

Análise comparativa de ferramentas computacionais para modelagem de lógica fuzzy.

Dayse Mourão Arruda
dayse.arruda@int.gov.br
INT

Gabriel Moreira Dias Abud
gabriel.abud@int.gov.br
INT

Felipe Arruda Pontes
felipe.arruda@int.gov.br
INT

Rogério Macedo Pontes
rogerio@pep.ufrj.br
UFRJ/COPPE

Bruno Barcellos Farias de Oliveira
bruno.oliveira@int.gov.br
INT

Resumo:A lógica fuzzy permite modelar matematicamente a tomada de decisão humana de forma a incorporar as imprecisões características da linguagem natural e os aspectos subjetivos e qualitativos inerentes ao processo decisório em problemas reais. Apesar da resistência inicial do meio acadêmico, atualmente já pode ser considerado um campo de conhecimento bastante consolidado, com aplicações nas mais diversas áreas da engenharia. O sistema GesPlan, é uma ferramenta computacional de apoio a atividade de definição e avaliação de planos de produção alternativos, e foi utilizada em sua modelagem a abordagem baseada em Sistemas de Inferência Fuzzy. No início de seu desenvolvimento, foram pesquisadas dentre as bibliotecas disponíveis para a construção de sistemas fuzzy, qual seria mais adequada para ser incorporada ao aplicativo em questão. Considerando o exposto, o objetivo deste texto é apresentar os resultados de um levantamento que apresenta uma análise comparativa dentre as diversas ferramentas disponibilizadas para a modelagem de lógica fuzzy. Este estudo apresenta esquematicamente as orientações gerais para a seleção da biblioteca de lógica fuzzy mais adequada, tendo em vista os diferentes perfis de utilização no contexto de novos desenvolvimentos. A análise leva em consideração aspectos como licença de distribuição, linguagem, recursos gráficos e algoritmos implementados.

Palavras Chave: lógica fuzzy - sistemas fuzzy - sistema inteligente - lógica nebulosa -

1. INTRODUÇÃO

Desde que o conceito de conjuntos fuzzy foi introduzido no já clássico artigo de Zadeh(1965) como uma forma de modelar incertezas e imprecisões da linguagem natural, houve uma grande expansão no uso de seus recursos e desdobramentos nas mais diversas áreas de aplicação. Já em meados da década de 70 pesquisadores japoneses voltaram sua atenção para o assunto e desde então ocorreram avanços significativos na utilização prática desses conceitos. Atualmente já existem milhares de patentes e produtos sendo comercializados empregando esta tecnologia. É comum o uso das ideias advindas da teoria dos conjuntos fuzzy em aplicações tais como: controle de processos industriais, reconhecimento de padrões, auxílio a tomada de decisão, controle de tráfego urbano, produtos de consumo como geladeiras, ar condicionado, máquinas fotográficas, aspirador de pó e TV dentre outras. Em Ross(1995) por exemplo, são detalhados vários casos de aplicação de lógica fuzzy em engenharia de um modo geral.

Na realidade a lógica fuzzy é um dos elementos que constituem a área de Inteligência Computacional ou Soft Computing, conforme denominado por Zadeh(1994, p.77), e corresponde a uma família de técnicas computacionais que estudam mecanismos adaptativos sob a forma de sistemas computacionais para implementar comportamento inteligente em ambientes dinâmicos e complexos. Dentre essas técnicas pode-se destacar as redes neurais, computação evolucionária, e sistemas fuzzy, sendo este último tópico o foco desse estudo.

No escopo do projeto *Faperj: E-26/110.322/2007 – “GesPlan – Desenvolvimento de um protótipo de sistema adequado ao planejamento mestre da produção com mecanismos inteligentes de apoio a decisão”* foi empregado o ferramental teórico oferecido pela lógica fuzzy para modelar o problema do planejamento mestre da produção de forma a incorporar no processo de tomada de decisão o aspecto subjetivo, impreciso e conflitante característico dessa atividade de planejamento, uma vez que a abordagem baseada em Sistemas de Inferência Fuzzy (SIF) é bastante adequada a contextos com essas peculiaridades (ARRUDA et al, 2012).

