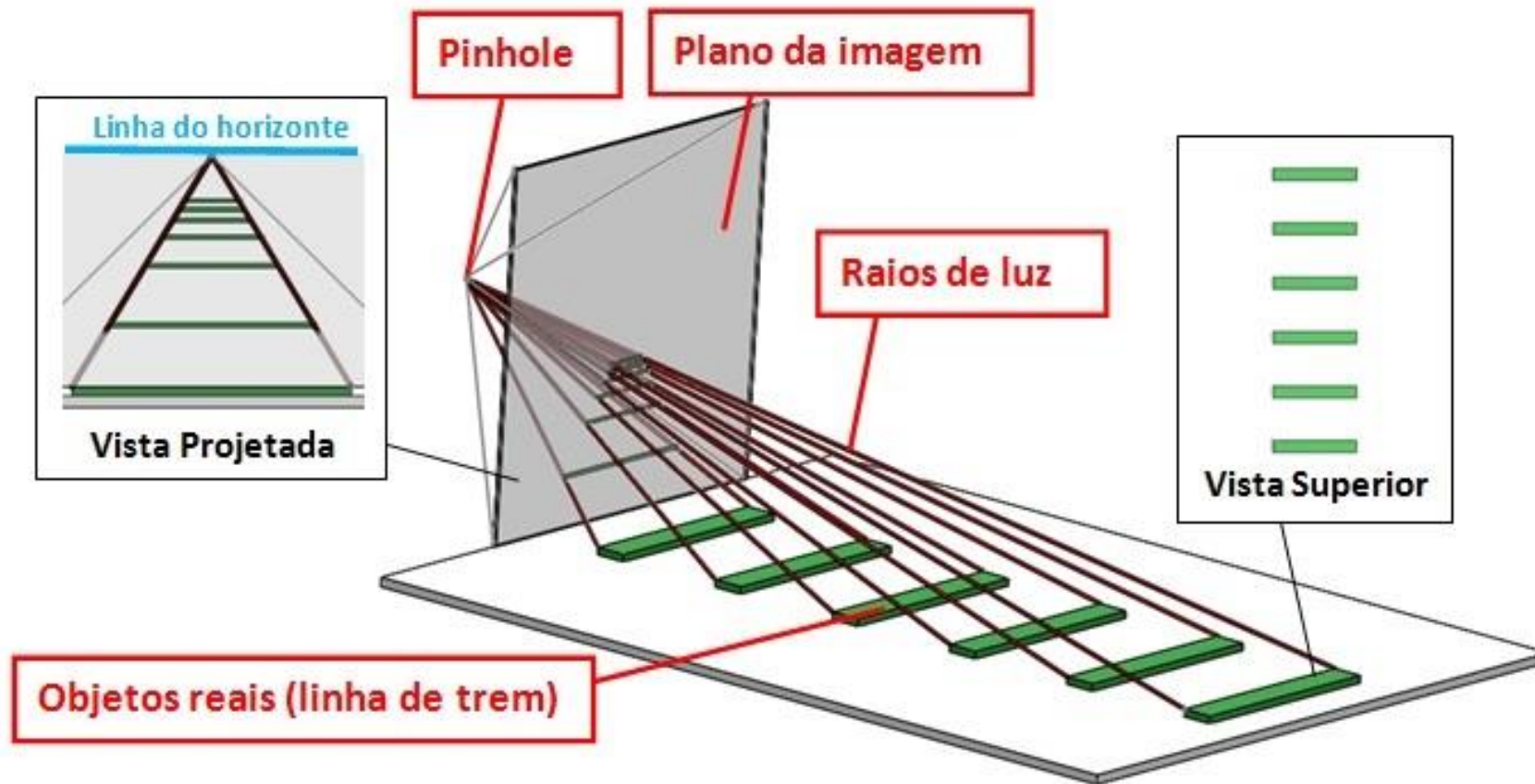
$$\mathbf{F}(t) = -\mathbf{R}^{-1}\mathbf{B}(t)^T\mathbf{K}$$

$$\mathbf{Q} = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}$$

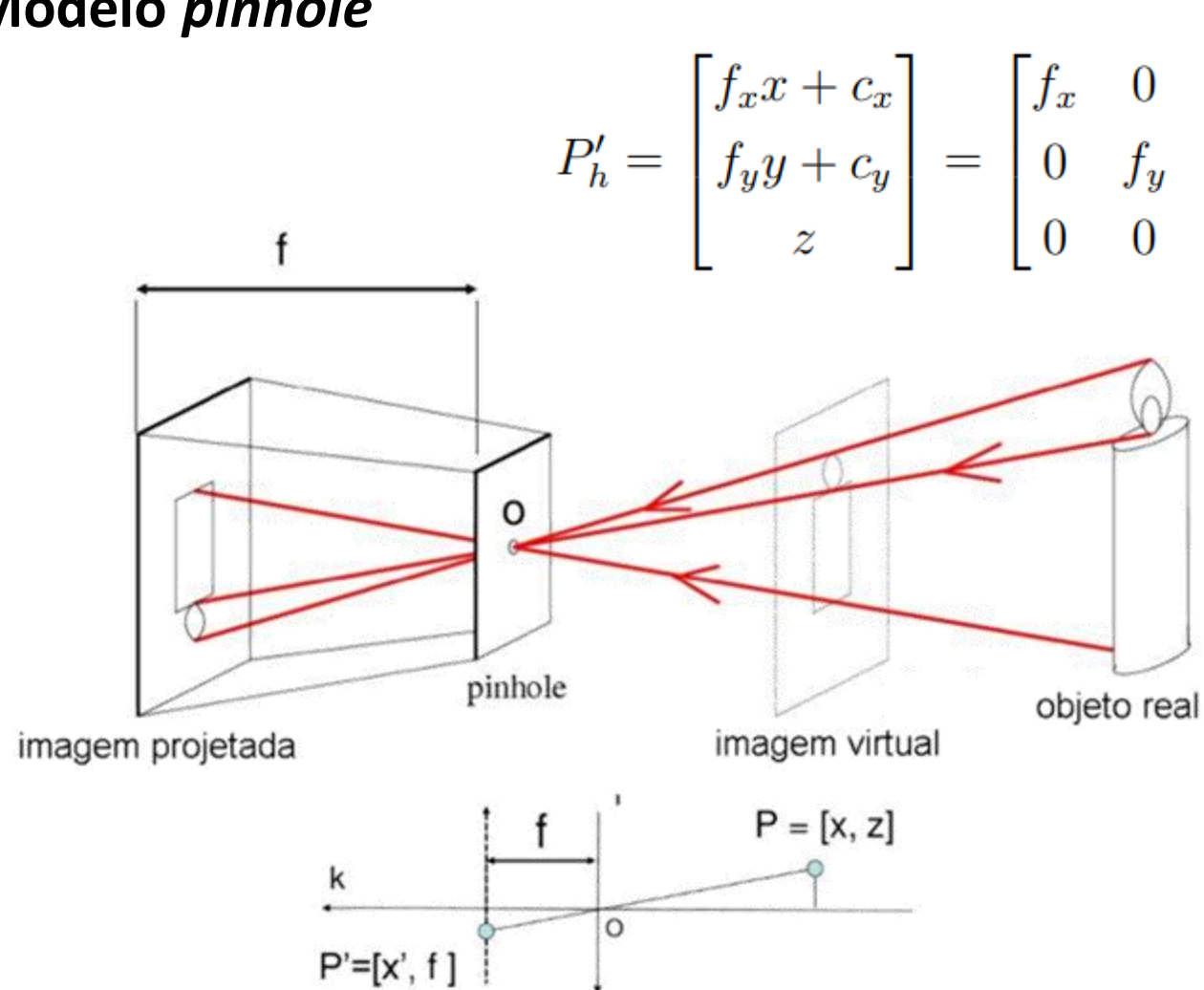
$$\mathbf{R} = 50$$

$$\mathbf{K} = \begin{bmatrix} -70.51 & -14.56 & 0.45 & 2.98 \end{bmatrix}$$

## Visão computacional



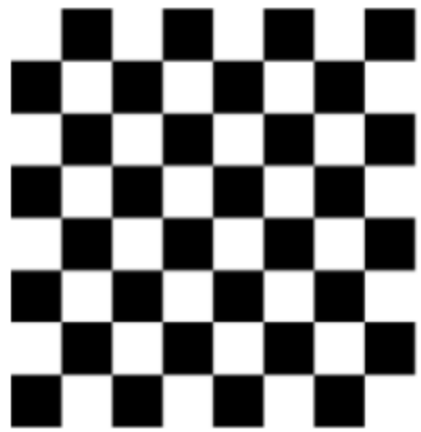
## Modelo *pinhole*



$$P'_h = \begin{bmatrix} f_x x + c_x \\ f_y y + c_y \\ z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_X & 0 \\ 0 & f_y & c_Y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_X & 0 \\ 0 & f_y & c_Y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_K P_j$$

$$P = K \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_{14} \\ r_{21} & r_{22} & r_{23} & t_{24} \\ r_{31} & r_{32} & r_{33} & t_{34} \end{bmatrix}}_{\begin{bmatrix} R_c & T_c \end{bmatrix}} P_w$$

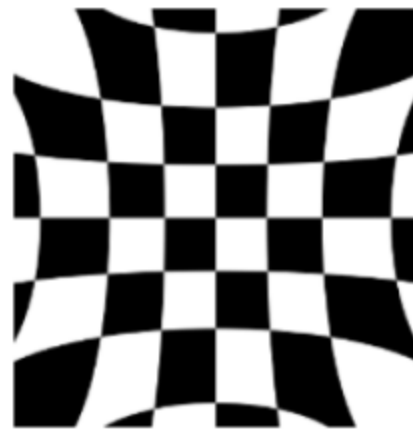
## Calibração da câmera



**Sem  
distorção**



**Distorção radial  
positiva**



**Distorção radial  
negativa**

$$x''_R = x'(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

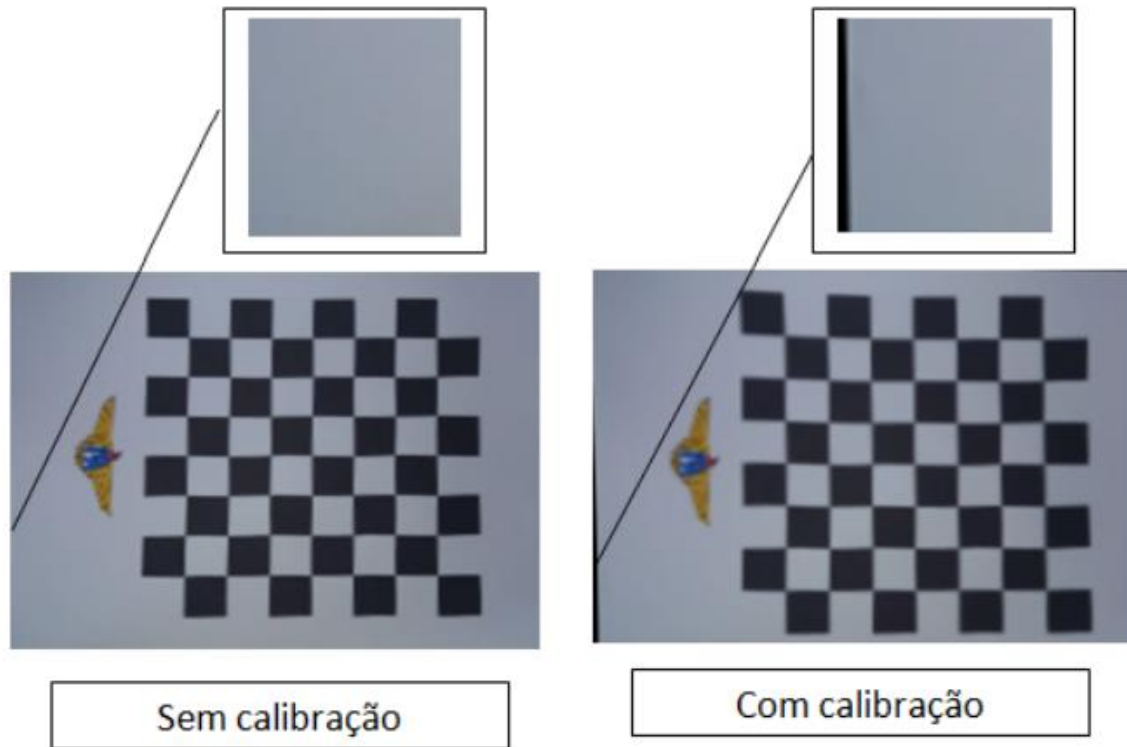
$$y''_R = y'(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

$$x''_T = x' + (2p_1x'y' + p_2(r^2 + x'^2))$$

$$y''_T = y' + (p_1(r^2 + x'^2) + 2p_2x'y')$$

onde:  $r^2 = x'^2 + y'^2$

## Uso do OpenCV para calibração

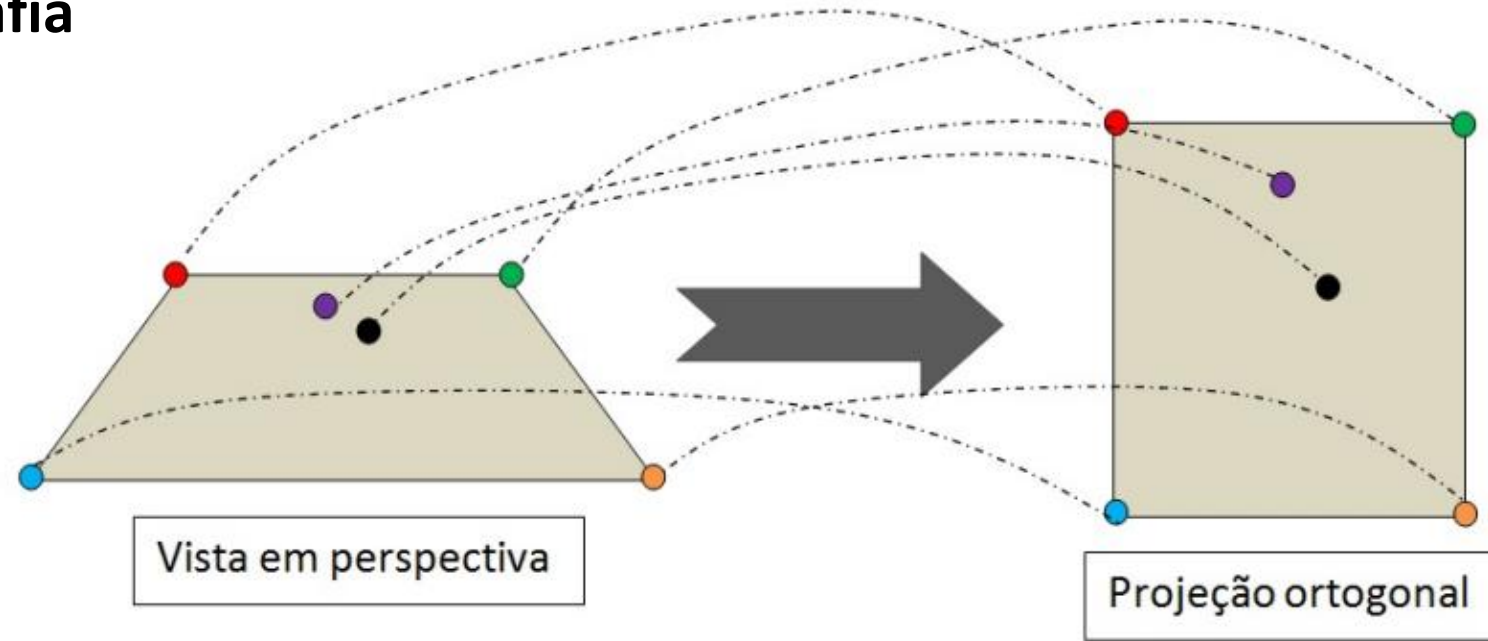


Método `calibrateCamera` permite calibrar usando imagens de *chessboard*

$$K_c = \begin{bmatrix} 2,14 \times 10^3 & 0 & 9 \times 10^2 \\ 0 & 2,11 \times 10^3 & 8,21 \times 10^2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$dist = \begin{bmatrix} \underbrace{9,84 \times 10^{-2}}_{k_1} & \underbrace{-0,31}_{k_2} & \underbrace{4,69 \times 10^{-3}}_{p_1} & \underbrace{3,92 \times 10^{-3}}_{p_2} & \underbrace{0,8}_{k_3} \end{bmatrix}$$

## Homografia



$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



## Navegação por waypoints

Neste tipo de navegação uma lista de *waypoints* é enviada para o robô seguir ponto a ponto. A missão é completada quando o robô seguir todos os pontos.

onde:

$R_w$  = raio do *waypoint* [m]

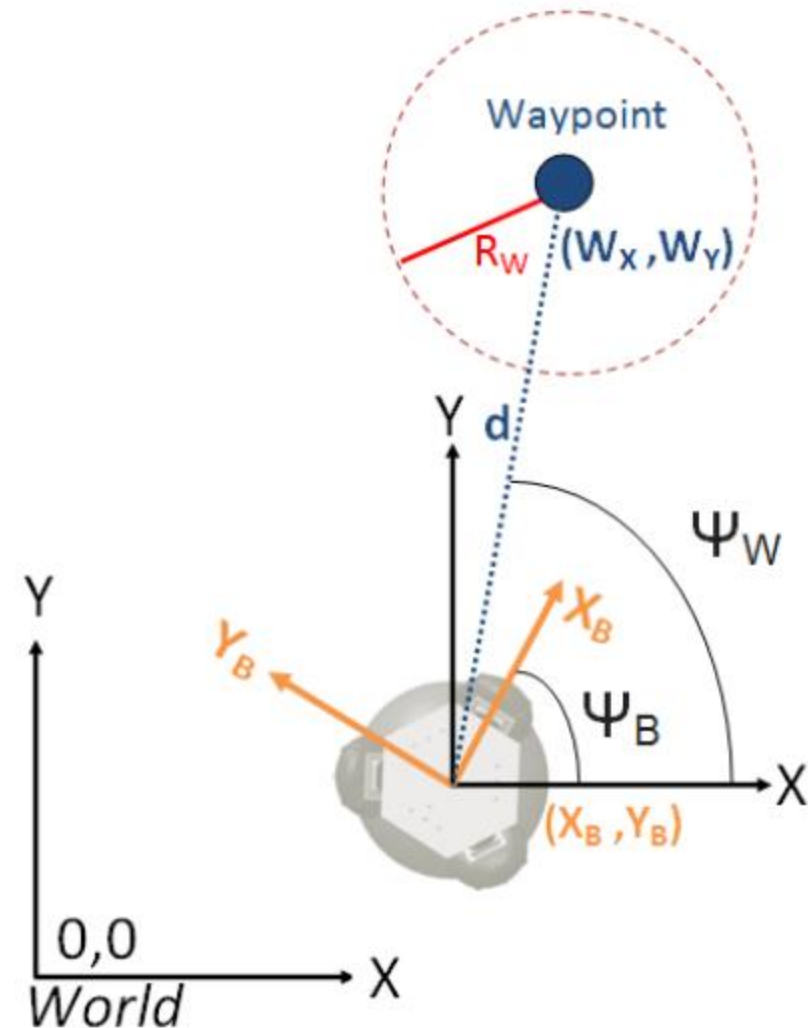
$W_x, W_y$  = posição do *waypoint* [m]

