UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

ALEXANDRE JACQUES MARIN
JÚLIO CESAR NARDELLI BORGES
YURI ANTIN WERGRZN

ANÁLISE QUALITATIVA DE ALGORITMOS DE NAVEGAÇÃO FUZZY

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2011

ALEXANDRE JACQUES MARIN
JÚLIO CESAR NARDELLI BORGES
YURI ANTIN WERGRZN

ANÁLISE QUALITATIVA DE ALGORITMOS DE NAVEGAÇÃO FUZZY

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Informática como requisito parcial para obtenção do grau de Engenheiro no Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: João Alberto Fabro

Co-orientador: Heitor Silvério Lopes

CURITIBA

2011

Texto da dedicatória.

AGRADECIMENTOS

Texto dos agradecimentos.

Texto da epígrafe.

RESUMO

MARIN, Alexandre; BORGES, Júlio; WERGRZN, Yuri . Análise Qualitativa de Algoritmos de Navegação Fuzzy. 26 f. Trabalho de Conclusão de Curso – Bacharelado em Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

Este documento descreve detalhadamente a execução do projeto Änálise Qualitativa de Algoritmos de Navegação Fuzzy, feito como trabalho de conclusão de curso de Engenharia de Computação na Universidade Tecnológica Federal do Paraná.

Palavras-chave: Navegação, Fuzzy, Robôs, Autônoma, ...

ABSTRACT

MARIN, Alexandre; BORGES, Júlio; WERGRZN, Yuri . Qualitative Analysis of Fuzzy Algorithms. 26 f. Trabalho de Conclusão de Curso – Bacharelado em Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

Abstract text (maximum of 500 words).

Keywords: Navigation, Fuzzy, Robots, Autonomous, ...

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

LISTA DE SÍMBOLOS

SUMÁRIO

1 INTRODUÇÃO

O problema do controle de navegação de robôs móveis autônomos é um campo da Engenharia da Computação que representa um grande desafio, devido ao fato de o ambiente ser dinâmico, haver sensoriamento sujeito a ruídos e exigências de controle e tomada de decisão em tempo real. Um sistema de navegação deve garantir que o robô móvel atinja satisfatoriamente o destino de sua trajetória, enviando ao robô os comandos necessários para a sua locomoção, de maneira precisa e suave, ao mesmo tempo em que permite reações rápidas às mudanças de ambiente para evitar colisões. Na robótica móvel, existem dois principais paradigmas que guiam os projetos de diversas arquiteturas de sistemas de navegação: o reativo e o deliberativo. O paradigma reativo procura reproduzir a reação imediata dos animais aos estímulos do ambiente. Geralmente, arquiteturas reativas são empregadas como uma camada de nível inferior na navegação de robôs móveis, pois apresentam a vantagem de resposta em tempo real uma vez que mapeiam a leitura dos sensores, diretamente, em ações. Arquiteturas deliberativas, por outro lado, intercalam o processo da tomada de decisão, desde a percepção até a ação, com uma etapa de planejamento a qual demanda grande tempo computacional, impedindo a atuação do robô em tempo real. Atualmente, são definidas arquiteturas híbridas, conjugando ambos os paradigmas(FRACASSO PAULO T.; COSTA, 2005). Ao realizar uma breve pesquisa para análise do estado da arte, foi possível perceber que existem vários trabalhos que apresentam novos métodos para navegação autônoma através do uso de lógica Fuzzy. Entretanto, não foi encontrado um trabalho propondo a comparação entre métodos já existentes. Com o intuito de preencher esta lacuna, a equipe optou desenvolver este projeto. Um dos modelos analisados neste trabalho é o modelo ED-FCM proposto por Mendonça no ano de 2010(MENDONÇA M.; ARRUDA, 2010). A sigla ED-FCM significa Event- Driven Fuzzy Cognitive Maps e quer dizer Mapas Cognitivos Difusos Dirigidos A Eventos. Para realizar a comparação prática destes algoritmos, provou-se necessária uma plataforma robótica, obtida através reconstrução do robô Bellator. O Bellator é um robô móvel que estava em construção em outro projeto mas foi abandonado. Este robô, sua reconstrução e adaptação às necessidades da equipe, é parte do projeto e é documentada em detalhes neste documento. Esta adaptação leva em conta também a possibilidade de utilização do Bellator para projetos futuros. As principais motivações deste tra-

