



PONTIFÍCIA
UNIVERSIDADE
CATÓLICA
DO RIO DE JANEIRO

Robótica Móvel Inteligente

Aluno: Leonardo de Paula Batista Benevides
Orientador: Marley Maria B. R. Velasco

Introdução

A robótica é uma área de pesquisa interdisciplinar, por natureza. Pode-se afirmar, de forma geral, que ela emprega ferramentas, metodologias e tecnologias inerentes a grandes áreas como a engenharia mecânica, engenharia mecatrônica (com história também recente e poucos cursos no Brasil), engenharia elétrica e eletrônica e engenharia de computação. A robótica utiliza-se de conceitos teóricos de grandes áreas como matemática, física, química, biologia, até educação, e busca também muita inspiração em áreas mais centradas no entendimento do cérebro e do corpo humano como neurologia, fisiologia e psicologia.

Sob a ótica da robótica, em processos industriais e de produção, o desenvolvimento da área é de extrema importância devido ao alto risco de acidentes existente em grande parte das tarefas nestes processos. Falhas humanas não só podem trazer graves consequências ao trabalhador e ao equipamento industrial, mas também podem causar, como frequentemente tem ocorrido, sérios danos ambientais e ao Homem.

A execução deste projeto contribuirá com o avanço do conhecimento para o desenvolvimento de máquinas mais inteligentes com o objetivo de substituir o Homem em atividades que possam causar o prejuízo à saúde ou mesmo à própria vida deste. Isso inclui a possibilidade de robôs auxiliando o Homem de forma a reduzir as possibilidades de erros humanos.

Dessa forma, a motivação deste projeto é desenvolver pesquisa na área de robótica através de modelos de robôs virtuais e reais a partir de modelos computacionais inteligentes para dotar robôs reais com capacidade de navegação para atingir metas específicas.

Portanto, utilizando um robô Lego MindStorms (Figura 1), será apresentado um modelo de sistemas de controle Fuzzy capaz de realizar uma tarefa especificada mais adiante. Este modelo foi escolhido devido a complexidade do problema, onde o robô terá que lidar com diferentes entradas simultâneas em diferentes situações, caso fosse utilizado um sistema de controle convencional, haveria a necessidade de mapear todas as situações possíveis, o que seria muito trabalhoso e poderia não se levar em conta todas as possíveis variáveis relevantes ao problema, seja por falta de conhecimento de alguma delas ou por mera distração, o que acarretaria em um modelo ineficiente, gerando mais empecilhos na etapa de ajustes.



Figura 1 – Robô LEGO MindStorms NXT utilizado com modelo de controle Fuzzy.

Controladores que utilizam Lógica Fuzzy diferem dos controladores tradicionais, pois não priorizam a modelagem matemática, mas sim utilizam regras lingüísticas, que são muito mais intuitivas e próximas do raciocínio humano, o qual é dotado de capacidade de tomar decisões mesmo com informações imprecisas.

Para a criação dessas regras é necessário um especialista no assunto ou uma base de dados da qual se possa extrair informação para que se possa descrever como o robô deve reagir de acordo com as informações colidas do ambiente. Suas reações devem respeitar um conjunto de regras da forma “SE....x.....ENTÃOy....”.

Portanto, esse método permite que se resolva problemas complexos de uma forma bem intuitiva, ágil e prática e utilizando pouca modelagem matemática.

Objetivos

Desenvolver e embarcar um Sistema de controle Fuzzy em um robô Lego Mindstorms NXT com o objetivo de seguir um caminho fechado e cumprir um objetivo(capturar bolas de uma determinada cor) e retornar ao ponto de origem. Para isso será necessário desenvolver o conhecimento sobre: a tecnologia de sensores de som e opto-eletrônicos para navegação autônoma, em substituição ao reconhecimento de imagens; a capacitação no uso de ferramentas de simulação virtual de robôs reais; capacitação de robôs reais a seguir caminhos definidos por linha fechada a partir de informações fornecidas pelo ambiente.

Serão investigadas técnicas baseadas em inteligência computacional tais como Lógica Fuzzy e Sistemas Neuro-Fuzzy capazes de dotar o agente-robô com raciocínio, para que este possa de forma autônoma realizar a tarefa proposta da forma mais eficiente possível.

Hardware Utilizado



Figura 2 – Brick LEGO MindStorms NXT

Resultado de uma parceria entre o MIT e o LEGO Group, o LEGO MindStorms é a linha que associa as peças de montar à evolução da tecnologia. Com blocos de encaixe, o brinquedo evoluiu passando pela linha *Technic*, cujos encaixes mais inteligentes permitiam movimentos de articulações, eixos e engrenagens, posteriormente sendo associados a motores, para finalmente permitir a participação de sensores e microcontroladores.

A primeira versão comercializada do MindStorms denominada “*Robotics Invention Systems*” (RIS) era controlada pelo *brick* RCX provido de um microcontrolador de 8 bits e 32k de memória RAM para armazenar o firmware e o programa do usuário que aceitava diversas linguagens de programação e era realizada por uma interface infravermelha entre o computador e brick.

Este produto passou a ser adotado em escolas e universidades com fins didáticos, pois todos os aspectos da dinâmica robótica eram aplicados, de forma simplificada, desafiando a construir projetos envolvendo estruturação, mecânica, programação e a capacidade de correlacionar todos estes formando o raciocínio mecatrônico.

Em 2006 os kits LEGO MindStorms NXT (Figura 3) evoluíram em relação ao seu antecessor, o *brick* passou a contar com dois microprocessadores, um principal de 32 bits, com 48MHz, 256k de memória flash e 64k de memória RAM, e outro secundário de 8 bits, 4MHz, 4k de memória flash e 512b de RAM, um adaptador *Bluetooth* interno e uma entrada USB. Esta nova versão é a disponibilização em código aberto do firmware no próprio site da empresa.

O uso no meio acadêmico possibilitou a criação de competições de robôs, onde instituições de ensino apresentam robôs desenvolvidos por seus alunos para competirem entre si. As modalidades presentes nas competições consistem desde uma simples tarefa que os participantes montem um robô que realize uma tarefa até as mais complexas, como que as instituições tragam times de robôs que joguem uma partida de futebol.

