



Universidade do Estado do Rio de Janeiro

Instituto Politécnico


Felipe Sertã Abicalil

**Controle com lógica Fuzzy e Neurofuzzy aplicada à análise e
programação de robôs móveis com visualização e simulação 3D**

Nova Friburgo
2007

Felipe Sertã Abicalil

Controle com lógica Fuzzy e Neurofuzzy aplicada à análise e programação de robôs móveis com visualização e simulação 3D



Dissertação apresentada como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Modelagem Computacional do Instituto Politécnico, da Universidade do Estado do Rio de Janeiro.

Orientador: Prof. Carlos Alberto Fialho Thompson Leite

Nova Friburgo
2007

CATALOGAÇÃO NA FONTE
UERJ/REDE SIRIUS/BIBLIOTECA CTC/E

A148

Abicalil, Felipe Sertã.

Controle com lógica Fuzzy e Neurofuzzy aplicada à análise e programação de robôs móveis com visualização e simulação 3D / Felipe Sertã Abicalil. - 2007.
108 f.: il.

Orientador: Carlos Alberto Fialho Thompson Leite.
Dissertação (Mestrado) - Universidade do Estado do Rio de Janeiro, Instituto Politécnico.

1. Robótica - Teses. 2. Lógica difusa – Teses.
3. Imagem tridimensional – Teses. 4. Inteligência artificial – Teses. .I. Leite, Carlos Alberto Fialho Thompson. II. Universidade do Estado do Rio de Janeiro. Instituto Politécnico. III. Título.

CDU 007.52

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação.

Assinatura

Data

Felipe Sertã Abicalil

Controle com lógica Fuzzy e Neurofuzzy aplicada à análise e programação de robôs móveis com visualização e simulação 3D

Dissertação apresentada como requisito parcial para obtenção do título de Mestre, ao Programa de Pós - Graduação em Modelagem Computacional do Instituto Politécnico, da Universidade do Estado do Rio de Janeiro.

Aprovada em 30 de agosto de 2007

Banca examinadora:

Carlos Alberto Fialho Thompson Leite, Ph.D. (Orientador)
Universidade do Estado do Rio de Janeiro. Instituto Politécnico

Washington Braga Filho, Ph.D.
Pontifícia Universidade Católica do Rio de Janeiro

Carlos Frederico Estrada Alves, D.Sc.
Universidade do Estado do Rio de Janeiro. Instituto Politécnico

Nova Friburgo
2007

DEDICATÓRIA

Dedico este trabalho à minha Mãe Lourdinha e
ao meu irmão Fábio, pelo apoio imprescindível
para a conclusão deste trabalho.

AGRADECIMENTOS

Agradeço a Deus por me dar força e perseverança nos momentos mais difíceis.

À minha família pelo apoio, durante a elaboração deste trabalho.

Ao meu orientador, Ph.D. Carlos A. Thompson , pela oportunidade e por todo o conhecimento compartilhado que permitiu a realização deste trabalho.

Aos funcionários da Secretaria da PGM-C-IPRJ/UERJ pela atenção e a sempre presente disposição de ajudar.

Aos funcionários da biblioteca do IPRJ, pelo bom atendimento as necessidades.

Aos colegas de trabalho, por todo o apoio dado.

A todos os que direta ou indiretamente contribuíram para a realização deste trabalho.

RESUMO

ABICALIL, Felipe Sertã. Controle com lógica Fuzzy e Neurofuzzy aplicada à análise e programação de robôs móveis com visualização e simulação 3D. 2007. 108 f. Dissertação (Mestrado em Modelagem Computacional) – Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2007.

Este trabalho tem como objetivo o estudo de uma área da robótica chamada robótica móvel. Um robô móvel deve realizar uma navegação segura e esta é a principal motivação deste trabalho. Para tal foi desenvolvido um simulador de robótica móvel com visualização em 3D. Um dos grandes interesses na área de robótica móvel é a utilização de algoritmos de inteligência artificial. O objetivo deste trabalho é a utilização e simulação de inteligência artificial para o controle destinado ao desvio de obstáculos. As simulações são dinâmicas, ou seja, o robô não tem informação previa do cenário. Os algoritmos de inteligência artificial implementadas neste trabalho são lógica Fuzzy e Neurofuzzy. As contribuições do simulador são: a simulação e visualização em 3D com o cenário modelado em um programa CAD/3D, permite testar diversas configurações antes de testar o robô real, simula o ruído de sensores, utiliza lógica fuzzy e neurofuzzy para o desvio de obstáculos. Os resultados mostram a capacidade do sistema fuzzy para lidar com os dados ruidosos dos sensores assim como a influência das variáveis antecedentes e conseqüentes do sistema fuzzy de no comportamento do robô móvel para o desvio de obstáculos além da capacidade do sistema neurofuzzy de aprender a partir dos dados de treinamento mostrando uma melhoria no resultado das simulações.

PALAVRAS-CHAVE: Robótica móvel; Lógica Fuzzy; Lógica Neurofuzzy; Visualização 3D.

ABSTRACT

This work has as objective the study of an area of the robotics named mobile robotics. A mobile robot must navigate in a safe way and this is the main motivation of this work. To do that a mobile robotics simulator with 3D visualization was developed. One of the great interests in mobile robotics is using artificial intelligence algorithms. The main point of this work is using and simulate artificial intelligence applied in obstacle avoidance control. The simulations are dynamics it means that the robot do not have previous information about the scenery. The artificial intelligence algorithms developed in this work are Fuzzy and Neurofuzzy logics. The simulator contributions are that the simulation and 3D visualization where the scenery is a 3D model from a CAD/3D software besides allows to test many configurations before testing the real robot and simulates noise from sensors and uses fuzzy and neurofuzzy logics to obstacle avoidance. The results show the fuzzy system capability to deal with the noisy data from sensors and how fuzzy variables influences the mobile robot behavior in obstacle avoidance besides the ability of neurofuzzy system to learn from training data showing improvements in the simulation results.

Keywords: Mobile robotics; Fuzzy logic; Neurofuzzy logic; 3D visualization.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Esquema gráfico da inferência crisp-fuzzy | 15 |
| Figura 2 - Diagrama Esquemático de um Sistema Neurofuzzy | 17 |
| Figura 3 – Imagem de Protótipo de Robô Móvel | 20 |
| Figura 4 - Representação das Entradas do Sistema de Controle | 22 |
| Figura 5 - Representação do Obstáculo e Reação do Robô | 22 |
| Figura 6 - Variável Antecedente Distância de Entrada 28 | 28 |
| Figura 7 – Variável Antecedente Ângulo de Entrada 28 | 28 |
| Figura 8 – Variável Conseqüente Velocidade de Saída 29 | 29 |
| Figura 9 – Variável Conseqüente Velocidade Angular de Saída 29 | 29 |
| Figura 10 – Simulação 01 | 32 |
| Figura 11 – Simulação 02 | 32 |
| Figura 12 – Simulação 03 | 33 |
| Figura 13 – Simulação 04 | 34 |
| Figura 14 – Distância percorrida nas simulações de 1-4 | 34 |
| Figura 15 – Velocidade Média nas simulações de 1-4 | 35 |
| Figura 16 – Velocidade Angular Média nas simulações de 1-4 | 35 |
| Figura 17 – Iterações nas simulações de 1-4 | 36 |
| Figura 18 – Trajetórias das simulações de 1-4 | 37 |
| Figura 19 - Simulação 05 | 38 |
| Figura 20 - Simulação 06 | 39 |
| Figura 21 - Simulação 07 | 39 |
| Figura 22 - Simulação 08 | 40 |
| Figura 23 - Simulação 09 | 41 |
| Figura 24 – Distância percorrida nas simulações de 5-9 | 41 |
| Figura 25 – Velocidade Média nas simulações de 5-9 | 42 |
| Figura 26 – Velocidade Angular Média nas simulações de 5-9 | 42 |
| Figura 27 - Iterações nas simulações de 5-9 | 43 |
| Figura 28 – Trajetórias das simulações de 5-9 | 44 |
| Figura 29 - Simulação 10 | 45 |
| Figura 30 - Variável Antecedente Distância de Entrada | 45 |
| Figura 31 - Simulação 11 | 46 |
| Figura 32 – Variável Antecedente Distância de Entrada | 46 |
| Figura 33 - Simulação 12 | 47 |
| Figura 34 – Variável Antecedente Distância de Entrada | 48 |
| Figura 35 - Simulação 13 | 49 |
| Figura 36 – Variável Antecedente Distância de Entrada | 49 |
| Figura 37 - Simulação 14 | 50 |
| Figura 38 – Variável Antecedente Distância de Entrada | 50 |
| Figura 39 – Distância percorrida nas simulações de 10-14 | 51 |
| Figura 40 – Velocidade Média nas simulações de 10-14 | 51 |
| Figura 41 – Velocidade Angular Média nas simulações de 10-14 | 52 |
| Figura 42 – Iterações nas simulações de 10-14 | 52 |
| Figura 43 – Trajetórias das simulações de 10-14 | 53 |
| Figura 44 – Variável Antecedente Ângulo de Entrada | 54 |
| Figura 45 - Simulação 15 | 55 |
| Figura 46 – Variável Antecedente Ângulo de Entrada | 55 |

| | |
|---|----|
| Figura 47 - Simulação 16 | 56 |
| Figura 48 – Variável Antecedente Ângulo de Entrada | 56 |
| Figura 49 – Distância percorrida nas simulações 13,15 e 16 | 57 |
| Figura 50 – Velocidade Média nas simulações 13,15 e 16 | 57 |
| Figura 51 – Velocidade Angular Média nas simulações 13,15 e 16 | 58 |
| Figura 52 – Iterações nas simulações 13,15 e 16 | 58 |
| Figura 53 – Trajetórias das simulações 13,15 e 16 | 59 |
| Figura 54– Variável Conseqüente Velocidade de Saída | 60 |
| Figura 55 - Simulação 17 | 61 |
| Figura 56 – Variável Conseqüente Velocidade de Saída | 61 |
| Figura 57 - Simulação 18 | 62 |
| Figura 58 – Variável Conseqüente Velocidade de Saída | 62 |
| Figura 59 – Distância percorrida nas simulações 10,17 e 18 | 63 |
| Figura 60 – Velocidade Média nas simulações 10,17 e 18 | 63 |
| Figura 61 – Velocidade Angular Média nas simulações 10,17 e 18 | 64 |
| Figura 62 – Iterações nas simulações 10,17 e 18 | 64 |
| Figura 63 – Trajetórias das simulações 10,17 e 18 | 65 |
| Figura 64 – Variável Conseqüente Velocidade Angular de Saída | 66 |
| Figura 65 - Simulação 19 | 67 |
| Figura 66 – Variável Conseqüente Velocidade Angular de Saída | 67 |
| Figura 67 - Simulação 20 | 68 |
| Figura 68 – Variável Conseqüente Velocidade Angular de Saída | 68 |
| Figura 69 – Distância percorrida nas simulações 10,19 e 20 | 69 |
| Figura 70 – Velocidade Média nas simulações 10,19 e 20 | 69 |
| Figura 71 – Velocidade Angular Média nas simulações 10,19 e 20 | 70 |
| Figura 72 – Iterações nas simulações 10,19 e 20 | 70 |
| Figura 73 – Trajetórias das simulações 10,19 e 20 | 71 |
| Figura 74 - Variável Antecedente Distância de Entrada | 72 |
| Figura 75 – Variável Antecedente Ângulo de Entrada | 73 |
| Figura 76 - Simulação 21 | 74 |
| Figura 77 - Simulação 22 | 75 |
| Figura 78 – Variável Antecedente Distância de Entrada | 75 |
| Figura 79 – Variável Antecedente Ângulo de Entrada | 76 |
| Figura 80 - Simulação 23 | 77 |
| Figura 81 - Simulação 24 | 79 |
| Figura 82 – Distância percorrida nas simulações 21, 22, 23 e 24 | 79 |
| Figura 83 – Velocidade Média nas simulações 21, 22, 23 e 24 | 80 |
| Figura 84 – Velocidade Angular Média nas simulações 21, 22, 23 e 24 | 80 |
| Figura 85 – Iterações nas simulações 21, 22, 23 e 24 | 81 |
| Figura 86 – Trajetórias das simulações 21, 22, 23 e 24 | 82 |
| Figura 87 – Treinamento neurofuzzy | 83 |
| Figura 88 - Simulação 25 | 84 |
| Figura 89 – Variável Antecedente Distância de Entrada | 85 |
| Figura 90 – Variável Antecedente Ângulo de Entrada | 85 |
| Figura 91 - Simulação 26 | 86 |
| Figura 92 - Simulação 27 | 87 |
| Figura 93 – Variável Antecedente Distância de Entrada | 88 |
| Figura 94 – Variável Antecedente Ângulo de Entrada | 88 |

| | |
|---|-----|
| Figura 95 - Simulação 28 | 89 |
| Figura 96 – Distância percorrida nas simulações 22, 25, 26, 27 e 28 | 90 |
| Figura 97 – Velocidade Média nas simulações 22, 25, 26, 27 e 28 | 90 |
| Figura 98 – Velocidade Angular Média nas simulações 22, 25, 26, 27 e 28 | 91 |
| Figura 99 – Iterações nas simulações 22, 25, 26, 27 e 28..... | 91 |
| Figura 100 – Trajetórias das simulações 22, 25, 26, 27 e 28 | 92 |
| Figura 101 – Treinamento neurofuzzy | 93 |
| Figura B.1 – Simulação 16 | 103 |
| Figura B.2 – Simulação 16 | 104 |
| Figura B.3 – Simulação 16 | 104 |
| Figura B.4 – Simulação 16 | 105 |
| Figura B.5 – Simulação 16 | 105 |
| Figura B.6 – Simulação 16 | 106 |
| Figura B.7 – Simulação 16 | 107 |
| Figura B.8 – Simulação 16 | 107 |
| Figura B.9 – Simulação 16 | 108 |

SUMÁRIO

| | |
|---|-----------|
| Capítulo 1 – INTRODUÇÃO | 1 |
| 1.1 - A Robótica Móvel | 1 |
| 1.2 - Lógica Fuzzy Aplicada à Robótica Móvel | 10 |
| Capítulo 2 – FUNDAMENTAÇÃO TEÓRICA | 13 |
| 2.2 A Lógica Neurofuzzy | 16 |
| 2.3 Lógica Fuzzy aplicada ao Robô Móvel | 20 |
| 2.4 O Simulador de Robótica Móvel | 23 |
| Capítulo 3 – RESULTADOS | 27 |
| Capítulo 4 – CONCLUSÕES | 94 |
| Capítulo 5 – TRABALHOS FUTUROS | 97 |
| Referências Bibliográficas | 98 |
| APÊNDICE | 99 |

Capítulo 1 – INTRODUÇÃO

1.1 - A Robótica Móvel

A difusão dos computadores pessoais ocorrida nas últimas duas décadas aproximou as mais diversas pessoas e profissões de uma nova ferramenta de interação com o mundo ao seu redor. A princípio, os programas seguiam os modelos pré-existent, tal como o processamento de texto segue a máquina de escrever ou o desenho arquitetônico e técnico por computador (CAD) segue o desenho tradicional a duas dimensões com o qual era dimensionado. Contudo, com a difusão dos computadores, outros modos de representação e operação com a realidade surgiram, ultrapassando largamente tudo o que se conhecia até então. Entre muitos exemplos possíveis, destacam-se os jogos de computador e a realidade virtual. Ao mesmo tempo, a Robótica tem evoluído a um ritmo ainda mais rápido, embora menos visível, e é hoje possível construir robôs sofisticados, com um grau de controle autônomo notável a partir de uma caixa de um brinquedo, por exemplo, ou enviar um robô móvel a Marte capaz de administrar sozinho a sua missão e enviar dados para a base. O terceiro elemento desta revolução tecnológica foi a banalização de equipamentos eletrônicos de qualidade superior e computadores com poder de cálculo cada vez maior, o que permite trazer o processamento de imagens a três dimensões (3D) em tempo real para os computadores domésticos. A Internet estendeu tornou o acesso à informação independente da distância.

Os modelos 3D descrevem um ambiente como um conjunto de superfícies adjacentes. Cada superfície é descrita pela sua posição no espaço euclidiano. Além da posição, a descrição das superfícies contém texturas que reproduzem o aspecto

fotográfico das mesmas. Depois de muito tempo, plantas e perspectivas que só os arquitetos e engenheiros percebiam como um volume tridimensional com objetos, criaram-se finalmente ferramentas que permitem ao usuário comum visualizar o espaço tridimensional de uma forma mais completa. É possível usar estes modelos 3D como base de informação espacial de um sistema de Realidade Virtual.

A robótica tem cada vez mais aplicações. Muitos protótipos e novos métodos de controle têm sido utilizados. Testes de laboratório demandam tempo e risco de colisões e acidentes com os robôs. O desenvolvimento de softwares de simuladores de robótica traz uma grande ferramenta para novas aplicações. Com os simuladores é possível testar novos algoritmos de controle e a inteligência artificial aos robôs. A robótica móvel é uma área de grande importância para o desenvolvimento tecnológico.

Uma das principais ações que um robô móvel autônomo deve ser capaz de realizar é desviar de obstáculos enquanto navega por um ambiente. Sensores podem permitir a identificação de obstáculos e a tomada de decisões para evitar possíveis colisões. Atualmente, novas aplicações na robótica têm surgido exigindo cada vez mais autonomia, mobilidade, adaptabilidade e interação dos robôs com o ambiente e com outros agentes. Para isso, os robôs móveis devem ser capazes de navegar e se localizar no ambiente, evitando obstáculos enquanto se movimentam. O sistema de navegação de um robô móvel deve chegar ao seu destino final evitando acidentes ocasionados por colisões ou quedas. Normalmente, os obstáculos encontrados podem ser fixos (paredes, mesas, armários) ou móveis (pessoas que se deslocam no ambiente). Realizar uma navegação segura é a principal motivação deste trabalho. A intenção é implementar um método de desvio de obstáculos baseado em sensores de distância que possa ser adicionado a estratégias de navegação.

Um sistema embarcado pode ser considerado um sistema computacional com propósitos específicos, normalmente construído em dimensões reduzidas, que deve funcionar de forma autônoma. Um requisito normalmente exigido de sistemas embarcados é a execução em tempo real, em especial quando a aplicação envolve

veículos autônomos inteligentes. Por exemplo, sistemas de freio ABS, que impedem o bloqueio completo das rodas durante a frenagem, devem analisar as informações fornecidas pelo sistema (velocidade, força de acionamento dos pedais de freio, etc.) e produzir resultados (frenagem que será efetivamente aplicada a cada uma das rodas) em uma pequena fração de tempo.

A robótica é uma área do conhecimento que tem evoluído de forma muito rápida nos últimos anos, entretanto, o estudo, o projeto e a implementação de robôs e autômatos vêm sendo desenvolvidos há algumas décadas.

Os primeiros robôs eram, na verdade, autômatos complexos, verdadeiras obras de arte mecânicas, que executavam tarefas de modo repetitivo. Estes sistemas deram origem aos atuais braços manipuladores de base fixa, que atualmente são largamente adotados na indústria (e.g. indústria automotiva). Somente mais recentemente surgiram os robôs móveis, que se caracterizam pela sua capacidade de se deslocar de modo guiado, semi-autônomo ou totalmente autônomo.

Portanto, o desenvolvimento da robótica evoluiu dos autômatos, passando pelos robôs manipuladores de base fixa, pelos dispositivos móveis guiados a distância, chegando mais recentemente aos robôs móveis semi-autônomos e mesmo os totalmente autônomos.

Tipos de Autômatos, Robôs e Veículos Robóticos

Funcionalidade:

Tipos de Aplicações [11]

- Mecanismos eletro-mecânicos com movimento repetitivo pré-definido e fixo (autômatos)
- Manipuladores: Braços de base fixa (e.g. robôs industriais)
- Manipuladores: Braços de base móvel (e.g. gruas robóticas)
- Robôs Móveis (terrestres, aquáticos, exploração espacial, dentre outros)

Mobilidade:

Dispositivos de Locomoção [11]

- Robôs móveis com deslocamento com rodas
- Robôs móveis com deslocamento com esteiras
- Robôs com deslocamento por propulsão (e.g. no ar, na água)
- Robôs com deslocamento por pernas (e.g. *animats*, bípedes)

Autonomia:

Grau de Inteligência do Robô [11]

- Totalmente comandados a distância (e.g. carro de “controle remoto” = tele-comandado, tele-operado)
- Veículos guiados: dependem de informações externas, sem as quais não podem operar (AGV – *Automated Guided Vehicle*)
- Veículos semi-autônomos: operam controlados remotamente, mas possuindo um certo grau de autonomia (e.g. podem parar de modo a evitar o choque com obstáculos)
- Veículos autônomos: operam sem intervenção humana, de forma totalmente automatizada. Uma vez iniciada sua operação, executam tarefas de modo autônomo.

Em meio a tantos tipos distintos de robôs e veículos, conforme descrito acima, com diferentes recursos e níveis de autonomia, é importante que se busque identificar e organizá-los em categorias outros aspectos importantes de um robô móvel são descritos logo abaixo [11].

Locomoção: A maneira como o robô irá se deslocar no ambiente. Os dispositivos de locomoção são de grande importância na caracterização de um robô móvel;

Percepção: Como o robô irá perceber o ambiente (e.g. sensores de contato – *bumpers* - e de distância em relação a obstáculos) e inclusive monitorar parâmetros próprios dele e de seu comportamento (e.g. carga da bateria, odometria);

Controle e Inteligência: Como o robô irá transformar suas percepções e conhecimentos prévios adquiridos (e.g. mapa do ambiente) em ações, ou seqüências de ações, a serem executadas;

Comunicação: Como o robô irá se comunicar com um operador humano, ou mesmo com outros dispositivos robóticos;

As pesquisas em robótica móvel têm avançado na direção de uma maior integração e aperfeiçoamento destes 4 aspectos descritos acima, cuja meta principal é o desenvolvimento de modelos cada vez mais sofisticados e robustos dos chamados **RMA**s - **Robôs Móveis Autônomos**. Os RMA's possuem portanto, como características fundamentais, as capacidades de locomoção e de operação de modo semi ou completamente autônomo. Também deve ser considerado que maiores níveis de autonomia serão alcançados a medida de que o robô passe a integrar 2 outros aspectos considerados da maior importância: robustez (capacidade de lidar com as mais diversas situações) e inteligência (de modo a resolver e executar tarefas por mais complexas que sejam).

Os robôs e veículos móveis devem ser dotados de atuadores, que serão os responsáveis pela execução de suas ações e, principalmente, pelo seu deslocamento no ambiente. Existem diferentes tipos de dispositivos usados na locomoção dos robôs, e conforme a solução/implementação adotada o veículo terá diferentes tipos de comportamento. Podem ser citados como exemplo os robôs providos com duas rodas laterais e acionadores independentes acoplados a cada uma delas, obtendo-se, assim, um veículo de *tração diferencial*.

A percepção é um dos componentes principais dos robôs móveis, pois é através dos sensores que pode ser garantido ao sistema um maior nível de autonomia e robustez. Os sensores empregados junto à robótica móvel são de diferentes tipos, onde alguns servem para se ter uma realimentação mais direta do resultado de uma ação, como por exemplo, os *encoders*. A “ordem” de execução de um comando não garante que o resultado perfeito e completo desta ação seja atingido, por exemplo, uma mesma tensão de acionamento aplicada a um motor, durante um mesmo período de tempo, poderá ser afetada pela inclinação do terreno, em termos da distância total percorrida (isto sem considerar questões relacionadas à curva de resposta a aceleração e o torque do motor). O uso de um *encoder* permite que seja medido o real deslocamento (leia-se giro) das rodas de um robô móvel. Os principais dispositivos sensores são descritos abaixo:

Encoder (de rodas): Mede o número de rotações das rodas, e permite que se obtenha informações de odometria (medida de deslocamento).

Sensor de Infravermelho: Mede a distância entre o sensor e um obstáculo posicionado em frente a ele, através da estimativa da distância calculada em função do retorno da reflexão da luz na superfície (medida de distância).

Sensor Laser: Mede a distância entre o sensor e um obstáculo posicionado em frente – sensor direcional de alta precisão (medida de distância).

Sonar (ultra-som): Estima a distância entre o sensor e os obstáculos ao seu redor, através da reflexão sonora. É um sensor mais sensível a perturbações, obtendo medidas aproximadas (medida de distância).

Sensores de Contato (bumpers): Identifica quando ocorre uma colisão entre o veículo (sensor) e um ponto de contato (medida de contato/pressão).

Bússola Eletrônica: Identifica a orientação do veículo em relação ao campo magnético da terra (medida de orientação – posição relativa).

GPS: Identifica a posição absoluta do veículo no globo terrestre, baseando-se na rede de satélites GPS – *Global Positioning System* (medida de posicionamento absoluto).

Visão Artificial: O uso de imagens permite que sejam adquiridas a partir de uma ou mais câmeras (visão monocular, estéreo ou omnidirecional) descrições do ambiente (imagem monocromática ou colorida). As imagens permitem que se implemente técnicas de determinação de posicionamento relativo, posicionamento absoluto, detecção e estimativa de deslocamento, assim como detecção de obstáculos. Outros sensores Acelerômetros, Giroscópios, Sensores de Inclinação, Radar, dentre outros.

O adequado uso dos sensores, bem como a correta interpretação dos dados fornecidos por estes (modelo sensorial, incluindo precisão, distância alcançada, suscetibilidade ao ruído), é que permitirá a correta implementação, ou simulação, de sistemas RMA (Robôs Móveis Autônomos) mais robustos. Um outro tema de grande relevância na robótica é a integração de sensores (fusão sensorial), de modo a integrar e explorar a complementaridade entre os diferentes sensores, que possuem diferentes escalas de sensibilidade e confiabilidade (e.g. fusão de dados de um radar, sonar e sensor laser).

Um sistema robótico móvel pode ter diferentes níveis de autonomia e inteligência, o que será definido pelo tipo de sistema de controle integrado ao sistema. Existem sistemas RMAs que possuem um nível de autonomia e inteligência extremamente limitado, como por exemplo os AGVs convencionais industriais. Um AGV (Automated Guided Vehicle) industrial é um robô móvel que usualmente segue uma marcação preestabelecida (e.g. uma faixa pintada no solo), e mesmo se este possui integrada uma capacidade de se deslocar de modo autônomo (locomoção) e de perceber o ambiente (sensor da faixa), seu sistema de controle depende desta informação para poder continuar atuando. Se um pedaço da faixa for danificado, o robô usualmente não terá como prosseguir sua tarefa e deverá parar de se movimentar. O mesmo acontece caso este robô detecte um obstáculo em seu caminho, ele irá parar e aguardar que o obstáculo seja

removido. Nota-se que os níveis de robustez, autonomia e inteligência de um veículo deste tipo são bastante reduzidos, não sendo capaz de suportar falhas na marcação ou de desviar de obstáculos e retornar para a execução de sua tarefa. Em função disto, as pesquisas em robótica móvel autônoma [11] têm avançado na direção de propor sistemas de controle, que incluam “comportamentos inteligentes”, de modo a tornar os robôs mais confiáveis, robustos e menos dependentes da intervenção de seres humanos. Este “comportamento inteligente” é implementado através de sistemas de controle que devem, entre outras funções, realizar tarefas como as descritas a seguir:

- Garantir a preservação da integridade física dos seres humanos, bem como não causar danos ao ambiente onde está inserido;
- Garantir a preservação da integridade física do robô: evitar colisões contra pessoas e objetos (obstáculos estáticos ou móveis);
- Garantir a manutenção de informações que permitam uma melhor execução das tarefas, o que pode incluir: construção e atualização de um mapa do ambiente, determinação da localização do robô em relação a este mapa, definição de trajetórias que permitam que ele se desloque de um ponto a outro (considerando o mapa disponível);
- Integrar as informações provenientes de diferentes sensores, interpretando estas informações e levando em consideração problemas de ruído e erros associados às mesmas;
- Gerar os comandos, na seqüência correta, de modo a realizar passo-a-passo as tarefas que lhe são atribuídas;
- Definir soluções alternativas para situações imprevistas que se apresentem e que por ventura possam prejudicar a execução das ações que estavam planejadas;
- Capacidade de se adaptar, aprender com as experiências passadas e corrigir seus erros e, quem sabe no futuro, ter a capacidade de evoluir.

A tecnologia *drive-by-wire* representa uma nova era, na qual soluções mecânicas serão substituídas por soluções eletro-mecânicas. A possibilidade de se dirigir um veículo a partir de sinais eletrônicos permite o uso de computadores embarcados atuando como

co-pilotos e ajudando o motorista em situações de emergência, ou na execução de tarefas, tais como estacionar um veículo.

Através dos equipamentos, como *scanners* a laser, ultra-som, radar e visão estereoscópica, diferentes obstáculos podem ser detectados, fazendo com que o veículo pare ou desvie, evitando assim colisões, que representam o principal tipo de acidente de trânsito com vítimas. Ter a redução dos acidentes nas estradas, com a conseqüente redução no número de vítimas como meta, mostra a importante contribuição que o desenvolvimento de tecnologias para a automação veicular pode trazer para a sociedade. Os sensores de navegação, como bússola, GPS, giroscópio, acelerômetros e sensores de velocidade das rodas, garantem, por sua vez, que os sistemas mecatrônicos existentes na base hierárquica da estrutura de controle permitam o veículo trafegar de forma segura e controlada, mantendo sua estabilidade dinâmica e o seguimento seguro da trajetória previamente determinada.

O sistema computacional de controle deve executar tarefas como: preservar a integridade do veículo, preservar a integridade dos objetos e entes presentes no ambiente onde este veículo atua, planejar/executar/encontrar soluções para a execução de tarefas, e em alguns casos deve inclusive interagir com outros sistemas. As características de um tal sistema de controle nos remete ao estudo de técnicas que vem sendo abordadas junto ao domínio da Inteligência Artificial (I.A.).

As arquiteturas computacionais de controle de veículos autônomos são as mais diversas, podem ser citadas algumas das arquiteturas de controle que se tornaram as mais conhecidas e reconhecidas pelas suas características e potencialidades: controle reativo, controle deliberativo e controle hierárquico. O presente trabalho trata de simular um veículo autônomo utilizando controle reativo para desviar de obstáculos utilizando lógica fuzzy.

O controle reativo consiste de um sistema de reação sensorial-motora, onde este tipo de controle normalmente é o mais simples de ser implementado (reativo puro), não

necessitando de muitos recursos computacionais para sua implementação. No controle reativo existe um laço de: (i) leitura dos sensores; (ii) imediato processamento destas informações; (iii) geração de um comando de resposta para os atuadores. Usualmente um esquema de controle reativo considera apenas as leituras sensoriais realizadas no presente para fins de tomada de decisão e geração de comandos de ação. Um sistema reativo é bastante útil para implementar comportamentos como: desviar de obstáculos (*avoid collision behaviour*: reage a presença de um obstáculo), e seguir um objeto (*wall-following/lane-follow behaviour*: acompanhar um elemento guia). Pode ser considerado como um sistema de controle reativo, caso da tomada de decisão do agente-robô não se utilize de mapas, de memória, ou de outras informações adicionais, fazendo uso apenas das informações sensoriais a fim de gerar uma nova ação.

A melhoria e a construção de simuladores robustos de veículos e robôs autônomos vêm sendo buscados por inúmeros grupos de pesquisa no mundo inteiro. Tanto no meio acadêmico, quanto abordagens como a da NASA nos EUA, vêm buscando o desenvolvimento da pesquisa em veículos autônomos através do uso de simuladores e do aperfeiçoamento dos sistemas computacionais de controle destes veículos.

1.2 - Lógica Fuzzy Aplicada à Robótica Móvel

Os robôs estão sendo cada vez mais empregados na indústria, na área de pesquisa e até nas casas. Interagir com ambientes cada vez mais dinâmicos é um dos grandes desafios que pode permitir uma maior difusão da robótica. A movimentação é o grande passo que aumenta a aplicabilidade de sistemas robóticos. A robótica móvel é a área que estuda e desenvolve em sistemas robóticos a capacidade de lidar com ambientes dinâmicos de forma eficiente e com o mínimo risco para a vida humana. Em função disso a robótica móvel é um campo da ciência de grande crescimento e estudo. Novos protótipos de robôs moveis são lançados com a capacidade de interagir com um determinado ambiente. Por exemplo, as pesquisas espaciais necessitam de robôs que possam interagir em ambientes, com informações imprecisas ou cuja modelagem

matemática é complexa, de forma a facilitar e mesmo tornar possível a operação a distância ou mesmo se tornar um sistema autônomo. A lógica fuzzy é utilizada para representar situações vagas e imprecisas cuja solução pode ser alcançada por diferentes caminhos. Existem diversas maneiras de se percorrer um determinado caminho e por meio de uma construção e modelagem mais natural e intuitiva, a lógica fuzzy não necessita de um modelo matemático do sistema. Tratando-se de metodologia fuzzy é o operador do sistema cujo modelo está sendo identificado enquanto ele está controlando o sistema [5].

O mapeamento da variável numérica para simbólica através de uma função(fuzzyficação). O sistema fuzzy permite o raciocínio sobre bases simbólicas (regras nebulosas). O processo de re-mapeamento de simbólica para numérica (de-fuzzyficação).