balho são o desenvolvimento de uma plataforma robótica adequada para pesquisas acadêmicas, tendo em vista a falta de plataformas disponíveis com este propósito, e o estudo de sistemas de controle de navegação robótica, que é um ramo de pesquisa de elevado valor acadêmico. A possibilidade de disponibilizar uma nova plataforma robótica para futuros estudos acadêmicos e estudar um algoritmo de navegação pouco explorado são os maiores incentivos da equipe para a realização deste trabalho.

1.1 MOTIVAÇÃO

Uma das principais vantagens do uso do estilo de formatação `abnt-UTFPR.cls` para LaTeX é a formatação *automática* dos elementos que compõem um documento acadêmico, tais como capa, folha de rosto, dedicatória, agradecimentos, epígrafe, resumo, abstract, listas de figuras, tabelas, siglas e símbolos, sumário, capítulos, referências, etc. Outras grandes vantagens do uso do LaTeX para formatação de documentos acadêmicos dizem respeito à facilidade de gerenciamento de referências cruzadas e bibliográficas, além da formatação – inclusive de equações matemáticas – correta e esteticamente perfeita.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Os principais objetivos do projeto são construir uma plataforma robótica estável cujo hardware e software deverão ser bem documentados, visando extensibilidade para projetos futuros, e utilizar-se desta plataforma para testar dois algoritmos de navegação fuzzy, comparando-os segundo diversos critérios, tais como: tamanho e complexidade do código, custo computacional, tempo de resposta, eficácia.

1.2.2 Objetivos Específicos

Os objetivos específicos deste projeto remetem não apenas à análise de algoritmos de navegação fuzzy mas também quanto à qualidade da re-construção da plataforma robótica. De maneira geral, a plataforma deve ser capaz de detectar obstáculos próximos a mesma através de sensores de distância, e deverá se deslocar por meio de um par de motores de corrente contínua acoplados a rodas revestidas de borracha. As rodas devem ser equipadas com encoders para auxílio à odometria. A plataforma deve executar comandos de movimentação, os quais são: translação para frente e para trás, e rotação, no sentido horário e anti horário. Um sistema mi-

crocontrolador deve processar os sinais dos sensores de distância e dos encoders das rodas, e implementar rotinas de baixo nível para acessar os dados dessa leitura. Esse sistema também deve implementar rotinas de baixo nível para controlar a potência dos motores e calcular deslocamento, velocidade e aceleração linear e angular da plataforma. A plataforma deve ser extensível a qualquer outro sistema de processamento. O microcontrolador desta deve implementar rotinas de baixo nível para comunicação de dados. A plataforma deve apresentar interface de alto nível para leitura dos dados dos sensores e recebimento de comandos de movimentação. A documentação de hardware e software da plataforma deve atender aos requisitos de um projeto de engenharia, incluindo: diagramas de blocos, diagramas esquemáticos, diagramas UML de classes, casos de uso, entre outros. Os dois algoritmos de navegação deverão ser implementados em uma hardware acoplado à plataforma robótica, visando reduzir atrasos de comunicação. Este hardware deverá ser capaz de ler os dados dos sensores do robô como fonte de informações para os algoritmos e enviar comandos de movimentação ao mesmo. Os testes dos algoritmos deverão realizados inicialmente em um ambiente aberto e livre de obstáculos, onde o robô deverá deslocar-se em uma trajetória de um ponto inicial a um ponto final com velocidade constante. Posteriormente, o ambiente será limitado progressivamente, acrescentando-se obstáculos entre os dois pontos da trajetória. Em cada caso, os dois algoritmos serão experimentados sob as mesmas condições.