O kit (Figura 5) conta com o *brick* (1), este com entrada para três servo-motores e quatro sensores, os três servo-motores (6), um sensor ultra-som (5), um sensor infra-vermelho (4 - utilizado para percepção luminosa de ambientes claros e escuros), um sonoro (3 - que pode ser ajustado na escala decibel (dB) para detectar todos os sons com a mesma sensibilidade) e dois de toque (2- capazes de perceberem quando são pressionado por algum objeto, e também quando é liberado novamente).



Figura 3 – Kit LEGO MindStorms NXT

Os servo-motores podem funcionar como entradas e/ou saídas de dados. Eles funcionam como um motor de passos, sendo possível controlar o quanto o motor deve girar, e com qual velocidade ele deve fazê-lo em uma escala de 0 a 100. Cada motor possui também um sensor de rotações embutido, que permite a leitura dos movimentos do motor, podendo esta ser apresentada em graus, com precisão de +/- um grau e total para uma volta de 360 graus, ou em voltas completas, aplicada à medição de distâncias percorridas.

O sensor ultra-sônico, juntamente com o luminoso, compõe a visão do robô. Funcionando com o mesmo princípio científico de localização dos morcegos, o sensor percebe obstáculos a até 255 centímetros com uma precisão de +/- 3 cm calculando o tempo de retorno de uma onda sonora refletida pelo objeto. Ele pode apresentar o resultado em centímetros ou em polegadas .

A LEGO disponibiliza também sensores de movimentos como o *Gyro*(percebe a velocidade de rotação), *Compass*(percebe a direção de rotação) e um sensor acelerômetro, que percebe aceleração nos três eixos ortogonais.

Arquitetura do Robô

Utilizou-se como base o modelo TriBot (Figura 4), sugerido pelo manual da Lego, , um triciclo composto por dois motores ligados a rodas e uma terceira roda de apoio livre.



Figura 4 – Estrutura sugerida no manual.

Este modelo foi escolhido como ponto de partida, pois possibilita, com um pequeno número de peças e apenas dois motores, o controle sobre a direção e o sentido de navegação do robô. O fato de utilizar apenas dois motores é de suma importância, pois teremos uma porta sobrando no brick para um servo que capturará as bolas.

Para ser capaz de percorrer um percurso fechado de qualquer tipo, o robô foi equipado com dois sensores Luminosidade na sua dianteira (Figura 6), voltados para baixo, pois a análise das leituras realizadas por estes sensores dão ao robô a visão ou percepção da variação da cor do caminho que ele deverá seguir. Os sensores devem estar o mais apontado para baixo possível, pois a leitura será mais precisa devido a menor interferência da reflexão dos raios luminosos.

Ao perceber uma variação na cor da pista(robô andarรก sobre uma linha preta em um ambiente de fundo branco), o robô toma a iniciativa de acertar seu percurso a fim de que seus sensores voltem a indicar que estآo lendo branco. Portanto, existe a obrigatoriedade de se utilizar dois sensores, um a direita e outro a esquerda do centro do robô para que ele possa decidir para qual lado deverรก girar se necessitar fazer uma curva. A pista de testes   composta por um fundo branco e o caminho em preto com bordas cinzas, esse modelo deve-se ao fato de que a L gica Fuzzy, diferentemente da L gica Tradicional, aceita que um elemento pertence com um grau de pertin ncia a um conjunto, ou seja a cor cinza ela pertence com um certo grau(podemos dizer que 50%) ao conjunto Preto , e ao mesmo tempo pertence com um grau(os restantes 50%) ao conjunto Branco.

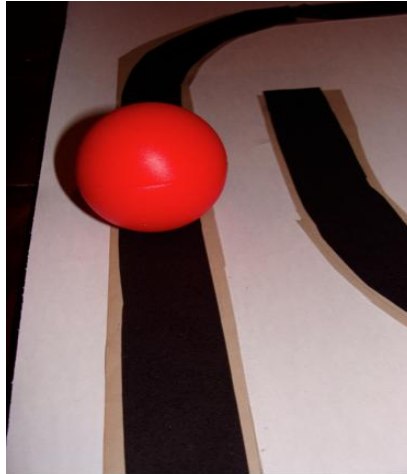


Figura 5 - Modelo de pista utilizada.

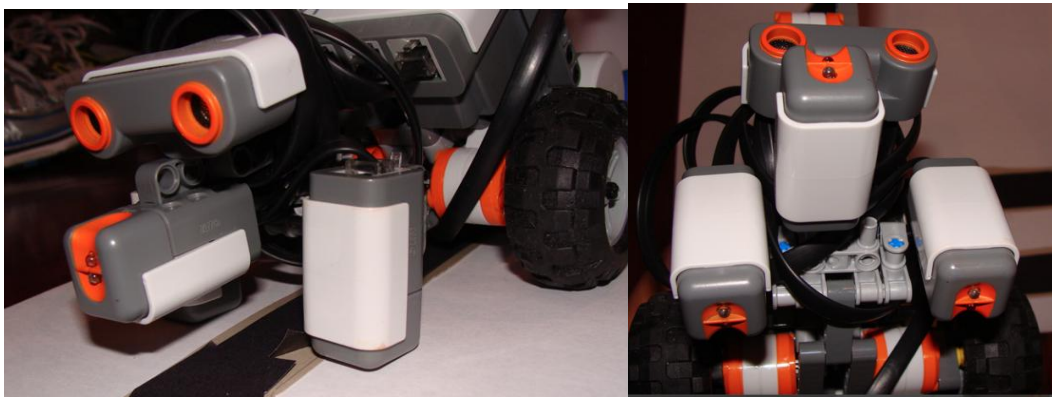


Figura 6– Disposição dos sensores infra-vermelho.

Quanto à tarefa de capturar bolas, o robô contará com dois sensores, um infra-vermelho e outro ultra-sônico, localizados na parte central de sua dianteira (Figura 7) – voltado para frente do veículo – que, em uma rotina serão responsáveis por verificar se existe algum elemento na frente do robô (se a distância a algum objeto é inferior ao tamanho que a garra pode capturar) e se o elemento é da cor que se deseja.