$\psi_w$  = ângulo do *waypoint* [°]

$X_B, Y_B$  = posição do *ballbot* [m]

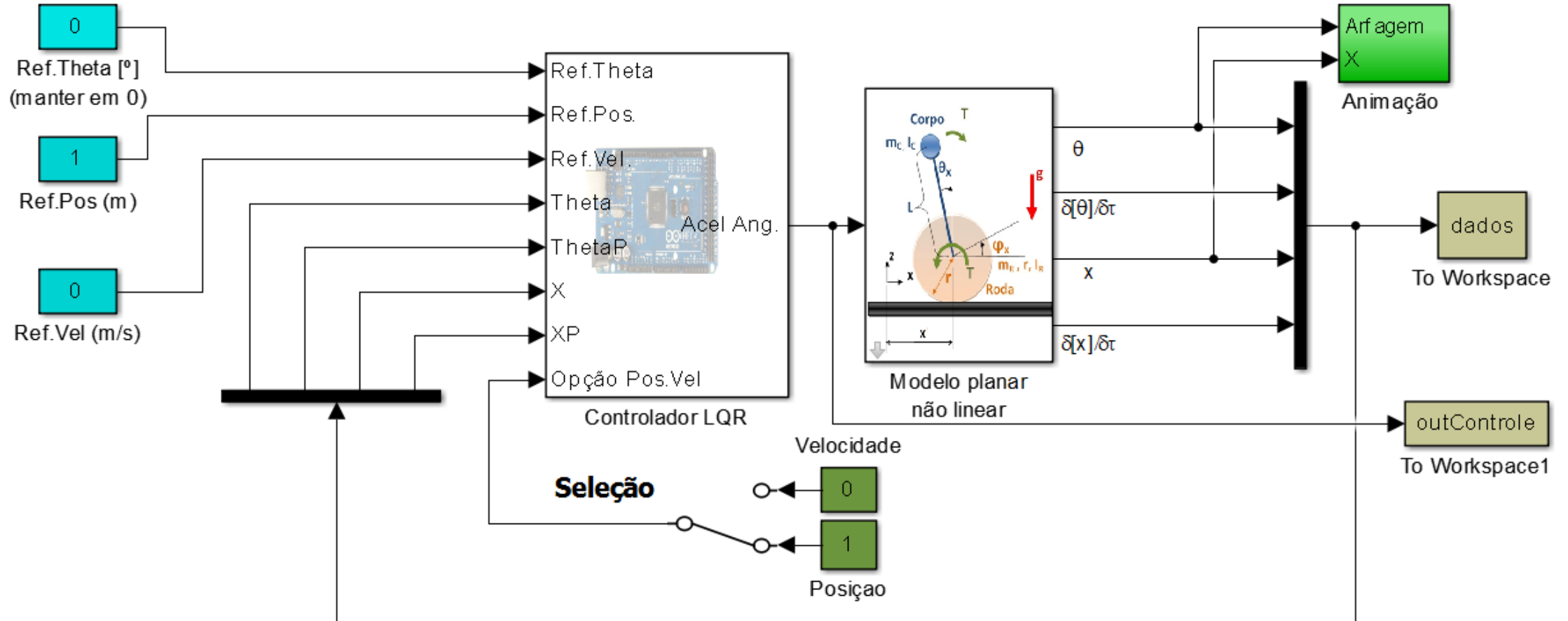
$\psi_B$  = ângulo do *ballbot* [°]

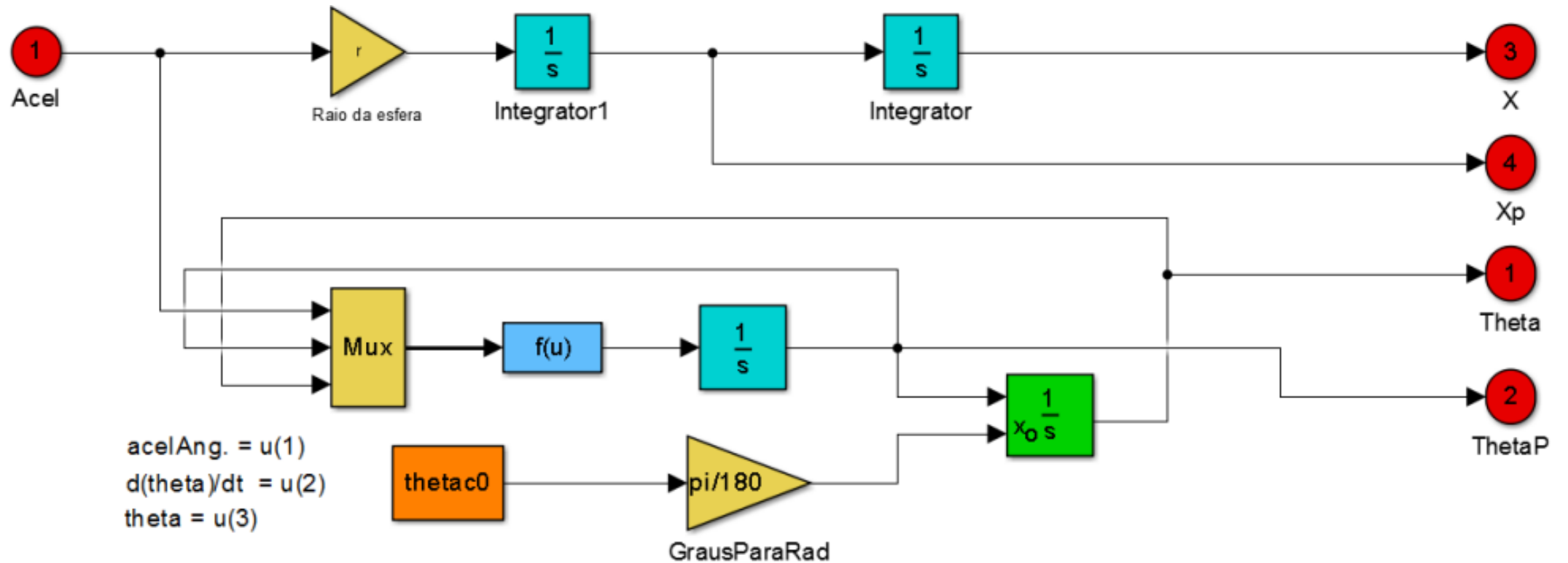
$d$  = distância euclidiana entre o *ballbot* e o *waypoint*



