

**Trabalho de Conclusão de Curso - 2021/2**  
Navegação em robôs móveis por Arbitragem e Fusão em Arquiteturas  
Comportamentais

Profa. Orientadora: Dra. Kathya Silvia Collazos Linares  
Acadêmico: Marcelo Gervazoni Carbonera

20 de Setembro de 2021

# Estrutura

- 1. Introdução.
- 2. Referencial Teórico.
- 4. Materiais e Método.
- 5. Desenvolvimento.
- 6. Conclusões e Trabalhos Futuros.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Objetivos

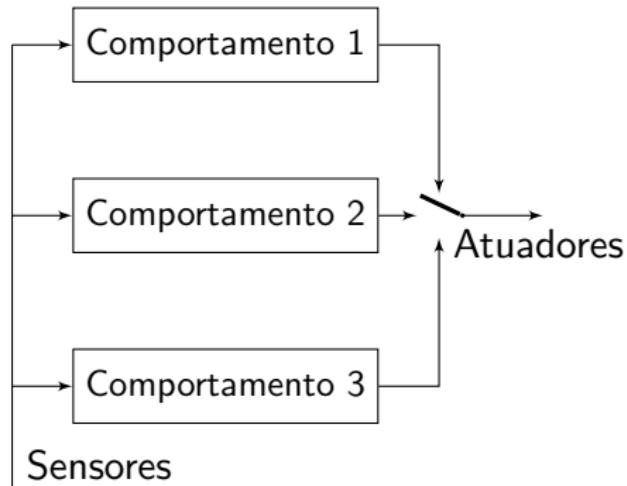
## Objetivo Geral

- Construir robô autônomo, sem armazenamento de mapa, capaz de desviar de obstáculos.
- Comparar qualitativamente: arbitragem e fusão de comportamentos.

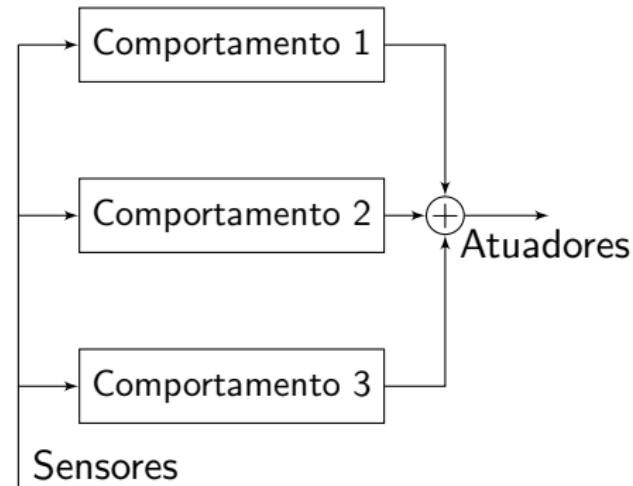
## Objetivos Específicos

- Modelagem.
- Implementar arbitragem de comportamentos usando controlador híbrido.
- Implementar fusão de comportamentos usando controlador *fuzzy*.
- Implementação física.
- Implementação do sistema embarcado.
- Teste e validação.
- Comparar arbitragem e fusão de comportamentos.

# Arquitetura comportamental

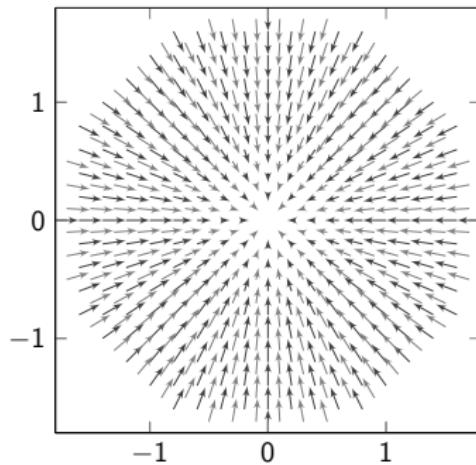


(a) Arbitragem

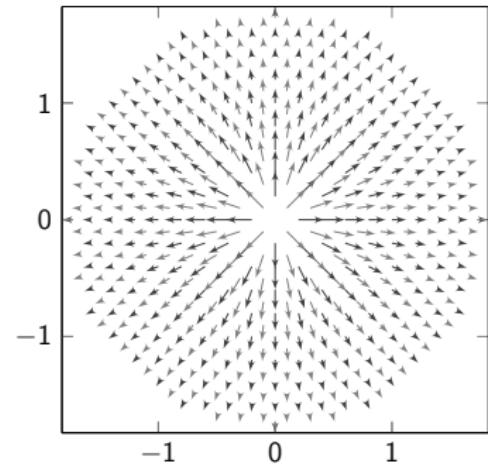


(b) Fusão

# Comportamentos como campos potenciais

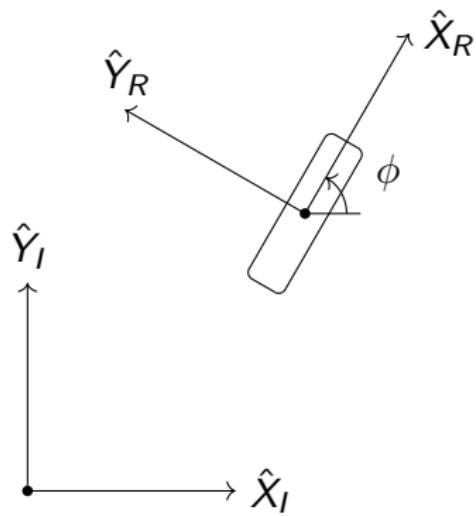


(a) Comportamento “Ir Para Objetivo”

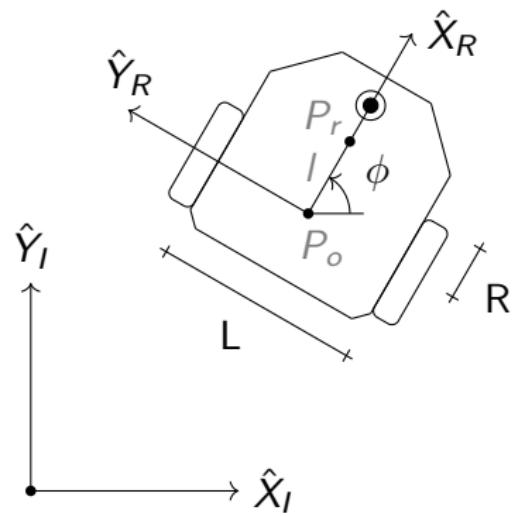


(b) Comportamento “Evitar Obstáculo”

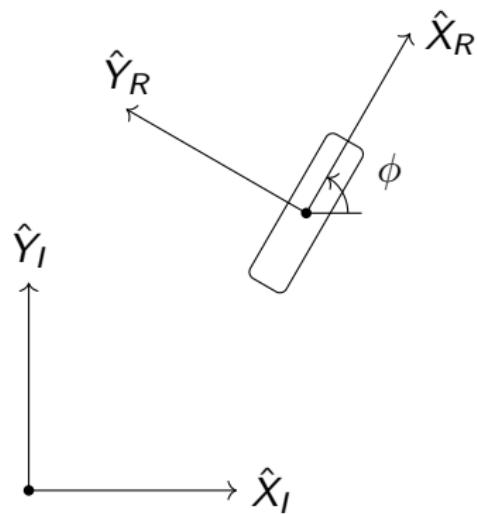
# Modelo Cinemático



(a) Uniciclo

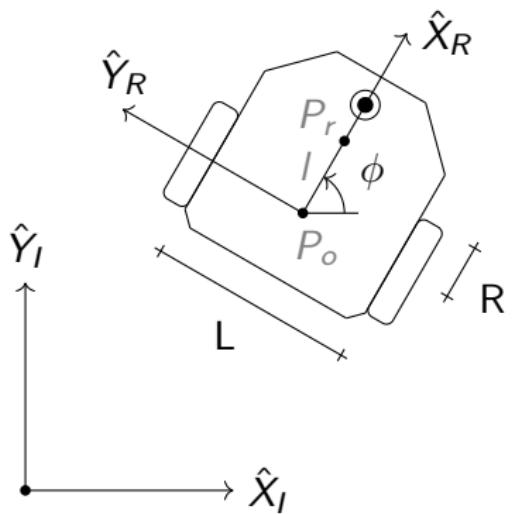


(b) Diferencial



$$\begin{cases} \dot{x} = v \cos \phi \\ \dot{y} = v \sin \phi \\ \dot{\phi} = \omega \end{cases}$$
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$\begin{cases} \dot{x} = \frac{R}{2}(\omega_I + \omega_r) \cos \phi \\ \dot{y} = \frac{R}{2}(\omega_I + \omega_r) \sin \phi \\ \dot{\phi} = \frac{R}{L}(\omega_r - \omega_I) \end{cases}$$



# Estabelecendo vínculo entre os modelos

$$\begin{cases} \dot{x} = \frac{R}{2}(\omega_l + \omega_r) \cos \phi \\ \dot{y} = \frac{R}{2}(\omega_l + \omega_r) \sin \phi \\ \dot{\phi} = \frac{R}{L}(\omega_r - \omega_l) \end{cases}$$

$$\begin{cases} \dot{x} &= v \cos \phi \\ \dot{y} &= v \sin \phi \\ \dot{\phi} &= \omega \end{cases}$$

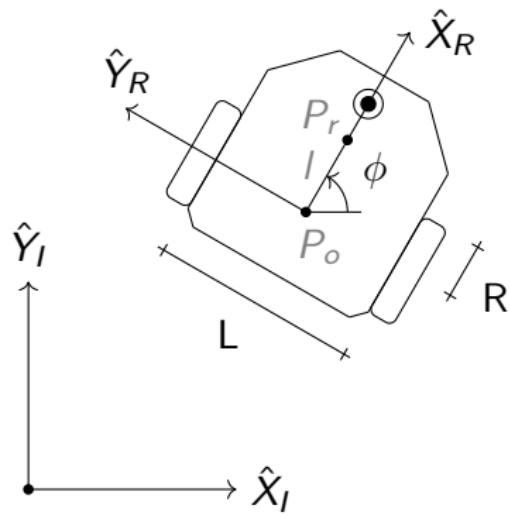
## Equivalência entre modelos

$$\omega_l = \frac{2v - L\omega}{2R}$$

$$\omega_r = \frac{2v + L\omega}{2R}$$

# O modelo de integrador único

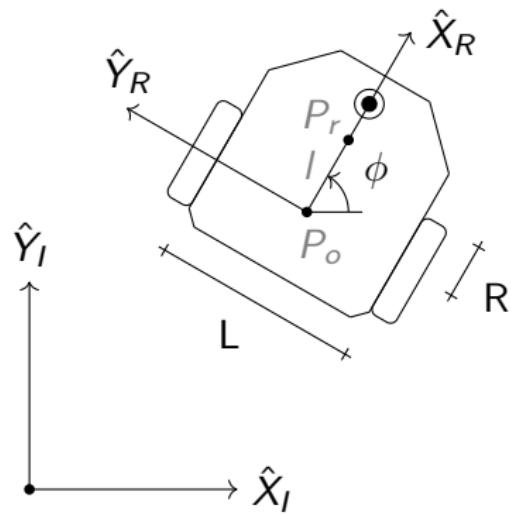
$$\begin{cases} x' = x_o + l \cos \phi \\ y' = y_o + l \sin \phi \end{cases}$$



# O modelo de integrador único

$$\begin{cases} x' = x_o + l \cos \phi \\ y' = y_o + l \sin \phi \end{cases}$$

$$\begin{cases} \dot{x}' = \dot{x}_o - \dot{\phi}l \sin \phi = v \cos \phi - l\omega \sin \phi \\ \dot{y}' = \dot{y}_o + \dot{\phi}l \cos \phi = v \sin \phi + l\omega \cos \phi \end{cases}$$

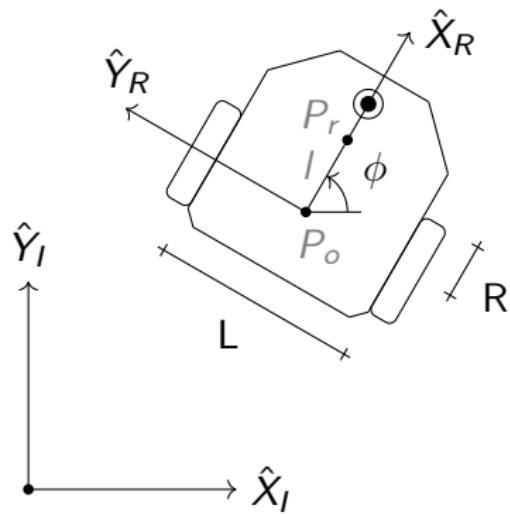


# O modelo de integrador único

$$\begin{cases} x' = x_o + l \cos \phi \\ y' = y_o + l \sin \phi \end{cases}$$

$$\begin{cases} \dot{x}' = \dot{x}_o - \dot{\phi}l \sin \phi = v \cos \phi - l\omega \sin \phi \\ \dot{y}' = \dot{y}_o + \dot{\phi}l \cos \phi = v \sin \phi + l\omega \cos \phi \end{cases}$$

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & l \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$



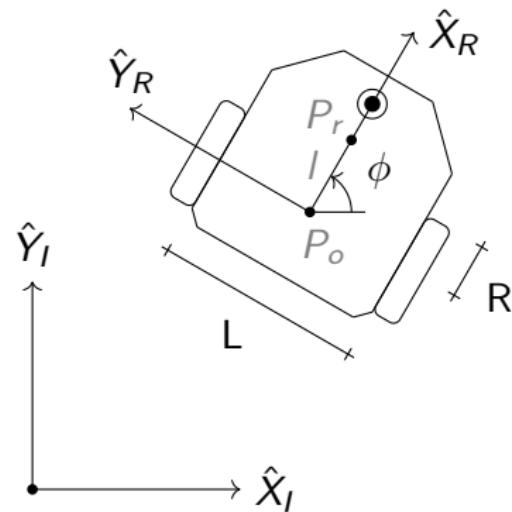
# O modelo de integrador único

$$\begin{cases} x' = x_o + l \cos \phi \\ y' = y_o + l \sin \phi \end{cases}$$

$$\begin{cases} \dot{x}' = \dot{x}_o - \dot{\phi}l \sin \phi = v \cos \phi - l\omega \sin \phi \\ \dot{y}' = \dot{y}_o + \dot{\phi}l \cos \phi = v \sin \phi + l\omega \cos \phi \end{cases}$$

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & l \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix}$$



# O modelo de integrador único

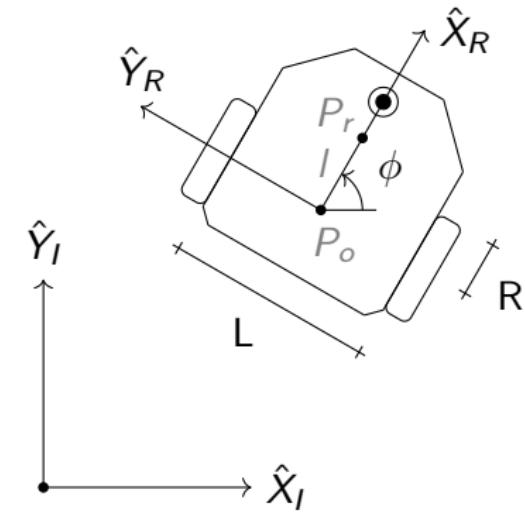
$$\begin{cases} x' = x_o + l \cos \phi \\ y' = y_o + l \sin \phi \end{cases}$$

$$\begin{cases} \dot{x}' = \dot{x}_o - \dot{\phi}l \sin \phi = v \cos \phi - l\omega \sin \phi \\ \dot{y}' = \dot{y}_o + \dot{\phi}l \cos \phi = v \sin \phi + l\omega \cos \phi \end{cases}$$

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & l \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix}$$

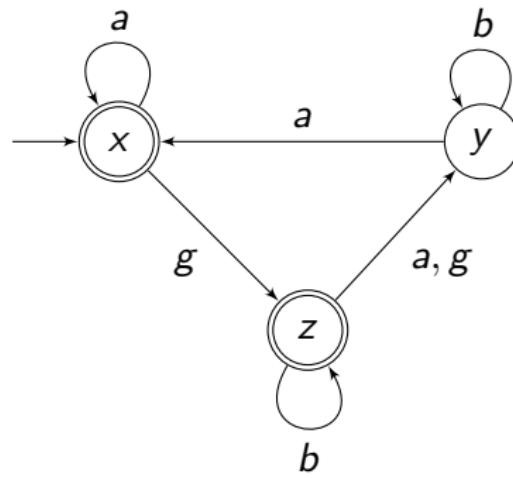
$$\begin{bmatrix} \omega_I \\ \omega_r \end{bmatrix} = \begin{bmatrix} \frac{1}{R} & \frac{-L}{2R} \\ \frac{1}{R} & \frac{L}{2R} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \dot{x}' \\ \dot{y}' \end{bmatrix}$$



# Sistemas a Eventos Discretos

## Definição

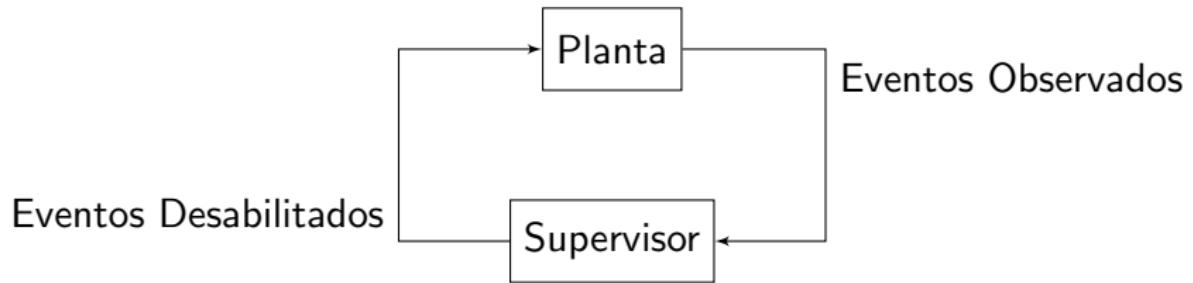
- Sistema Dinâmico.
- Estados discretos.
- Eventos assíncronos provocam transições instantâneas de estado.
- Arbitragem!



# Controle Supervisório

## Definição

- Captura eventos observáveis.
- Permite ou suprime eventos passíveis de ocorrência.



