

Dispositivo Robótico para Assistência à Locomoção de Pessoas Idosas em Ambientes Urbanos

Daniel Leite¹, Karla Figueiredo², Marley Vellasco¹,

¹Laboratório de Inteligência Robótica Aplicada - LIRA,
Departamento de Engenharia Elétrica,
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro
daniels.leite@hotmail.com, marley@ele.puc-rio.br

²Departamento de Ciência da Computação, UERJ -
São Francisco Xavier, 524 - Maracanã - Rio de Janeiro
karla.figueiredo@gmail.com

Abstract. This work presents the development of a navigation algorithm based on fuzzy logic for the displacement of a mobile robot. The goal is to develop an autonomous navigation for an assistance mobility device for elderly people. The robot, which is the base of the robotic device, must be able to autonomously navigate in an unstructured environment, guiding the user to a target, avoiding possible obstacles that could cause the fall of the user. Navigation should be in a smooth and natural way for the user, so that the device itself will not make the user miss the balance. This work was developed in a virtual environment through Stage simulator integrated with robotics framework Player.

Keywords: Robô Móvel, Lógica Fuzzy, Navegação Autônoma, PlayerStage

1 Introdução

Com o aumento da expectativa de vida, o perfil etário da população brasileira vem sofrendo uma mudança já vivenciada por países ricos: o seu envelhecimento. Segundo dados do IBGE [1] e [2], projeções indicam que em 2050 a população idosa, ou seja, pessoas com 65 anos ou mais, estará em torno de 25% da população do país.

No guia das Cidades Amigas dos Idosos [3] são descritas várias características que uma cidade deve ter para atender às necessidades de uma população idosa, dentre as quais, pode-se destacar que a cidade deve ser adaptada para as necessidades de mobilidade, tais como: calçadas largas, degraus baixos, boa sinalização, poucos desníveis, entre outros. Contudo, mesmo em uma cidade adaptada, parte da população idosa faz uso de dispositivos de assistência à locomoção, como bengalas e andadores, por possuir alguma deficiência motora, cognitiva, visual, ou mesmo para ter maior conforto e segurança ao se locomover.

O uso de dispositivos de assistência à locomoção proporciona, à parcela da população idosa mais frágil, o aumento de sua capacidade funcional, gerando maior autonomia para locomoção, tendo como consequências diretas o fortalecimento

muscular e a melhora da qualidade de vida. Esses dispositivos também auxiliam na prevenção de quedas, cujas consequências causam grande impacto para a qualidade de vida da população idosa, dentre as mais impactantes podem-se destacar a deterioração funcional, a hospitalização, a institucionalização e as restrições às atividades de vida diária (AVD), geradas de forma direta, devido a fraturas ou indireta, pelas dores, incapacidade e receios de sofrerem novas quedas [4].

O presente trabalho tem como objetivo o desenvolvimento de um dispositivo robótico para auxiliar a locomoção de pessoas idosas em ambientes urbanos. O dispositivo deve ser capaz de guiar o usuário para um determinado destino de maneira autônoma, desviando de qualquer obstáculo que possa levar o idoso a queda, além de ter uma estrutura que ofereça apoio para o seu deslocamento. As manobras realizadas pelo dispositivo devem ser suaves, evitando realizar movimentos bruscos ou oscilatórios, pois podem gerar desequilíbrio ao usuário. Além da atuação autônoma, o dispositivo deverá ter a capacidade de receber comandos diretos do usuário, que por sua vez deverá ter prioridade sobre qualquer atuação autônoma.

Nesse artigo serão apresentadas as etapas do desenvolvimento do algoritmo de navegação autônoma, bem como sua aplicação e validação em um ambiente virtual. Esse agente será criado e testado por meio do simulador Stage [5]. Essa plataforma foi escolhida devido a sua compatibilidade com o framework de robótica Player [6], que será utilizado no projeto do dispositivo real. As condições de contorno, como velocidade de deslocamento, dimensões do protótipo, funcionalidades e comportamento durante a navegação, foram obtidas a partir da pesquisa realizada por Rivero [7] e [8].

O restante desse trabalho está organizado em mais três seções: a seção 2 descreve a metodologia aplicada no desenvolvimento do algoritmo de navegação; a seção 3 os testes e resultados dos experimentos realizados em ambiente virtual; e por último, na seção 4 são apresentadas as conclusões e indicações para trabalhos futuros.

