

APLICAÇÃO DE CONTROLADORES FUZZY E PROPORCIONAL PARA UM ROBÔ SEGUIDOR DE PAREDE AUTÔNOMO EM AMBIENTE ESTÁTICO

Wilson L. Rodrigues Junior¹, Dyogo M. Reis¹, Ranulfo P. Bezerra Neto¹, Ronnasayd S. Machado², Wanderson A. S. Silva¹, José O. Brito Neto², Ricardo A. L. Rabêlo¹, André M. Santana¹

Universidade Federal do Piauí¹, Universidade Estadual do Piauí²

wilsonlealjunior@hotmail.com, dyogomedeiross@hotmail.com, ronnasayd@hotmail.com,
ranulfo0s@hotmail.com, wass10@hotmail.com, jbritoneto@globo.com, ricardoalr@ufpi.edu.br,
andremacedo@ufpi.edu.br,

Resumo: A navegação de robôs móveis apresenta uma variedade de problemas, como a presença de ruídos e distúrbios, necessidade de fusão sensorial e a dificuldade na obtenção do modelo matemático que representa as interações físicas do ambiente. Cabe ao sistema de controle prover comandos para a correta execução dos movimentos do robô, de forma a alcançar algum objetivo proposto mesmo diante das adversidades do ambiente. Este artigo apresenta a aplicação de diferentes tipos de sistemas de controle, como controladores fuzzy convencional, fuzzy hierárquico e proporcional, para a navegação de um robô seguidor de parede em ambiente estático, cujo objetivo é evitar colisões e manter o robô a uma distância segura da parede. Foram feitas simulações em dois ambientes 3D e um experimento real para validação da abordagem proposta. As simulações foram feitas por meio do simulador V-REP em associação com a linguagem C++, utilizando um robô diferencial autônomo munido de sensores ultrassônicos. O experimento real fez uso de sensores e da placa Arduino acoplados ao chassi de um robô diferencial. Os resultados mostram que o objetivo é alcançado de forma satisfatória por todas as técnicas propostas.

Palavras-chave: Robô móvel; Controlador fuzzy; fuzzy hierárquico; PID

Abstract: The Mobile robot navigation presents a variety of problems, such as the presence of noises and disturbances, necessity of sensorial fusion and struggle in obtaining the mathematical model that represents the environment's physical interactions. The role of the control system is to provide commands for the correct execution of the robot's movements, in a way to reach a proposed target even when faced with the adversities of the environment. This paper presents contributions of different types of control systems, such as fuzzy control system, Hierarchical fuzzy control system and proportional control, for a navigation of a wall following robot in a static environment, which their aims are to avoid collisions and maintain the robot at a safe distance from the wall. Simulations were made in two 3D environments and also a real experiment to validate the proposed approach. The simulations were made through the V-REP simulator associated to C++, using an autonomous differential robot equipped with ultrasonic sensors. The real experiment operated sensors and arduino board attached to a chassi of a differential robot. The results show that the objective is reached satisfactorily through all the proposed techniques.

Keywords: Mobile robot; Fuzzy control; PID; Hierarchical fuzzy

I. INTRODUÇÃO

A navegação de um robô móvel é uma importante área de pesquisa e também fundamental para a construção do sistema de controle que levará o robô a um correto comportamento no ambiente. A fim de construir um mapa para navegação em ambientes internos, o robô deve ser equipado com o comportamento de seguidor de parede. Esse comportamento habilita o robô a seguir contornos de objetos como por exemplo, paredes e obstáculos em ambientes fechados e também pode ser combinado com outros comportamentos considerados inteligentes para alcançar tarefas de alta complexidade [1].

Para conseguir tal comportamento, o robô deve ser dotado de algum mecanismo de controle que forneça comandos para a correta execução de movimentos por parte do robô. Dentre alguns sistemas de controle podemos citar o controlador proporcional integral derivativo (PID), fuzzy e fuzzy hierárquico.

Sistemas proporcionais integrais derivativos são largamente utilizados no processo de controle industrial retroalimentado. A utilidade dos controladores PID está na sua aplicabilidade à maioria dos sistemas de controle, estrutura simples, de fácil desenvolvimento e custo inexpressível. Controladores PID podem ter seus parâmetros sintonizados de forma analítica ou experimental a depender da disponibilidade do modelo matemático do sistema. [2], [3].

Os controladores fuzzy convencional, fuzzy hierárquico e proporcional integral derivativo são tipos de sistemas de controle que apresentam características diferentes na modelagem, definição da estrutura, refinamento e no desempenho. Em [3], [4], [5] são apresentadas características dos controladores PID, fuzzy convencional e fuzzy hierárquico respectivamente.

A lógica fuzzy, ao contrário da lógica clássica, é tolerante a imprecisões, incertezas e verdades parciais. Devido a isso é mais simples implementar um sistema de controle baseado em lógica fuzzy para modelos não lineares. Mais do que isso,

o controlador fuzzy oferece uma possibilidade de mimetizar o conhecimento e a expertise humana [4].

O uso de um sistema hierárquico fuzzy permite diminuir o número de regras constituintes da base de regras fuzzy linearmente com o número de variáveis de entrada. A vantagem é então reduzir o número de regras fuzzy, comparado com o fuzzy convencional, onde o número de regras cresce exponencialmente com o número de variáveis de entrada, implicando em um menor custo computacional [5].

A proposta do artigo é aplicar estes tipos de sistemas de controle a um robô com rodas de acionamento diferencial de forma a verificar qual controlador realiza curvas mais suaves e mantém uma distância mínima em relação a parede.

II. REVISÃO LITERÁRIA

A. Controle Proporcional integral derivativo (PID)

Mais da metade dos controladores industriais em uso atualmente empregam esquemas de controle PID ou PID modificado. Como a maioria dos controladores PID é ajustada em campo, diferentes tipos de regras de sintonia vêm sendo propostas na literatura. Com a utilização dessas regras de sintonia, ajustes finos no controlador PID podem ser feitos em campo. Além disso, métodos de sintonia automática vêm sendo desenvolvidos e alguns controladores PID têm a capacidade de fazer sintonia automática online [6], [7], [8], [9]. Estruturas PID modificadas, como o controle I-PD e o controle PID com vários graus de liberdade, atualmente estão em uso. Vários métodos práticos de comutação suave (de operação manual para operação automática) e ganho programado estão comercialmente disponíveis.