Um robô móvel que emprega lógica fuzzy para o desvio de obstáculos utiliza de uma base de conhecimento (regras fuzzy) para obter tal comportamento. O desvio de obstáculos é uma das funções mais importantes de um robô móvel. Os sensores do robô móvel são as leituras do ambiente ao redor do robô móvel. O controle do tipo desvio de obstáculos obtém uma entrada vinda dos sensores indicando um possível obstáculo crítico e sua localização aproximada. A localização do obstáculo fornece a informação que o sistema fuzzy necessita para que por meio de sua base de conhecimento possa produzir uma saída que é como reagir ao obstáculo em questão. A saída pode ser por exemplo um comando para os sistemas de controle de velocidade do robô móvel. O entendimento sobre o funcionamento deste tipo de sistema de controle é fundamental no desenvolvimento de uma aplicação real que é o robô móvel.

O objetivo deste trabalho é a utilização e simulação de inteligência artificial para o controle estinado ao desvio de obstáculos. As simulações são dinâmicas, ou seja, o robô não tem informação previa do cenário. As técnicas de inteligência artificial implementadas neste trabalho são lógica Fuzzy e Neurofuzzy. Os resultados das

simulações apresentam a influencia das configurações do controle, nas características e no comportamento do robô móvel.

Capítulo 2 – FUNDAMENTAÇÃO TEÓRICA

2.1 Lógica Fuzzy

Muitas estruturas de controladores fuzzy baseados nos vários métodos de inferência que vêm sendo apresentados. Dentre eles os métodos mais comumente usados na prática estão [5]: o método proposto por Mamdani o qual associara a regra de composição min-max baseado na interpretação das regras de controle como um conjunto de antecedentes e conseqüentes, já o método produto-soma proposto por Mizumoto que sugeriu introduzir o produto e a média aritmética como operadores de agregação para substituir o AND lógico (mínimo) e o OU lógico (máximo) na composição das regras de inferência do método min-max. No algoritmo de um controlador fuzzy, o cálculo da defuzzificação é uma tarefa complicada e que consome tempo. Takagi e Sugeno [5] propuseram um modelo do tipo *crisp* (definido) no qual a parte conseqüente das regras de controle são representações funcionais *crisp* ou números reais *crisp* no caso mais simples ao invés de conjuntos fuzzy. Com o modelo de números reais *crisp* de saída, o conjunto fuzzy conseqüente de inferência será um conjunto discreto com um finito número de pontos, isso pode simplificar o cálculo do algoritmo de defuzzificação.

Os sistemas fuzzy são baseados em regras como

IF Distancia N AND ângulo ZP THEN vel_saida IS M AND velocidade angular IS MN

Ou seja, se a distância é N(near) e o ângulo ZP (zero positive) então a velocidade de saída é M e a velocidade angular é MN. Sendo a distância e o ângulo variáveis de entrada (antecedentes) e a velocidade de saída e a velocidade angular variáveis de saída (consequentes).

A variável distância possui funções de pertinência ou funções membro para representar seus valores fuzzy como: N (Near), M (Medium) e F (Far).

A parte consequente das regras fuzzy podem ser funções de pertinência ou *crisp*. O desenvolvimento abaixo [1] mostra a inferência para consequentes *crisp*, para referencias sobre consequentes com funções, membro ver referencia [5] .

Suponha que as funções membro A_i e B_j são $A_i(e)$ e $B_j(\dot{e})$. Em uma certa instância de controle t , os valores observados de e e \dot{e} são e_t e \dot{e}_t , respectivamente, então os valores reais de A_i e B_j são $A_i(e_t)$ e $B_j(\dot{e}_t)$ ($i \in I, j \in J$). O valor da parte antecedente da regra de controle vai ser:

$$f_{ij} = \begin{cases} A_i(e_t)B_j(\dot{e}_t), & \text{Método Produto-soma} \\ \text{Min}\{A_i(e_t), B_j(\dot{e}_t)\}, & \text{Método Min-max} \end{cases} \quad (i \in I, j \in J) \quad (2.1)$$

Argumentando da parte antecedente para a parte consequente vai gerar um subconjunto conclusão o qual denominados como C. C será um subconjunto discreto com finitos números de pontos. No caso do método produto-soma, $C = \{ f_{ij} / u_{ij}, \mid i \in I, j \in J \}$, onde u_{ij} não é necessariamente diferente dos outros. Usando o método do centróide para defuzzificar o conjunto C, o valor real de saída do controlador u_t é dado por:

$$u_t = \frac{\sum_{i,j} f_{ij} u_{ij}}{\sum_{i,j} f_{ij}} \quad (2.2)$$

E desta forma se obtém o valor de saída de um sistema crisp-fuzzy.

Segue abaixo uma simulação do cálculo de uma inferência fuzzy para um sistema com duas regras tendo duas variáveis de entrada(distancia de entrada e ângulo de entrada) e duas de variáveis de saída(velocidade linear e velocidade angular), sendo as saídas do tipo crisp. Para uma distância de entrada de 40% e um ângulo de entrada de -70° a inferência é calculada conforme ilustra a figura abaixo:

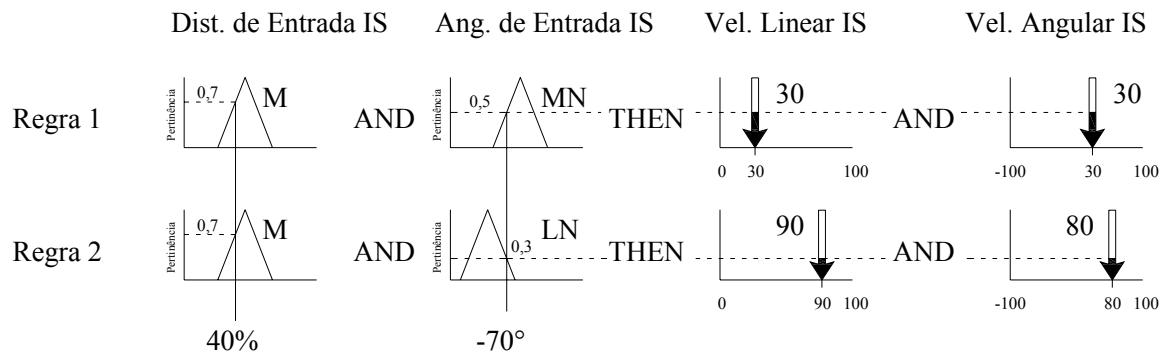


Figura 1 – Esquema gráfico da inferência crisp-fuzzy

Para se calcular as saídas do sistema crisp-fuzzy considerando apenas as duas regras acima se procede da seguinte forma:

$$Velocidade_Linear_de_Saída = \frac{\min(0.7, 0.5) \times 30.0 + \min(0.7, 0.3) \times 90.0}{0.5 + 0.3} = 52.50$$

$$Velocidade_Angular_de_Saída = \frac{\min(0.7, 0.5) \times 30.0 + \min(0.7, 0.3) \times 80.0}{0.5 + 0.3} = 48.75$$

2.2 A Lógica Neurofuzzy

A lógica Neurofuzzy é uma das tecnologias de inteligência. A lógica neurofuzzy combina a lógica fuzzy com o aprendizado e capacidades adaptativas de redes neurais. Isto significa que o usuário pode inicializar a rede neural usando conhecimento apurado expressado com regras fuzzy, e então treiná-la usando dados numéricos. Após treinar a rede, as regras podem ser avaliadas e editadas para melhorar alguma característica no controlador não incorporado pelos dados de treinamento.

A lógica neurofuzzy traz os benefícios tanto de sistemas de redes neurais quanto de sistemas de lógica fuzzy.

Diferentes arquiteturas de sistemas neurofuzzy têm sido investigadas. Tais arquiteturas têm sido utilizadas em muitas aplicações, especialmente no controle de processos.

Redes neurais e sistemas fuzzy têm sido intensamente aplicados a muitos problemas incluindo identificação de sistema, predição, classificação e controle.

Redes neurais são sistemas "black box" que podem aprender a prever e se adaptar ao mundo real.

Um sistema neurofuzzy é um sistema que utiliza um algoritmo de aprendizado derivado ou inspirado em teoria de redes neurais para determinar seus parâmetros, conjuntos e regras fuzzy, através do processamento de exemplos de dados.

Um sistema neurofuzzy está baseado em um sistema fuzzy que é treinado por um algoritmo de aprendizagem. Um sistema neurofuzzy pode ser sempre interpretado como um sistema de regras fuzzy.

Sistemas neurofuzzy combinam os atributos positivos de lógica fuzzy e redes neurais produzindo sistemas fuzzy com habilidade de aprender e se adaptar ao mundo real.

Um sistema fuzzy e neurofuzzy consiste de algumas camadas como mostra a figura abaixo.

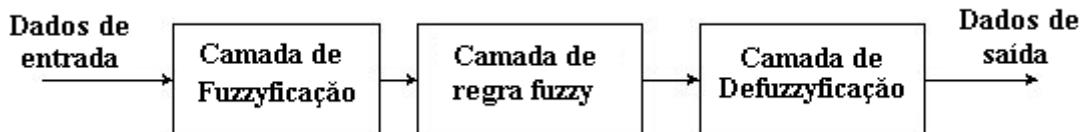


Figura 2 - Diagrama Esquemático de um Sistema Neurofuzzy

A diferença entre o Fuzzy e o Neurofuzzy é que no Neurofuzzy as suas saídas durante treinamento são comparadas a ‘targets’ ou saídas esperadas para uma dada entrada para computar um erro que deve ser minimizado por um algoritmo de aprendizagem.

Para a aprendizagem do sistema Neurofuzzy é necessário um conjunto de dados de treinamento, ou seja, um conjunto de dados de entradas e saídas esperadas. Os dados de treinamento são selecionados em geral de forma randômica ou pseudo-randômica. O erro entre a saída esperada e a saída do sistema neurofuzzy é minimizada por meio de modificações nos parâmetros do sistema neurofuzzy. Um ‘loop’ percorre o número de padrões e cada ‘loop’ é denominado época. É calculado o erro para os padrões de aprendizado a cada época e a parada do algoritmo de aprendizagem neurofuzzy geralmente é feito quando o erro é minimizado ou quando se passarem um determinado número de épocas. O ajuste dos parâmetros é mostrado no desenvolvimento abaixo de acordo com uma taxa de aprendizado.

Desenvolvimento Matemático Neurofuzzy

Suponha um mapeamento não-linear não conhecido a ser realizado por um sistema fuzzy que pode ser representado por

$$y^k = f(x^k) = f(x_1^k, \dots, x_n^k)$$

Onde y representa a saída e x as entradas do mapeamento.

Para $k = 1, \dots, K$ i.e. temos o seguinte conjunto de treinamento

$$\{(x^1, y^1), \dots, (x^K, y^K)\}$$

Para modelar o mapeamento desconhecido f, empregamos regras fuzzy IF-THEN simplificadas do seguinte tipo

$$\mathfrak{R}_i: \text{if } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \text{ then } y = z_i,$$

$i = 1, \dots, m$, onde A_{ij} são funções triangulares e z_i são números reais.

Neste contexto, regras simplificadas significa que as saídas individuais de cada regra são números crisp, portanto podemos utilizar a média ponderada para obter a saída do sistema.

Seja O^k a saída do sistema fuzzy correspondente a entrada x^k .

Suponha a saída da regra i, dado por α_i , é definido pelo operador produto de Larsen

$$\alpha_i = \prod_{j=1}^n A_{ij}(x_j^k)$$

A saída do sistema é dada pelo calculo discreto do centróide

$$o^k = \frac{\sum_{i=1}^m \alpha_i z_i}{\sum_{i=1}^m \alpha_i}$$

e sendo a medida do erro para o padrão de treinamento k como

$$E_k = \frac{1}{2}(o^k - y^k)^2$$

Onde o^k é a saída computada pelo sistema fuzzy correspondente ao padrão de entrada x^k e a saída desejada y^k , $k = 1, \dots, K$.

O método *steepest descent* é utilizado para aprender os parâmetros do sistema como por exemplo

$$z_i(t+1) = z_i(t) - \eta \frac{\partial E_k}{\partial z_i} = z_i(t) - \eta(o^k - y^k) \frac{\alpha_i}{\alpha_1 + \dots + \alpha_m},$$

Onde η é a taxa de aprendizado.

No presente trabalho as funções de pertinência A_i são do tipo gaussianas dadas pela fórmula

$$A_i(x) = e^{-b(x-a_i)^2}$$

Para o aprendizado dos parâmetros ‘a’ e ‘b’ das funções gaussianas de índice 1 seria utilizada uma equação do tipo

$$a_1(t+1) = a_1(t) - \eta \frac{\partial E_k}{\partial a_1},$$

$$b_1(t+1) = b_1(t) - \eta \frac{\partial E_k}{\partial b_1}$$

2.3 Lógica Fuzzy aplicada ao Robô Móvel

O Robô Móvel

O robô simulado neste trabalho possui uma capacidade de giro e variação de velocidade e sensores distribuídos em ao seu redor de forma a obter informações sobre as distâncias de obstáculos em algumas direções em relação ao corpo do robô. A figura abaixo exemplifica um protótipo deste tipo de robô.

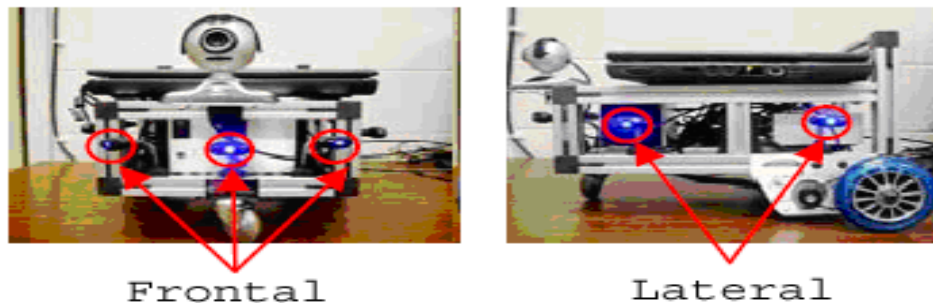


Figura 3 – Imagem de Protótipo de Robô Móvel

As variáveis fundamentais para o sistema de controle

As variáveis utilizadas para o controle do robô móvel no caso de entrada ou antecedentes são o ângulo do sensor (ângulo de entrada) que indica o obstáculo mais próximo e a distância do mesmo (distância de entrada). As variáveis de saída ou conseqüentes seriam a velocidade angular de saída e sua respectiva velocidade linear de saída.

A variável fuzzy antecedente ângulo de entrada representa a direção do obstáculo mais próximo e é definido pelas funções de pertinência LN(Large Negative), MN(Medium Negative), ZN(Zero Negative), ZP(Zero Positive), MP(Medium Positive) e

LP(Large Positive). As funções de pertinência são definidas na faixa de ângulos de entrada de -90 a 90 graus.

A variável fuzzy antecedente distância de entrada representa o quando o obstáculo está afastado do robô móvel e é definido pelas funções de pertinência N(Near), M(Medium) e F(Far). As funções de pertinência são definidas na faixa de 0 a 100 por cento do alcance do sensor.

A variável fuzzy conseqüente velocidade linear de saída representa o comando de velocidade linear para o sistema de controle de velocidade do robô móvel. Poder ser definido pelas funções de pertinência N(Near), M(Medium) e F(Far). A variável conseqüente também pode ser definida por valores crisp dentro da faixa de 0 a 100 por cento da velocidade linear máxima do robô movel para o sistema crisp fuzzy.

A variável fuzzy conseqüente velocidade angular de saída representa o comando de velocidade angular para o sistema de controle de velocidade do robô móvel. Pode ser definido pelas funções de pertinência LN(Large Negative), MN(Medium Negative), ZN(Zero Negative), ZP(Zero Positive), MP(Medium Positive) e LP(Large Positive). A variável conseqüente também pode ser definida por valores crisp dentro da faixa de -100 a 100 por cento da velocidade angular máxima do robô movel para o sistema crisp fuzzy.

A figura abaixo ilustra a configuração espacial das variáveis do sistema de controle.

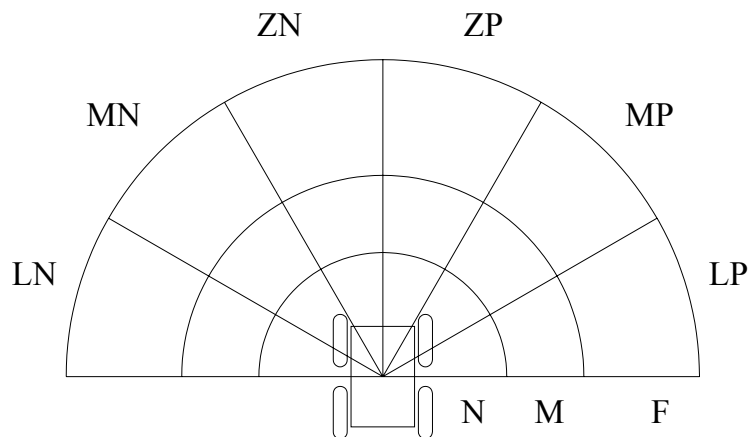


Figura 4 - Representação das Entradas do Sistema de Controle

A figura abaixo ilustra a reação do robô móvel ao encontrar um obstáculo a em seu trajeto indicando um obstáculo na direção MN(Medium Negative) e a uma distância M(Medium).

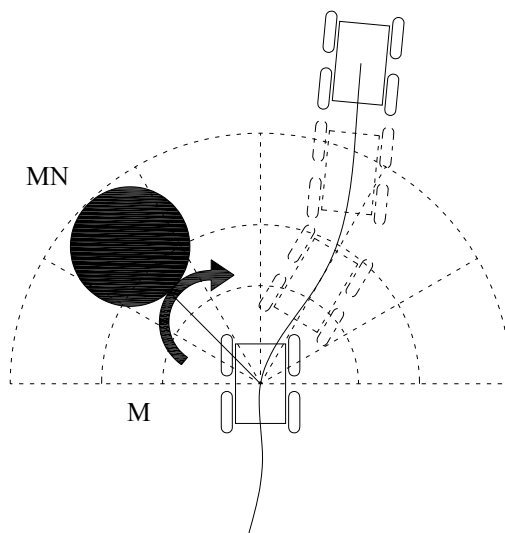


Figura 5 - Representação do Obstáculo e Reação do Robô

Regras Fuzzy

Por meio do conhecimento humano podem ser formuladas regras para incorporar inteligência ao sistema de robótica móvel que utiliza controle fuzzy. As regras foram formuladas conforme descrito abaixo para o sistema de controle do tipo desvio de obstáculos.

IF distancia N AND ângulo ZP THEN vel_saida IS M AND vel angular de saída IS MN
IF distancia M AND ângulo ZP THEN vel_saida IS N AND vel angular de saída IS MN
IF distancia N AND ângulo MP THEN vel_saida IS M AND vel angular de saída IS ZN
IF distancia M AND ângulo MP THEN vel_saida IS N AND vel angular de saída IS MN
IF distancia N AND ângulo ZN THEN vel_saida IS M AND vel angular de saída IS MP
IF distancia M AND ângulo ZN THEN vel_saida IS N AND vel angular de saída IS MP
IF distancia N AND ângulo MN THEN vel_saida IS M AND vel angular de saída IS ZP
IF distancia M AND ângulo MN THEN vel_saida IS N AND vel angular de saída IS MP
IF distancia N AND ângulo LP THEN vel_saida IS M AND vel angular de saída IS ZN
IF distancia N AND ângulo LN THEN vel_saida IS M AND vel angular de saída IS ZP

Por exemplo a primeira regra diz que se a distância do obstáculo fornecida pelo sensor for muito próximo (Near) e o sensor for o de ângulo positivo zero ou seja de pequeno ângulo em relação a frente do robô então a velocidade do veículo deve ser média com uma velocidade angular de saída média negativa ou seja na outra direção.

2.4 O Simulador de Robótica Móvel

Programa Computacional

O programa foi desenvolvido em C\C++ utilizando a biblioteca 3D OpenGL. O ambiente no qual o robô móvel deve interagir pode ser desenhado em um programa de modelagem 3D o qual é importado para dentro do simulador. Este recurso torna este simulador flexível o suficiente para um cenário específico. A partir de um modelo CAD

um laboratório de teste de protótipos é possível uma pré-configuração e mesmo a otimização antes do teste e prevenindo uma série de possíveis acidentes no laboratório de teste de protótipos.

O simulador considera um robô móvel cujo tempo de resposta do sistema de controle de velocidade angular e linear é próximo de zero.

O simulador permite testar as modificações implementadas no sistema de controle. Demonstra a forma como o robô virtual desvia de forma suave dos obstáculos. As linhas vermelhas são representações virtuais dos sensores do robô.

O sistema de controle fuzzy utilizado é multivariável tanto na entrada como na saída. As informações de entrada são: o ângulo do sensor mais crítico ou de menor distância até o obstáculo e a distância em questão. As saídas do controlador são a velocidade linear e angular do robô.

A função controle de movimento é a chave do funcionamento do programa computacional do controle e simulação de robótica móvel. Por meio desta lógica o conhecimento representado pelas regras fuzzy permitem o robô móvel tomar decisões a respeito do comportamento a ser adotado em função das condições e obstáculos apresentados.

Pseudo-Código da função controle de movimento (responsável pelo controle de movimentação do robô no deslocamento e desvio de obstáculos)

Função Controle de Movimento

Entradas:

Qual o sensor mais crítico

Ângulo e distância até o target

SE (A distância do Obstáculo mostrada pelo sensor < Distância crítica)

***CALCULA** Lógica Fuzzy*

***MOVIMENTA** o Robô de acordo com a velocidade linear e velocidade angular calculadas pelo sistema Fuzzy*

SENÃO

MOVIMENTA o Robô em direção ao target

SE(A distância até o target < mínimo)

PARAR o robô

FINALIZA a simulação

A função de colisão permite que uma geometria tridimensional possa ser considerada no cálculo da distância do obstáculo mais próximo do robô móvel. A geometria 3D para efeito de cálculo é representada por faces triangulares das quais representam planos com uma delimitação de área para o cálculo da colisão com um segmento de reta 3D. A função sensores carrega os segmentos de reta que representam os sensores de distância do robô.

O algoritmo utilizado para o cálculo da colisão se baseia na forma como a geometria é representada no espaço tridimensional. Por meio de cálculos geométricos, o programa visa simular os sensores de distância do robô, poderia ser por exemplo um sensor de ultra-som. A geometria é representada por meio de faces triangulares e o cálculo utilizado verifica a interseção de um segmento de reta 3D nesta face. No caso de haver uma colisão a distância é calculada. Os parâmetros de entrada do controlador são: o ângulo do sensor e a distância até o ponto de colisão do eixo que representa o sensor. O procedimento então pode ser descrito como para cada sensor é varrido em todas as faces para verificar uma possível colisão. O menor ou algum outro critério de valor encontrado na varredura é retornado assim como a identificação do respectivo sensor responsável por este sinal mais crítico. A partir desta informação podemos achar o ângulo do sensor e então alimentar o sistema de controle fuzzy.

Vantagens do Simulador

- Simulação e Visualização em 3D com OpenGL
- Cenário modelável em um programa CAD/3D
- Permite testar diversas configurações antes de testar o robô real

- Simula ruído de sensores
- Utiliza Lógica Fuzzy (Capacidade de lidar com dados ruidosos)
- Utiliza Lógica NeuroFuzzy (Capacidade de sintonizar o sistema fuzzy a partir de dados de treinamento)

Capítulo 3 – RESULTADOS

O simulador desenvolvido no presente trabalho possui vários arquivos de configuração que permitem modificar uma série de parâmetros tanto nas características do controle fuzzy/neurofuzzy como nos parâmetros do controlador. A visualização mostra em forma de uma animação gráfica 3d o comportamento do robô na simulação. Os principais parâmetros de cada simulação são mostradas para ilustrar as diferentes configurações. As simulações visam demonstrar as características de uma série de fatores que afetam o comportamento do robô no desvio de obstáculos. Fatores de modificações nas variáveis de entrada e de saída e valores de ruído dos sensores, dentre outros fatores. A primeira parte das simulações demonstra a reação do robô ao ruído dos sensores. A segunda parte o efeito de modificações nas variáveis antecedentes e conseqüentes e a parte final a funcionalidade do sistema neurofuzzy.

Configuração Básica do Controle Fuzzy para o Desvio de Obstáculos

As variáveis e regras a seguir mostram a configuração básica do sistema fuzzy para desvio de obstáculos. As demais simulações são em grande parte pequenas modificações nesta configuração inicial.

Variáveis Fuzzy Antecedentes

Variável Distância de Entrada

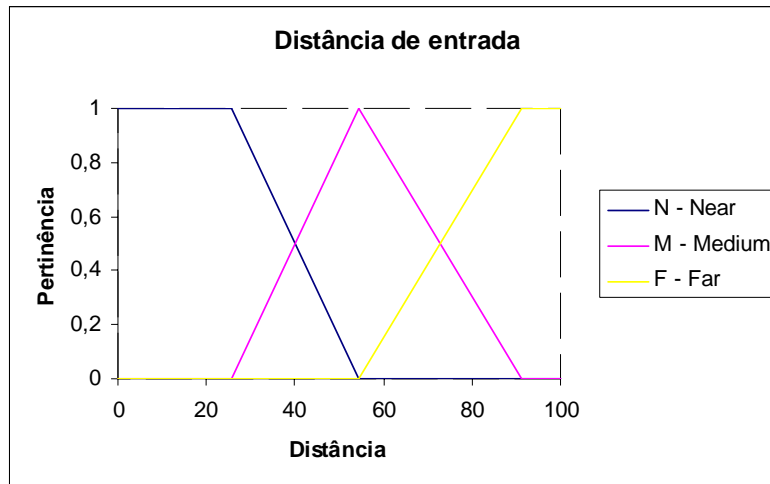


Figura 6 - Variável Antecedente Distância de Entrada

Variável Ângulo de Entrada

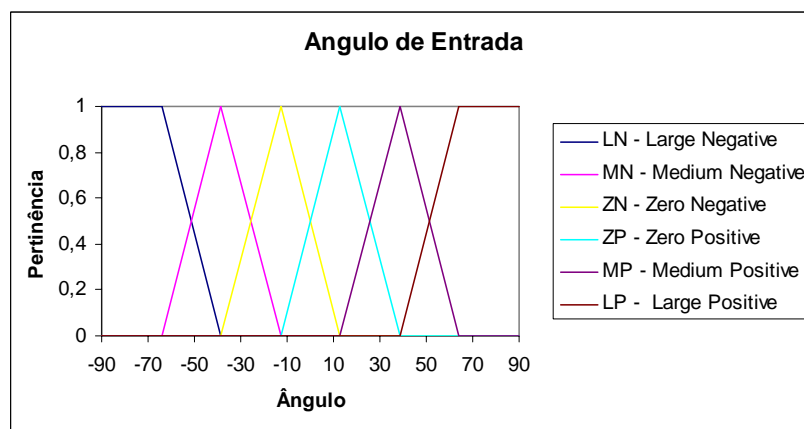


Figura 7 – Variável Antecedente Ângulo de Entrada

Variáveis Fuzzy Conseqüentes

Variável Velocidade de Saída

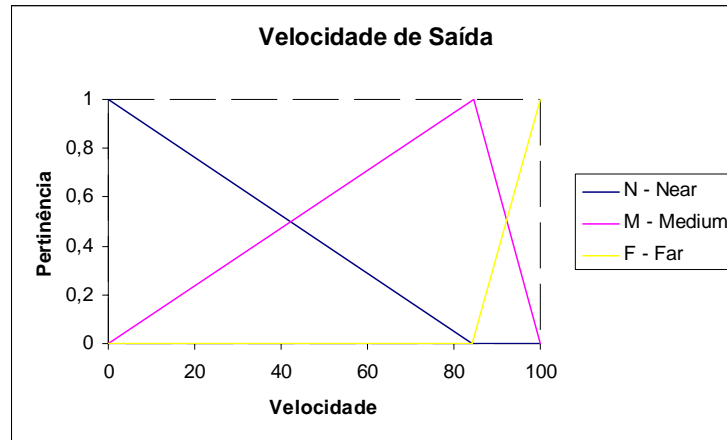


Figura 8 – Variável Consequente Velocidade de Saída

Variável Velocidade Angular de Saída

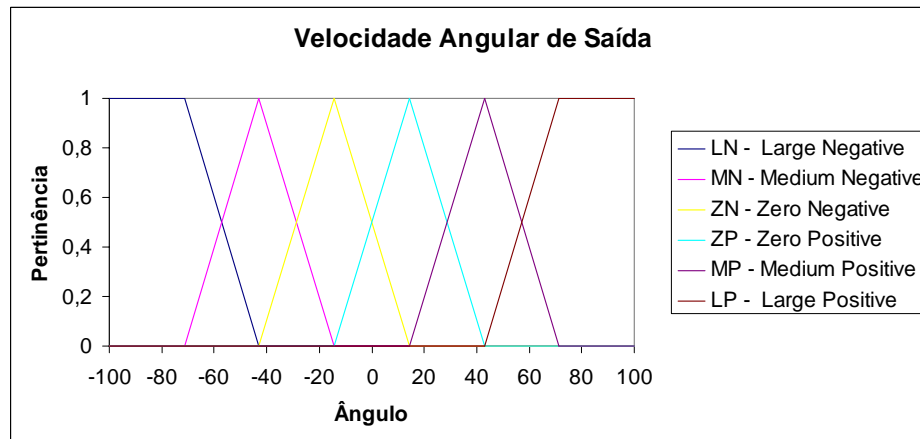


Figura 9 – Variável Consequente Velocidade Angular de Saída

Regras Fuzzy

São dez regras que regem o comportamento do robô para o desvio de obstáculos.

- Regra 1: N **AND** ZP **THEN** N **AND** MN
- Regra 2: M **AND** ZP **THEN** M **AND** MN
- Regra 3: N **AND** MP **THEN** N **AND** ZN
- Regra 4: M **AND** MP **THEN** M **AND** MN
- Regra 5: N **AND** ZN **THEN** N **AND** MP
- Regra 6: M **AND** ZN **THEN** M **AND** MP
- Regra 7: N **AND** MN **THEN** N **AND** ZP
- Regra 8: M **AND** MN **THEN** M **AND** MP

Regra 9: N AND LP THEN M AND ZN

Regra 10: N AND LN THEN M AND ZP

Os dados de configuração da simulação são diversos parâmetros que são descritos abaixo:

Arquivo de interface fuzzy: arquivo onde estão alocadas as principais informações para a construção da interface fuzzy. Dentre as informações destacam-se as regras fuzzy, arquivos de variáveis fuzzy com suas respectivas funções de pertinência tanto para os antecedentes quanto para os consequentes.

Arquivo dos sensores: arquivo no qual estão alocadas as informações das geometrias dos sensores (segmentos de reta 3D).

Seleção dos sensores: método pelo qual um sensor é considerado mais crítico do que outro. Este método determina qual o sensor crítico que vai fornecer o ângulo de entrada e a distância de entrada para o sistema fuzzy de desvio de obstáculos. Nas simulações utilizam-se o sensor que apresenta a menor distância até o obstáculo.

Escala dos sensores: fator de escala aplicado aos sensores que permite testar sensores com um alcance maior ou menor sem ter que alterar o arquivo da geometria dos sensores.

Percentual da distância para inferência fuzzy: percentual do alcance dos sensores a partir do qual o sistema considera que a distância até o obstáculo necessita ser tratada pelo sistema fuzzy para o desvio de obstáculos.

Velocidade linear: velocidade máxima linear do robô simulado em unidades de desenho por iteração. A iteração está relacionada ao tempo, pois seria o tempo de um ciclo do programa do robô, se este ciclo demorar um segundo, logo esta é sua unidade de tempo. As unidades de desenho como em todo programa CAD pode ser a que o usuário considerar como metros ou milímetros.

Velocidade angular máxima: velocidade angular máxima do robô simulado em graus por iteração.

Arquivo da geometria do robô móvel: arquivo com os dados da geometria 3D no formato “3ds” que representam graficamente o robô móvel.

Arquivo da geometria do cenário: arquivo com os dados da geometria 3D no formato “3ds” que representam graficamente o cenário.

Arquivo de trajetória: arquivo onde está armazenada a trajetória da posição inicial até a posição de destino ou fim da simulação.

Posição inicial: posição de partida do robô móvel na simulação. O eixo de origem considerado na simulação é a posição central da tela de simulação e os valores estão em unidades de desenho.

Target: posição que marca o destino onde o robô móvel deve chegar.

Percentual de ruído: percentual do alcance dos sensores que pode variar pseudo-randomicamente da medida de distância calculada. Este pode simular características de sensores de diferentes tipos como: sensores de maior ruído como sonar e sensores com ruído próximo de zero como o sensor laser.

Simulações

A simulação 16 está descrita com mais detalhe no Apêndice B para ilustrar passo a passo o funcionamento do programa durante as simulações de Robótica Móvel.

Simulação Ruído dos sensores de distância do robô móvel

Este primeiro conjunto de simulações de 1 até 4 visam verificar o efeito do ruído nos sensores na eficiência do controlador no desvio de obstáculos. Serão simulados ruídos em cada um dos sensores tendo como amplitude um percentual do valor medido. Os valores medidos serão somados ao ruído pseudo-randômico e segue para a etapa de seleção do sensor mais critico neste caso o de menor valor, ou seja, com o obstáculo mais próximo.