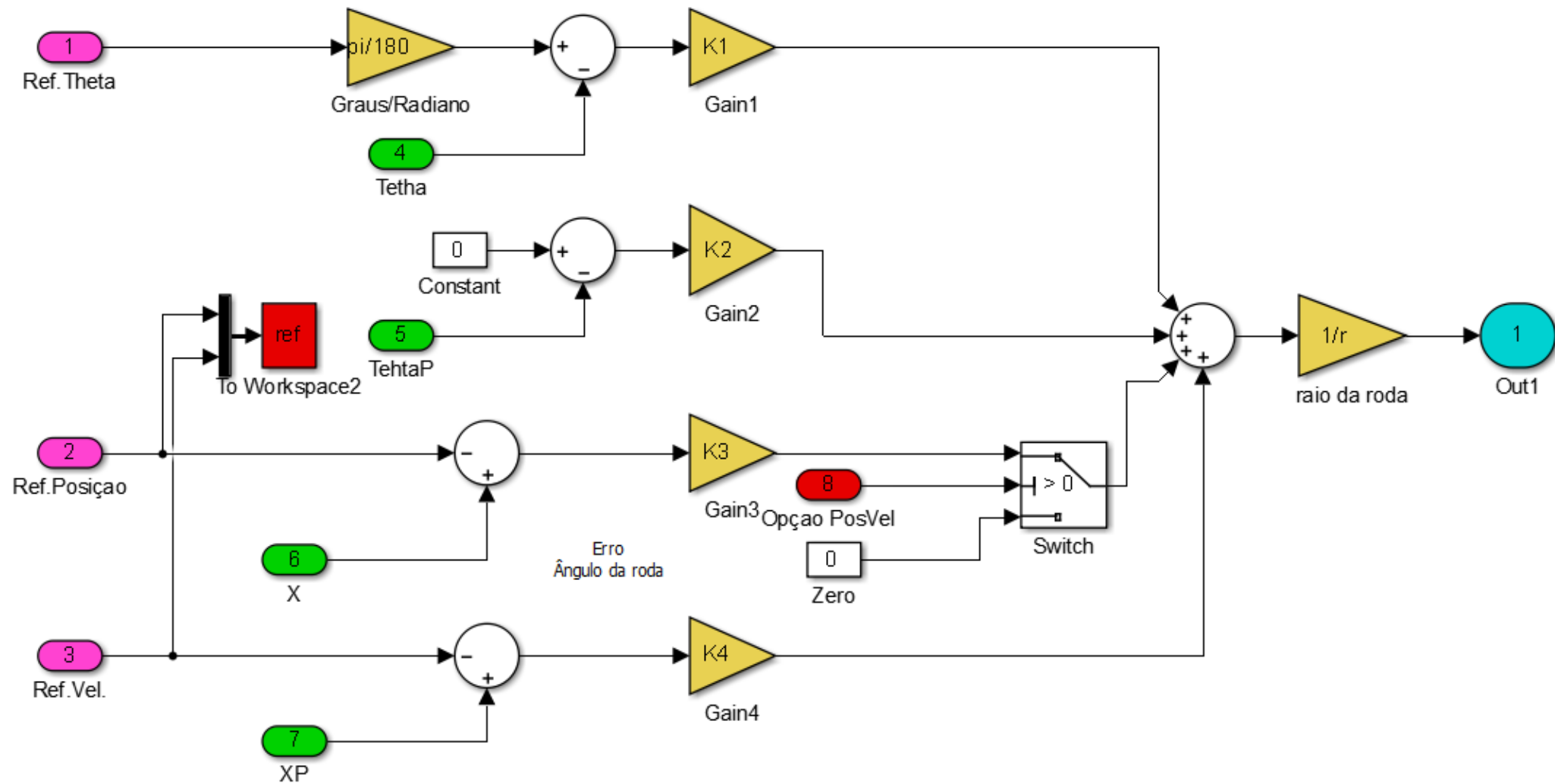
## MATLAB / Simulink

### Referência



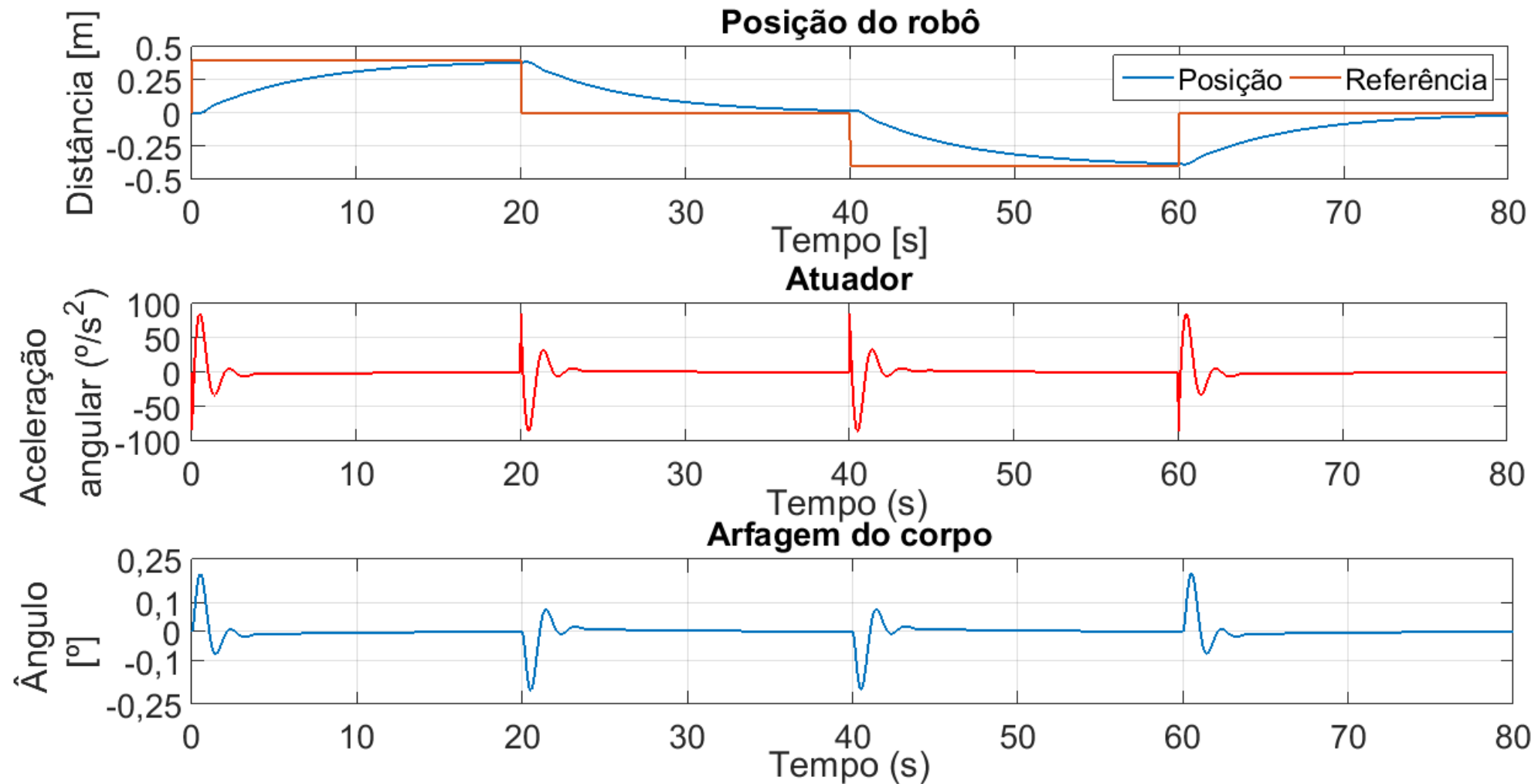


Bloco do modelo planar não linear

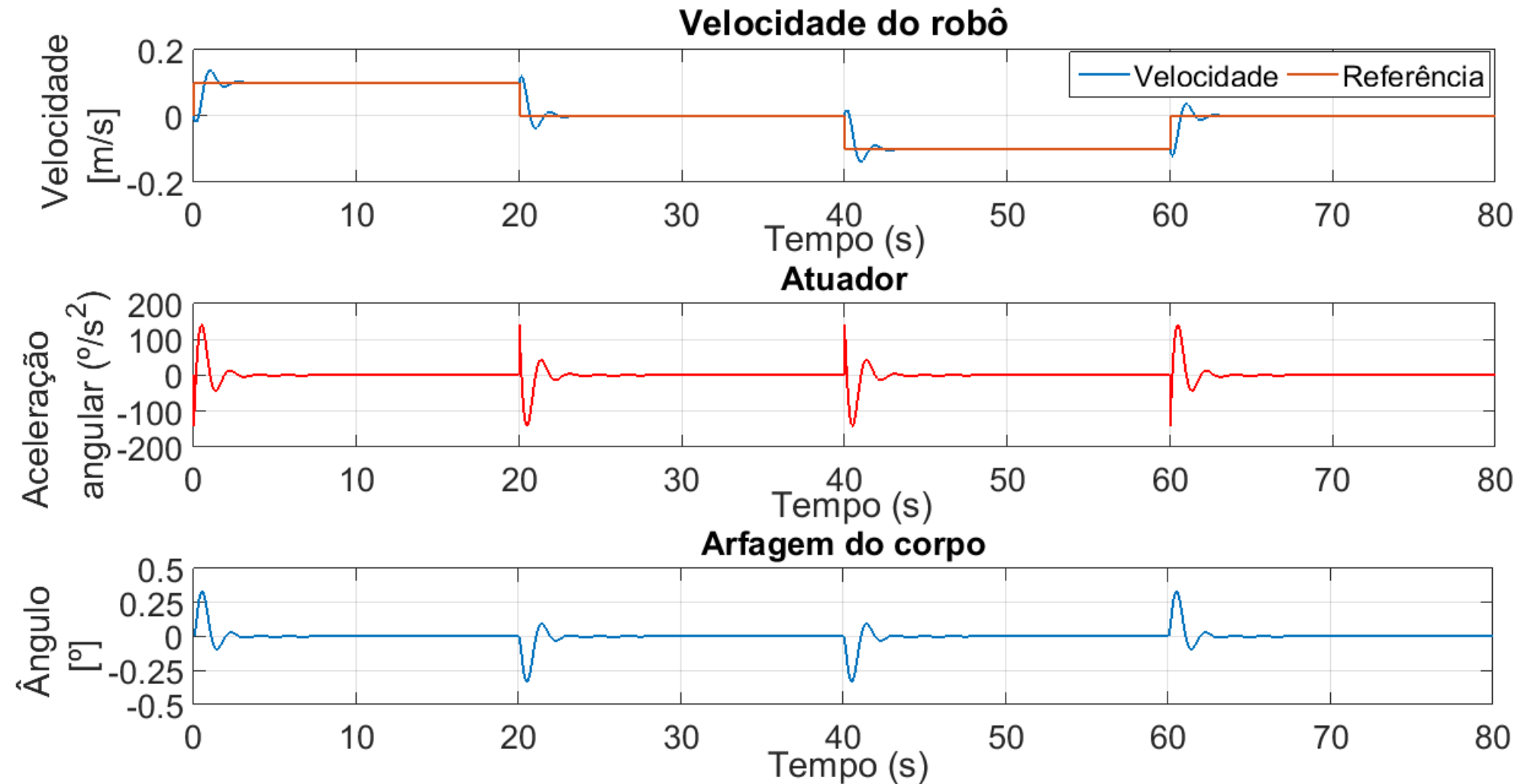


Bloco do controlador LQR

## Resultados da simulação do modelo planar (MATLAB/Simulink)

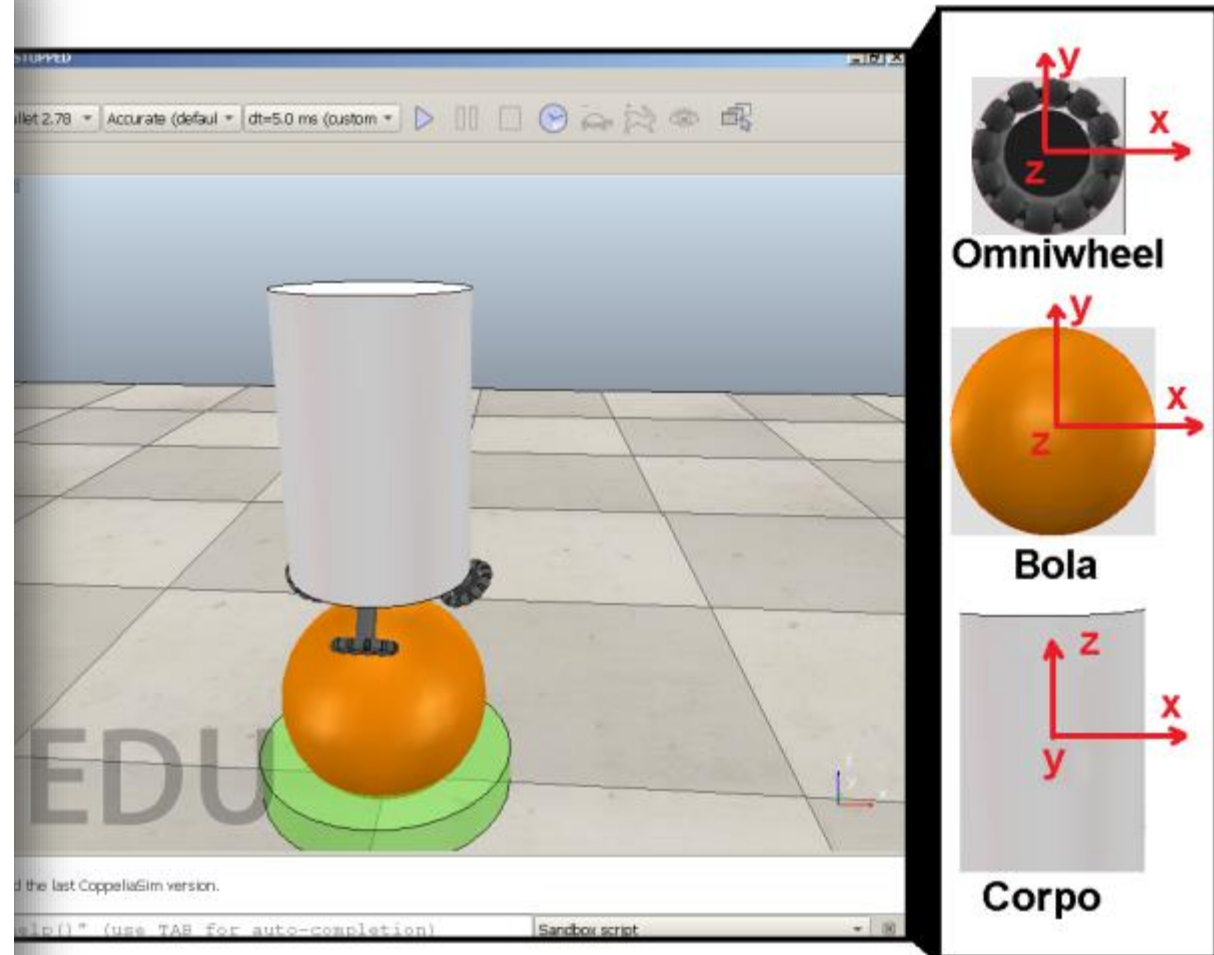


## Resultados da simulação do modelo planar (MATLAB/Simulink)

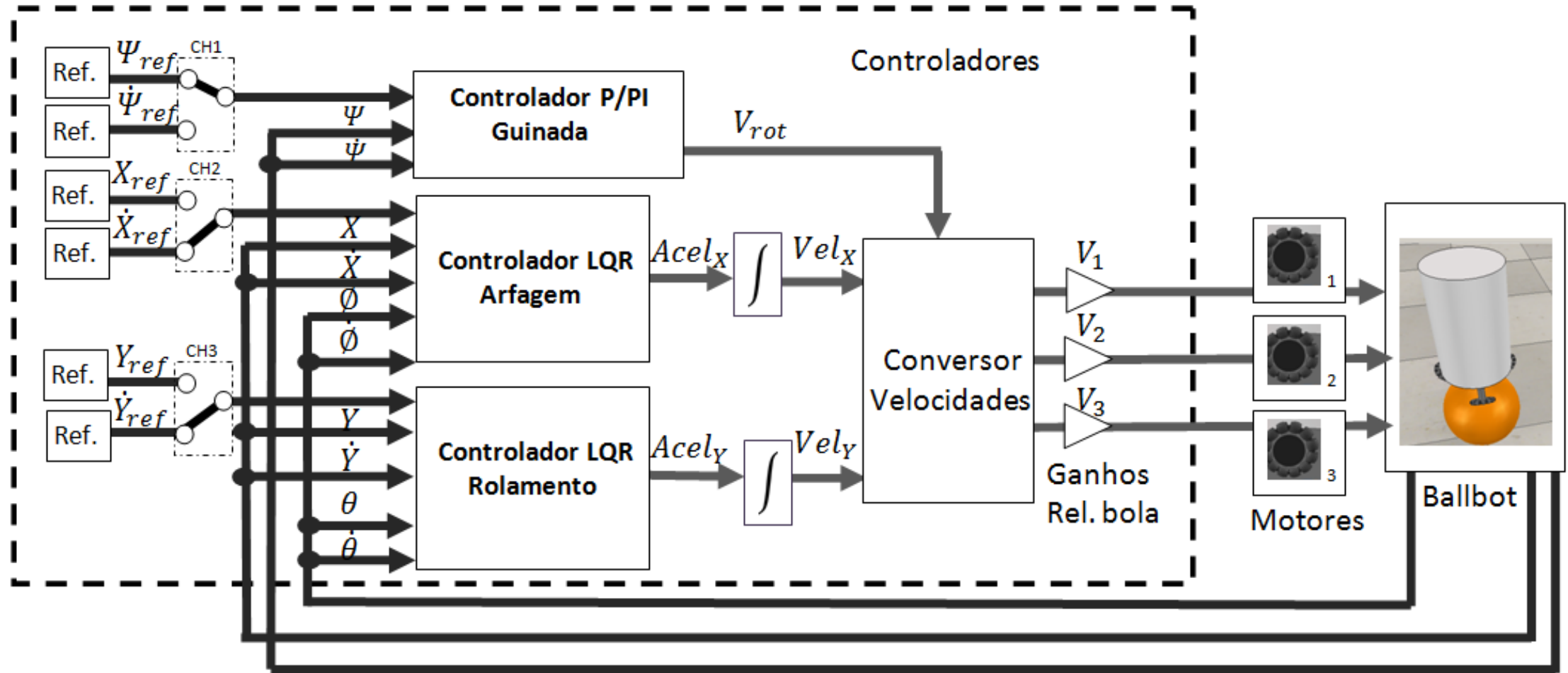


## Simulação 3D usando o CoppeliaSim (V-REP)

Parâmetro	Valor	Unidade
Aceleração da gravidade	9,81	$m/s^2$
Diâmetro da esfera	0,24	m
Massa da esfera	0,58	kg
Momento de inércia da esfera em X	0,0134	$kgm^2$
Momento de inércia da esfera em Y	0,0134	$kgm^2$
Momento de inércia da esfera em Z	0,0134	$kgm^2$
Diâmetro do corpo cilíndrico	0,2	m
Altura do corpo cilíndrico	0,3	m
Massa do corpo cilíndrico	1,6	kg
Momento de inércia do corpo em X	0,0413	$kgm^2$
Momento de inércia do corpo em Y	0,0413	$kgm^2$
Momento de inércia do corpo em Z	0,0720	$kgm^2$
Diâmetro da <i>omniwheel</i>	7	cm
Espessura da <i>omniwheel</i>	1,3	cm
Massa da <i>omniwheel</i>	50	g
Momento de inércia da <i>omniwheel</i> em X	$6,16 \times 10^{-6}$	$kgm^2$
Momento de inércia da <i>omniwheel</i> em Y	$6,16 \times 10^{-6}$	$kgm^2$
Momento de inércia da <i>omniwheel</i> em Z	$1,22 \times 10^{-4}$	$kgm^2$



## Arquitetura de controle





## Modos de operação

Modo de operação	Chave 1 (CH1)	Chave 2 (CH2)	Chave 3 (CH3)
Guinada	UP ( $\Psi_{\text{ref}} = REF$ )	DOWN ( $\dot{X}_{\text{ref}} = 0$ )	DOWN ( $\dot{Y}_{\text{ref}} = 0$ )
Velocidade de guinada	DOWN ( $\dot{\Psi}_{\text{ref}} = REF$ )	DOWN ( $\dot{X}_{\text{ref}} = 0$ )	DOWN ( $\dot{Y}_{\text{ref}} = 0$ )
Posição	DOWN ( $\dot{\Psi}_{\text{ref}} = 0$ )	UP ( $X_{\text{ref}} = REF$ )	UP ( $Y_{\text{ref}} = REF$ )
Velocidade de translação	DOWN ( $\dot{\Psi}_{\text{ref}} = 0$ )	DOWN ( $\dot{X}_{\text{ref}} = REF$ )	DOWN ( $\dot{Y}_{\text{ref}} = REF$ )
Guinada e posição	UP ( $\Psi_{\text{ref}} = REF$ )	UP ( $X_{\text{ref}} = REF$ )	UP ( $Y_{\text{ref}} = REF$ )

## Conversão de velocidades

$$V_1 = -Vel_x \cos(\beta) + V_{rot}$$

$$V_2 = (0,86Vel_x + 0,5Vel_y) \cos(\beta)$$

$$V_3 = (-0,86Vel_x + 0,5Vel_y) \cos(\beta)$$

onde:

$V_{rot}$  = velocidade de guinada;

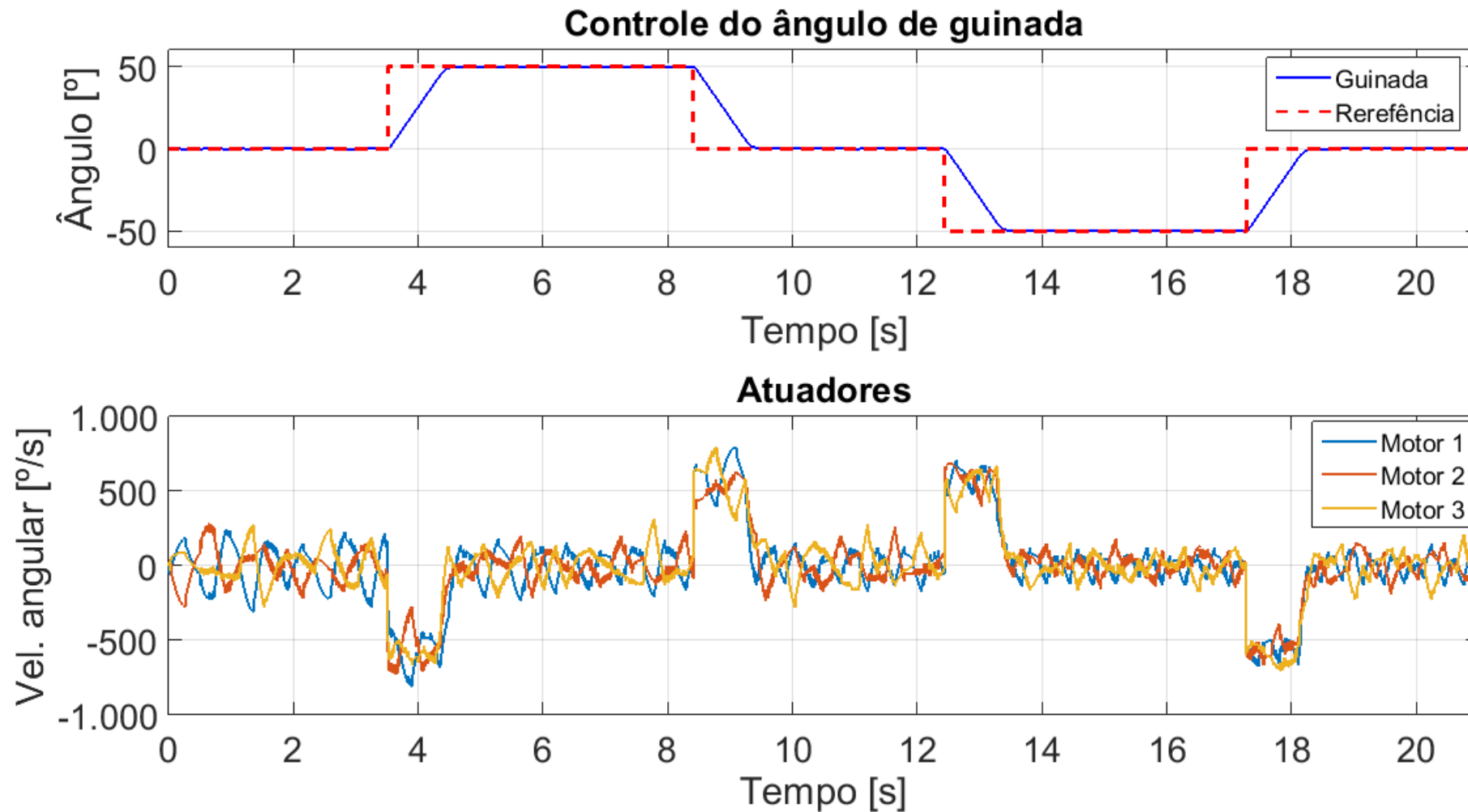
$\beta$  = ângulo dos motores;

$V_1, V_2$  e  $V_3$  = velocidade dos motores;

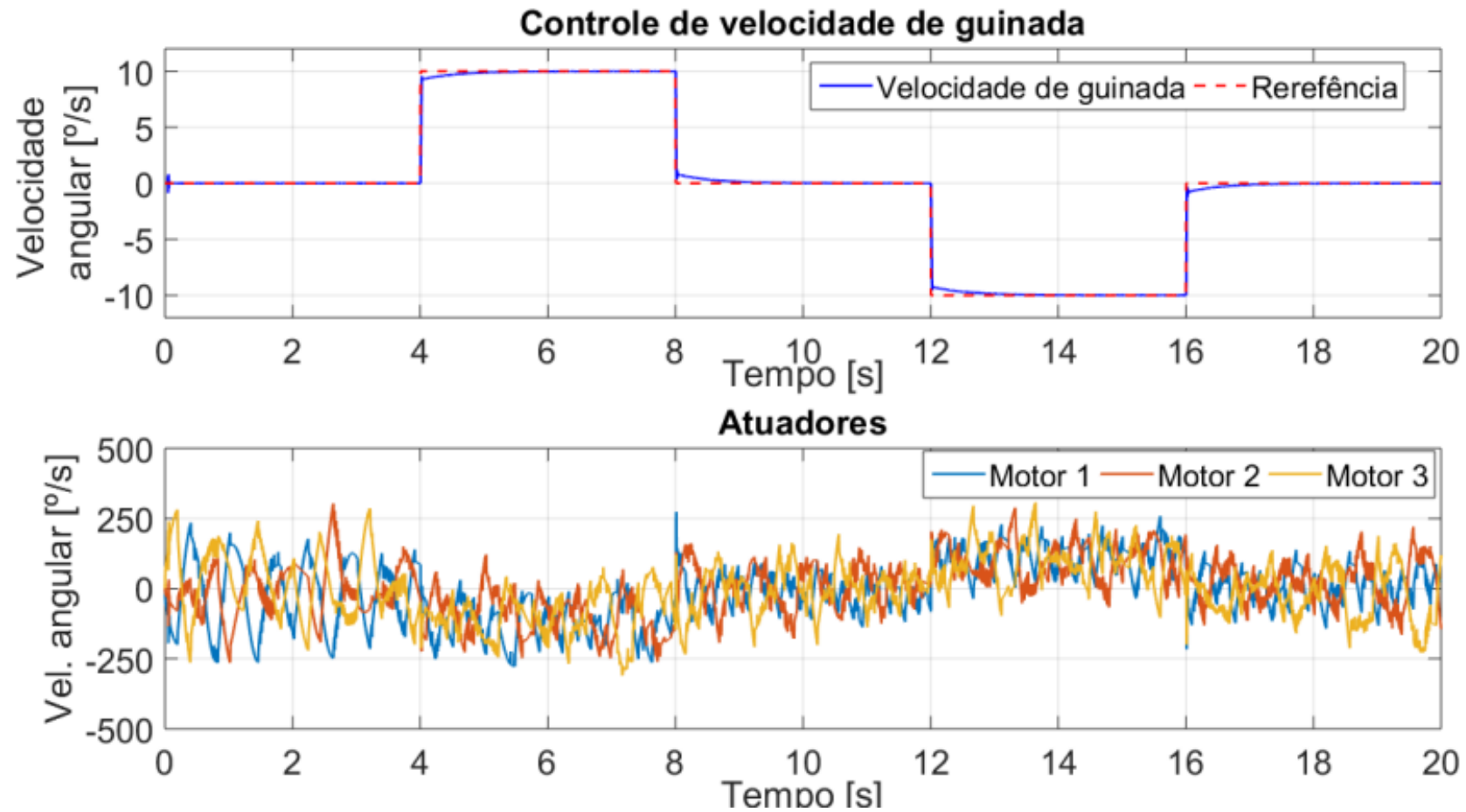
$Vel_x$  = velocidade em x;

$Vel_y$  = velocidade em y.