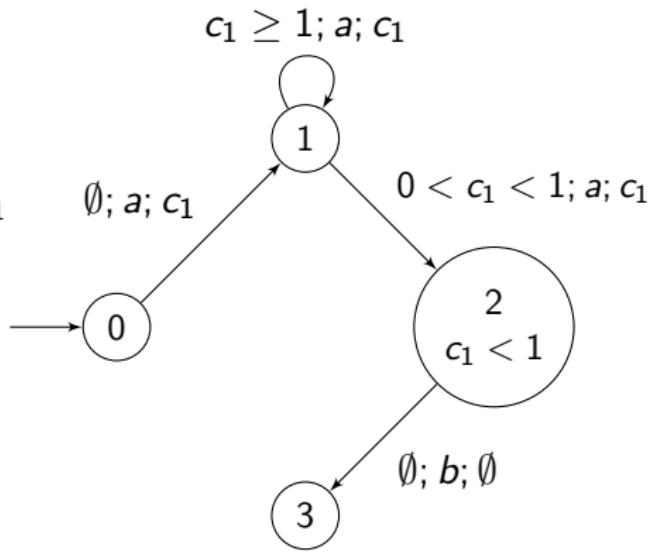
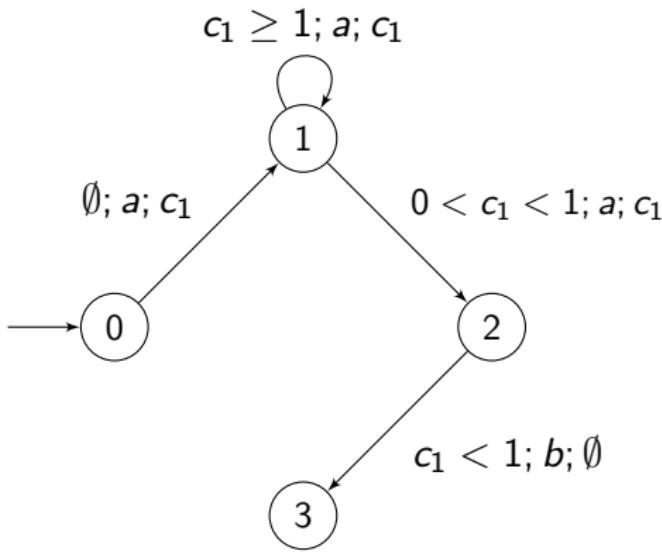
# Automatos Temporizados com guarda

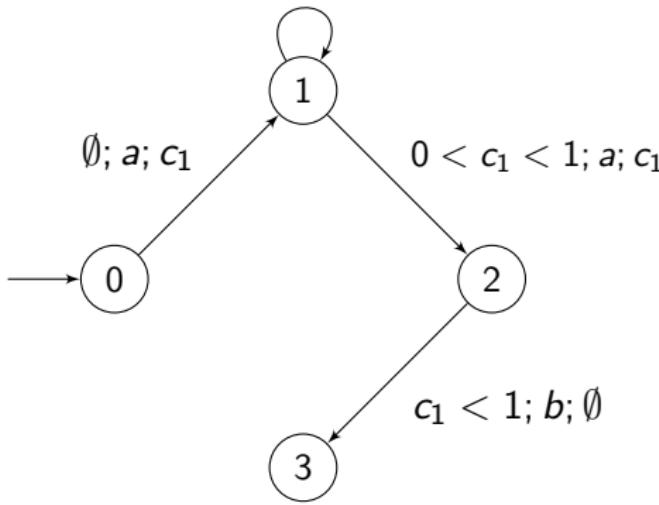
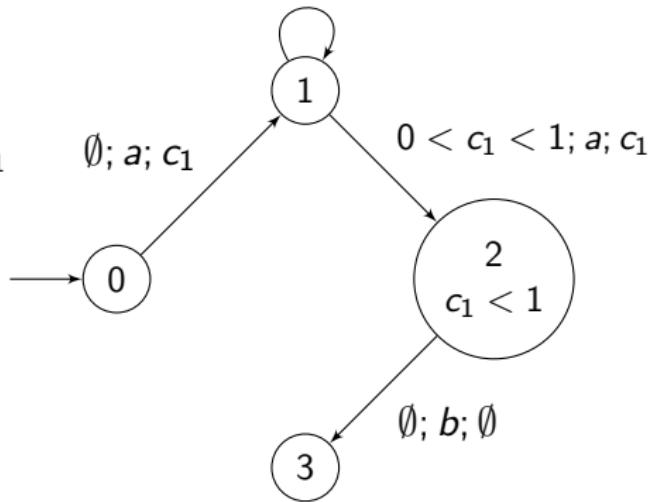
## clocks

- “Contador”: associado a tempo.
- Definido por uma função linear ( $\dot{x} = 1$ ).

## Função de transição

- Transições: (Estado, Evento) → (Estado)
- Transições Temporizadas: (Guardas, Evento, *Reset*) → (Estado)
  - Guardas: condições nas variáveis contínuas.
  - *Reset*: conjunto de variáveis *clock* às quais deve-se atribuir valor zero.



$c_1 \geq 1; a; c_1$  $c_1 \geq 1; a; c_1$ 

## Invariante

Evita *deadlock* por tempo ao forçar transição!

# Sistemas Híbridos

## Definição

- Estados formados por tuplas  $(q, x)$ .
  - $q$ : Estados discretos .
  - $x$ : Estados contínuos .
- Autômato híbrido extende o autômato temporizado com guardas.
- Estados são chamados “**modos de operação**”, pois alteram a dinâmica do sistema.

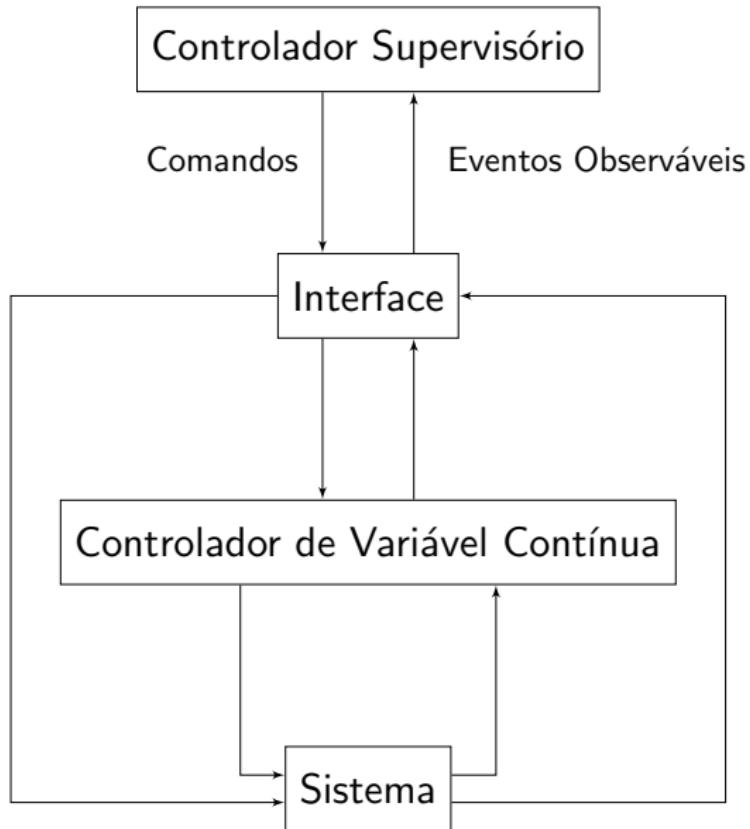
# Sistemas Híbridos

## Definição

- Estados formados por tuplas  $(q, x)$ .
  - $q$ : Estados discretos .
  - $x$ : Estados contínuos .
- Autômato híbrido extende o autômato temporizado com guardas.
- Estados são chamados “**modos de operação**”, pois alteram a dinâmica do sistema.

## Arbitragem

Chaveamento de dinâmicas (controladores).

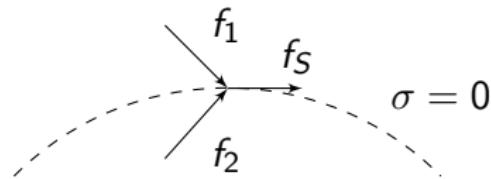


## Campos vetoriais descontínuos

- Regiões de costura.
- Deslizamento.
- Comportamento de Zenão.
- Regularização.

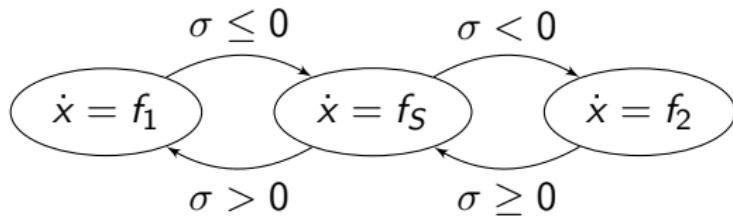
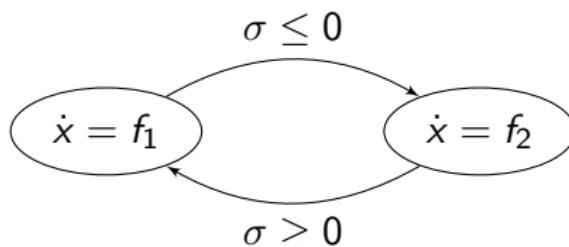
## Campos vetoriais descontínuos

- Regiões de costura.
- Deslizamento.
- Comportamento de Zenão.
- Regularização.

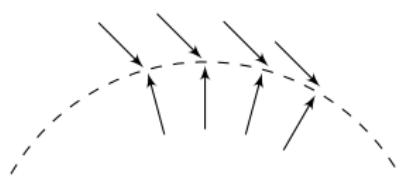


## Campos vetoriais descontínuos

- Regiões de costura.
- Deslizamento.
- Comportamento de Zenão.
- Regularização.



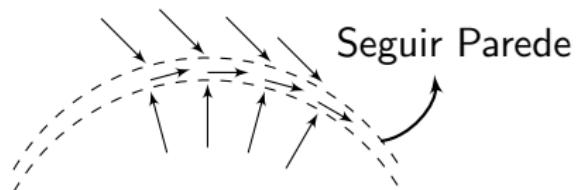
Ir para Objetivo



Evitar Obstáculo



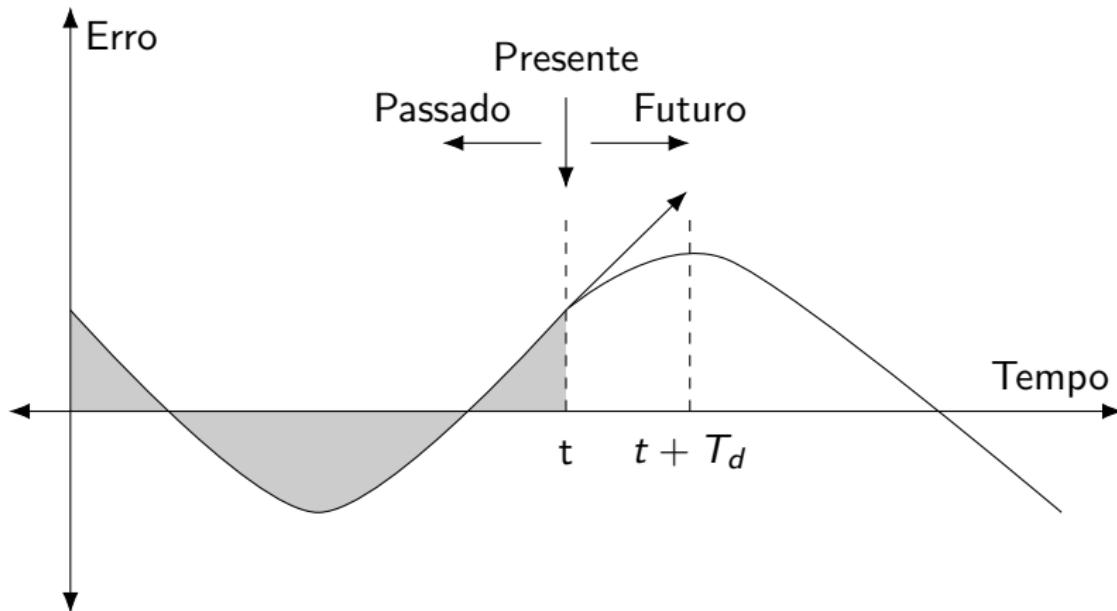
Ir para Objetivo



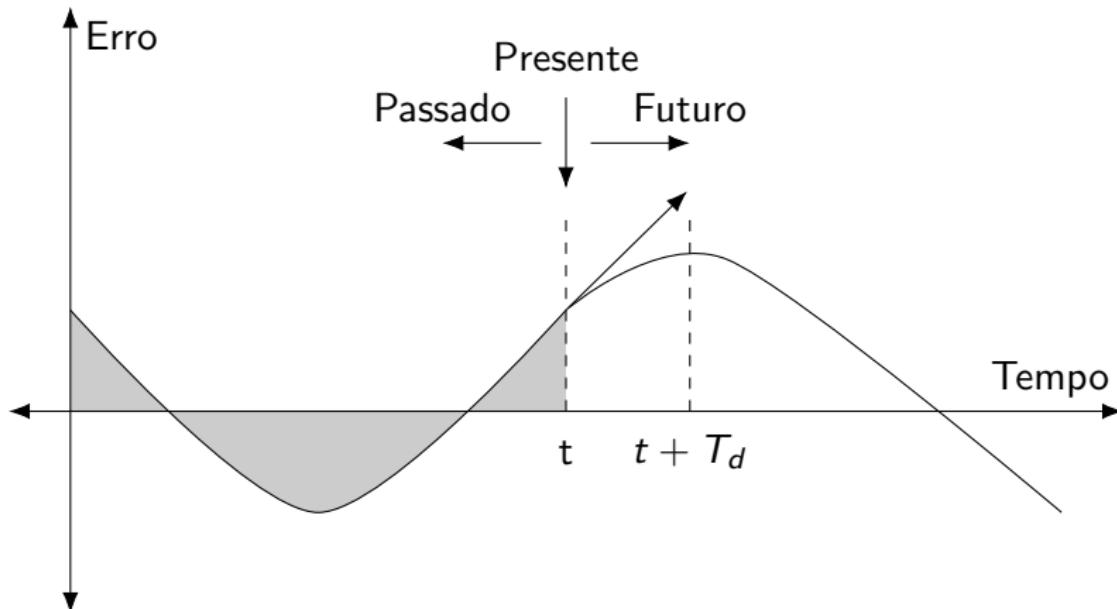
Evitar Obstáculo



# Controlador PID



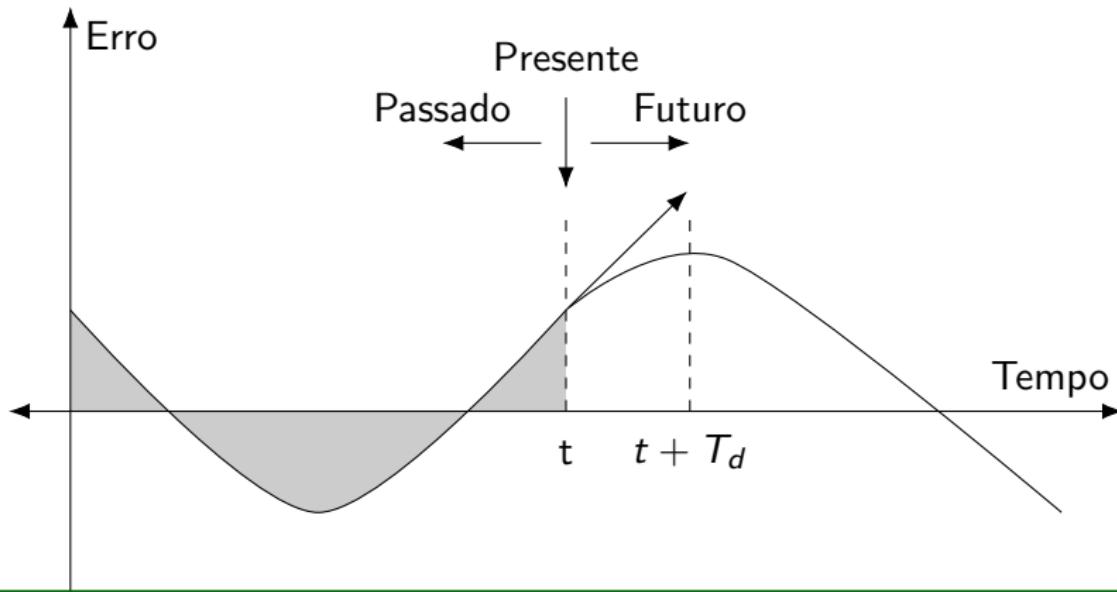
# Controlador PID



PID tempo contínuo

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \quad (1)$$

# Controlador PID



PID tempo discreto

$$u(k) = k_p e(k) + k_i T_d \sum_0^k e(k) + k_d \frac{e(k) - e(k-1)}{T_d} \quad (2)$$

# Lógica Fuzzy em controle

## Conjuntos fuzzy

- Variável linguística.
- Valor linguístico.
- Grau de pertinência.

# Lógica Fuzzy em controle

## Conjuntos fuzzy

- Variável linguística.
- Valor linguístico.
- Grau de pertinência.

## Motivação para uso

- Mescla aspectos exatos e elementos qualitativos.

# Lógica Fuzzy em controle

## Conjuntos fuzzy

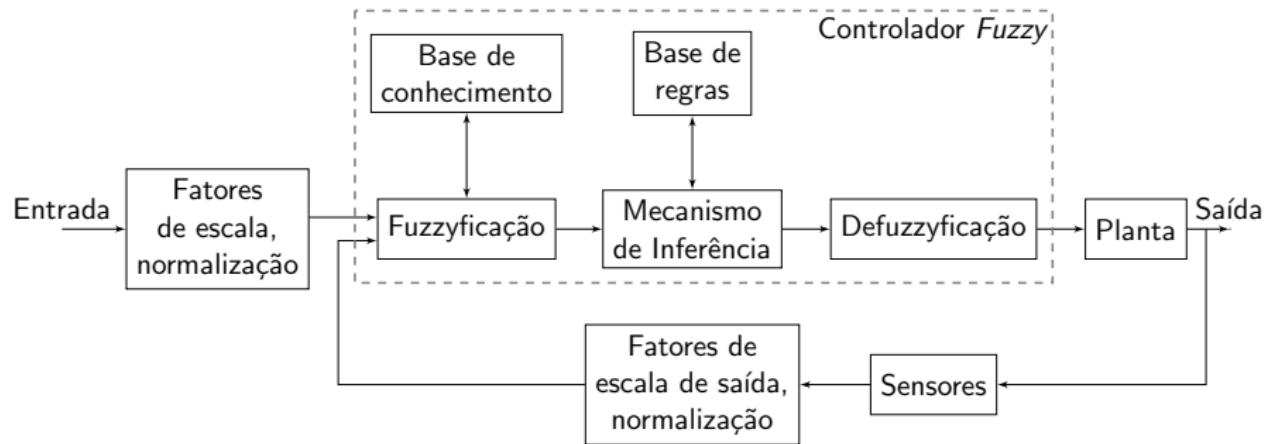
- Variável linguística.
- Valor linguístico.
- Grau de pertinência.

## Motivação para uso

- Mescla aspectos exatos e elementos qualitativos.

## Indicação e contraindicação

- Indicada em sistemas **complexos**, compreendidos parcialmente.
- Deseja-se solução rápida.
- Não apropriada para sistemas simples (LIT).



## Fuzzyficação

- Associa variáveis numéricas a conjuntos fuzzy.
- Saída: conjunto de graus de pertinência vinculados a valores linguísticos.

## Fuzzyficação

- Associa variáveis numéricas a conjuntos fuzzy.
- Saída: conjunto de graus de pertinência vinculados a valores linguísticos.

## Mecanismo de Inferência

- Base de regras ( $P \rightarrow Q$ ).
- Pertinência das premissas determinam pertinências das saídas.
- Conjunto de “recomendações”.