Dados da Simulação 01:

Arquivo de interface fuzzy = interface_rob001.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 1.8
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Arquivo de trajetória = trajeto01.txt
Posição inicial X = 5.0 e Y = -55.0
Target_X = 10.0 e Y = 110.0
Percentual de ruído nos sensores = 0.0

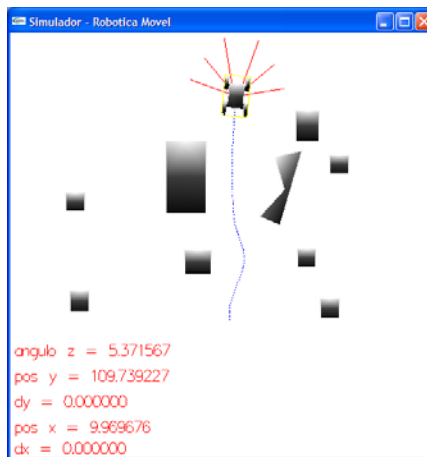


Figura 10 – Simulação 01

Dados da Simulação 02:

Arquivo de interface fuzzy = interface_rob001.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene01.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = 5.0 e Y = -55.0
 Target_X = 10.0 e Y = 110.0
 Percentual de ruído nos sensores = 5.0

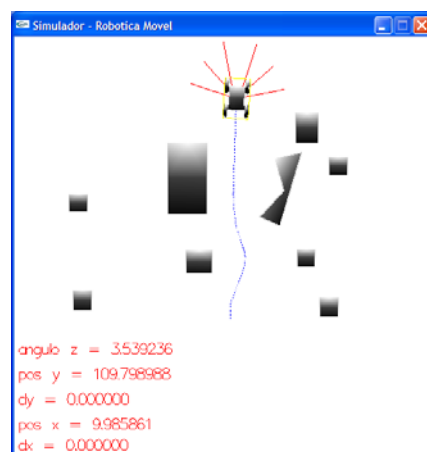


Figura 11 – Simulação 02

Dados da Simulação 03:

Arquivo de interface fuzzy = interface_rob001.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 1.8
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Arquivo de trajetória = trajeto01.txt
Posição inicial X = 5.0 e Y = -55.0
Target_X = 10.0 e Y = 110.0
Percentual de ruído nos sensores = 10.0

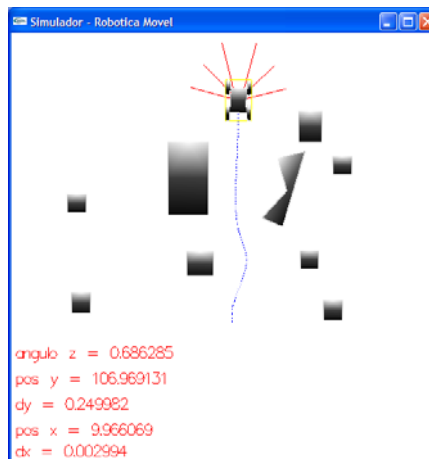


Figura 12 – Simulação 03

Dados da Simulação 04:

Arquivo de interface fuzzy = interface_rob001.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 1.8
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Arquivo de trajetória = trajeto01.txt
Posição inicial X = 5.0 e Y = -55.0
Target_X = 10.0 e Y = 110.0
Percentual de ruído nos sensores = 15.0

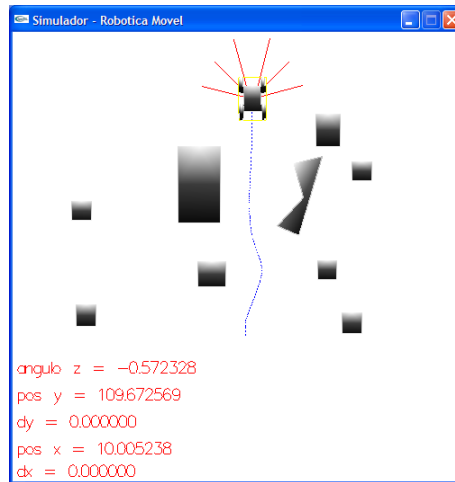


Figura 13 – Simulação 04

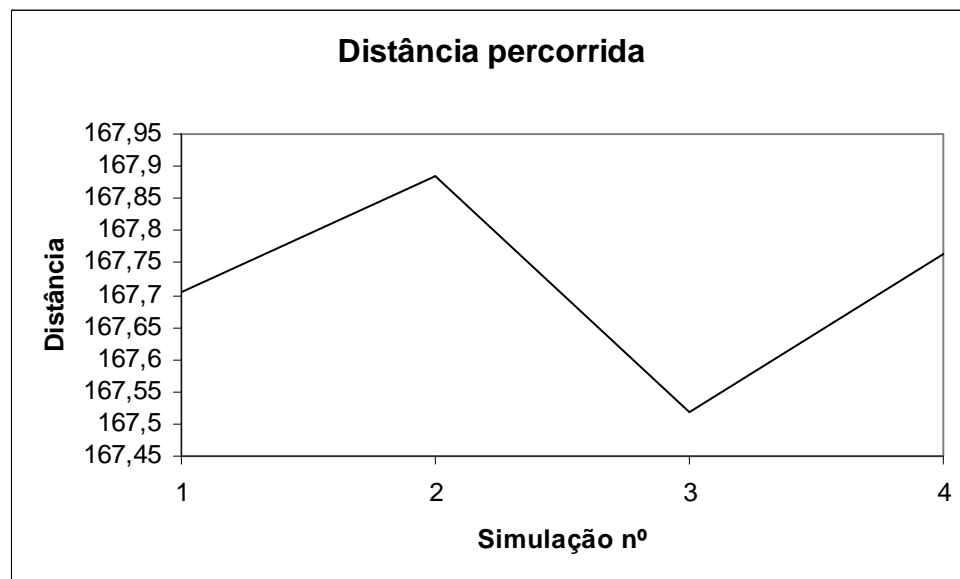


Figura 14 – Distância percorrida nas simulações de 1-4

As simulações de 1 a 4 não apresentam diferenças significativas na distância percorrida.

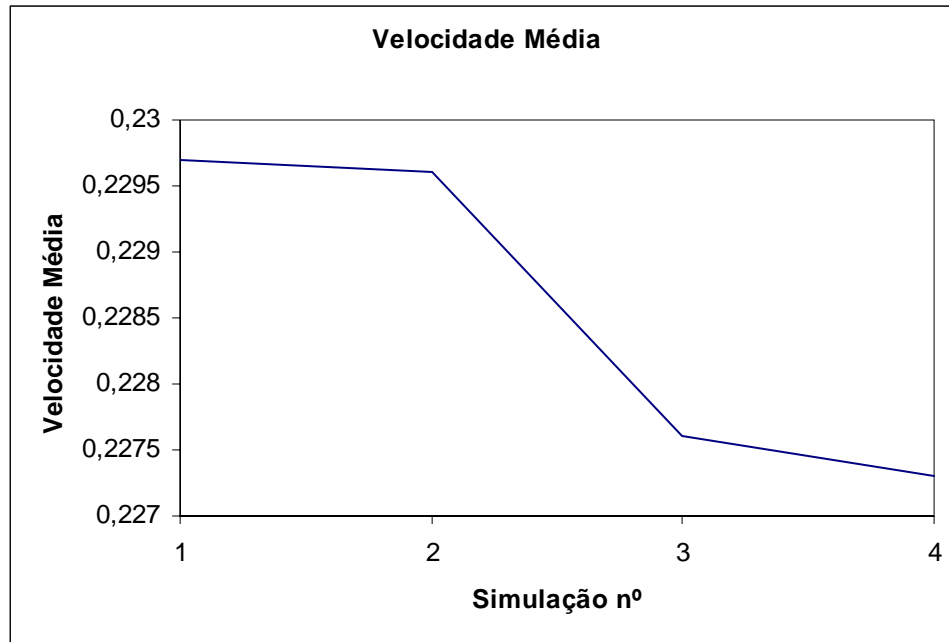


Figura 15 – Velocidade Média nas simulações de 1-4

As simulações de 1 a 4 não apresentam diferenças significativas nas velocidades médias.

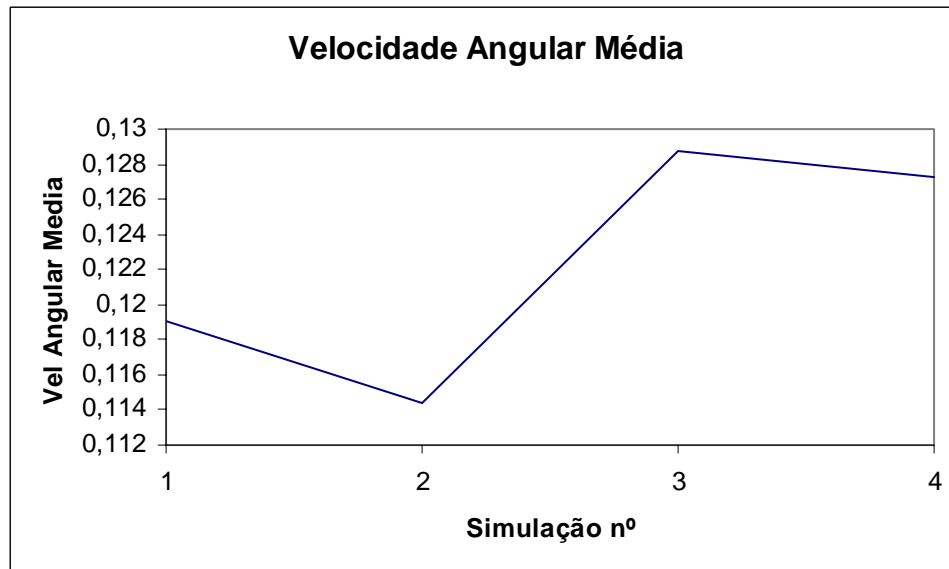


Figura 16 – Velocidade Angular Média nas simulações de 1-4

Nas velocidades angulares a simulação 4 apresenta um maior valor.

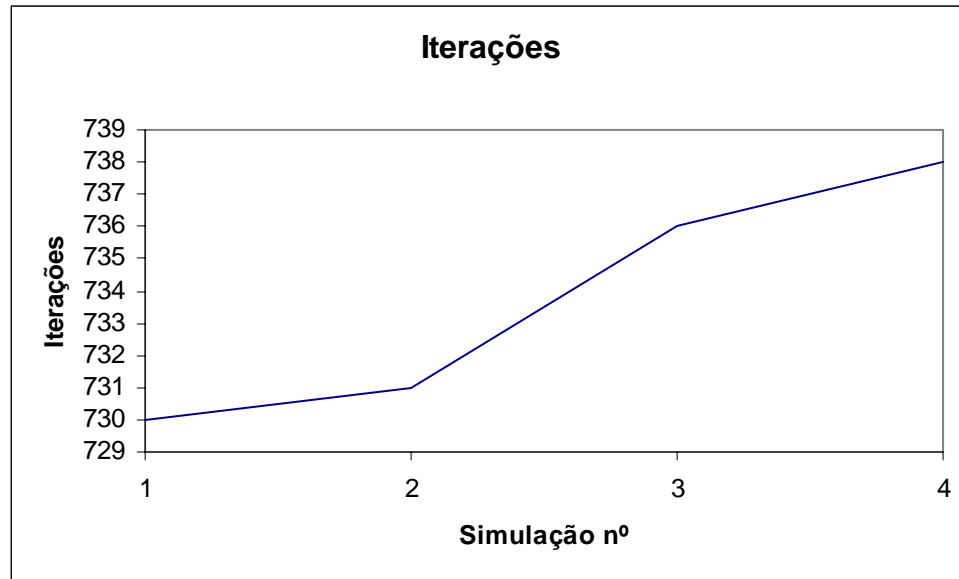


Figura 17 – Iterações nas simulações de 1-4

Quanto ao número de iterações a simulação 4 apresenta o maior valor.

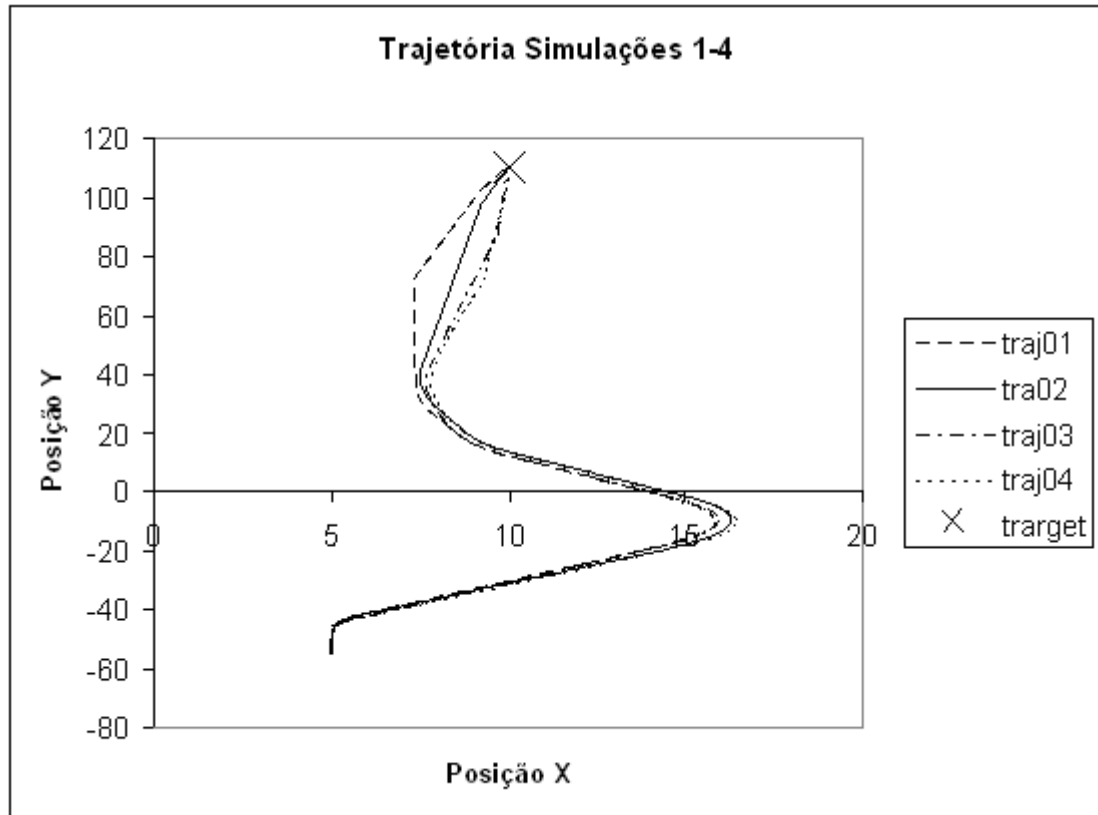


Figura 18 – Trajetórias das simulações de 1-4

Na trajetória da simulação 01 nota-se um pequeno desvio antes de o robô móvel se direcionar ao target, pois como o ruído está em 0% o robot aciona a regra 9 que produz um pequeno desvio praticamente seguindo em linha reta até a saída do sensor do obstáculo e então convergindo para o target. As simulações 2, 3 e 4 seguem praticamente a mesma trajetória convergindo para o target.

As simulações de 5 até 9 testam a sensibilidade ao ruído em um cenário de maior dificuldade no desvio dos obstáculos.

Dados da Simulação 05:

Arquivo de interface fuzzy = interface_rob001.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8

Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene02.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = 5.0 e Y = -55.0
 Target_X = 10.0 e Y = 115.0
 Percentual de ruído nos sensores = 5.0

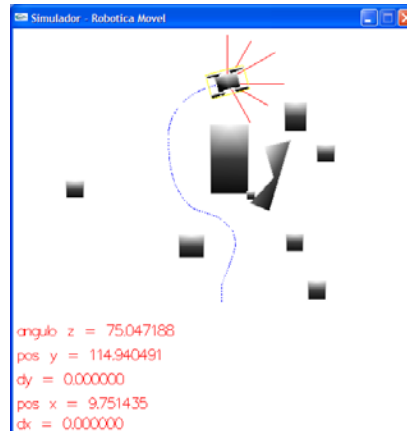


Figura 19 - Simulação 05

Dados da Simulação 06:

Arquivo de interface fuzzy = interface_rob001.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene02.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = 5.0 e Y = -55.0
 Target_X = 10.0 e Y = 115.0
 Percentual de ruído nos sensores = 7.0

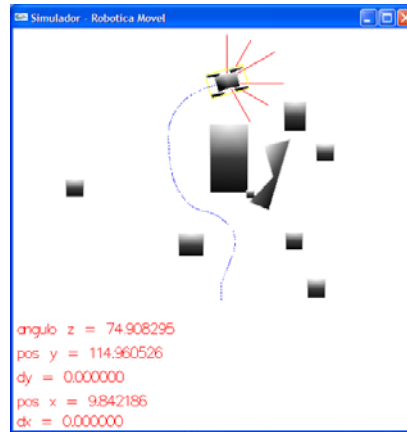


Figura 20 - Simulação 06

Dados da Simulação 07:

Arquivo de interface fuzzy = interface_rob001.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene02.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = 5.0 e Y = -55.0
 Target_X = 10.0 e Y = 115.0
 Percentual de ruído nos sensores = 10.0

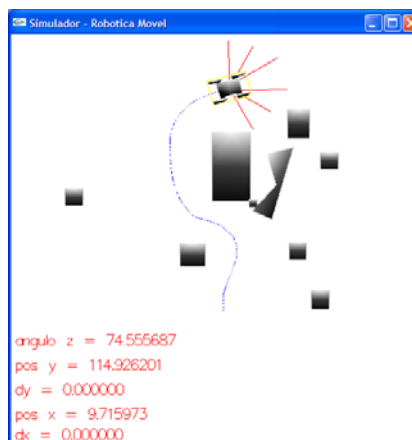


Figura 21 - Simulação 07

Dados da Simulação 08:

Arquivo de interface fuzzy = interface_rob001.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 1.8
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene02.3ds
Arquivo de trajetória = trajeto01.txt
Posição inicial X = 5.0 e Y = -55.0
Target_X = 10.0 e Y = 115.0
Percentual de ruído nos sensores = 15.0

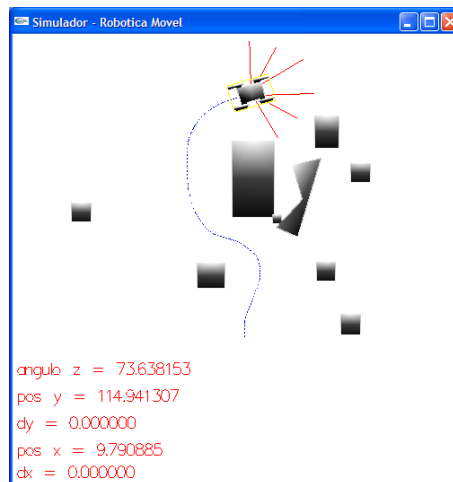


Figura 22 - Simulação 08

Dados da Simulação 09:

Arquivo de interface fuzzy = interface_rob001.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 1.8
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene02.3ds
Arquivo de trajetória = trajeto01.txt
Posição inicial X = 5.0 e Y = -55.0
Target_X = 10.0 e Y = 115.0
Percentual de ruído nos sensores = 45.0

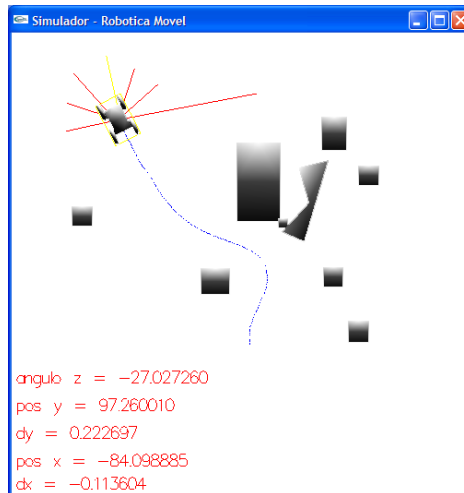


Figura 23 - Simulação 09

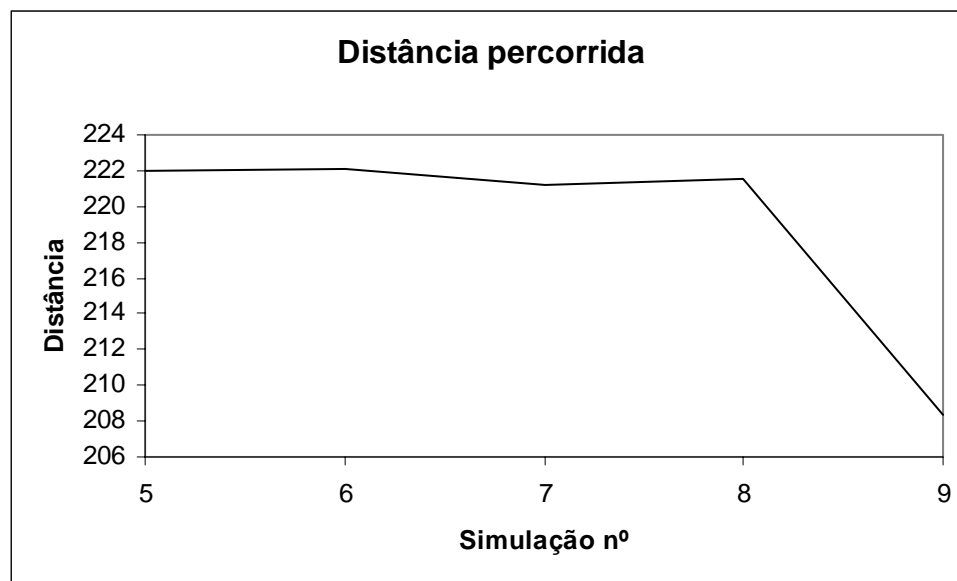


Figura 24 – Distância percorrida nas simulações de 5-9

Menor distancia na simulação 9 devido a simulação não convergir para o target então a simulação foi terminada assim que desviou de todos os obstáculos e as simulações 5 até a 8 mostra uma pequena variação distancia com o aumento do ruído.

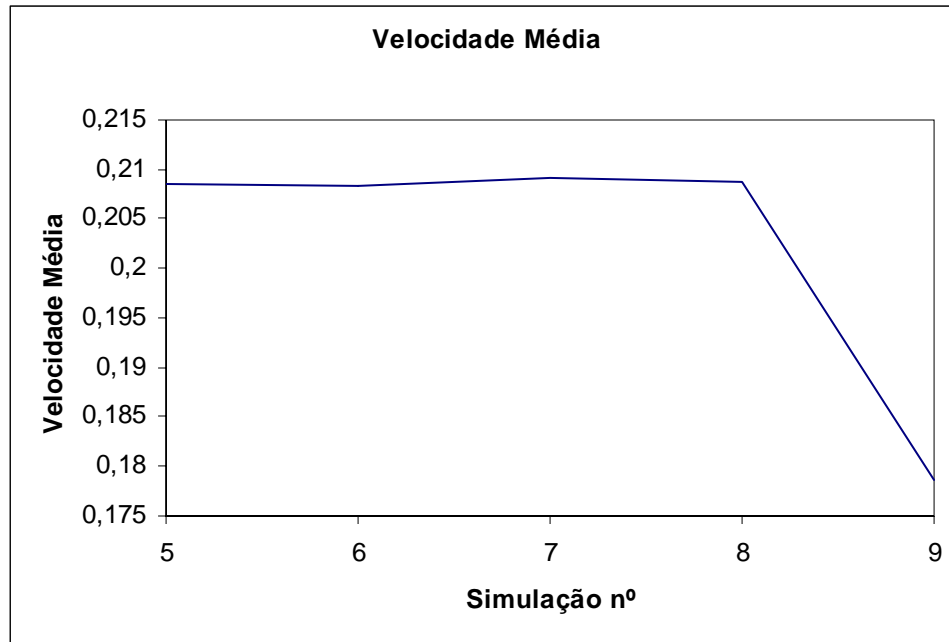


Figura 25 – Velocidade Média nas simulações de 5-9

Na simulação 9 devido ao alto ruído faz com que em boa parte do tempo as velocidade seja mais baixa como se houvesse um obstáculo mais próximo ainda que o real.

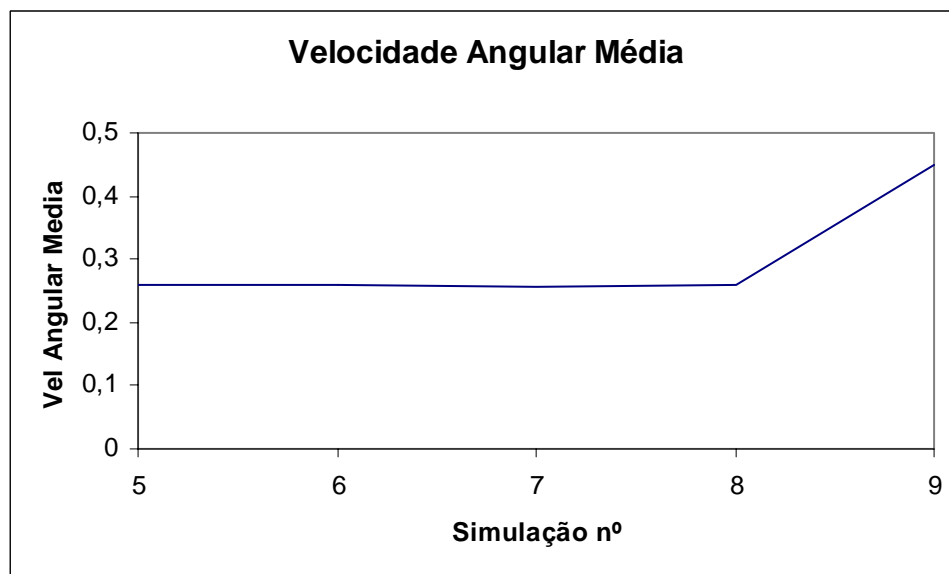


Figura 26 – Velocidade Angular Média nas simulações de 5-9

A maior velocidade angular da simulação 9 se deve ao alto valor de ruído.

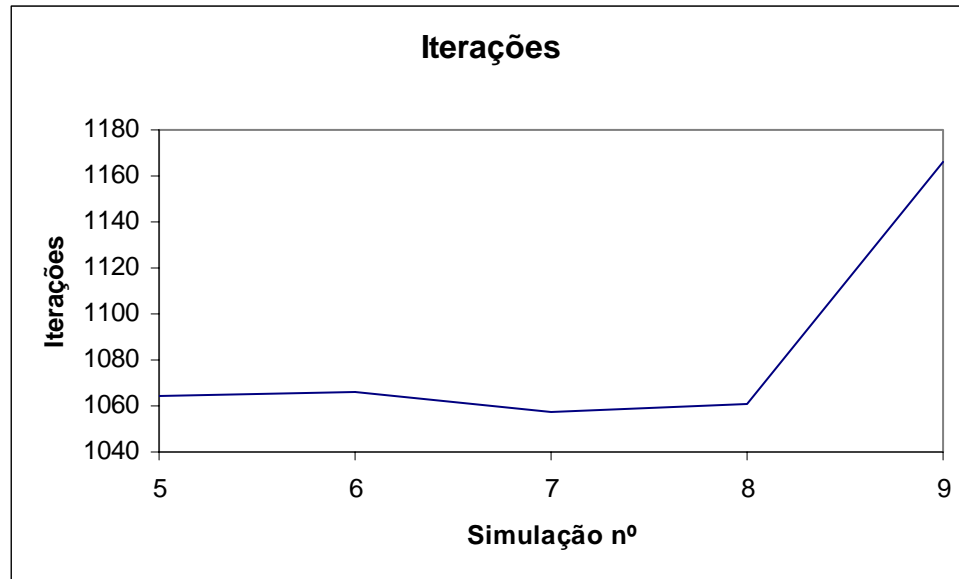


Figura 27 - Iterações nas simulações de 5-9

O maior valor nas iterações da simulação 9 é devido ao robô não convergir para o target as demais não apresentam grande variação.

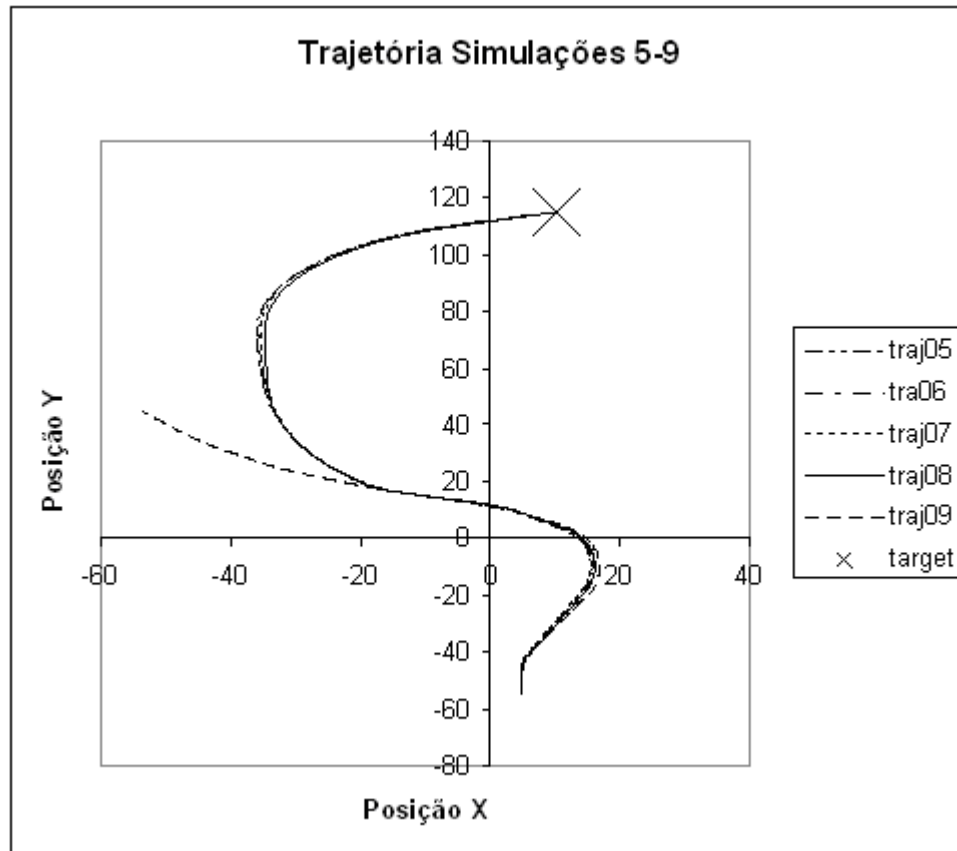


Figura 28 – Trajetórias das simulações de 5-9

As simulações 5, 6, 7 e 8 seguem praticamente a mesma trajetória convergindo para o target. Na simulação 9 o robô não colide no trecho inicial porém não converge para o target pois pelo ruído alto considera que sempre tem um obstáculo por perto.

As próximas simulações baseiam-se em regulagens do sistema fuzzy. O efeito de modificações nas funções de pertinência das variáveis fuzzy tanto dos antecedentes quanto dos conseqüentes vão ser analisadas em separado, assim como uma pequena modificação nas regras fuzzy de desvio de obstáculos. Depois de determinado seus efeitos individuais pode-se fazer uma regulação mais otimizada da interface fuzzy para uma melhor performance em um determinado cenário.

As simulações de 10-14 visam mostrar o efeito de diversas configurações na variável antecedente distância de entrada e a diferença entre as interfaces fuzzy está na configuração da variável fuzzy distância de entrada. Seguem abaixo as configurações e comentários.

