

Atividade 3 - Robótica Comportamental

Mateus Soares Marques

Abstract—Este trabalho expõe os resultados obtidos da implementação da arquitetura baseada em comportamento para navegação de robôs móveis em ambiente desconhecido apresentada no artigo [1]. A solução foi implementada através do MATLAB/Simulink conectados ao simulador de robôs CoppeliaSim, onde o robô considerado foi o PioneerP3DX. Para adaptar o projeto original ao caso específico do PioneerP3DX, um controlador de velocidade foi inserido na arquitetura para interfacear o árbitro de comportamento com o robô. Os comportamentos foram modelados como regras de um único controlador fuzzy, em substituição aos controladores individuais para cada comportamento. Por fim, uma abordagem que utiliza de todos os sensores disponíveis no modelo do P3DX no CoppeliaSim foi proposta e comparada com arquitetura original.

I. INTRODUÇÃO

O problema da navegação do robô móvel em ambiente desconhecido consiste em levar o robô, cujas coordenadas instantâneas são conhecidas, à um alvo, também de coordenadas conhecidas, através de uma área de trabalho sobre a qual não se tem conhecimento prévio da disposição de possíveis obstáculos. Para desviar de obstáculos e completar a tarefa de alcançar o ponto alvo, o robô dispõe de sensores de distância.

A. Abordagem original modificada

A abordagem original apresentada por [1] define quatro possíveis situações envolvendo obstáculos detectados ou não que são respondidas por quatro respectivos comportamentos:

- Ir ao alvo: o robô se move em direção ao alvo caso não haja obstáculos detectados por seus sensores;
- Evitar obstáculo: o robô desvia de obstáculos detectados à frente, à frente e esquerda ou à frente e direita;
- Seguir parede: o robô segue uma parede lateral caso não haja obstáculos à frente e o alvo esteja na mesma direção da parede encontrada;
- Sair de trancamento: o robô realiza uma manobra de forma a sair de uma situação onde é cercado por obstáculos em todas as direções.

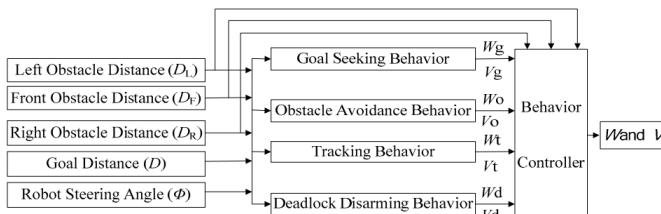


Fig. 1. Arquitetura do controle do robô.

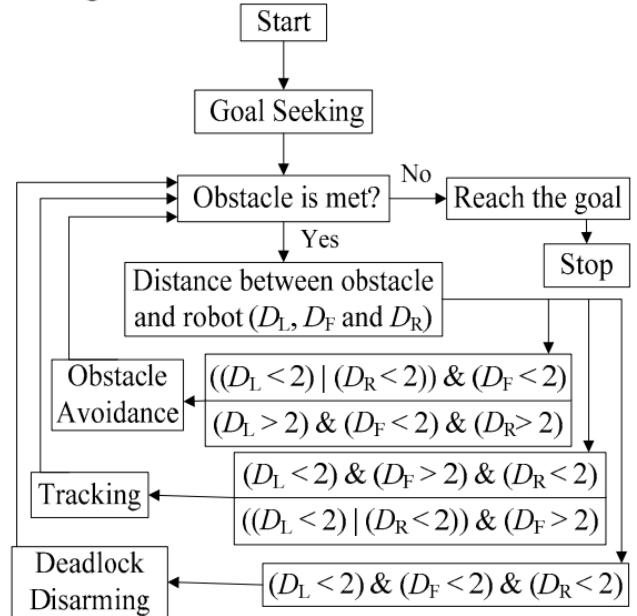


Fig. 2. Controlador de comportamento.

A arbitragem dos comportamentos é realizada por uma estrutura de decisão que compara as distâncias medidas em cada direção considerada (no caso do artigo, esquerda, direita e frente) com um limiar (dois metros em [1]) e decide se as medidas correspondem a um determinado comportamento dentre os apresentados. A arquitetura geral do sistema original é apresentada na figura 1. A arquitetura do controlador ou árbitro de comportamento é apresentada na figura 2.