## Fuzzyficação

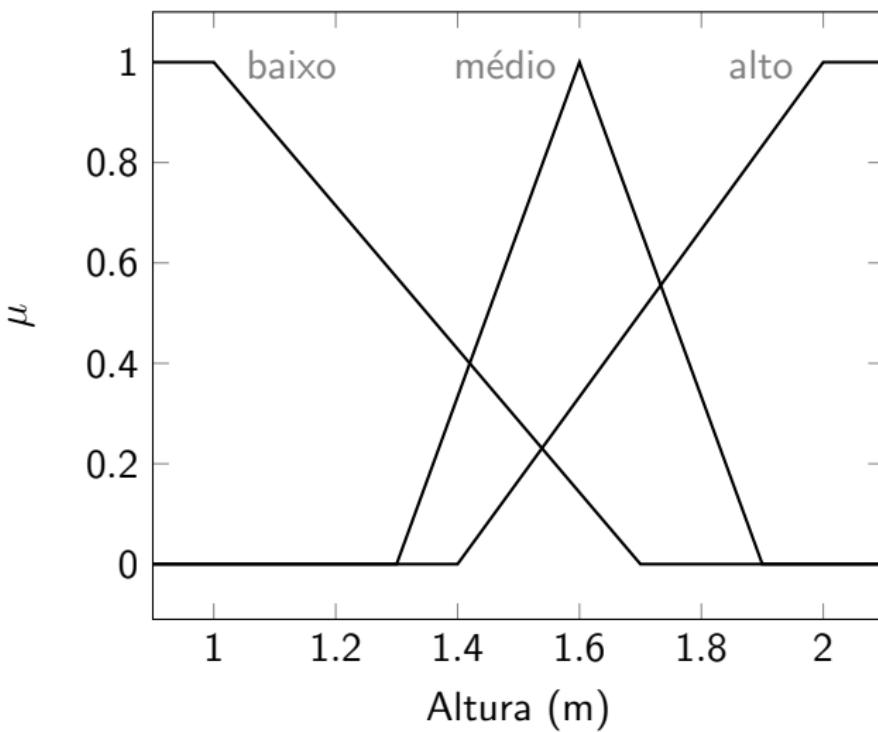
- Associa variáveis numéricas a conjuntos fuzzy.
- Saída: conjunto de graus de pertinência vinculados a valores linguísticos.

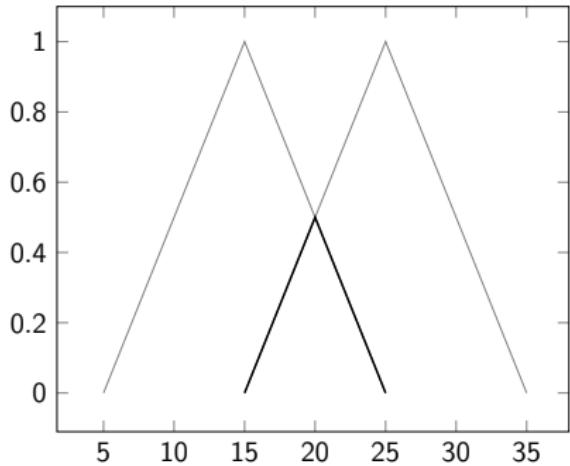
## Mecanismo de Inferência

- Base de regras ( $P \rightarrow Q$ ).
- Pertinência das premissas determinam pertinências das saídas.
- Conjunto de “recomendações”.

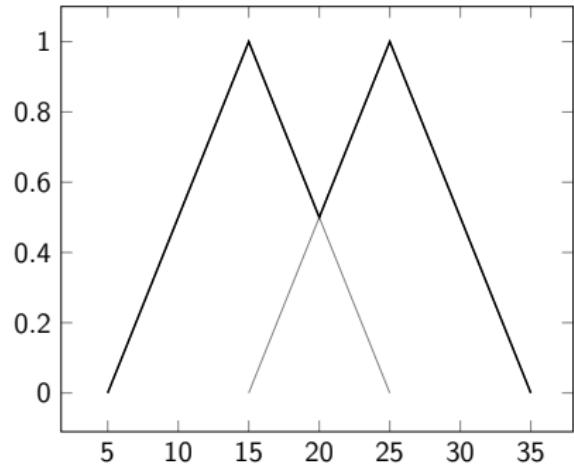
## Defuzzificação

- A partir de recomendações distintas, determinam-se as saídas.

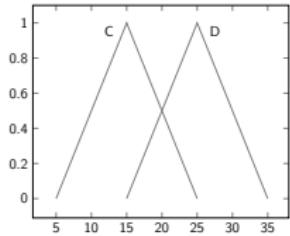




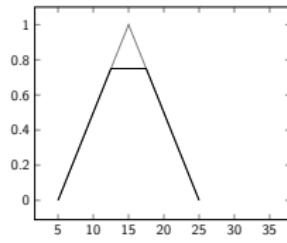
(a) Intersecção usando função “min”



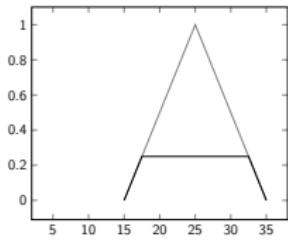
(b) União usando função “max”



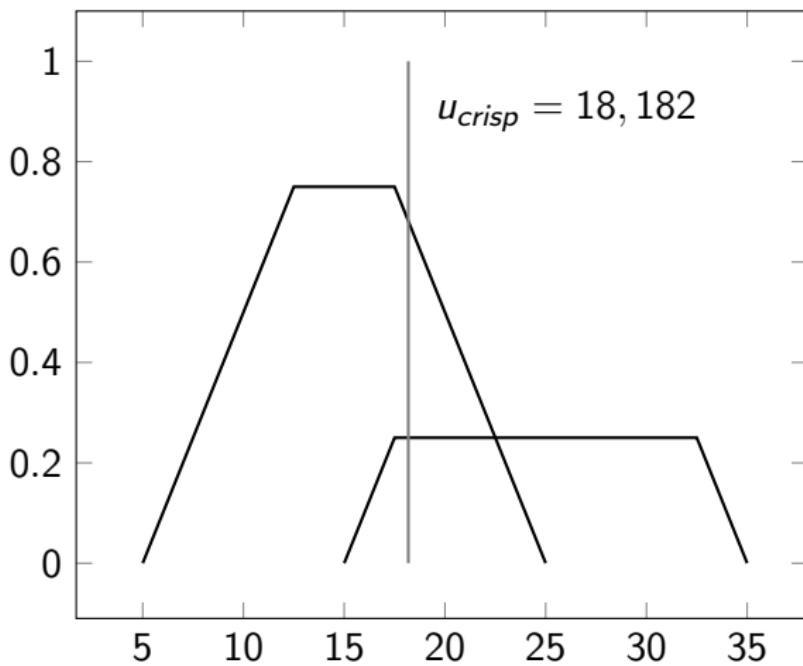
(a) Funções de pertinência para C e D



(b) Função de pertinência para regra  $A \rightarrow C$ , com  $\mu_A = 0,75$



(c) Função de pertinência para regra  $B \rightarrow D$ , com  $\mu_B = 0,25$



## Defuzzificação COG

$$u_{crisp} = \frac{\sum_i b_i \int \mu_i}{\sum_i \int \mu_i} \quad (3)$$

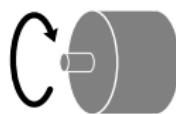
- $b_i$  é o centro da função de pertinência.
- $\int \mu_i$  para funções triangulares vale  $b(h - \frac{h^2}{2})$

## Defuzzificação COG

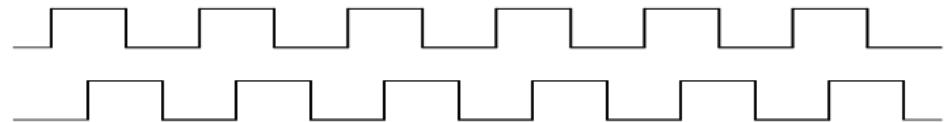
$$u_{crisp} = \frac{\int \mu^*(x) x dx}{\int \mu^*(x) dx} \quad (4)$$

- $\mu^*$  é calculada pela função max() sobre funções de pertinência parciais.

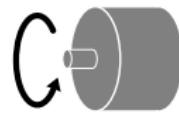
# Odometria



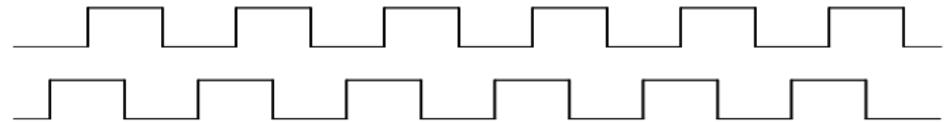
Canal A



Canal B

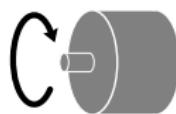


Canal A

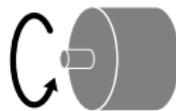
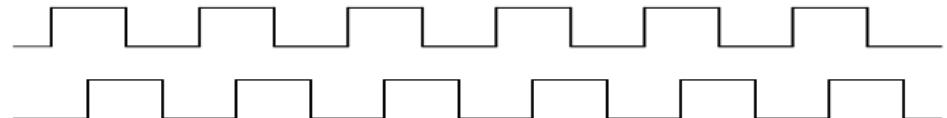


Canal B

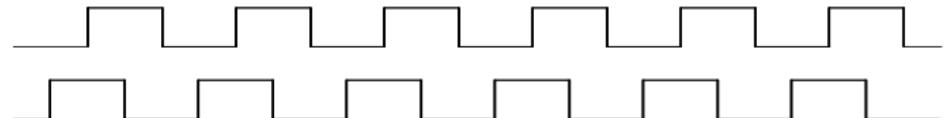
# Odometria



Canal A



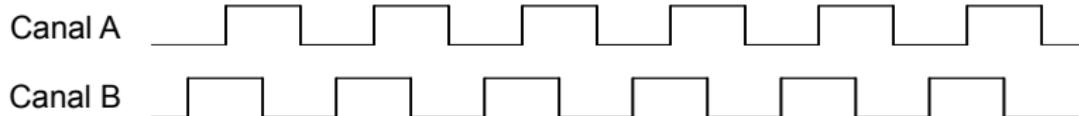
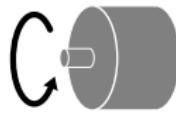
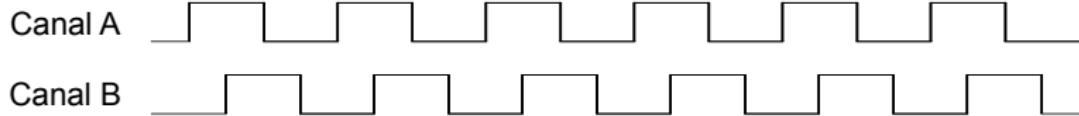
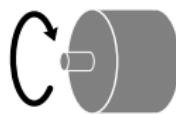
Canal A



## Estimativa de velocidade

$$V_{pneu} = \frac{N_{ticks}}{N_{PPR}} \frac{2\pi R}{T} \quad (5)$$

# Odometria



## Estimativa de velocidade

$$\begin{cases} x' = x + \frac{d_{esq} + d_{dir}}{2} \cos \phi \\ y' = y + \frac{d_{esq} + d_{dir}}{2} \sin \phi \\ \phi' = \phi + \frac{d_{dir} - d_{esq}}{L} \end{cases} \quad (6)$$

# Materiais

## Tabela de Custos

Item	Qty	Custo Unitário	Custo Total
Bateria LiPo Limskey 11,1v 3S 4200mAh	1	112,05	112,05
Ponte H dupla L298N	1	6,12	6,12
Sensor infravermelho GP2Y0A21YK0F, alcance de 10 a 80 cm	5	13,07	65,35
Módulo conversor Buck DC-DC, saída 5v	1	9,14	9,14
Módulo Bluetooth XM-15	1	25,00	25,00
Conjuntos de motores, caixa de redução 1:34 e sensores de efeito hall (341,2 PPR)	2	47,015	94,03
Esfera de rolagem	1	2,002	2,002
Tiva Connected LaunchPad TM4C1294	1	80,00	80,00
	Total:		393,69

# Metodologia

- Projetar controladores.
- Validar o funcionamento usando simulador Simiam.
- Montagem física.
- Sistema embarcado.
- Teste e validação.

# Simulação

## Simiam

- Simulador implementado no Matlab.
- Implementado como sistema híbrido (Supervisor escalona controladores).

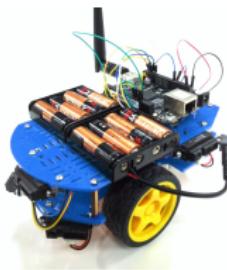
# Simulação

## Simiam

- Simulador implementado no Matlab.
- Implementado como sistema híbrido (Supervisor escalona controladores).



(a) Robô Khepera 3

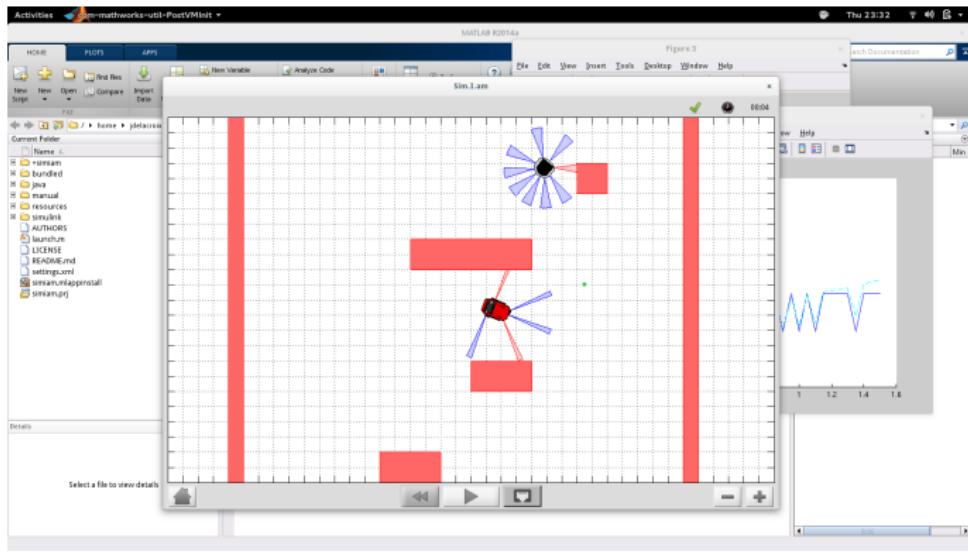


(b) Robô QuickBot

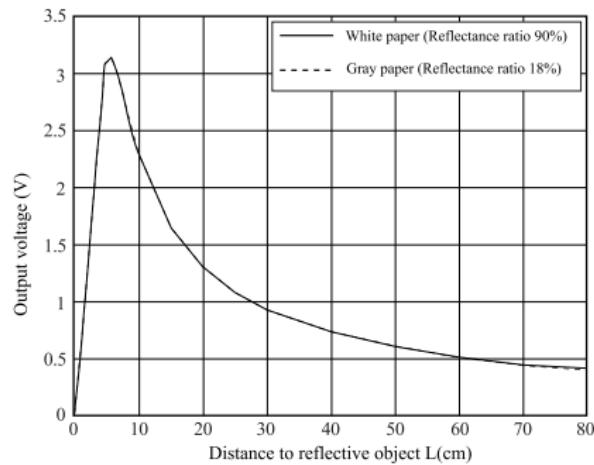
# Simulação

## Simiam

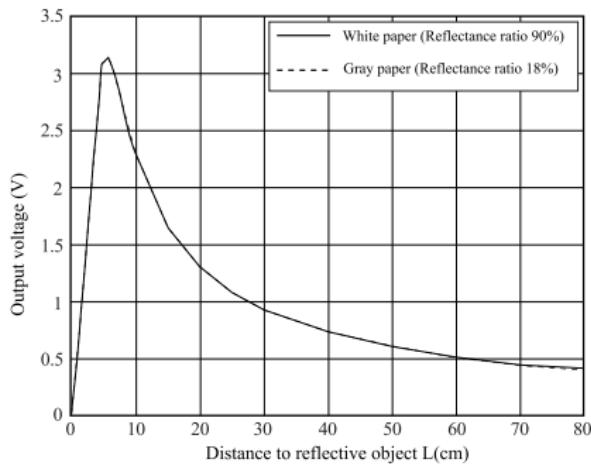
- Simulador implementado no Matlab.
- Implementado como sistema híbrido (Supervisor escalona controladores).



# Especificações



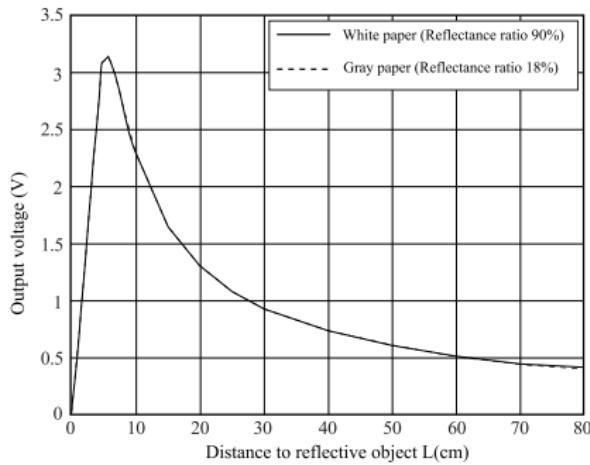
# Especificações



## Parâmetros

- Sensor IR: intervalo entre 10 e 80cm.
- $L = 18\text{cm}$ .
- $R = 3.4\text{cm}$ .