2 ESTADO DO PROJETO

Este capítulo visa descrever com quais recursos a equipe iniciou a execução do projeto, tanto materiais como imateriais. Ou seja, este capítulo descreve a situação do robô e seus componentes físicos, o principal recurso material do projeto e, também, de seus componentes de software e a documentação dos mesmos, da forma como entregue à equipe no início do projeto.

2.1 VISÃO GERAL

O robô disponibilizado à equipe para a realização do projeto, o Bellator, é resultado de um projeto não concluído, visando a implementação de uma plataforma robótica controlada remotamente por joystick. Para tal, o robô foi dividido em três camadas: Baixo nível, Alto nível e Supervisório. A camada de baio nível seria responsável por controlar os motores do robô, bem como receber leituras dos sensores do mesmo. A camada de alto nível comunicaria-se com a camada de baixo nível via conexão serial, faria obtenção de vídeo através de uma webcam e comunicaria-se com a camada supervisória através de uma conexão sem-fio. Esta última seria a interface com o usuário para realizar o controle remoto do robô. O diagrama esquemático a seguir demonstra esta situação:

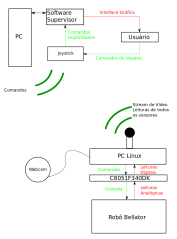

Figura 1: Diagrama original do projeto Bellator, retirado da monografia do mesmo.
Fonte: (DESENVOLVIMENTO..., 2010)

A partir da figura 1, o funcionamento do projeto bellator pode ser analisado. A camada de baixo nível é composta pelo robô bellator, equipado com dois motores elétricos Bosch FPG

12V, seis sensores de distância ”2Y0A02F98”da Sharp, uma bateria Unybatt 12V-7.2 Ampére hora, duas pontes H e a placa micro-controlada C8051F340, capaz de ler e converter leituras de tensão analógicas dos sensores do robô bem como produzir sinais de controle para os motores do robô. Esta placa está conectada à camada de alto nível, composta por um PC Embarcado VIA EPIA ME60000 Mini-ITX com sistema operacional Linux, através de uma conexão serial. Utilizando-se de um protocolo de comunicação, este PC Embarcado envia comandos de movimentação para a camada de baixo nível e recebe as leituras dos sensores obtidas pela camada de baixo nível. O PC Embarcado também comunica-se com a camada de alto nível para receber comandos de movimentação do usuário e enviar as leituras dos sensores para o mesmo. Além disso, o PC Embarcado envia, também via comunicação sem-fio, um stream de vídeo gerado por uma webcam Genius iLook 316. Finalmente, o software supervisor remoto, executado em um PC com máquina virtual java, fornece as informações recebidas da camada de alto nível para o usuário, ajudando-o a tomar decisões sobre a locomoção do robô e recebe comandos de movimentação do usuário, gerados em um Joystick do videogame Sony Playstation 2, enviando-os para a camada de alto nível.

O leitor pode observar desde já que o projeto Bellator apresenta objetivos muito distintos do projeto ”Algoritmos de Navegação Fuzzy Uma Análise Qualitativa”. Assim, não será feita uma descrição detalhada de todos os componentes do projeto bellator. A seguir, será descrito como o robô foi recebido pela equipe e quais componentes foram re-aproveitados.