Os comportamentos são modelados individualmente por controladores nebulosos. As entradas dos controladores fuzzy são as distâncias medidas com os sensores, o erro de distância e o erro de orientação do robô em relação ao alvo, como mostrado na figura 1. As saídas são a velocidade linear e o ângulo de guinada do robô, dados por V e W , respectivamente.

Tentar reproduzir o projeto descrito no PioneerP3DX com total fidelidade ao apresentado no artigo original é uma tarefa difícil. Isso porque com o PioneerP3DX, que é um robô de tração diferencial, determinar o ângulo de guinada como saída envolveria controlar o ângulo com que o robô faz curvas, um vez que, diferente de um veículo de Ackerman, por exemplo, o P3DX não apresenta juntas para controle da orientação das rodas.

Para adaptação do projeto original ao P3DX, pode-se tratar a saída W não como ângulo de guinada, mas como velocidade angular. No entanto, tentativas iniciais de fazer

tal paralelo levaram a resultados pouco consistentes e sem a robustez esperada. Por fim, decidiu-se por interfacear o controlador de comportamento com as velocidades do robô através de um controlador não-linear de velocidade, projetado à partir da teoria de Lyapunov [2]. A lei de controle desse controlador é dada por

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

onde u e ω são as velocidades linear e angular do robô, respectivamente, γ e k são os ganhos do controlador de velocidade e e e α são a magnitude e orientação do vetor erro \hat{e} passado como entrada ao controlador de velocidade pelo controlador de comportamento. Esse vetor é definido no sistema de coordenadas do robô, como mostrado na figura 3

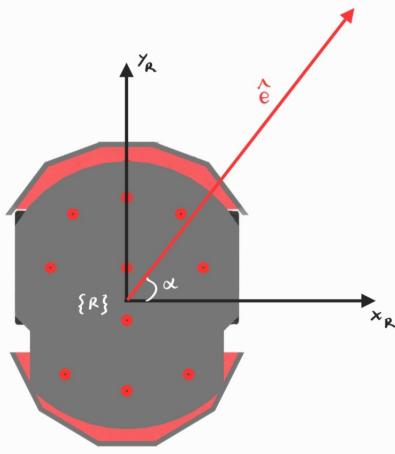


Fig. 3. Votor erro ou "força de velocidade" gerado pelo controlador de comportamento como entrada para o controlador de velocidade.

Pode-se então fazer o paralelo do que antes era V e W com e e α .

Outra modificação realizada, foi a substituição dos controladores fuzzy para os comportamentos individuais por um único controlador nebuloso que agrupa todas as regras de cada comportamento em um único sistema de inferência.

Para reproduzir o projeto original, trabalhou-se apenas com os sensores apontados para frente, esquerda e direita do robô. Considerando a figura 4 que enumera os sensores disponíveis no P3DX, os sensores para cada distância são

- Distância Frontal (D_F): sensores 4 e 5;
- Distância à Esquerda (D_L): sensores 1 e 16;
- Distância à Direita (D_R): sensores 8 e 9.

Portanto, os valores de D_F , D_L e D_R são dados por

$$D_F = \min(d_4, d_5)$$

$$D_L = \min(d_1, d_{16})$$

$$D_R = \min(d_8, d_9)$$

onde d_i indica a distância medida pelo i -ésimo sensor.

Dadas as distâncias e o erro de distância e orientação do alvo (e_G e α_G), o banco de regras do controlador fuzzy projetado pode ser resumido em

- Se D_F é grande (maior do que o limiar de detecção) e não há obstáculos na mesma direção do alvo, o vetor \hat{e} na direção α_G e tem magnitude proporcional à e_G ;
- Se D_F é menor que o limiar de detecção, a magnitude de \hat{e} é proporcional a D_F ;
- Se D_F e D_L (mas não D_R) são menores que o limiar de detecção, \hat{e} tem orientação para direita, sendo α tão à direita quanto menor for D_F ;
- Se D_F e D_R (mas não D_L) são menores que o limiar de detecção, \hat{e} tem orientação para esquerda, sendo α tão à esquerda quanto menor for D_F ;
- Se D_F (mas nem D_L nem D_R) é menor que o limiar de detecção, \hat{e} aponta para esquerda, se o alvo está à esquerda ou para direita, se o alvo está à direita, sendo mod α tão grande quanto menor for D_F . Estando o alvo à frente, α aponta para direita;
- Se D_L ou D_R (mas não D_F) é menor que o limiar e o alvo está, respectivamente, à esquerda ou à direita, \hat{e} aponta para frente e tem magnitude média;
- Se D_F , D_L e D_R são menores que o limiar, o vetor \hat{e} aponta para trás e tem viés à direita ou à esquerda para o alvo à direita ou à esquerda, respectivamente. Se o alvo está à frente, o viés é à direita.