Figura 7 – Disposição dos sensores responsáveis pela rotina de captura da bola.

O terceiro servo-motor é o responsável por capturar o elemento identificado pelos dois últimos sensores, para isso ele foi fixo na parte superior do robô para que possa sustentar uma garra e movimentá-la quando acionado (figura 8).

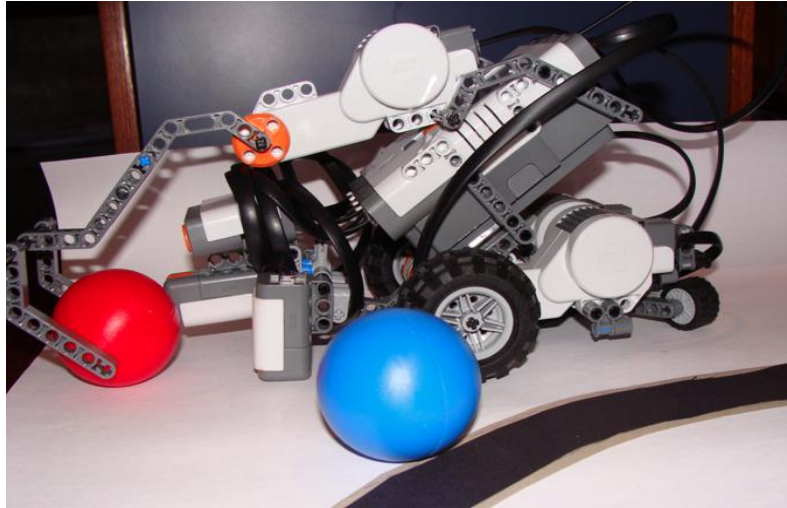


Figura 8 – Ilustração do funcionamento do Servo e da Garra .

Software

Idealmente, utiliza-se plataformas de simulação, onde o problema é modelado e as soluções propostas são avaliadas virtualmente para depois serem embarcadas no hardware e necessitar poucos ajustes. Apesar disso, neste trabalho não foi possível realizar esse procedimento, que seria uma integração com outro trabalho de iniciação científica de outro aluno, entretanto, os resultados obtidos foram muito bons e não foram necessários muitos ajustes.

1.1.LEGO MINDSTORMS Education's NXT Software v1.1

Este é o software que acompanha o Robô(Figura 9). É desenvolvido pela *National Instruments* (NI), o programa é baseado no software LabVIEW, também da mesma empresa. A linguagem de programação bastante intuitiva “*drag-and-drop*” (blocos que representam elementos são “arrastados para uma posição”).

O kit LEGO MINDSTORMS possui um enorme potencial computacional e é vendido também como brinquedo para crianças por ser muito simples, entretanto essa simplicidade traz limitações, apenas funções básicas de programação,portanto ele não é capaz de executar as rotinas desejadas.

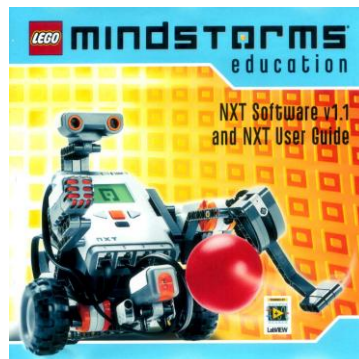


Figura 9 – Software LEGO NXT.

1.2.National Instruments LabVIEW 8.2

O LabVIEW utiliza-se de um ambiente gráfico de caráter extremamente intuitivo e fácil de operar (não é necessário conhecimentos das tradicionais linguagens textual de programação), baseado em ícones de funções, rotinas, variáveis e controles .

O programa é utilizado em larga escala pala indústria, por sua alta performance em aquisição, análise e manipulação de dados em todas as etapas do processo de produção.

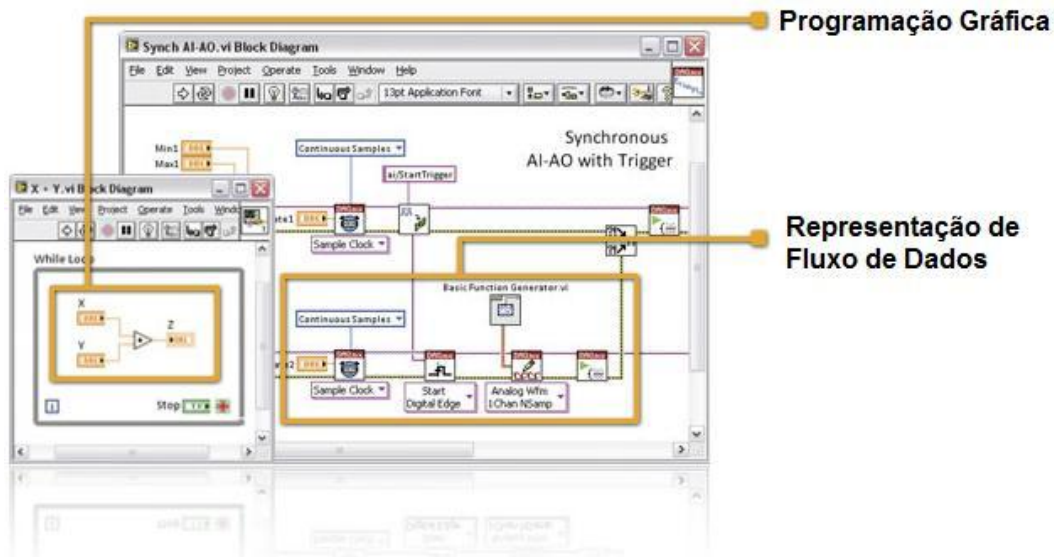


Figura 10 – LabVIEW.

A *National Instruments*, desenvolveu um toolkit para LabVIEW das funções de controle do NXT, o qual dispõe-se de todo o poder computacional do LabVIEW para realizarmos as rotinas mais elaboradas que não eram permitidas pelo programa fornecido junto com o robô.