2.2 RECEBIMENTO DO ROBÔ

O robô Bellator foi entregue a equipe no dia TODO, em uma caixa, desmontado, juntamente de toda a documentação disponível do mesmo, por mídia digital. A caixa continha os seguintes itens:

- Chassi do robô Bellator com dois motores Bosch FPG12V e pontes H acoplados
- Cinco sensores de distância ”2Y0A02F98”da Sharp
- Uma bateria Unybatt 12V-7.2 Ampére hora
- Uma placa micro-controlada C8051F340
- Uma placa de roteamento, produzida pelo projeto Bellator
- Um PC Embarcado VIA EPIA ME60000 Mini-ITX

O chassi do robô e seus componentes acoplados são a base da plataforma robótica a ser utilizada pela equipe e críticas para a execução do projeto. Os sensores de distância, um a menos que os disponíveis no projeto Bellator, são essenciais para a localização do robô e de obstáculos, sendo críticoos para o projeto, também. A placa micro-controlada também é um recurso crítico para o projeto pois é um componente responsável por todo o controle de baixo nível do robô que, além disso, possui software reaproveitável e documentado. A bateria é um recurso necessário para o projeto, porém é de fácil acquisição, ao contrário dos outros componentes aqui citados. Visto a importância destes elementos para o projeto, eles serão descritos em maiores detalhes na seção **??**, com apoio da documentação produzida pelo projeto Bellator. Alguns itens mencionados na seção anterior, referentes ao projeto Bellator, não foram recebidos ou não serão utilizados no novo projeto. A webcam e joystick mencionados na seção anterior não foram entregues, pois não serão necessários para este projeto, visto que um sistema de navegação autonoma não necessita de Joystick e, no caso deste projeto, não será capaz de utilizar-se de um stream de vídeo. A placa de roteamento será utilizada apenas como auxílio para testes iniciais dos componentes do robô, visto que é essencial para o funcionamento do robô, embora seja muito rudimentar. Esta placa será reprojetada e reconstruída. O PC Embarcado foi entregue destituído de qualquer documentação sobre os componentes de software do mesmo. Além disso, como o novo objetivo do robô não necessita de comunicação sem-fio e não utilizará stream de vídeo, motivo principal para a utilização deste PC no projeto anterior, a equipe optou por descartar este recurso do projeto e utilizar outra placa mais simples e menor, que será descrita em detalhes na seção **??**.

A documentação disponível à equipe, produzida pelo projeto Bellator, descreve em detalhes os componentes de hardware da plataforma robótica Bellator, do software de controle supervisório e da camada de baixo nível do robô, ou seja, o software da placa C8051F340**??**. Esta documentação será utilizada como referência para o reaproveitamento da plataforma robótica, com exceção da documentação referente ao software de controle supervisório, que não será utilizado.

2.2.1 Teste dos Componentes

Após o recebimento do robô, foram realizados testes para garantir a funcionalidade dos componentes recebidos, visto que a falha de alguns destes componentes poderiam implicar na impossibilidade de continuar o projeto ou em atrasos significativos. Estes testes também foram necessários para determinar de forma mais precisa o que poderia ser reaproveitado do projeto Bellator. De acordo com a documentação do projeto Bellator**??**, o robô deveria ser

capaz de, se montado conforme as instruções na mesma, funcionar como um sistema controlado remotamente. A equipe optou por testar apenas a camada de baixo nível, como definida no projeto Bellator, isoladamente. O robô foi montando utilizando-se do chassi e seus componentes acoplados, a placa de roteamento, a bateria e a placa C8051F340. Foi observado mal-funcionamento da placa de roteamento, que foi fácilmente reparada através da troca de um transistor.

Com os componentes mais críticos testados, a equipe tentou testar o PC Embarcado. Porém, a ausência de documentação sobre este documento impossibilitou o teste e, após inúmeras tentativas falhas de utilização do mesmo, a equipe optou por não utilizar o PC Embarcado VIA EPIA ME60000 para o projeto, visando reduzir o crescente atraso causado por este. A camada supervisória do projeto Bellator não foi testada, visto que o robô passará a ser autônomo, não necessitando de uma camada supervisória.

3 LÓGICA FUZZY

Esta seção tem como função descrever o que é a lógica fuzzy, como surgiu, como funciona e como projetar um sistema fuzzy.