2 Metodologia

Para dotar o dispositivo da capacidade de se deslocar de maneira autônoma em ambientes desconhecidos e não estruturados, será aplicado um algoritmo de navegação baseado na Lógica Fuzzy. Essa técnica de inteligência computacional foi escolhida principalmente devido a sua capacidade de modelar os modos imprecisos do raciocínio. Essa característica a torna uma ferramenta muito poderosa para a realização de tarefas de navegação, pois a navegação é uma tarefa simples e natural de ser executada por um ser humano, mas complexa de ser modelada por ferramentas matemáticas convencionais, principalmente em ambientes desconhecidos e não estruturados, como é o caso de ambientes urbanos. Além da capacidade de exprimir um conhecimento a partir de um conjunto de regras, a Lógica Fuzzy é uma técnica de fácil aplicação e baixo custo computacional, o que a torna ideal para a tarefa proposta.

Para que o robô se localize no ambiente e receba a informação de seu objetivo, será considerada a existência de uma fonte externa que identifique a posição global do robô e do objetivo, semelhante ao que ocorre em uma navegação com um dispositivo GPS

comercial ou aplicativo de navegação, como o Google Maps©. Para que o robô consiga corrigir sua orientação com o objetivo ele estará embarcado com uma bússola.

A informação dos obstáculos presentes na sua rota de navegação é obtida por sensores do tipo ultrassom. Esse tipo de sensor de distância foi o escolhido, pois é menos suscetível a ruídos devido à intensidade luminosa do ambiente, principalmente gerada de ondas de comprimento infravermelho (o sol), quando comparados a sensores de visão (câmeras) e infravermelho (PSD e LIDAR) [9]. Em relação aos sensores LIDAR, que são excelentes sensores para detecção de obstáculos, os sensores ultrassom ainda tem como vantagem o custo, que chega a ser centenas de vezes menor. Em relação aos sensores de visão, o ultrassom ainda tem como vantagem o baixo custo computacional exigido para obter informações de distância.

As informações provenientes dos sensores de distância, da orientação ao objetivo e da distância ao objetivo serão entradas do Sistema de Inferência Fuzzy (SIF) que terá como saída as velocidades angulares de cada motor.

2.1 Modelo Virtual

Para realizar os ajustes e validação do algoritmo de navegação Fuzzy, conforme já mencionado, foi criado um robô móvel no ambiente do simulador Stage. Esse robô móvel possui uma base diferencial e suas dimensões podem ser observadas na Fig. 1. Essas dimensões foram escolhidas como sendo próximas às do modelo real pretendido.

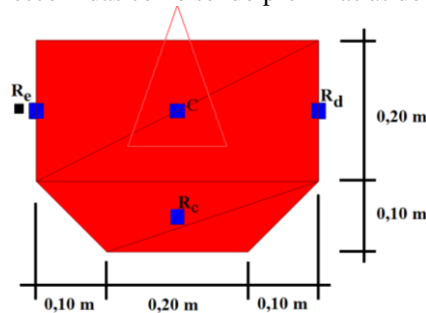


Fig. 1. Representação das dimensões do robô virtual

Na Fig. 1, o ponto C representa o centro de rotação do robô, R_e e R_d as localizações das rodas esquerda e direita, R_c a localização da roda castor e o triângulo central a direção da frente do robô.

O robô está embarcado com oito sensores de distância. Os sensores S1 a S5, cujos posicionamentos podem ser observados na Fig. 2, foram utilizados para identificar obstáculos de colisão. Esses sensores foram configurados, dentro das limitações do simulador, para representar os sensores de ultrassom que serão aplicados na plataforma real [10]. As características desses sensores são: campo de visão 40 graus e alcance 6,45 metros. Os outros três sensores de distância, identificados na Fig. 2 como S6, S7 e S8, foram utilizados para detectar a presença de uma calçada, portanto estão posicionados na parte inferior do modelo. Com isso, ele detecta objetos em um nível mais baixo que os sensores S1 a S5.

Para que haja uma diferenciação entre a calçada e os demais objetos do mapa, objetos que não representam calçadas foram posicionados acima da região de atuação dos sensores S6 a S8, mas dentro da região de atuação dos sensores S1 a S5. A cada valor de leitura foi inserido um ruído gaussiano com desvio padrão duas vezes a resolução linear do sensor, ou seja, 5,08 cm. Na Fig. 2 também é possível observar o campo de visão dos sensores S1 a S8, onde os sensores que detectam as calçadas, S6 a S8, são aqueles cujos campos de visão estão em verde. Esses sensores foram criados com um campo de atuação de 10 graus e distância de leitura 1m. Cada quadrado da grade do fundo da Fig. 2 possui área de 1 m^2 , o que permite avaliar o tamanho do robô e o alcance dos seus sensores.