Todo este poder computacional, porém, exige algumas limitações. O *Brick* do LEGO NXT não reconhece a integração com o software MATLAB nem arredondamentos numéricos e conversões entre variáveis. Por este motivo pode-se apenas gravar programas pouco complexos na memória do *Brick* para execução no mesmo. Assim, fica-se então limitado a uma conexão *online* entre o *Brick* e o computador, seja por *Bluetooth* ou via cabo USB, onde todo o trabalho computacional é realizado no micro, que o *Brick* apenas retransmitirá os comandos para os servos e dos sensores.

A escolha deste software para elaboração do algoritmo de controle para este projeto foi devido a sua interface ágil e eficiente e o fato de ser a base do programa desenvolvido em parceria com a LEGO para controlar o NXT.

O LabVIEW possui uma interface de comunicação com o MATLAB(apresentado a seguir) , portanto será construído o controlador Fuzzy no MATLAB para posteriormente executá-lo através do software da *National Instruments* através da janela "*MATLAB Script Node*"(Figura 11) que executa as funções perfeitamente sem acréscimo significativo de necessidade computacional nem de tempo de execução.

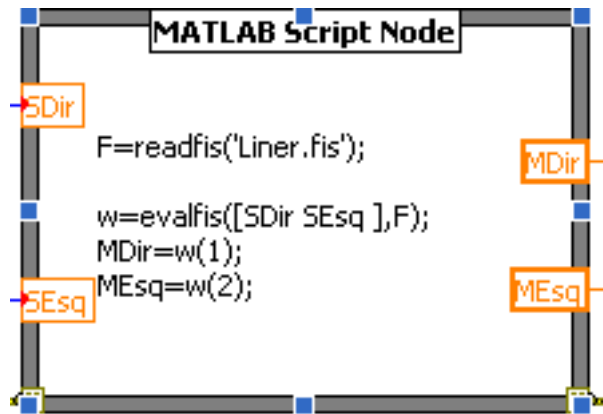


Figura 11 – MATLAB Script Node com algoritmo de chamada do controle Fuzzy.

1.3.MATLAB

O MATLAB, abreviação de “*Matrix Laboratory*”, juntamente com o Simulink, seu principal “*addon*” são amplamente utilizados tanto para fins didáticos, em universidades, quanto para fins comerciais em empresas de pesquisa, de novas tecnologias e instituições financeiras.

As áreas de aplicação destas ferramentas englobam praticamente todos os ramos que envolvem análises matemáticas. Alguns exemplos são controle, com modelagem para sistemas de controle e produção de código para sistemas externos; processamento de sinais, com modelagem para processamento de sinais, contando também com ambiente de simulação e produção e verificação de código; computação técnica, com desenvolvimento, visualização e análises de algoritmos matemáticos; processamento de imagens, com rotinas de aquisição, análise, visualização e ambiente de desenvolvimento de algoritmo; finanças, com modelos financeiros, ambientes de análises e aplicações.

No MATLAB, utilizou-se o “*Fuzzy Logic Toolbox*” (Figura 12) para desenvolvimento do controlador que será apresentado mais adiante no trabalho.

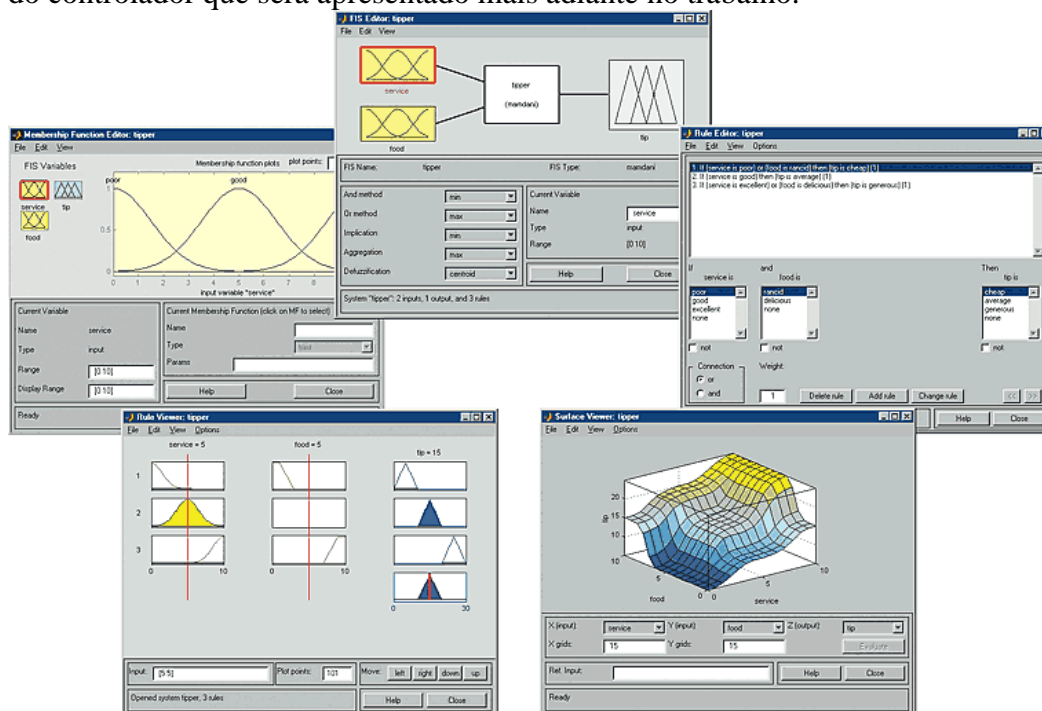


Figura 12 – Fuzzy Logic Toolbox.

Controle

Para atingir o objetivo do controle proposto existe uma rotina principal, que mantém o robô sobre a linha preta e testa se a bola da cor procurada está perto e se ela deve ser capturada. A rotina será executada sucessivamente até que se dê a tarefa por cumprida.

O Controle principal, aplicando Fuzzy é o da rotina de locomoção inteligente, enquanto o teste de localização do objetivo é um algoritmo de identificação. O programa em LabVIEW será apresentado no apêndice deste trabalho.