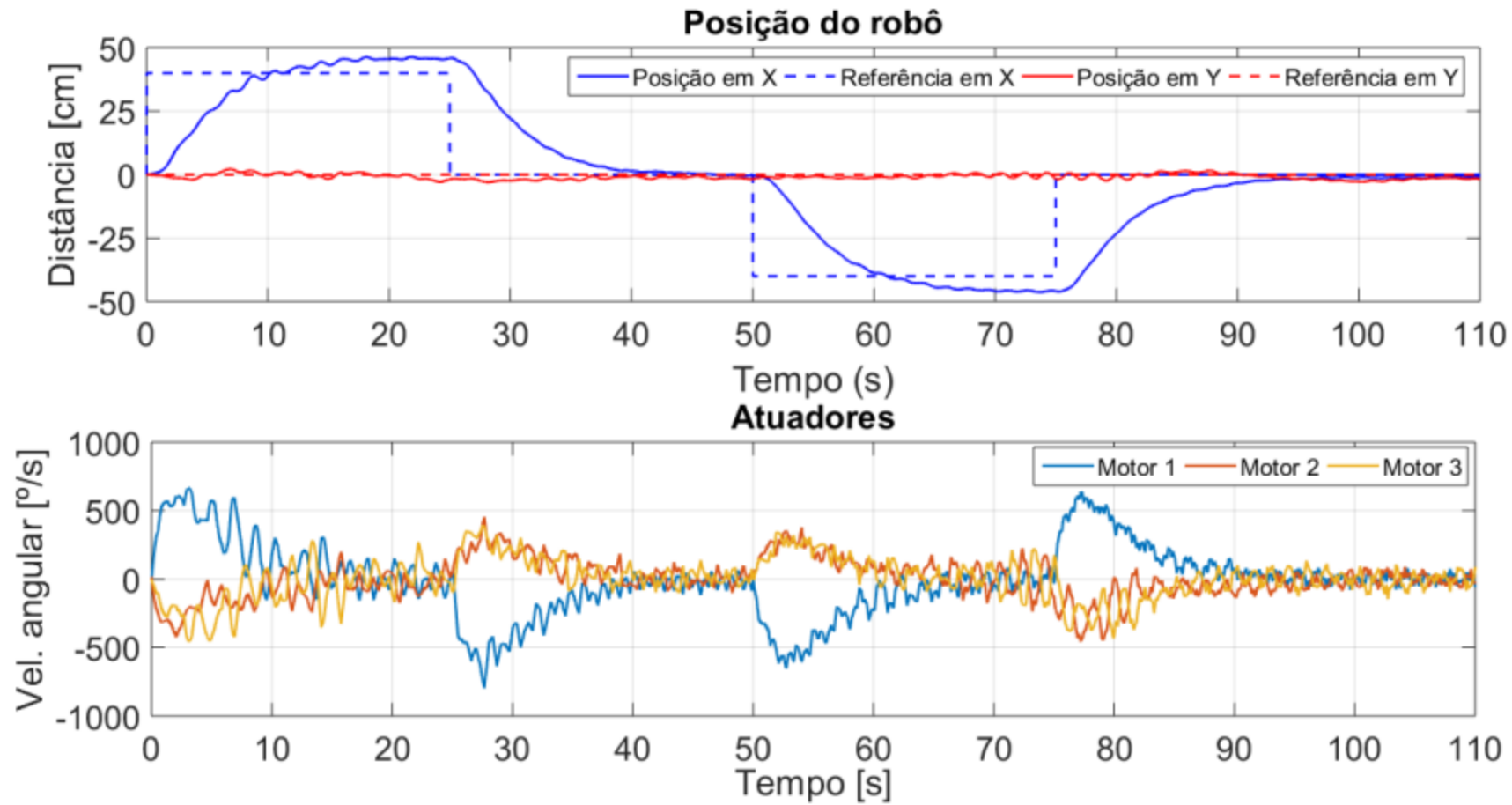
## Controle do ângulo de guinada, passo de simulação = 10 ms