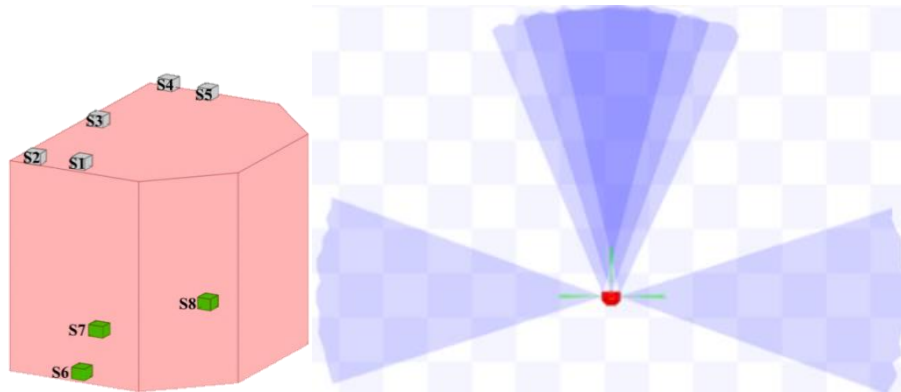


Fig. 2. Representação do robô virtual com os sensores ultrassom e ilustração da área de atuação dos sensores ultrassom do robô virtual

2.2 Ambiente de Navegação

Para realizar os testes do algoritmo de navegação foi criado um mapa representando uma região com vários quarteirões. Esse mapa (Fig. 3) possui uma área de $50 \times 50\text{m}$, e as calçadas possuem espaçamento de 2 m. As dimensões do ambiente criado para navegação, bem como os objetos nele colocados, foram escolhidas de maneira a melhor representar um ambiente urbano, que é o objetivo final de navegação do modelo real. Como referência para as dimensões do mapa, assim como na Fig. 2, cada quadrado da grade do fundo da figura possui área de 1 m^2 .

Devido à limitação do simulador de não permitir posicionar os sensores de modo a identificar desníveis, como a transição entre a calçada e a rua, para emular a presença do meio fio da calçada foram criadas duas alturas de obstáculos. Os obstáculos posicionados mais próximos ao chão, destacados em azul na Fig. 3, representam o meio fio e os objetos pretos, os demais obstáculos do mapa.

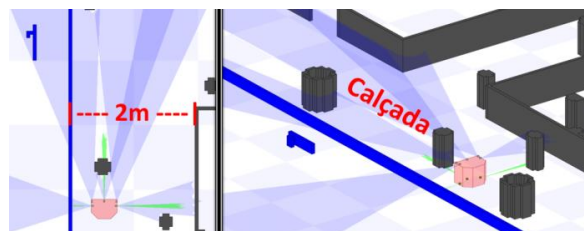


Fig. 3. Trecho do mapa de navegação

Nesse ambiente de navegação foram espalhados diversos objetos para criar regiões de navegação com diferentes graus de complexidade. Por meio da interação do robô móvel com esse ambiente, foi feito o refinamento das regras fuzzy para o controle de navegação.

Na seção seguinte será apresentado o SIF utilizado no algoritmo de navegação do robô.

2.3 Fuzzyficação

As variáveis de entrada utilizadas para o SIF são os dados dos sensores de distância, a orientação em relação ao objetivo e a distância ao objetivo. Já as saídas são os níveis de atuação desejados para cada motor. Os sensores de distância S1 a S5, mostrados na Fig. 2, geraram as seguintes variáveis linguísticas:

- S1: Sensor Lateral Esquerdo
- S2: Sensor Frente Esquerdo
- S3: Sensor Frente
- S4: Sensor Frente Direita
- S5: Sensor Lateral Direita

Cada uma dessas variáveis linguísticas possui três possíveis conjuntos fuzzy: Muito Perto (MP), Perto (P) e Longe (L); e seu universo de discurso vai de 0 à 1,5m. Como o alcance do sensor de distância é superior ao universo de discurso da variável que o representa, caso o sensor detecte um objeto a uma distância maior que 1,5m o valor lido será trocado pela distância de 1,5m. Na Fig. 4 estão representadas as funções de pertinência de cada um dos valores das variáveis linguísticas dos sensores de distância.