Dados da Simulação 10:

Arquivo de interface fuzzy = interface_rob001.dat

Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

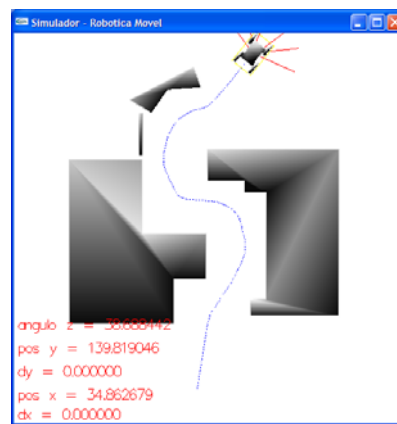


Figura 29 - Simulação 10

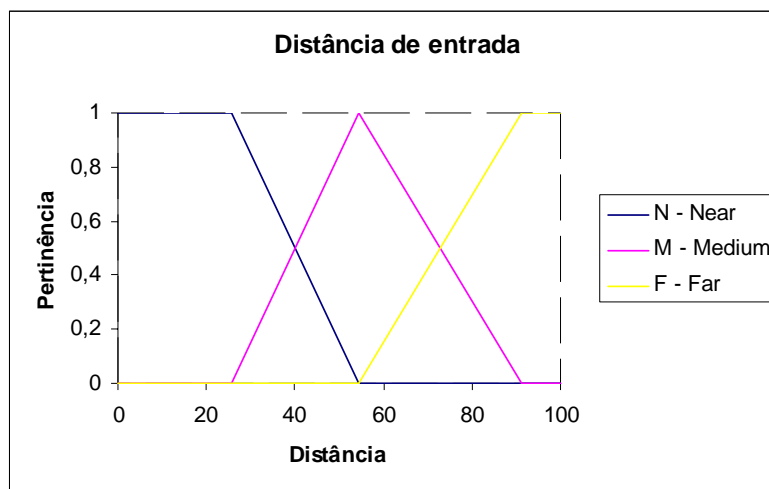


Figura 30 - Variável Antecedente Distância de Entrada

Dados da Simulação 11:

Arquivo de interface fuzzy = interface_rob002.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

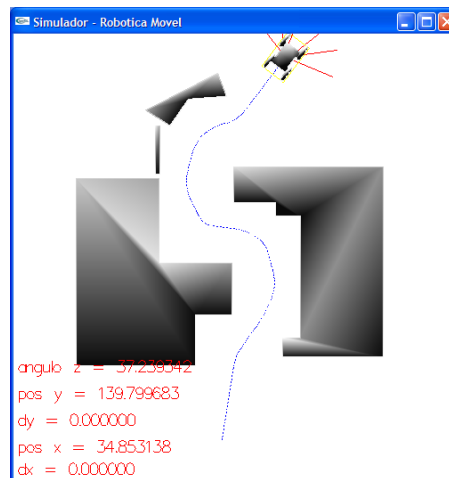


Figura 31 - Simulação 11

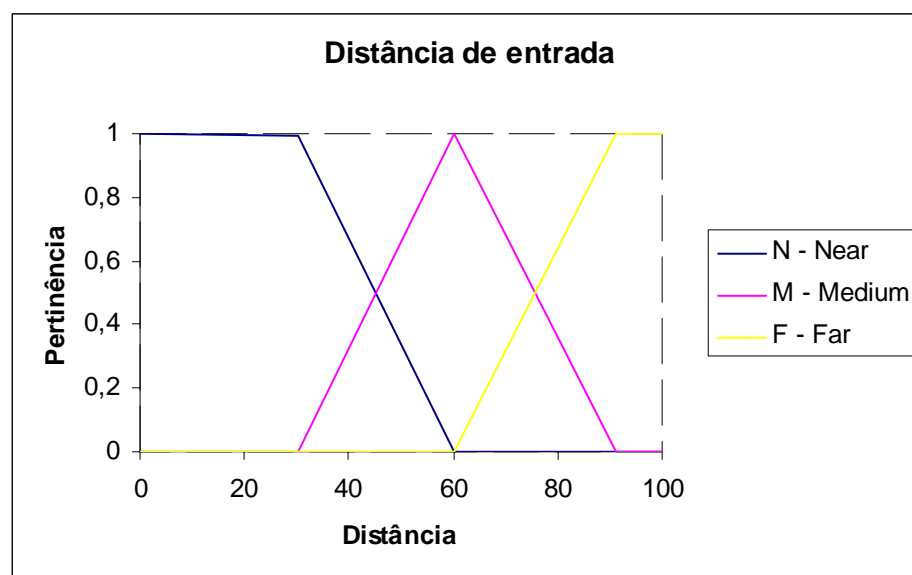


Figura 32 – Variável Antecedente Distância de Entrada

Com esta variável antecedente tanto o maior valor da função de pertinência M quanto da N aumentaram no eixo das distâncias.

Dados da Simulação 12:

Arquivo de interface fuzzy = interface_rob003.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 1.8
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene03.3ds
Arquivo de trajetória = trajeto01.txt
Posição inicial X = -10.0 e Y = -130.0
Target_X = 35.0 e Y = 140.0
Percentual de ruído nos sensores = 5.0

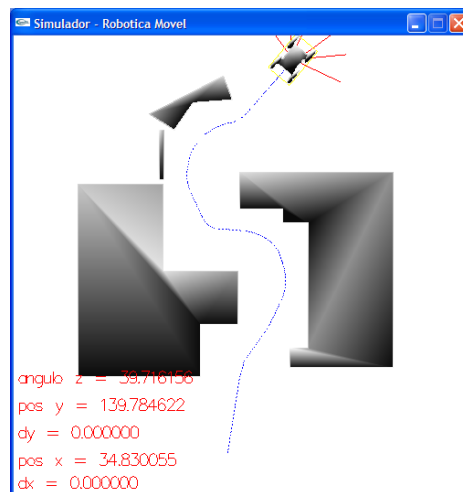


Figura 33 - Simulação 12

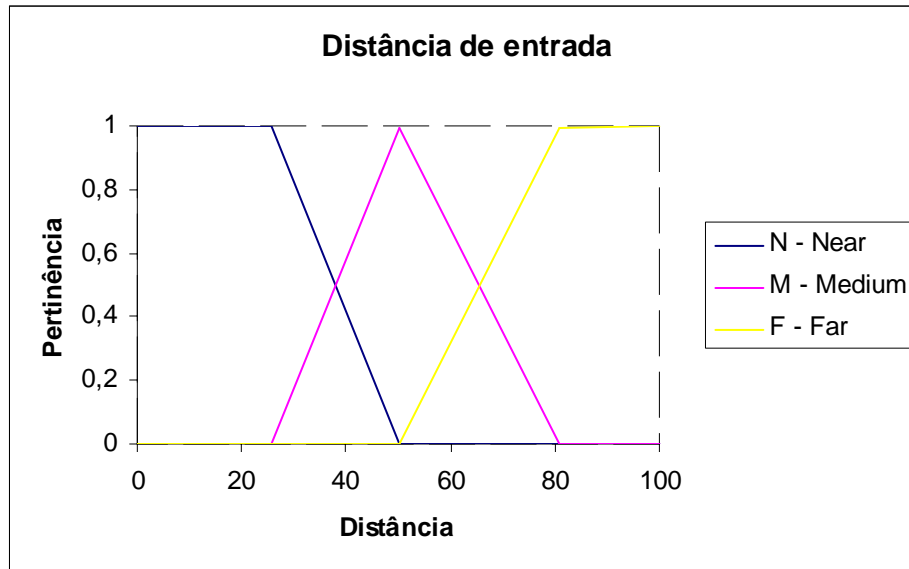


Figura 34 – Variável Antecedente Distância de Entrada

Com esta variável antecedente o maior valor da função de pertinência M diminuiu no eixo das distâncias.

Dados da Simulação 13:

Arquivo de interface fuzzy = interface_rob004.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

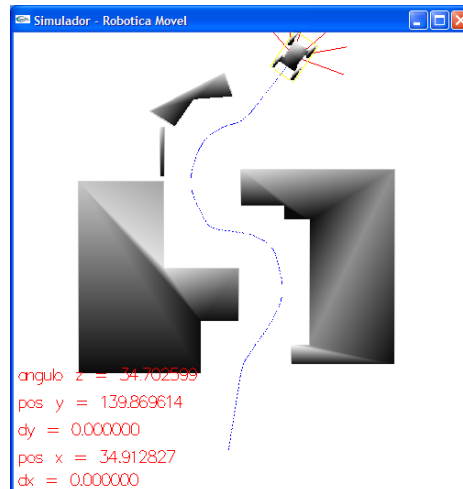


Figura 35 - Simulação 13

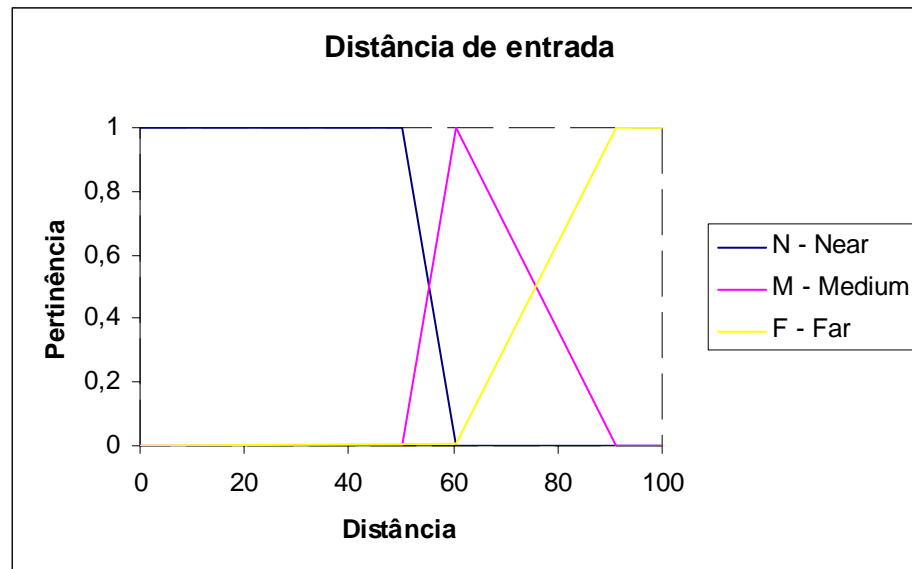


Figura 36 – Variável Antecedente Distância de Entrada

Com esta variável antecedente tanto o maior valor da função de pertinência M quanto da N aumentaram no eixo das distâncias. Porém a N indica um maior aumento.

Dados da Simulação 14:

Arquivo de interface fuzzy = interface_rob005.dat

Arquivo dos sensores = sens3.txt

Seleção de sensores = menor valor

Escala dos sensores = 0.7

Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

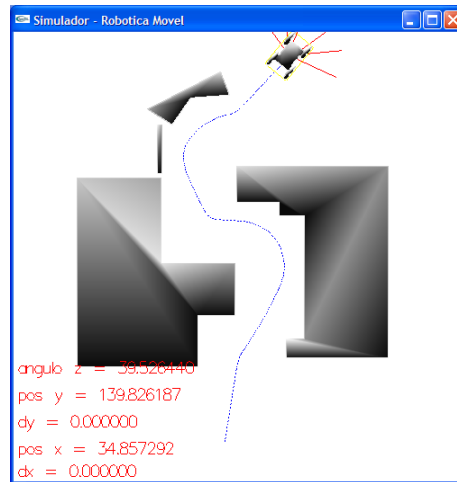


Figura 37 - Simulação 14

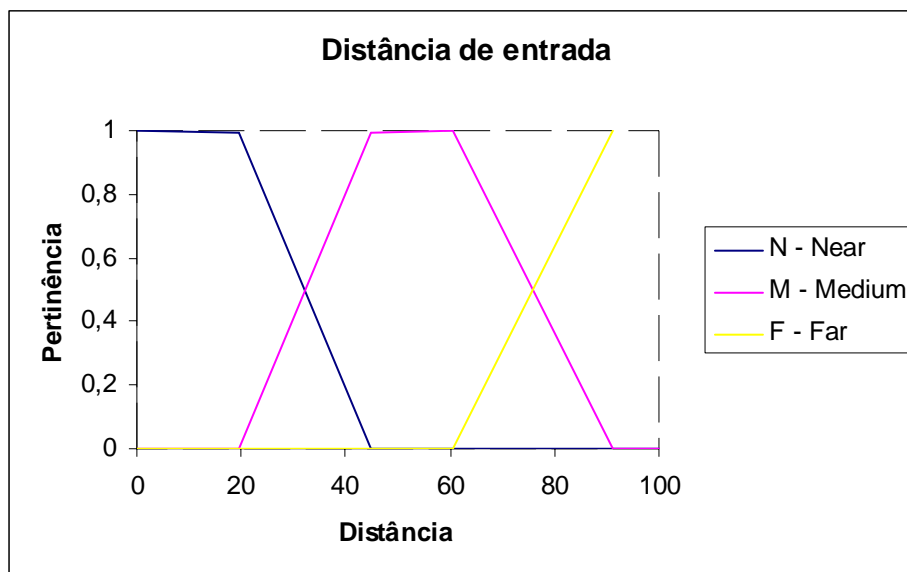


Figura 38 – Variável Antecedente Distância de Entrada

Com esta variável antecedente houve uma modificação no formato da função de pertinência M passando para o formato trapezoidal, logo a função tem seu valor de pertinência máximo em um faixa de valores.

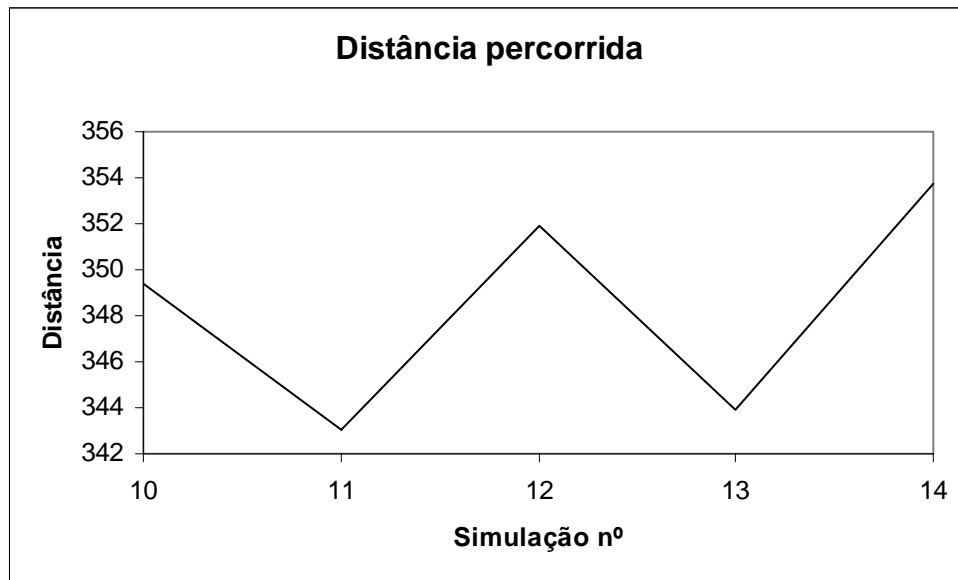


Figura 39 – Distância percorrida nas simulações de 10-14

Nas simulações 11 e 13 apresentam os menores valores de distância percorrida.

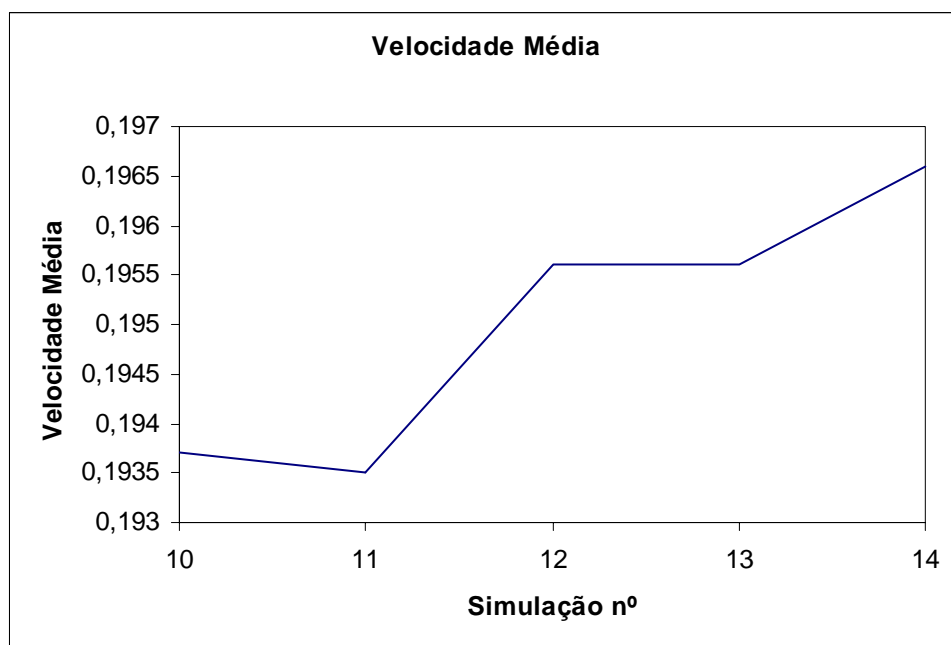


Figura 40 – Velocidade Média nas simulações de 10-14

A simulação 11 apresenta o menor valor de velocidade média e a simulação 14 apresenta um dos maiores valores de velocidade média.

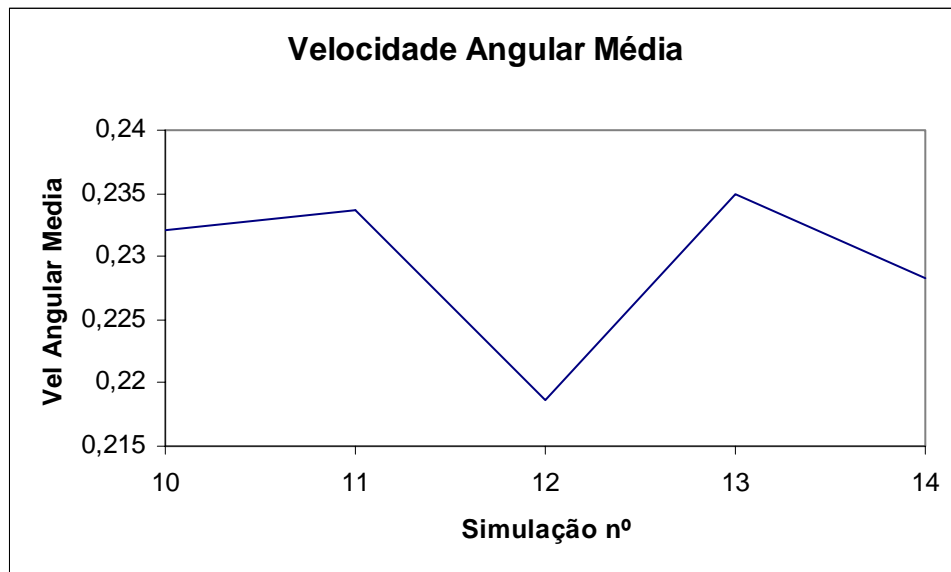


Figura 41 – Velocidade Angular Média nas simulações de 10-14

As simulações 11 e 13 apresentam os maiores valores de velocidade angular média.

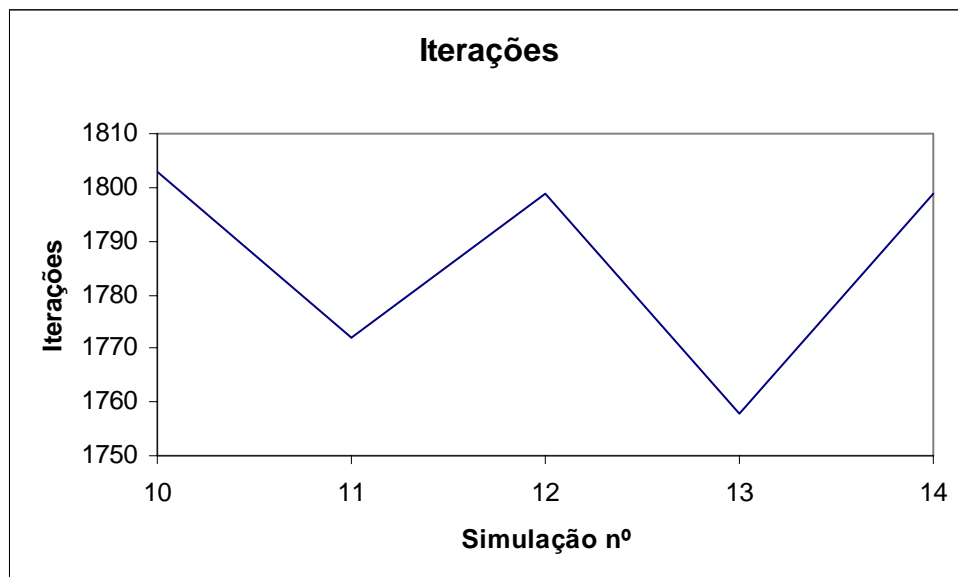


Figura 42 – Iterações nas simulações de 10-14

A simulação 13 apresenta o menor número de iterações.

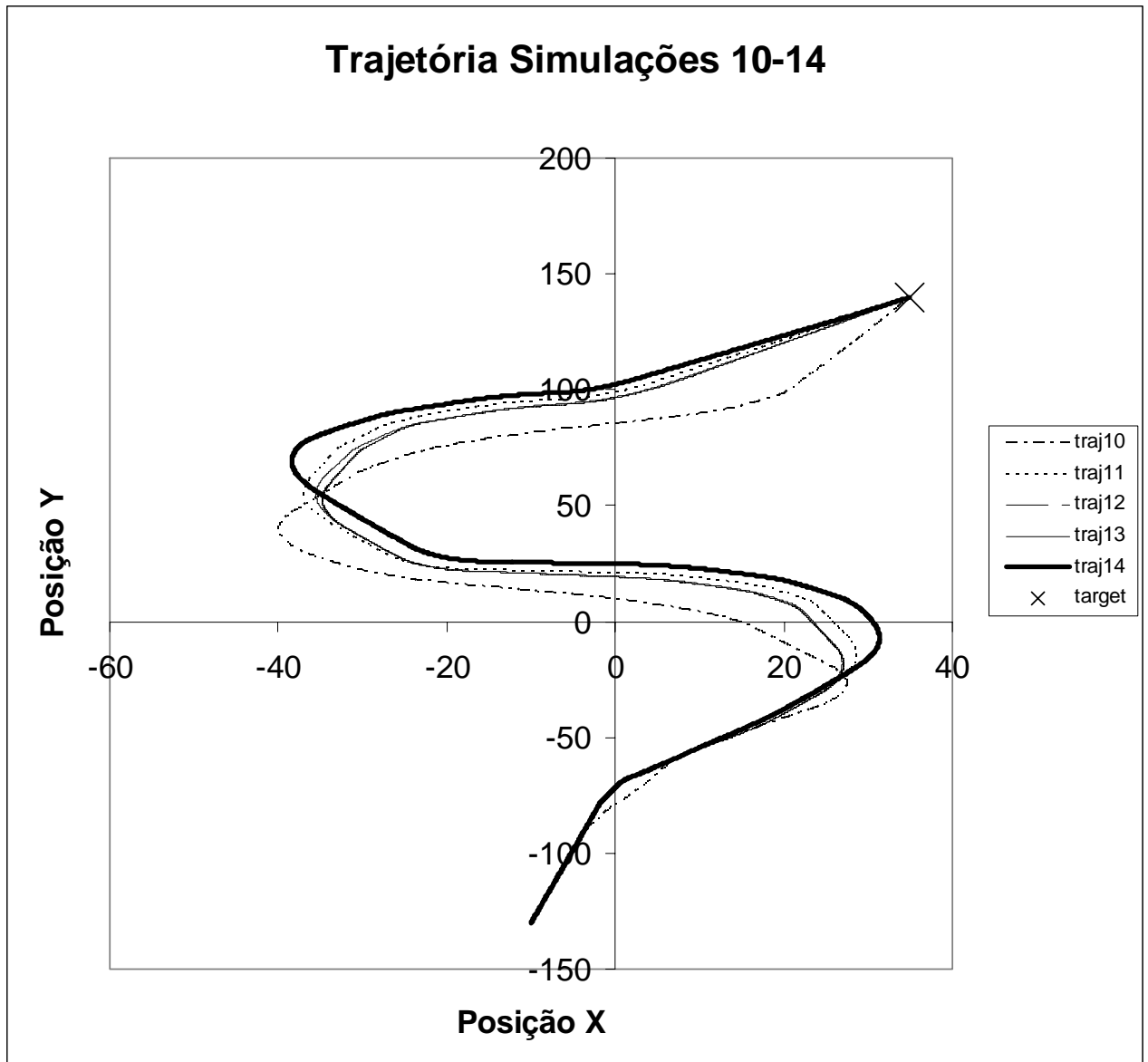


Figura 43 – Trajetórias das simulações de 10-14

A simulação 14 apresenta uma maior velocidade média. As trajetórias 11, 12 e 13 apresentam trajetos semelhantes e a simulação 10 um trajeto um pouco diferente e as curvas acontecem um pouco antes das demais simulações. A simulação 13 se destaca por apresentar tanto uma distancia percorrida quanto um número de iterações menor.

As simulações de 15 e 16 visam mostrar o efeito de diversas configurações na variável antecedente ângulo de entrada. A referência é a simulação 13 modificando apenas a variável ângulo de entrada. Seguem abaixo as configurações e comentários.

A simulação 13 tem a seguinte configuração na variável ângulo de entrada

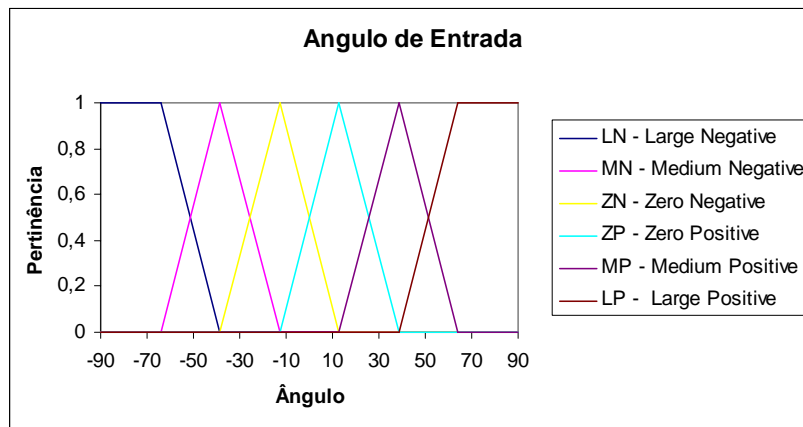


Figura 44 – Variável Antecedente Ângulo de Entrada

Dados da Simulação 15:

Arquivo de interface fuzzy = interface_rob006.dat

Arquivo dos sensores = sens3.txt

Seleção de sensores = menor valor

Escala dos sensores = 0.7

Percentual da distância para inferência fuzzy(%) = 95%

Velocidade linear = 0.25

Velocidade angular = 1.8

Arquivo geometria robô móvel = robot2.3ds

Arquivo geometria cenário = scene03.3ds

Arquivo de trajetória = trajeto01.txt

Posição inicial X = -10.0 e Y = -130.0

Target_X = 35.0 e Y = 140.0

Percentual de ruído nos sensores = 5.0

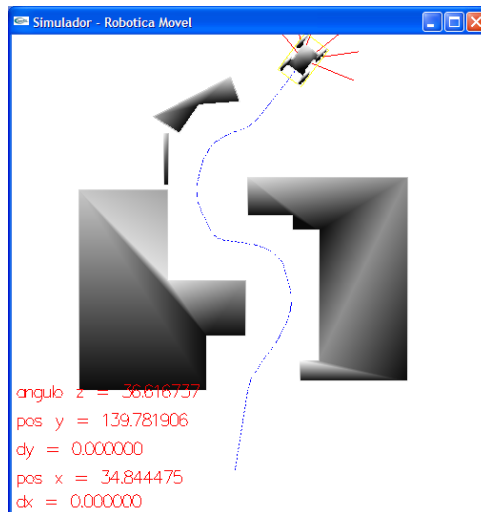


Figura 45 - Simulação 15

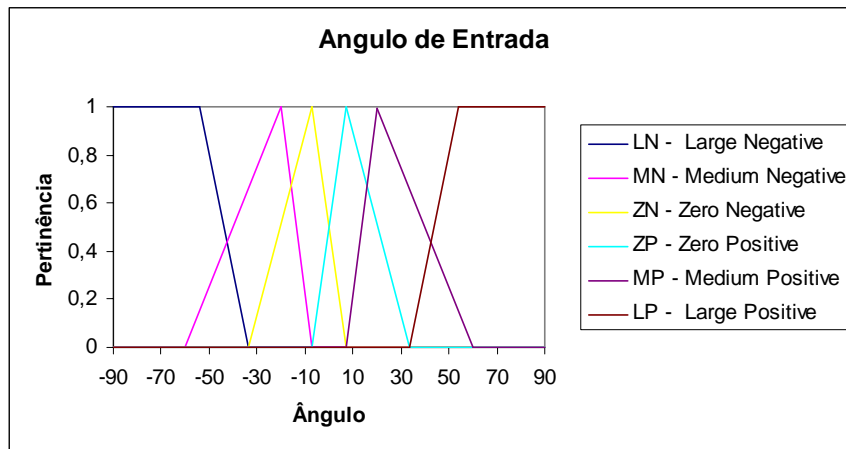


Figura 46 – Variável Antecedente Ângulo de Entrada

A variável antecedente ângulo de entrada neste caso foi modificada de modo que as funções de pertinência sejam mais próximas do ângulo 0°. Deste modo uma pequena mudança de ângulo pode acionar outra regra, por exemplo da função ZP para ZN ou para MP com apenas aproximadamente 20°.

Dados da Simulação 16:

Arquivo de interface fuzzy = interface_rob007.dat

Arquivo dos sensores = sens3.txt

Seleção de sensores = menor valor

Escala dos sensores = 0.7

Percentual da distância para inferência fuzzy(%) = 95%

Velocidade linear = 0.25

Velocidade angular = 1.8

Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

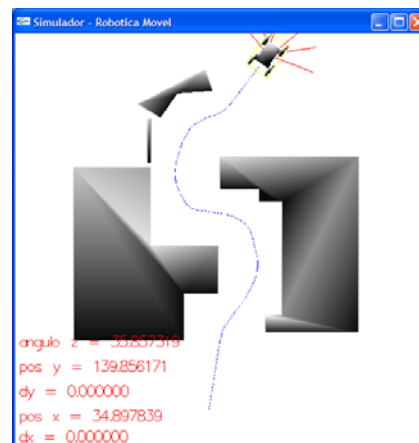


Figura 47 - Simulação 16

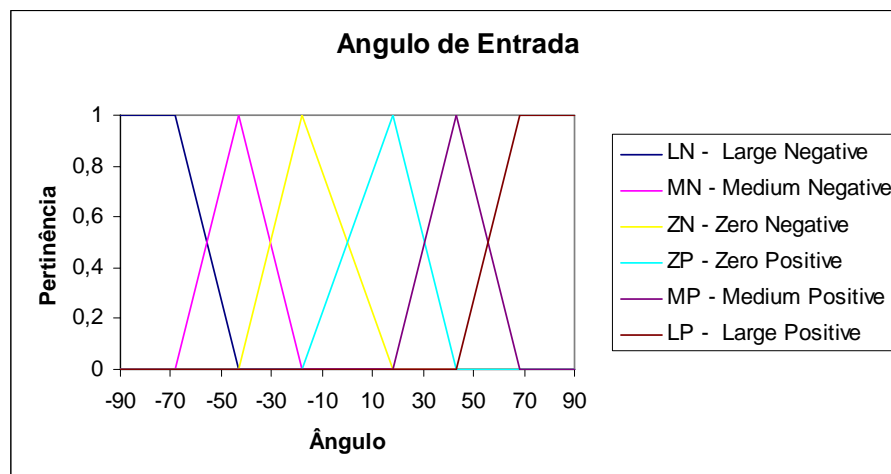


Figura 48 – Variável Antecedente Ângulo de Entrada

A variável antecedente ângulo de entrada neste caso foi modificada de modo que as funções de pertinência sejam mais afastadas do ângulo 0°. Deste modo maiores ângulos são necessários para atingir o maior valor de pertinência das funções como ZP e MP.

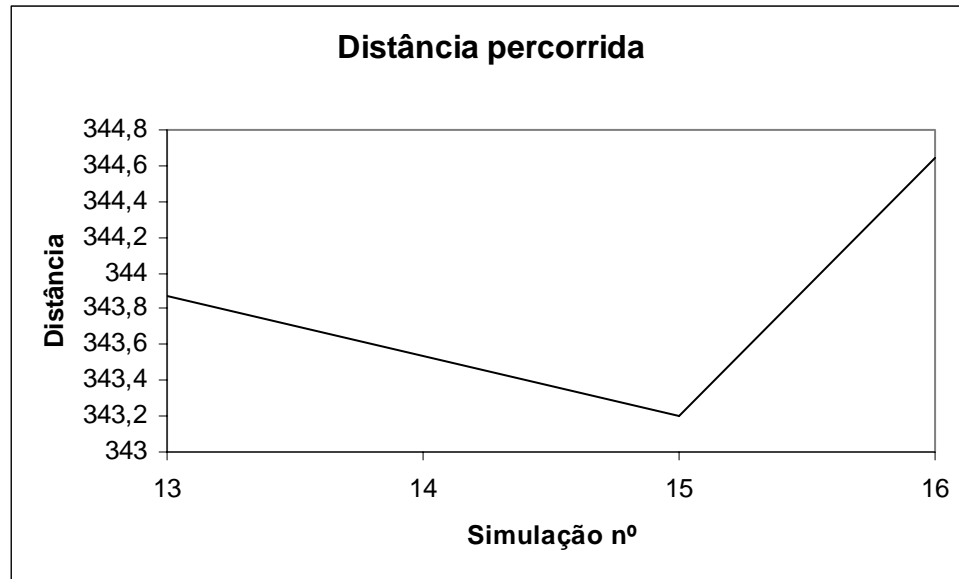


Figura 49 – Distância percorrida nas simulações 13,15 e 16

A simulação 16 apresenta um valor um pouco maior de distância.