Rotina de Captura da Bola

Enquanto o robô estiver se movimentando pela linha preta encontrará bolas pelo caminho e ao se aproximar de objetos, o sensor de ultra-som acoplado na frente informará um valor (distância da bola ao robô), quando este valor for menor que quinze(valor empírico), ele fará um outro teste para verificar a cor da bola.

Este teste consiste em sabendo que há um objeto a sua frente e sua garra esta vazia, utilizar o sensor infra-vermelho para verificar se a cor do objeto é a cor procurada. Cada cor tem uma intensidade, a cor azul tem intensidade 37, e a cor vermelha em torno de 42. Se a intensidade da bola vista pelo sensor for maior ou igual a 40, a garra será acionada, caso a contrário o robô irá ignorar a bola e continuar o percurso.

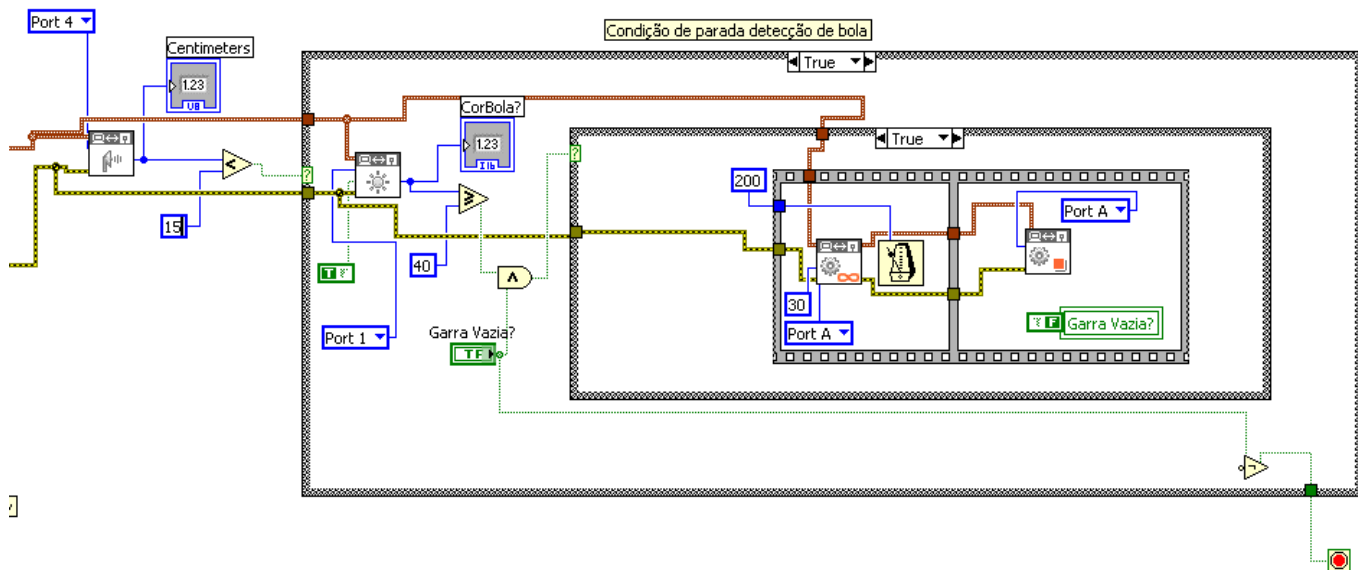


Figura 13 – Rotina de Captura.

Rotina de Navegação Autônoma

Foi implementado um sistema Fuzzy (Figura 14) capaz de julgar as condições do ambiente ao seu redor, e tomar decisões da forma de atuação nos motores.

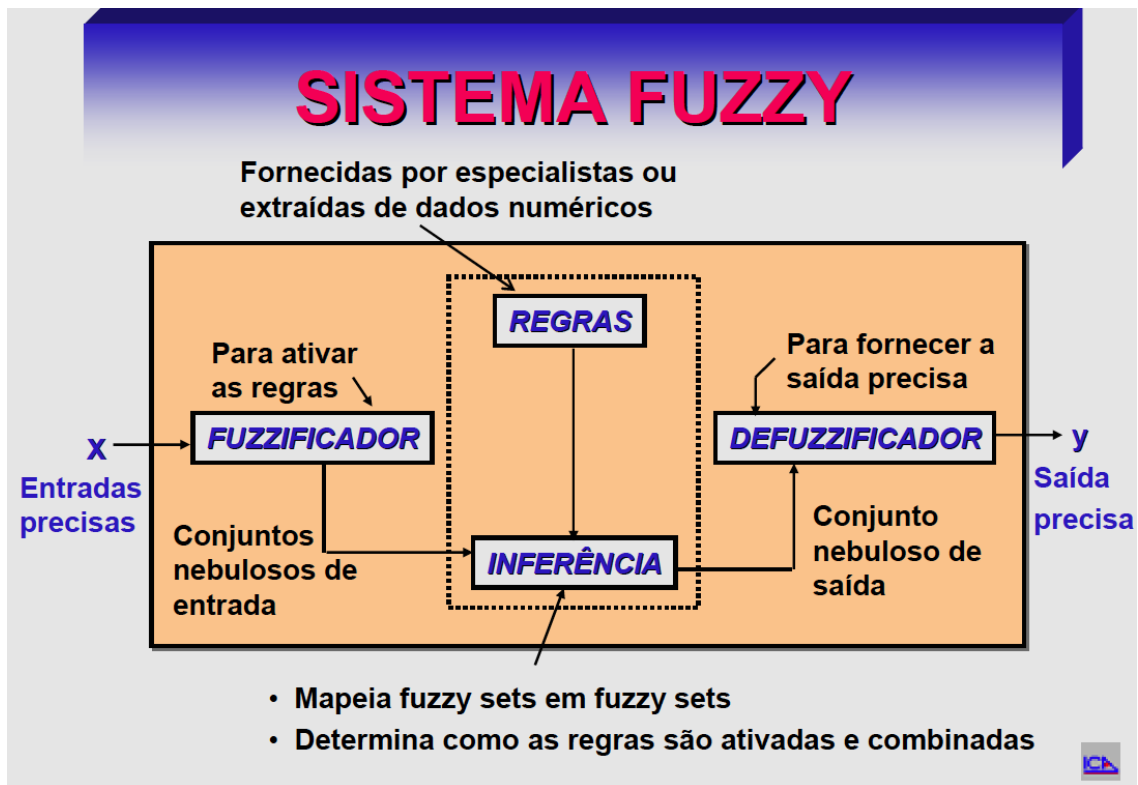


Figura 14 – Estrutura do sistema.