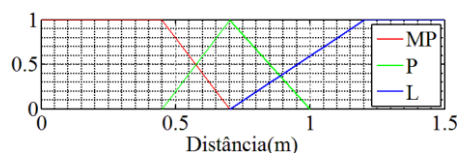


Fig. 4. Funções de pertinência das variáveis dos sensores de distância

Os sensores de distância S6 a S8, utilizados para detectar a divisão entre a calçada e a rua, geraram as seguintes variáveis linguísticas:

- S6: Sensor Altura Esquerdo

- S7: Sensor Altura Frente
- S8: Sensor Altura Direito

Cada uma das variáveis dos sensores de altura possui dois conjuntos fuzzy possíveis, Baixo (BX) e Alto (AL), cujas funções de pertinência podem ser observadas na Fig. 5. No desenvolvimento das regras foi considerada como uma zona segura valor de distância Baixo. Portanto, o robô tenderá a manter uma distância acima de 50 cm do meio fio.

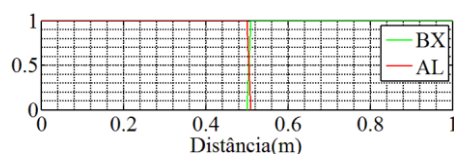


Fig. 5. Funções de pertinência das variáveis Sensor Altura

A orientação do dispositivo em relação ao objetivo é representada pela variável linguística Ângulo. A variável Ângulo possui cinco conjuntos fuzzy possíveis: Negativo Grande (NG), Negativo Médio (NM), Zero (Z), Positivo Médio (PM) e Positivo Grande (PG); e seu universo de discurso vai de -180° a 180° . As funções de pertinência de cada um dos valores da variável Ângulo estão representadas na Fig. 6. O valor entrada de orientação ao objetivo (distância angular) é obtido pela diferença entre a orientação do robô e a direção global do objetivo, ou seja, este será positivo quando o robô está orientado a direita do objetivo até um ângulo máximo de 180° .

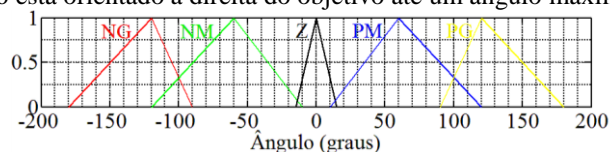


Fig. 6. Funções de pertinência da variável Ângulo

A distância do robô ao objetivo é representada pela variável linguística Distância. Essa variável possui dois conjuntos fuzzy possíveis: Perto (P) e Longe (L); e seu universo de discurso vai de 0 a 3m. Assim como no caso dos sensores ultrassom, ao se obter um valor de distância acima do universo de discurso da variável linguística Distância, esse valor é substituído pelo limite do universo de discurso, ou seja, 3m. Na Fig. 7 estão representadas as funções de pertinência dos valores fuzzy da variável Distância.

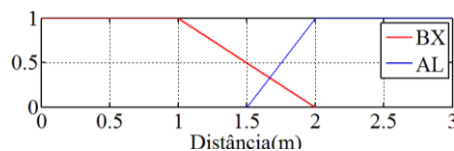


Fig. 7. Funções de pertinência da variável Distância

Para representar a atuação desejada no motor esquerdo e direito foram, respectivamente, criadas as variáveis linguísticas MotorE e MotorD. As variáveis linguísticas dos motores possuem sete conjuntos fuzzy possíveis cada: Negativo Alto (NA), Negativo Médio (NM), Negativo Baixo (NB), Zero (Z), Positivo Baixo (PB),

Positivo Médio (PM) e Positivo Alto (PA); e seu universo de discurso vai de -1 à 1. Essa maior quantidade de valores fuzzy, em relação às outras variáveis do SIF, foi adotada para tentar gerar maior suavidade na transição de atuação, devido a mudanças de regras. Na Fig. 8 estão representadas as funções de pertinência de cada um dos valores das variáveis linguísticas dos motores.

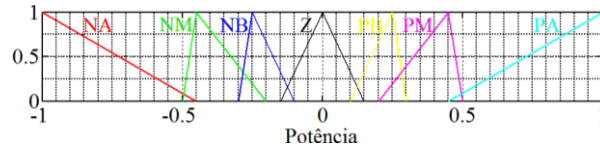


Fig. 8. Funções de pertinência das variáveis MotorE(D)

2.4 Base de Regras

Conforme exposto anteriormente, a base de regras foi criada a partir do conhecimento de um “especialista”. Ela foi escrita visando uma navegação com conversões suaves, com pouca oscilação no valor de velocidade angular do robô, uma velocidade linear média de 0,7 m/s e que não executasse movimentos de ré. Essas características de navegação foram consideradas para que o dispositivo não gere desequilíbrio no usuário.