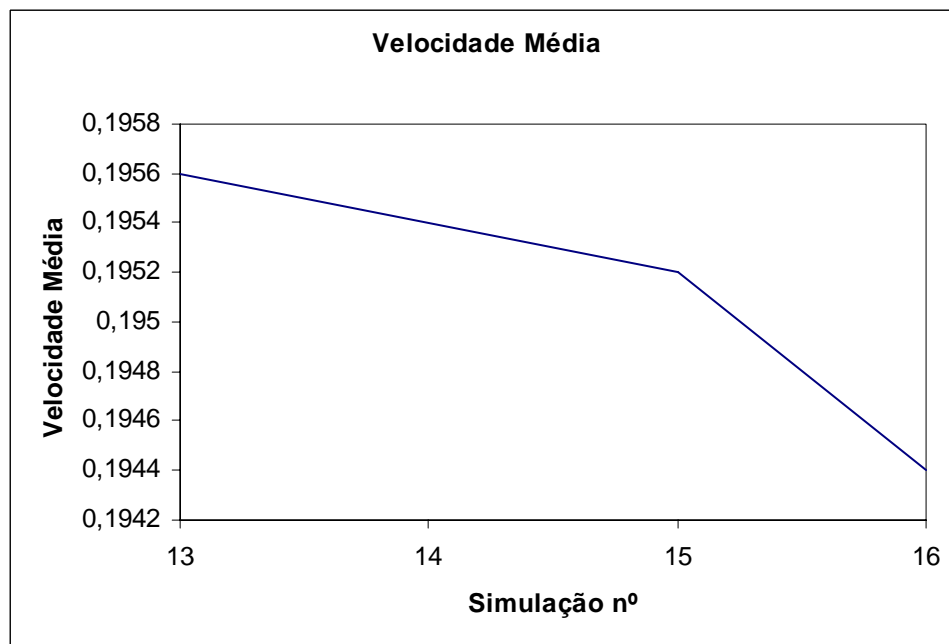


Figura 50 – Velocidade Média nas simulações 13,15 e 16

Não apresenta mudança significativa nos valores de velocidade média.

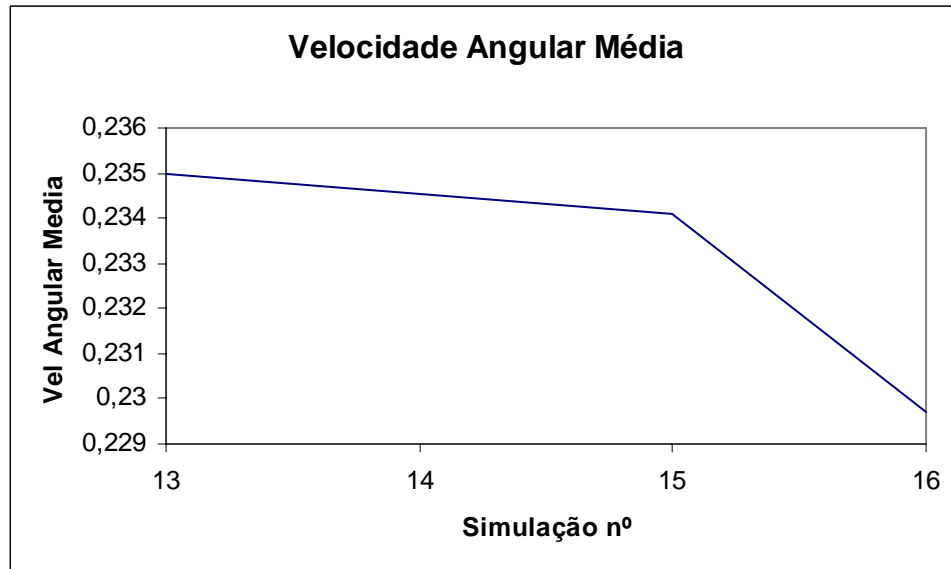


Figura 51 – Velocidade Ângular Média nas simulações 13,15 e 16

Não apresenta mudança significativa nos valores de velocidade angular média.

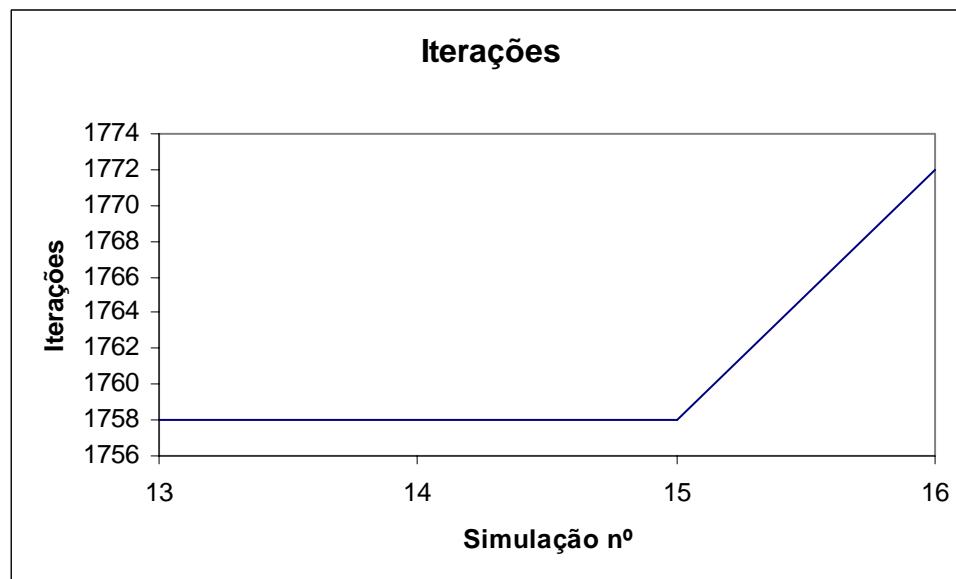


Figura 52 – Iterações nas simulações 13,15 e 16

A simulação 16 apresenta um valor um pouco maior nas iterações.

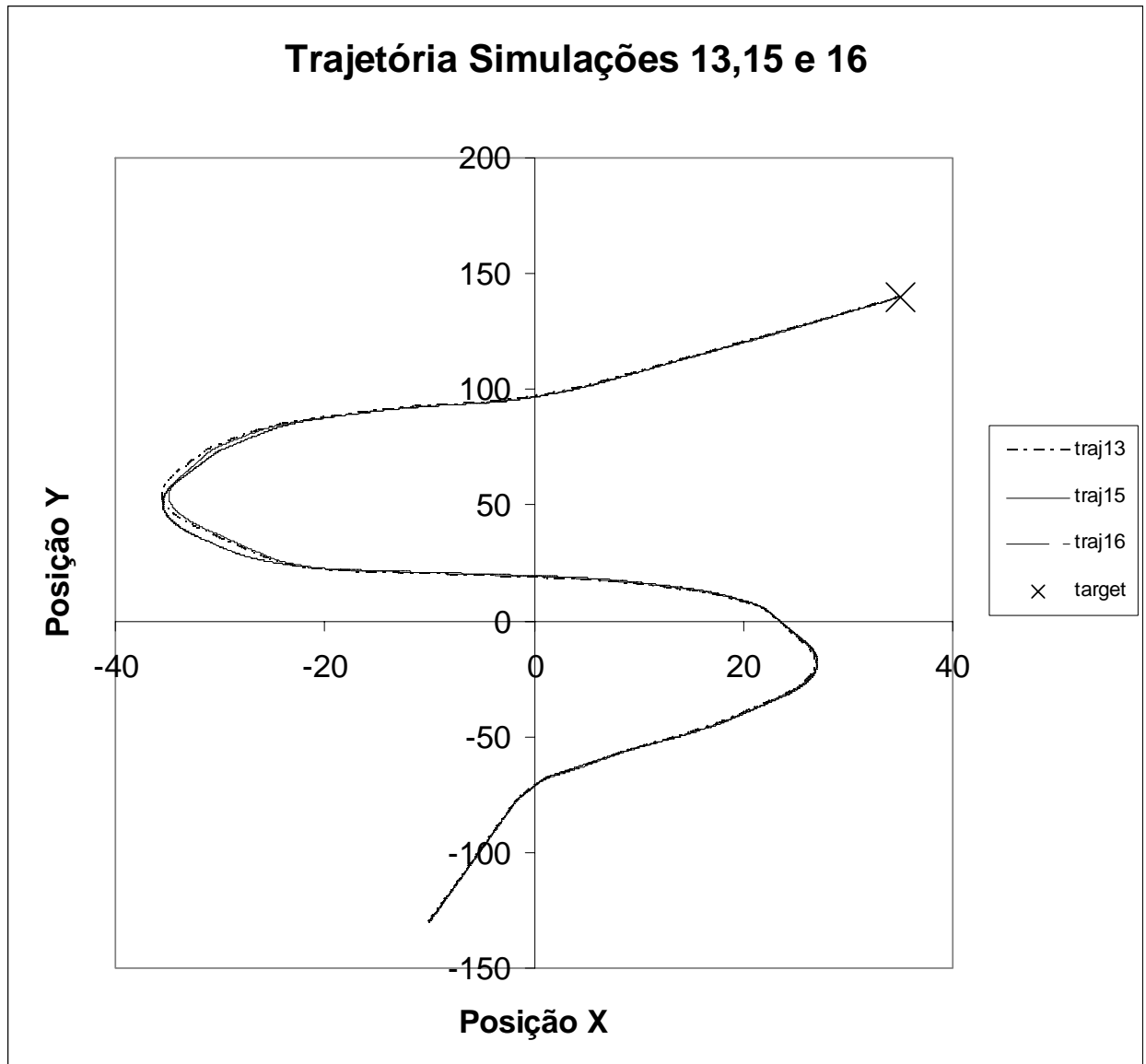


Figura 53 – Trajetórias das simulações 13,15 e 16

Os gráficos acima mostram que não houve modificação significativa no comportamento nas simulações. Apenas que a simulação 16 apresenta uma trajetória um pouco mais externa.

As simulações 17 e 18 visam verificar o efeito de modificações na variável velocidade de saída, aproximando e afastando da velocidade máxima de navegação do robô móvel.

A referência é a simulação 10 cuja variável velocidade de saída é mostrada abaixo.

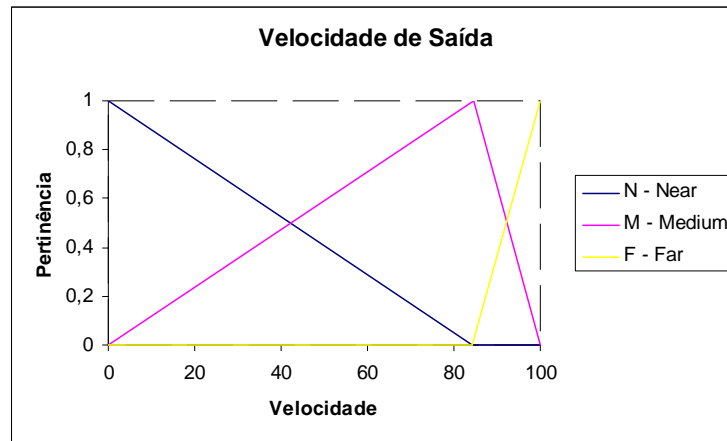


Figura 54– Variável Consequente Velocidade de Saída

Dados da Simulação 17:

Arquivo de interface fuzzy = interface_rob008.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

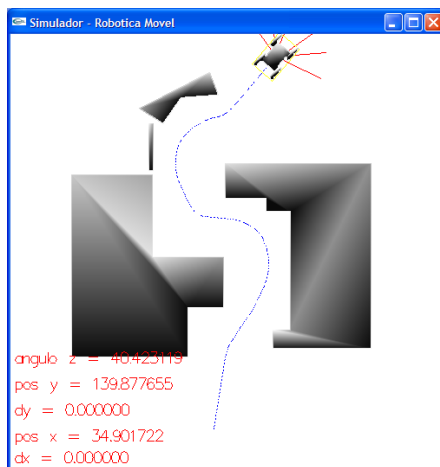


Figura 55 - Simulação 17

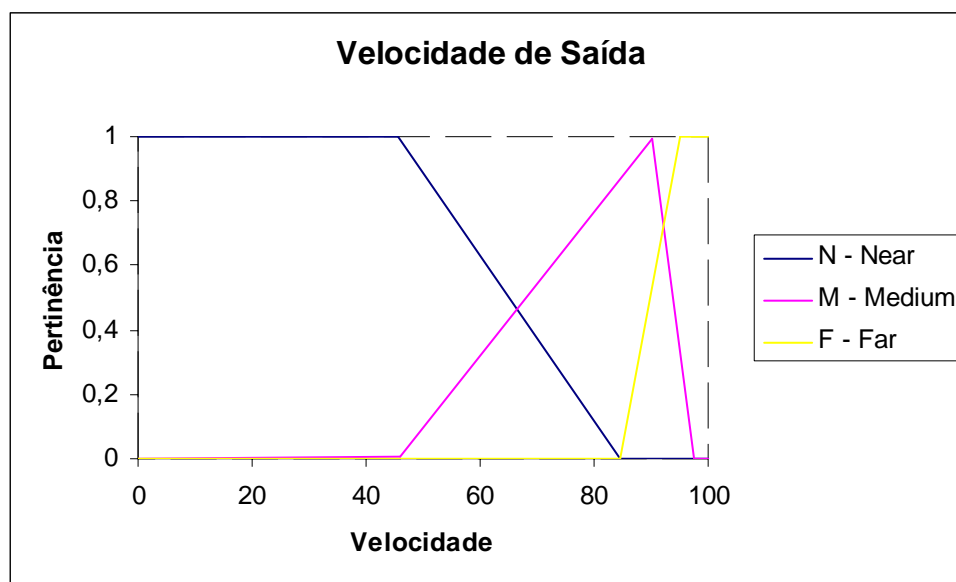


Figura 56 – Variável Conseqüente Velocidade de Saída

Com esta variável antecedente o maior valor da função de pertinência M aumenta um pouco no eixo da velocidade e a função N avança neste eixo.

Dados da Simulação 18:

Arquivo de interface fuzzy = interface_rob009.dat

Arquivo dos sensores = sens3.txt

Seleção de sensores = menor valor

Escala dos sensores = 0.7

Percentual da distância para inferência fuzzy(%) = 95%

Velocidade linear = 0.25

Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

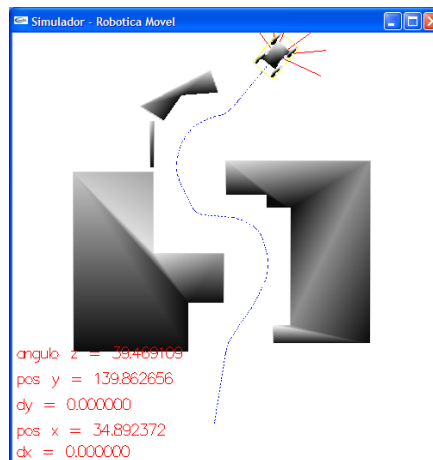


Figura 57 - Simulação 18

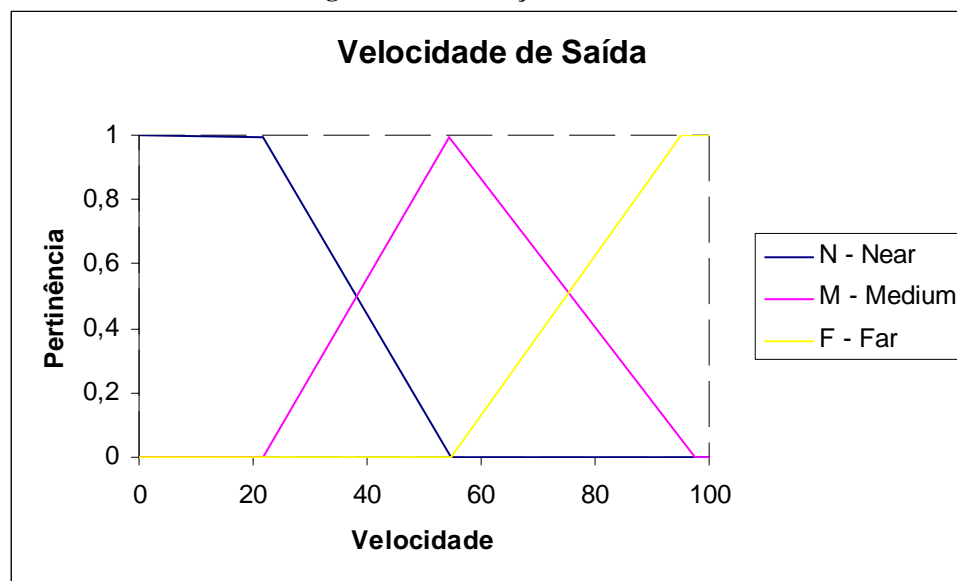


Figura 58 – Variável Conseqüente Velocidade de Saída

Com esta variável antecedente o maior valor da função de pertinência M diminuiu no eixo da velocidade.

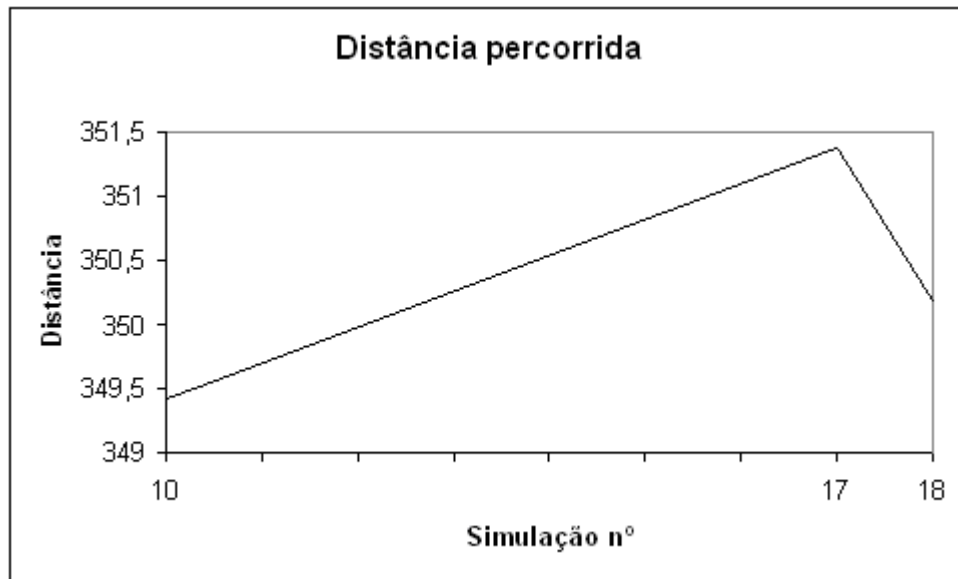


Figura 59 – Distância percorrida nas simulações 10,17 e 18

A simulação 17 apresenta um valor um pouco maior de distância.

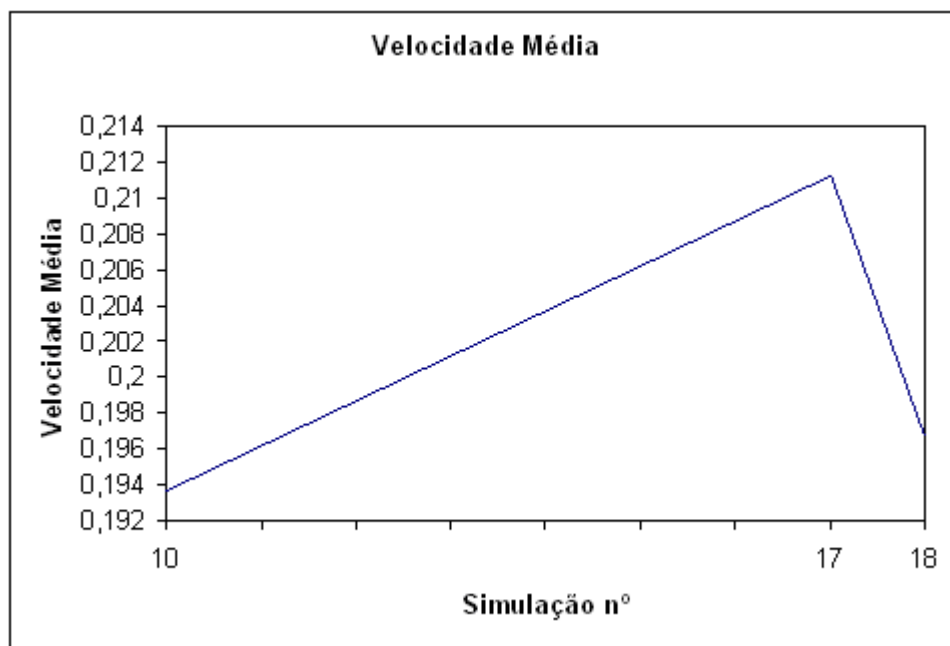


Figura 60 – Velocidade Média nas simulações 10,17 e 18

A simulação 17 apresenta uma velocidade média um pouco maior.

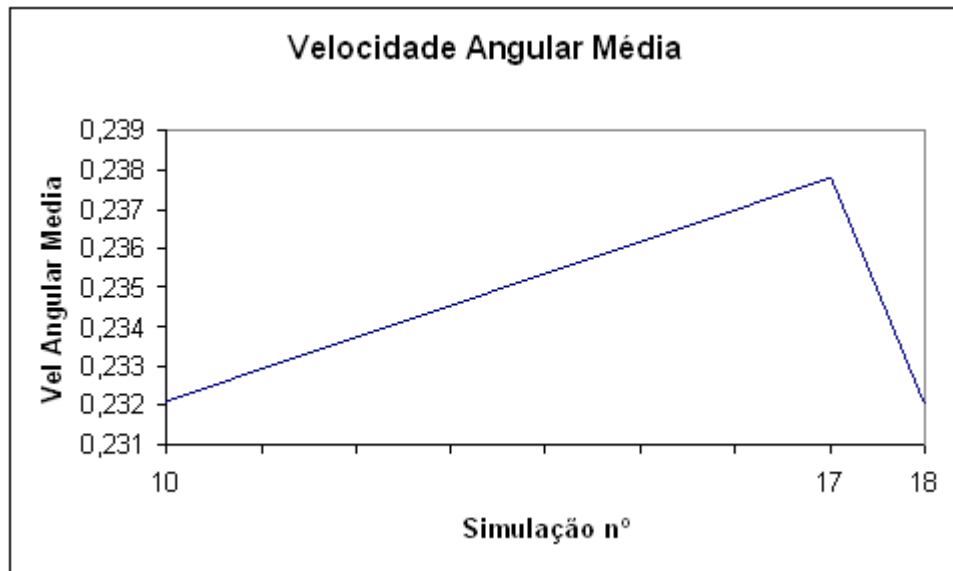


Figura 61 – Velocidade Ângular Média nas simulações 10,17 e 18

A simulação 17 apresenta uma velocidade angular média um pouco maior.

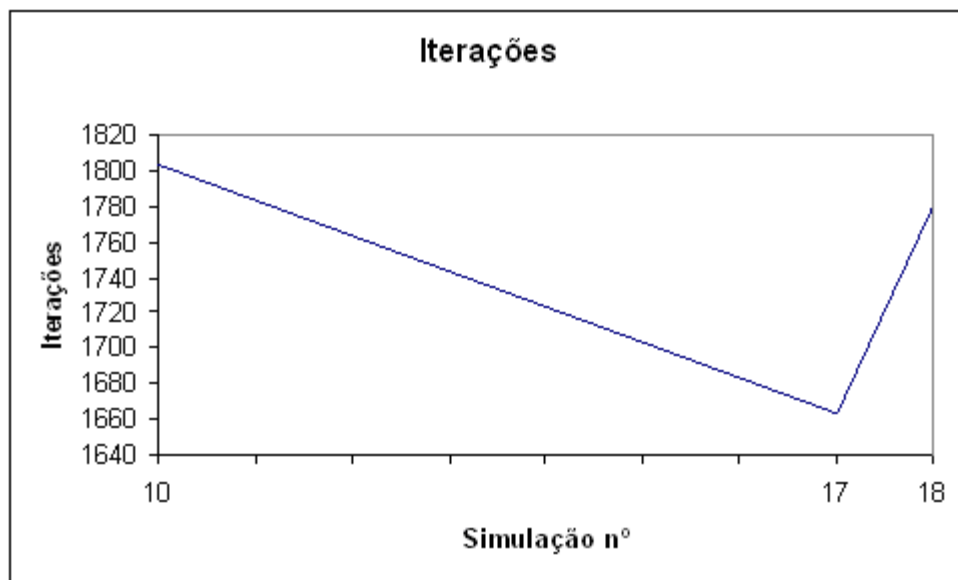


Figura 62 – Iterações nas simulações 10,17 e 18

A simulação 17 apresenta um valor menor nas iterações.

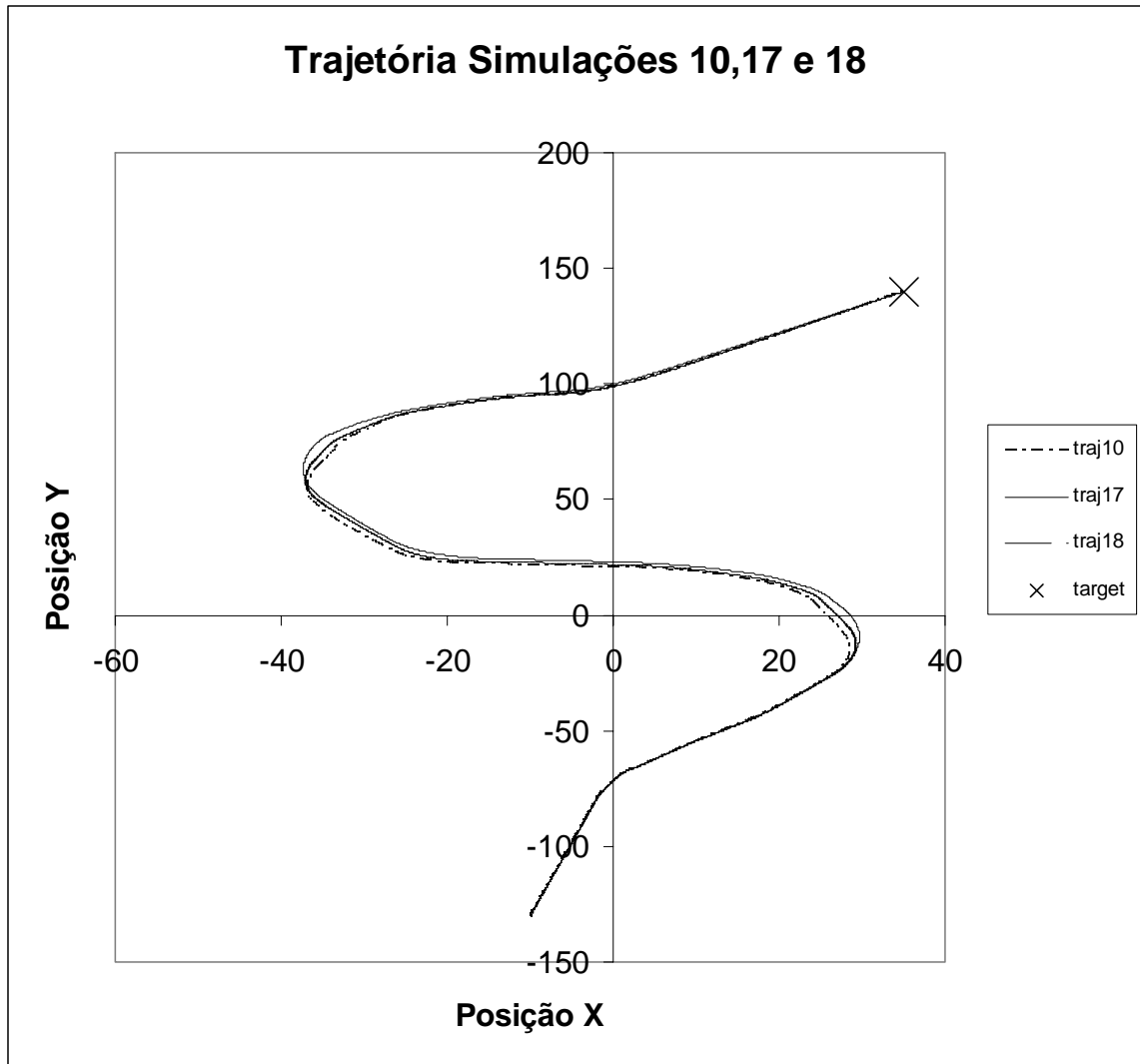


Figura 63 – Trajetórias das simulações 10,17 e 18

A simulação 17 apresenta uma velocidade média e angular maior, contudo, apresenta uma distância percorrida um pouco maior em relação às demais. As trajetórias das simulações 10 e 18 não apresentam grande diferença, apenas que a trajetória da simulação 18 é um pouco mais externa devido a uma velocidade média ligeiramente maior que a simulação 10.

As simulações 19 e 20 visam verificar o efeito de modificações na variável velocidade angular de saída, aproximando e afastando da velocidade máxima de navegação do robô móvel.

A referência é a simulação 10 cuja variável velocidade angular de saída é mostrada abaixo.

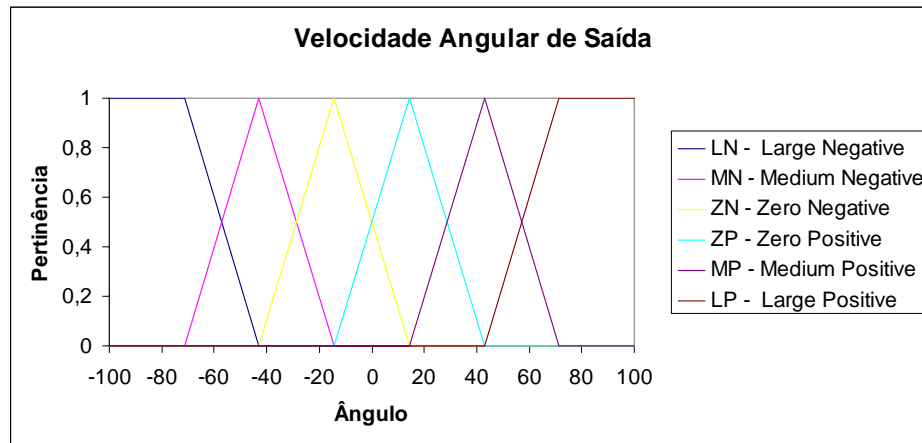


Figura 64 – Variável Consequente Velocidade Angular de Saída

Dados da Simulação 19:

Arquivo de interface fuzzy = interface_rob010.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

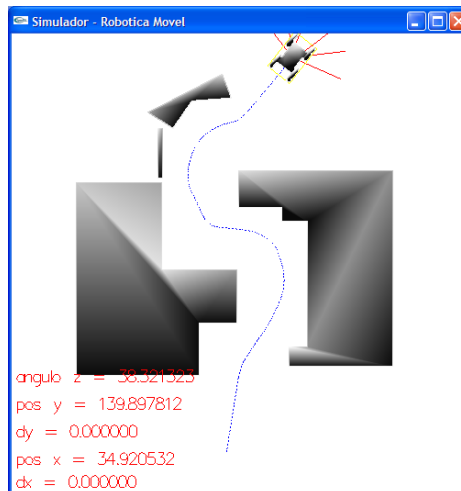


Figura 65 - Simulação 19

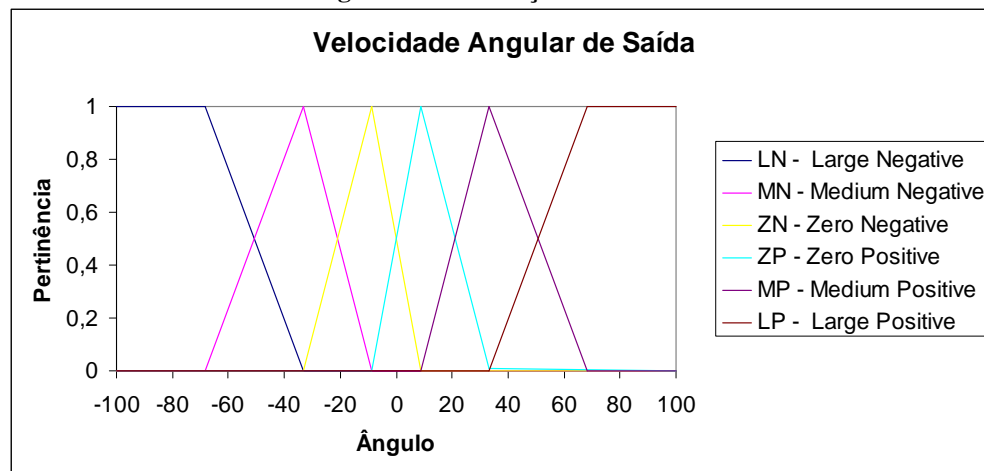


Figura 66 – Variável Conseqüente Velocidade Angular de Saída

A variável conseqüente velocidade angular de saída neste caso foi modificada de modo que as funções de pertinência sejam mais próximas de zero.