A utilidade dos controles PID está na sua aplicabilidade geral à maioria dos sistemas de controle para sistemas lineares invariantes no tempo. Em particular, quando o modelo matemático da planta não é conhecido e, portanto, métodos de projeto analítico não podem ser utilizados, controles PID se mostram os mais úteis. Na área dos sistemas de controle de processos, sabe-se que os esquemas básicos de controle PID e os controles PID modificados provaram sua utilidade conferindo um controle satisfatório, embora em muitas situações eles possam não proporcionar um controle ótimo [10], [11], [12], [2], [13].

O algoritmo PID pode ser descrito de acordo com a equação (1), onde $\mu(t)$ é a variável de controle que é a grandeza ou a condição modificada pelo controlador, de modo a afetar o valor da variável controlada e $e(t)$ é o erro de controle dado pela diferença do estado desejado e o estado atual que esta sendo medido. A variável de controle é então a soma de três termos: o termo proporcional (o qual é proporcional ao erro de controle), o termo integral (o qual é proporcional a integral do erro), e o termo derivativo (o qual é proporcional a derivada do erro). Os parâmetros de controle a serem ajustados são o ganho proporcional K_p , o ganho integrativo K_i , e o ganho derivativo K_d [12]. A Figura 3 demonstra o esquema de um sistema de controle PID aplicado ao controle de distância em relação a parede para um robô diferencial. Onde a variável de

controle proposta é a velocidade angular do robô em relação ao ponto central entre suas rodas.

$$\mu(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

1) Ação de controle proporcional: Para um controlador com ação de controle proporcional, a relação entre a saída do controlador $\mu(t)$ e o sinal de erro atuante $e(t)$ é :

$$\mu(t) = K_p e(t) \quad (2)$$

Onde K_p é denominado ganho proporcional. Qualquer que seja o mecanismo real e o tipo de energia utilizada na operação, o controlador proporcional é essencialmente um amplificador de ganho ajustável.

2) Ação de controle integral: Em um controlador com ação de controle integral, o valor da saída $\mu(t)$ do controlador é modificado a uma taxa de variação proporcional ao sinal de erro atuante $e(t)$. A principal função da ação integral é fazer com que a saída do processo concorde com o sinal de referência em estado estacionário. Com a ação de controle proporcional, normalmente existe um erro em estado estacionário. Com a ação de controle integral, um pequeno erro positivo sempre levará a um aumento no sinal de controle, e um erro negativo a uma diminuição no sinal de controle não importa qual pequeno esse erro seja.

$$\frac{d\mu(t)}{dt} = K_i e(t) \quad (3)$$

ou

$$\mu(t) = K_i \int_0^t e(\tau) d\tau \quad (4)$$

Onde K_i é uma constante ajustável.

3) Ação de controle proporcional-integral: A ação de controle de um controlador proporcional integral é definida por:

$$\mu(t) = K_p e(t) + \frac{K_p}{\tau_i} \int_0^t e(\tau) d\tau \quad (5)$$

Onde τ_i é o tempo integrativo.

4) Ação de controle proporcional-derivativo: O propósito da ação de controle derivativo é prover a estabilidade ao sistema de malha fechada. Por causa da dinâmica do processo, levará algum tempo antes que a mudança na variável de controle seja noticiada a saída do processo. Portanto, o sistema de controle estará atrasado na correção do erro. A ação de um controle proporcional-derivativo é proporcional a saída do sistema prevista, onde a previsão é feita pela extrapolação do erro pela tangente da curva do erro. A ação de controle de um controlador proporcional-derivativo é definida por:

Onde τ_d é chamado de tempo derivativo.

$$\mu(t) = K_p e(t) + K_p \tau_d \frac{de(t)}{dt} \quad (6)$$

5) Ação de controle proporcional-integral-derivativo: A combinação das ações de controle proporcional, de controle integral e de controle derivativo é denominada ação de controle proporcional-integral-derivativo. Essa combinação tem as vantagens individuais de cada uma das três ações de controle. A equação de um controlador com essas combinações é dada por:

$$\mu(t) = K_p e(t) + \frac{K_p}{\tau_i} \int_0^t e(\tau) d\tau + K_p \tau_d \frac{de(t)}{dt} \quad (7)$$

B. Sistemas fuzzy

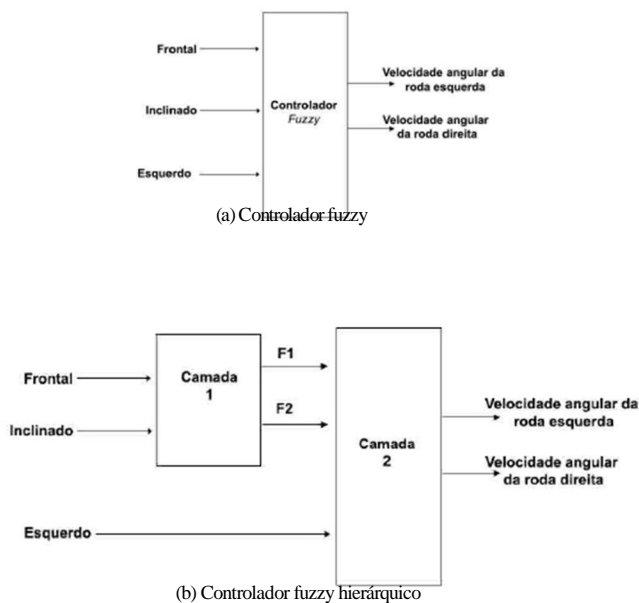


Figura 1. Controladores fuzzy

O controle convencional muitas vezes requer o completo conhecimento do sistema e tende a não apresentar uma boa performance em ambientes com ruídos e incertezas. Estas dificuldades de modelar o sistema completo pode ser resolvida pelo emprego de técnicas de soft computing. Dentre essas técnicas, a lógica fuzzy oferece uma solução promissora para lidar com as incertezas do mundo real por meio da imitação da experiência humana na forma de regras [1].

Muitas das ações humanas e comportamentos podem ser realizadas utilizando-se de um estrutura de conjuntos de regras se-então que, implicitamente, foram desenvolvidas ao longo de anos de conhecimento e experiência.