Os demais detalhes da implementação, como as funções de pertencimento das entradas e saídas, a listagem de todas as regras e seus pesos e os ganhos do controlador de velocidade seguem na seção Desenvolvimento.

B. Abordagem proposta

Para inclusão dos demais sensores do P3DX e simplificação do banco de regras, uma nova abordagem que faz proveito do controlador de velocidade incluído é apresentada.

Nessa nova abordagem, se considera que a presença de um obstáculo gera uma "força de velocidade" no sentido oposto ao da detecção, repelindo o robô de obstáculos enquanto uma força na direção do alvo o atrai. Essa chamada "força de velocidade" é uma re-significação do vetor \hat{e} produzido pelo controlador de comportamento e fornecido ao controlador de velocidade. As figuras 5 e 6 ilustram as ações atratoras e repulsoras do alvo e dos obstáculos sobre o robô.

O caso de seguimento de parede, quando há um obstáculo detectado no mesmo lado do robô onde está o alvo e não há obstáculos à frente, é tratado de maneira diferente. Nesse caso uma força é aplicada direção norte, fazendo o robô

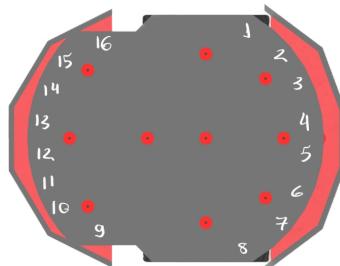


Fig. 4. Localização e índice de cada sensor disponível no modelo do P3DX no Coppelia.

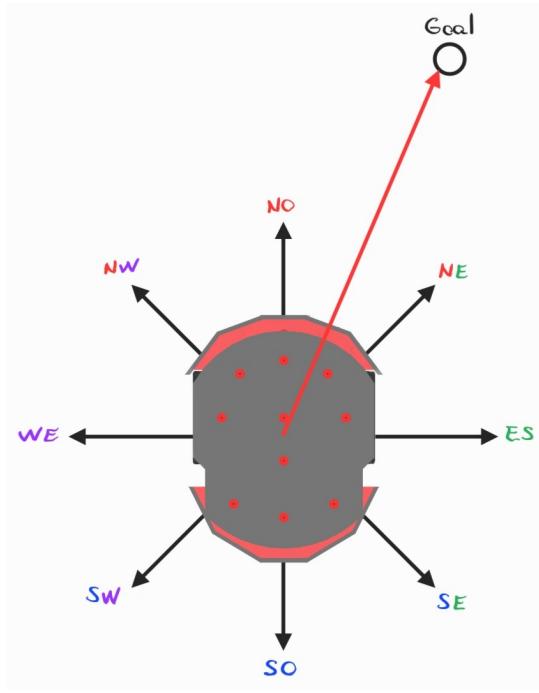


Fig. 5. Ação atradora do alvo sobre o robô. A seta em vermelho indica o vetor "força de velocidade" \hat{e}

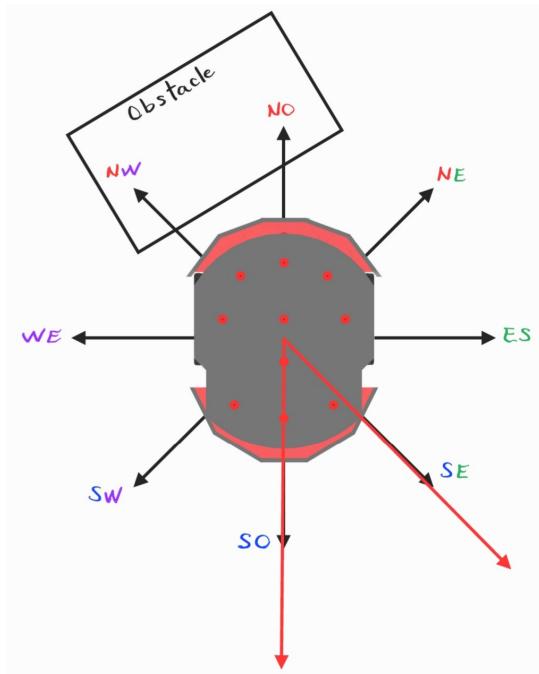


Fig. 6. Ação repulsora dos obstáculos sobre o robô. As setas em vermelho indicam a ação gerada pelas distâncias medidas nas direções Norte e Noroeste.