A escolha de um controlador baseado em Lógica Fuzzy, foi devido a necessidade de se inferir, de forma paralela, todas as entradas para então atuar nas saídas. Outro motivo da sua escolha foram a suavidade das respostas nos motores, pois como o controle é constante e não booleano, não há trancos ou movimentos súbitos por parte do robô como foram notados nos testes utilizando um controlador “Crisp”(Ordinário).

Esta rotina se resume em locomover o NXT sobre a linha preta, os sensores infravermelhos devem ser mantidos sobre uma superfície branca caso verifiquem uma mudança na leitura (comecem a se aproximar da linha preta), deverá obedecer uma das regras para consertar sua trajetória. Numa segunda fase de testes foram introduzidas linhas cinzas entre a linha preta e o fundo branco, afim de tornar mais natural ainda o movimento do robô, pois seguindo um dos preceitos base do modelo Fuzzy, o cinza não é nem branco nem preto, logo a ação do robô deve avaliar o nível de pertinência das variáveis e fornecer a melhor resposta.

As crenças embarcadas no NXT são os valores dos sensores ultra-sônicos. As suas atitudes serão baseadas na base de regras que compõem a inteligência embarcada (lógica Fuzzy).

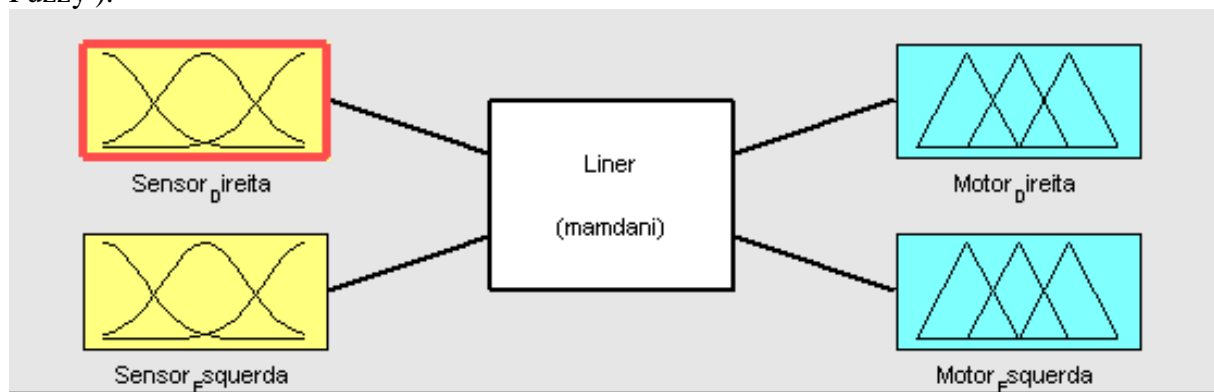


Figura 15 – Sistema Fuzzy Utilizado.

Variáveis Lingüísticas

Existem duas variáveis lingüísticas como entrada do sistema de controle Fuzzy, que são referentes aos valores obtidos pelos sensores de infra-vermelho localizadas a direita e a esquerda do robô.

Para as saídas do sistema Fuzzy de controle, existem duas variáveis lingüísticas, referentes à potência aplicada aos motores direito e esquerdo.

As leituras dos sensores de infra-vermelho são limitadas na faixa entre 0 e 100 pelo próprio hardware.

Quanto à saída dos motores como agentes de atuação do sistema, eles possuem uma limitação imposta também pelo hardware de -100 a 100, sendo estes as velocidades máximas de ré e para frente respectivamente.

Conjuntos Fuzzy

As variáveis lingüísticas dos sensores de infra-vermelho, tanto o esquerdo quanto o direito foram divididas em dois conjuntos fuzzificadores (Branco e Preto), cujo arranjo é apresentado abaixo (Figura 16).

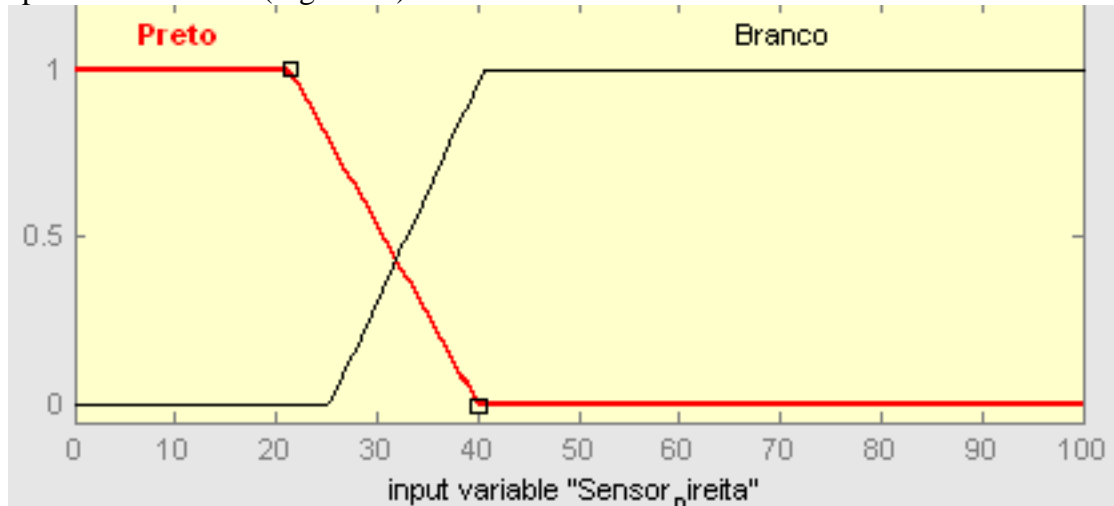


Figura 16 – Conjuntos Fuzzy de entrada para os sensores infra-vermelho.