Para combinar os termos dos antecedentes das regras, foram utilizados os conectores “E” e “OU”. O operador de norma triangular aplicado para representar o conector “E” foi o mínimo e o de co-norma triangular para o conector “OU” foi o máximo. Para representar a implicação da regra utilizou-se o operador mínimo e para combinar as ativações dos consequentes das regras, o operador máximo. Cada regra possui dois conjuntos em seu consequente, que representam os valores desejados para cada motor.

Na composição das regras, o conector “OU” foi aplicado apenas para combinar a variável Ângulo. O resultado dessa operação, assim como os demais termos na regra, é combinado pelo conector “E”. Por exemplo, a regra linguística representada pela equação 1 é reescrita com os operadores mínimo e máximo, conforme a equação 2.

$$\text{Se}[(\hat{\text{Ângulo}}_{\text{NG}} \text{ OU } \hat{\text{Ângulo}}_{\text{NM}}) \text{ E Sensor Frente}_L] \rightarrow \begin{cases} \text{MotorE}_{\text{PM}} \\ \text{MotorD}_{\text{PB}} \end{cases} \quad (1)$$

$$\text{Mín}_{\rightarrow} \left\{ \text{Mín}_E \left[\text{Máx}_{\text{OU}}(\hat{\text{Ângulo}}_{\text{NG}}, \hat{\text{Ângulo}}_{\text{NM}}), \text{Sensor Frente}_L \right], \begin{cases} \text{MotorE}_{\text{PM}} \\ \text{MotorD}_{\text{PB}} \end{cases} \right\} \quad (2)$$

Na Fig. 9, é exibido um trecho do conjunto de regras com a indicação de seus operadores. O raciocínio empregado para a criação das regras foi sempre tentar realizar desvios de obstáculos direcionados ao objetivo. Por essa razão, as regras de navegação e desvio não foram separadas. No exemplo da Fig. 9, as regras 114 e 115 exemplificam esse raciocínio. Em ambas as regras, o robô possui um obstáculo localizado no seu lado direito, tanto a frente quanto na lateral, e seu movimento de desvio o obriga a virar para a esquerda. Contudo, quando mais à esquerda estiver o seu objetivo, ou seja, quanto mais positiva for a variável Ângulo, maior será o seu desvio. Vale observar que ambos os casos o desvio empregado é considerado suficiente para evitar a colisão, mas se torna mais acentuado quando coincide com o objetivo. Outro ponto a se destacar na

metodologia de criação as regras, é que se procurou evitar regras que resultassem em um movimento de rotação puro e que a atuação negativa de um motor fosse mais acentuada que a atuação positiva do outro. A primeira medida foi adotada para evitar configurações nas quais o robô ficasse preso em um movimento oscilatório, já a segunda, foi para que a soma das regras nunca resultasse em uma velocidade linear negativa, ou seja, que o dispositivo realizasse um movimento de ré.

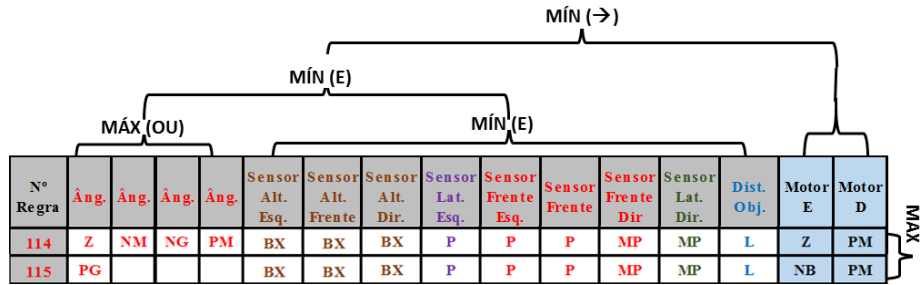


Fig. 9. Exemplo da base de regras com indicação dos operadores

2.5 Defuzzyficação

Para realizar a operação de defuzzyficação dos conjuntos resultantes das regras ativadas, utilizou-se a média ponderada dos máximos, conforme a equação 3.

$$\text{Defuzzy}_{\text{MotorE(D)}} = \frac{\sum_{i=NA}^{PA} \mu_i * \text{MédiaMax}_i}{\sum_{i=NA}^{PA} \mu_i} \quad (3)$$

Onde: i representa os valores fuzzy da variável MotorE(D) , μ_i a pertinência resultante do valor fuzzy i e MédiaMax_i a média dos máximos da função de pertinência do valor fuzzy i .

