

# Atividade 3

## Robótica Comportamental

Aluno: Mateus Soares Marques  
Disciplina: Automação Inteligente  
Professor: Antonio Marcus Nogueira Lima

# Objetivos

- Implementar a solução original proposta no artigo de referência indicado;
- Adaptar a solução original para o Pioneer3DX no CoppeliaSim;
- Alterar e/ou aumentar o número de funções de pertencimento do controlador fuzzy;
- Adicionar os demais sensores do Pioneer3DX na solução;
- Testar o sistema com o modelo de inferência de Sugeno.

# Roteiro

- Introdução
- Desenvolvimento
- Resultados
- Análise e Conclusão

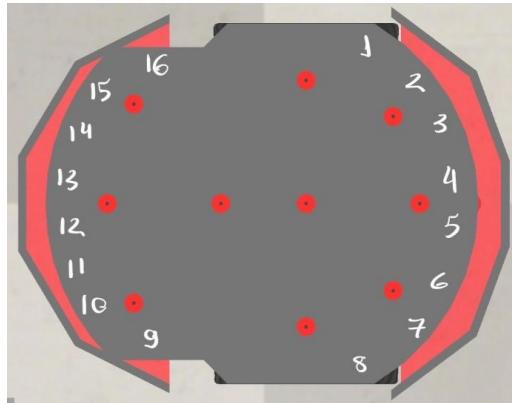
# Introdução

## Navegação em ambiente desconhecido

- O robô navega em uma área de trabalho de dimensões conhecidas;
- As coordenadas instantâneas do robô ou a distância e orientação relativa do alvo são conhecidas;
- Não se tem informações prévias sobre a presença de obstáculos na área de trabalho;
- A distância entre o robô e possíveis obstáculos é medida através de sensores acoplados ao chassi.

# Introdução

## Abordagem original



- Os sensores 4 e 5 são usados para medir a distância frontal
- Os sensores 1 e 16 medem a distância à esquerda
- Os sensores 8 e 9 medem a distância à direita
- O valor de distância considerado é o menor entre as leituras do par em cada direção.

# Introdução

## Abordagem Original

- Se não há obstáculos, mova-se em direção ao alvo, com velocidade proporcional a distância ao alvo;
- Se há obstáculo à frente, reduza a velocidade;
- Se há obstáculo à frente esquerda (direita), vire à direita (esquerda);
- Se há não há obstáculo à frente, há obstáculo à esquerda (direita) e o alvo está à esquerda (direita), siga em frente com velocidade média;
- Se há obstáculo à esquerda (direita) e o alvo está à direita (esquerda) vire em direção ao alvo;
- Se há obstáculo à frente, esquerda e direita, diminua a velocidade ao mínimo e vire até não haver mais obstáculos à frente.

# Introdução

## Abordagem Proposta

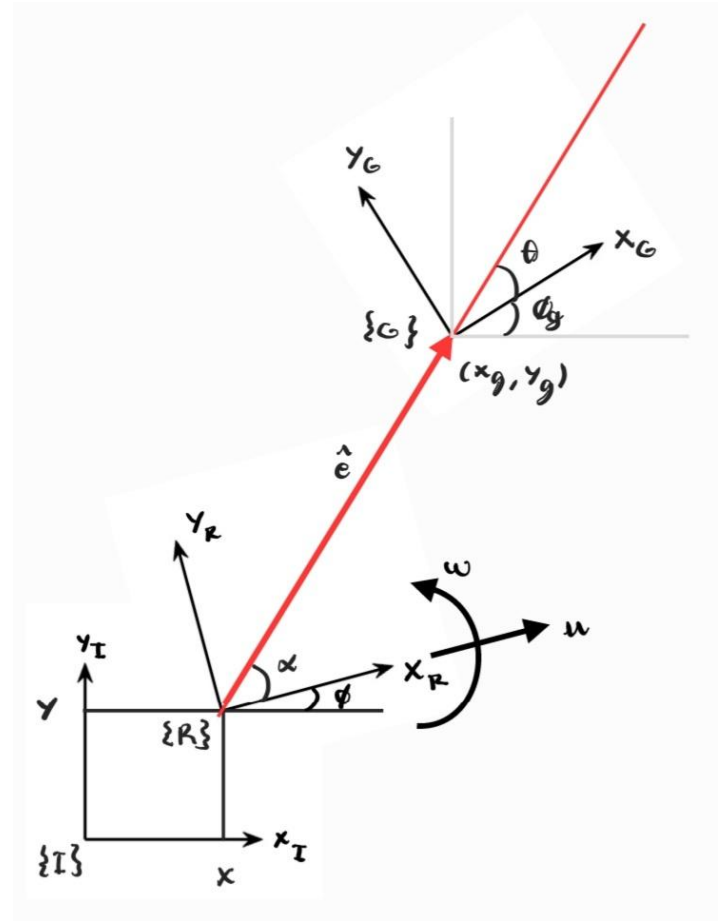
- Um vetor “força de velocidade” é aplicado sobre o robô. O alvo puxa o robô enquanto obstáculos o repelem. Se um obstáculo é percebido à direita, uma força empurra o robô para esquerda e vice-versa. Obstáculos à frente empurram o robô para trás.
- No caso de seguimento de parede, a presença de um obstáculo lateral na mesma direção do objetivo gera uma força na direção norte do referencial do robô.
- Para tal abordagem, o controlador de comportamento é separado do controlador de velocidade. O controlador de comportamento gera a “força” resultante, que é passada ao controlador de velocidade para gerar as velocidades linear e angular do robô.