A lógica fuzzy é baseada no trabalho de Lofti Zadeh, que propôs a teoria de conjuntos fuzzy em 1965 (WIKIPEDIA, 2011). Na teoria clássica de conjuntos, um elemento tem seu grau de pertinência ao conjunto representado por uma variável booleana, ou seja, ou o elemento pertence ao conjunto ou não pertence. Num conjunto fuzzy, no entanto, um elemento tem seu grau de pertinência representado por um valor real no intervalo [0, 1].

Uma aplicaçao básica e clássica da lógica fuzzy é a classificação de uma variável contínua. Por exemplo, se fosse necessário classificar um valor de temperatura, poderia ser utilizado um conjunto fuzzy com as variáveis FRIO, MORNO e QUENTE. Para saber a classificação de certo valor de temperatura, poderia ser utilizado algo como o que está representado na figura 2.

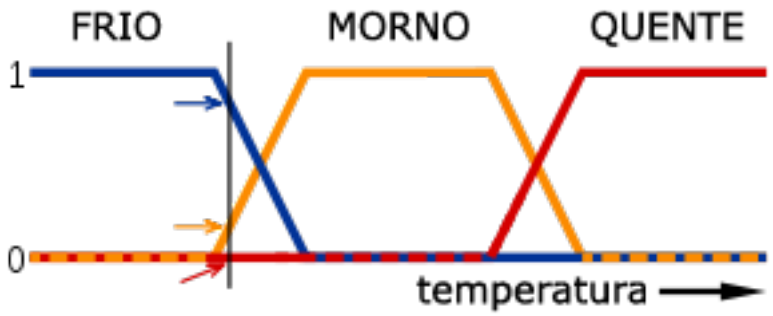

Figura 2: Classificação de valores de temperatura utilizando lógica fuzzy.
Fonte: (WIKIPEDIA, 2011)

A temperatura destacada na figura tem os seguintes graus de pertinência: 0.8 FRIO, 0.2 MORNO e 0 QUENTE. É comum também associar outras variáveis linguísticas para caracterizar os valores. No caso do exemplo, uma associação possível seria: bastante frio, levemente morno, nada quente.

Com base na saída do classificador, um sistema fuzzy pode tomar decisões. Por exem-

plo, caso o objetivo fosse controlar a temperatura de uma caldeira, para deixá-la bem quente, a observação de que está bastante frio implicaria na necessidade da elevação da quantidade de energia disponibilizada para seu aquecimento. Conforme a temperatura subisse, novas observações seriam feitas e classificadas, levando a outras regulações da fonte de calor, até chegar à temperatura esperada. Quanto mais próximo do nível esperado, menores as variações causadas na regulação da fonte de calor.

As decisões tomadas pelo sistema são definidas por um conjunto de regras fuzzy, cujo formato básico é:

SE X ENTÃO Y

onde X é uma expressão contendo uma ou mais variáveis linguísticas e Y a ação a ser tomada. Retomando o exemplo da caldeira, uma regra possível seria: SE bastante frio ENTÃO aumentar bastante o fornecimento de calor.

É evidente que os conceitos FRIO, MORNO e QUENTE são totalmente subjetivos (cada aplicação tem parâmetros específicos). O mesmo é válido para outras grandezas físicas, tais como: velocidade, distância, pressão, força, aceleração etc. Os valores e funções utilizadas para avaliar tais grandezas (algo como o modelo mostrado na figura 2) são determinados por quem está desenvolvendo o sistema fuzzy, com os devidos ajustes para melhor adequação à aplicação em questão. Da mesma forma, as decisões a serem tomadas com base nos dados classificados são definidas pelo usuário. Isso é interessante quando se considera a adição de novos sensores e, consequentemente, novos dados. Uma simples alteração ou adição de novas regras é o suficiente para adaptar o sistema às novas entradas.