A teoria de conjunto fuzzy foi desenvolvida e inspirada na capacidade humana de lidar com processos ambíguos. Dentre

as principais características dos controladores baseados na lógica fuzzy está sua habilidade de tomar decisões adequadas por um processo de inferência linguística. O controlador baseado em logica fuzzy (CLF) representa esse processo de inferência

por meio de regras fuzzy baseada em conhecimento heurístico e experiência, que são frequentemente usados para controlar um dado sistema. Um mecanismo de inferência processa a informação guardada na base de conhecimento para determinar a ação de controle adequada a ser tomada em qualquer situação [14]. Em [15] é demonstrado de forma clara o funcionamento de sistemas fuzzy do tipo mamdani.

A Figura 2 mostra a representação de um sistema fuzzy por meio de blocos. O bloco de fuzzificação é responsável por receber as entradas numéricas resultantes de medições ou observações e transformá-las em variáveis linguísticas. A interpretação dessas variáveis linguísticas é feita pelo bloco de inferência que por meio de uma pesquisa na base de regras determina quais regras fuzzy serão ativadas. Cabe a defuzzificação receber essas regras e combiná-las para fornecer uma saída precisa.

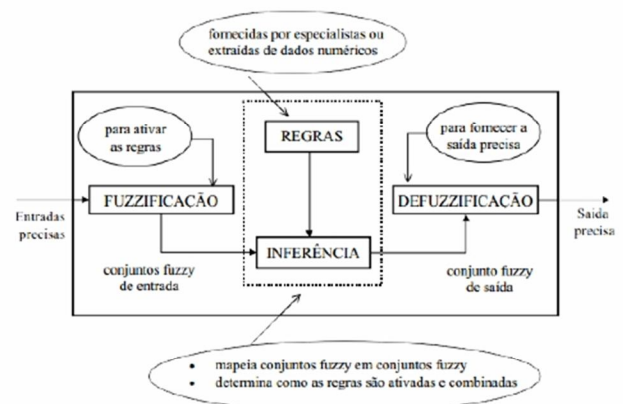


Figura 2. Sistema de inferência fuzzy [15].

C. Sistemas fuzzy hierárquico

O projeto do CLF é comumente uma atividade demorada que envolve aquisição de conhecimento, definição da estrutura de controle, definição das regras, e outros parâmetros de controle.

Atualmente, uma das mais importantes questões em sistemas fuzzy é como reduzir o número total de regras envolvidas e suas exigências computacionais. Em um sistema fuzzy convencional, o número de regras cresce exponencialmente em relação ao número de variáveis de entrada [16]. Esse problema é chamado de maldição da dimensionalidade e, afim de resolvê-lo, em [17] é aplicado e proposto o conceito de controlador fuzzy hierárquico (CFH). Em [16] é mostrado a diferença no número de regras fuzzy utilizadas em um sistema fuzzy e um sistema fuzzy hierárquico.

A Figura 1(b) mostra um sistema fuzzy hierárquico. Suponha um sistema fuzzy com 4 entradas e 6 funções de pertinência

para cada variável de entrada, então para cada sistema fuzzy de baixa dimensionalidade temos $6^2(m^2)$ regras, portanto o total de número de regras é $3 \times 6^2 = 108((n-1)m^2)$, que é uma função linear em relação ao número de variáveis de entrada. No sistema fuzzy convencional temos $6^4 = 1296$.

D. Ambiente Simulado

O uso de um ambiente virtual de simulação de robôs tem se demonstrado uma ferramenta poderosa, apresentando uma série de vantagens:

- Economia de recursos financeiros, pois diversos testes podem ser realizados antes de ser implementado fisicamente o robô;
- Economia de tempo, pois podemos realizar um maior número de experimentos por meio de simulação, nos quais a realização de um experimento requer um menor tempo para configurar o experimento (não há necessidade de recarregar baterias e posicionar equipamentos e objetos), além de se poder "acelerar o tempo" do relógio virtual na execução das simulações;
- Evitar danos ao robô, pois por meio das simulações pode-se verificar previamente as situações que podem provocar danos ao robô, devido por exemplo a fortes colisões, acionamento indevido dos motores, ou exposição do robô à ambientes perigosos para testes;
- Evitar acidentes e aumentar a segurança, por meio da simulação pode-se realizar diversos testes visando garantir uma maior segurança e robustez do sistema robótico, evitando assim a incidência de acidentes com pessoas e com elementos presentes no ambiente de atuação do robô, permitindo inclusive uma melhor análise de como adicionar novos dispositivos de salvaguarda em hardware e software que permitam aumentar a segurança e confiabilidade dos robôs;
- Aperfeiçoamento do hardware e software, uma vez que se pode, por meio de simulações, testar diferentes configurações de hardware, bem como testar e avaliar novas implementações e ajustes nos parâmetros do software de controle, permitindo assim uma melhoria do sistema como um todo e a otimização do uso dos recursos disponíveis a fim de obter uma maior eficiência do sistema robótico [18].

Em um ambiente simulado todas as variáveis são controladas facilitando assim a modelagem do controle do robô

E. Ambiente real

Embora o projeto de um sistema robótico móvel inteligente simulado apresente suas vantagens, ele não é um sistema real. Suas desvantagens são: apresentação de todas as variáveis controladas, tem pouco ou quase nenhum ruído para se preocupar, o hardware está com suas funcionalidades sempre funcionando corretamente, a recepção de dados dos sensores é sincronizada e etc. Para verificar o desempenho dos controladores em um ambiente real é necessário implementar um projeto de um sistema robótico móvel inteligente.

Esse projeto envolve duas tarefas principais: o projeto de hardware e o projeto de software. O hardware é responsável pela implementação física de sensores, atuadores, interfaces e do processamento dos dados por meio do uso de um processador embarcado: microprocessadores, microcontroladores, hardware dedicado, hardware reconfigurável (FPGAs, Soft-Cores, SoPC, PSOC - Programmable System-on-Chip). O software implementa o sistema de controle robótico, responsável pela análise dos sinais dos sensores, planejamento e tomada de decisões, e controle dos atuadores responsáveis pelo deslocamento e ações do robô.