# Introdução

## Abordagem Proposta

- A “força” mencionada é um vetor de erro que alimenta um controlador não-linear projetado através do método de Lyapunov. As entradas do controlador são a magnitude e o argumento desse vetor no referencial do robô:

$$\begin{cases} u = \gamma e \cos \alpha \\ \omega = k\alpha + \gamma \cos \alpha \sin \alpha \end{cases}$$





# Desenvolvimento

## Abordagem original modificada

A adaptação da solução do artigo para o caso do P3DX mostrou-se uma tarefa difícil. Fazer das saídas do controlador nebuloso as velocidades linear e angular ( $V$  e  $W$ , respectivamente) levou a um projeto baseado na tentativa e erro, com resultados pouco robustos e muito dependentes dos parâmetros dos sensores e da disposição dos obstáculos na cena. Como forma de mitigar esses problemas, modificou-se a solução original fazendo das saídas vetores de erro (tal qual na abordagem proposta), que alimentam um controlador não-linear de velocidade.

A modificação não implicou em mudança do banco de regras, pois foi feita uma associação direta entre o que antes era  $W$  e o que agora é  $\alpha$  (argumento do vetor erro ou “força de velocidade”) e o que antes era  $V$  e o que agora é  $e$  (magnitude do vetor erro).

# Desenvolvimento

## Parâmetros

Sensores: 1 metros,  $30^\circ$ ;

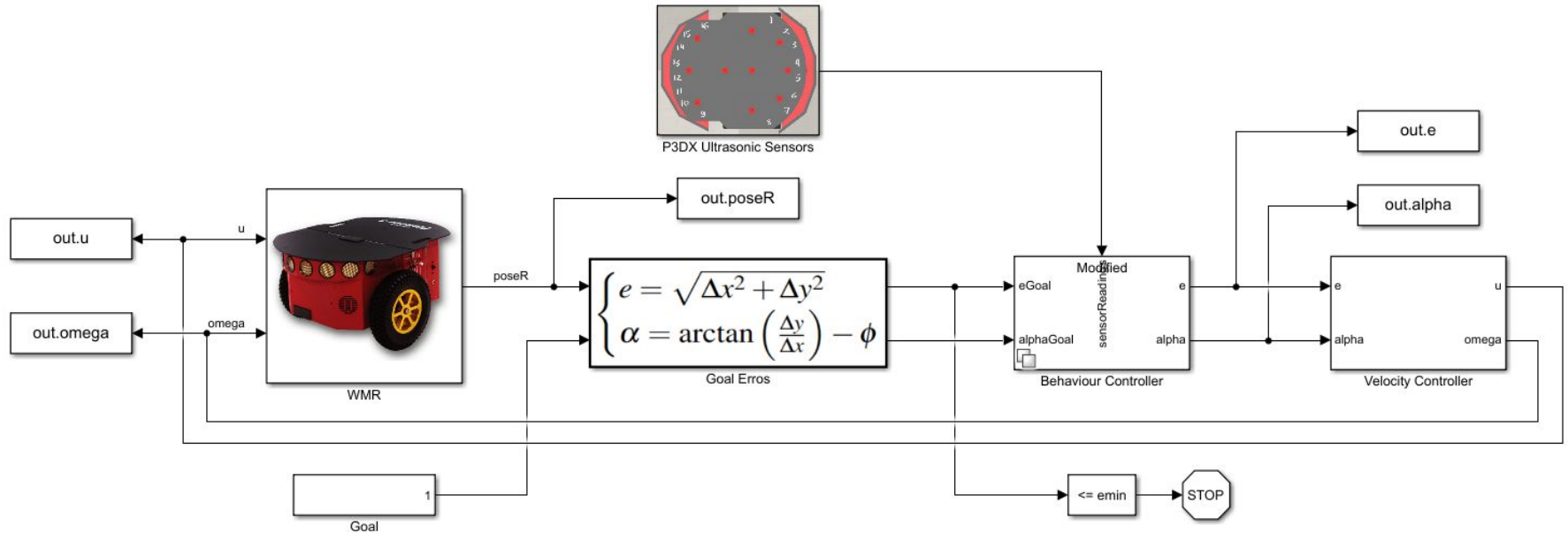
Velocidade linear máxima: 1 m/s;

Velocidade angular máxima:  $300^\circ/\text{s}$ ;

Controlador não-linear de velocidade:  $\gamma = 0.3$ ;  $k = 1$ .