Pelo fato de permitir a classificação de dados de uma forma mais flexível (em comparação com a teoria clássica de conjuntos), os dados de entrada para um sistema fuzzy não precisam ser tão precisos, sem ruídos, já que a presença de ruídos causaria apenas pequenas variações na classificação (enquanto numa classificação binária um nível baixo de ruído pode mudar uma classificação totalmente, ao passar um limite pré-definido).

Para projetar um sistema fuzzy, o seguinte conjunto de passos pode ser seguido (KAEHLER, 1998):

1. Definição do que vai ser controlado, com quais critérios, qual o tipo de saída desejado.
2. Determinação da relação entre a entrada e a saída, escolha do número de variáveis de entrada para o sistema.
3. Utilizando a estrutura de regras da lógica fuzzy, dividir o problema principal em várias

tarefas menores. Estas regras devem descrever o comportamento desejado conforme definido no item 1. O número e complexidade das regras depende do número de parâmetros de entrada a serem processados e do número de variáveis fuzzy associadas aos mesmos.

4. Criação das funções de classificação, que definem os significados (comumente através de variáveis linguísticas) dos valores de entrada e saída. As saídas das funções de classificação serão utilizadas nas regras

5. Criação das funções para pré e pós-processamento necessárias (para tratamento dos dados de entrada e saída)

6. Testar o sistema, avaliar os resultados, ajustar as regras e funções de classificação. Repetir este passo até obter um comportamento satisfatório.

4 MAPAS COGNITIVOS FUZZY

Esta seção tem como objetivo explicar o que são mapas cognitivos fuzzy, também conhecidos como FCM (Fuzzy Cognitive Maps), abordando o conceito, a estrutura, as propriedades e vantagens desse modelo, apresentar exemplos, descrever uma metodologia para construí-lo e uma metodologia de aprendizado de máquina em sistemas inteligentes utilizando FCM.

O modelo FCM é abordado na tese de doutorado de (MENDONÇA, 2011). Mapas cognitivos são diagramas que representam ligações entre palavras, idéias, tarefas ou outros itens ligados a um conceito central, em volta do qual os elementos são dispostos radialmente e de forma intuitiva, de acordo com a importância dos conceitos. Um mapa cognitivo pode ser tratato matematicamente por meio de operações com matrizes e baseado na teoria de grafos. Desse modo, crenças ou afirmações a respeito de um domínio de conhecimento limitado são expressas por palavras ou expressões linguísticas, interligadas por relações de causa e efeito. Esse modelo propõe uma organização de idéias onde é possível predizer as consequências que essa organização implica ao universo representado, sendo considerado um modelo matemático da ”estrutura de crenças”. O mapa cognitivo fuzzy é gerado quando se incluem a essa estrutura incertezas através da lógica Fuzzy.

A estrutura de um mapa cognitivo fuzzy é um grafo direcionado, figura 3, em que os valores numéricos são variáveis ou conjuntos fuzzy, os ”nós”são conceitos linguísticos, representados por conjuntos fuzzy e cada ”nó”é associado a outros através de conexões, a cada qual está associado um peso numérico, o qual representa uma variável fuzzy relacionado ao nível de causalidade entre os conceitos.