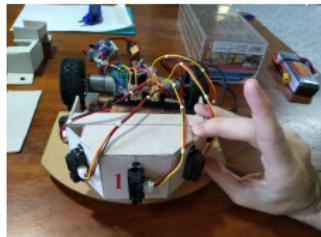
# Especificações



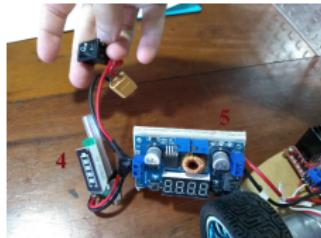
## Equação para distâncias do sensor IR

$$\begin{aligned} d(v) = & 2.7802212625v^6 - 35.1150300110v^5 + 179.6031433005v^4 \\ & - 477.9449116299v^3 + 706.3400747125v^2 - 569.7367375002v \quad (7) \\ & + 221.2678651473 \end{aligned}$$

# Montagem Física



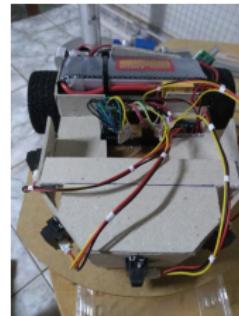
(a) Posicionamento dos sensores IR



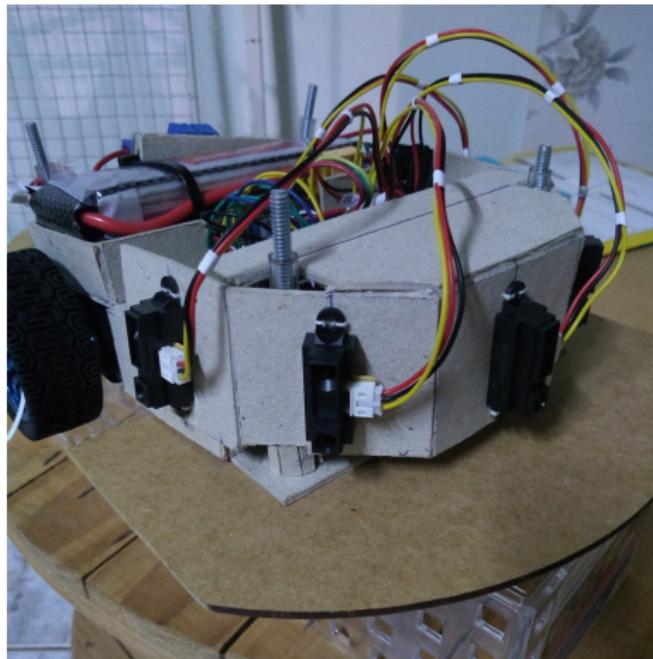
(c) Regulador de Tensão



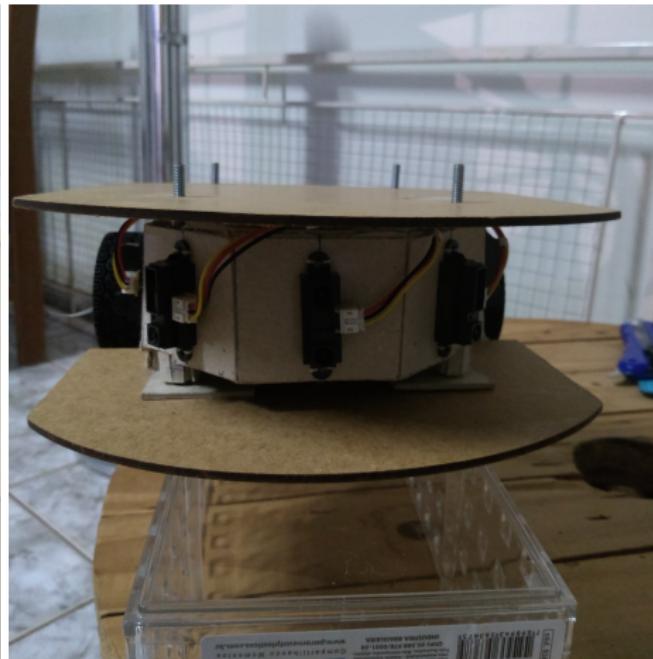
(b) Disposição dos motores e microcontrolador



(d) Posicionamento dos componentes



(a) Posição dos parafusos de rosca



(b) Robô montado

# Curva para motores

Figure: Curva do motor sem carga

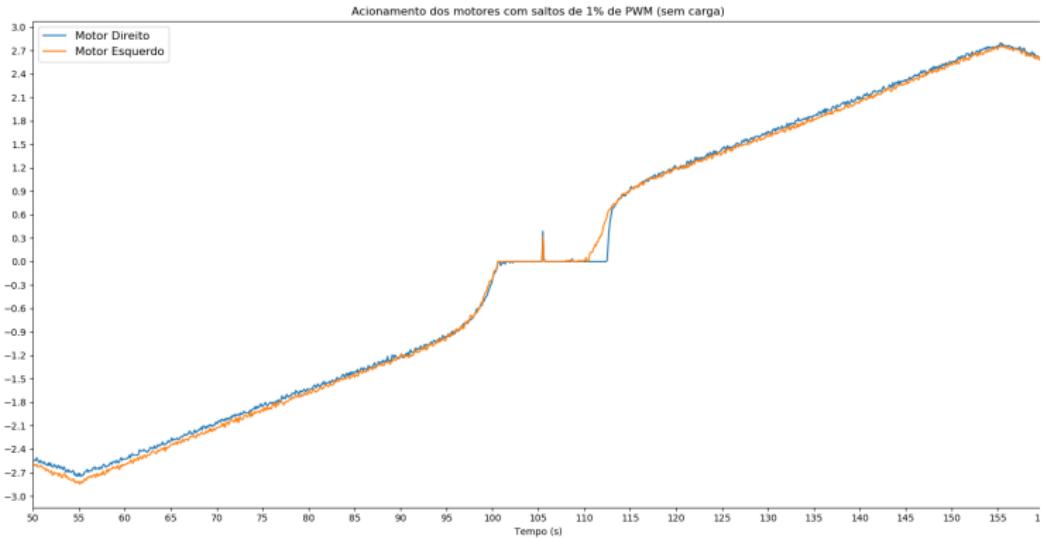


Figure: Curva do motor com carga

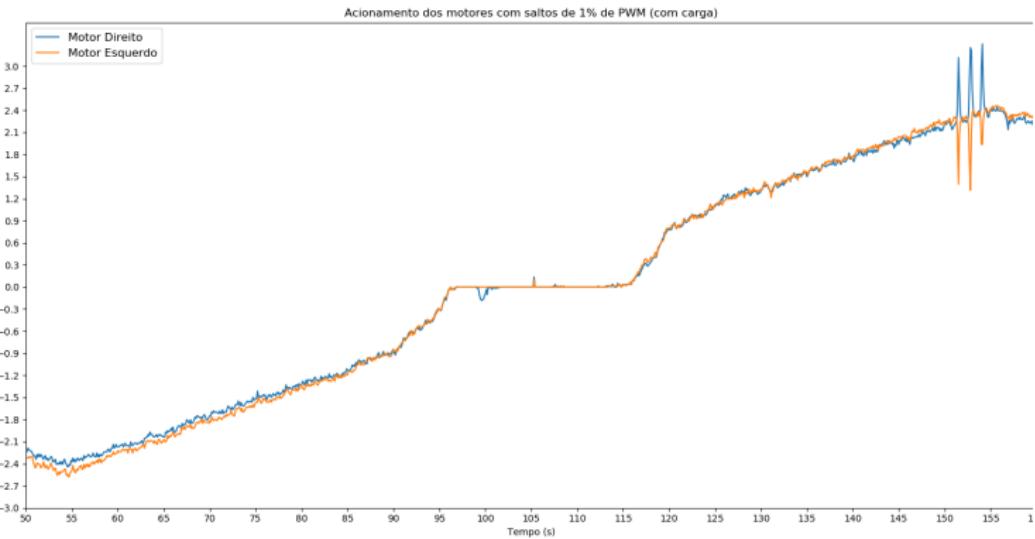
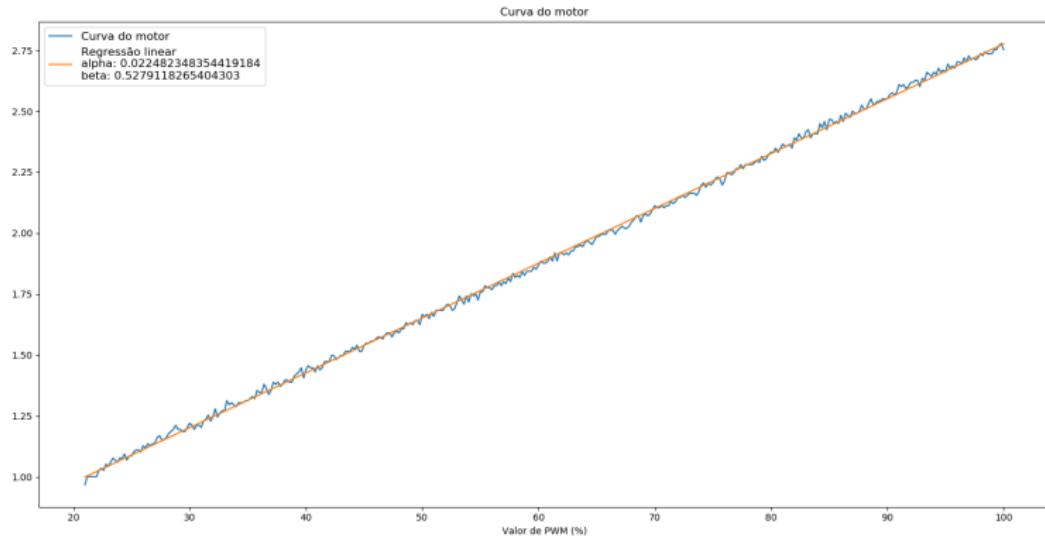


Figure: Regressão Linear para a curva do motor



# Desenvolvimento

## Controlador Híbrido

Implementa arbitragem de comportamentos.

## Controlador Fuzzy

Implementa fusão de comportamentos.

# Controlador Híbrido

## Controlador PID

$$\omega = K_p \epsilon(k) + K_i \sum_0^k \epsilon(k) T + k_d \frac{\epsilon(k) - \epsilon(k-1)}{T} \quad (8)$$

# Controlador Híbrido

## Controlador PID

$$\omega = K_p \epsilon(k) + K_i \sum_0^k \epsilon(k) T + k_d \frac{\epsilon(k) - \epsilon(k-1)}{T} \quad (8)$$

## Considerações

- Velocidade angular tem prioridade sobre velocidade linear.
- É necessário sacrificar velocidade linear em caso de saturação.

## Algoritmo para conversão de modelos

### Algorithm Uniciclo para Acionamento Diferencial priorizando $\omega$

**Entrada:**  $v, \omega$

**Saída:**  $\omega_l, \omega_r$

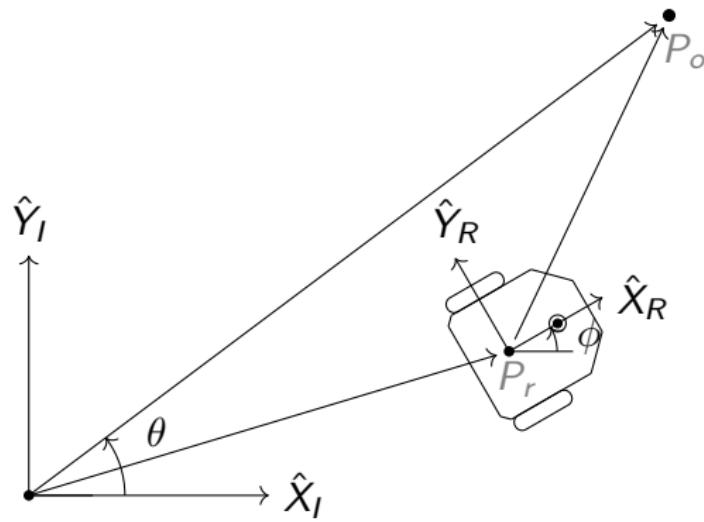
```
1: Se  $v > 0$  Faça
2:    $vlim \leftarrow \max(\min(\text{abs}(v), R \cdot VEL\_MAX), R \cdot VEL\_MIN)$ 
3:    $wlim \leftarrow \max(\min(\text{abs}(\omega), (R/L) \cdot (VEL\_MAX - VEL\_MIN)), 0)$ 
4:    $w_{l,lim}, w_{r,lim} \leftarrow \text{uniToDiff}(vlim, wlim)$ 
5:    $velocidadeMaior \leftarrow \max(w_{l,lim}, w_{r,lim})$ 
6:    $velocidadeMenor \leftarrow \min(w_{l,lim}, w_{r,lim})$ 
7:   Se  $velocidadeMaior > VEL\_MAX$  Faça
8:      $w_{l,lim} \leftarrow w_{l,lim} - (velocidadeMaior - VEL\_MAX)$ 
9:      $w_{r,lim} \leftarrow w_{r,lim} - (velocidadeMaior - VEL\_MAX)$ 
10:  Senão Se  $velocidadeMenor < VEL\_MIN$  Faça
11:     $w_{l,lim} \leftarrow w_{l,lim} + (VEL\_MIN - velocidadeMenor)$ 
12:     $w_{r,lim} \leftarrow w_{r,lim} + (VEL\_MIN - velocidadeMenor)$ 
13:  Fim Se
14:  Se  $v \geq 0$  Faça
15:     $vlim \leftarrow 1$ 
16:  Senão
17:     $vlim \leftarrow -1$ 
18:  Fim Se
```

## Algoritmo para conversão de modelos

### Algorithm Uniciclo para Acionamento Diferencial priorizando $\omega$

```
19:  Se  $\omega \geq 0$  Faça
20:     $wlim \leftarrow 1$ 
21:  Senão
22:     $wlim \leftarrow -1$ 
23:  Fim Se
24:   $v, \omega \leftarrow diffToUni(w_{l,lim}, w_{r,lim})$ 
25:   $v \leftarrow v \cdot vlim$ 
26:   $\omega \leftarrow \omega \cdot wlim$ 
27:  Senão
28:    Se  $abs(\omega) > (R/L) \cdot 2 \cdot VEL\_MIN$  Faça
29:      Se  $\omega \geq 0$  Faça
30:         $\omega \leftarrow max(min(abs(\omega), (R/L) \cdot 2 \cdot VEL\_MAX), (R/L) \cdot 2 \cdot VEL\_MIN)$ 
31:      Senão
32:         $\omega \leftarrow -max(min(abs(\omega), (R/L) \cdot 2 \cdot VEL\_MAX), (R/L) \cdot 2 \cdot VEL\_MIN)$ 
33:    Fim Se
34:    Senão
35:       $\omega \leftarrow 0$ 
36:    Fim Se
37:  Fim Se
38:   $w_l, w_r \leftarrow uniToDiff(v, \omega)$ 
```

# Comportamento Ir Para Objetivo (IPO)



## Equações para Comportamento IPO

$$\mathbf{u_{ipo}} = P_o - P_r \quad (9)$$

## Equações para Comportamento IPO

$$\mathbf{u}_{\text{ipo}} = P_o - P_r \quad (9)$$

$$e_\theta = \text{atan2}(u_{ipo,y}, u_{ipo,x}) - \phi \quad (10)$$

## Equações para Comportamento IPO

$$\mathbf{u}_{\text{ipo}} = P_o - P_r \quad (9)$$

$$e_\theta = \text{atan}2(u_{ipo,y}, u_{ipo,x}) - \phi \quad (10)$$

- $K_p = 4$

## Equações para Comportamento IPO

$$\mathbf{u}_{\text{ipo}} = P_o - P_r \quad (9)$$

$$e_\theta = \text{atan}2(u_{ipo,y}, u_{ipo,x}) - \phi \quad (10)$$

- $K_p = 4$
- $K_i = 0,01$

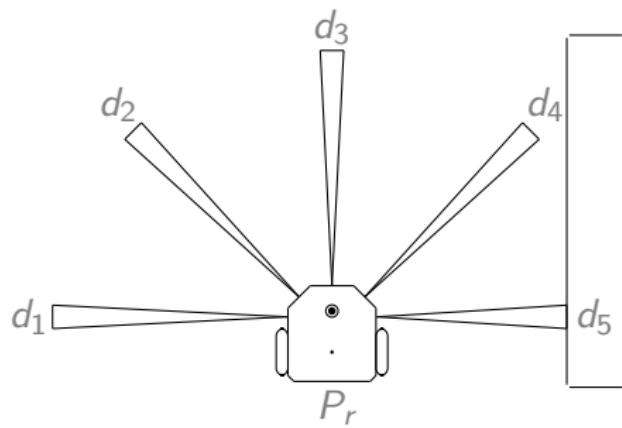
## Equações para Comportamento IPO

$$\mathbf{u}_{\text{ipo}} = P_o - P_r \quad (9)$$

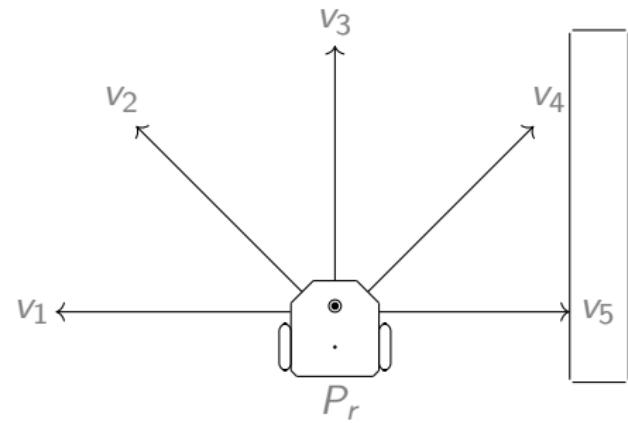
$$e_\theta = \text{atan}2(u_{ipo,y}, u_{ipo,x}) - \phi \quad (10)$$

- $K_p = 4$
- $K_i = 0,01$
- $K_d = 0,01$

# Comportamento Evitar Obstáculo (EO)



(a) Sensores infravermelho



(b) Vetores utilizados

## Conversão de sistemas de coordenadas

$$\begin{bmatrix} x \\ y \end{bmatrix}_i^R = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} d_i \\ 0 \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \end{bmatrix}_i^R \quad (11)$$

## Conversão de sistemas de coordenadas

$$\begin{bmatrix} x \\ y \end{bmatrix}_i^R = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} d_i \\ 0 \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \end{bmatrix}_i^R \quad (11)$$

## Combinação Linear

$$\mathbf{u}_{eo2} = k_1 \mathbf{v}_1 + k_2 \mathbf{v}_2 + k_3 \mathbf{v}_3 + k_4 \mathbf{v}_4 + k_5 \mathbf{v}_5 \quad (12)$$

## Conversão de sistemas de coordenadas

$$\begin{bmatrix} x \\ y \end{bmatrix}_i^R = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} d_i \\ 0 \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \end{bmatrix}_i^R \quad (11)$$

## Combinação Linear

$$\mathbf{u}_{eo2} = k_1 \mathbf{v}_1 + k_2 \mathbf{v}_2 + k_3 \mathbf{v}_3 + k_4 \mathbf{v}_4 + k_5 \mathbf{v}_5 \quad (12)$$

## Equilíbrio

$$\mathbf{u}_{eo1} = k_1 \mathbf{v}_1 + k_2 \mathbf{v}_2 + k_3 \mathbf{v}_3 + k_4 \mathbf{v}_4 + k_5 \mathbf{v}_5 + \mathbf{v}_{eq} \quad (13)$$

## Parâmetros

- Constantes  $k_i$ :

$$\rightarrow k_1 = 0,7$$

$$\rightarrow k_2 = 2$$

$$\rightarrow k_3 = 1,2$$

$$\rightarrow k_4 = 2$$

$$\rightarrow k_5 = 0,7$$

## Parâmetros

- Constantes  $k_i$ :

$$\rightarrow k_1 = 0,7$$

$$\rightarrow k_2 = 2$$

$$\rightarrow k_3 = 1,2$$

$$\rightarrow k_4 = 2$$

$$\rightarrow k_5 = 0,7$$

- $\mathbf{v}_{\text{eq}} = [-240 \ 0]^T$

## Parâmetros

- Constantes  $k_i$ :
  - $k_1 = 0,7$
  - $k_2 = 2$
  - $k_3 = 1,2$
  - $k_4 = 2$
  - $k_5 = 0,7$
- $\mathbf{v}_{\text{eq}} = [-240 \ 0]^T$
- $D_{\text{insegura}} = 25$

## Comportamento EO completo

$$\mathbf{u}_{eo} = \begin{cases} \mathbf{u}_{eo1} \\ \left[ \begin{matrix} \cos(90) & -\sin(90) \\ \sin(90) & \cos(90) \end{matrix} \right] \mathbf{u}_{eo2} \\ \left[ \begin{matrix} \cos(-90) & -\sin(-90) \\ \sin(-90) & \cos(-90) \end{matrix} \right] \mathbf{u}_{eo2} \end{cases}$$

para  $|\mathbf{v}_3| \geq D_{insegura}$   
 para  $|\mathbf{v}_3| < D_{insegura} \wedge$   
 $k_1 (|\mathbf{v}_5| - |\mathbf{v}_1|) + k_2 (|\mathbf{v}_4| - |\mathbf{v}_2|) >$   
 para  $|\mathbf{v}_3| < D_{insegura} \wedge$   
 $k_1 (|\mathbf{v}_5| - |\mathbf{v}_1|) + k_2 (|\mathbf{v}_4| - |\mathbf{v}_2|) \leq$