Esse método foi utilizado por ter representatividade da forma e do grau de ativação do conjunto dos consequentes das regras e por sua simples implementação, quando comparado ao método do Centroide, que possui grande representatividade da forma e do grau de ativação dos conjuntos dos consequentes das regras.

3 Testes e Resultados

Nessa seção serão apresentados os resultados da navegação em ambiente simulado. Ao todo serão apresentados 2 trajetos, cujas disposições dos obstáculos geram diferentes níveis de dificuldade para a navegação.

3.1 Trajeto 1

A primeira parte do trajeto 1 simula uma condição de navegação em uma calçada. Nessa condição, o robô não possui obstáculos de colisão a acima da linha contínua azul,

conforme a Fig. 10, que representa o meio fio da calçada, mas deve sempre se manter afastado da mesma, pois representa uma condição de risco de queda para o idoso. Portanto, esse trajeto visa avaliar a capacidade do algoritmo de ponderar a ausência de obstáculos de colisão em relação à presença do meio fio, com o intuito de conduzir o idoso de maneira segura.

O primeiro trajeto inicia com o robô alinhado com o primeiro objetivo, destacado em amarelo (Fig. 10). A esquerda do robô, linha sólida azul, está o meio fio da calçada, detectável apenas pelos sensores de distância S6 a S8. Os demais objetos do mapa, desenhados em preto, são detectáveis apenas pelos sensores S1 a S5. O centro do círculo amarelo indica o objetivo da navegação e o círculo amarelo, de raio 2m, a tolerância para alcançar esse objetivo.

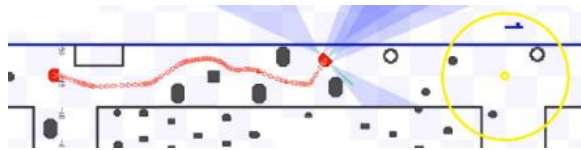


Fig. 10. Trajeto 1 parte 1 início até parada

Nessa primeira etapa do trajeto 1 o robô não consegue chegar ao objetivo, pois ao desviar de um obstáculo ele entra em uma configuração no qual detecta uma calçada a sua frente e lateral, Fig. 11 $t = 46s$. Nessa configuração o SIF tem como resposta uma atuação nula, pois não há informação suficiente garantir a segurança do usuário ao manobrar. Portanto, nesse caso, o usuário deverá ativar o controle manual para alterar o estado do robô.

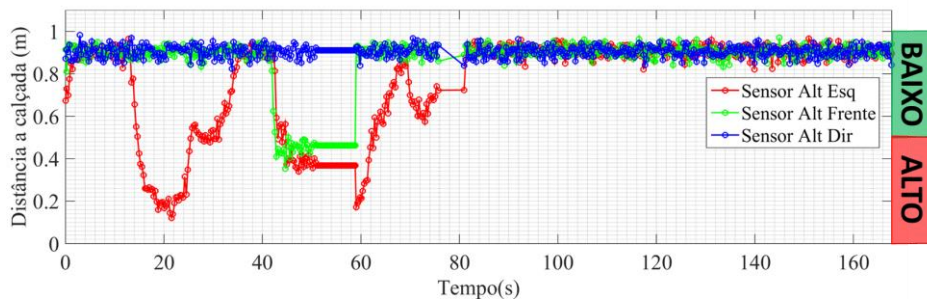


Fig. 11. Leitura dos sensores de altura para o percurso do trajeto 1

Esse procedimento foi realizado no simulador, reposicionando o robô manualmente de modo que estivesse alinhado com o meio fio, Fig. 11 $t = 59s$. A partir do reposicionamento do robô o SIF passou a ter a possibilidade de se deslocar de maneira mais segura para o usuário. A sequência dos passos do robô até atingir o objetivo pode ser observada na Fig. 12.

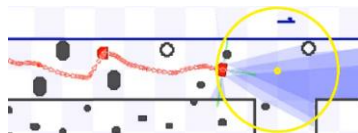


Fig. 12. Término trajeto 1 parte 1

Na segunda etapa do trajeto 1, o robô passa a ter um novo objetivo, indicado na Fig. 13 como o círculo amarelo 2. Essa segunda etapa simula a condição de corredor, onde em ambos os lados do trajeto existem paredes. Entre o robô e o objetivo existem objetos que estão centralizados, em relação ao “corredor”. Esse trajeto visa avaliar a capacidade de manobra do robô, em regiões de pouco espaço, e o desenvolvimento de sua velocidade, em regiões mais livres.