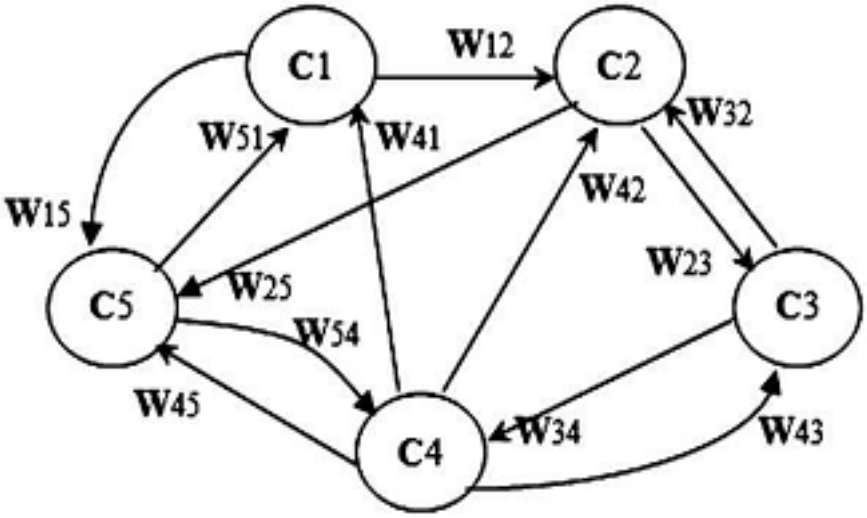

Figura 3: Exemplo de um FCM (grafo).

Fonte: (MENDONÇA, 2011)

5 CONCLUSÃO

Espera-se que o uso do estilo de formatação LaTeX adequado às Normas para Elaboração de Trabalhos Acadêmicos da UTFPR (`abnt-UTFPR.cls`) facilite a escrita de documentos no âmbito desta instituição e aumente a produtividade de seus autores. Para usuários iniciantes em LaTeX, além da bibliografia especializada já citada, existe ainda uma série de recursos (**??**) e fontes de informação (**????**) disponíveis na Internet.

Recomenda-se o editor de textos Kile como ferramenta de composição de documentos em LaTeX para usuários Linux. Para usuários Windows recomenda-se o editor TeXnicCenter (**??**). O LaTeX normalmente já faz parte da maioria das distribuições Linux, mas no sistema operacional Windows é necessário instalar o software MiKTeX (MIKTEX, 2009).

Além disso, recomenda-se o uso de um gerenciador de referências como o JabRef (**??**) ou Mendeley (**??**) para a catalogação bibliográfica em um arquivo BibTeX, de forma a facilitar citações através do comando `\cite{}` e outros comandos correlatos do pacote abnTeX. A lista de referências deste documento foi gerada automaticamente pelo software LaTeX + BibTeX a partir do arquivo `reflatex.bib`, que por sua vez foi composto com o gerenciador de referências JabRef.

O estilo de formatação LaTeX da UTFPR e este exemplo de utilização foram elaborados por Diogo Rosa Kuiaski (diogo.kuiaski@gmail.com) e Hugo Vieira Neto (hvieir@utfpr.edu.br). Sugestões de melhorias são bem-vindas.

REFERÊNCIAS

DESENVOLVIMENTO do Projeto Bellator. [S.l.: s.n.], 2010.

FRACASSO PAULO T.; COSTA, A. H. **Navegação Reativa de Robôs Móveis Autônomos Utilizando Lógica Nebulosa com Regras Ponderadas**. [S.l.: s.n.], 2005.

KAEHLER. **Fuzzy Logic Tutorial**. 1998.

MENDONÇA, M. **Uma Contribuição ao Desenvolvimento de Sistemas Inteligentes Utilizando Redes Cognitivas Dinâmicas**. [S.l.: s.n.], 2011.

MENDONÇA M.; ARRUDA, L. N. F. **Qualitative Autonomous Navigation System Employing Event Drive-Fuzzy Cognitive Maps and Fuzzy Logic**. [S.l.: s.n.], 2010.

MIKTEX. **The MiKTEX project**. 2009. Disponível em: <http://www.miktex.org>. Acesso em: 8 de novembro de 2009.

WIKIPEDIA. **Fuzzy Logic**. 2011. Disponível em: <http://en.wikipedia.org/wiki/Fuzzy_logic>. Acesso em: 29 de julho de 2011.

APÊNDICE A – NOME DO APÊNDICE

Use o comando `\apendice` e depois comandos `\chapter{}` para gerar títulos de apêndices.

ANEXO A – NOME DO ANEXO

Use o comando `\anexo` e depois comandos `\chapter{}` para gerar títulos de anexos.