Grande parte da dificuldade em se programar robôs móveis deve-se ao fato de que os elementos que compõem o robô, como sensores e atuadores, apresentam um certo grau de incerteza nos dados obtidos ou nas ações desempenhadas. Além disso, o ambiente em que esses robôs operam, normalmente, é dinâmico e imprevisível. A simples tarefa de se mover de uma extremidade de um corredor até a outra pode se tornar consideravelmente mais difícil se os sensores e atuadores do robô não funcionarem com precisão [18].

III. ABORDAGEM

Os robôs móveis têm como característica principal, não estarem restritos a um espaço de trabalho fixo tendo assim liberdade de circulação no ambiente. Estes robôs são concebidos para movimentação planar 2D em ambientes terrestres e são capazes de translação e rotação em torno do seu centro de gravidade. Afim de avançar em áreas apertadas e evitar entaves, robôs móveis devem ter boa mobilidade e capacidade de manobra. Estas características, só são possíveis dotando o robô com sistemas de sensores e atuadores, que forneçam informação correta sobre o ambiente de operação e capacidade de movimentação adequada.

Em todas as aplicações robóticas, a execução de uma tarefa genérica exige uma correta movimentação por parte do robô a fim de percorrer um caminho sem que ocorram colisões, quer com obstáculos fixos ou obstáculos móveis. A correta execução de tal movimento é designada ao sistema de controle que irá prover os atuadores do robô com comandos consistentes ao movimento desejado. Estes sistemas de controle requerem em sua maioria a existência de modelos dinâmicos ou cinemáticos do sistema que se deseja controlar [19].

Os controles propostos nesse artigo tem como objetivo fornecer comandos de forma que o robô possa seguir uma parede, evitando colisões, mantendo-se a uma distância segura da parede.

A. Controlador proporcional

Levando-se em consideração a não linearidade intrínseca da planta de um robô diferencial, foram feitos controladores proporcionais para estados específicos em que o robô poderia se encontrar no ambiente, sendo que o parâmetro K_p de cada controlador foi sintonizado individualmente por meio de métodos heurísticos em ambientes de testes, que representava o estado desejado, sendo estes, detecção de parede pelo sensores: esquerdo, inclinado e frente. O controlador proporcional foi

escolhido devido a sua maior facilidade de sintonia em um ambiente real para um robô diferencial.

Tendo encontrado os valores de parâmetros, os três controladores proporcionais são executados simultaneamente e o valor da ação controle que acionará a roda esquerda e direita é a soma do valor da ação controle dos três controladores proporcionais. O sistema de controle proporcional atua em cima do erro dado pela diferença entre a distância de segurança e a distância real que o robô se encontra em relação a parede, como ilustrado na Figura 3

B. Sistemas fuzzy

O CLF proposto foi modelado utilizando a biblioteca eFLL [20] de sistemas fuzzy para C++ sendo que este contém três variáveis de entrada, que são os sensores que estão distribuídos na frente, esquerda, e entre o sensor da frente e da esquerda do robô. Cada uma dessas variáveis contém quatro funções de pertinência: muito próximo (MP), próximo (P), médio (M) e distante (D), que representam a distância do robô em relação a parede (Figura 4). O CLF tem como saída a velocidade angular das duas rodas que tem três funções de pertinência: baixo (B), moderado (M) e alto (A) (Figura 5). O CLF proposto resultou em um total de 64 regras fuzzy para cada roda. A base de regra usada pelo CLF foi modelada, testada e refinada por meio de conhecimento heurístico. A Figura 1(a) mostra o CLF proposto.

As Tabelas I e II são uma representação da base de regras do CLF utilizada para o controle autônomo do robô, onde frontal, inclinado, esquerdo são as entradas, e B, M, A são as saídas.

C. Sistemas fuzzy hierárquico

Após a modelagem do CLF proposto atender aos objetivos, foi utilizado o método L-HFS proposto em [16] que atua sobre a base de regras do sistema fuzzy convencional para gerar a base de regras do CFH. O CFH proposto é mostrado na Figura 1(b). Ele foi modelado usando duas camadas, tal que na camada 1 as variáveis de entrada são as leituras do sensor da frente e o inclinado, e na camada 2 as entradas são a variável artificial, que é a saída da camada anterior, e a leitura do sensor da esquerda. O CFH tem como saída a velocidade angular das duas rodas que tem três funções de pertinência (baixo, moderado e alto), assim como no CLF proposto. O CFH proposto resultou em um total de 16 + 12 regras fuzzy para as camadas 1 e 2, respectivamente, totalizando 28 regras fuzzy no controle da roda esquerda, para o controle da roda direita resultou em 16 + 8 regras fuzzy para as camadas 1 e 2, respectivamente, totalizando 24 regras. Em [16] e [21] são explanadas algumas desvantagens que podem tornar difícil o projeto do CFH e acabam por degradar o desempenho.

Porém no CFH, em muitos casos as saídas intermediárias são artificiais e não possuem significado físico. Assim, para as unidades de lógica fuzzy (ULF), se elas são usadas como variável de entrada da próxima camada como a flecha pontilhada na Figura 6, que é o caso usual, então a regra fuzzy envolvida na camada do meio da estrutura hierárquica tem um pequeno significado físico e consequentemente são difíceis de

se projetar. Este fenômeno toma-se proeminente a medida que o número de camadas cresce em um SFH [16]. Se as regras de composição das saídas fuzzy no nível médio não são razoáveis, ela causará a degradação do desempenho do controle [21].

Intuitivamente, existem dois tipos de coluna na Tabela III e três tipos Tabela IV, então são requeridas duas variáveis artificiais, F1 e F2, a variável artificial do controlador da roda esquerda, F1, necessita de três funções de pertinência para classificá-la E1, E2 e E3 (Figura 7(b)), e para a variável da roda direita, F2, são necessárias duas funções de pertinência, D1 e D2 (Figura 7(a)), então basta relacionar as entradas frontal e inclinado, e cada tipo de coluna agora corresponde a uma função de pertinência. As Tabelas V e VI são obtidas relacionando a variável esquerdo com as variáveis F1 e F2, e como saídas temos as velocidades angulares das rodas esquerda e direita, respectivamente.