Dados da Simulação 20:

Arquivo de interface fuzzy = interface_rob011.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 1.8
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene03.3ds
 Arquivo de trajetória = trajeto01.txt

Posição inicial $X = -10.0$ e $Y = -130.0$
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 5.0

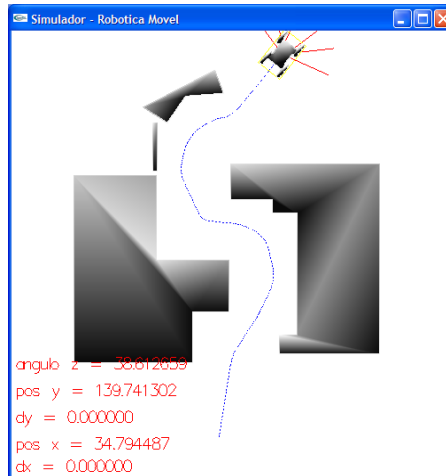


Figura 67 - Simulação 20

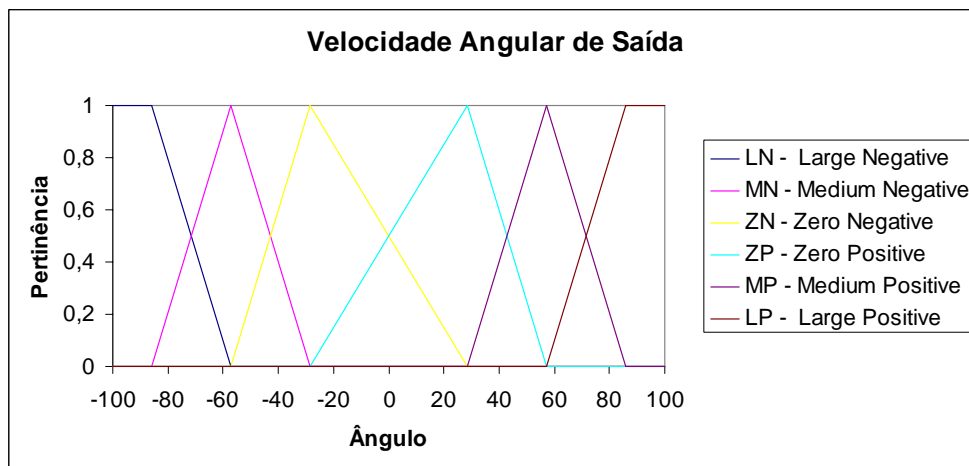


Figura 68 – Variável Conseqüente Velocidade Angular de Saída

A variável conseqüente velocidade angular de saída neste caso foi modificada de modo que as funções de pertinência sejam mais afastadas de zero.

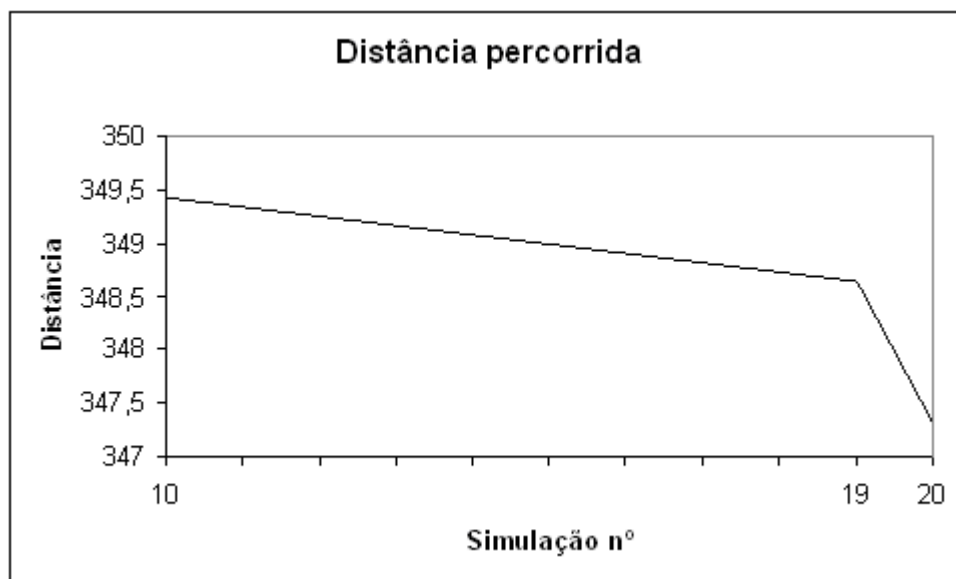


Figura 69 – Distância percorrida nas simulações 10,19 e 20

A simulação 10 apresenta um valor um pouco maior de distância e simulação 20 um valor menor.

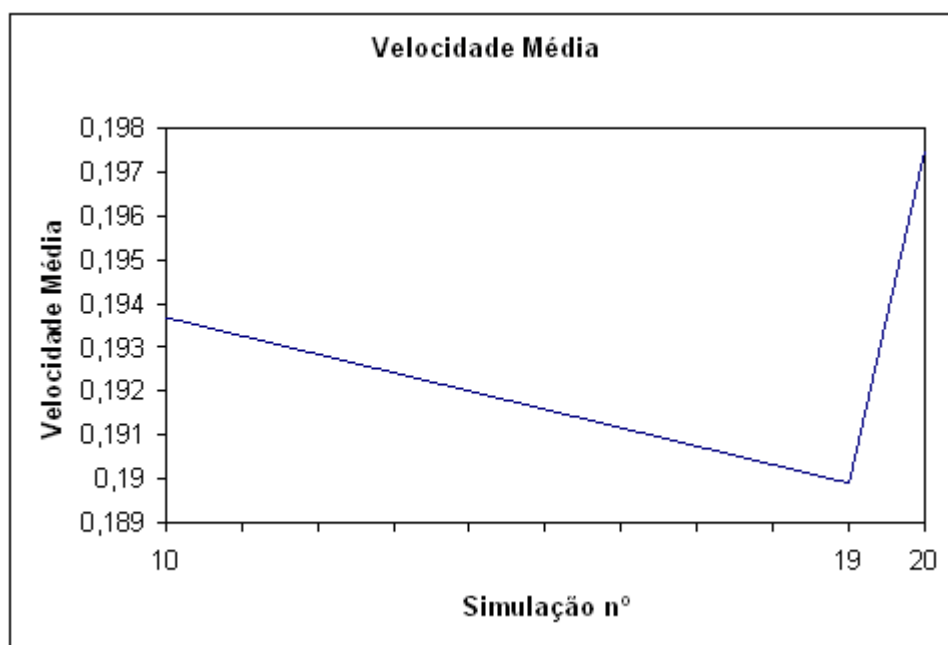


Figura 70 – Velocidade Média nas simulações 10,19 e 20

A simulação 19 apresenta uma velocidade média um pouco menor e a simulação 20 um valor um pouco maior.

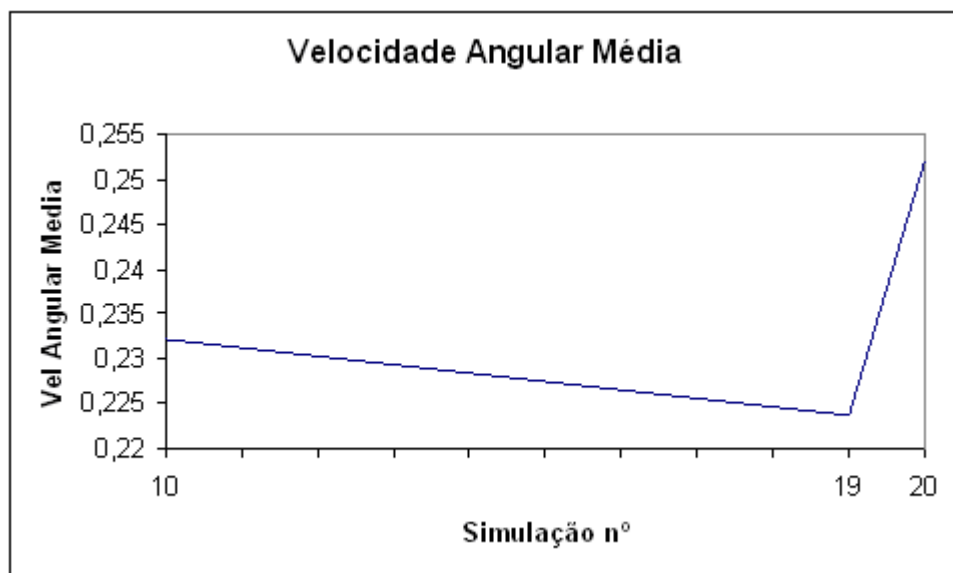


Figura 71 – Velocidade Ângular Média nas simulações 10,19 e 20

A simulação 19 apresenta uma velocidade angular média um pouco menor e a simulação 20 uma velocidade angular média maior.

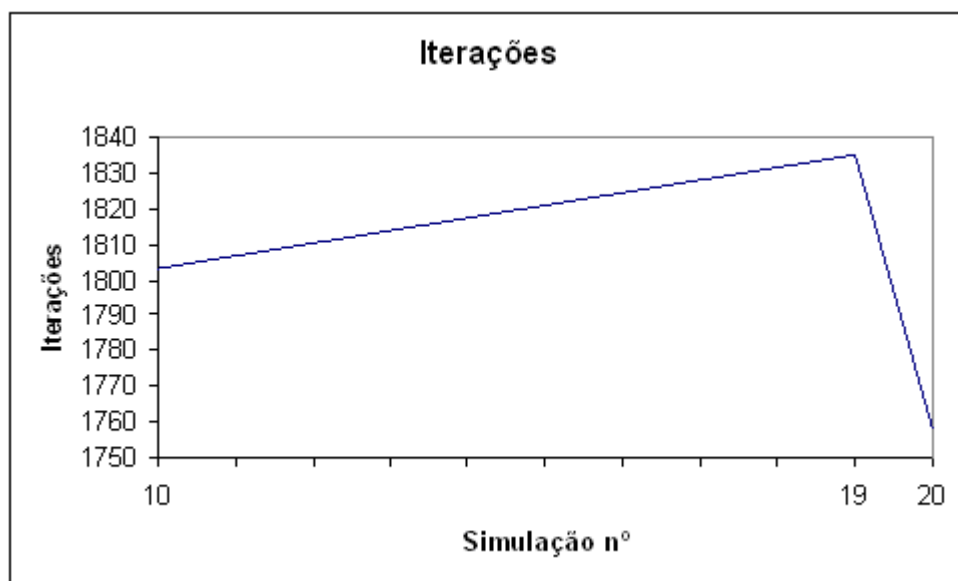


Figura 72 – Iterações nas simulações 10,19 e 20

A simulação 19 apresenta um valor maior nas iterações e a simulação 20 um valor menor.

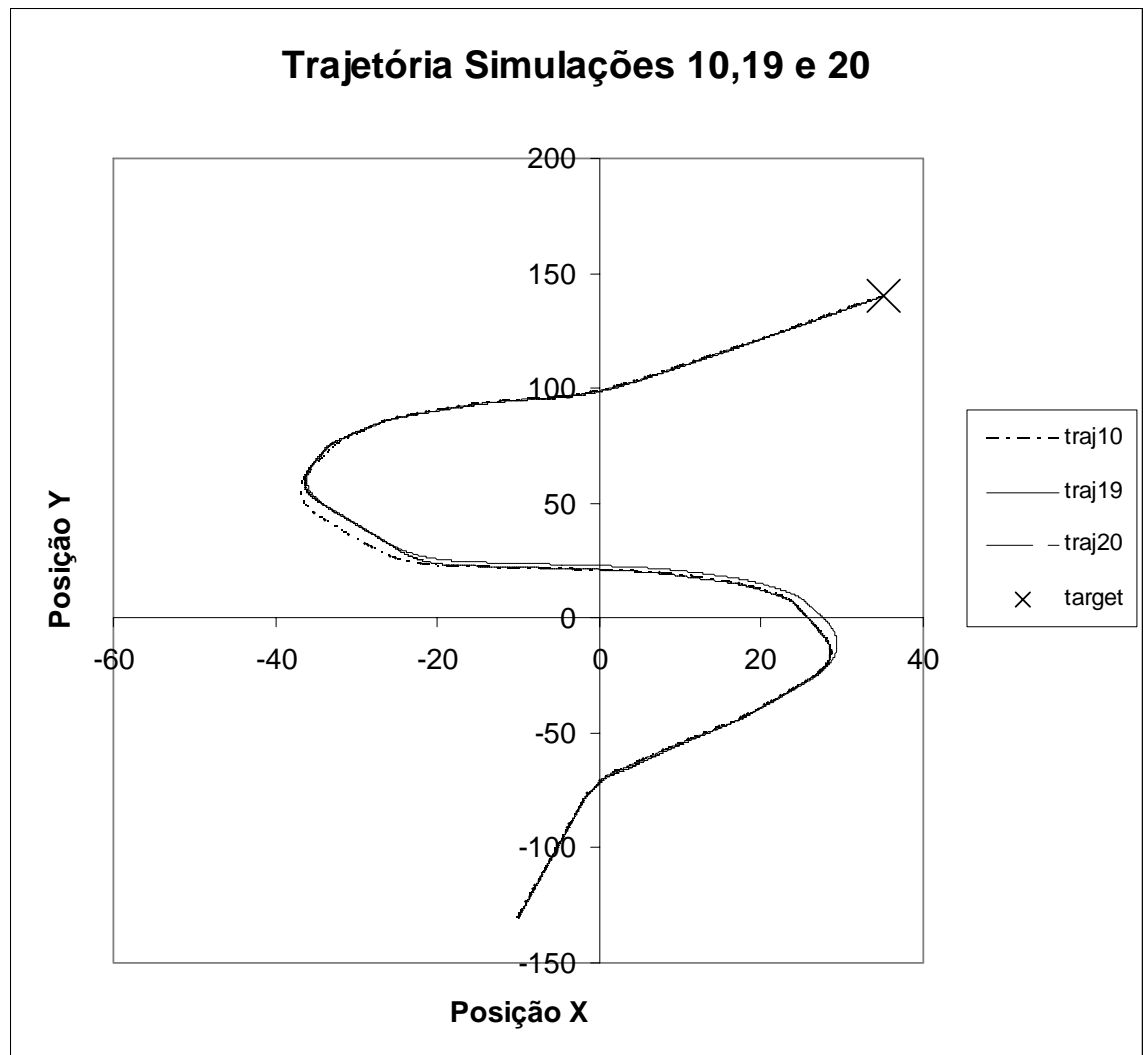


Figura 73 – Trajetórias das simulações 10,19 e 20

Tendo como base a trajetória 10 é possível perceber que a trajetória 19 segue mais externa e a simulação 20 mais interna.

Neurofuzzy

As próximas simulações tratam da funcionalidade neurofuzzy do simulador de robótica móvel. As simulações 21 e 22 apenas geram os dados para o treinamento neurofuzzy.

Dados da Simulação 21:

Arquivo de interface fuzzy = interface_robneuro01.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 0.5
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Arquivo gerado para treino = treinoneuro01.txt
Número de padrões gerados = 200
Arquivo de trajetória = trajeto01.txt
Posição inicial X = -10.0 e Y = -130.0
Target_X = 35.0 e Y = 140.0
Percentual de ruído nos sensores = 10.0

Regras e valores de saída crisp

Esta simulação utiliza as mesmas variáveis antecedentes da simulação 1 com a diferença de os consequentes serem valores crisp.

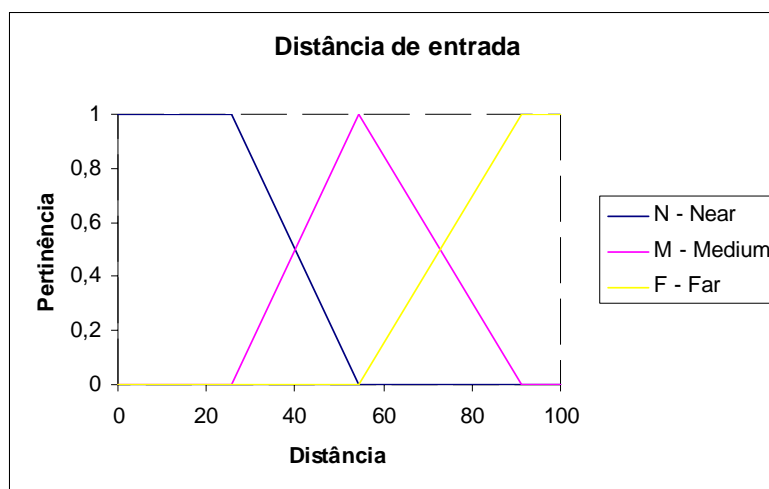


Figura 74 - Variável Antecedente Distância de Entrada

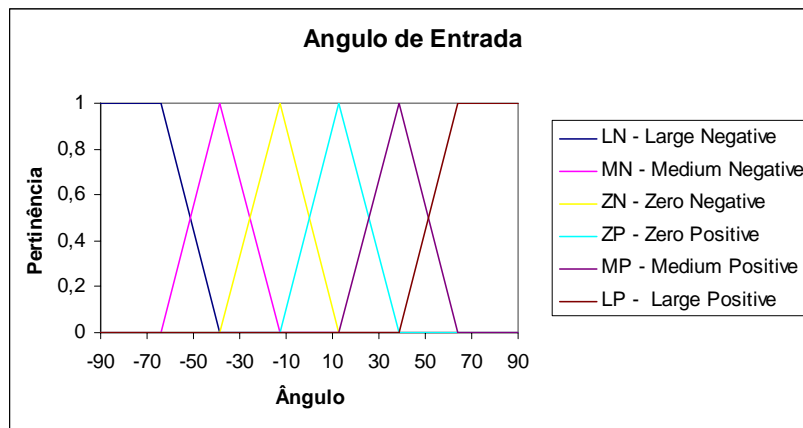


Figura 75 – Variável Antecedente Ângulo de Entrada

- Regra 1: N **AND** ZP **THEN** 30 **AND** -90
 Regra 2: M **AND** ZP **THEN** 90 **AND** -90
 Regra 3: N **AND** MP **THEN** 30 **AND** -30
 Regra 4: M **AND** MP **THEN** 90 **AND** -90
 Regra 5: N **AND** ZN **THEN** 30 **AND** 90
 Regra 6: M **AND** ZN **THEN** 90 **AND** 90
 Regra 7: N **AND** MN **THEN** 30 **AND** 30
 Regra 8: M **AND** MN **THEN** 90 **AND** 90
 Regra 9: N **AND** LP **THEN** 90 **AND** -30
 Regra 10: N **AND** LN **THEN** 90 **AND** 30

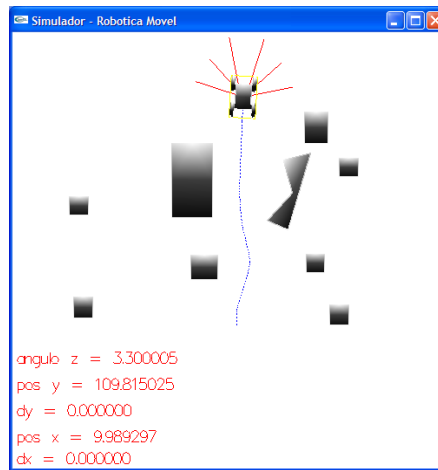


Figura 76 - Simulação 21

Dados da Simulação 22:

Arquivo de interface fuzzy = interface_robneuro02.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 0.5
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene01.3ds
 Arquivo gerado para treino = treinoneuro02.txt
 Número de padrões gerados = 200
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 10.0

Regras e valores de saída crisp

Esta simulação utiliza variáveis antecedentes com funções de pertinência do tipo gaussianas e nos consequentes valores crisp.

Regra 1: N **AND** ZP **THEN** 30 **AND** -90

Regra 2: M **AND** ZP **THEN** 90 **AND** -90

Regra 3: N AND MP THEN 30 AND -30
 Regra 4: M AND MP THEN 90 AND -90
 Regra 5: N AND ZN THEN 30 AND 90
 Regra 6: M AND ZN THEN 90 AND 90
 Regra 7: N AND MN THEN 30 AND 30
 Regra 8: M AND MN THEN 90 AND 90
 Regra 9: N AND LP THEN 90 AND -30
 Regra 10: N AND LN THEN 90 AND 30

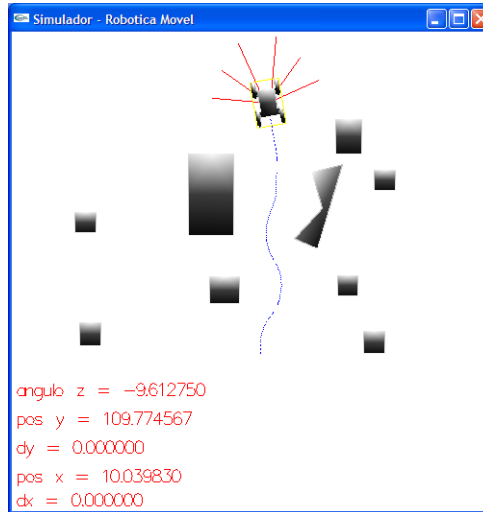


Figura 77 - Simulação 22

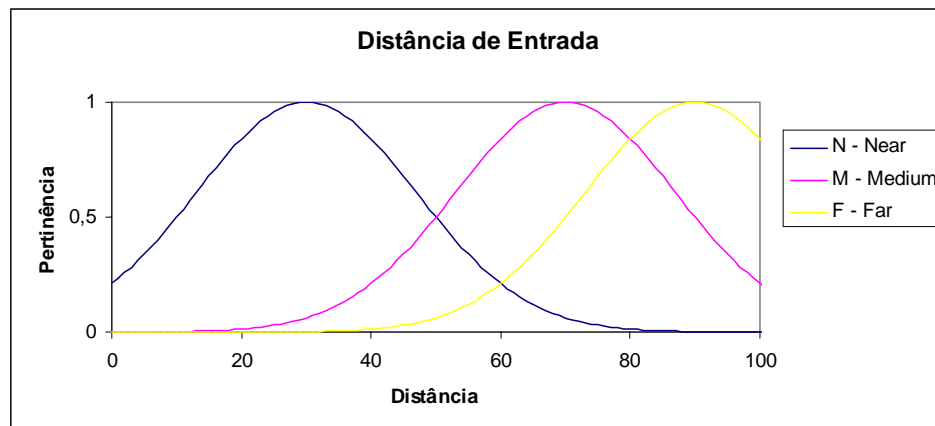


Figura 78 – Variável Antecedente Distância de Entrada

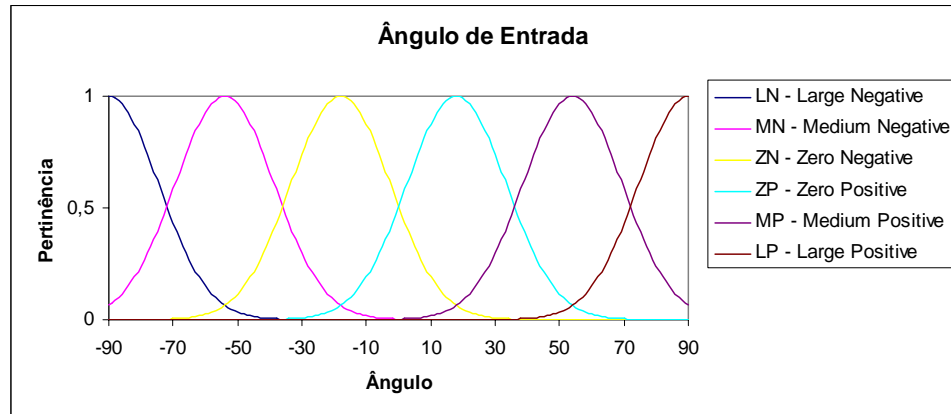


Figura 79 – Variável Antecedente Ângulo de Entrada

Dados da Simulação 23:

Esta simulação visa o aprendizado apenas dos valores crisp dos consequentes que são os valores z_i .

Arquivo de interface fuzzy = interface_robneurotr01.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 0.5
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene01.3ds
 Taxa de aprendizado $L_z = 0.05$
 Taxa de aprendizado $L_a = 0$ (não corrige os parâmetros a)
 Taxa de aprendizado $L_b = 0$ (não corrige os parâmetros b)
 Arquivo para treino = treinoneuro01.txt
 Numero de Epocas = 500
 Arquivo de trajetória = trajeto01.txt
 Posição inicial $X = -10.0$ e $Y = -130.0$
 Target $X = 35.0$ e $Y = 140.0$
 Percentual de ruído nos sensores = 10.0

Regras e valores de saída crisp

Esta simulação utiliza variáveis antecedentes com funções de pertinência do tipo gaussianas e nos conseqüentes valores crisp sem lógica para aprender as saídas crisp que fazem o desvio de obstáculos de acordo com o conjunto de dados provenientes da simulação 21. Antes do treinamento o robô móvel ficava preso ao passar perto de um dos obstáculos a esquerda devido a regra 8 que manda uma velocidade zero para o robô.

Regra 1: N **AND** ZP **THEN** 13 **AND** 1
Regra 2: M **AND** ZP **THEN** 11 **AND** 1
Regra 3: N **AND** MP **THEN** 110 **AND** 10
Regra 4: M **AND** MP **THEN** 15 **AND** 10
Regra 5: N **AND** ZN **THEN** 110 **AND** 1
Regra 6: M **AND** ZN **THEN** 110 **AND** 20
Regra 7: N **AND** MN **THEN** 11 **AND** 2
Regra 8: M **AND** MN **THEN** 0 **AND** 10
Regra 9: N **AND** LP **THEN** 110 **AND** 1
Regra 10: N **AND** LN **THEN** 10 **AND** 10

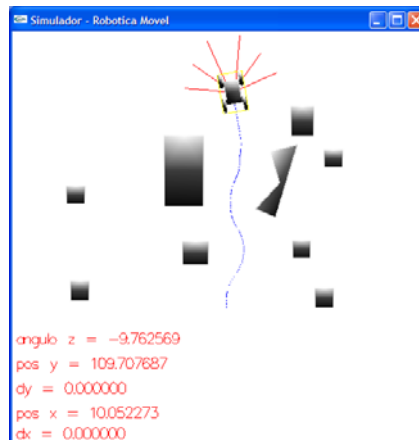


Figura 80 - Simulação 23

Dados da Simulação 24:

Esta simulação visa o aprendizado apenas dos valores crisp dos conseqüentes que são os valores z_i .

Arquivo de interface fuzzy = interface_robneurotr01.dat

Arquivo dos sensores = sens3.txt

Seleção de sensores = menor valor

Escala dos sensores = 0.7

Percentual da distância para inferência fuzzy(%) = 95%

Velocidade linear = 0.25

Velocidade angular = 0.5
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Taxa de aprendizado Lz = 0.05
Taxa de aprendizado La = 0 (não corrige os parâmetros a)
Taxa de aprendizado Lb = 0 (não corrige os parâmetros b)
Arquivo para treino = treinoneuro02.txt
Numero de Epocas = 500
Arquivo de trajetória = trajeto01.txt
Posição inicial X = -10.0 e Y = -130.0
Target_X = 35.0 e Y = 140.0
Percentual de ruído nos sensores = 10.0

Regras e valores de saída crisp

Esta simulação utiliza variáveis antecedentes com funções de pertinência do tipo gaussianas e nos consequentes valores crisp sem lógica para aprender as saídas crisp que fazem o desvio de obstáculos de acordo com o conjunto de dados provenientes da simulação 22. Antes do treinamento o robô móvel ficava preso ao passar perto de um dos obstáculos a esquerda devido a regra 8 que manda uma velocidade zero para o robô.

Regra 1: N **AND** ZP **THEN** 13 **AND** 1
Regra 2: M **AND** ZP **THEN** 11 **AND** 1
Regra 3: N **AND** MP **THEN** 110 **AND** 10
Regra 4: M **AND** MP **THEN** 15 **AND** 10
Regra 5: N **AND** ZN **THEN** 110 **AND** 1
Regra 6: M **AND** ZN **THEN** 110 **AND** 20
Regra 7: N **AND** MN **THEN** 11 **AND** 2
Regra 8: M **AND** MN **THEN** 0 **AND** 10
Regra 9: N **AND** LP **THEN** 110 **AND** 1
Regra 10: N **AND** LN **THEN** 10 **AND** 10

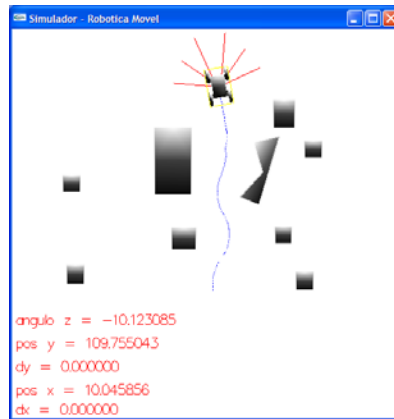


Figura 81 - Simulação 24

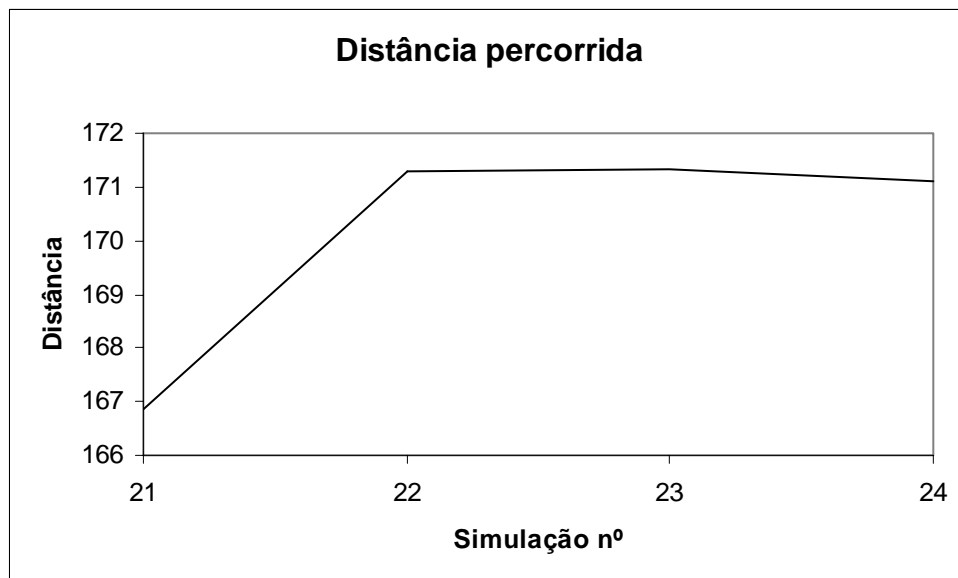


Figura 82 – Distância percorrida nas simulações 21, 22, 23 e 24

Apesar da pequena diferença a simulação 21 apresenta uma menor distância percorrida.

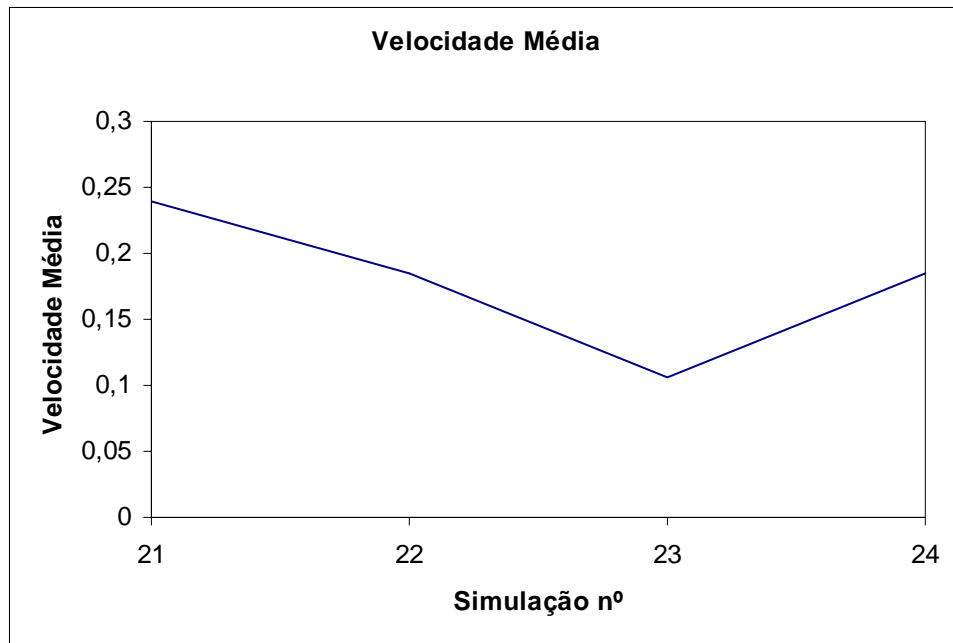


Figura 83 – Velocidade Média nas simulações 21, 22, 23 e 24

A simulação 23 apresenta a menor velocidade média.

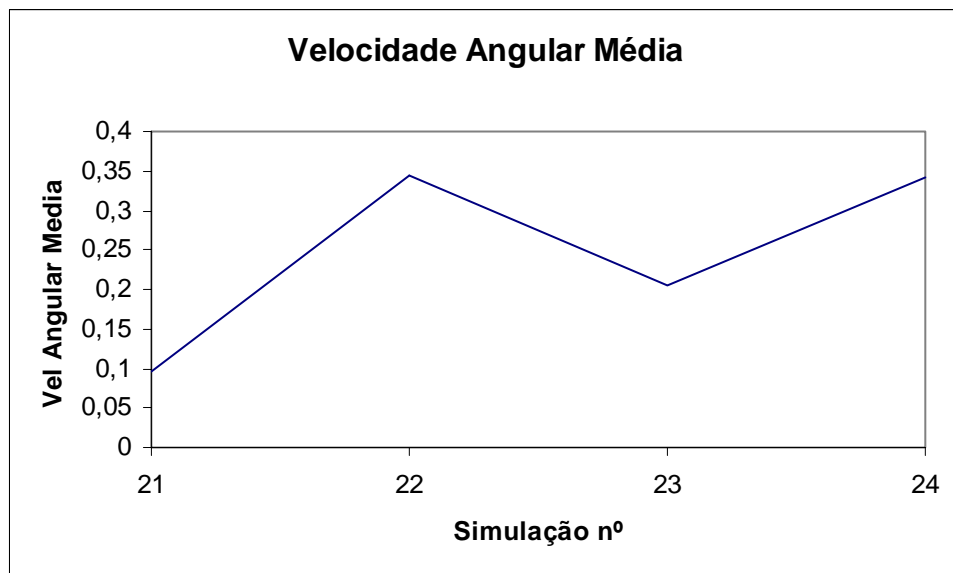


Figura 84 – Velocidade Angular Média nas simulações 21, 22, 23 e 24

As simulações 21 e 23 apresentam as menores velocidades angulares médias.