Como se pode perceber, os conjuntos ativos para fuzificação realmente operam entre 15 e 70 . Após este valor, somente o conjunto Branco estará ativo com grau de pertinência igual a 1 e antes de 15 o conjunto Preto que estará ativo com grau de pertinência igual a 1. Este conjunto poderia ter sido reduzido para apenas essa faixa de valores, mas se decidiu manter os limites do hardware.

Para as variáveis dos servo-motores direito e esquerdo foram construídos os conjuntos Fuzzy –Frente, Zero e Ré, conforme demonstrado abaixo (Figura 17). Neste caso se optou por restringir os limites de atuação do motor porque não se queria que fossem mostrados casos onde a velocidade de Ré fosse maior que 40.

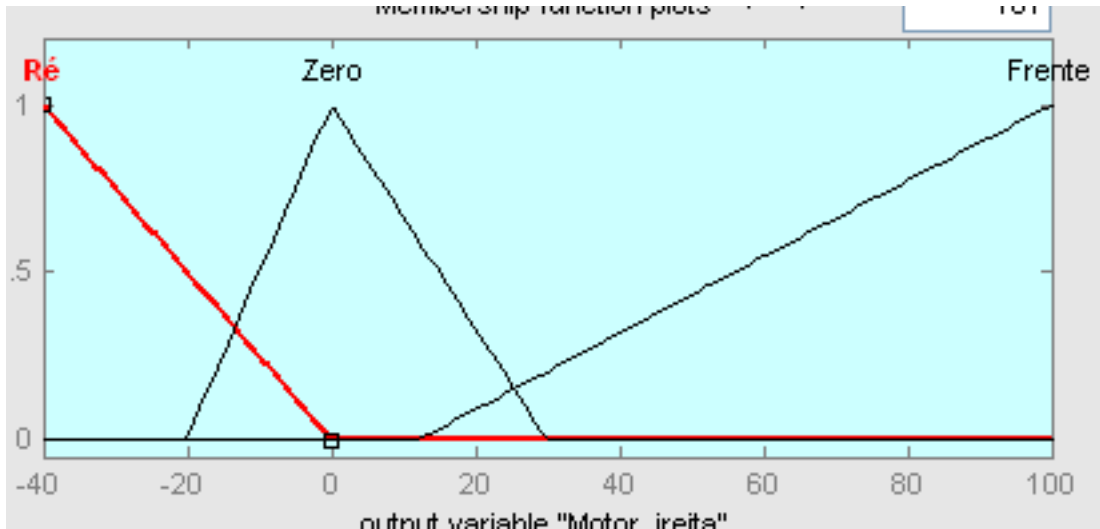


Figura 17 – Conjuntos Fuzzy de saída para os motores.

1.3.1. Base de Regras

O sistema Fuzzy proposto, com duas entradas (provenientes dos sensores de infravermelho para navegação) para duas saídas (atuação em ambas as rodas), poderia ser controlado por até 4 regras.

As regras foram formuladas seguindo uma ordem lógica de raciocínio, como era previsto de ser em um controle baseado em sistema Fuzzy.

As regras utilizadas foram:

- Se o Sensor da Direita é Preto e o Sensor da Esquerda é Preto, então a saída no Motor da Direita é Ré e a saída no Motor da Esquerda é Frente.
- Se o Sensor da Direita é Branco e o Sensor da Esquerda é Branco, então a saída no Motor da Direita é Frente e a saída no Motor da Esquerda é Frente.
- Se o Sensor da Direita é Preto e o Sensor da Esquerda é Branco, então a saída no Motor da Direita é Ré e a saída no Motor da Esquerda é Frente.
- Se o Sensor da Direita é Branco e o Sensor da Esquerda é Preto, então a saída no Motor da Direita é Frente e a saída no Motor da Esquerda é Ré.