IV. EXPERIMENTOS E RESULTADOS

Inicialmente o comportamento de cada controlador foi simulado utilizando os cenários A e B. Sendo que o cenário A retrata um ambiente de fácil locomoção com angulação obtusa entre as paredes e o cenário B representando a simulação de um ambiente real, com paredes em curvatura de 90 graus. Após simulação foi realizado o experimento real utilizando o cenário C. A Figura IV ilustra os cenários supracitados. A proposta da experimentação prática é analisar o comportamento do robô diante de todas adversidades que o ambiente pode oferecer, ruídos dos sensores, irregularidades do piso, fusão sensorial e etc. A seguir são apresentados os experimentos simulados e reais, bem como os resultados obtidos.

As simulações foram feitas utilizando o virtual experimentation platform (V-REP)[22] em associação a um programa desenvolvido na linguagem C++, utilizando um modelo de robô diferencial com sensores ultrassônicos distribuídos na frente, esquerda e entre o sensor da frente e da esquerda do robô. O V-REP coleta as medições dos sensores em relação ao ambiente e envia os dados para o programa em C++. No programa em C++ foi modelado todo o sistema fuzzy, por meio da biblioteca eFLL, que tem como entrada os valores dos sensores e como saída a velocidade angular das duas rodas (esquerda e direita). As Figuras 8(a) e 8(b) mostram o cenário de simulação que todos os controladores foram submetidos.

O experimento real foi feito utilizando um robô com rodas de acionamento diferencial, três sonares, uma placa Arduino [23] e duas baterias de alimentação, como pode ser visto na Figura 9. Todos os controladores foram implementados diretamente no Arduino utilizando a mesma biblioteca usada na simulação.

A Figura 10 ilustra as respectivas trajetórias dos controladores proporcional, fuzzy convencional e fuzzy hierárquico para o cenário A. O qual consiste de um octógono não simétrico com a finalidade de avaliar a eficiência dos controladores para trajetórias com curvas suaves.

A Figura 10(a) mostra a trajetória do controlador Proporcional, do qual observa-se que o controlador apresenta uma

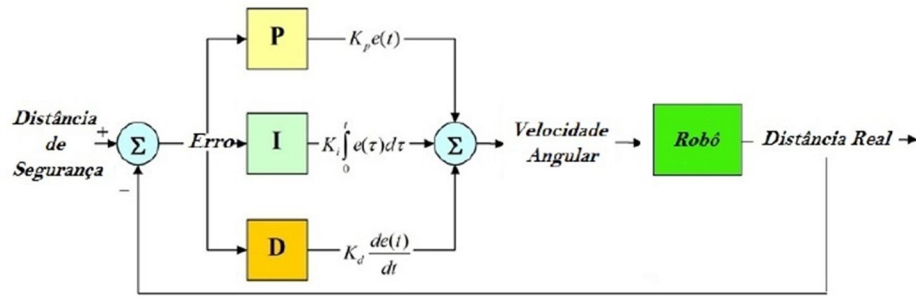


Figura 3. Sistema de Controle PID proposto.

Tabela I
BASE DE REGRAS DO CONTROLADOR FUZZY PARA A RODA ESQUERDA

		FRONTAL															
		MP INCLINADO				P INCLINADO				M INCLINADO				D INCLINADO			
ESQUERDO	MP	A	A	A	A	A	M	M	M	A	M	M	M	A	M	M	A
	P	A	A	A	A	A	M	M	M	A	M	M	M	A	M	M	M
	M	A	A	A	A	A	M	M	M	A	M	M	M	A	M	M	B
	D	A	A	A	A	A	M	M	M	A	M	M	M	A	M	M	B
		E1	E1	E1	E1	E1	E2	E2	E2	E1	E2	E2	E2	E1	E2	E2	E3

Tabela II
BASE DE REGRAS DO CONTROLADOR FUZZY PARA A RODA DIREITA

		FRONTAL															
		MP INCLINADO				P INCLINADO				M INCLINADO				D INCLINADO			
ESQUERDO	MP	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	M
	P	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	M
	M	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	M
	D	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	M
		D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D2

Tabela III
PRIMEIRA CAMADA DO CONTROLADOR FUZZY HIERÁRQUICO PARA A RODA ESQUERDA

		INCLINADO			
		MP	P	M	D
FRONTAL	MP	E1	E1	E1	E1
	P	E1	E2	E2	E2
	M	E1	E2	E2	E2
	D	E1	E2	E2	E3

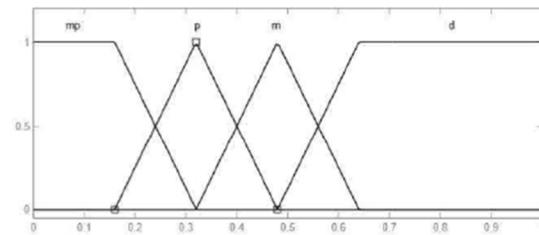


Figura 4. Funções de pertinência da distância.

Tabela IV
PRIMEIRA CAMADA DO CONTROLADOR FUZZY HIERÁRQUICO PARA A RODA DIREITA

		INCLINADO			
		MP	P	M	D
FRONTAL	MP	D1	D1	D1	D1
	P	D1	D1	D1	D1
	M	D1	D1	D1	D1
	D	D1	D1	D1	D2

trajetória mais sinuosa devido a não linearidade do robô, que implica em uma dificuldade para o controlador fazer ajustes finos em torno da referência de distância proposta.

As Figuras 10(b) e 10(c) representam as trajetórias para os controladores fuzzy convencional e hierárquico respectivamente. Percebe-se que o controlador fuzzy tem um comportamento mais linear no seu percurso do que o controlador Proporcional. Isto se deve ao fato de que sistemas de controle fuzzy lidam melhor com incertezas o que garante que os

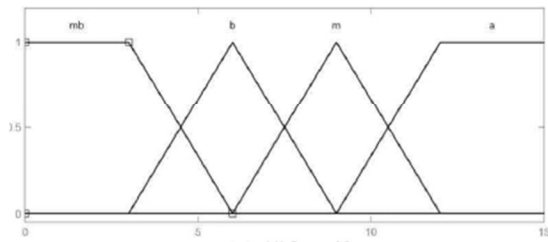


Figura 5. Funções de pertinência da velocidade.