(14)

# Comportamento Mesclado IPO + EO

## Cálculo da recomendação

$$\mathbf{u}_{eo\_e\_ipo} = k \hat{\mathbf{u}}_{ipo} + (1 - k) \hat{\mathbf{u}}_{eo} \quad (15)$$

# Comportamento Mesclado IPO + EO

## Cálculo da recomendação

$$\mathbf{u}_{eo\_e\_ipo} = k \hat{\mathbf{u}}_{ipo} + (1 - k) \hat{\mathbf{u}}_{eo} \quad (15)$$

- $k = 0,3$

# Comportamento Mesclado IPO + EO

## Cálculo da recomendação

$$\mathbf{u}_{eo\_e\_ipo} = k \hat{\mathbf{u}}_{ipo} + (1 - k) \hat{\mathbf{u}}_{eo} \quad (15)$$

- $k = 0,3$

## IPO

$$\mathbf{u}_{ipo} = P_o - P_r \quad (16)$$

# Comportamento Mesclado IPO + EO

## Cálculo da recomendação

$$\mathbf{u}_{eo\_e\_ipo} = k \hat{\mathbf{u}}_{ipo} + (1 - k) \hat{\mathbf{u}}_{eo} \quad (15)$$

- $k = 0,3$

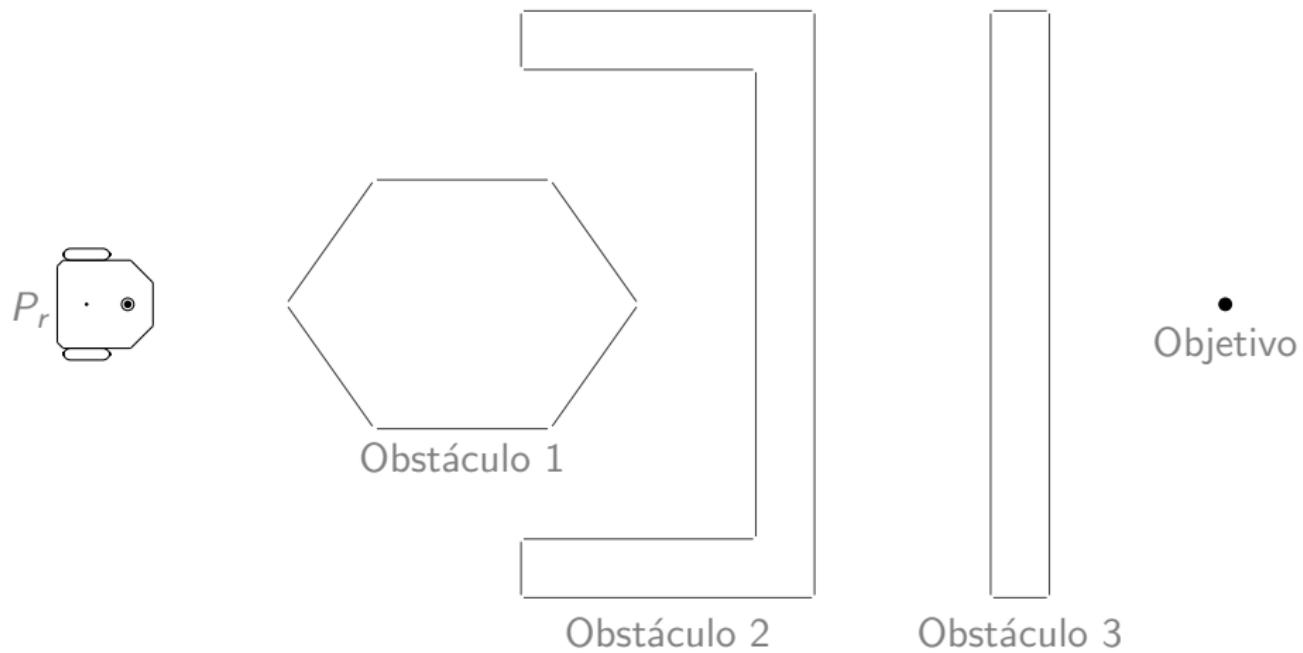
## IPO

$$\mathbf{u}_{ipo} = P_o - P_r \quad (16)$$

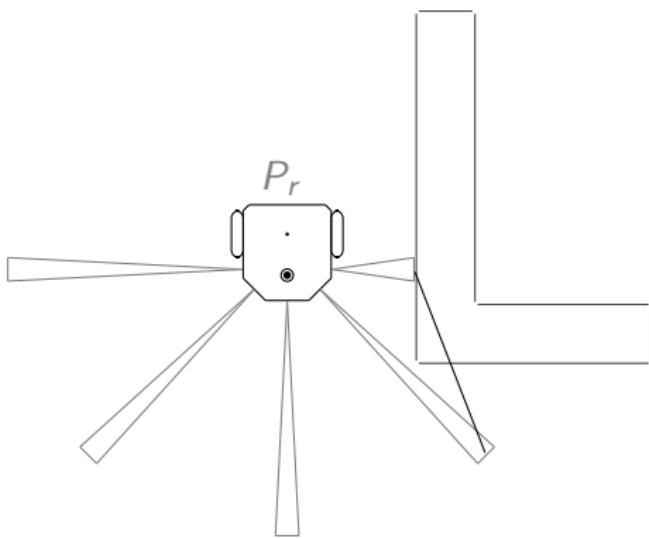
## EO

$$\mathbf{u}_{eo1} = k_1 \mathbf{v}_1 + k_2 \mathbf{v}_2 + k_3 \mathbf{v}_3 + k_4 \mathbf{v}_4 + k_5 \mathbf{v}_5 + \mathbf{v}_{eq} \quad (17)$$

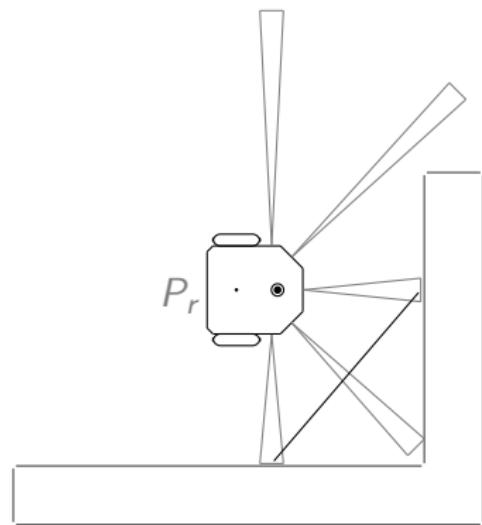
# O problema dos Mínimos Locais



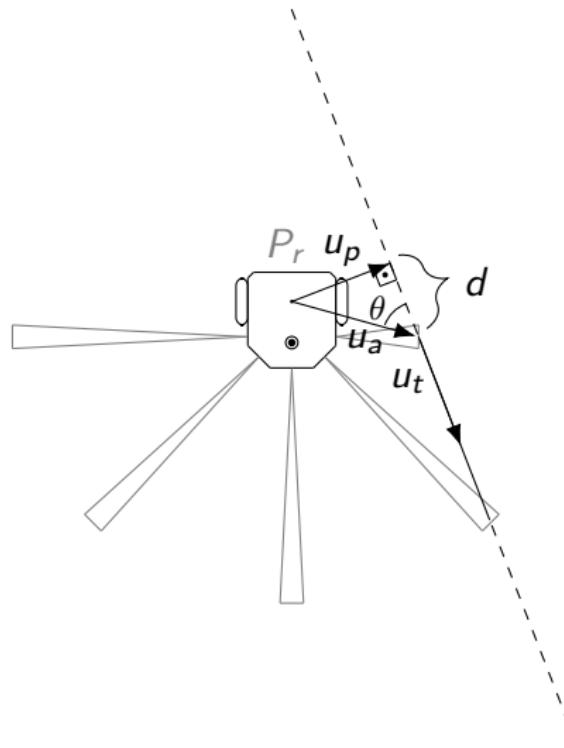
# Comportamento Seguir Parede



(a) Parede subestimada



(b) Parede superestimada



## Equação para Comportamento Seguir Parede

$$\mathbf{u_p} = \mathbf{u_a} - d \cdot \mathbf{u_t} \quad (18)$$

## Equação para Comportamento Seguir Parede

$$\mathbf{u_p} = \mathbf{u_a} - d \cdot \mathbf{u_t} \quad (18)$$

$$\mathbf{u_a} \cdot \mathbf{u_t} = |\mathbf{u_a}| \cdot |\mathbf{u_t}| \cos(\theta) \quad (19)$$

## Equação para Comportamento Seguir Parede

$$\mathbf{u_p} = \mathbf{u_a} - d \cdot \mathbf{u_t} \quad (18)$$

$$\mathbf{u_a} \cdot \mathbf{u_t} = |\mathbf{u_a}| \cdot |\mathbf{u_t}| \cos(\theta) \quad (19)$$

$$\mathbf{u_a} \cdot \mathbf{u_t} = |\mathbf{u_a}| \cos(\theta) = d \quad (20)$$

## Equação para Comportamento Seguir Parede

$$\mathbf{u_p} = \mathbf{u_a} - d \cdot \mathbf{u_t} \quad (18)$$

$$\mathbf{u_a} \cdot \mathbf{u_t} = |\mathbf{u_a}| \cdot |\mathbf{u_t}| \cos(\theta) \quad (19)$$

$$\mathbf{u_a} \cdot \mathbf{u_t} = |\mathbf{u_a}| \cos(\theta) = d \quad (20)$$

$$\mathbf{u_{fw}} = \mathbf{u_t} + \beta \cdot \left( \mathbf{u_p} - d_{fw} \cdot \frac{\mathbf{u_p}}{|\mathbf{u_p}|} \right) \quad (21)$$

## Parâmetros

- $\beta = 5,5$

## Parâmetros

- $\beta = 5,5$
- $d_{fw} = 50cm$

## Parâmetros

- $\beta = 5,5$
- $d_{fw} = 50cm$
- Constantes do Controlador PID:
  - $k_p = 2$
  - $k_i = 0$
  - $k_d = 0$

# Autômato para arbitragem

## Estados

Índice	Comportamento	Descrição
0	P	Parar
1	IPO	Ir para Objetivo
2	EO	Evitar Obstáculos
3	IPO E EO	Ir para Objetivo e Evitar Obstáculos
4	SP AH	Seguir Parede sentido Anti-horário
5	SP H	Seguir Parede sentido Horário

## Eventos

Legenda	Condições
a	No objetivo
b	Fez progresso
c	Tem Obstáculo
d	Está inseguro
e	Livre de obstáculo
f	Contornando pela esquerda
g	Contornando pela direita

## Evento No Objetivo (a)

$$| \mathbf{v}_{\text{objetivo}} - \mathbf{v}_{\text{robô}} | < D_{STOP} \quad (22)$$

## Evento No Objetivo (a)

$$| \mathbf{v}_{\text{objetivo}} - \mathbf{v}_{\text{robô}} | < D_{STOP} \quad (22)$$

## Parâmetro

- $D_{STOP} = 15$

## Evento Fez progresso (b)

### Algorithm Verificação de progresso

**Entrada:**  $d_{prog}$ ,  $v_{objetivo}$ ,  $v_{robô}$

**Saída:**  $d_{prog}$ ,  $retornoBooleano$

- 1: **Se**  $|v_{objetivo} - v_{robô}| < (d_{prog} - D_{PROG\_EPSILON})$  **Faça**
- 2:      $d_{prog} \leftarrow \min(|v_{objetivo} - v_{robô}|, d_{prog})$
- 3:      $retornoBooleano \leftarrow Verdadeiro$
- 4: **Senão Se**  $\text{abs}(|v_{objetivo} - v_{robô}| - d_{prog}) \leq D_{PROG\_EPSILON}$  **Faça**
- 5:      $retornoBooleano \leftarrow Verdadeiro$
- 6: **Senão**
- 7:      $retornoBooleano \leftarrow Falso$
- 8: **Fim Se**

## Evento Fez progresso (b)

### Algorithm Verificação de progresso

**Entrada:**  $d_{prog}$ ,  $v_{objetivo}$ ,  $v_{robô}$

**Saída:**  $d_{prog}$ ,  $retornoBooleano$

```
1: Se  $|v_{objetivo} - v_{robô}| < (d_{prog} - D_{PROG\_EPSILON})$  Faça
2:    $d_{prog} \leftarrow \min(|v_{objetivo} - v_{robô}|, d_{prog})$ 
3:    $retornoBooleano \leftarrow Verdadeiro$ 
4: Senão Se  $\text{abs}(|v_{objetivo} - v_{robô}| - d_{prog}) \leq D_{PROG\_EPSILON}$  Faça
5:    $retornoBooleano \leftarrow Verdadeiro$ 
6: Senão
7:    $retornoBooleano \leftarrow Falso$ 
8: Fim Se
```

## Parâmetro

- $d_{prog}$  inicial vale 1000.

## Evento Fez progresso (b)

### Algorithm Verificação de progresso

**Entrada:**  $d_{prog}$ ,  $v_{objetivo}$ ,  $v_{robô}$

**Saída:**  $d_{prog}$ ,  $retornoBooleano$

```
1: Se  $|v_{objetivo} - v_{robô}| < (d_{prog} - D_{PROG\_EPSILON})$  Faça
2:    $d_{prog} \leftarrow \min(|v_{objetivo} - v_{robô}|, d_{prog})$ 
3:    $retornoBooleano \leftarrow Verdadeiro$ 
4: Senão Se  $\text{abs}(|v_{objetivo} - v_{robô}| - d_{prog}) \leq D_{PROG\_EPSILON}$  Faça
5:    $retornoBooleano \leftarrow Verdadeiro$ 
6: Senão
7:    $retornoBooleano \leftarrow Falso$ 
8: Fim Se
```

## Parâmetro

- $d_{prog}$  inicial vale 1000.
- $D_{PROG\_EPSILON} = 2$

## Evento Tem Obstaculo (c)

$$\text{any}(\mathbf{d}_s < D_{\text{EM\_OBSTÁCULO}}) \quad (23)$$

## Evento Tem Obstaculo (c)

$$\text{any}(\mathbf{d}_s < D_{\text{EM\_OBSTÁCULO}}) \quad (23)$$

## Parâmetro

- $D_{\text{EM\_OBSTÁCULO}} = 75$

## Evento Está inseguro (d)

$$\text{any}(\mathbf{d}_s < D_{INSEGURO}) \quad (24)$$

## Evento Está inseguro (d)

$$\text{any}(\mathbf{d_s} < D_{INSEGURO}) \quad (24)$$

### Parâmetro

- $D_{INSEGURO} = 25$

## Evento Livre de Obstáculo (e)

$$\text{all}(\mathbf{d}_s > D_{\text{EM\_OBSTÁCULO}}) \quad (25)$$

## Condição de contorno (eventos f e g)

$$[\mathbf{u}_{\text{ipo}} \ \mathbf{u}_{\text{eo}}] \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \mathbf{u}_{\text{sp}} \quad (26)$$

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = [\mathbf{u}_{\text{ipo}} \ \mathbf{u}_{\text{eo}}]^{-1} \cdot \mathbf{u}_{\text{sp}} \quad (27)$$

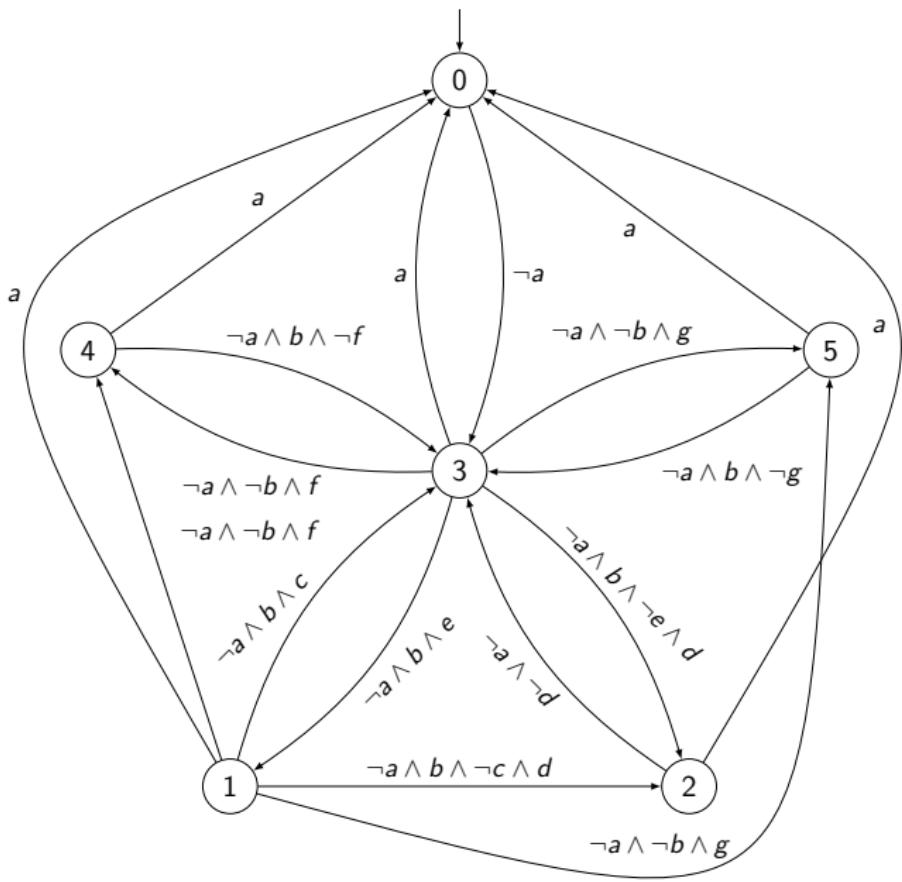
## Algoritmo para verificação de situação de contorno

### Algorithm Verificação de situação de deslize em fronteira

**Entrada:** sentidoDeContorno

**Saída:** retornoBooleano

```
1:  $u_{ipo,x}, u_{ipo,y} \leftarrow vetorIrParaObjetivo()$ 
2:  $u_{eo,x}, u_{eo,y} \leftarrow vetorEvitarObstaculo()$ 
3:  $u_{sp,x}, u_{sp,y} \leftarrow vetorSeguirParede(sentidoDeContorno)$ 
4:  $determinante \leftarrow u_{ipo,x}u_{ao,y} - u_{ipo,y}u_{ao,x}$ 
5:  $\sigma_1 \leftarrow (u_{ao,y}u_{sp,x} - u_{ao,x}u_{sp,y})/determinante$ 
6:  $\sigma_2 \leftarrow (-u_{ipo,y}u_{sp,x} + u_{ipo,x}u_{sp,y})/determinante$ 
7: Se obstaculoPresente(sentidoDeContorno) E  $\sigma_1 > 0$  E  $\sigma_2 > 0$  Faça
8:     retornoBooleano  $\leftarrow$  Verdadeiro
9: Senão
10:    retornoBooleano  $\leftarrow$  Falso
11: Fim Se
```



## Explicação

- De todos os estados é possível chegar ao estado inicial “Parar” (0) por meio da condição “No Objetivo” (a), mas para sair dele, uma redefinição de objetivo provoca a negação da condição de parada ( $\neg a$ ) e o estado mesclado “Ir para Objetivo e Evitar Obstáculo” (3) assume controle.