No âmbito do sistema desenvolvido em java, e funcionando em plataforma Web resultante do projeto, foi empregado um conjunto de algoritmos fuzzy com o objetivo de auxiliar o processo de geração de planos de produção alternativos e na avaliação destes planos segundo critérios pré-estabelecidos. Para realizar essa tarefa foram pesquisadas dentre as ferramentas disponibilizadas para modelar sistemas de inferência fuzzy qual a mais indicada para ser utilizada/adaptada no projeto em questão, considerando os seguintes requisitos:

- ser um Software Livre, de tal forma que fosse possível acessar seu código fonte, modificá-lo se necessário e distribuí-lo junto às aplicações implementadas;
- ser desenvolvido em JAVA para permitir incorporá-lo mais facilmente ao projeto;
- ter documentação disponível para sua utilização.

Considerando o exposto, optou-se pela utilização do XFuzzy 3.0 que é um ambiente de desenvolvimento baseado em sistemas de inferência fuzzy que será descrito mais adiante.

O objetivo deste texto é apresentar de forma esquematizada o levantamento que foi realizado para selecionar a ferramenta mais adequada ao contexto da pesquisa, descrevendo as características observadas em cada um dos softwares de forma a disponibilizar informações que facilitem a escolha dos mesmos em outras circunstâncias para a comunidade científica em geral.

No próximo tópico é discutida a metodologia empregada na pesquisa. No item 3 é apresentado de forma resumida o referencial teórico no campo da lógica fuzzy e no quarto

tópico é feita uma categorização das ferramentas segundo alguns critérios definidos com um posterior detalhamento das características das bibliotecas que se destacaram considerando sua robustez e o tipo de recursos oferecidos na seção 5.

2. METODOLOGIA

Em relação a sua natureza, este trabalho pode ser classificado como sendo *pesquisa aplicada*, conforme Silva & Menezes (2005), uma vez que objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos, envolvendo verdades e interesses locais. No que se refere aos seus objetivos, é uma *pesquisa exploratória*, pois visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses, envolvendo levantamento bibliográfico ou análise de exemplos que estimulem a compreensão do problema.

O levantamento foi realizado a partir da bibliografia disponível na área e principalmente acessando os repositórios relativos às diversas ferramentas na internet e os sites dos projetos. Sempre que possível realizou-se testes com os softwares analisados. Ao longo da pesquisa o foco foi direcionado para a avaliação de algumas características mais determinantes na escolha do software, considerando sua compatibilidade com o restante do sistema GesPlan ao qual seria integrado, conforme descrito inicialmente.

3. REFERENCIAL TEÓRICO EM LÓGICA FUZZY

A teoria fuzzy se constitui de uma poderosa ferramenta adequada a modelagem matemática de problemas reais onde incertezas e imprecisões estão presentes. Este tipo de abordagem possibilita a representação de conceitos vagos e imprecisos (ZADEH, 1988, p.83).

A teoria dos conjuntos fuzzy é uma extensão da teoria dos conjuntos clássica e é baseada na ideia de que os conjuntos existentes no mundo real não possuem limites precisos. Um conjunto fuzzy é caracterizado como um agrupamento impreciso e indefinido, onde a transição da pertinência para a não-pertinência ocorre de forma gradual e não abrupta.

Na definição formal de um conjunto fuzzy diz-se que se X é uma coleção de objetos denotados genericamente por x , então um conjunto nebuloso A em X é um conjunto de pares ordenados, da forma:

$$A = \{ (x, \mu_A(x)) \mid x \in X \}$$

onde: $\mu_A(x)$ é denominada **função de pertinência** de x em A , que mapeia X em um grau de pertinência que varia entre 0 e 1.

Quando este intervalo contém apenas 0 e 1 então $\mu_A(x)$ equivale a função característica de um conjunto clássico (JANG et al, 1997, p. 14).

O valor de pertinência de um elemento reflete a sua compatibilidade com as características e propriedades do conjunto, podendo variar desde a situação da exclusão completa, onde $\mu_A(x) = 0$, até o ponto da inclusão completa, com $\mu_A(x) = 1$

Assim como a teoria dos conjuntos clássicos dispõe dos operadores de união, interseção e complemento, a teoria dos conjuntos fuzzy emprega uma extensão desses operadores, alterados de forma a considerar o aspecto da pertinência parcial, conforme definidos por Zadeh(1965).

Várias formulações para operadores fuzzy generalizados podem ser encontradas na literatura. As **t-normas** (ou normas triangulares) são operadores que generalizam a operação de interseção, enquanto que as **s-normas** (ou t-conormas) generalizam a operação de união. Cox(1994), por exemplo, discute os efeitos da escolha do operador no resultado do problema,

pois a seleção do operador vai depender principalmente das características da aplicação. Vide, por exemplo, Klir e Yuan(1995) que apresentam uma boa variedade desses operadores.