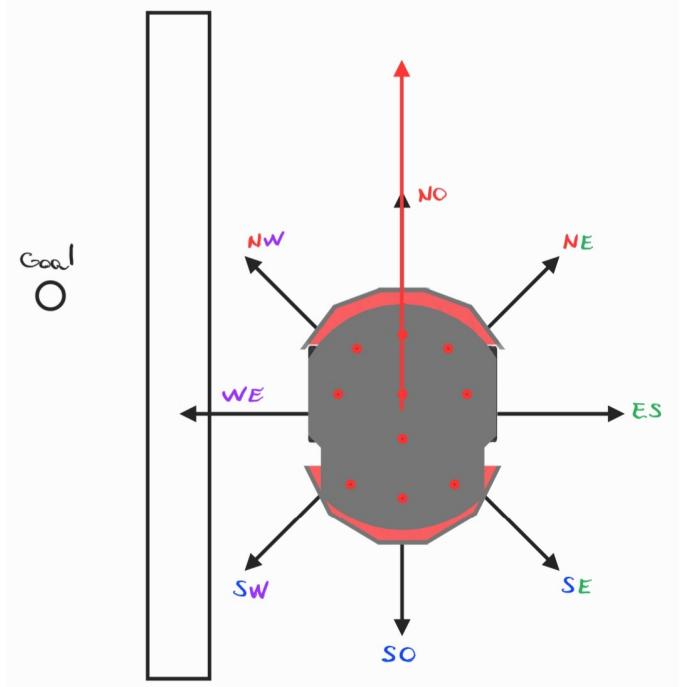


Fig. 7. Caso do seguimento de parede na abordagem proposta. A seta em vermelho indica o vetor "força de velocidade" aplicado sobre o robô.

acompanhar a parede. Esse caso especial é ilustrado na figura 7.

Na abordagem proposta, se considerou as direções para as quais apontam os 8 pares de sensores de proximidade disponíveis no P3DX do CoppeliaSim, sendo elas os quatro pontos cardinais e os quatro pontos colaterais no referencial do robô. As distâncias em cada direção são dadas por

$$\begin{aligned}d_{NO} &= \min(d_4, d_5) \\d_{NE} &= \min(d_6, d_7) \\d_{ES} &= \min(d_8, d_9) \\d_{SE} &= \min(d_{10}, d_{11}) \\d_{SO} &= \min(d_{12}, d_{13}) \\d_{SW} &= \min(d_{14}, d_{15}) \\d_{WE} &= \min(d_{16}, d_1) \\d_{NW} &= \min(d_2, d_3)\end{aligned}$$

As regras que do sistema de inferência fuzzy projetado podem ser resumidas em

- Se não há obstáculos à frente ou atrás, o vetor \hat{e} é aplicado na direção do alvo com magnitude proporcional a distância ao alvo;
- Se um obstáculo é detectado em uma direção, um vetor \hat{e} é aplicado no sentido oposto com magnitude proporcional a distância medida;
- Se um obstáculo é detectado numa direção lateral (ES ou WE) que contem o alvo, o vetor \hat{e} é aplicado na direção NO e tem magnitude média.

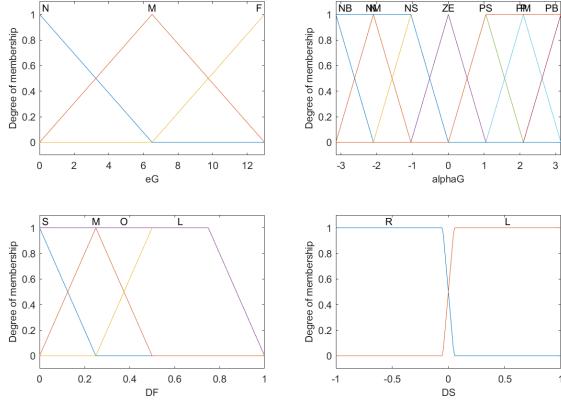


Fig. 8. Funções de pertencimento das entradas no caso de apenas três direções. As funções de DR foram omitida, uma vez que são as mesmas de DL. DS serve ao propósito de dar um viés ao desvio de obstáculos sem detecção lateral.