## Explicação

- De todos os estados é possível chegar ao estado inicial “Parar” (0) por meio da condição “No Objetivo” (a), mas para sair dele, uma redefinição de objetivo provoca a negação da condição de parada ( $\neg a$ ) e o estado mesclado “Ir para Objetivo e Evitar Obstáculo” (3) assume controle.
- A partir desse último, as transições para “Seguir Parede” (4 e 5) dependem da condição “Não Fez Progresso” ( $\neg b$ ), mas a escolha do sentido de contorno depende das condições “contornando pela esquerda” e “contornando pela direita” (f e g). A partir dos estados 4 ou 5, o retorno para o estado 3 depende de não ter encontrado objetivo ( $\neg a$ ), voltar a fazer progresso (b) e deixar a situação de seguir fronteira ( $\neg f$  ou  $\neg g$ ).

## Explicação

- O estado 3 pode alcançar “Ir Para Objetivo” (1) quando está “livre de obstáculo” (e) e retorna quando volta a “ter obstáculo” (c). O estado 1 alcança “Seguir Parede” com as mesmas condições utilizadas a partir do estado 3.

## Explicação

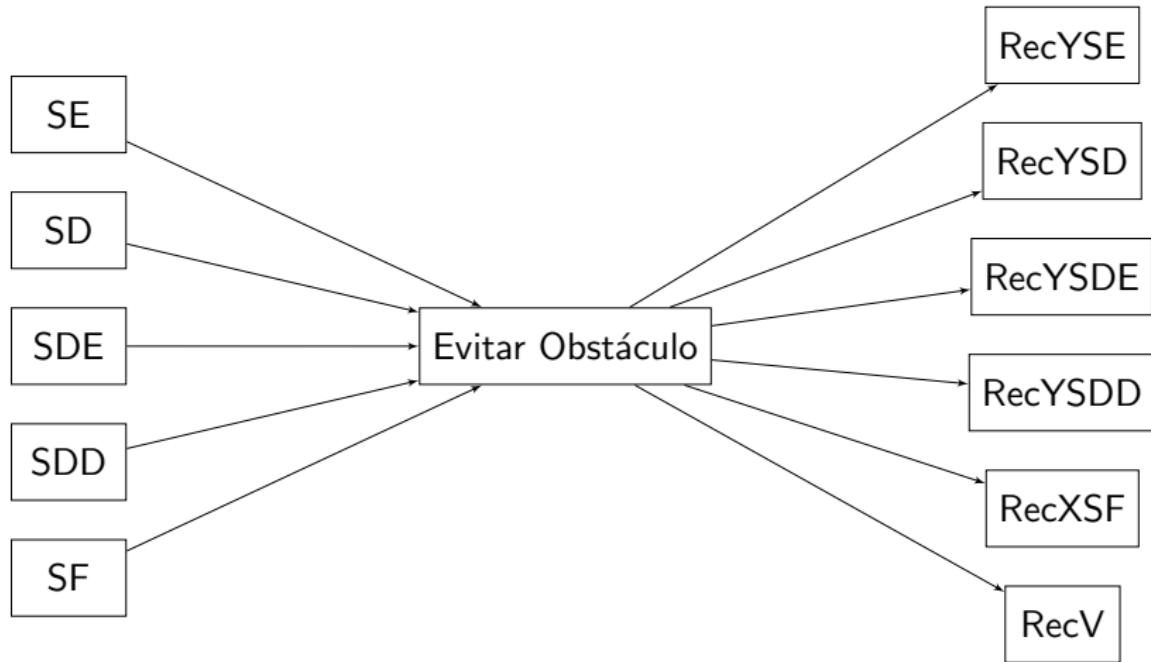
- O estado 3 pode alcançar “Ir Para Objetivo” (1) quando está “livre de obstáculo” (e) e retorna quando volta a “ter obstáculo” (c). O estado 1 alcança “Seguir Parede” com as mesmas condições utilizadas a partir do estado 3.
- Os estados 1 e 3 podem alcançar “Evitar Obstáculo” (2) por meio da condição “está inseguro” (d), mas uma vez neste estado, só pode retornar ao estado 3, com condição de “não estar inseguro” ( $\neg d$ ).

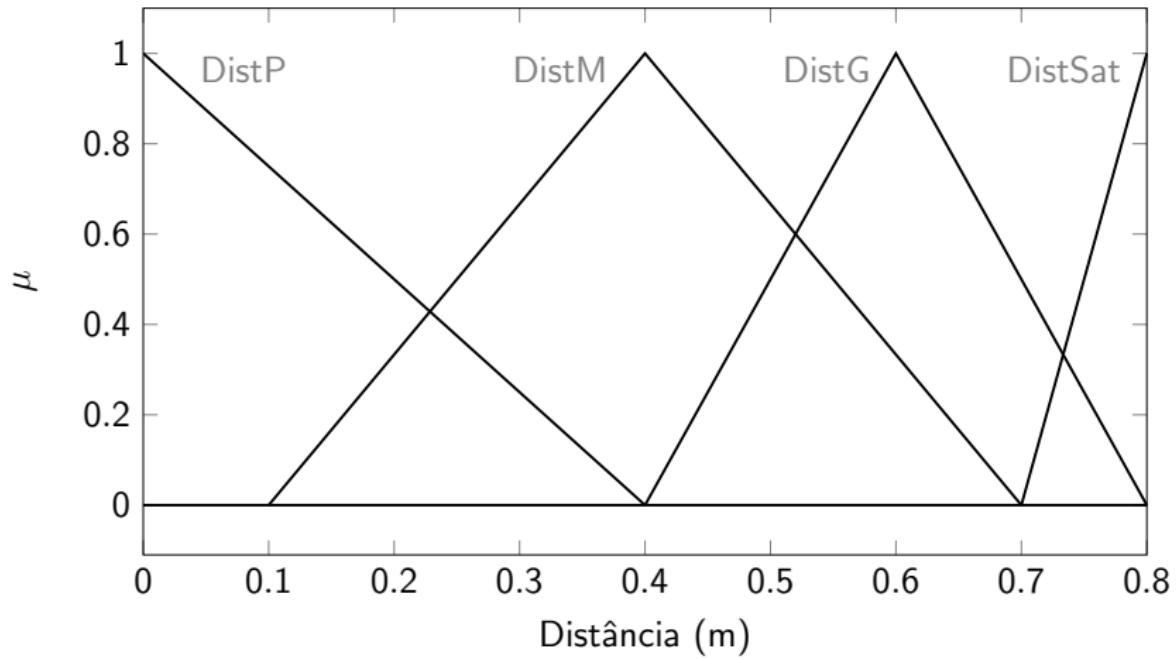
# Controlador Fuzzy

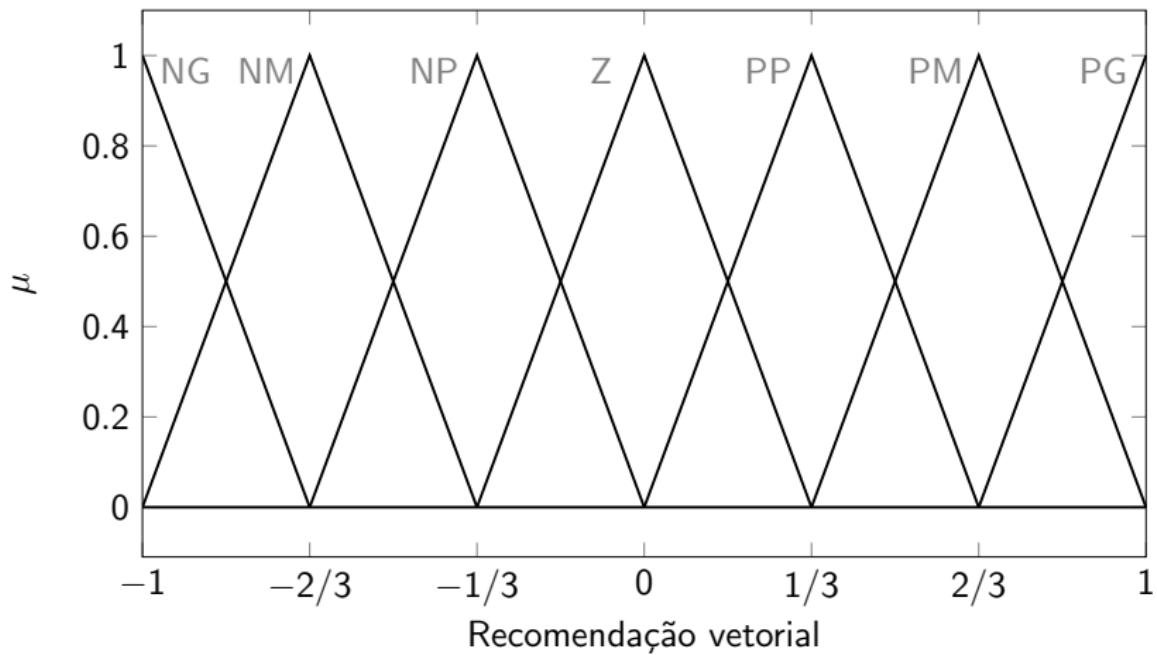
## Cálculo do comportamento Ir Para Objetivo (IPO)

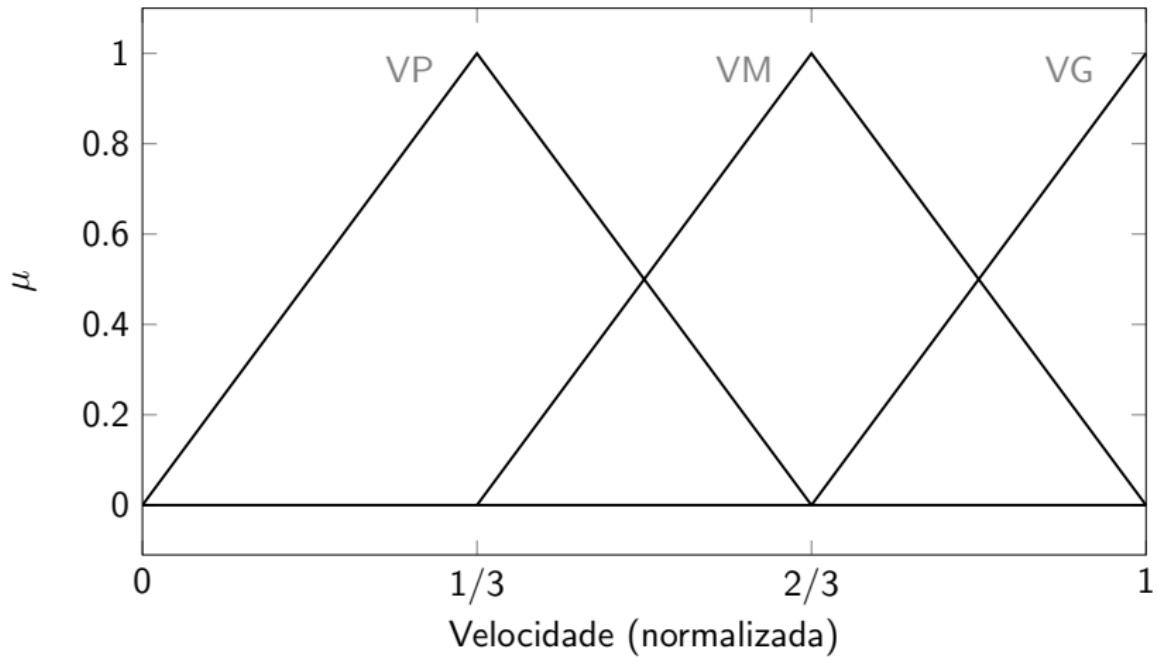
$$\mathbf{u}_{\text{ipo}, \text{fuzzy}} = \begin{cases} \begin{bmatrix} \cos(e_\theta) \\ \sin(e_\theta) \end{bmatrix} & , \text{ para } |\mathbf{u}_{\text{ipo}}| \geq 1 \\ |\mathbf{u}_{\text{ipo}}| \cdot \begin{bmatrix} \cos(e_\theta) \\ \sin(e_\theta) \end{bmatrix} & , \text{ para } |\mathbf{u}_{\text{ipo}}| < 1 \end{cases} \quad (28)$$

# Comportamento Evitar Obstáculo (EO)









## Cálculo do Comportamento

$$\mathbf{u}_{eo,temp} = \left[ \frac{2 \cdot RecXSF}{RecYSE + RecYSD + RecYSDE + RecYSDD} \right] \quad (29)$$

## Cálculo do Comportamento

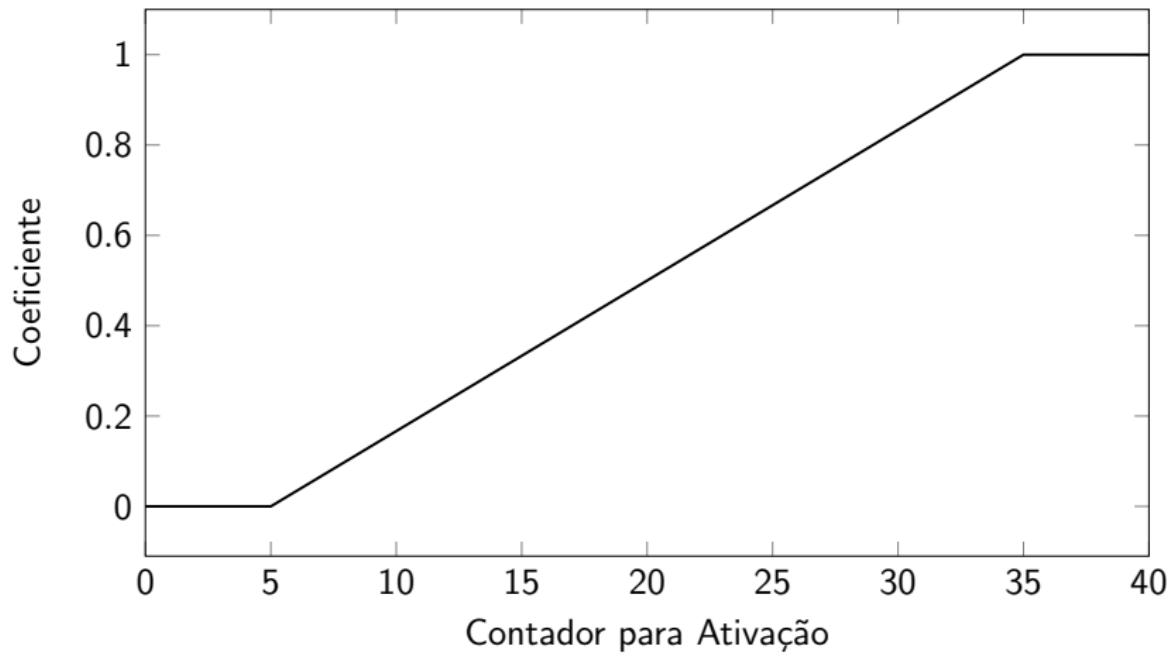
$$\mathbf{u}_{eo,temp} = \left[ \frac{2 \cdot RecXSF}{RecYSE + RecYSD + RecYSDE + RecYSDD} \right] \quad (29)$$

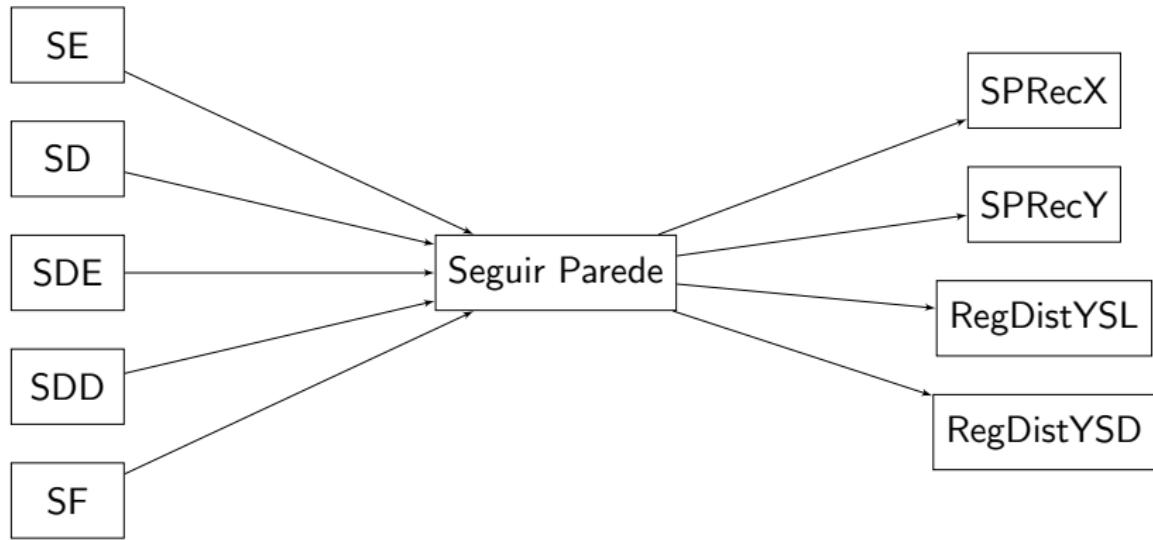
$$\mathbf{u}_{eo,fuzzy} = \begin{cases} \mathbf{u}_{eo,temp} & , \text{ para } |\mathbf{u}_{eo,temp}| \leq 1 \\ \frac{\mathbf{u}_{eo,temp}}{|\mathbf{u}_{eo,temp}|} & , \text{ para } |\mathbf{u}_{eo,temp}| > 1 \end{cases} \quad (30)$$