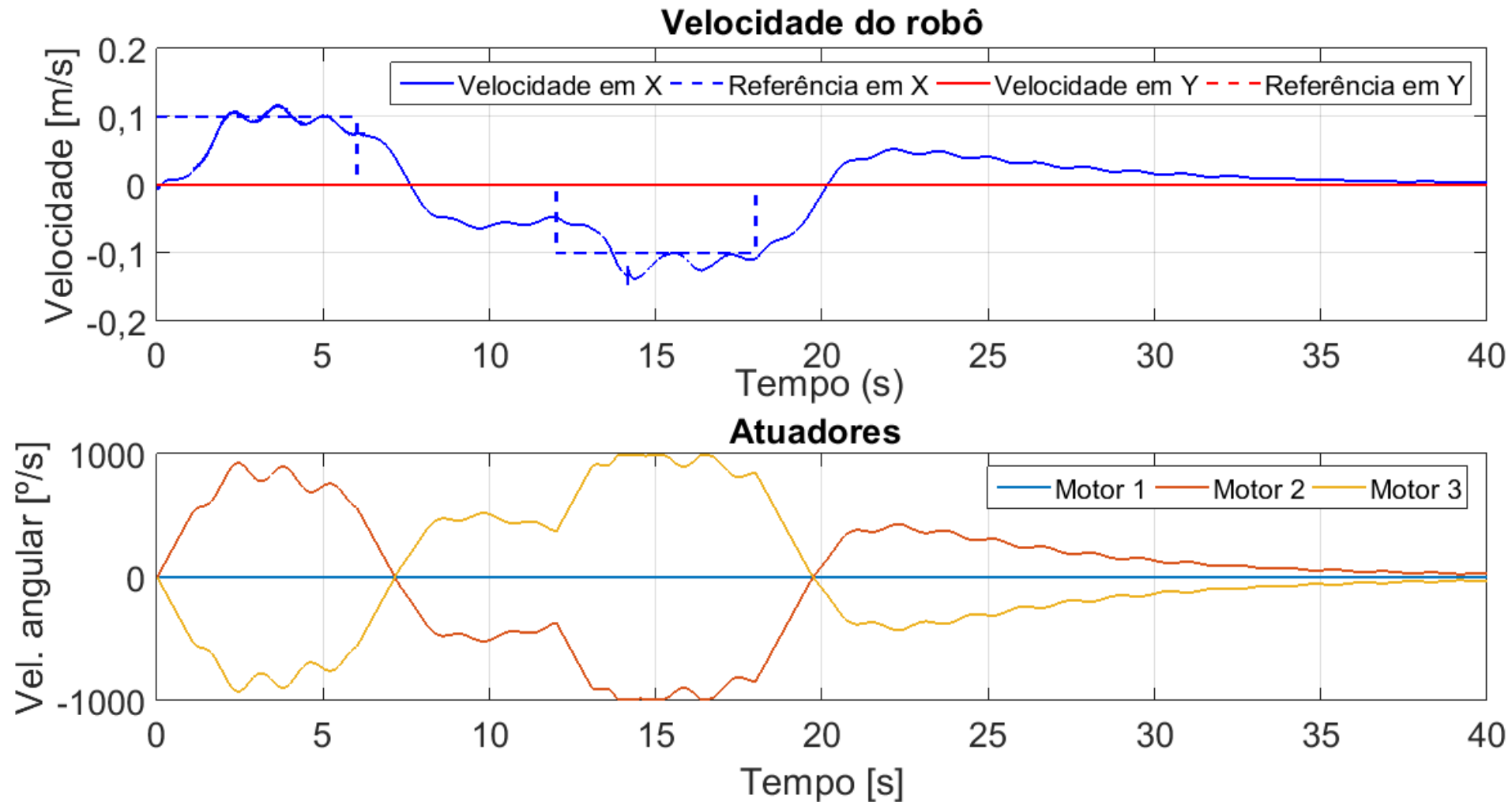
## Controle de velocidade de guinada



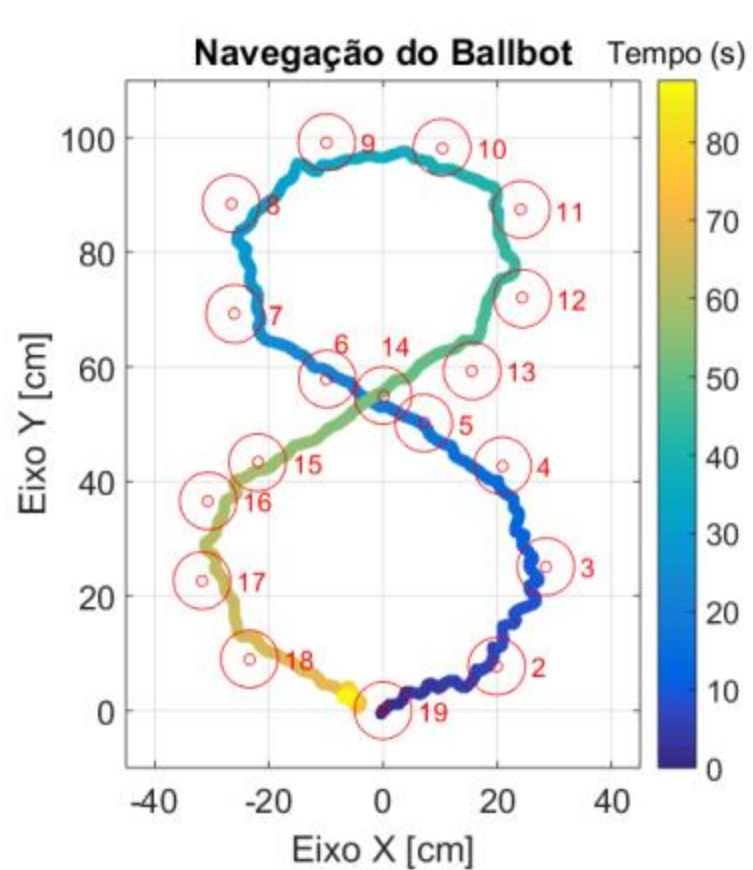
## Controle de posição



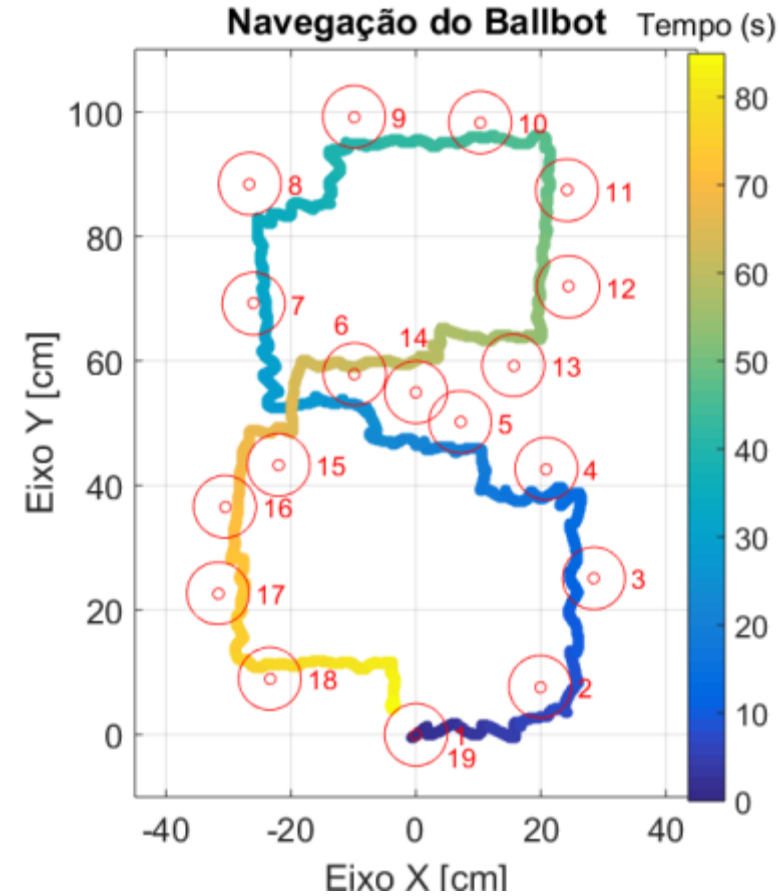
## Controle de velocidade



## Navegação autônoma por *waypoints*

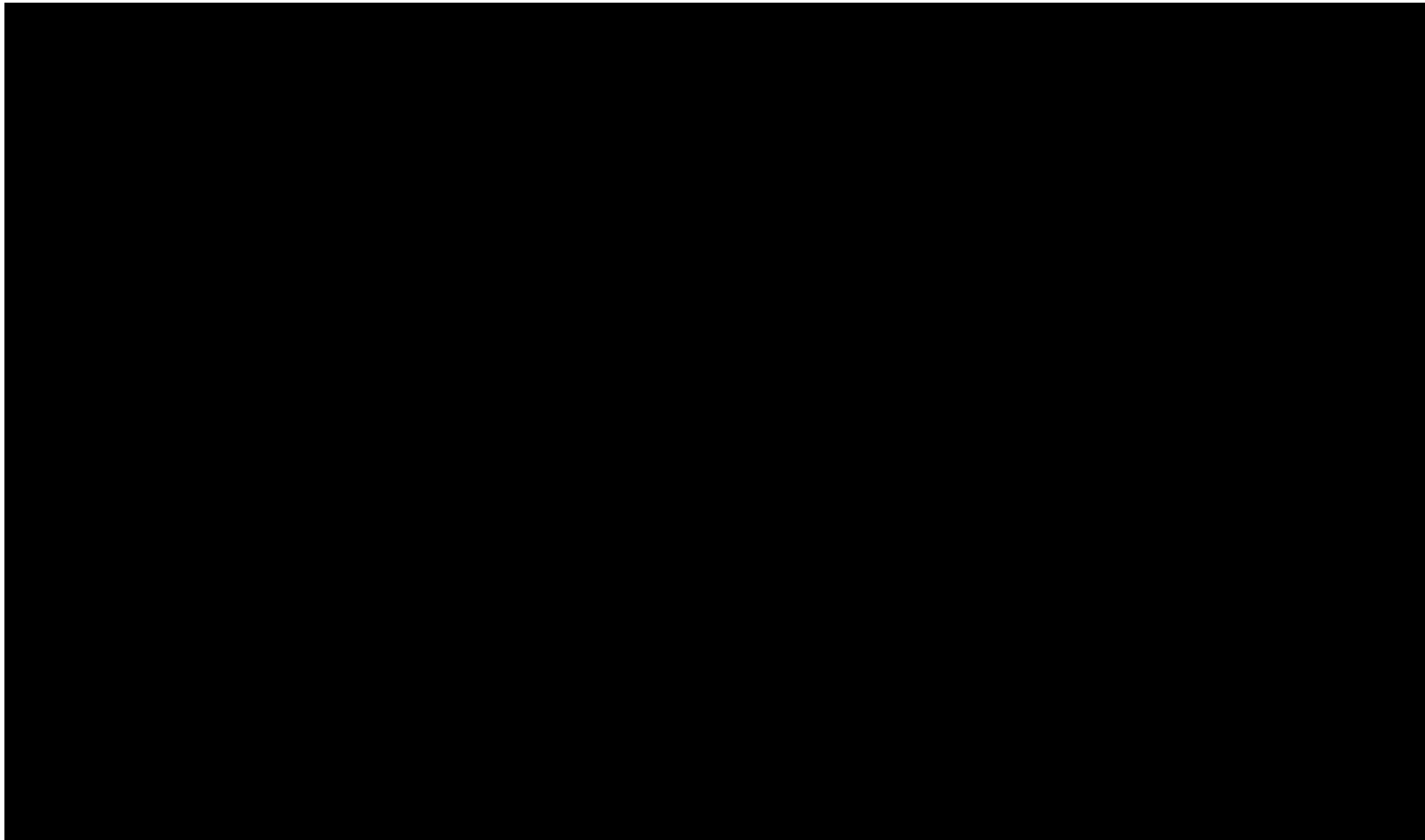


Algoritmo com prioridade de orientação



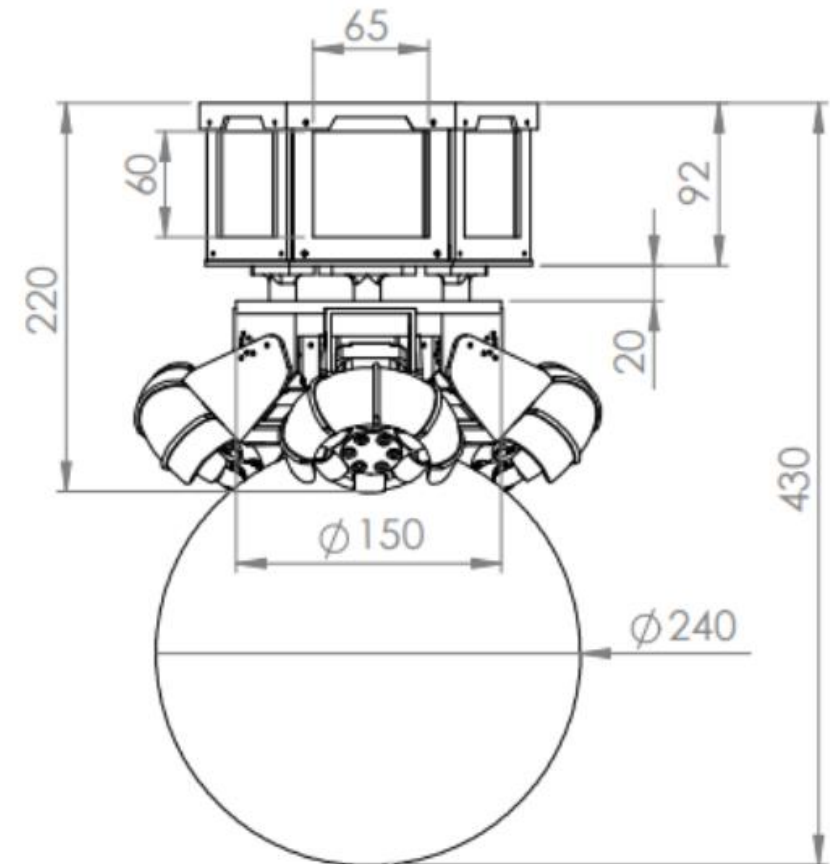
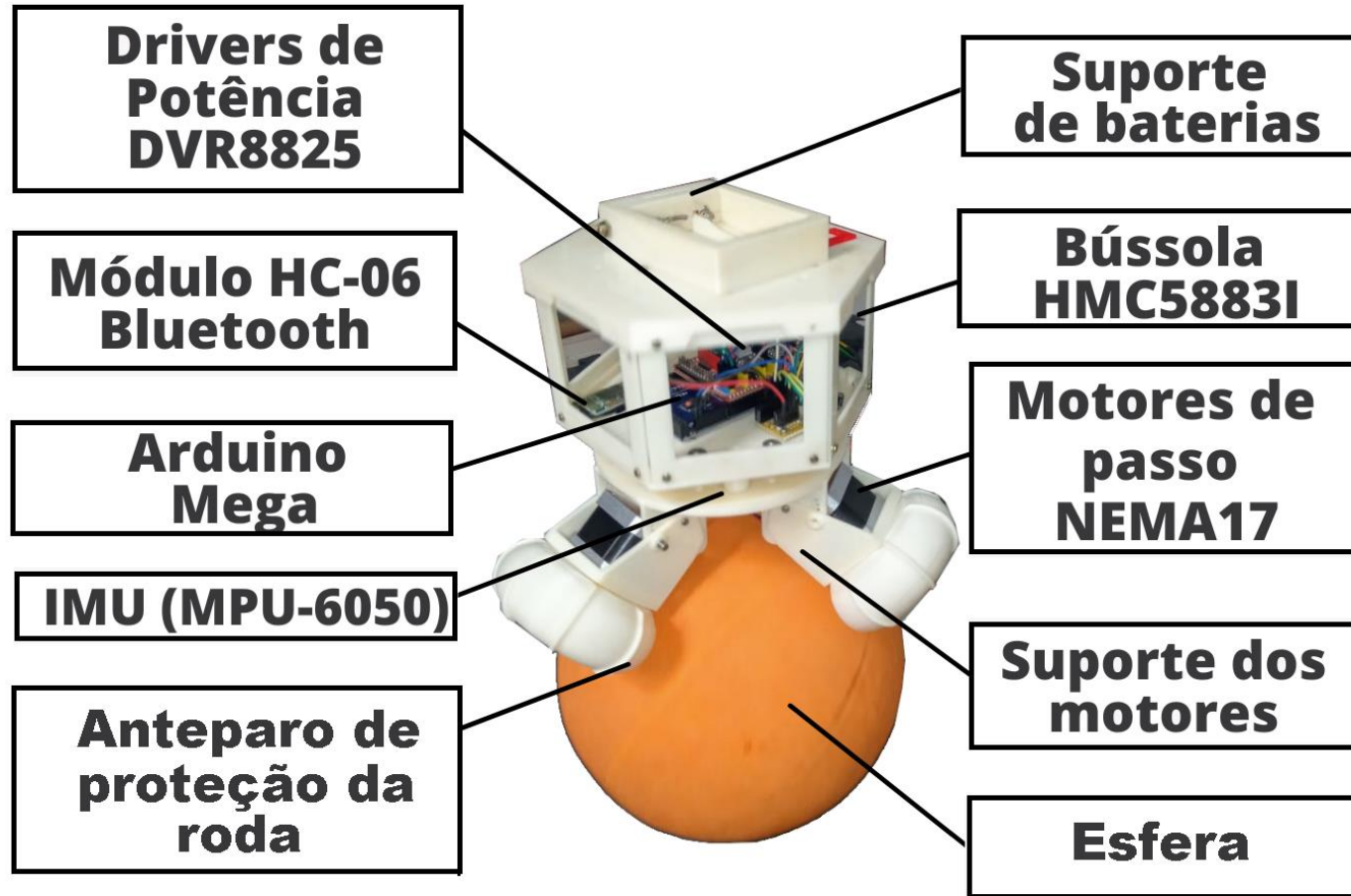
Robô com orientação fixa

## Vídeo da navegação autônoma por *waypoints* no CoppeliaSim

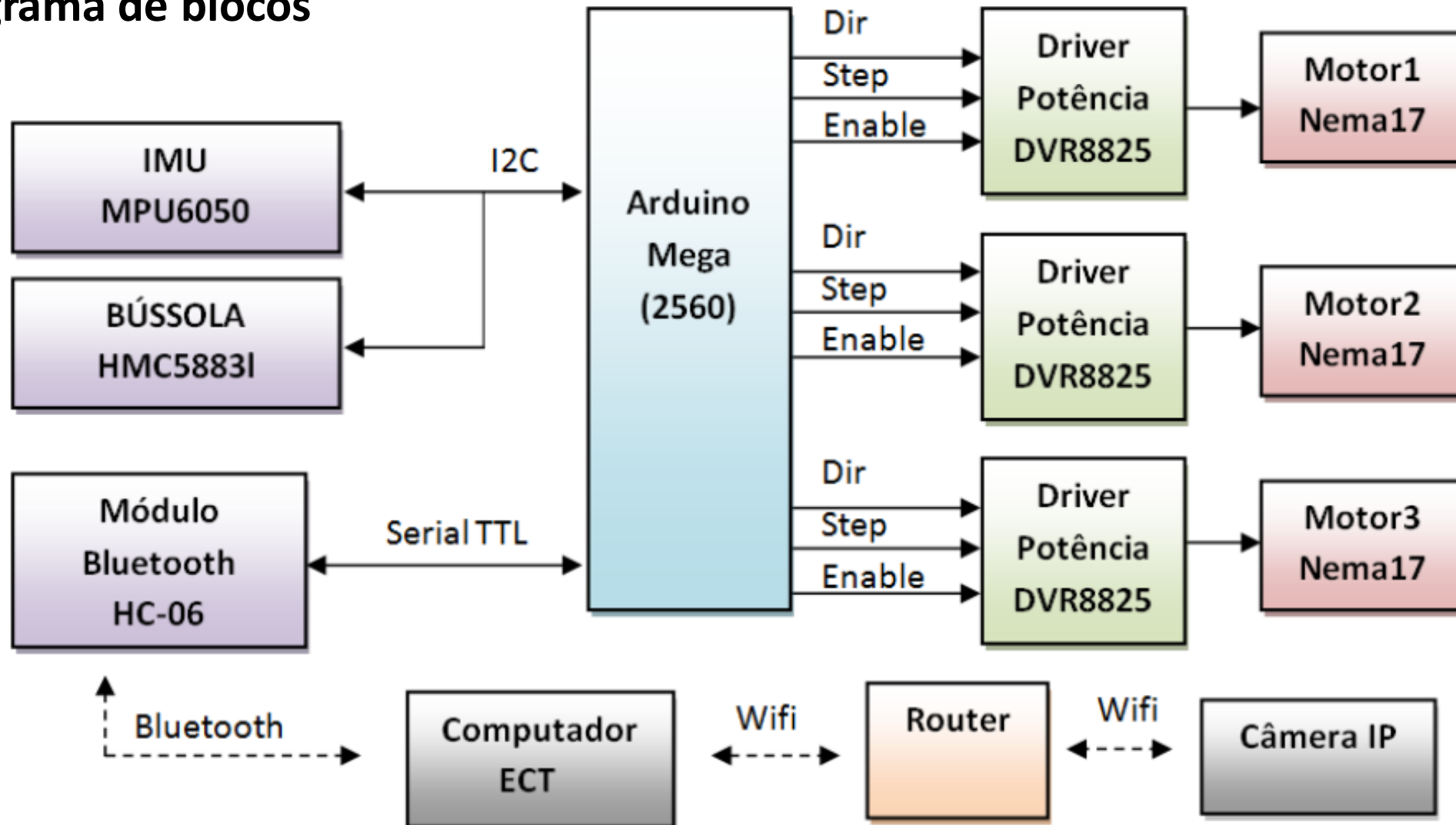




## Estrutura do robô Ballbot



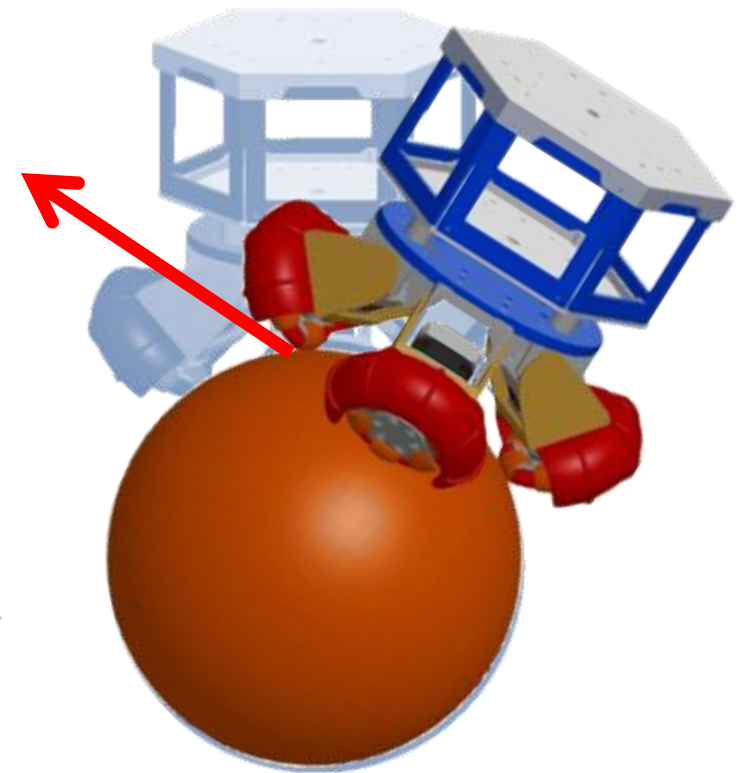
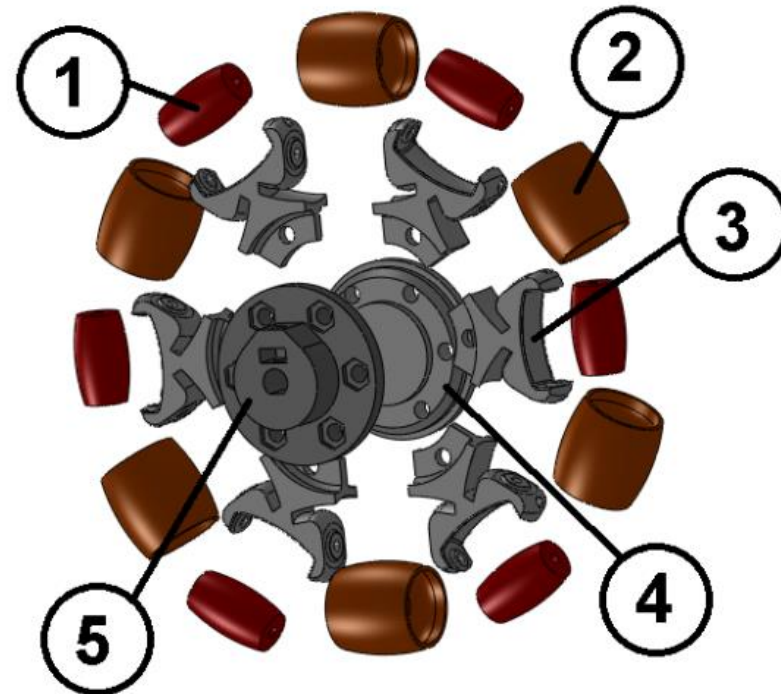
## Diagrama de blocos

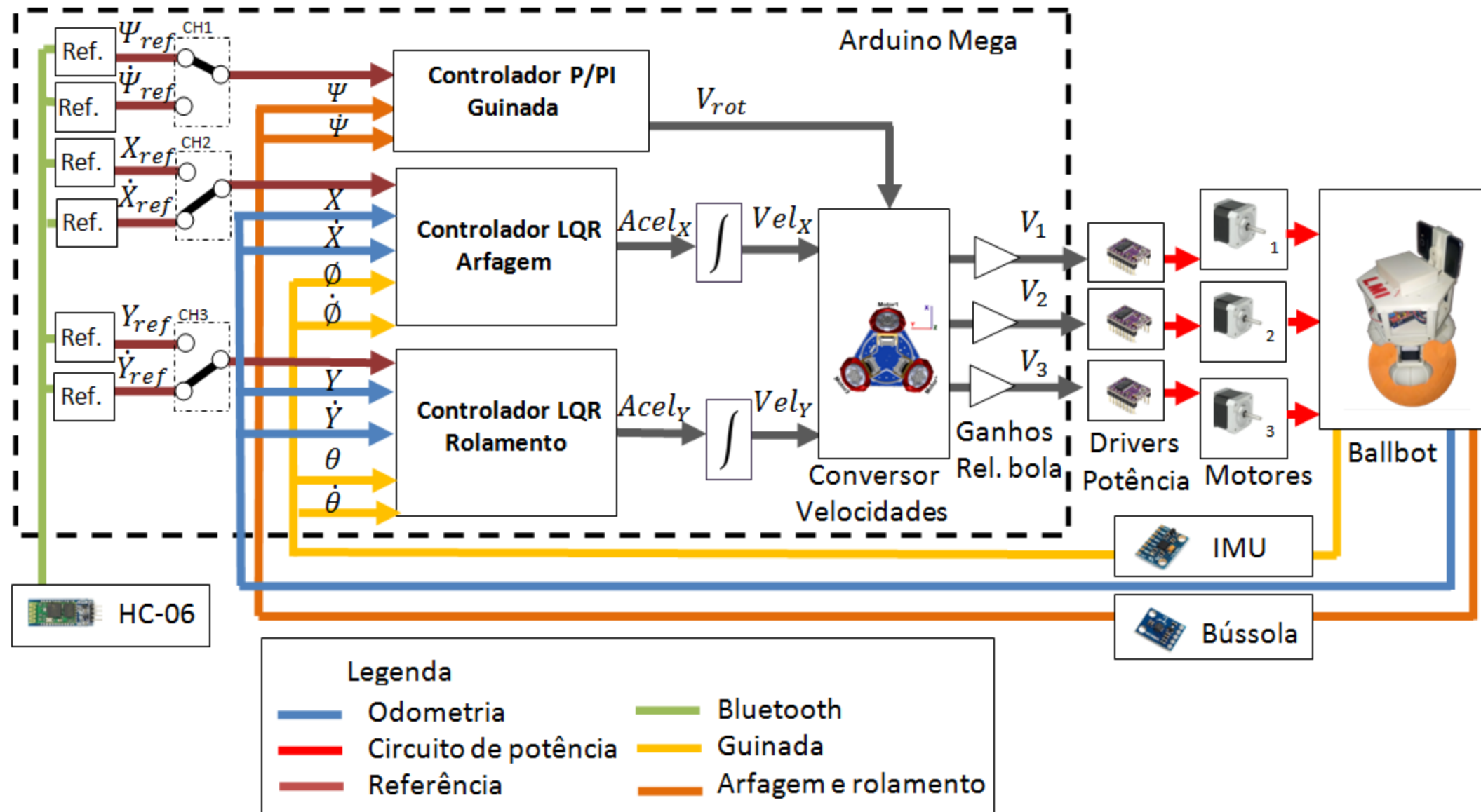


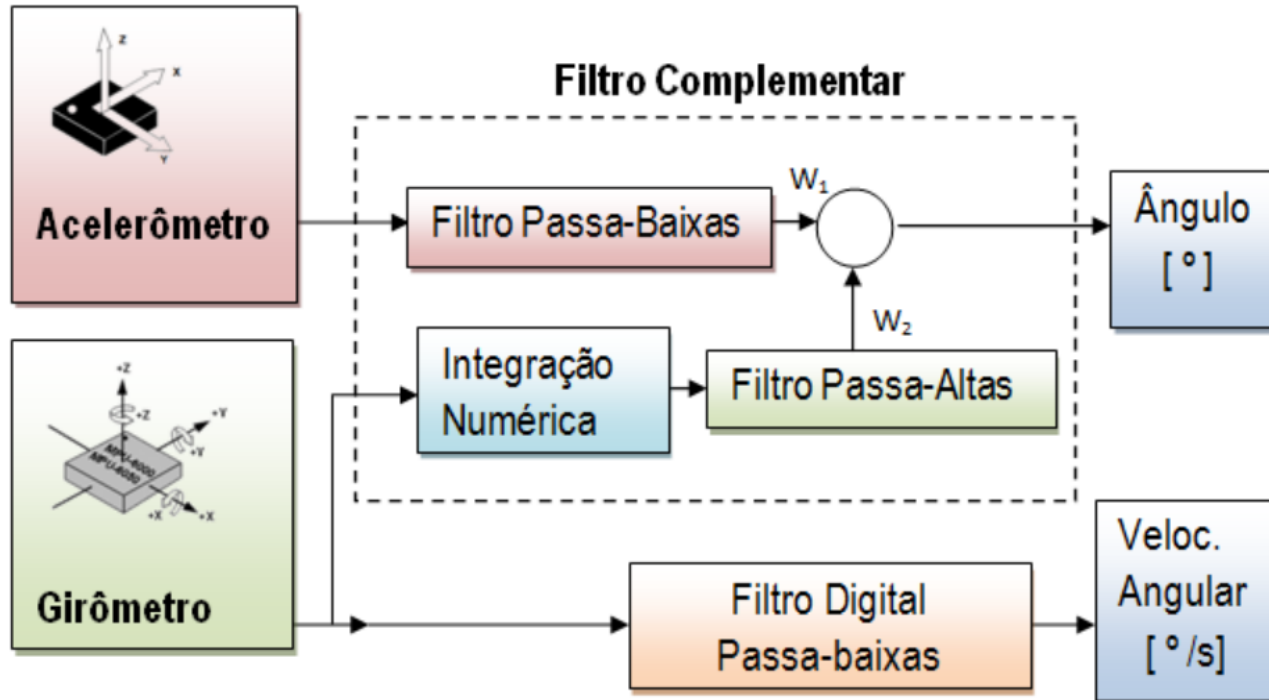
## Técnicas para melhorar o desempenho do sistema embarcado

- A taxa de comunicação serial entre o Arduino e o módulo *Bluetooth* foi aumentada para 115200 bps.
- Uso da interrupção de temporizadores do Arduino para gerar os sinais para os *drivers* de potência dos motores.
- Geração dos pulsos para os *drivers* de potência dos motores usando os pinos de saída dos registradores das portas do Arduino.
- Uso da técnica de deslocamento de bits para operações de multiplicação/divisão no programa executado no Arduino.