Outro elemento central na lógica nebulosa é o conceito das **variáveis linguísticas**, que são constituídas de palavras ou sentenças curtas em linguagem natural, e usadas no lugar das variáveis numéricas (ZADEH, 1988, p.91).

Associados às **variáveis linguísticas** são usados **modificadores** na modelagem com o objetivo de conseguir maior semelhança com a linguagem natural. Sua função equivale àquela que os adjetivos e advérbios assumem na linguagem. Ou seja, da mesma forma que estes mudam as características de substantivos e verbos, os modificadores, na teoria dos conjuntos nebulosos, alteram a forma das funções de pertinência, transformando um conjunto nebuloso em um novo conjunto.

Um modificador de uma variável linguística é descrito por uma função matemática definida de forma heurística. Sua especificação é subjetiva, e depende grandemente da percepção do projetista.

Maiores informações sobre o tema dos modificadores podem ser encontradas em Aguiar e Oliveira Jr (1999), ou Bárdossy e Duckstein(1965). Destaca-se ainda Cox(1994), que dedica um capítulo inteiro ao assunto e faz uma abordagem bem detalhada.

As **regras fuzzy** são um elemento chave para representar “pedaços” de conhecimento no escopo da lógica fuzzy. O conjunto de regras de um sistema nebuloso é a sua linguagem de representação de conhecimento e elas são usadas para descrever a interdependência entre as variáveis de entrada e de saída do sistema.

Os dois modelos de regras mais comumente empregados são as regras de Mamdani, também denominado modelo linguístico e as do tipo Takagi-Sugeno-Kang (TSK) (JANG et al. ,1997).

Os **Sistemas de Inferência Fuzzy**, por sua vez, tem sua estrutura computacional baseada nos conceitos da teoria dos conjuntos fuzzy e empregam um processo de inferência denominado **raciocínio aproximado**, que deriva conclusões a partir de um conjunto de regras fuzzy e fatos conhecidos.

Em linhas gerais um sistema fuzzy comumente executa as seguintes etapas:

- **Fuzzificação:** Corresponde a etapa de mapeamento dos valores reais das variáveis de entrada e saída em conjuntos fuzzy, o que passa pela definição de quantos conjuntos fuzzy terão cada uma das variáveis, da forma das suas funções de pertinência e os respectivos intervalos;
- **Inferência:** Determina as ações de controle que devem ser empregadas, de acordo com as regras expressas na base de regras. Em outras palavras, transforma os conjuntos nebulosos de entrada num conjunto nebuloso de saída, através da relação definida pela base de regras, e a partir de um determinado mecanismo de inferência em particular.
- **Defuzzificação:** Determina uma saída numérica a partir do conjunto nebuloso de saída obtido na etapa de inferência. Ou seja, traduz o valor linguístico obtido como resultado, num valor discreto que melhor o represente. É uma operação inversa àquela feita inicialmente, que transformou os valores originais para o domínio nebuloso. Existem vários métodos de defuzzificação e sua escolha depende do contexto sendo analisado.

Um dos métodos mais utilizados na etapa de defuzzificação é o do centro de gravidade, que calcula o centróide da área do conjunto nebuloso resultante, ou seja, informa o valor que divide a área do conjunto nebuloso final em duas partes iguais (KLIR & YUAN, p.336, 1995).

4. CATEGORIZAÇÃO DAS FERRAMENTAS

Ao longo do levantamento dos softwares foram analisadas questões tais como a linguagem de programação que foi empregada em seu desenvolvimento, sua forma de distribuição, algoritmos fuzzy implementados, existência de ferramenta gráfica para auxílio a modelagem, documentação disponível e compatibilidade com o padrão para *Fuzzy Control Language* (FCL) estabelecido no relatório publicado pela IEC, International Electrotechnical Commission(1997).

O objetivo da definição e utilização deste padrão pela comunidade científica é permitir que o desenvolvedor tenha mais flexibilidade e possa trocar entre bibliotecas fuzzy alternativas que empreguem o mesmo padrão para FCL, mantendo o resto de seu programa intacto.

No âmbito deste levantamento, um dos requisitos é que a ferramenta deveria funcionar em plataforma Web, preferencialmente empregando Java ou outra linguagem que possibilitasse alguma forma de integração com Java. Outro aspecto considerado relaciona-se a forma de distribuição do software em análise - era preferível que fosse uma ferramenta distribuída em alguma das modalidades de licença de Software Livre uma vez que o software de gestão resultante poderia ser distribuído nesses moldes.