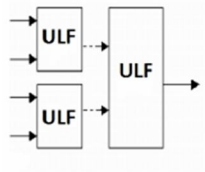
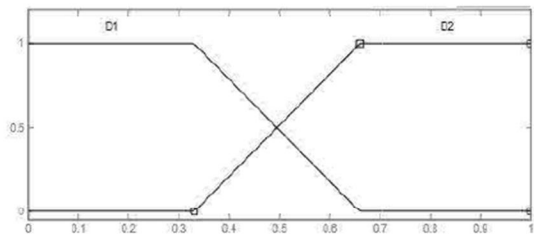
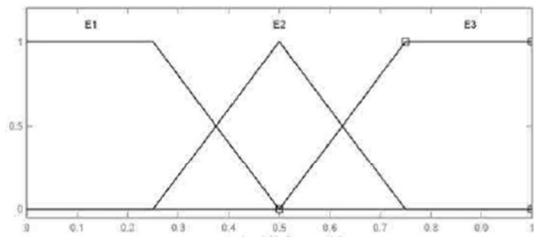


Figura 6. Sistema fuzzy hierárquico [16].



(a) Funções de pertinência da variável artificial direita.



(b) Funções de pertinência da variável artificial esquerda.

Figura 7. Funções de pertinência das variáveis artificiais.

mesmos tratem melhor sistemas não lineares. Em relação ao sistema hierárquico pode-se verificar que o mesmo consegue atingir um alto nível de similaridade com o sistema fuzzy convencional devido ao fato das variáveis artificiais estarem bem definidas.

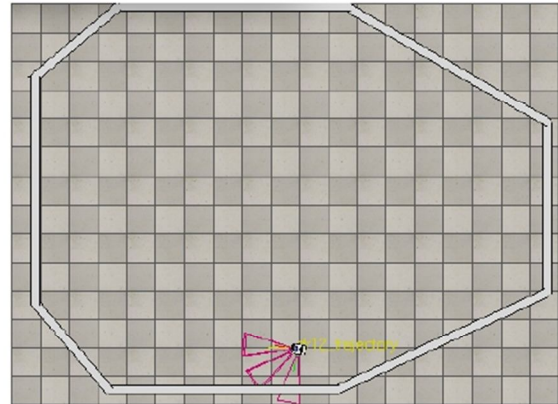
A Figura 11 ilustra as respectivas trajetórias dos controladores Proporcional, Fuzzy convencional e fuzzy hierárquico para

Tabela V
SEGUNDA CAMADA DO CONTROLADOR FUZZY HIERÁRQUICO PARA A RODA ESQUERDA

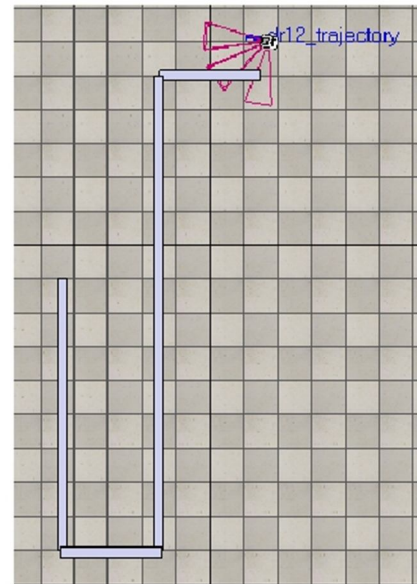
		ESQUERDO			
		MP	P	M	D
F1	1	A	A	A	A
	E2	M	M	M	M
	E3	A	M	B	B

Tabela VI
SEGUNDA CAMADA DO CONTROLADOR FUZZY HIERÁRQUICO PARA A RODA DIREITA

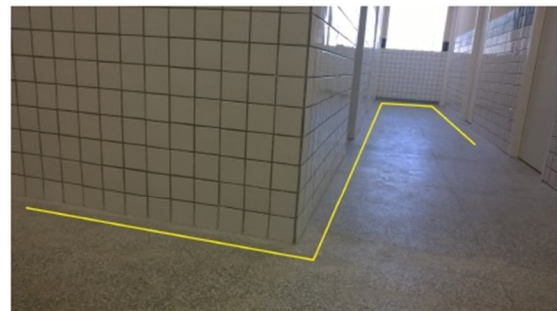
		ESQUERDO			
		MP	P	M	D
F2	D1	B	B	B	B
	D2	M	M	M	M



(a) Cenário A.



(b) Cenário B.



(c) Cenário C.

Figura 8. Cenários

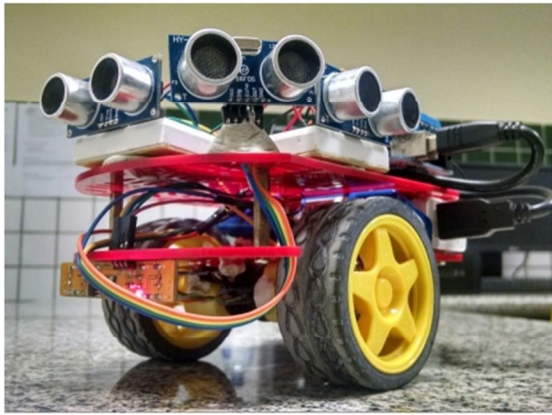
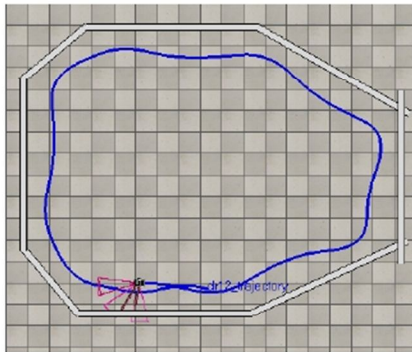
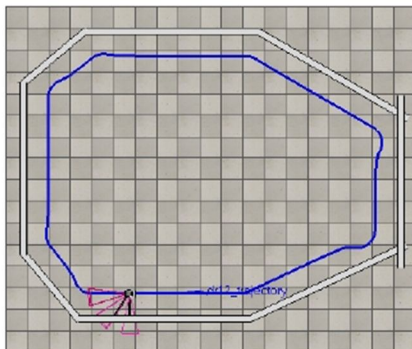


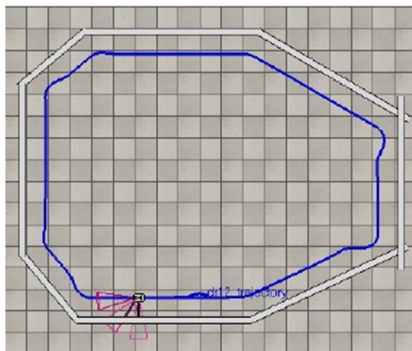
Figura 9. Robô utilizado para os experimentos.