## Detalhes da roda omnidirecional







$$\theta = \alpha(\theta + \theta_{\text{gyro}} dt) + (1 - \alpha) \theta_{\text{acc}}$$

onde:

$\theta$  = ângulo de arfagem filtrado [°]

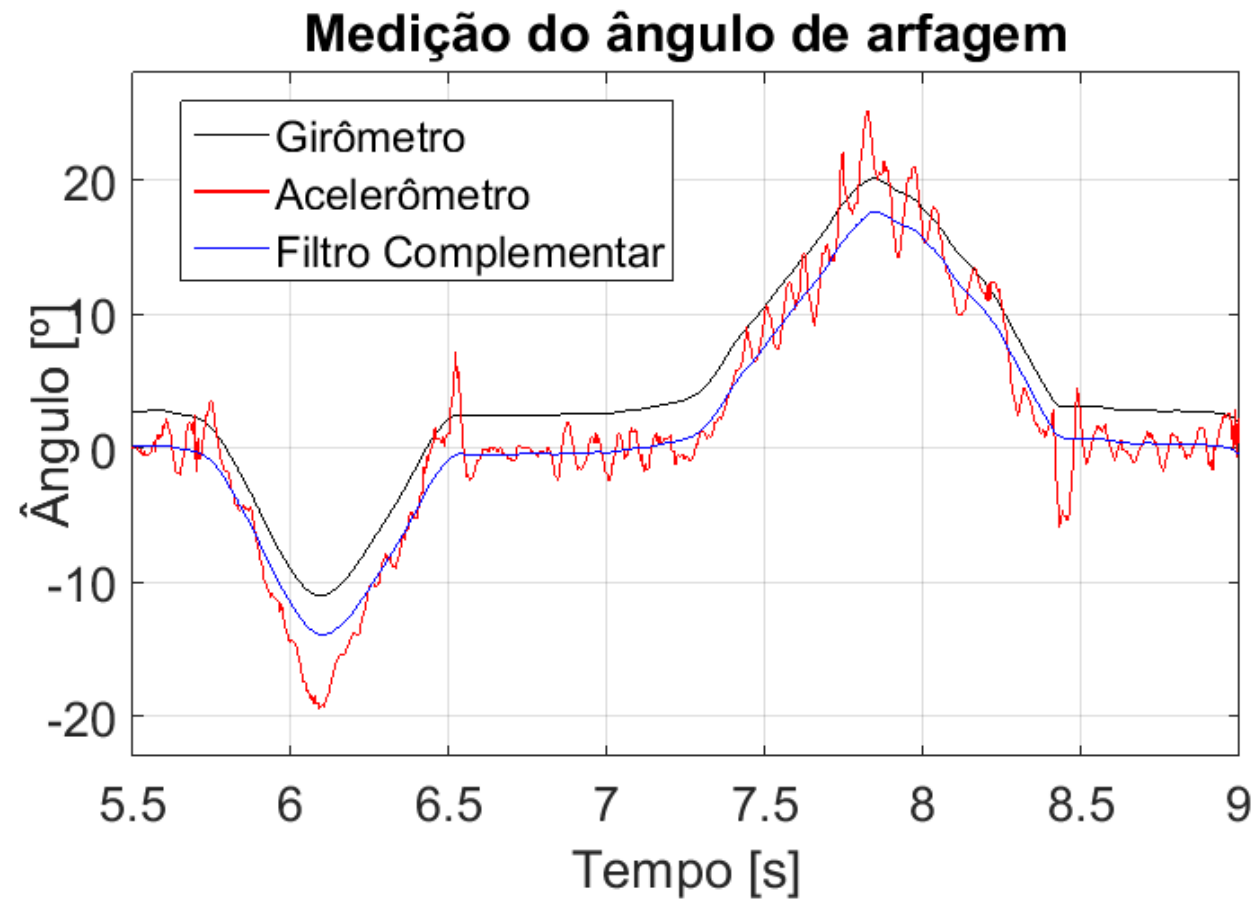
$\theta_{\text{acc}}$  = ângulo obtido pelo acelerômetro [°]

$\theta_{\text{gyro}}$  = ângulo obtido pelo girômetro [°]