II. DESENVOLVIMENTO

Para o primeiro caso, que reproduz as regras e as direções consideradas no artigo de referência [1], as funções de pertencimento das entradas e saídas são dadas pelas figuras 8 e 9, respectivamente. O banco de regras é apresentado na figura 10.

Para o caso da abordagem proposta, as funções de pertencimento das entradas e saídas bem como o banco de regras constam nas imagens 11, 12 e 13.

Os parâmetros do controlador de velocidade escolhidos para as duas abordagens foram $\gamma = 0.3$ e $k = 1$. Já os parâmetros dos sensores são o padrão dos sensores do modelo do P3DX no Coppelia: 1 metro de alcance com 30° de abertura. As velocidades geradas são limitadas à 1.2 m/s e 300°/s, como indicado no manual do PioneerP3DX.

Um modelo para co-simulação Simulink-CoppeliaSim foi implementado no Simulink. A figura 16 traz o modelo geral do sistema, que contem as variantes necessárias para o caso de três direções e para o caso de oito direções.

III. RESULTADOS

Simulou-se os sistemas para os cenários considerados no artigo [1]. O resultados seguem nas imagens da figura 14. Um sexto cenário foi considerado, no entanto apenas na abordagem com todos os sensores o robô conseguiu completar a tarefa. Esse caso consta na figura 15

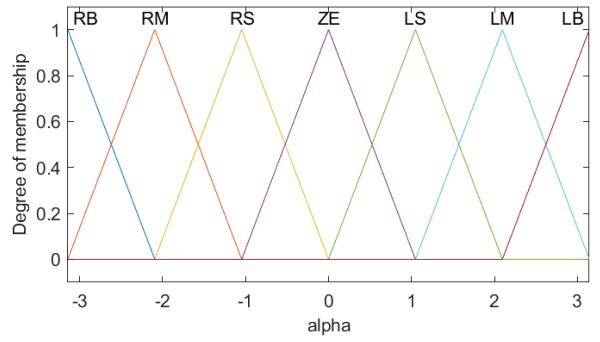
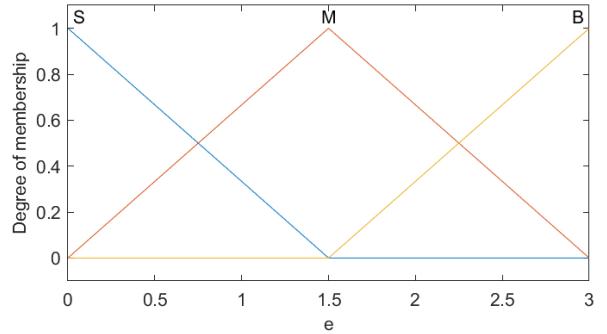


Fig. 9. Funções de pertencimento das saídas do controlador de comportamento da abordagem original com três direções.

```
'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)
```

Fig. 10. Banco de regras do controlador fuzzy da abordagem do artigo [1].

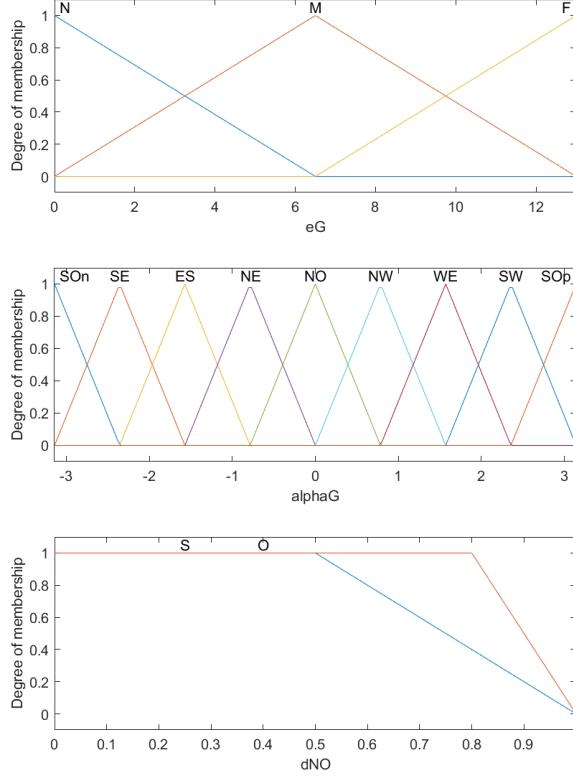


Fig. 11. Funções de pertencimento das entradas do controlador da abordagem proposta. Todas as direções possuem as mesmas funções de pertencimento, por isso apenas a direção Norte é apresentada.