(a) Trajetória PID.

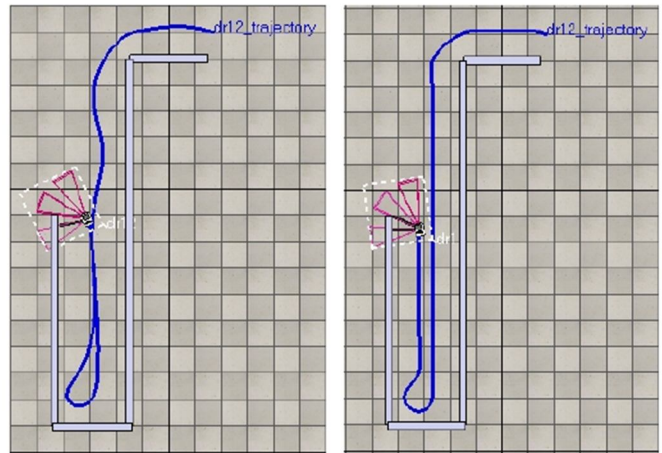


(b) Trajetória Fuzzy Convencional.



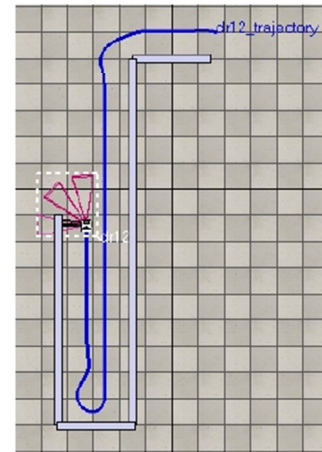
(c) Trajetória Fuzzy Hierárquico.

Figura 10. Trajetórias no Cenário A



(a) Trajetória PID.

(b) Trajetória Fuzzy Convencional.



(c) Trajetória Fuzzy Hierárquico.

Figura 11. Trajetórias no Cenário B

o cenário B, o qual consiste na representação em ambiente simulado do cenário real proposto.

A Figura 11(a) representa a trajetória do controlador proporcional para o cenário B. Observa-se que o controlador proporcional apresenta curvas sinuosas durante o percurso e aproxima-se levemente da parede esquerda ao final do corredor.

As Figuras 11(b) e 11(c) representam as trajetórias dos controladores fuzzy convencional e hierárquico respectivamente. Observa-se que o controlador fuzzy convencional apresenta movimentos suaves nas curvas com ângulos retos e se mantém a uma distância confortável das paredes durante todo o percurso. O controlador fuzzy hierárquico consegue alcançar o mesmo desempenho do fuzzy convencional com uma base de regras reduzida.

As Tabelas VII, VIII e IX apresentam um conjunto de métricas que avaliam o comportamento dos controladores propostos. Dentre as métricas avaliadas estão Velocidade Angular Média (VAM), Velocidade Linear Média (VLM) e Distância Média em Relação a Parede (DMRP). As métricas consistem na média e desvios padrões retirados a partir dos dados de

Tabela VII
MÉTRICAS DE AVALIAÇÃO NO CENÁRIO A

Sistema de Controle	DMRP	VLM	VAM
PID	0, 4691 □ □ 0, 0430	5, 0000 □ □ 0	-1, 3348 □ □ 2, 7996
Fuzzy Convencional	0, 4713 □ □ 0, 0382	7, 5328 □ □ 0, 1639	-1, 8941 □ □ 7, 0254
Fuzzy Hierárquico	0, 4223 □ □ 0, 0351	7, 0939 □ □ 0, 4862	-1, 6552 □ □ 8, 4726

Tabela VIII
MÉTRICAS DE AVALIAÇÃO NO CENÁRIO B

Sistema de Controle	DMRP	VLM	VAM
PID	0, 4589 □ □ 0, 0812	5, 015 □ □ 0, 0854	-0, 3494 □ □ 7, 7638
Fuzzy Convencional	0, 4605 □ □ 0, 0652	7, 6497 □ □ 0, 4146	-0, 6763 □ □ 9, 9931
Fuzzy Hierárquico	0, 4196 □ □ 0, 0545	7, 1229 □ □ 0, 6335	-0, 6382 □ □ 9, 9695

Tabela IX
MÉTRICAS DE AVALIAÇÃO NO CENÁRIO C

Sistema de Controle	DMRP	VLM	VAM
PID	0, 3468 □ □ 0, 1928	7, 8878 □ □ 0, 0737	4, 2539 □ □ 4, 5120
Fuzzy Convencional	0, 4526 □ □ 0, 0743	7, 5792 □ □ 0, 3119	3, 7473 □ □ 2, 4936
Fuzzy Hierárquico	0, 4489 □ □ 0, 0652	7, 2576 □ □ 0, 4831	3, 9075 □ □ 2, 5417

três simulações feitas para cada controlador em cada um dos cenários propostos.

Das Tabelas VII e VIII observa-se que para a métrica DMRP os três controladores apresentaram valores similares tanto para a média como para o desvio padrão.

Para ambientes simulados o fuzzy hierárquico apresenta os menores valores de DMRP e o fuzzy convencional os maiores valores.

O fuzzy convencional tem a maior velocidade linear em ambiente simulado e o fuzzy hierárquico o maior desvio padrão para a velocidade linear. O controlador proporcional apresenta a menor velocidade linear nas simulações.

Para o ambiente real (Tabela IX) observa-se que os controladores fuzzy apresentaram valores similares aos simulados. O controlador proporcional apresentou uma performance pior em relação aos resultados obtidos durante as simulações devido aos ruídos inerentes de um cenário real.

O projeto do CLF envolveu aquisição de conhecimento, definição da estrutura de controle, regras, funções de pertinências, refinamentos. Os comandos fornecido pelo CLF fizeram com que o robô seguisse a parede e se movimentasse de forma suave como pode ser constatado pelas Figuras 10(b) e 11(b).