$\alpha$  = constante do filtro

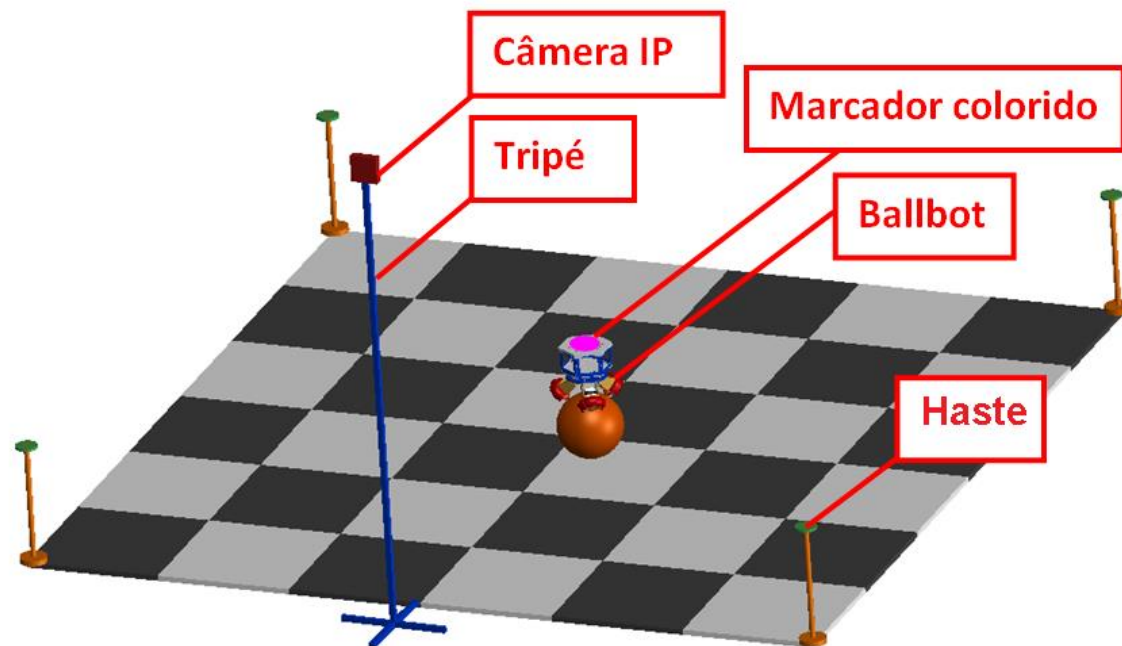
$dt$  = passo de integração

## Resultado do filtro complementar



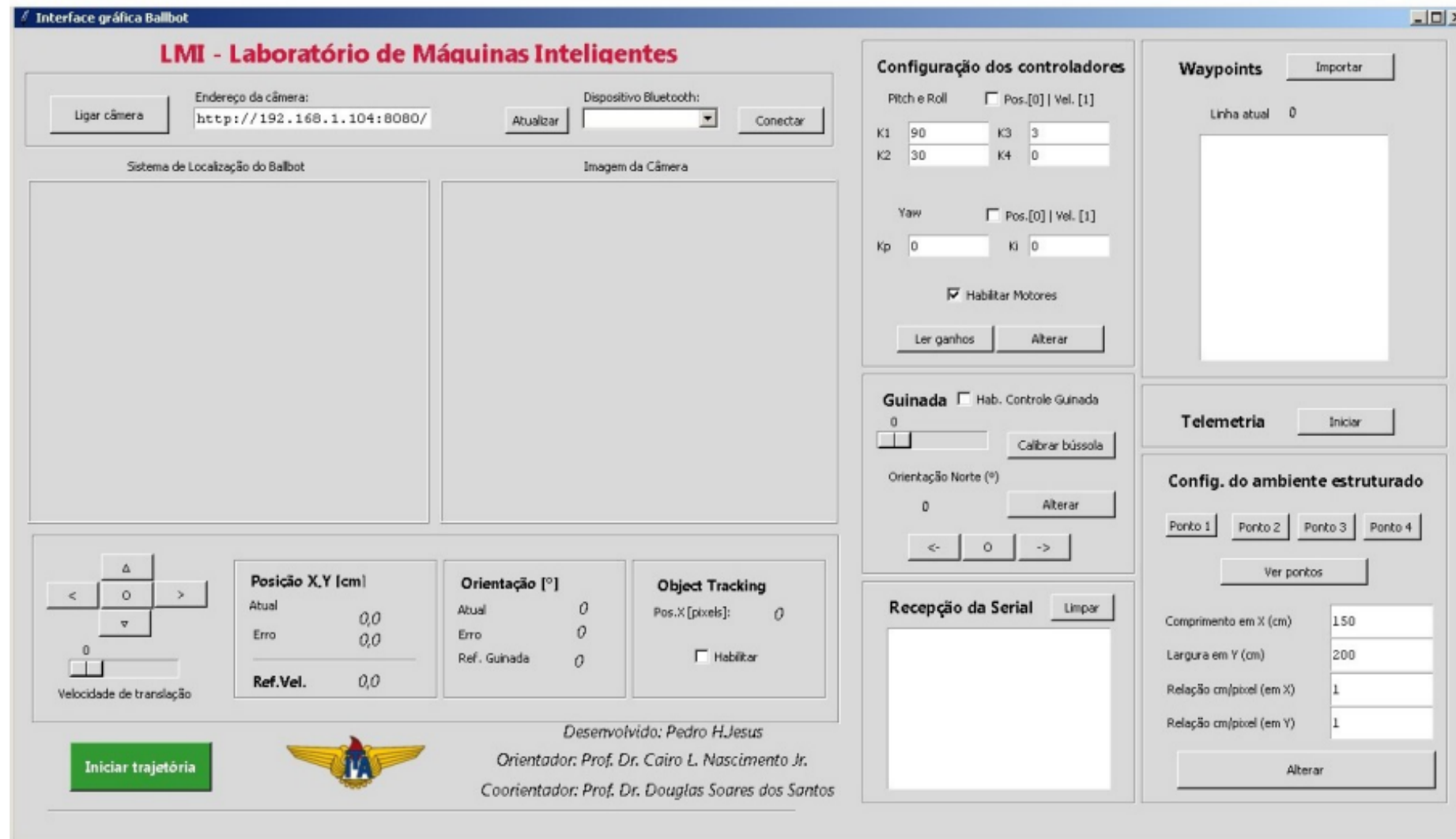


## Ambiente de testes

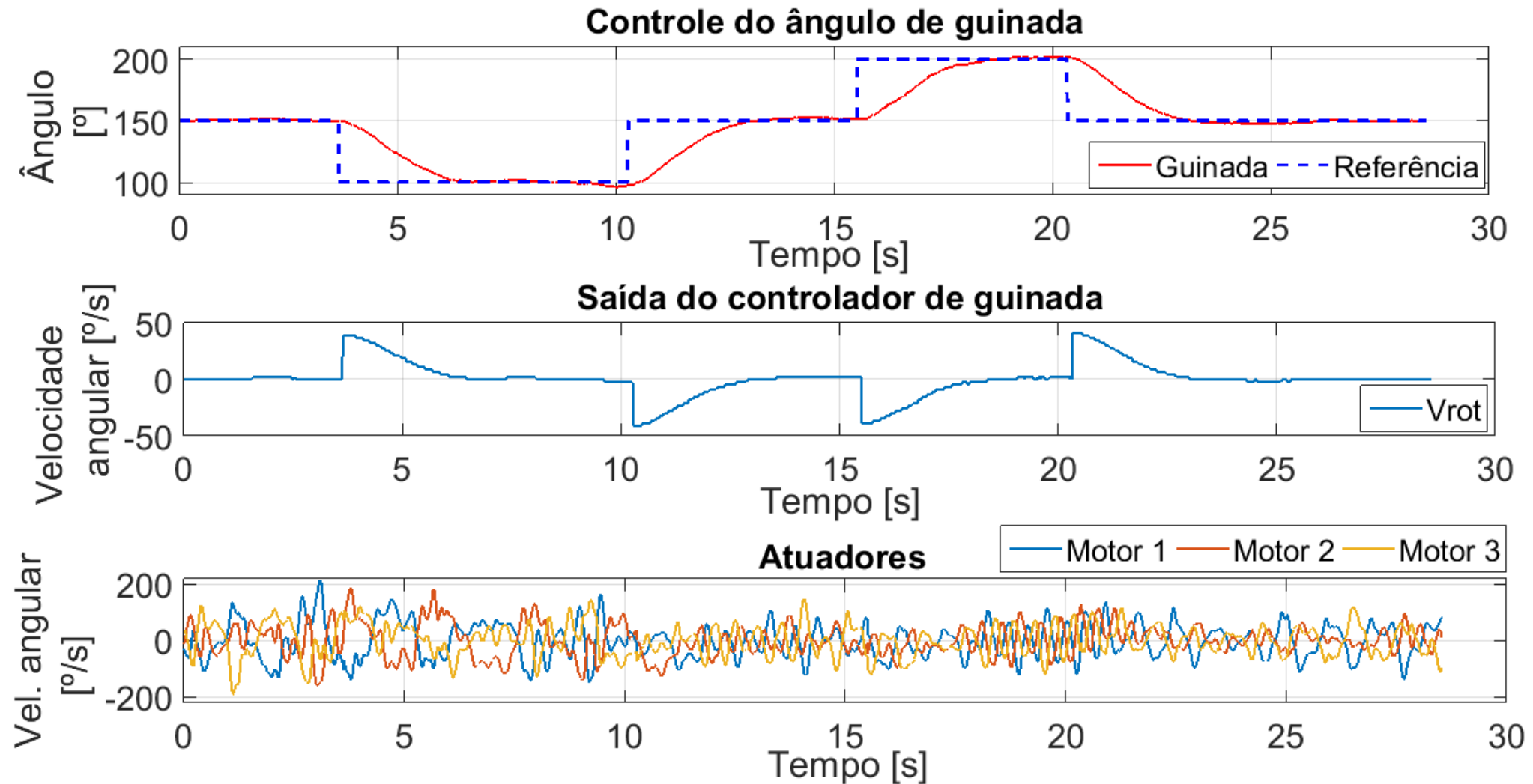




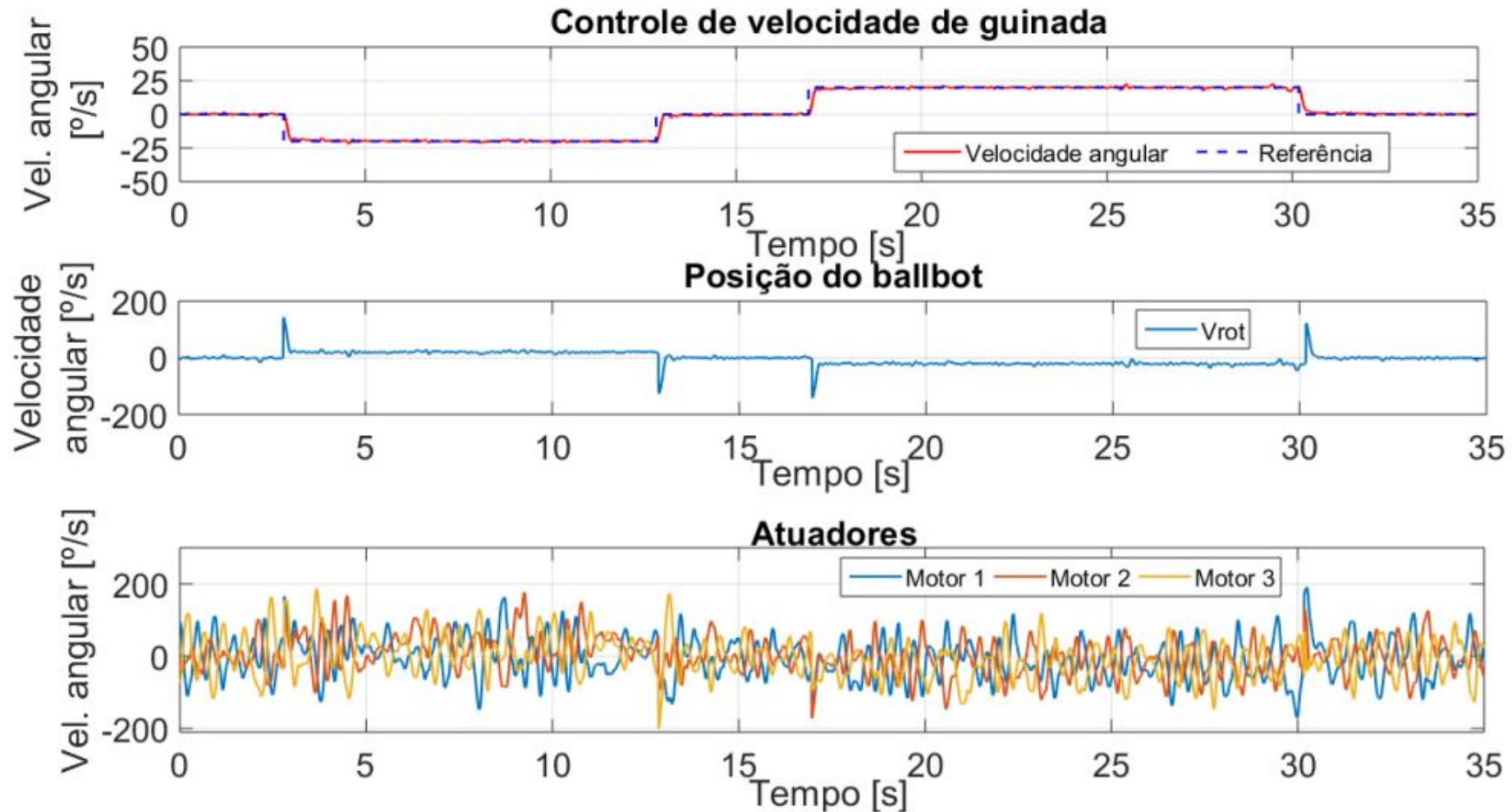
## Software de controle



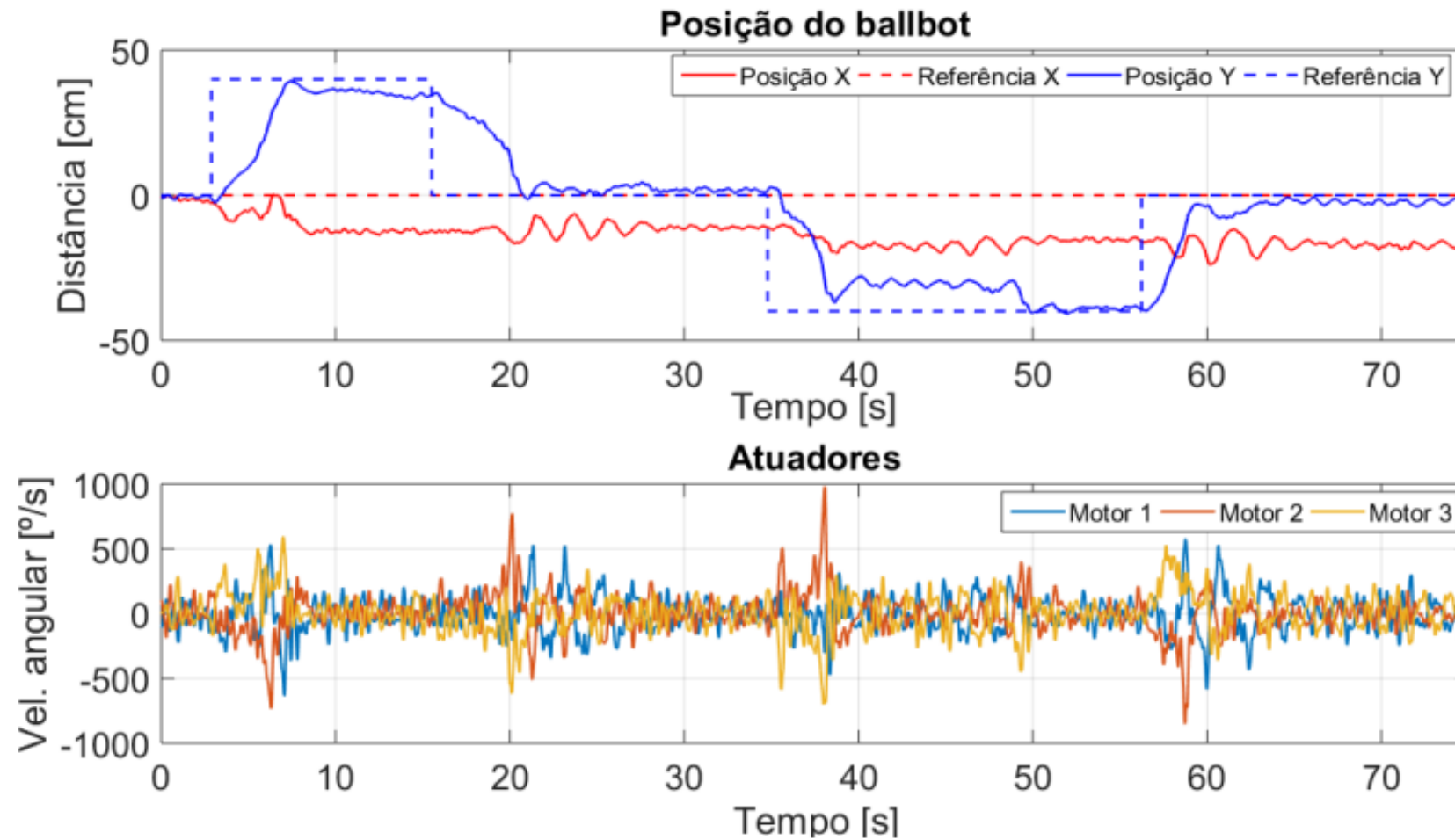
## Controle do ângulo de guinada



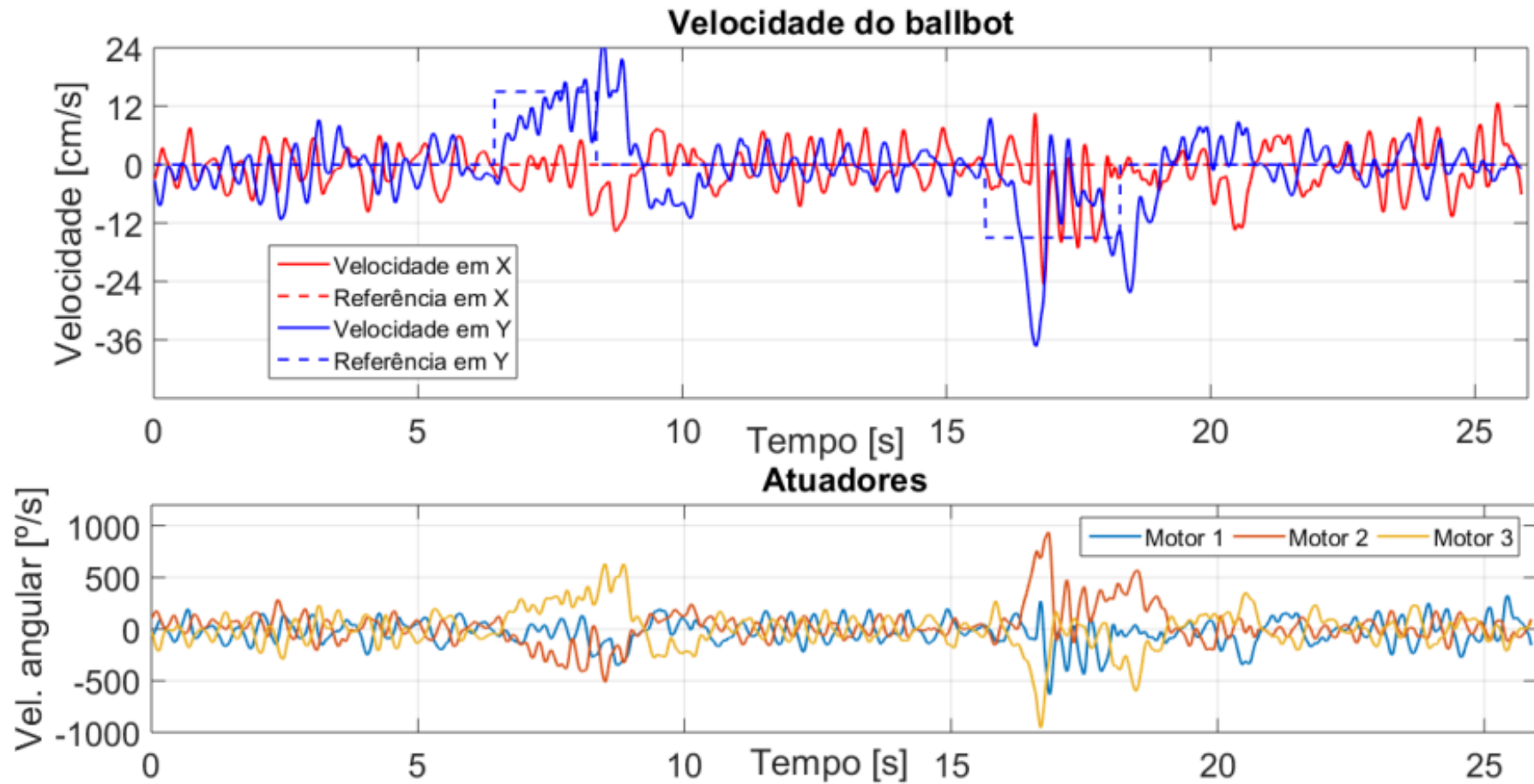
## Controle de velocidade de guinada



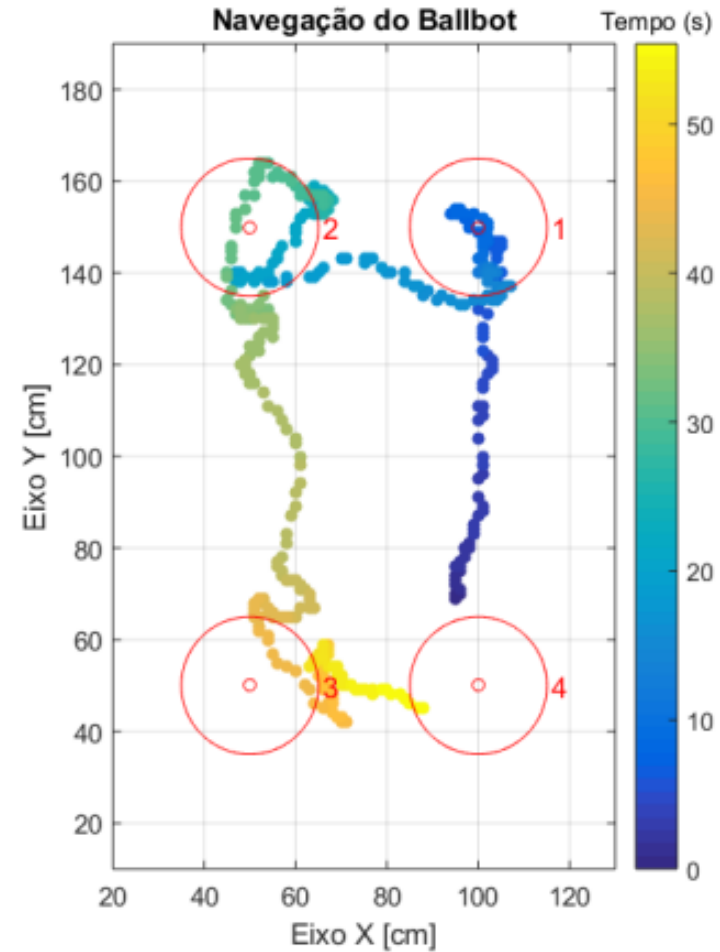
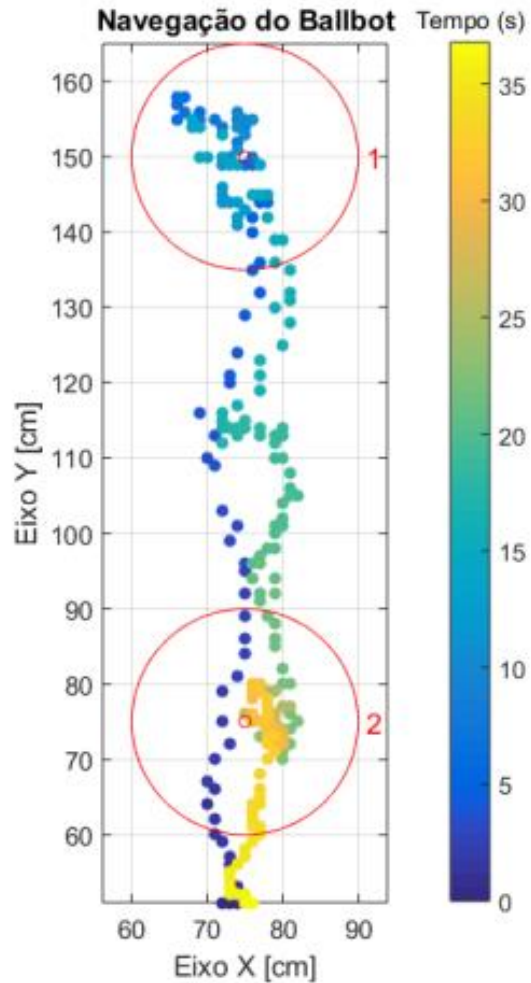
## Controle de posição



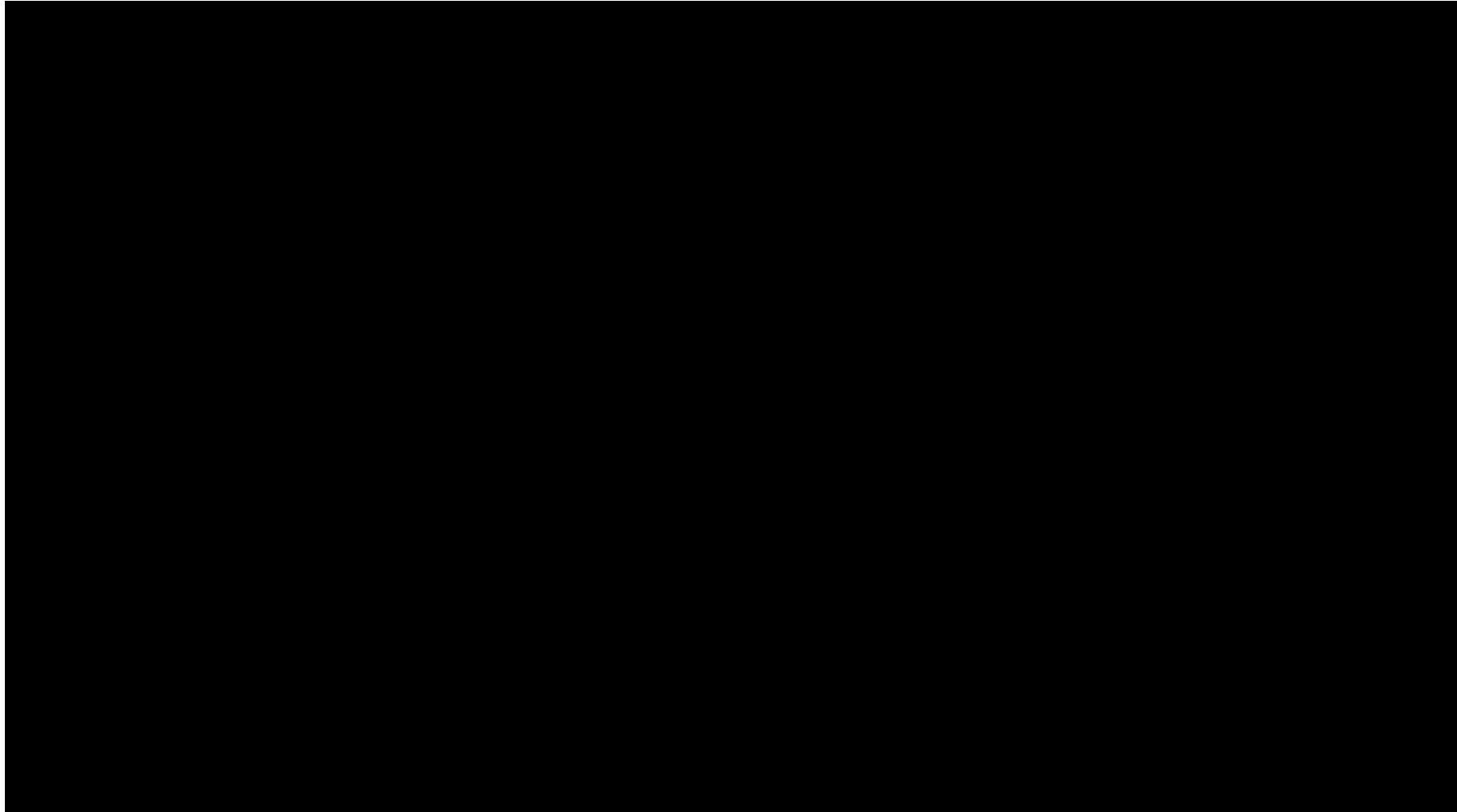
## Controle de velocidade



## Resultado da navegação por *waypoints*



## Vídeo da navegação autônoma por *waypoints*



Experimental



## Rastreamento de objetos

Posição desejada do objeto é sempre no centro da imagem captada pela câmera embarcada.

Uso de filtro de cor para detectar o objeto colorido.