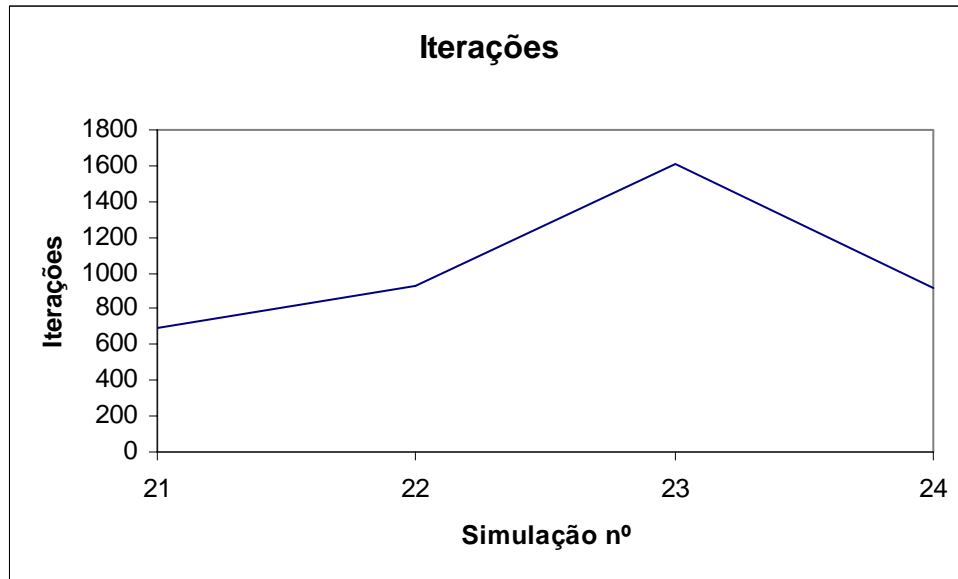


Figura 85 – Iterações nas simulações 21, 22, 23 e 24

As simulações 22 e 24 apresentam praticamente o mesmo número de iterações e a simulação 23 um maior número de iterações em relação a 21 que tem o menor número de iterações. Recapitulando a simulação 24 aprendeu com os dados da simulação 22 e a simulação 23 com os dados da 21.

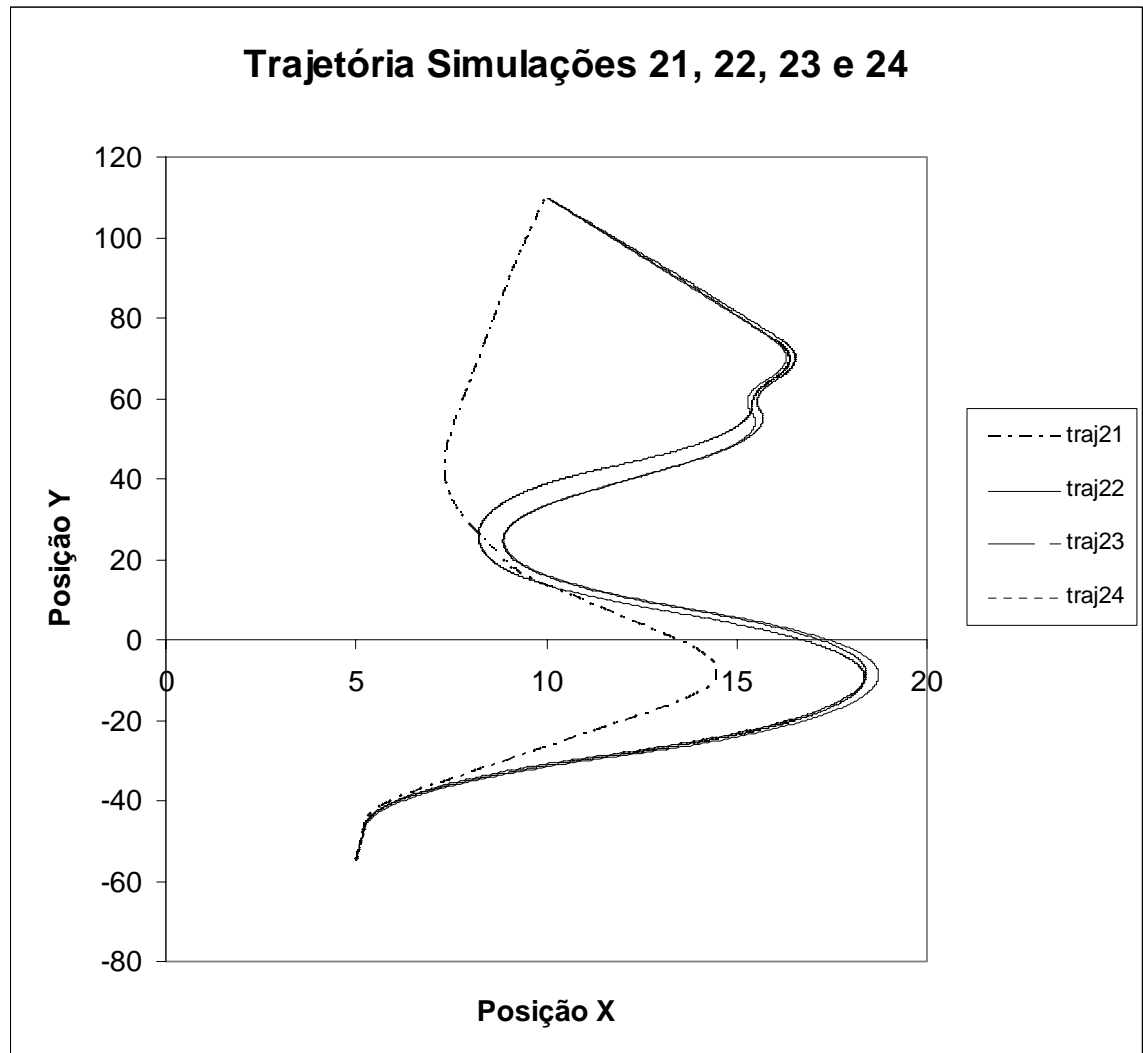


Figura 86 – Trajetórias das simulações 21, 22, 23 e 24

A simulação 21 apresenta a trajetória mais curta e a 23 tem a trajetória mais longa. As trajetórias 22 e 24 são praticamente idênticas.

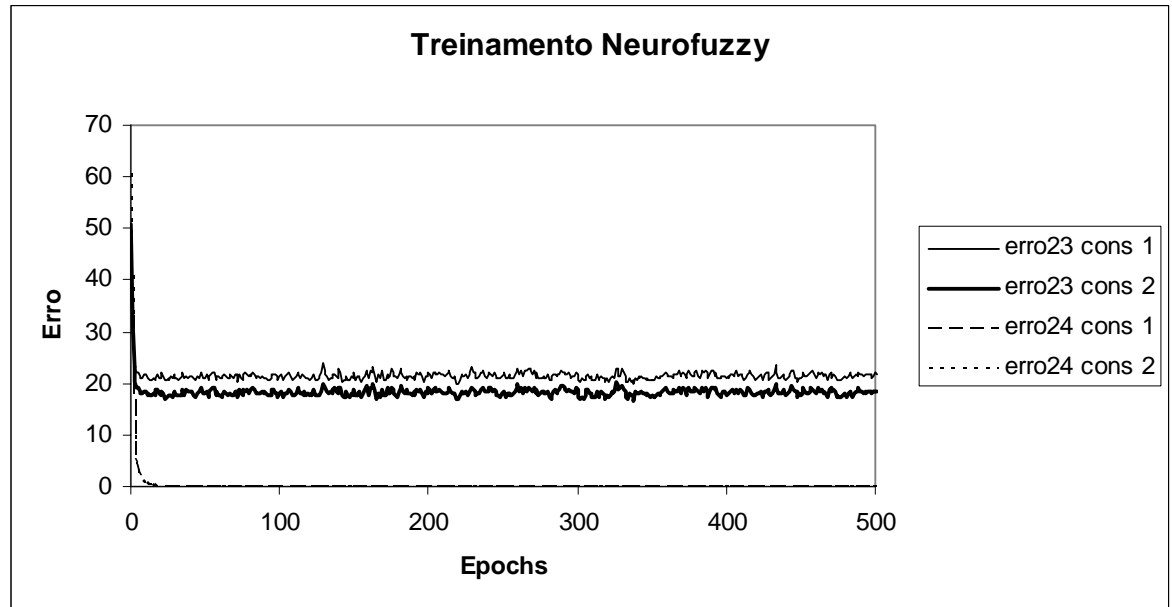


Figura 87 – Treinamento neurofuzzy

A simulação 23 apresenta um erro em torno de 20 e a simulação 24 um valor próximo de zero.

A simulação 25 apenas mostra o comportamento antes do treinamento com os dados da simulação 22.

Dados da Simulação 25:

Arquivo de interface fuzzy = interface_robneurotr02.dat
 Arquivo dos sensores = sens3.txt
 Seleção de sensores = menor valor
 Escala dos sensores = 0.7
 Percentual da distância para inferência fuzzy(%) = 95%
 Velocidade linear = 0.25
 Velocidade angular = 0.5
 Arquivo geometria robô móvel = robot2.3ds
 Arquivo geometria cenário = scene01.3ds
 Arquivo para treino = treinoneuro02.txt
 Numero de Epocas = 0
 Arquivo de trajetória = trajeto01.txt
 Posição inicial X = -10.0 e Y = -130.0
 Target_X = 35.0 e Y = 140.0
 Percentual de ruído nos sensores = 10.0

Regras e valores de saída crisp

Esta simulação utiliza variáveis antecedentes com funções de pertinência do tipo gaussianas e nos consequentes valores crisp com uma configuração previa de acordo com um conhecimento sofre uma sintonia nas saídas crisp e nas variáveis antecedentes no parâmetro 'a' para que ajustem o desvio de obstáculos de acordo com o conjunto de dados provenientes da simulação 22.

Regra 1: N **AND** ZP **THEN** 20 **AND** -80
Regra 2: M **AND** ZP **THEN** 80 **AND** -80
Regra 3: N **AND** MP **THEN** 20 **AND** -40
Regra 4: M **AND** MP **THEN** 80 **AND** -80
Regra 5: N **AND** ZN **THEN** 20 **AND** 80
Regra 6: M **AND** ZN **THEN** 80 **AND** 80
Regra 7: N **AND** MN **THEN** 20 **AND** 40
Regra 8: M **AND** MN **THEN** 80 **AND** 80
Regra 9: N **AND** LP **THEN** 80 **AND** -40
Regra 10: N **AND** LN **THEN** 20 **AND** 40

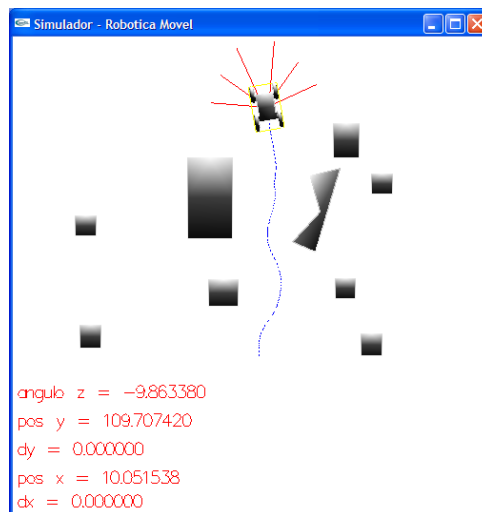


Figura 88 - Simulação 25

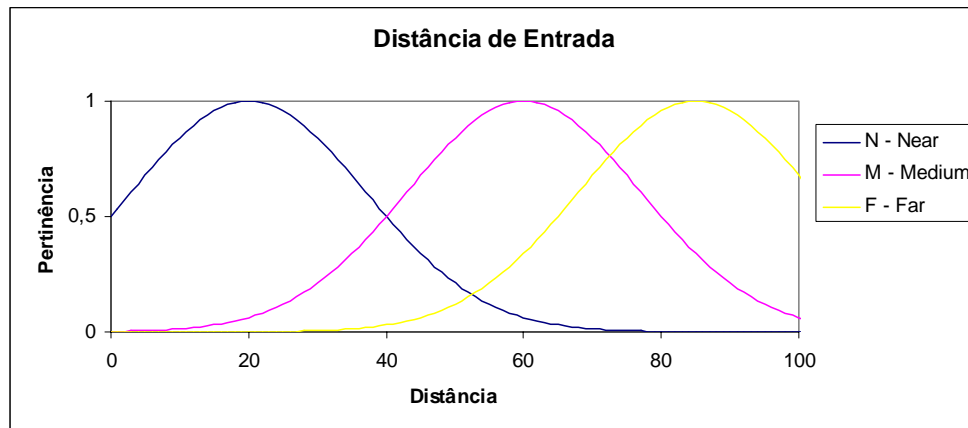


Figura 89 – Variável Antecedente Distância de Entrada

A variável antecedente distância de entrada difere da mesma variável da simulação 22 no parâmetro ‘a’ das funções de pertinência gaussianas.

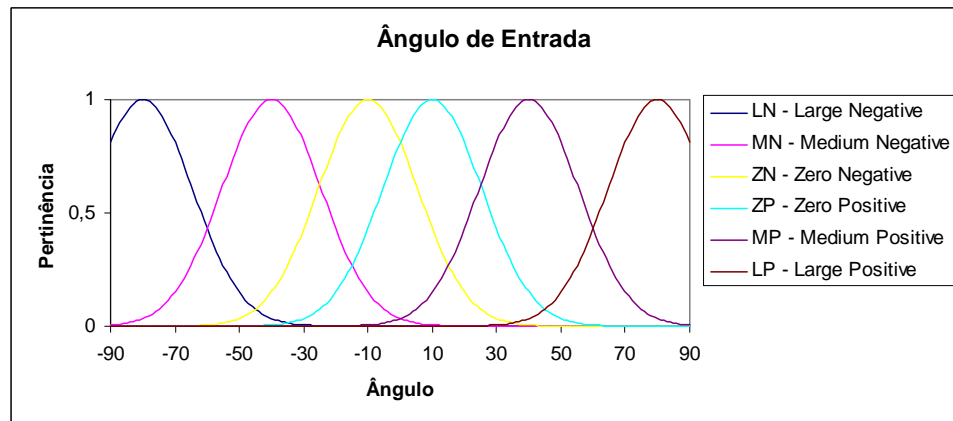


Figura 90 – Variável Antecedente Ângulo de Entrada

A variável antecedente ângulo de entrada difere da mesma variável da simulação 22 no parâmetro ‘a’ das funções de pertinência gaussianas.

A simulação 26 mostra o comportamento da configuração 25 após o treinamento com os dados da simulação 22 sendo que os parâmetros ‘z’ e ‘a’ serão modificados durante o treinamento.

Dados da Simulação 26:

Arquivo de interface fuzzy = interface_robneurotr02.dat

Arquivo dos sensores = sens3.txt

Seleção de sensores = menor valor

Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 0.5
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Taxa de aprendizado Lz = 0.001
Taxa de aprendizado La = 0.0005
Taxa de aprendizado Lb = 0 (não corrige os parâmetros b)
Arquivo para treino = treinoneuro02.txt
Numero de Epocas = 500
Arquivo de trajetória = trajeto01.txt
Posição inicial X = -10.0 e Y = -130.0
Target_X = 35.0 e Y = 140.0
Percentual de ruído nos sensores = 10.0

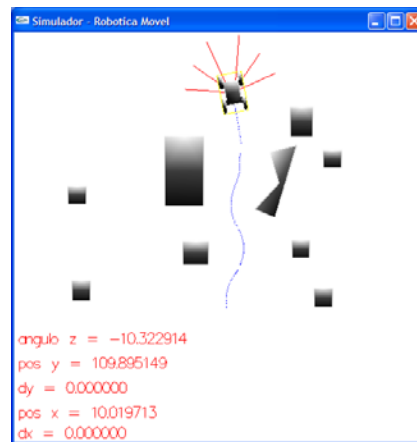


Figura 91 - Simulação 26

A simulação 27 apenas mostra o comportamento antes do treinamento com os dados da simulação 22.

Dados da Simulação 27:

Arquivo de interface fuzzy = interface_robneurotr03.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 0.5

Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Arquivo para treino = treinoneuro02.txt
Numero de Epocas = 0
Arquivo de trajetória = trajeto01.txt
Posição inicial X = -10.0 e Y = -130.0
Target_X = 35.0 e Y = 140.0
Percentual de ruído nos sensores = 10.0

Regras e valores de saída crisp

Esta simulação utiliza variáveis antecedentes com funções de pertinência do tipo gaussianas e nos consequentes valores crisp com uma configuração previa de acordo com um conhecimento sofre uma sintonia nas saídas crisp e nas variáveis antecedentes tanto no parâmetro 'a' como no 'b' para que ajustem o desvio de obstáculos de acordo com o conjunto de dados provenientes da simulação 22.

Regra 1: N **AND** ZP **THEN** 20 **AND** -80
Regra 2: M **AND** ZP **THEN** 80 **AND** -80
Regra 3: N **AND** MP **THEN** 20 **AND** -40
Regra 4: M **AND** MP **THEN** 80 **AND** -80
Regra 5: N **AND** ZN **THEN** 20 **AND** 80
Regra 6: M **AND** ZN **THEN** 80 **AND** 80
Regra 7: N **AND** MN **THEN** 20 **AND** 40
Regra 8: M **AND** MN **THEN** 80 **AND** 80
Regra 9: N **AND** LP **THEN** 80 **AND** -40
Regra 10: N **AND** LN **THEN** 20 **AND** 40

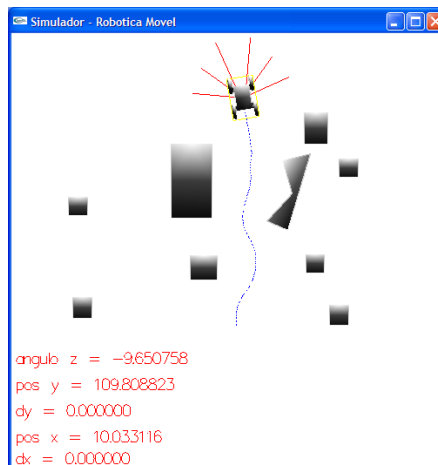


Figura 92 - Simulação 27

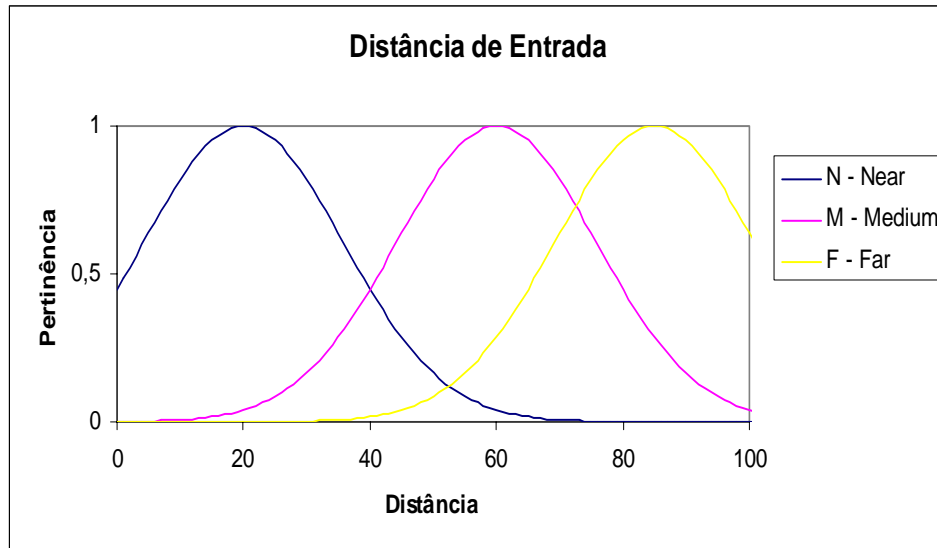


Figura 93 – Variável Antecedente Distância de Entrada

A variável antecedente distância de entrada difere da mesma variável da simulação 22 nos parâmetros ‘a’ e ‘b’ das funções de pertinência gaussianas.

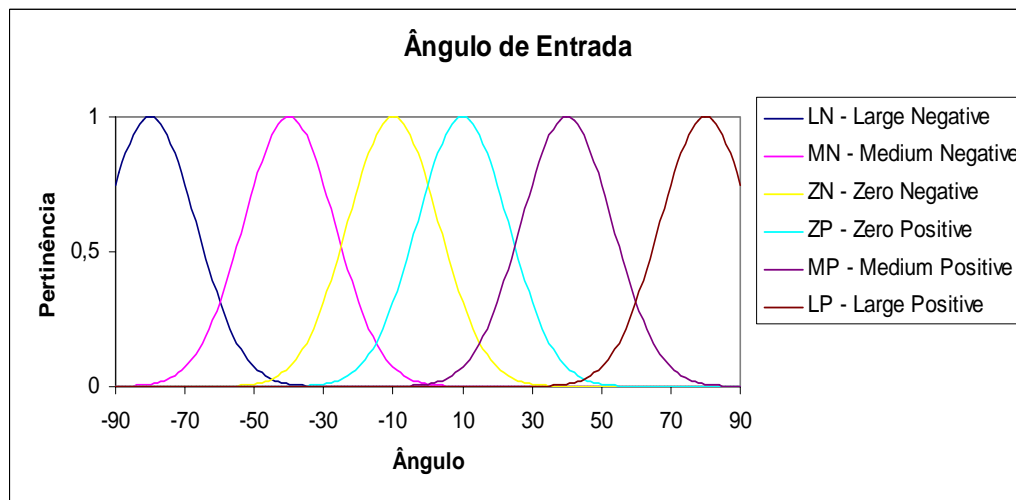


Figura 94 – Variável Antecedente Ângulo de Entrada

A variável antecedente ângulo de entrada difere da mesma variável da simulação 22 nos parâmetros ‘a’ e ‘b’ das funções de pertinência gaussianas.

A simulação 28 mostra o comportamento da configuração 27 após o treinamento com os dados da simulação 22 sendo que os parâmetros ‘z’, ‘a’ e ‘b’ serão modificados durante o treinamento.

Dados da Simulação 28:

Arquivo de interface fuzzy = interface_robneurotr03.dat
Arquivo dos sensores = sens3.txt
Seleção de sensores = menor valor
Escala dos sensores = 0.7
Percentual da distância para inferência fuzzy(%) = 95%
Velocidade linear = 0.25
Velocidade angular = 0.5
Arquivo geometria robô móvel = robot2.3ds
Arquivo geometria cenário = scene01.3ds
Taxa de aprendizado Lz = 0.001
Taxa de aprendizado La = 0.0005
Taxa de aprendizado Lb = 1e-10
Arquivo para treino = treinoneuro02.txt
Numero de Epocas = 500
Arquivo de trajetória = trajeto01.txt
Posição inicial X = -10.0 e Y = -130.0
Target_X = 35.0 e Y = 140.0
Percentual de ruído nos sensores = 10.0

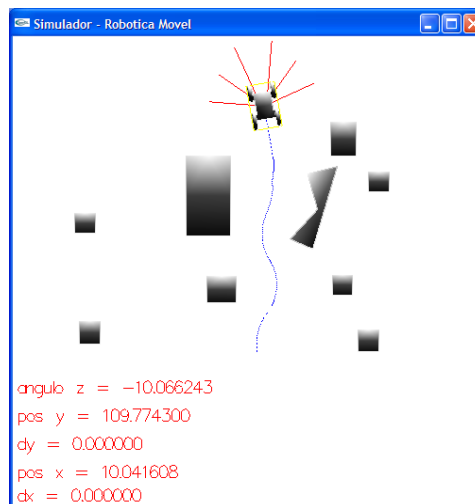


Figura 95 - Simulação 28

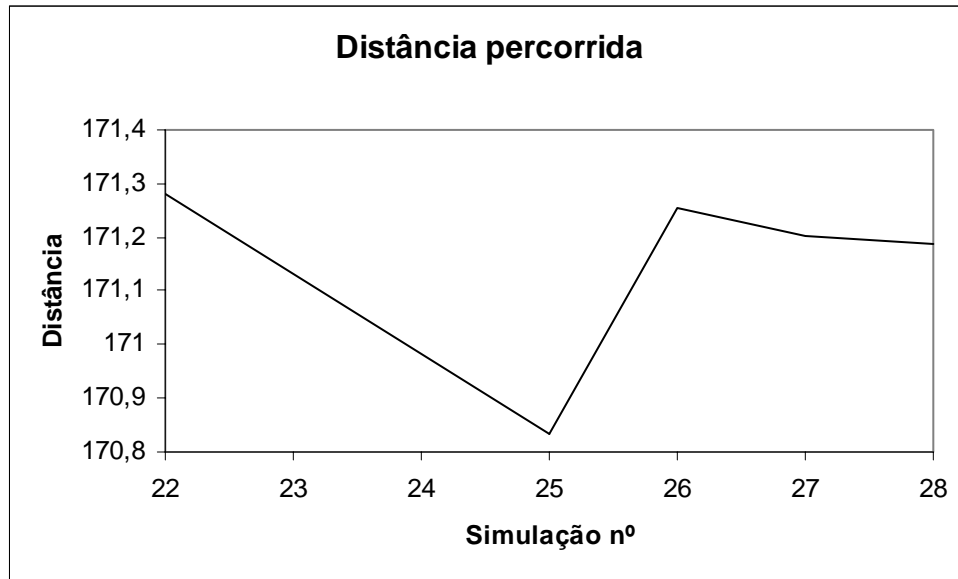


Figura 96 – Distância percorrida nas simulações 22, 25, 26, 27 e 28

Não apresenta mudança significativa nos valores de distância percorrida.

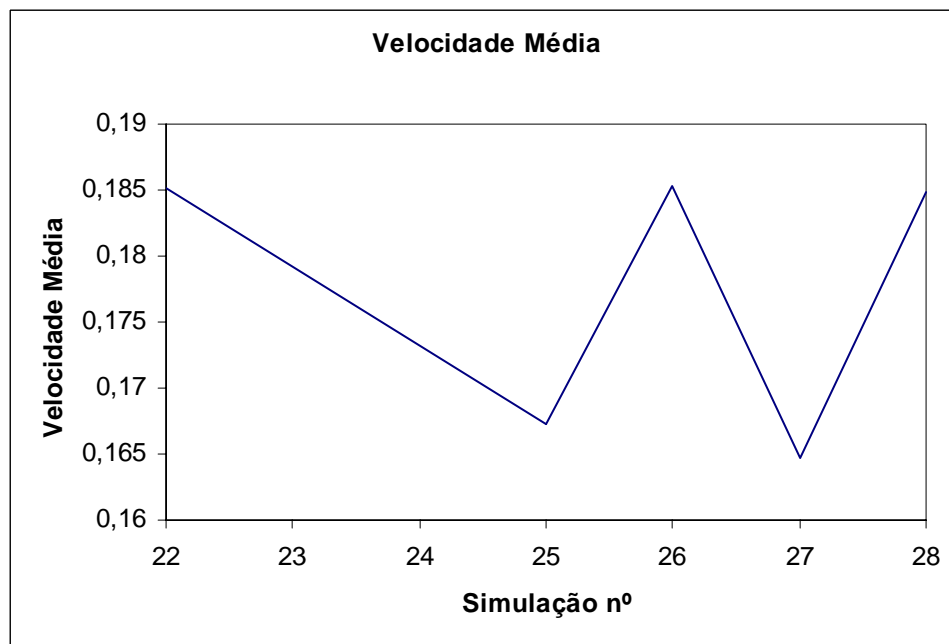


Figura 97 – Velocidade Média nas simulações 22, 25, 26, 27 e 28

As simulações 25 e 27 apresentam as menores velocidades médias e a 22, 26 e 28 apresentam as maiores velocidades.

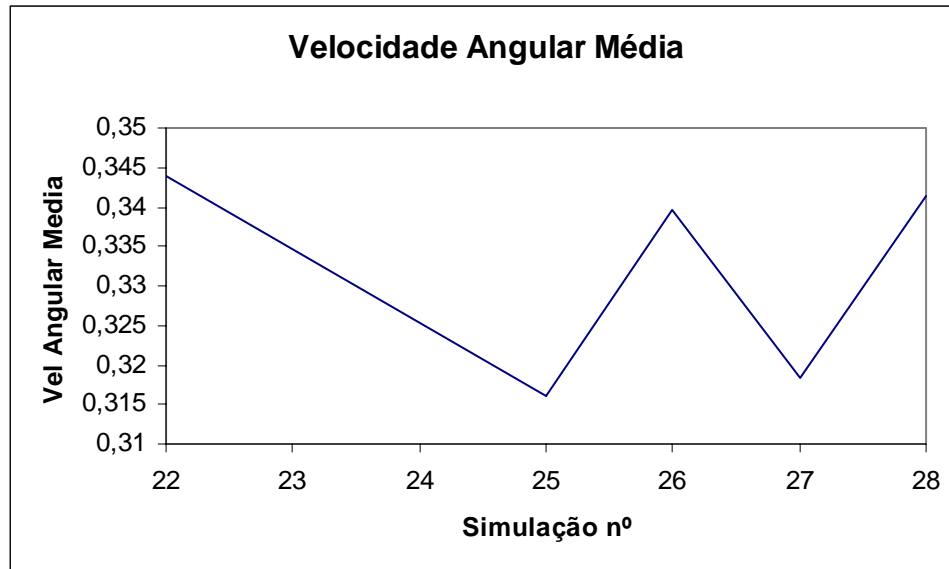


Figura 98 – Velocidade Angular Média nas simulações 22, 25, 26, 27 e 28

As simulações 25 e 27 apresentam as menores velocidades angulares médias e a 22, 26 e 28 apresentam as maiores velocidades angulares.

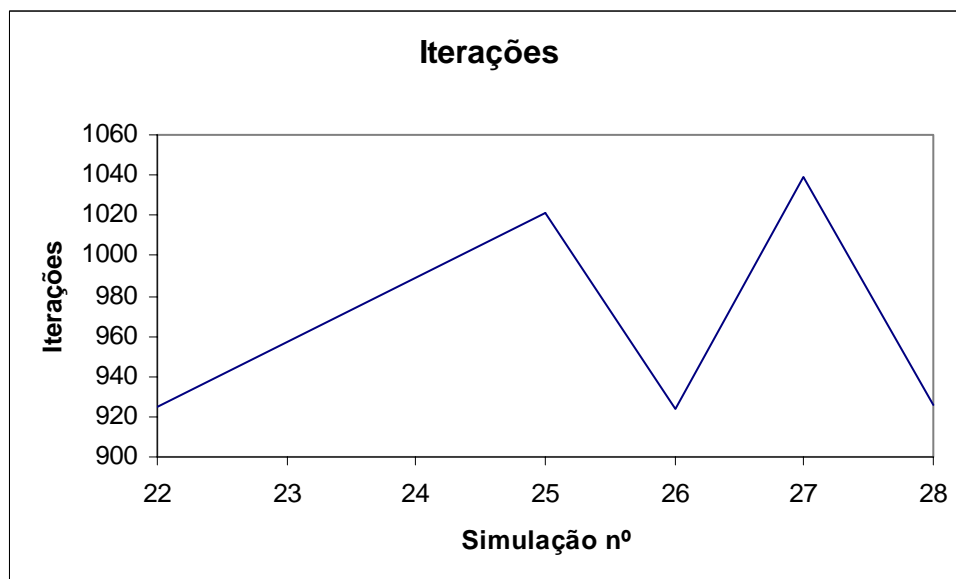


Figura 99 – Iterações nas simulações 22, 25, 26, 27 e 28

As simulações 25 e 27 apresentam os maiores números de iterações e as simulações 22, 26 e 28 apresentam os menores números de iterações.