# Desenvolvimento

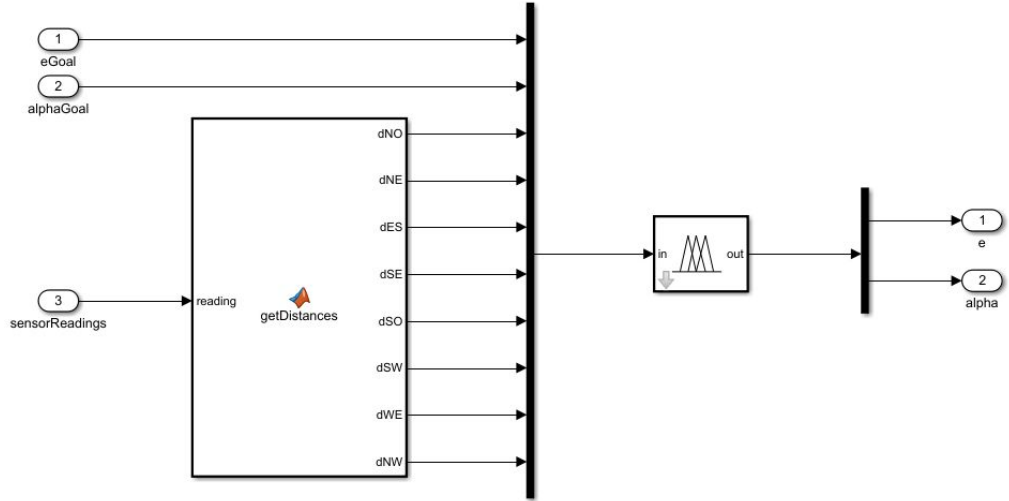
## Modelo geral do sistema



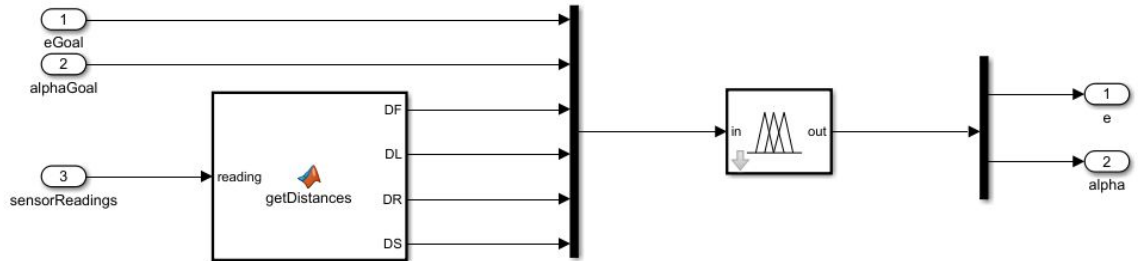
# Desenvolvimento

## Modelos dos controladores

Com todos os sensores



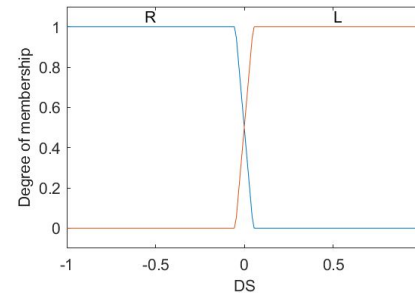
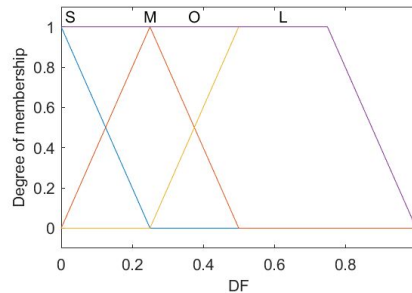
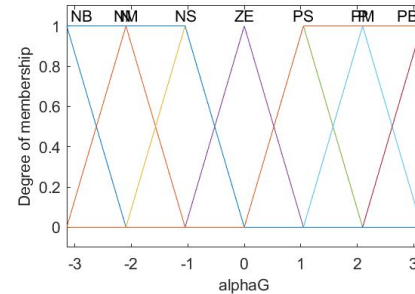
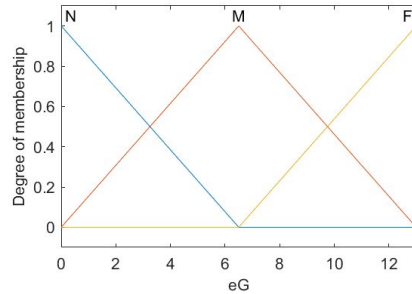
Apenas os originais



# Desenvolvimento

## Abordagem original modificada

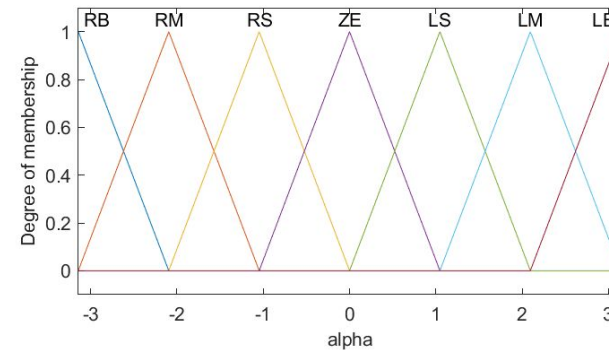
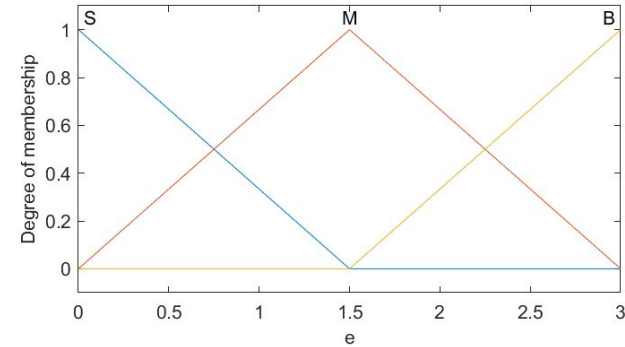
- Funções de pertencimento das entradas (DL e DR possuem apenas a função O de DF):