IV. ANÁLISE E CONCLUSÕES

A comparação entre os resultados mostra que a inclusão de todos os sensores suavisa a trajetória do robô. O uso de um controlador de velocidade facilitou a sintonia do controlador de comportamento e o uso de um único controlador nebuloso que agrupa todos os comportamento mitigou problemas associados ao chaveamento excessivo em situações de conflito entre um comportamento e outro. Por fim, a abordagem que define a ação de um vetor "força de velocidade" simplificou o processo de definição das regras do controlador fuzzy.

REFERENCIAS

- [1] Q. -y. Bao, S. -m. Li, W. -y. Shang and M. -j. An, "A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments," 2009 International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, China, 2009, pp. 257-261, doi: 10.1109/AICL.2009.125.
- [2] M. Aicardi, G. Casalino, A. Bicchi and A. Balestrino, "Closed loop steering of unicycle like vehicles via Lyapunov techniques," in IEEE Robotics & Automation Magazine, vol. 2, no. 1, pp. 27-35, March 1995, doi: 10.1109/100.388294.

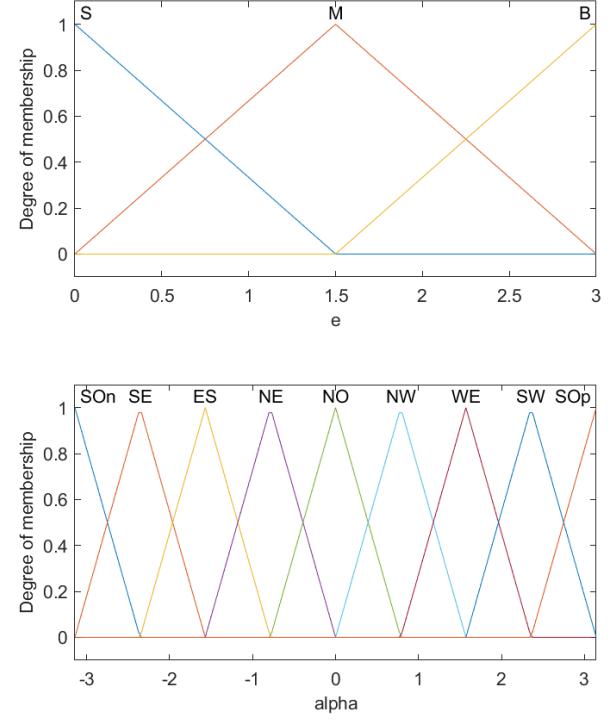


Fig. 12. Funções de pertencimento das saídas do controlador da abordagem proposta.

```

'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. (dNE==S) | (dNO==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=SOn) (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)

```

Fig. 13. Banco de regras do controlador da abordagem proposta.