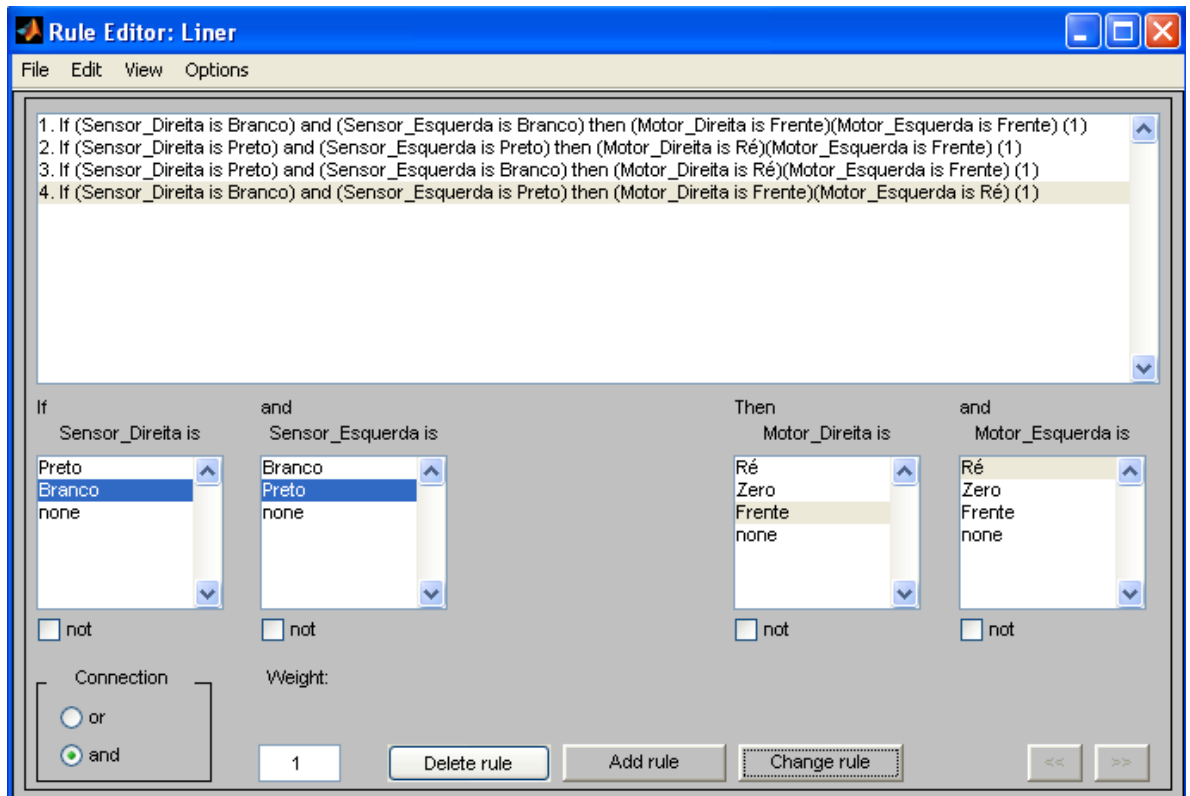


Figura 18 – Tela do editor de regras.

	Inputs		Outputs	
	SenDir	SenEsq	MotDir	MotEsq
1	P	P	R	R
2	P	B	F	R
3	B	P	R	F
4	B	B	F	F

Tabela 1 – Tabela de regras para o Controlador Fuzzy (P – Preto, B – Branco, F- Frente, R- Ré, Z-zero).

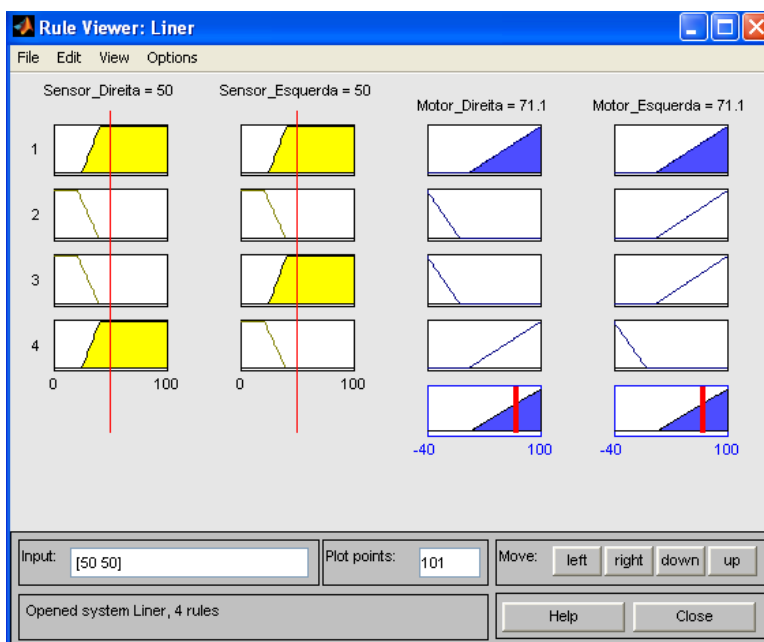


Figura 19 – Tela do visualizador de regras.

Defuzzificação

O método de inferência utilizado foi o Max-Min, onde o operador mínimo é utilizado nas implicações das entradas para cada regra, e então as regras serão compostas pela atuação do operador máximo na pertinência resultante de cada regra (no conectivo “e” das regras lingüísticas é utilizado operador mínimo e no conectivo “ou” das regras lingüísticas é utilizado operador máximo).

O Método de defuzzificação utilizado foi o método da Centróide, onde o valor de saída final do sistema é dado pela posição do centróide do gráfico formado pelos conjuntos Fuzzy de saída formados pela inferência das entradas.

Conclusões

Estudar os conceitos de inteligência computacional foi importante para produzir agentes robóticos inteligentes que podem realizar tarefas independente de uma supervisão humana e que possam substituir estes em situações de risco. Este projeto abranje os fundamentos básicos de um mecanismo de navegação autônoma em ambientes que se tem um padrão de caminho, a partir deste, pode-se adaptar para situações reais que possuam uma base commercial, como seria o caso de automóveis de navegação autônoma, os quais possuiriam um GPS onde se poderia escolher um destino, e sem mais ações do passageiro o veículo chegaria ao destino com o menor tempo possível, com uma velocidade moderada. Outras aplicações que já começaram a surgir seriam na automatização das tarefas domésticas, com aspiradores de pó que limpam a casa sozinhos, e futuramente podemos prever ferros de passar autônomos, limpadores de janelas e desentupidores.

Além das aplicações comerciais poderíamos ter robôs que possam procurar sobreviventes e desaparecidos em terremotos, desmoronamentos, quedas de aviões e naufrágios.

Portanto, estamos encarando um problema que pode ser a solução de problemas de muitas áreas diferentes, praticamente qualquer produto que envolva independência humana poderia ser resolvido utilizando esta base.

Comparando os resultados produzidos, os que utilizavam essas técnicas na maioria das vezes apresentavam respostas mais precisas e suaves do que controles que implementavam suas rotinas baseadas em métodos convencionais.

A próxima etapa do projeto consiste em estender o uso de agentes inteligentes para que estes possam trabalhar em conjunto e o aprendizado individual possa ser compartilhado com o grupo de agentes, a fim de que estes realizem melhor a tarefa que lhes foi proposta.

Referências

- 1 - George J. Klir & Bo Yuan, Fuzzy Sets and Fuzzy Logic – Theory and Applications, Prentice Hall PTR, 1995.
- 2 - J. M. Mendel, Fuzzy Logic Systems for Engineering: a Tutorial, Proc. IEEE, V. 83, No. 3, pp. 345-377, 1995.
- 3 - Haykin, S. Redes neurais, princípios e prática. 2 edição. Porto Alegre: Bookman, 2001.

4 - Wasserman, P.D. Neural computing: theory and practice. New York: van Nostrand Reinhold. 1989

5 – www.ica.ele.puc-rio.br/Downloads%5C31/ICA-cursop7-Defuzzificação.pdf - visitado em 28 de Julho de 2010.

6 – www.mathworks.com - visitado em 28 de Julho de 2010.

Apêndice