# Desenvolvimento

## Abordagem original modificada

- Funções de pertencimento das saídas



# Desenvolvimento

## Abordagem original modificada

- Regras

```
'1. (eG==N) & (DF~=O) => (e=S) (1)      '  
'2. (eG==M) & (DF~=O) => (e=M) (1)      '  
'3. (eG==F) & (DF~=O) => (e=B) (1)      '  
'4. (alphaG==NB) & (DF~=O) & (DR~=O) => (alpha=RB) (1) '  
'5. (alphaG==NM) & (DF~=O) & (DR~=O) => (alpha=RM) (1) '  
'6. (alphaG==NS) & (DF~=O) & (DR~=O) => (alpha=RS) (1) '  
'7. (alphaG==ZE) & (DF~=O) => (alpha=ZE) (1) '  
'8. (alphaG==PS) & (DF~=O) & (DL~=O) => (alpha=LS) (1) '  
'9. (alphaG==PM) & (DF~=O) & (DL~=O) => (alpha=LM) (1) '  
'10. (alphaG==PB) & (DF~=O) & (DL~=O) => (alpha=LB) (1) '  
'11. (DF==S) & (DS==L) => (e=S) (alpha=RB) (1) '  
'12. (DF==M) & (DS==L) => (e=M) (alpha=RM) (1) '  
'13. (DF==L) & (DS==L) => (e=B) (alpha=RS) (1) '  
'14. (DF==S) & (DS==R) => (e=S) (alpha=LB) (1) '  
'15. (DF==M) & (DS==R) => (e=M) (alpha=LM) (1) '  
'16. (DF==L) & (DS==R) => (e=B) (alpha=LS) (1) '  
'17. (alphaG==N) & (DF~=O) & (DR==O) => (e=M) (alpha=ZE) (1) '  
'18. (alphaG~=N) & (DF~=O) & (DL==O) => (e=M) (alpha=ZE) (1) '  
'19. (DF~=O) & (DL==S) => (e=M) (alpha=RS) (1) '  
'20. (DF~=O) & (DR==S) => (e=M) (alpha=LS) (1)      '
```

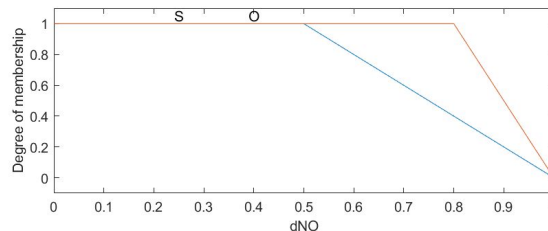
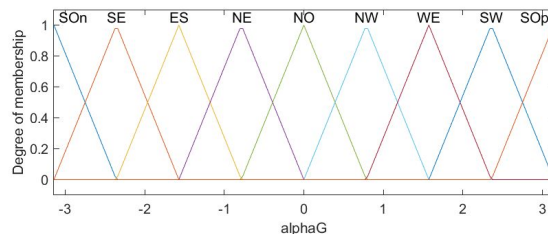
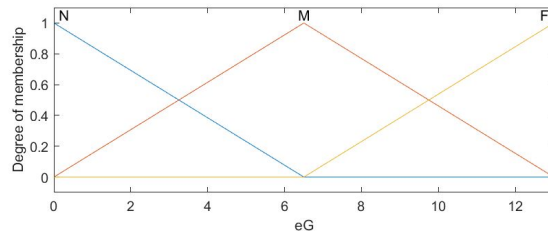
# Desenvolvimento

## Abordagem Proposta

- Funções de pertencimento das entradas:

Todos os sensores foram utilizados. Os 8 pares disponíveis nos dão distâncias nas 8 direções da rosa dos ventos.

As entradas das 7 direções que faltam na imagem possuem as mesmas funções S e O da entrada dNO (norte). S indica uma distância pequena e O a presença de obstáculo.

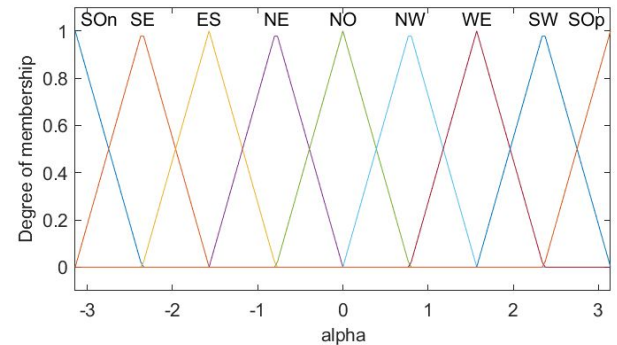
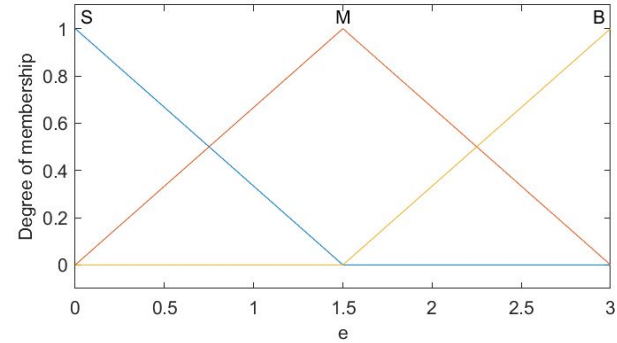