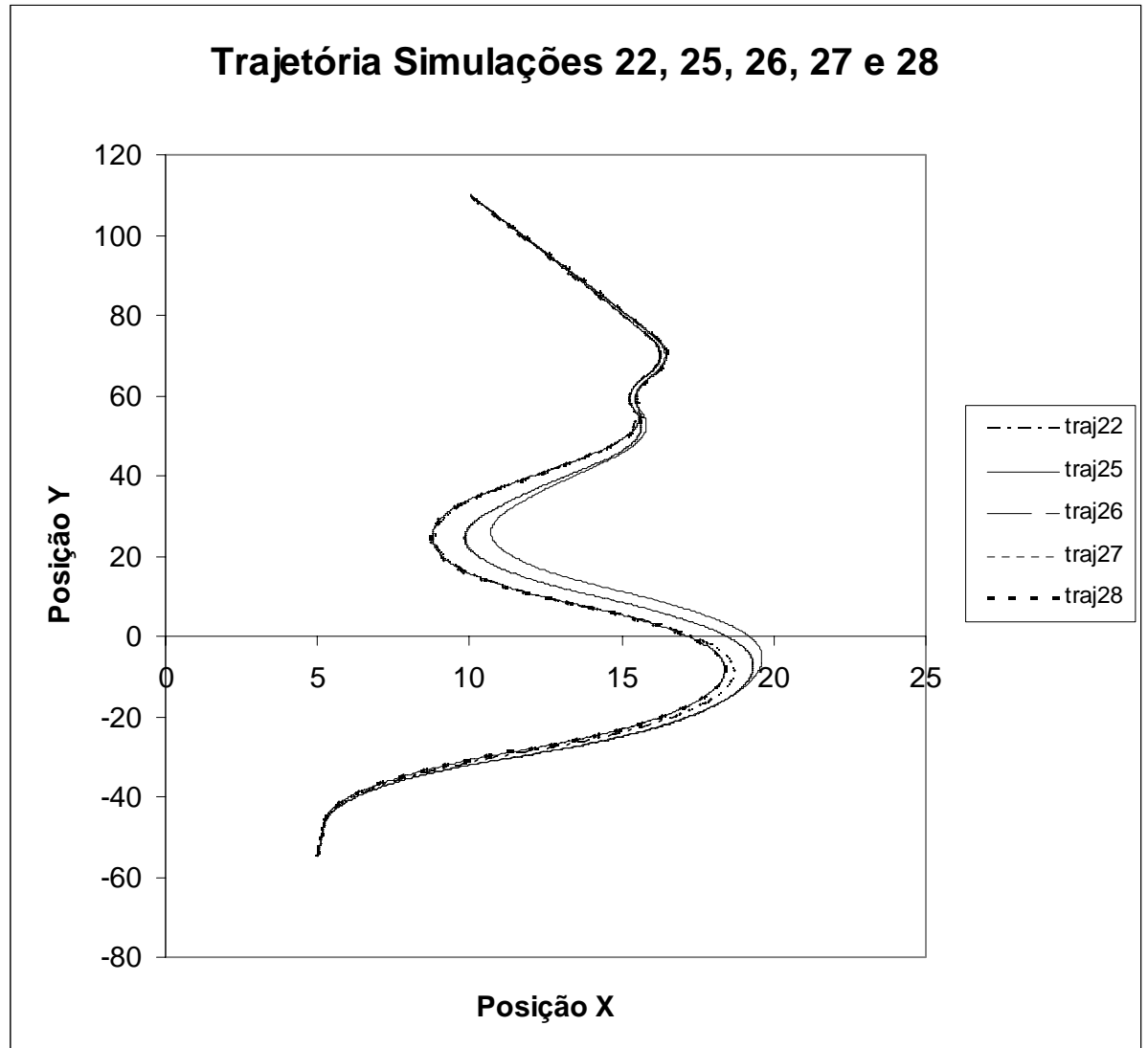


Figura 100 – Trajetórias das simulações 22, 25, 26, 27 e 28

As simulações 26 e 28 apresentam a trajetória muito semelhante a 22 que é o controlador que gerou o conjunto de dados de treinamento.

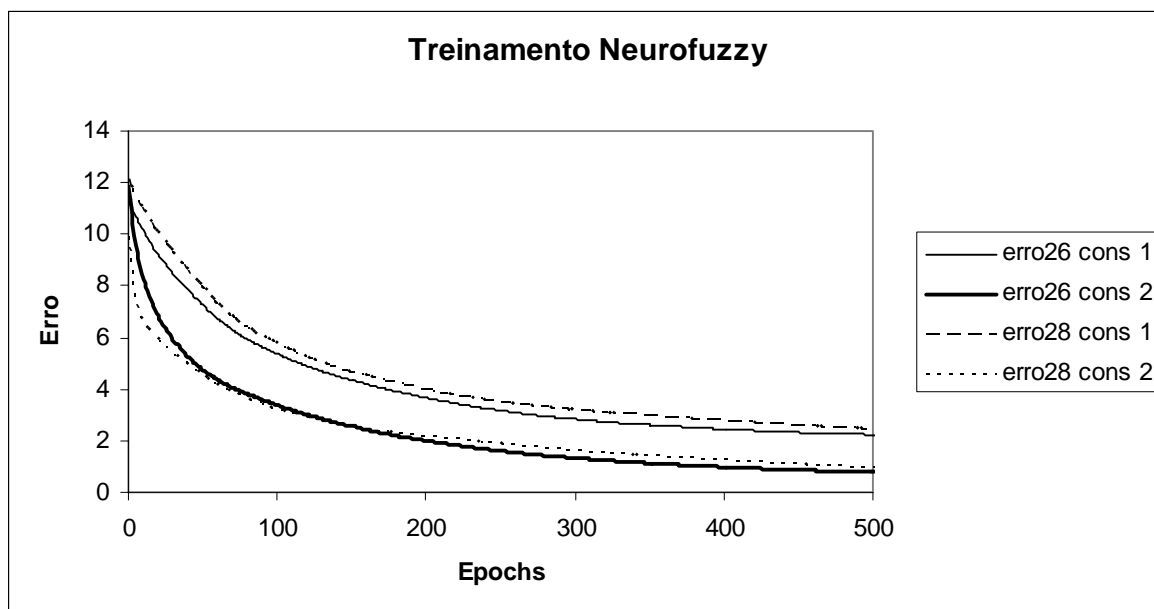


Figura 101 – Treinamento neurofuzzy

Os erros da simulações convergem para valores próximos de 2 em 500 épocas.

Capítulo 4 – CONCLUSÕES

Por meio das simulações número 01 até 09 pode-se concluir que o sistema fuzzy tem uma grande capacidade de trabalhar com dados ruidosos e mesmo no caso de maior ruído o robô móvel não colidiu com os obstáculos.

A respeito da variável antecedente distância de entrada nas simulações 10 até 14 é possível dizer que ela determina o ponto onde o sensor atua determinadas regras. Avançando a função M as regras associadas ao M podem atuar geralmente relacionadas a uma velocidade de saída M(medium). O mesmo com a função N cujas regras geralmente associam uma velocidade menor de saída. A simulação 14 apresenta uma maior velocidade media devido a sua função de pertinência M ser trapezoidal ou seja em uma região maior a distância é considerada média.

A respeito da variável antecedente ângulo de entrada os gráficos mostram que não houve modificação significativa no comportamento nas simulações. Tal fato se deve ao fato que as regras tanto para ângulos médios quanto para ângulos menores apresentam a mesma ação para distâncias médias. Tais regras (2 e 4) são ilustradas abaixo em negrito para ângulos positivo:

Regra 1: N AND ZP THEN N AND MN

Regra 2: M AND ZP THEN M AND MN

Regra 3: N AND MP THEN N AND ZN

Regra 4: M AND MP THEN M AND MN

A simulação 16 apresenta uma trajetória um pouco mais externa indicando uma menor reatividade aos obstáculos.

A respeito da variável consequente velocidade de saída nota-se uma relação grande com a velocidade média nas simulações 17 e 18 onde a configuração da simulação 17 com a função de pertinência “M” avançada no eixo das velocidades resultou em uma velocidade média maior em relação as simulações 10 e 18.

A respeito da variável consequente velocidade angular de saída por meio das simulações se percebe que a configuração 19 tem uma menor reação aos obstáculos e a configuração 20 tem uma maior reação e apresenta os melhores resultados com velocidade media e velocidade angular media além de uma distância percorrida menor. Pode-se concluir que quanto mais próximas estiverem as funções de pertinência da variável consequente velocidade angular de saída do valor zero menor a reação do simulador e quanto mais afastados mais reativo será o robô móvel.

Neurofuzzy

As trajetórias 22 e 24 são praticamente idênticas mostrando a convergência do erro no treinamento na simulação 24.

Mesmo com o erro do treinamento da simulação 23 estar próximo de 20 o controlador absorveu o conhecimento a respeito do desvio de obstáculos mesmo com a trajetória mais longa as reações foram corretas para o desvio dos obstáculos. Na simulação 24 o erro converge para um valor próximo de zero logo nas primeiras épocas.

As simulações 26 e 28 apresentam a trajetória muito semelhante a 22 que é o controlador que gerou o conjunto de dados de treinamento. Os erros da simulações convergem para valores próximos de 2 em 500 épocas. Sendo que este valor de erro já apresenta uma melhoria no funcionamento dos controladores comprovado pelo resultado das simulações que mostra um número de interações menor nas simulações 26 e 28 para chegar ao mesmo target em relação às simulações 25 e 27 que não passaram pelo treinamento.

Por meio das simulações se demonstra a grande versatilidade do sistema desenvolvido. As influências das variáveis e configurações apresentadas mostraram as reações do robô móvel no desvio de obstáculos. Tendo este conhecimento sobre o efeito das variáveis e configurações é possível raciocinar em como melhorar os parâmetros para melhorar uma ou outra característica de comportamento do robô móvel para uma dada situação. O uso da simulação com visualização 3D facilita o entendimento do comportamento do robô móvel. A inteligência artificial se mostra como uma das grandes tecnologias para um sistema de controle para robótica móvel. O entendimento de como regular o sistema fuzzy assim como fazer uso do sistema neurofuzzy são importantes para explorar a potencialidade de programas como o desenvolvido neste trabalho.

Capítulo 5 – TRABALHOS FUTUROS

O desenvolvimento e aprimoramento de novos simuladores e novas técnicas são necessidades para os avanços nesta área da tecnologia. Algumas características e funcionalidades podem ser desenvolvidas tais como:

- Simuladores de robótica com membros articulados utilizando cinemática inversa;
- Cinemática inversa por meio de algoritmo de inteligência artificial;
- Modelos de “Rigid Bodies” para o robô interagir com o ambiente podendo deslocar objetos;
- Incorporar ao simulador modelos que possam se adequar a robôs reais como características de motorização, dimensões, dentre outros fatores;
- Incorporar Algoritmos genéticos ao treinamento de robótica para o aprendizado do controlador fuzzy;
- Simular a fusão de sensores, diferentes tipos de sensores trabalhando juntos assim como simular características de cada tipo de sensor;
- Simulação de diversos meios de posicionamento incorporando erros de odometria e posicionamento;
- Implementação de algoritmos *octrees* para redução de demanda computacional para ambientes gráficos 3D de maior complexidade;

Referências Bibliográficas

- [1] Mizumoto, M., Realization of PID Controls by Fuzzy Control Methods, *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego, pp. 709-715, 1992.
- [2] Fayad, C., Optimized Fuzzy Logic Based Algorithm for a Mobile Robot Collision Avoidance in an Unknown Environment, University of Nottingham, UK, 2000.
- [3] Kawamura, K., Application of Fuzzy Control to a Sonar-Based Obstacle Avoidance Mobile Robot, *Proceedings of 2000 IEEE International Conference on Control Applications*, Alaska, , pp. 425-430.
- [4] Webb, P., The Integration of an Optimized Fuzzy Logic Navigation Algorithm into a Semi-Autonomous Robot Control System, University of Nottingham, UK, 2000.
- [5] Shaw, I., Godoy, M., *Controle Modelagem Fuzzy*, Editora Edgard Blücher Ltda.
- [6] Mizrahi, V., *Treinamento em Linguagem C++ módulo 2*, MAKRON Books do Brasil Editora Ltda.
- [7] Schildt, H., *C Completo e Total*, MAKRON Books do Brasil Ltda, 1991.
- [8] Stroustrup, B., *The C++ Programming Language*, Addison-Wesley Publishing Company, 2^a ed., 1991.
- [9] Fullér, R., *Neural Fuzzy Systems*, Åbo Akademis tryckeri, Åbo, 1995.
- [10] Sertã, F., Thompson, C. PhD, *Controles Moderno e Fuzzy*, IPRJ-UERJ, Nova Friburgo.
- [11] Jung, C., *Computação Embarcada : Projeto e Implementação de Veículos Autônomos Inteligentes*, XXV Congresso da Sociedade Brasileira de Computação, UNISINOS, São Leopoldo.
- [12] Thompson, M., *Lógica Neuro-Fuzzy: Aplicação na Identificação do Perfil do Investidor*, Universidade Cândido Mendes.

APÊNDICE

A: Desenvolvimento Matemático da Lógica Neuro-Fuzzy

No capítulo sobre lógica Neurofuzzy as derivadas abaixo não foram desenvolvidas no capítulo 2.2 e o desenvolvimento baseia-se nas referências [9] e [12]

As correções dos parâmetros ‘a’ e ‘b’ da função de pertinência A_1 são:

$$\begin{aligned} a_1(t+1) &= a_1(t) - \eta \frac{\partial E_k}{\partial a_1}, \\ b_1(t+1) &= b_1(t) - \eta \frac{\partial E_k}{\partial b_1} \end{aligned}$$

Segue abaixo o desenvolvimento de suas derivadas

Para as correções do parâmetro ‘a’ da função de pertinência A_1 segue o desenvolvimento baixo:

sendo

$$erro = (o^k - y^k)$$

$$\frac{\partial E_k}{\partial a_1} = \frac{\partial E_k}{\partial erro} \frac{\partial erro}{\partial a_1} = \frac{\partial E_k}{\partial erro} \frac{\partial o_k}{\partial a_1} = erro \frac{\partial o_k}{\partial a_1}$$

$$\frac{\partial o_k}{\partial a_1} = \frac{\partial}{\partial a_1} \left[\frac{\sum_{i=1}^n \alpha_i z_i}{\sum_{i=1}^n \alpha_i} \right] = \left[\frac{\sum_{i=1}^n \alpha_i z_i \frac{\partial \sum_{i=1}^n \alpha_i}{\partial a_1} - \sum_{i=1}^n \alpha_i \frac{\partial \sum_{i=1}^n \alpha_i z_i}{\partial a_1}}{\left(\sum_{i=1}^n \alpha_i \right)^2} \right]$$

sendo

$$soma_fu = \sum_{i=1}^n \alpha_i z_i$$

sendo

$$soma = \sum_{i=1}^n \alpha_i$$

E sendo

$$A_1(x) = e^{-b_1(x-a_1)^2}$$

$$\frac{\partial \sum_{i=1}^n \alpha_i z_i}{\partial a_1} = z_1 e^{-b(x-a_1)^2} 2b(x-a_1) \cdot \frac{\alpha_1}{e^{-b(x-a_1)^2}} + \dots = nra_1 \times z_1 2b(x-a_1) \cdot (\alpha_1 + \dots)$$

Onde nra_1 é o número de regras que possuem a função de pertinência A_1 e α com o índice da respectiva regra que no caso é igual a 1 as regras que não possuem a função de pertinência A_1 a derivada é nula as derivadas das demais regras com a função A_1 diferem apenas no termo a_i que deve ser o da regra em questão.

$$\frac{\partial \sum_{i=1}^n \alpha_i}{\partial a_1} = e^{-b(x-a_1)^2} 2b(x-a_1) \cdot \frac{\alpha_1}{e^{-b(x-a_1)^2}} + \dots = nra_1 \times 2b(x-a_1) \cdot (\alpha_1 + \dots)$$

Onde nra_1 é o número de regras que possuem a função de pertinência A_1 e α com o índice da respectiva regra que no caso é igual a 1 as regras que não possuem a função de pertinência A_1 a derivada é nula as derivadas das demais regras com a função A_1 diferem apenas no termo a_i que deve ser o da regra em questão.

logo

$$a_1(t+1) = a_1(t) - \eta \frac{\partial E_k}{\partial a_1} = (o^k - y^k) \left[\frac{(soma_fu \times nra_1 \times 2b(x-a_1) \cdot (\alpha_1 + \dots)) - (z_1 \times soma \times nra_1 \times 2b(x-a_1) \cdot (\alpha_1 + \dots))}{soma^2} \right]$$

Para as correções do parâmetro ‘b’ da função de pertinência A_1 segue o desenvolvimento baixo:

$$\frac{\partial E_k}{\partial b_1} = \frac{\partial E_k}{\partial erro} \frac{\partial erro}{\partial b_1} = \frac{\partial E_k}{\partial erro} \frac{\partial o_k}{\partial b_1} = erro \frac{\partial o_k}{\partial b_1}$$

$$\frac{\partial o_k}{\partial b_1} = \frac{\partial}{\partial b_1} \left[\frac{\sum_{i=1}^n \alpha_i z_i}{\sum_{i=1}^n \alpha_i} \right] = \left[\frac{\sum_{i=1}^n \alpha_i z_i \frac{\partial \sum_{i=1}^n \alpha_i}{\partial b_1} - \sum_{i=1}^n \alpha_i \frac{\partial \sum_{i=1}^n \alpha_i z_i}{\partial b_1}}{\left(\sum_{i=1}^n \alpha_i \right)^2} \right]$$

sendo

$$soma_fu = \sum_{i=1}^n \alpha_i z_i$$

sendo

$$soma = \sum_{i=1}^n \alpha_i$$

E sendo

$$A_1(x) = e^{-b_1(x-a_1)^2}$$

$$\frac{\partial \sum_{i=1}^n \alpha_i z_i}{\partial b_1} = -z_1 e^{-b_1(x-a_1)^2} (x-a_1)^2 \cdot \frac{\alpha_1}{e^{-b_1(x-a_1)^2}} + \dots = -nra_1 \times z_1 (x-a_1)^2 \cdot (\alpha_1 + \dots)$$

Onde nra_1 é o número de regras que possuem a função de pertinência A_1 e α com o índice da respectiva regra que no caso é igual a 1 as regras que não possuem a função de pertinência A_1 a derivada é nula as derivadas das demais regras com a função A_1 diferem apenas no termo a_i que deve ser o da regra em questão.

$$\frac{\partial \sum_{i=1}^n \alpha_i}{\partial b_1} = -e^{-b(x-a_1)^2} (x-a_1)^2 \cdot \frac{\alpha_1}{e^{-b(x-a_1)^2}} + \dots = -nra_1 \times (x-a_1)^2 \cdot (\alpha_1 + \dots)$$

Onde nra_1 é o número de regras que possuem a função de pertinência A_1 e α com o índice da respectiva regra que no caso é igual a 1 as regras que não possuem a função de pertinência A_1 a derivada é nula as derivadas das demais regras com a função A_1 diferem apenas no termo α_i que deve ser o da regra em questão.

logo

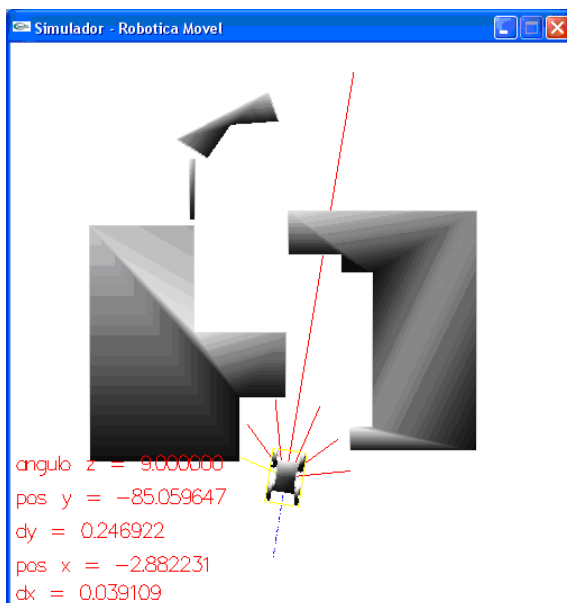
$$b_1(t+1) = b_1(t) - \eta \frac{\partial E_k}{\partial b_1} =$$

$$(o^k - y^k) \left[\frac{(z_1 \times soma \times nra_1 \times (x-a_1)^2 \cdot (\alpha_1 + \dots)) - (soma_fu \times nra_1 \times (x-a_1)^2 \cdot (\alpha_1 + \dots))}{soma^2} \right]$$

B: Detalhamento da Simulação de Robótica Móvel com Lógica Fuzzy

A simulação 16 é mostrada abaixo de forma a descrever como o robô móvel se comporta ao decorrer da simulação. Dessa forma se demonstra em diversas situações ao decorrer da simulação qual a reação do sistema de controle Fuzzy de desvio de obstáculos.

Simulação 16 na Iteração 185

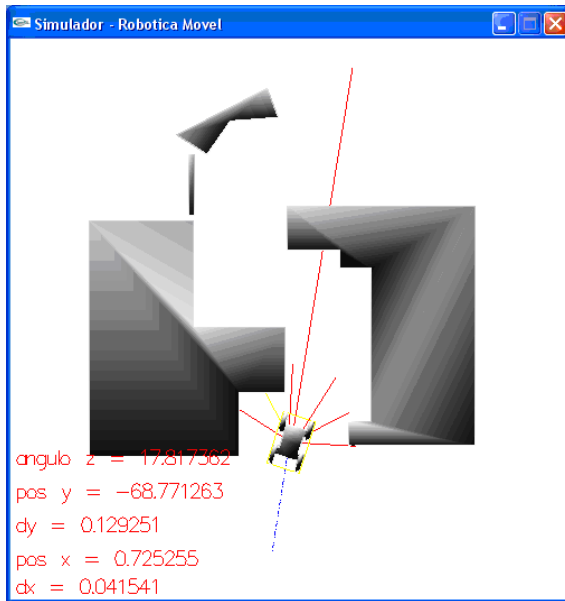


Sensor Ativado: Sensor LN
Distância de Entrada: 76,47%
Ângulo de Entrada: -75,42°
Vel. Linear de Saída: Não Altera
Vel. Angular de Saída: Não Altera
Distância do Target: 228,47
Posição X = -2,88 Y = -85,05
Ângulo em relação à origem: 9,0°
Velocidade Linear: 0,25

Figura B.1 – Simulação 16

Obstáculo à esquerda do robô é detectado e sua base de regras não indica modificação nas velocidades linear e angular.

Simulação 16 na Iteração 259

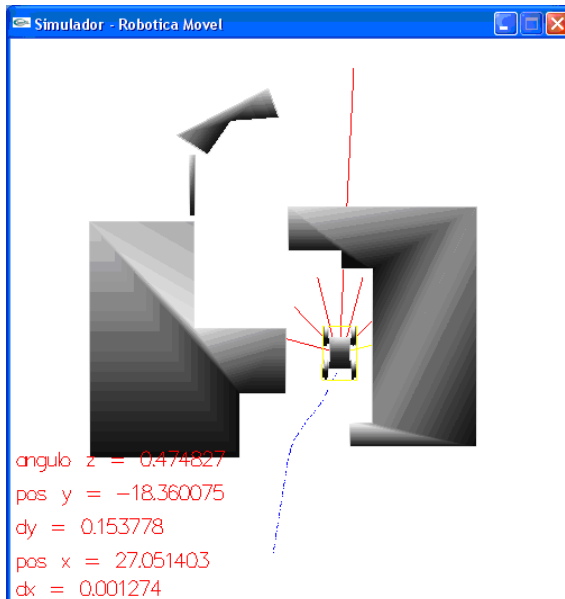


Sensor Ativado: Sensor MN
Distância de Entrada: 84,18%
Ângulo de Entrada: -45,00°
Vel. Linear de Saída: 54,30%
Vel. Angular de Saída: 42,85%
Distância do Target: 211,70
Posição X = 0,72 Y = -68,77
Ângulo em relação à origem: 17,81°
Velocidade Linear: 0,13

Figura B.2 – Simulação 16

Obstáculo à esquerda do robô é detectado e sua reação é uma rotação para a direita.

Simulação 16 na Iteração 575

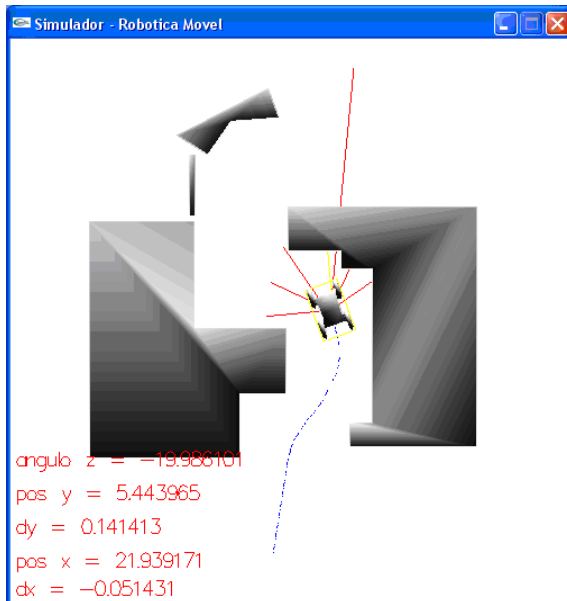


Sensor Ativado: Sensor LP
Distância de Entrada: 52,04%
Ângulo de Entrada: 75,42°
Vel. Linear de Saída: 61,51%
Vel. Angular de Saída: -14,28%
Distância do Target: 158,71
Posição X = 27,05 Y = -18,36
Ângulo em relação à origem: 0,47°
Velocidade Linear: 0,15

Figura B.3 – Simulação 16

Obstáculo à direita robô é detectado e sua reação é uma ligeira rotação para a esquerda.

Simulação 16 na Iteração 701

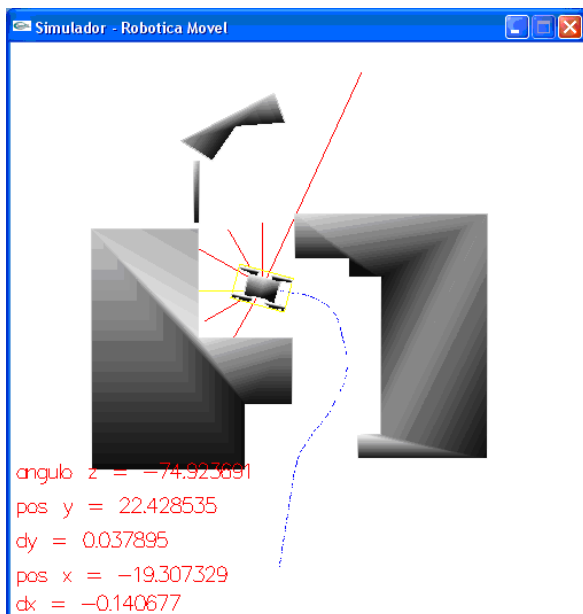


Sensor Ativado: Sensor ZP
Distância de Entrada: 68,73%
Ângulo de Entrada: 14,57°
Vel. Linear de Saída: 60,19%
Vel. Angular de Saída: -31,16%
Distância do Target: 135,32
Posição X = 21,93 Y = 5,44
Ângulo em relação à origem: -19,98°
Velocidade Linear: 0,15

Figura B.4 – Simulação 16

Obstáculo à direita e a frente do robô é detectado e sua reação é uma rotação para a esquerda.

Simulação 16 na Iteração 954

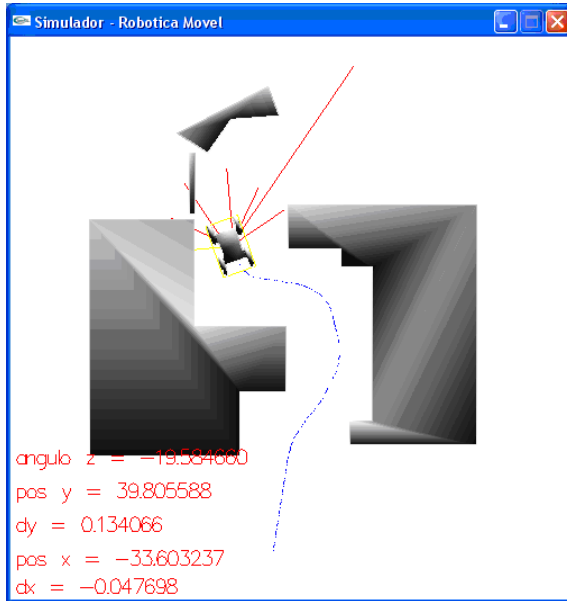


Sensor Ativado: Sensor ZN
Distância de Entrada: 74,40%
Ângulo de Entrada: -14,57°
Vel. Linear de Saída: 58,27%
Vel. Angular de Saída: 32,20%
Distância do Target: 129,50
Posição X = -19,30 Y = 22,42
Ângulo em relação à origem: -74,92°
Velocidade Linear: 0,14

Figura B.5 – Simulação 16

Obstáculo à esquerda e a frente do robô é detectado e sua reação é uma rotação para a direita.

Simulação 16 na Iteração 1093

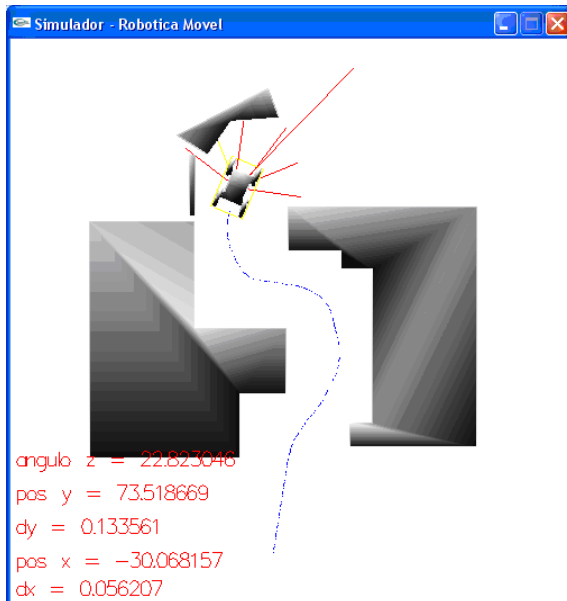


Sensor Ativado: Sensor LN
Distância de Entrada: 56,49%
Ângulo de Entrada: -75,42°
Vel. Linear de Saída: 56,91%
Vel. Angular de Saída: 14,27%
Distância do Target: 121,43
Posição X = -33,60 Y = 39,80
Ângulo em relação à origem: -19,58°
Velocidade Linear: 0,14

Figura B.6 – Simulação 16

Obstáculo à esquerda do robô é detectado e sua reação é uma rotação para a direita.

Simulação 16 na Iteração 1289

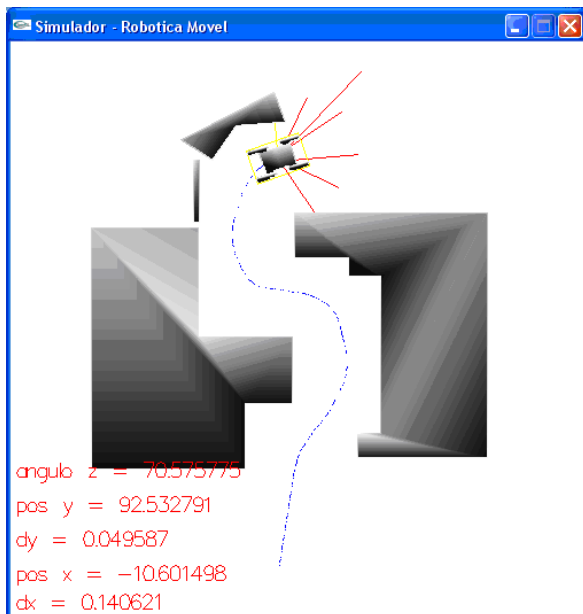


Sensor Ativado: Sensor MN
Distância de Entrada: 75,85%
Ângulo de Entrada: -45,00°
Vel. Linear de Saída: 57,96%
Vel. Angular de Saída: 42,84%
Distância do Target: 93,15
Posição X = -30,06 Y = 73,51
Ângulo em relação à origem: 22,82°
Velocidade Linear: 0,14

Figura B.7 – Simulação 16

Obstáculo à esquerda do robô é detectado e sua reação é uma rotação para a direita.

Simulação 16 na Iteração 1444

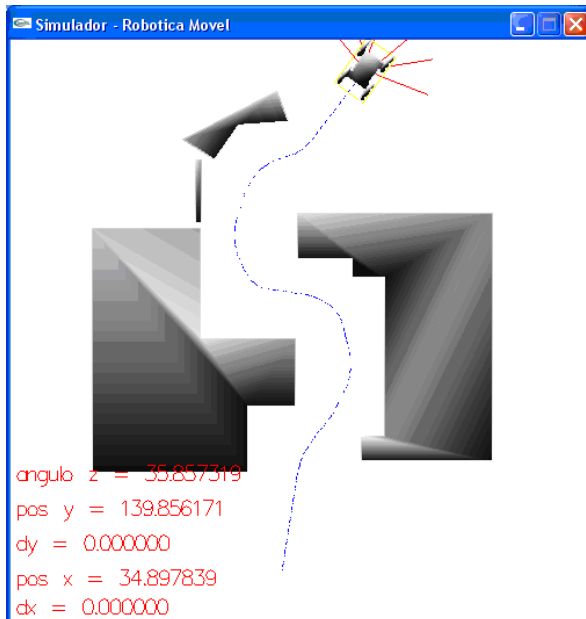


Sensor Ativado: Sensor LN
Distância de Entrada: 54,23%
Ângulo de Entrada: -75,42°
Vel. Linear de Saída: 59,64%
Vel. Angular de Saída: 14,27%
Distância do Target: 65,95
Posição X = -10,60 Y = 92,53
Ângulo em relação à origem: 70,57°
Velocidade Linear: 0,15

Figura B.8 – Simulação 16

Obstáculo à esquerda do robô é detectado e sua reação é uma ligeira rotação para a direita.

Simulação 16 na Iteração 1772



Sensor Ativado: Nenhum Sensor
Distância de Entrada: Não Ativado
Ângulo de Entrada: Não Ativado
Vel. Linear de Saída: Zero
Vel. Angular de Saída: Zero
Distância do Target: Target Atingido
Posição X = 34,89 Y = 139,85
Ângulo em relação à origem: 35,85°
Velocidade Linear: 0,00

Figura B.9 – Simulação 16

Objetivo atingido o robô móvel chega ao seu destino sem colisões em 1772 iterações.