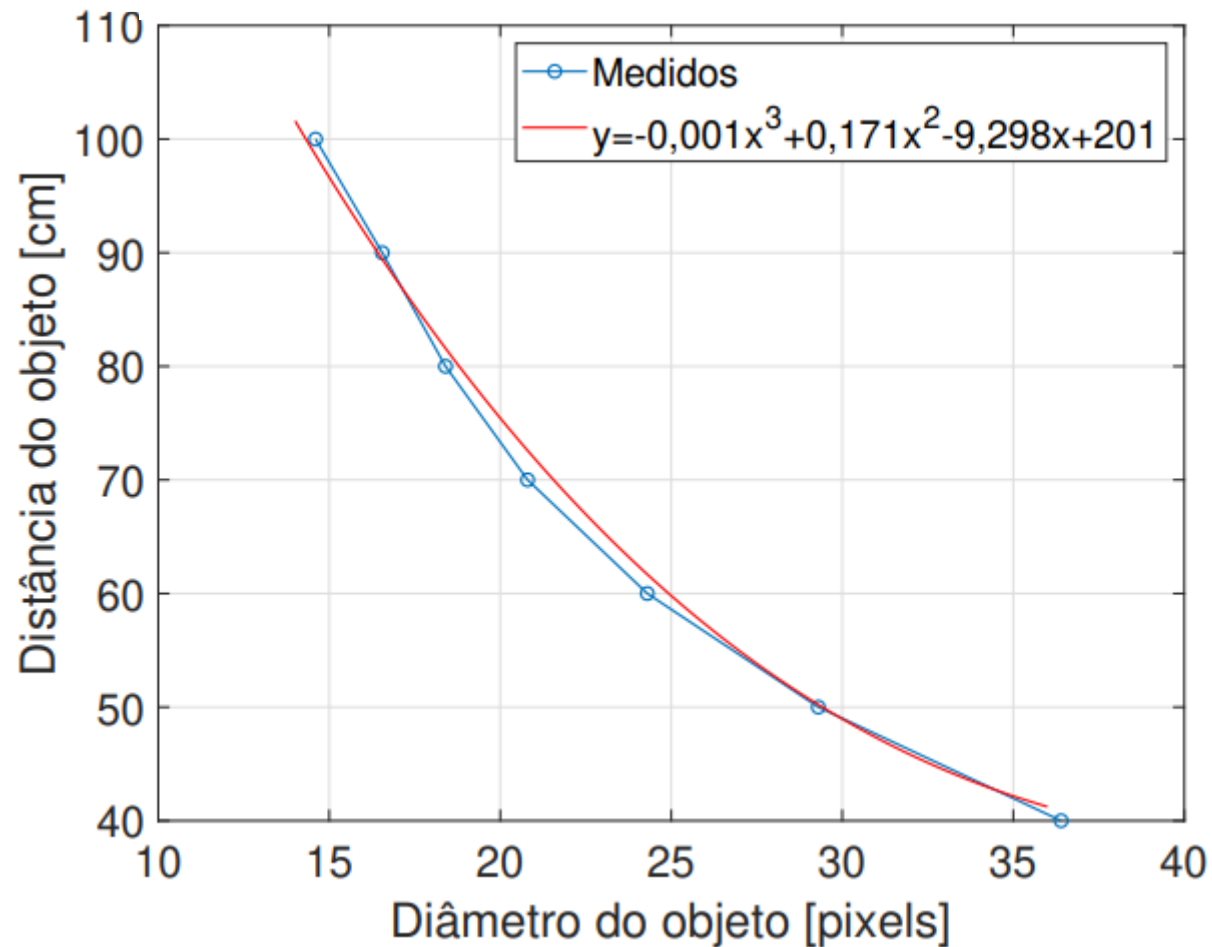
Primeiro o robô se orienta em relação ao objeto e depois executa o movimento de translação.

A estimativa de distância entre o *ballbot* e o objeto colorido foi feita com base no diâmetro do objeto.

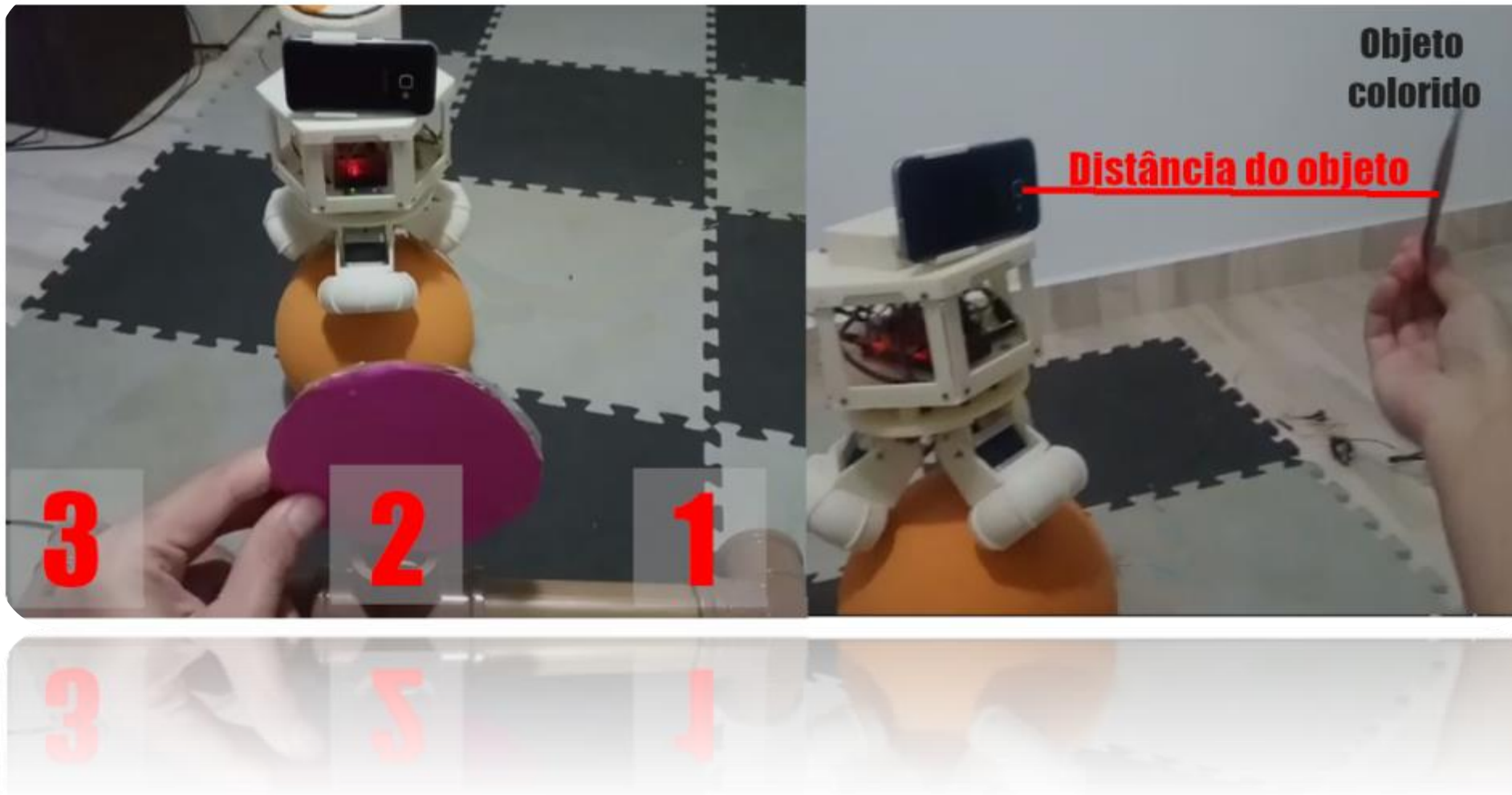




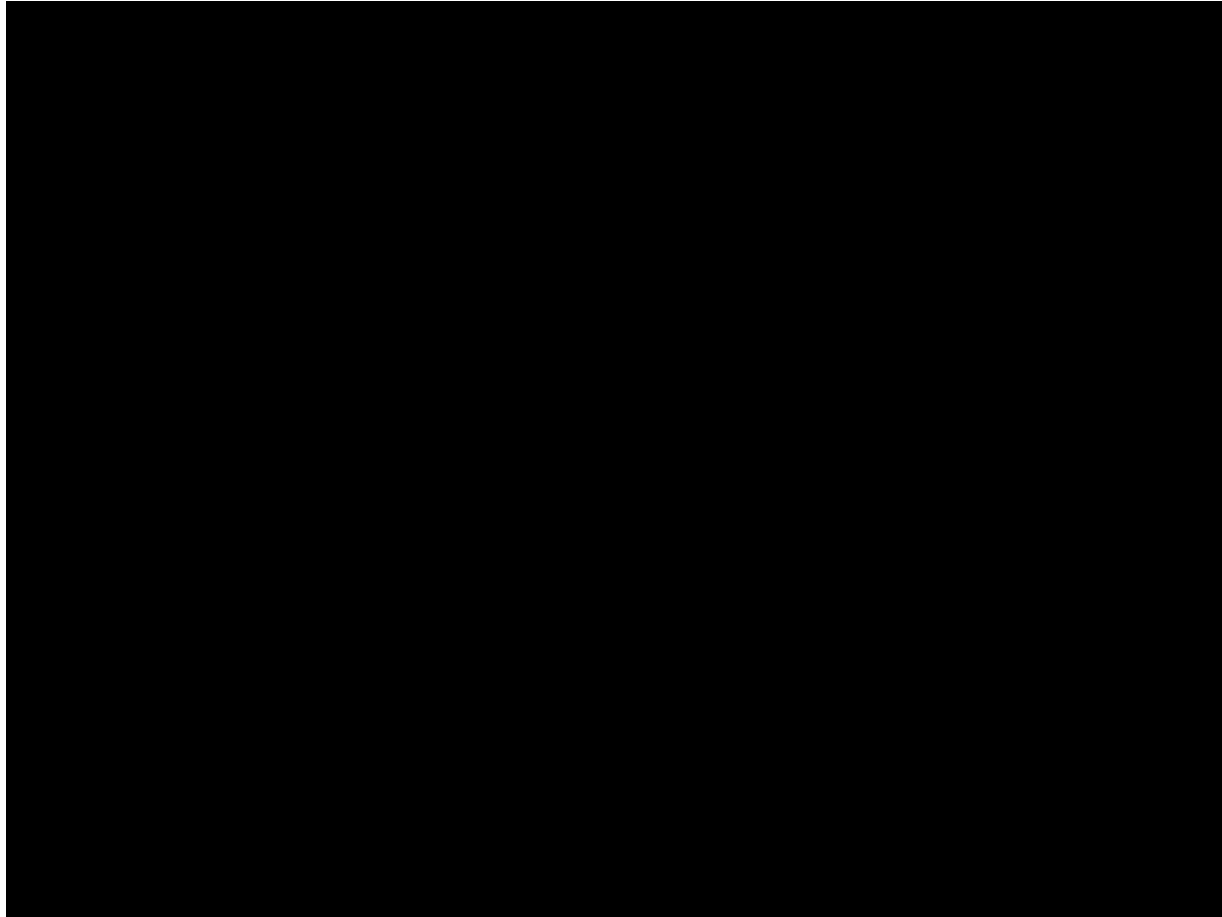
## Estimativa da distância entre o objeto e o robô



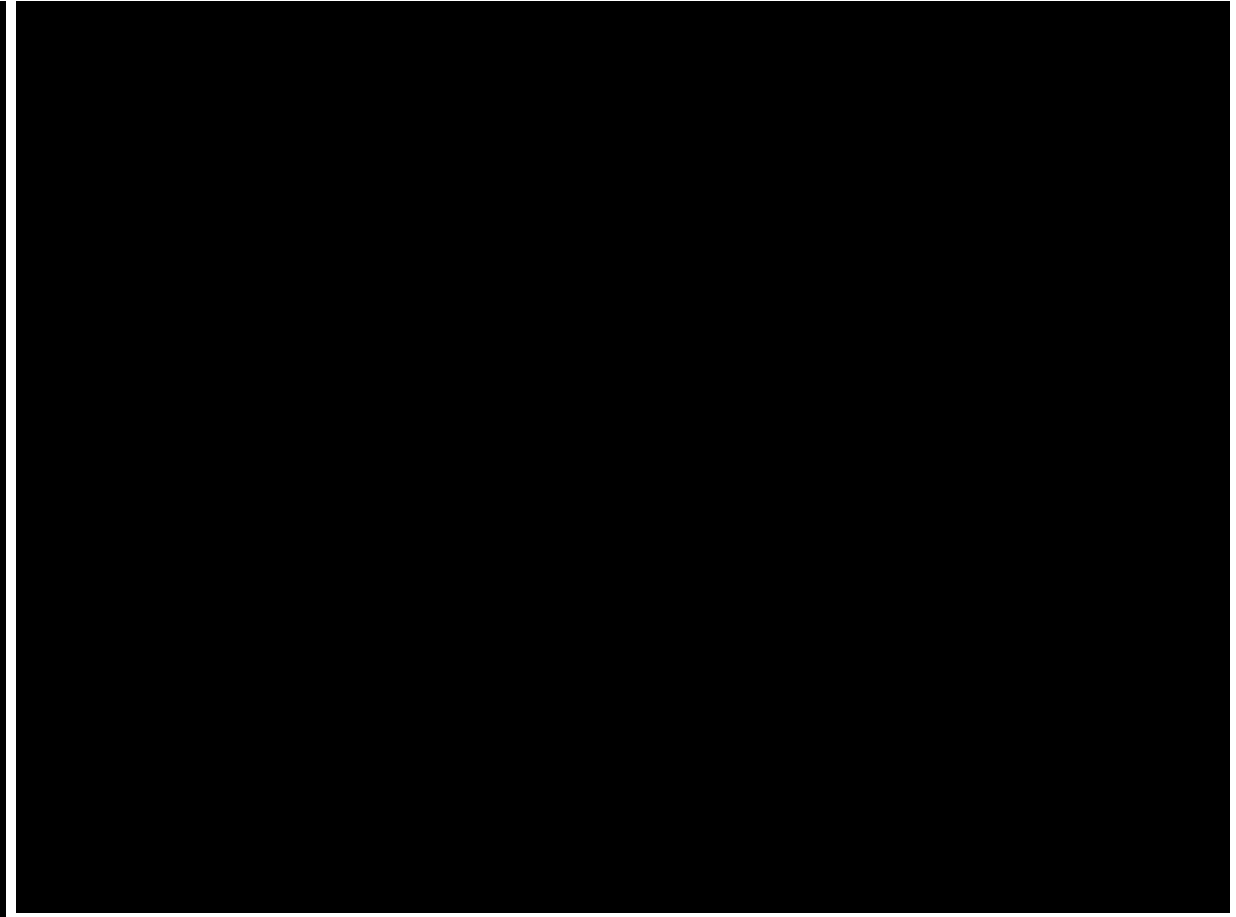
## Experimento de rastreamento de objeto



## Vídeos de rastreamento de objeto

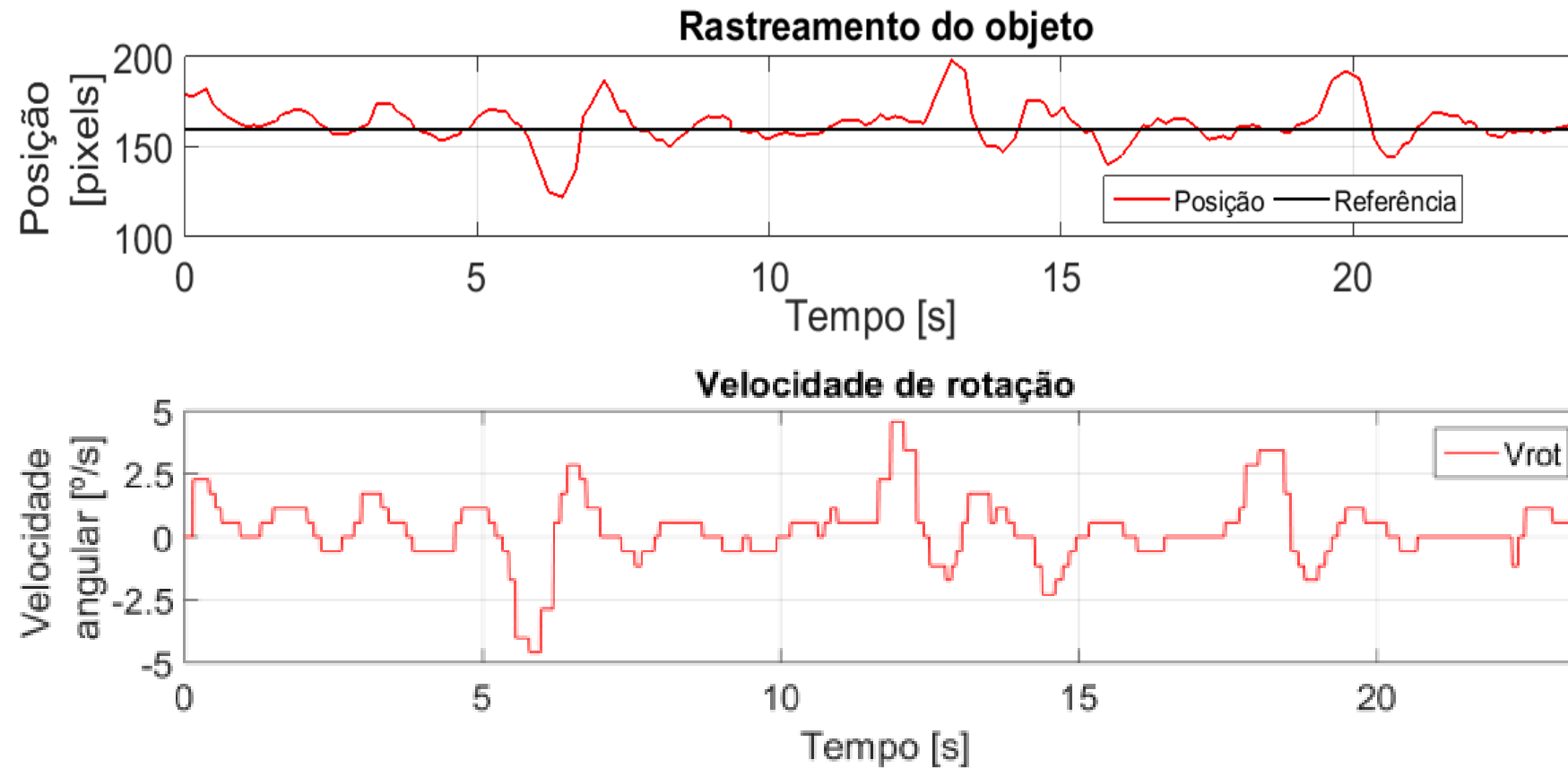


Somente ação de guinada



Com ações de guinada e translação

## Resultado do rastreamento de guinada



## Considerações finais

Apresentou-se neste trabalho um *ballbot* de baixo custo capaz de executar translação e rotação, o que permite que o robô tenha a capacidade de executar uma determinada missão por meio da navegação por *waypoints* e também rastrear objetos.

Principais dificuldades:

- Construção de rodas omnidirecionais de boa qualidade;
- Ruídos causados pela vibração dos motores;
- Limitação da área útil de navegação;
- Baixo poder computacional do sistema embarcado.

## Trabalhos futuros

Alguns pontos que podem ser explorados em trabalhos futuros:

- Construção de outros *ballbots*, para tarefas colaborativas;
- Uso de *ballbot* com SLAM (*Simultaneous Localization and Mapping*);
- Embarcar um LIDAR (*Light Detection and Ranging*) ou câmera de profundidade para a detecção de obstáculos;
- Navegação *outdoor* com o uso de GPS;
- Uso de uma ou mais câmeras para aumentar o ambiente de navegação.

Obrigado!



ITA - Instituto Tecnológico de Aeronáutica