# Desenvolvimento

## Abordagem Proposta

- Funções de pertencimento das saídas



# Desenvolvimento

## Abordagem Proposta

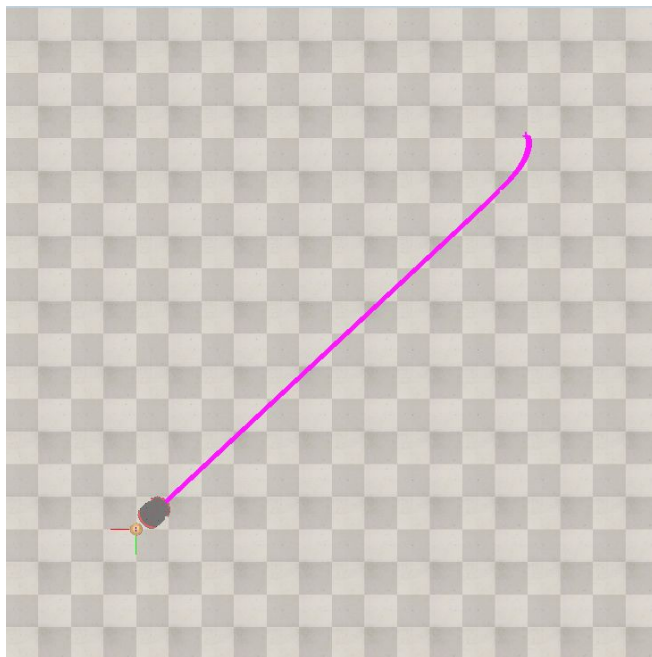
- Regras

```
'1. (eG==N) & (dNO~~S) & (dSO~~S) => (e=S) (1)
'2. (eG==M) & (dNO~~S) & (dSO~~S) => (e=M) (1)
'3. (eG==F) & (dNO~~S) & (dSO~~S) => (e=B) (1)
'4. (alphaG==Son) & (dNO~~S) & (dSO~~S) => (alpha=Son) (1)
'5. (alphaG==SE) & (dNO~~S) & (dSO~~S) => (alpha=SE) (1)
'6. (alphaG==ES) & (dNO~~S) & (dSO~~S) => (alpha=ES) (1)
'7. (alphaG==NE) & (dNO~~S) & (dSO~~S) => (alpha=NE) (1)
'8. (alphaG==NO) & (dNO~~S) & (dSO~~S) => (alpha=NO) (1)
'9. (alphaG==NW) & (dNO~~S) & (dSO~~S) => (alpha=NW) (1)
'10. (alphaG==WE) & (dNO~~S) & (dSO~~S) => (alpha=WE) (1)
'11. (alphaG==SW) & (dNO~~S) & (dSO~~S) => (alpha=SW) (1)
'12. (alphaG==SOp) & (dNO~~S) & (dSO~~S) => (alpha=SOp) (1)
'13. (dNO==S) | (dNE==S) | (dSE==S) | (dSO==S) | (dSW==S) | (dNW==S) => (e=S) (1)
'14. (dNE==S) => (alpha=SW) (0.8)
'15. (dES==S) => (alpha=WE) (0.5)
'16. (dSE==S) => (alpha=NW) (0.8)
'17. (dSO==S) => (alpha=NO) (1)
'18. (dSW==S) => (alpha=NE) (0.8)
'19. (dWE==S) => (alpha=ES) (0.5)
'20. (dNW==S) => (alpha=SE) (0.8)
'21. (dNO==S) & (dES==O) => (alpha=SOp) (1)
'22. (dNO==S) & (dWE==O) => (alpha=Son) (1)
'23. (dNO==S) & (dES~~O) & (dWE~~O) => (alpha=Son) (1)
'24. (alphaG==SE) & (dNO~~S) & (dNE==S) & (dES==S) & (dSE==S) => (e=M) (alpha=NO) (1)
'25. (alphaG==ES) & (dNO~~S) & (dNE==S) & (dES==S) & (dSE==S) => (e=M) (alpha=NO) (1)
'26. (alphaG==NE) & (dNO~~S) & (dNE==S) & (dES==S) & (dSE==S) => (e=M) (alpha=NO) (1)
'27. (alphaG==NW) & (dNO~~S) & (dSW==S) & (dWE==S) & (dNW==S) => (e=M) (alpha=NO) (1)
'28. (alphaG==WE) & (dNO~~S) & (dSW==S) & (dWE==S) & (dNW==S) => (e=M) (alpha=NO) (1)
'29. (alphaG==SW) & (dNO~~S) & (dSW==S) & (dWE==S) & (dNW==S) => (e=M) (alpha=NO) (1)
'
```