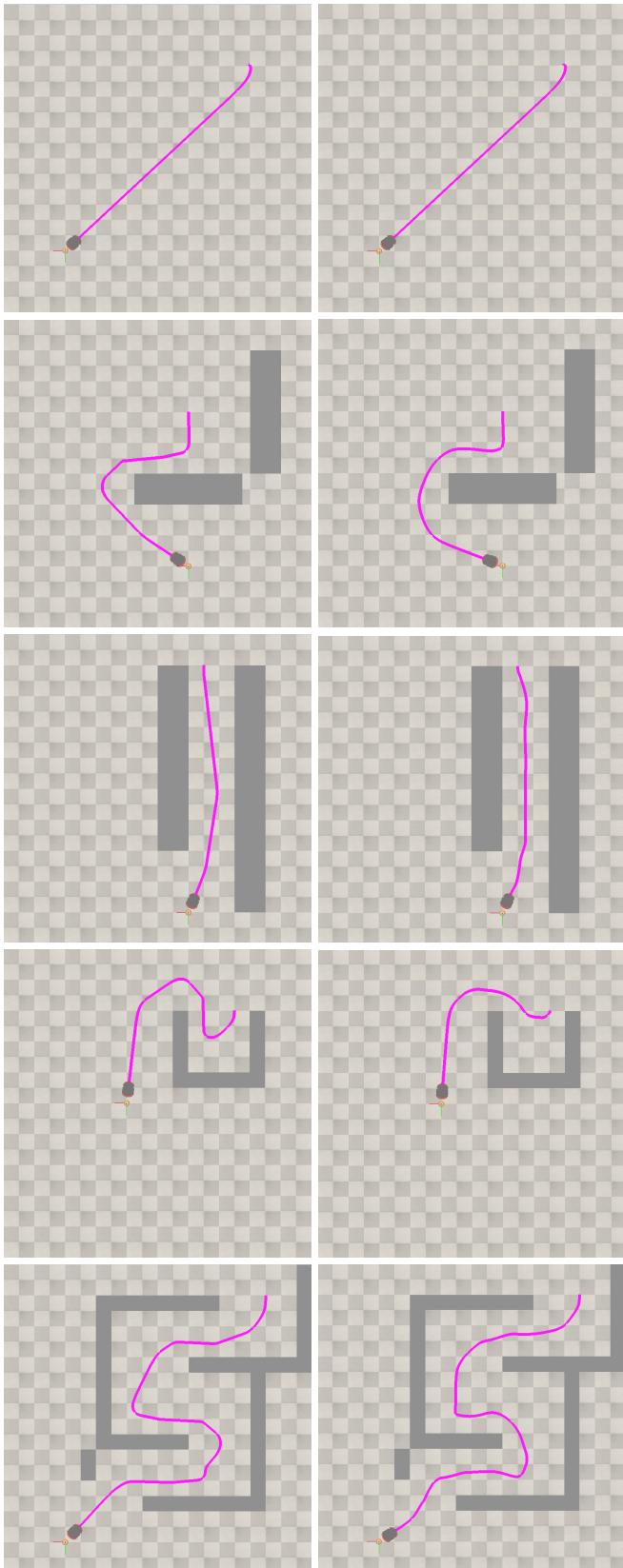


Fig. 14. Navegação do robô em diferentes cenários. As imagens da coluna à esquerda são para o caso de três direções sensoreadas. As imagens da coluna à direita são para a abordagem proposta com oito direções.

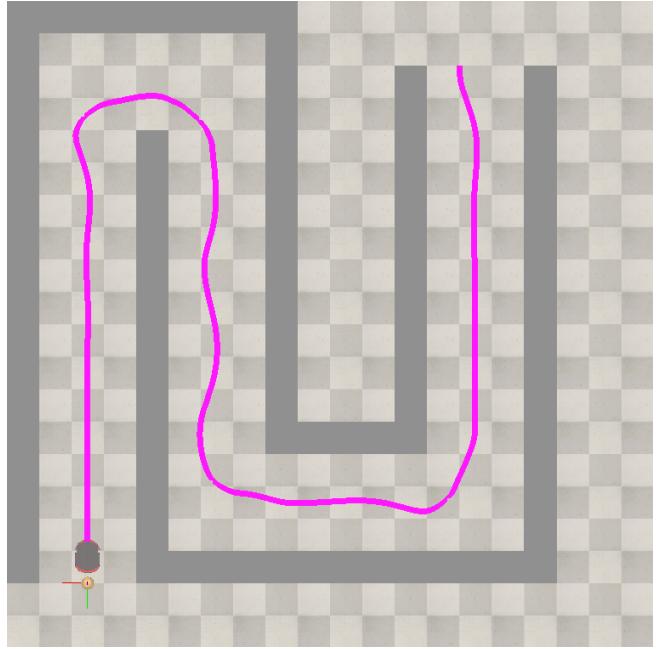


Fig. 15. último cenário, completado unicamente pela abordagem com todos os sensores do P3DX.

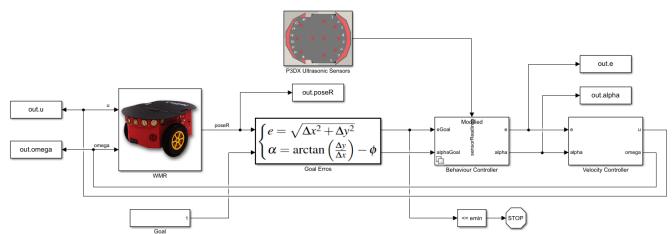


Fig. 16. Modelo geral do sistema de controle.