De posse da base de regra do CLF bastou aplicar o método L-HFS para gerar a base de regras do CFH, porém tratar a variável artificial é uma tarefa bastante complexa, pois é uma variável que não pertence ao problema e para que o CFH obtenha o bom desempenho essa variável tem que ser refinada. O CFH também forneceu comandos que fizeram o robô seguir a parede e se movimentar de forma suave alcançando o mesmo desempenho do CLF com menor custo computacional, como pode ser constatado pelas similaridades de trajetórias em relação ao fuzzy convencional.

O sistema de controle proporcional alcançou o objetivo desejado. Apesar deste sistema de controle não apresentar um

comportamento tão suave, é o que apresenta implementação mais simples, além de não necessitar de conhecimento especialista ou do modelo matemático do robô.

V. CONCLUSÃO

Em navegação de um robô móvel, o comportamento de um seguidor de paredes atribui ao robô a habilidade de seguir e contornar objetos. Para conseguir tal comportamento, o robô deve estar equipado com um controlador que ofereça menor risco de colisão e complete seu trajeto em menor tempo possível. Neste trabalho, a DMRP foi utilizada com o propósito de medir a proximidade que o robô está do objeto detectado. Dessa maneira, o controlador que apresentasse menor DMRP seria aquele com maior risco de colisão. Para a VLM, o controlador que executasse a trajetória em um menor tempo, seria o controlador que apresentasse maior velocidade linear. Ainda com relação a colisão, o robô precisa executar curvas mais suaves para aumentar sua segurança. Para alcançar esse objetivo, o robô deve executar uma trajetória com menor angulação possível em relação ao seu eixo central. A VAM é a métrica utilizada aqui para avaliar a suavidade da trajetória executada pelo robô nas curvas.

A partir da análise, notou-se que todos os controladores propostos chegaram ao objetivo desejado. Diante do problema exposto, observou-se as contribuições de cada controlador, o CLF lida com incertezas e ruídos do ambiente, o SFH tem uma base de regras menores do que a do CLF possibilitando assim um menor custo computacional, e o controle PID é o mais simples de ser implementado. Além disso, em todos os cenários, o SFH apresentou DMRP, VLM e VAM muito próximo do CLF. Indicando assim, menor risco de colisão e menor tempo na execução da trajetória, porém, com menor custo computacional. Concluindo que o controlador que apresenta o menor risco de colisão e o menor tempo de execução

da trajetória para o seguidor de paredes nos cenários aqui propostos é o SFH.

[23] "Getting started with arduino," <https://www.arduino.cc/en/Guide/HomePage>, 2003, online; Acessado em 10/05/2016.

REFERÊNCIAS

- [1] C.-H. Kuo et al., "Development of a fuzzy logic wall following controller for steering mobile robots," in *International Conference on Theory and Its Applications*. IEEE, 2013, pp. 7-12.
- [2] K. Ogata, *Modern control engineering*. Prentice Hall PTR, 2010.
- [3] Y. Huang and S. Yasunobu, "A general practical design method for fuzzy pid control from conventional pid control," in *The Ninth IEEE International Conference on Fuzzy Systems*, vol. 2. IEEE, 2000, pp. 969-972.
- [4] C. Rusu, I. Birou, and E. Szöke, "Fuzzy based obstacle avoidance system for autonomous mobile robot," in *2010 IEEE International Conference on Automation Quality and Testing Robotics*, vol. 1. IEEE, 2010, pp. 1-6.
- [5] M. Boujelben, C. Rekik, and N. Derbel, "Hierarchical fuzzy controller to avoid mobile obstacle for a mobile robot," in *013 10th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2013, pp. 1-8.
- [6] M. Zhuang and D. Atherton, "Automatic tuning of optimum pid controllers," in *IEE Proceedings D-Control Theory and Applications*, vol. 140, no. 3. IET, 1993, pp. 216-224.
- [7] Q.-G. Wang, T.-H. Lee, H.-W. Fung, Q. Bi, and Y. Zhang, "Pid tuning for improved performance," *IEEE Transactions on control systems technology*, vol. 7, no. 4, pp. 457-465, 1999.
- [8] P. Cominos and N. Munro, "Pid controllers: recent tuning methods and design to specification," *IEE Proceedings-Control Theory and Applications*, vol. 149, no. 1, pp. 46-53, 2002.
- [9] C. C. Hang, K. J. Astrom, and W. K. Ho, "Refinements of the ziegler-nichols tuning formula," in *IEE Proceedings D-Control Theory and Applications*, vol. 138, no. 2. IET, 1991, pp. 111-118.
- [10] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559-576, 2005.
- [11] K. J. Åström and T. Hägglund, *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.
- [12] K. J. Åström and T. Hägglund, "Pid controllers: Theory, design, and tuning," *Instrument Society of America, Research Triangle Park, NC*, vol. 10, 1995.
- [13] R. C. Dorf and R. H. Bishop, "Modern control systems," 1998.
- [14] W. Gueaieb and M. S. Miah, "An intelligent mobile robot navigation technique using rfid technology," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 9, pp. 1908-1917, 2008.
- [15] R. Tanscheit, "Sistemas fuzzy," *Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro*, 2004.
- [16] M.-L. Lee, H.-Y. Chung, and F.-M. Yu, "Modeling of hierarchical fuzzy systems," *Fuzzy sets and systems*, vol. 138, no. 2, pp. 343-361, 2003.
- [17] S. Yang-zhi, X. Shi-de, H. Meng-jie, and P. Shao-fei, "Study on obstacle avoidance for intelligent robot based on hierarchical fuzzy control," in *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*. IEEE, 2015, pp. 525-528.
- [18] D. F. Wolf, E. do Valle Simões, F. S. Osório, and O. T. Junior, "Robótica móvel inteligente: Da simulação às aplicações no mundo real," in *Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC*, 2009, p. 13.
- [19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [20] A. J. de Oliveira Alves, "efll - a fuzzy library for arduino and embedded systems" <http://www.zerokol.com/2012/09/arduinfuzzy-fuzzy-library-for-arduino.html>, 2012, [Online; Acessado em 10/05/2016].
- [21] T. Aoki, M. Matsuno, T. Suzuki, and S. Okuma, "Motion planning for multiple obstacles avoidance of autonomous mobile robot using hierarchical fuzzy rules," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 1994, pp. 265- 271.
- [22] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1321- 1326.