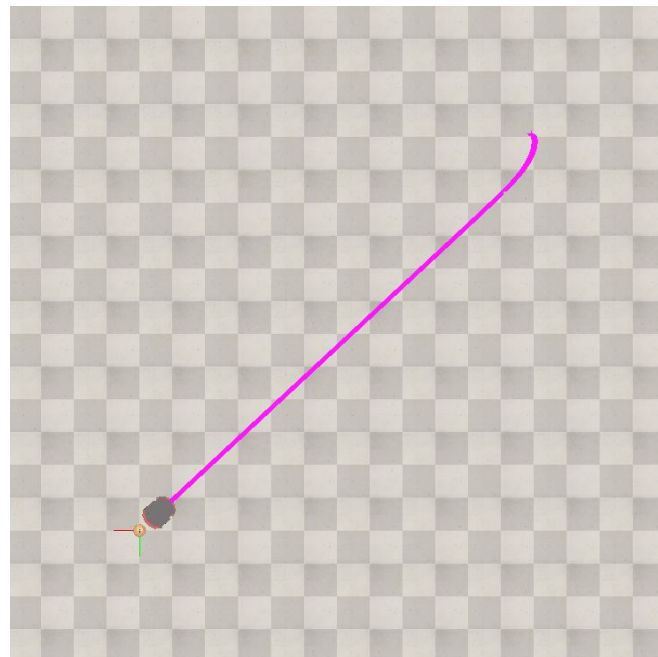
# Resultados

## Cenário 1

Original



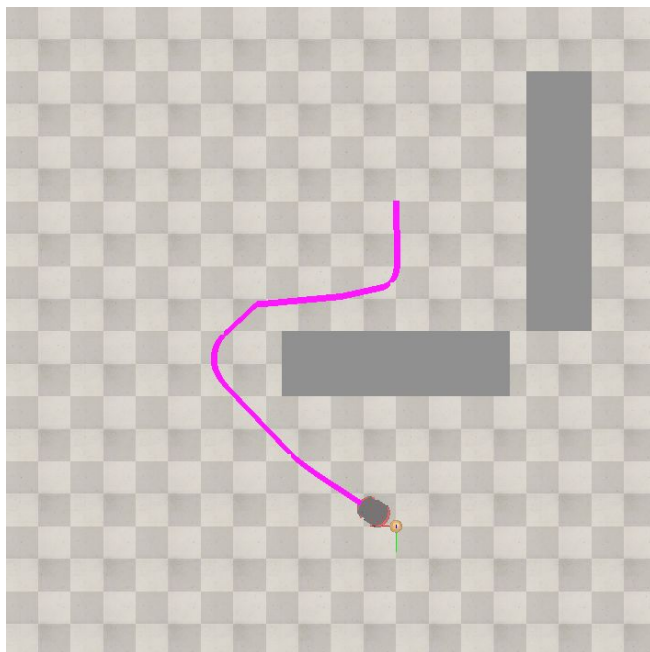
Proposto



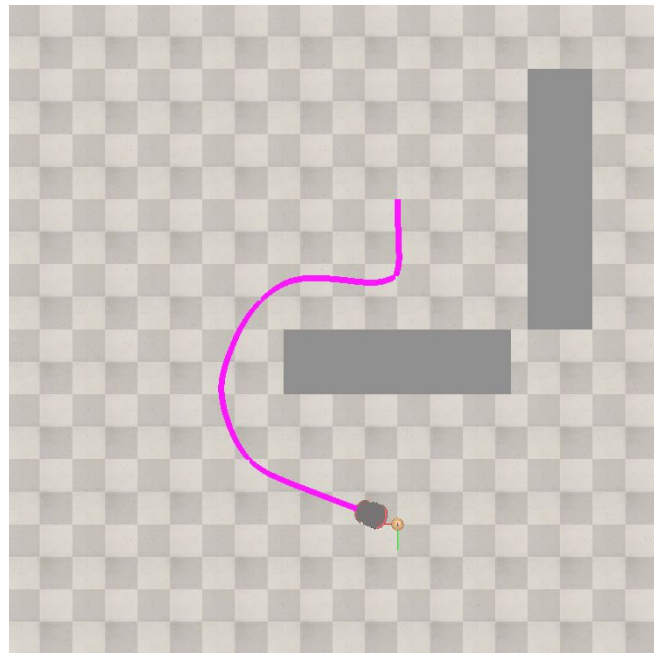
# Resultados

## Cenário 2

Original



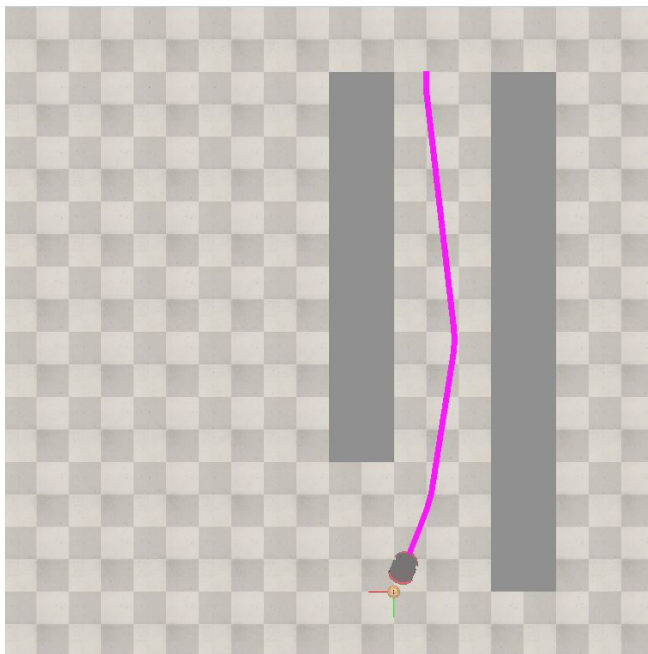
Proposto



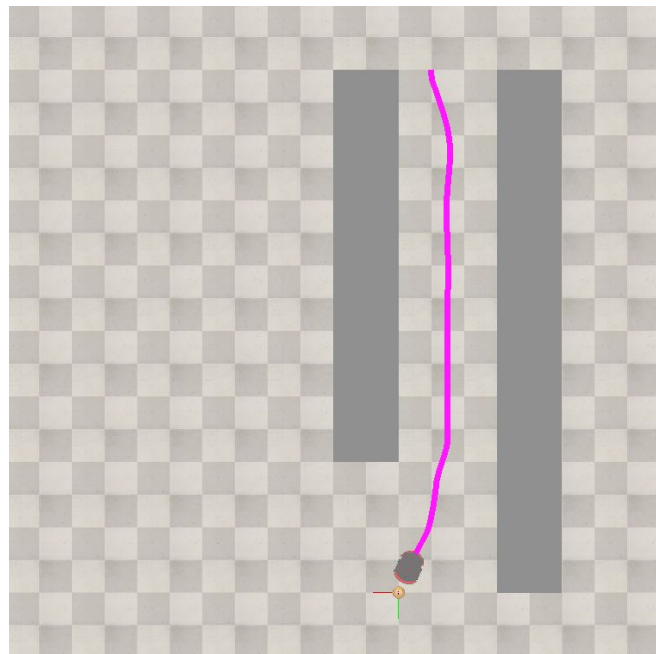
# Resultados

## Cenário 3

Original



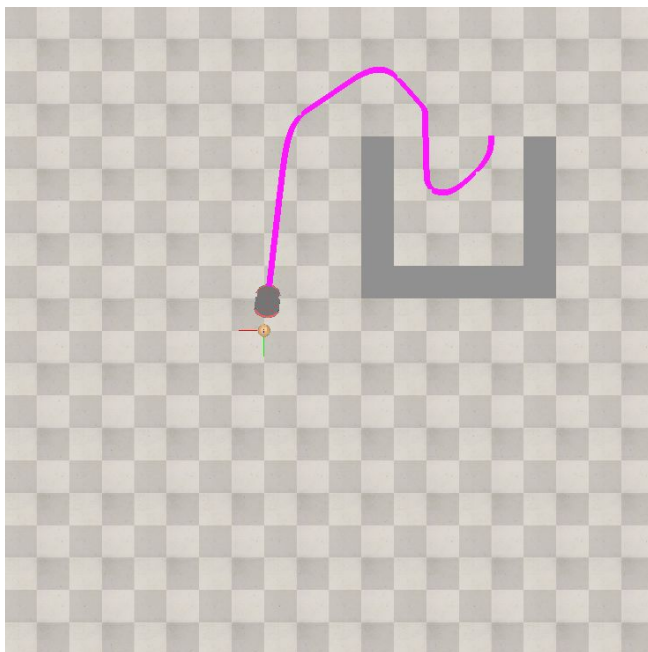
Proposto



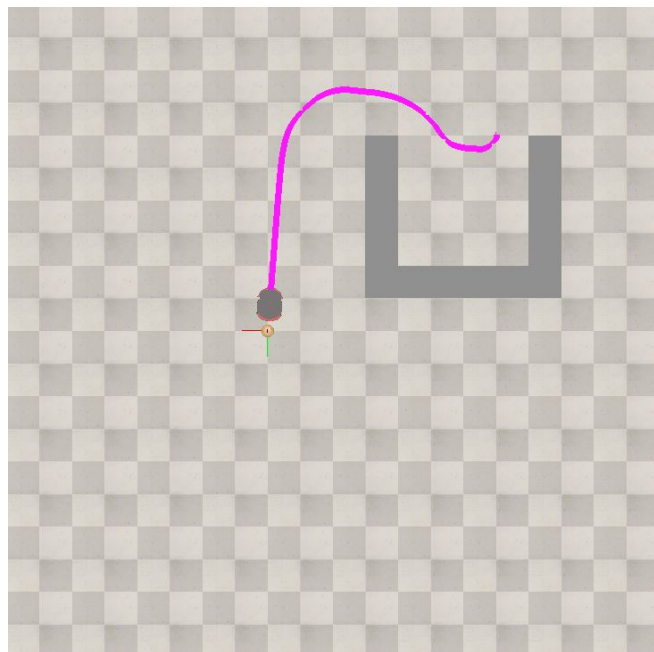
# Resultados

## Cenário 4

Original



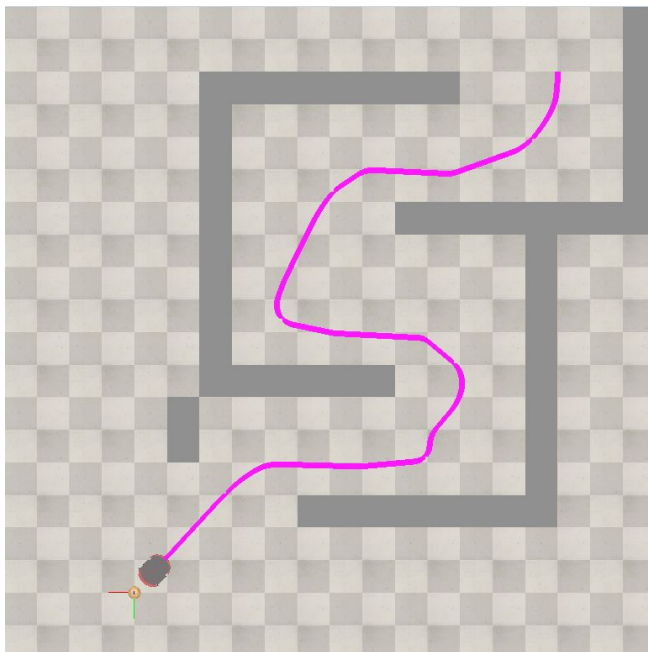