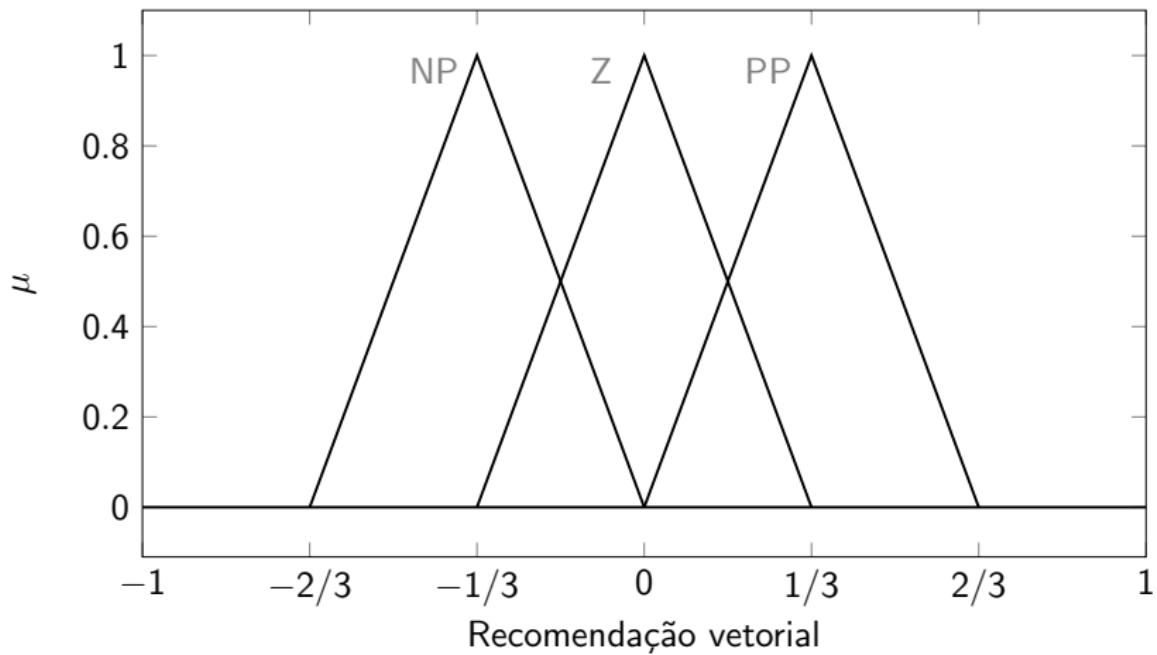
# Regras Fuzzy - Evitar Obstáculo

	Entradas					Saídas					
	SE	SD	SDE	SDD	SF	RecYSE	RecYSD	RecYSDE	RecYSDD	RecXSF	RecV
1	DistP	-	-	-	-	NG	-	-	-	-	-
2	DistM	-	-	-	-	NM	-	-	-	-	-
3	DistG	-	-	-	-	NP	-	-	-	-	-
4	DistSat	-	-	-	-	Z	-	-	-	-	VG
5	-	DistP	-	-	-	-	PG	-	-	-	-
6	-	DistM	-	-	-	-	PM	-	-	-	-
7	-	DistG	-	-	-	-	PP	-	-	-	-
8	-	DistSat	-	-	-	-	Z	-	-	-	VG
9	-	-	DistP	-	-	-	-	NG	-	-	VP
10	-	-	DistM	-	-	-	-	NM	-	-	VM
11	-	-	DistG	-	-	-	-	NP	-	-	VM
12	-	-	DistSat	-	-	-	-	Z	-	-	VG
13	-	-	-	DistP	-	-	-	-	PG	-	VP
14	-	-	-	DistM	-	-	-	-	PM	-	VM
15	-	-	-	DistG	-	-	-	-	PP	-	VM
16	-	-	-	DistSat	-	-	-	-	Z	-	VG
17	-	-	-	-	DistP	-	-	-	-	NG	VP
18	-	-	-	-	DistM	-	-	-	-	NM	VM
19	-	-	-	-	DistG	-	-	-	-	NP	VM
20	-	-	-	-	DistSat	-	-	-	-	Z	VG

# Comportamento Seguir Parede (SP)







## Cálculo Para Seguir Parede

### Algorithm Cálculo Final da Recomendação “Seguir Parede”

**Entrada:**  $t$

**Saída:**  $u_{sp,fuzzy,x}, u_{sp,fuzzy,y}$

- 1:  $C \leftarrow verificaAtivacaoSP()$
- 2:  $u_{sp,temp,x}, u_{sp,temp,y} \leftarrow calculaSPFuzzy()$
- 3: **Se**  $C > 0$  **Faça**
- 4:    **definirSentidoDeContorno()**
- 5:    **verificarPerdaDeReferencia()**
- 6:    **normalizarEntradasVetor()**
- 7: **Senão**
- 8:     $u_{sp,fuzzy,x} \leftarrow u_{sp,temp,x}$
- 9:     $u_{sp,fuzzy,y} \leftarrow u_{sp,temp,y}$
- 10: **Fim Se**

## Função verificaAtivacaoSP()

### Algorithm Verificar constante de ativação

**Entrada:** *ativacaoSP*

**Saída:** *ativacaoSP, SPDdir, C*

```
1: Se fezProgresso() Faça
2:   Se ativacaoSP > 0 Faça
3:     ativacaoSP ← ativacaoSP – 1
4:   Fim Se
5:   Se ativacaoSP < ATIVACAO_MARGEM Faça
6:     SPDdir ← 0
7:   Fim Se
8: Senão
9:   Se ativacaoSP < ATIVACAO_PASSOS Faça
10:    ativacaoSP ← ativacaoSP + 1
11:  Fim Se
12: Fim Se
13: C ←  $\frac{\min(\max(ativacaoSP, ATIVACAO_MARGEM), ATIVACAO_PASSOS - ATIVACAO_MARGEM)}{(ATIVACAO_PASSOS - 2 * ATIVACAO_MARGEM)}$ 
```

## Função definirSentidoDeContorno()

### Algorithm Verificação do sentido de contorno

**Entrada:**  $SpDir$ ,  $RegDistYSL$ ,  $RegDistYSD$

**Saída:**  $SpDir$

```
1: Se  $SpDir = 0$  Faça
2:   Se  $RegDistYSL + 2,5 \cdot RegDistYSD > \epsilon$  Faça
3:      $SpDir \leftarrow 1$ 
4:   Senão Se  $RegDistYSL + 2,5 \cdot RegDistYSD < -\epsilon$  Faça
5:      $SpDir \leftarrow -1$ 
6:   Senão
7:      $SpDir \leftarrow SpDir$ 
8:   Fim Se
9: Fim Se
```

## Função verificarPerdaDeReferencia()

### Algorithm Verificar Perda de Referência

**Entrada:**  $SpDir, u_{sp,fuzzy,x}, u_{sp,fuzzy,y}$

**Saída:**  $u_{sp,fuzzy,y}$

- 1: **Se**  $\sqrt{u_{sp,fuzzy,x}^2 + u_{sp,fuzzy,y}^2} < 0,05$  **Faça**
- 2:     **Se**  $SpDir = 1$  **Faça**
- 3:          $u_{sp,fuzzy,y} \leftarrow 0,3$
- 4:     **Senão Se**  $SpDir = -1$  **Faça**
- 5:          $u_{sp,fuzzy,y} \leftarrow -0,3$
- 6:     **Senão**
- 7:          $u_{sp,fuzzy,y} \leftarrow u_{sp,fuzzy,y}$
- 8:     **Fim Se**
- 9: **Fim Se**

## Função normalizarEntradasVetor()

### Algorithm Normalização de Componentes do vetor

**Entrada:**  $u_{sp,fuzzy,x}, u_{sp,fuzzy,y}$

**Saída:**  $u_{sp,fuzzy,x}, u_{sp,fuzzy,y}$

- 1:  $modulo \leftarrow | u_{sp,fuzzy,x} |$
- 2: **Se**  $modulo > 1$  **Faça**
- 3:      $u_{sp,fuzzy,x} \leftarrow u_{sp,fuzzy,x} / modulo$
- 4:      $u_{sp,fuzzy,y} \leftarrow u_{sp,fuzzy,y} / modulo$
- 5: **Fim Se**
- 6:  $modulo \leftarrow | u_{sp,fuzzy,y} |$
- 7: **Se**  $modulo > 1$  **Faça**
- 8:      $u_{sp,fuzzy,x} \leftarrow u_{sp,fuzzy,x} / modulo$
- 9:      $u_{sp,fuzzy,y} \leftarrow u_{sp,fuzzy,y} / modulo$
- 10: **Fim Se**

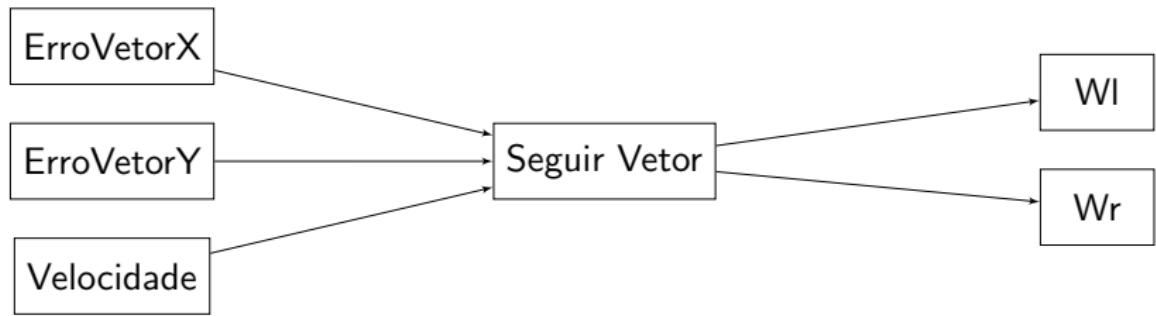
# Regras Fuzzy para Seguir Parede

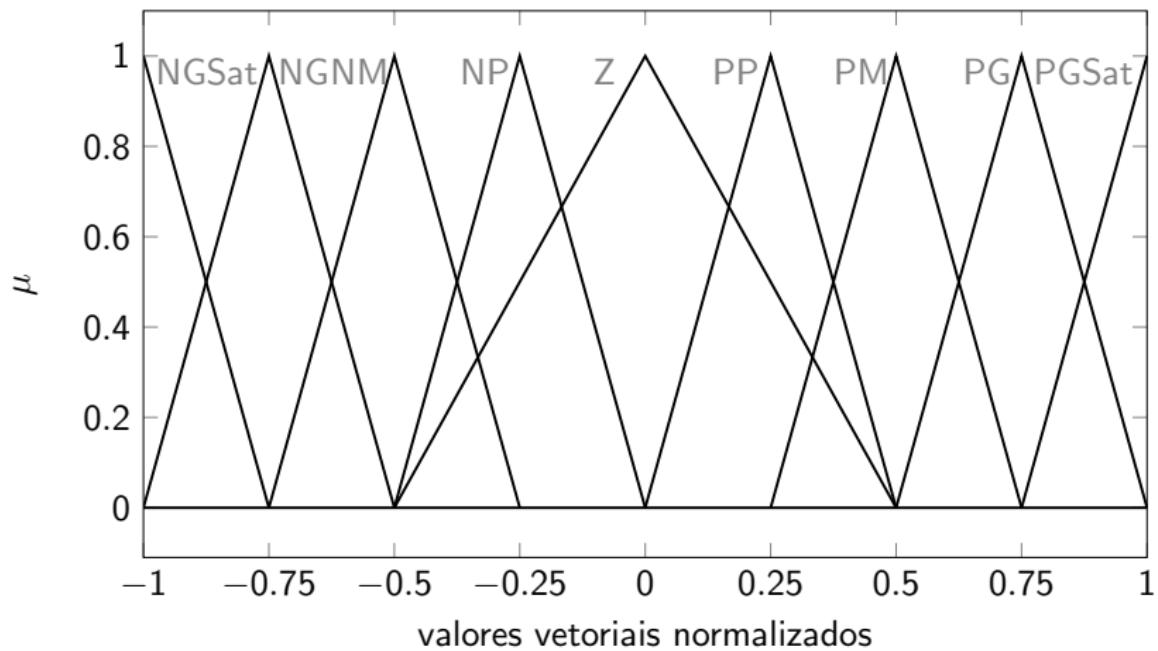
Relação das Entradas		Entradas					Saídas			
		SE	SD	SDE	SDD	SF	SPRecX	SPRecY	RegDistYSL	RegDistYSD
AND	1	DistSat	DistSat	DistSat	DistSat	-	Z	Z	-	-
OR	2	$\neg$ DistSat	$\neg$ DistSat	$\neg$ DistSat	$\neg$ DistSat	-	PP	-	-	-
AND	3	$\neg$ DistSat	-	DistSat	-	-	-	PP	Z	-
AND	4	-	$\neg$ DistSat	-	DistSat	-	-	NP	Z	-
AND	5	$\neg$ DistSat	-	-	-	$\neg$ DistSat	-	NP	Z	-
AND	6	-	$\neg$ DistSat	-	-	$\neg$ DistSat	-	PP	Z	-
-	7	DistP	-	-	-	-	-	-	Z	-
-	8	DistM	-	-	-	-	-	-	PP	-
-	9	DistG	-	-	-	-	-	-	PP	-
-	10	-	DistP	-	-	-	-	-	Z	-
-	11	-	DistM	-	-	-	-	-	NP	-
-	12	-	DistG	-	-	-	-	-	NP	-
-	13	-	-	DistP	-	-	-	-	-	Z
-	14	-	-	DistM	-	-	-	-	-	PP
-	15	-	-	DistG	-	-	-	-	-	PP
-	16	-	-	-	DistP	-	-	-	-	Z
-	17	-	-	-	DistM	-	-	-	-	NP
-	18	-	-	-	DistG	-	-	-	-	NP
AND	19	$\neg$ DistSat	-	$\neg$ DistSat	-	$\neg$ DistSat	-	NP	Z	-
AND	20	-	$\neg$ DistSat	-	$\neg$ DistSat	$\neg$ DistSat	-	PP	Z	-

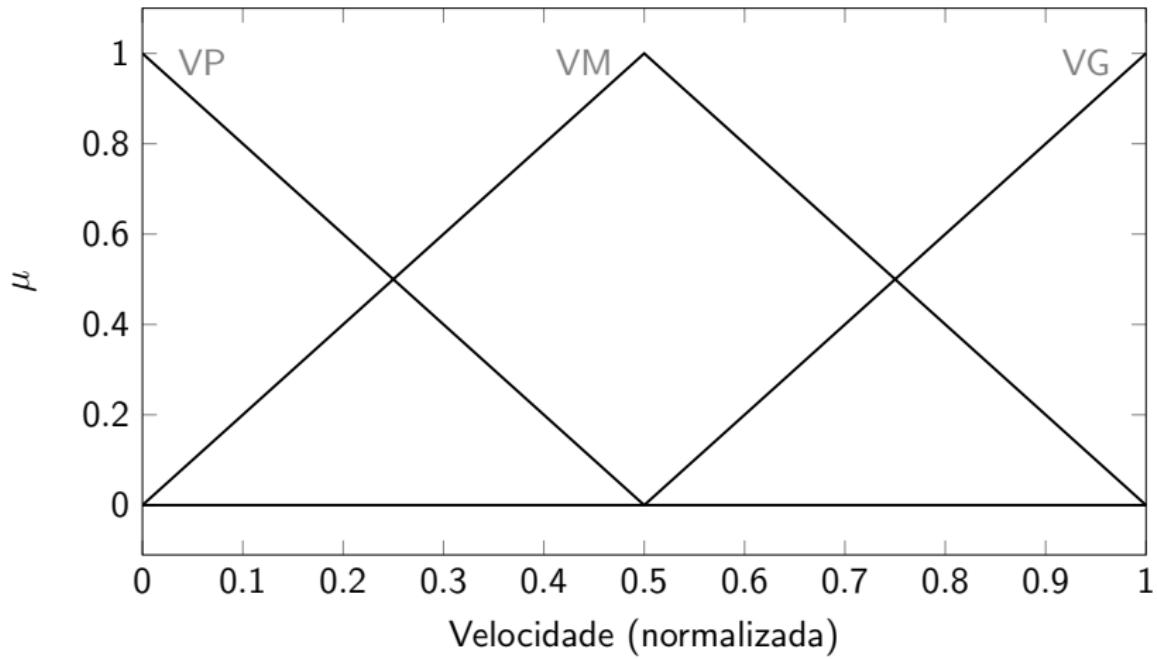
# Controlador Fuzzy para Seguir Recomendação

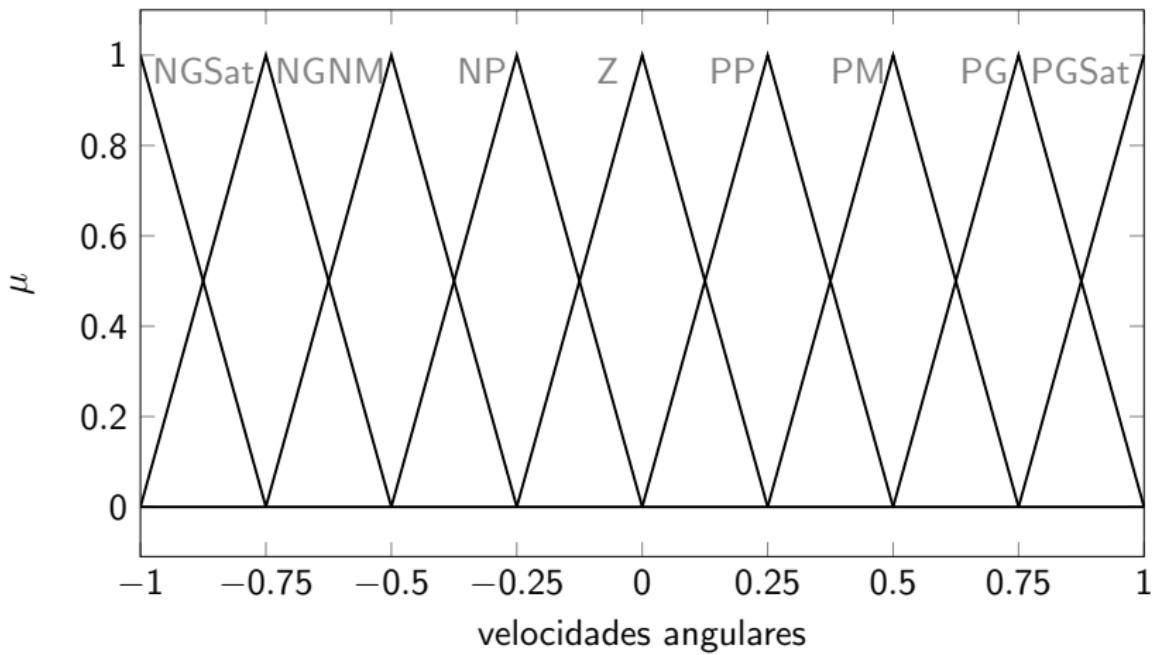
Equação para fusão de comportamentos

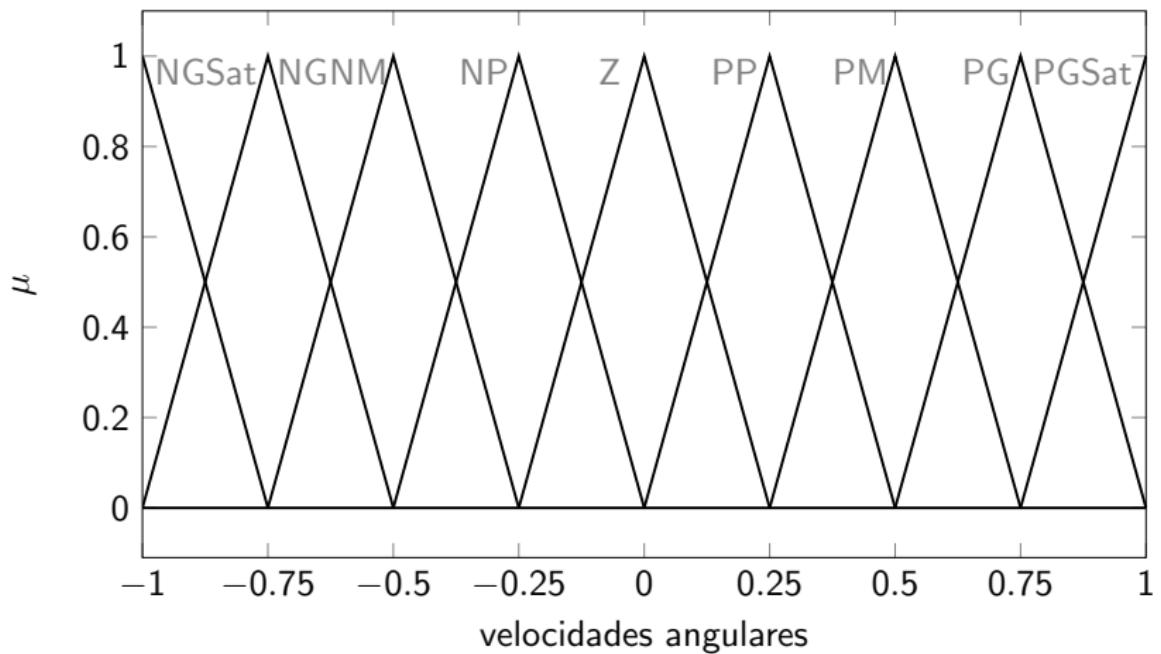
$$\mathbf{u}_{\text{final}} = (1 - C) \cdot (\mathbf{u}_{\text{ipo}} + \alpha \cdot \mathbf{u}_{\text{ao,fuzzy}}) + C \cdot \mathbf{u}_{\text{sp,fuzzy}} \quad (31)$$