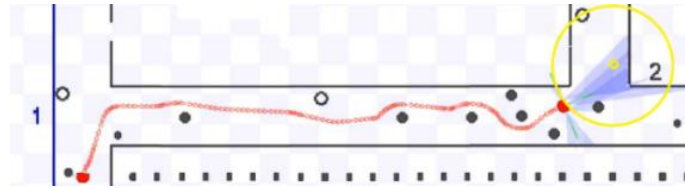


Fig. 13. Trajeto 1 parte 2

Na execução dessa segunda etapa, o robô conseguiu desviar de todos os obstáculos, realizando conversões suaves e aplicando sua velocidade máxima nas regiões em que possuía obstáculos a distância Longe (L). Na Fig. 14 é apresentado o gráfico da saída do SIF e das velocidades para todo o trajeto 1. A primeira parte do trajeto 1 ocorre entre os instantes $t = 0s$ e $t = 76s$, já a segunda parte ocorre entre $t = 81s$ e $t = 168s$.

Para que a velocidade linear máxima (de segurança) executada pelo robô fosse de $0,70 \text{ m/s}$, a saída do SIF foi multiplicada pela constante $4,67$, portanto os valores máximo e mínimo de SIF passaram a ser $\pm 4,67 \text{ rad/s}$.

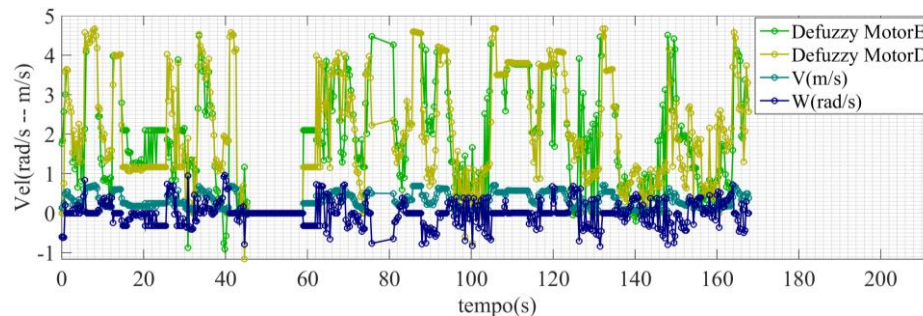


Fig. 14. Saída SIF e velocidades linear e angular do dispositivo na execução do trajeto 1

3.2 Trajeto 2

Esse trajeto, assim como o trajeto 1, está dividido em duas partes. A primeira parte, objetivo 3, possui obstáculos que forçam o robô a se alinhar com o centro do “corredor” e em seguida o robô confronta um novo obstáculo alinhado entre ele e objetivo. Essa etapa tem como objetivo avaliar a capacidade de manobra do robô ao encontrar um obstáculo alinhado com o objetivo. Na segunda parte, objetivo 4, não possui paredes, como nos casos anteriores, mas diversos obstáculos espalhados entre o robô e o objetivo. Essa segunda parte tem como objetivo avaliar a capacidade do robô em manter

alinhado ao objetivo. Na Fig. 15 é possível visualizar o percurso executado pelo robô para alcançar os objetivos 3 e 4 do trajeto 2.

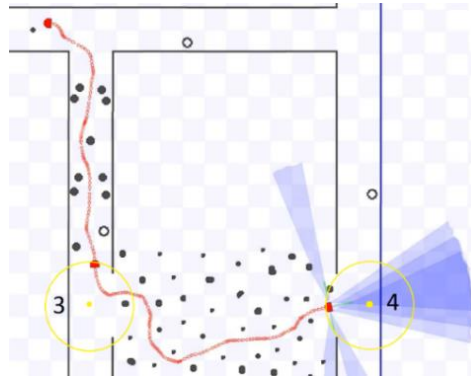


Fig. 15. Trajeto 2 partes 1 e 2

Assim como na segunda etapa do trajeto 1, o robô consegue realizar o trajeto 2 de maneira suave e sem apresentar colisões ou paradas. Na Fig. 16 é possível visualizar a saída do SIF e as velocidades linear e angular executadas pelo robô durante todo o percurso 2. No intervalo de tempo entre de $t = 0s$ e $t = 65s$ foi onde ocorreu a primeira parte do trajeto 2, já a segunda parte ocorreu entre os instantes $t = 70s$ e $t = 135s$.