Proposto



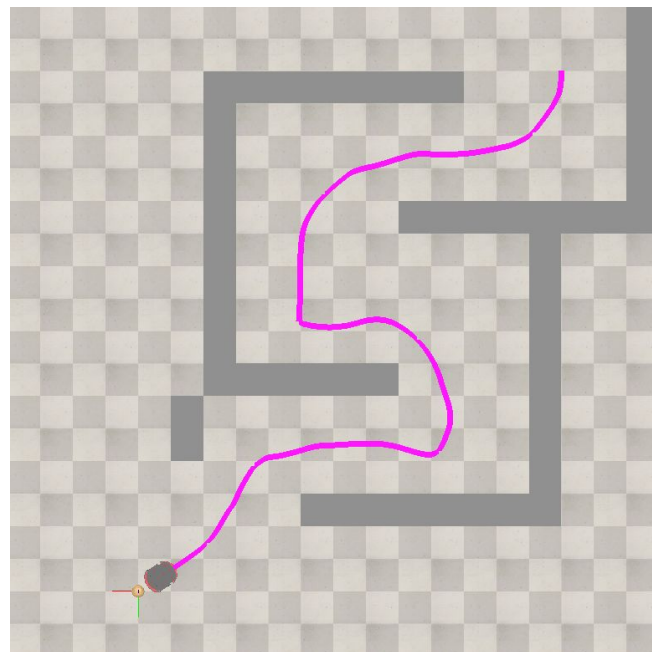
# Resultados

## Cenário 5

Original



Proposto





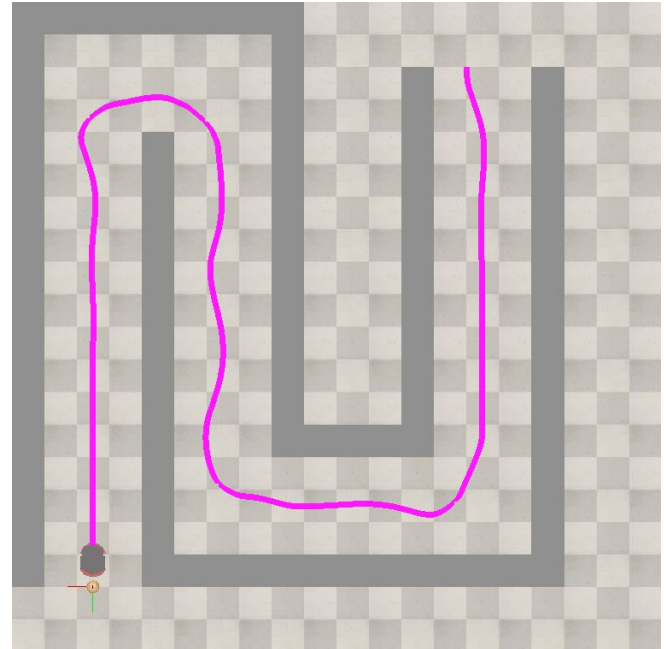
# Resultados

## Cenário 6

Original

Não completou a tarefa

Proposto





# Análise e Conclusão

- A inclusão de um controlador de velocidade facilitou a sintonia do controlador nebuloso e proporcionou uma resposta mais robusta;
- O projeto do sistema de navegação se tornou menos problemático para o P3DX uma vez que o controlador nebuloso foi utilizado para tratar de todos os comportamentos. A soma disso com o controle de velocidade deram ao movimento do robô maior suavidade;
- Uma maior quantidade de sensores é mais facilmente tratada na abordagem proposta, uma vez que a construção de um banco de regras se resume a gerar vetores de erro em oposição ao sentido de detecção do obstáculo para cada distância medida, e um vetor no sentido do objetivo. O seguimento de parede é obtido gerando-se um vetor paralelo ao obstáculo lateral.