## Tabela para Regras Fuzzy

X \ Y	-4	-3	-2	-1	0	1	2	3	4
-4	1	1	1	1	1	1	1	1	1
-3	1	1	1	1	1	1	1	1	1
-2	1	1	1	1	1	1	1	1	1
-1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1

X \ Y	-4	-3	-2	-1	0	1	2	3	4
-4	2	2	2	2	2	2	2	2	2
-3	2	1	1	1	1	1	1	1	2
-2	2	1	1	1	1	1	1	1	2
-1	2	1	1	1	1	1	1	1	2
0	2	1	1	1	0	1	1	1	2
1	2	1	1	1	1	1	1	1	2
2	2	1	1	1	1	1	1	1	2
3	2	1	1	1	1	1	1	1	2
4	2	2	2	2	2	2	2	2	2

(a) Recomendacão para velocidade VP

X \ Y	-4	-3	-2	-1	0	1	2	3	4
-4	3	3	3	3	3	3	3	3	3
-3	3	2	2	2	2	2	2	2	3
-2	3	2	2	2	2	2	2	2	3
-1	3	2	2	1	1	1	2	2	3
0	3	2	2	1	0	1	2	2	3
1	3	2	2	1	1	1	2	2	3
2	3	2	2	2	2	2	2	2	3
3	3	2	2	2	2	2	2	2	3
4	3	3	3	3	3	3	3	3	3

(c) Recomendacão para velocidade VG

(b) Recomendacão para velocidade VM

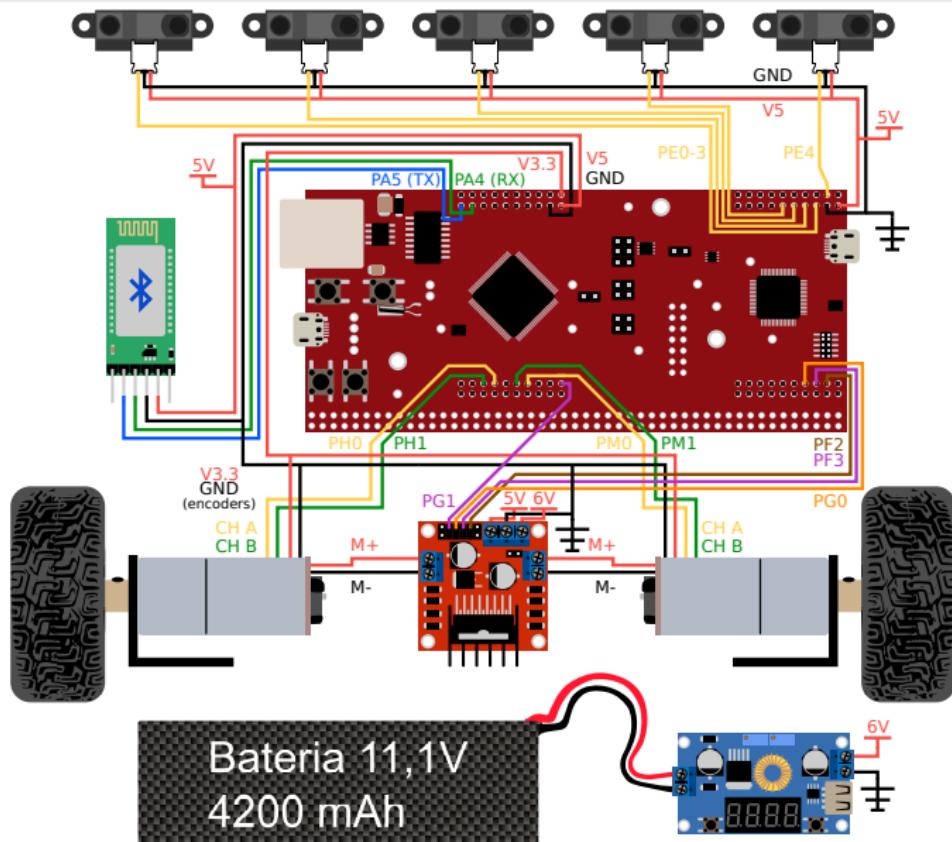
X \ Y	-4	-3	-2	-1	0	1	2	3	4
-4	4	4	4	4	4	4	4	4	4
-3	4	4	4	3	3	3	3	3	3
-2	4	4	3	2	2	2	2	2	2
-1	4	3	2	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0
1	-4	-3	-2	-1	-1	-1	-1	-1	-1
2	-4	-4	-3	-2	-2	-2	-2	-2	-2
3	-4	-4	-4	-3	-3	-3	-3	-3	-3
4	-4	-4	-4	-4	-4	-4	-4	-4	-4

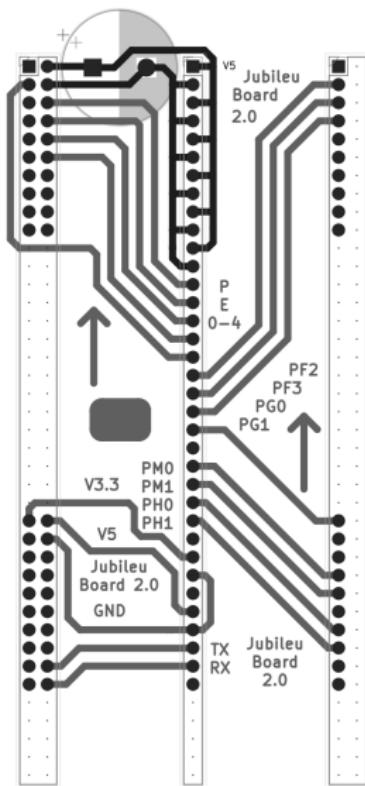
(d) Recomendação para velocidade angular

## Equação para Regras Fuzzy

$$\begin{bmatrix} RecWI \\ RecWr \end{bmatrix} = \begin{bmatrix} \min(\max(RecV - RecW, -4), 4) \\ \min(\max(RecV + RecW, -4), 4) \end{bmatrix} \quad (32)$$

# Circuitos



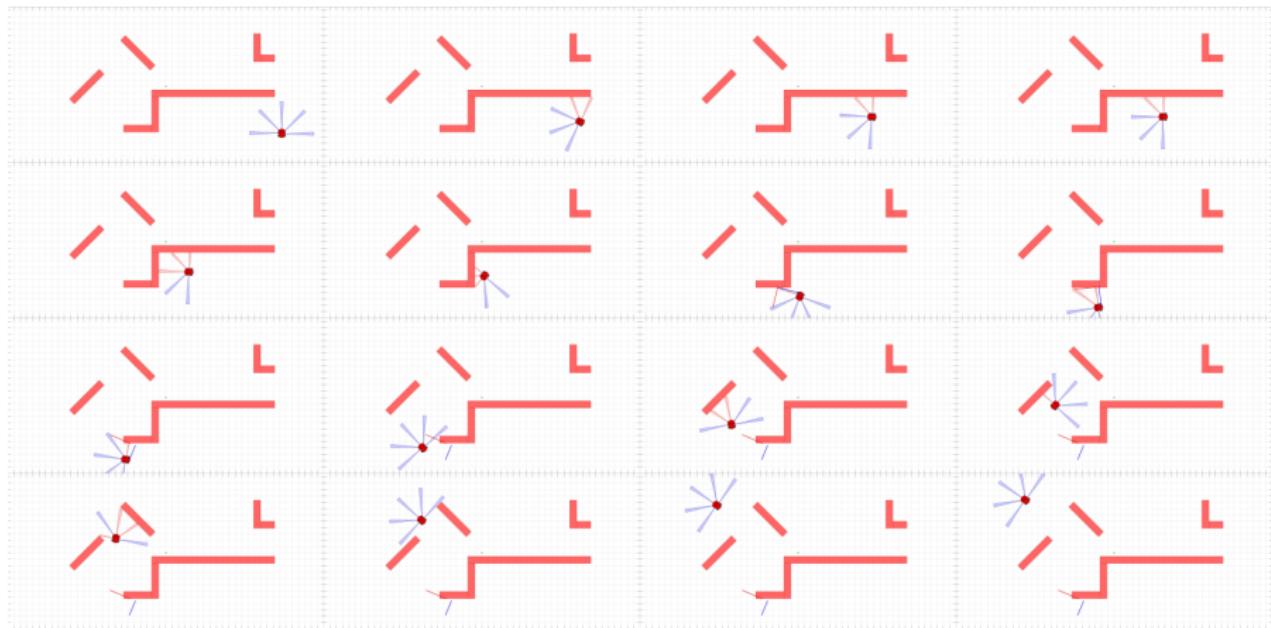


# Comunicação

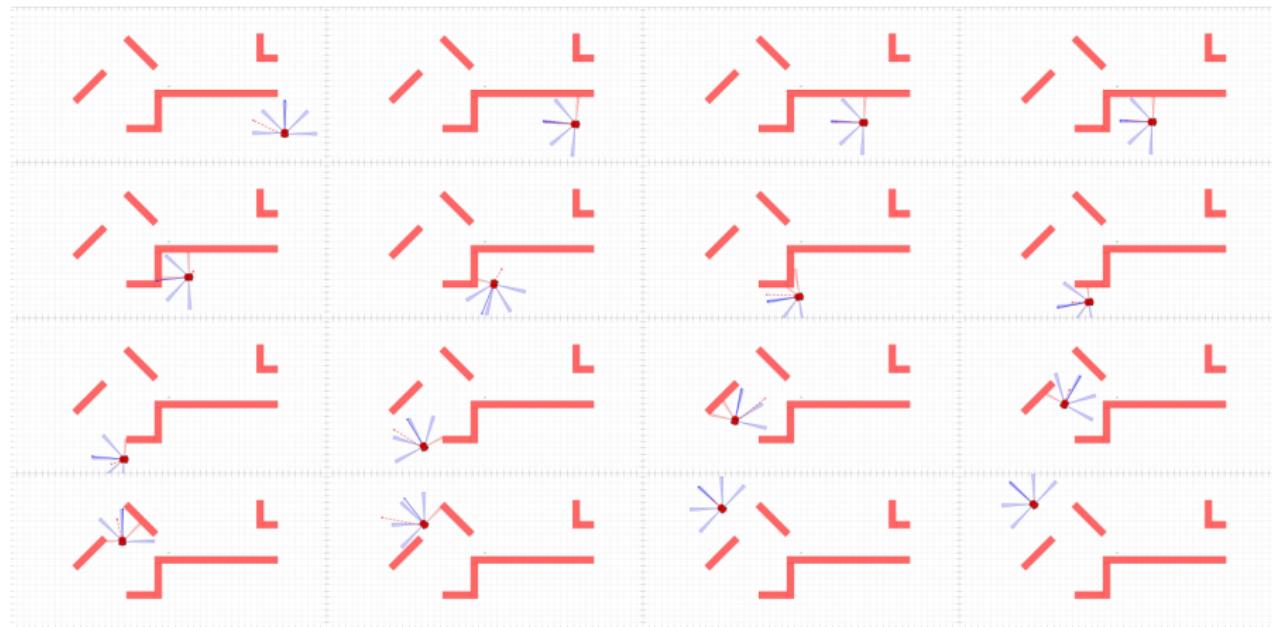
## Interface por Linha de Comando

Comando	Descrição
SUPERVISOR_HIBRIDO	Adota o controlador híbrido como a estratégia de navegação utilizada
SUPERVISOR_FUZZY	Adota o controlador <i>fuzzy</i> como a estratégia de navegação utilizada
SETCOORDOBJ(X.XX,Y.YY)	Redefine a coordenada objetivo
GETSTATE	Utilizado para obter a resposta do estado do robô ( $x$ , $y$ e $\theta$ ) a cada iteração, bem como o valor de seus sensores
EXIT	Necessário para solicitar ao robô para que deixe de informar seu estado a cada iteração

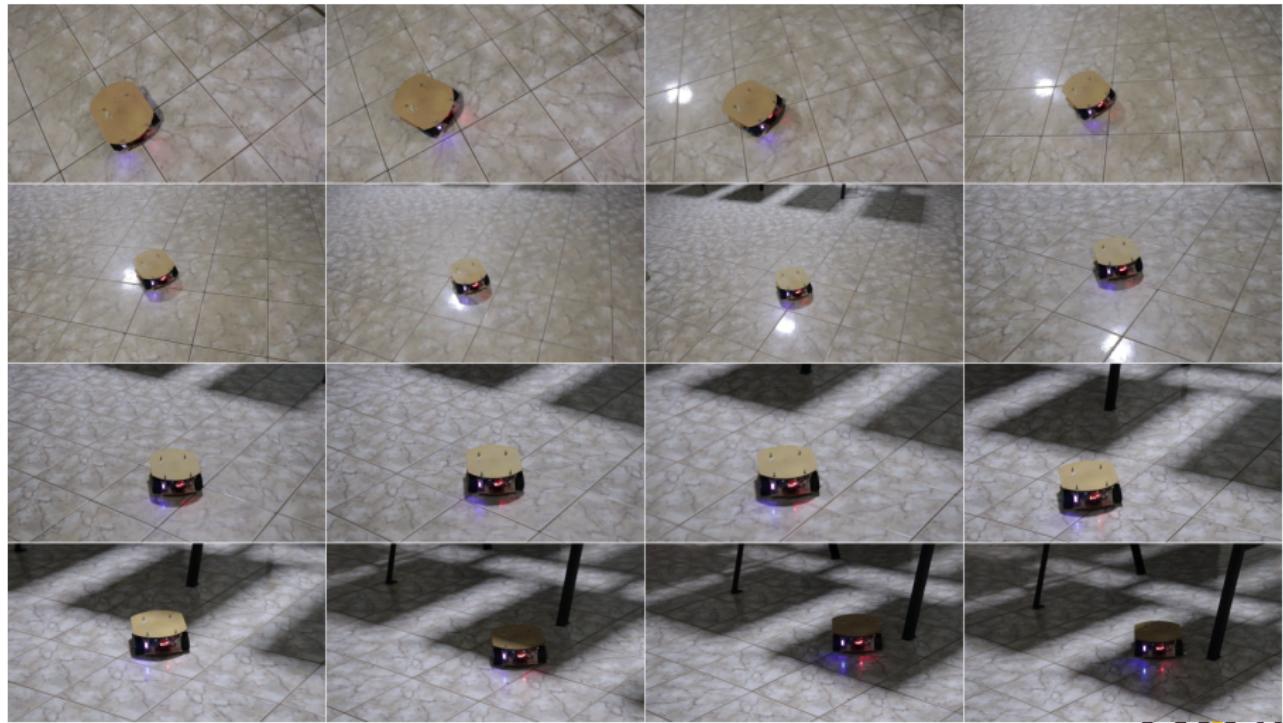
# Resultado para Simulação - Controlador Híbrido



# Resultado para Simulação - Controlador Fuzzy



# Resultado Para o Robô - Controlador Híbrido



# Resultado Para o Robô - Controlador Híbrido



# Resultado Para o Robô - Controlador Híbrido



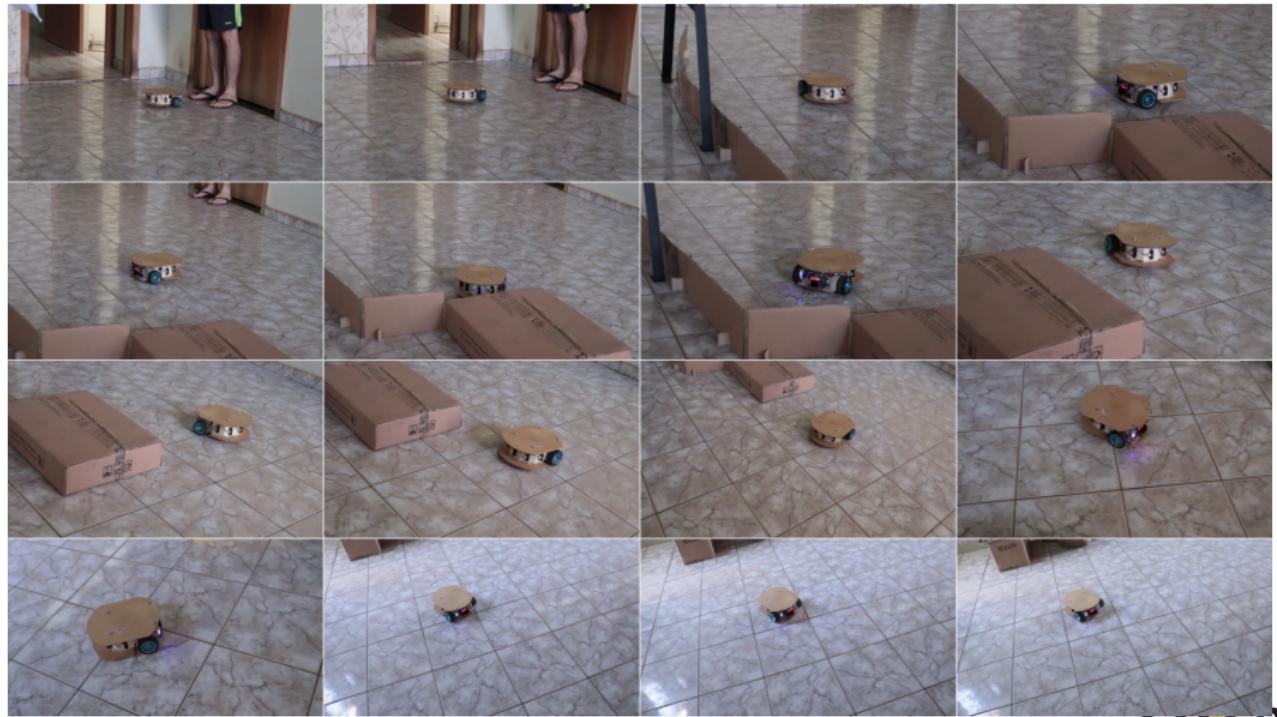
# Resultado Para o Robô - Controlador Híbrido



# Resultado Para o Robô - Controlador Híbrido



# Resultado Para o Robô - Controlador Fuzzy



# Considerações Finais e Trabalhos Futuros

- Arbitragem X Fusão

# Considerações Finais e Trabalhos Futuros

- Arbitragem X Fusão
- Infraestrutura

**Universidade Tecnológica Federal do Paraná**  
Engenharia de Computação  
*Campus Pato Branco*

**Trabalho de Conclusão de Curso - 2021/2**  
Navegação em robôs móveis por Arbitragem e Fusão em Arquiteturas  
Comportamentais

Profa. Orientadora: Dra. Kathya Silvia Collazos Linares  
Acadêmico: Marcelo Gervazoni Carbonera

20 de Setembro de 2021