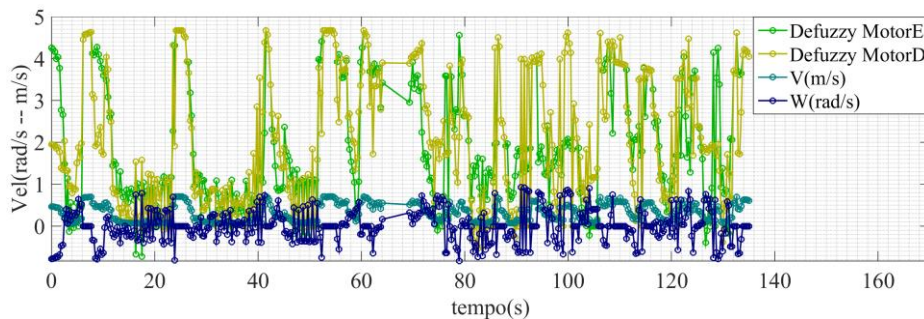


Fig. 16. Saída SIF e velocidades linear e angular do dispositivo na execução do trajeto 2

4 Conclusão

Com base nos resultados apresentados na seção anterior, pode-se concluir que o algoritmo de navegação se mostrou eficiente para a tarefa proposta. Esse fato pode ser afirmado, pois o robô conseguiu realizar os trajetos sem colidir, manobrando de maneira suave para contornar os obstáculos.

Na primeira etapa do trajeto 1 evidenciou-se a capacidade do algoritmo de navegar ponderando a presença do meio fio da calçada, ao realizar os desvios dos obstáculos de

colisão. Mesmo não sendo capaz de completar o percurso sem intervenção, enquanto as informações dos sensores foram suficientes para realizar manobras de modo a garantir a segurança do usuário, o dispositivo conseguiu navegar de maneira suave e sem colidir. Com relação aos demais trajetos, em todos os casos o robô conseguiu se conduzir de maneira suave e sempre direcionada ao objetivo.

Cabe mais uma vez ressaltar a necessidade que a navegação ocorra de maneira natural para o usuário que acompanha o dispositivo. Caso contrário, o dispositivo irá gerar desconforto ou, no pior dos casos, desequilibrar o usuário. O uso da Lógica Fuzzy teve como vantagem, em relação a técnicas matemáticas mais convencionais, a maior facilidade de traduzir a naturalidade da navegação de um usuário humano para o dispositivo robótico proposto.

Como trabalhos futuros, será realizado o desenvolvimento desse algoritmo em uma plataforma real. Essa plataforma estará embarcada com sensores de distância, uma central inercial e *encoders* nas rodas. Todo o processamento ocorrerá em uma CPU de baixo custo, como a Raspberry Pi [11], embarcada no próprio dispositivo e os acessos aos hardwares e softwares de controle do dispositivo serão gerenciados pelo framework de robótica Player [6].

5 Referências

1. IBGE, “Projeção da população por sexo e idade: Brasil 2000-2060 ”, <http://www.ibge.gov.br/home/presidencia/noticias/imprensa/ppts/00000014425608112013563329137649.pdf>
2. IBGE, “Sala de Imprensa ” (2008), <http://saladeimprensa.ibge.gov.br/noticias?view=noticia&id=1&idnoticia=1272&busca=1&t=ibge-populacao-brasileira-envelhece-ritmo-acelerado>.
3. OMS, Guia Global das Cidades Amigas das Pessoas Idosas, 1 ed. (2007).
4. A. Ribeiro, E. Souza, S. Atie, A. Souza e A. Schilithz, “A influência das quedas na qualidade de vida de idosos” *Ciência e Saúde Coletiva*, vol. 13, nº 4, pp. 1265-1273 (2008).
5. R. Vaughan, “Massively multi-robot simulation in stage,” *Swarm Intelligence*, vol. 2, nº 2ª, pp. 189--208 (2008)
6. B. P. Gerkey, R. T. Vaughan e A. Howard, “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems,” In *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, pp. 317--323 (2003)
7. A. M. Rivero, “Novo método Quality by Design – Fuzzy no desenvolvimento de tecnologia biomédica,” PUC-RIO, Rio de Janeiro, 2015.
8. A. M. Rivero, E. Costa Monteiro, D. d. S. Leite, F. Tannenbaum, M. Pinto e K. Figueiredo, *Implementing first stages of Quality by Design approach in the development of an assistive robotics technology* (2015).
9. R. Siegwart e I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)*, 1ª ed., Massachusetts Institute of Technology, 2004.
10. Maxbotix, “LV-MaxSonar-EZ Datasheet”, http://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf.
11. RASPBERRY PI, <https://www.raspberrypi.org/>