Para facilitar a análise das informações disponibilizadas no levantamento, uma breve explanação sobre os principais conceitos relacionados a Softwares Livres é apresentada a seguir, considerando somente aquelas modalidades de licença usadas nas ferramentas pesquisadas. Na figura 1 pode ser visualizado um esquema agrupando as ferramentas pesquisadas quanto a sua forma de distribuição.

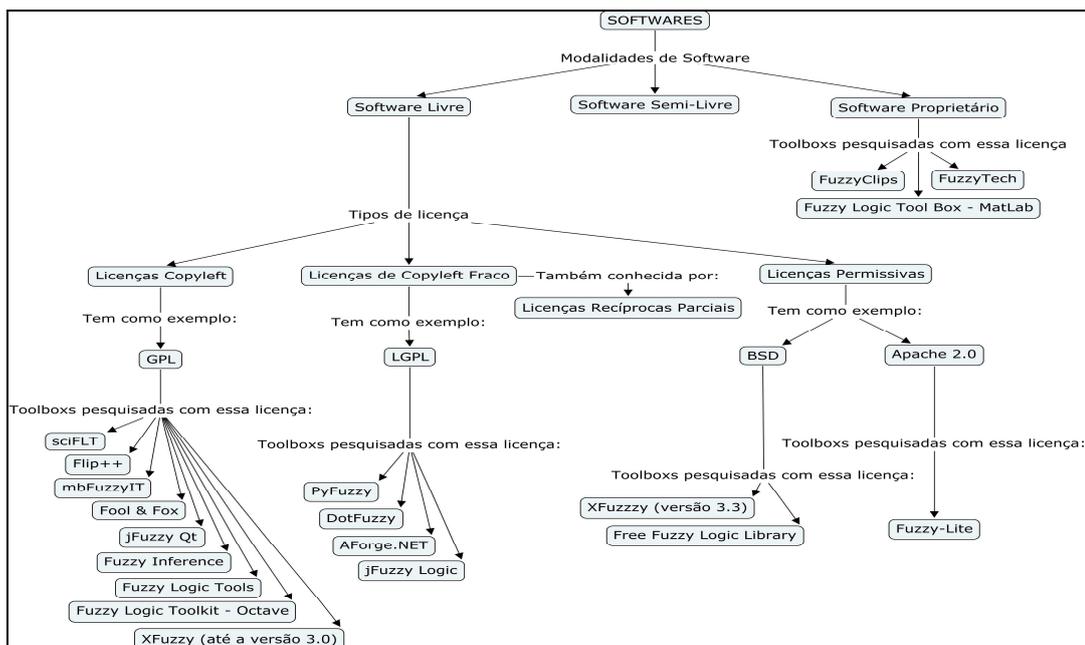


Figura 1: ferramentas pesquisadas agrupadas quanto às suas modalidades de licenciamento

Um Software Livre é definido pela Free Software Foundation (FSF) como sendo um software que pode ser usado, copiado ou redistribuído por qualquer pessoa, com ou sem

modificação, gratuitamente ou por um preço. É importante observar, que para que qualquer uma dessas permissões seja efetiva é imprescindível que o código fonte do sistema esteja disponível.

Pode-se dizer, em outras palavras, que o usuário do Software Livre tem resguardadas as quatro liberdades essenciais: liberdade para executar o programa; liberdade para redistribuir cópias exatas; liberdade para estudar e mudar o código-fonte do programa; e liberdade para redistribuir versões modificadas.

Já o Software Semi-Livre é aquele que vem com permissão para uso, copia, distribuição e modificação, desde que seja para fins não lucrativos.

Em oposição direta ao conceito de Software Livre existe a categoria de Softwares Proprietários, que correspondem aos softwares que não são Livres nem Semi-Livres. Ou seja, seu uso, redistribuição ou modificação é proibido, ou requer pedido de permissão, ou é restrito de alguma forma.

As duas licenças mais comuns para Software Livre são a GPL (GNU General Public License¹) definida pela Free Software Foundation e a BSD² (Berkeley Software Distribution) originária do sistema operacional de mesmo nome, desenvolvido pela Universidade da Califórnia. Esta última licença é definida pela Open Source Initiative (OSI), que juntamente com a FSF correspondem às duas maiores instituições representantes do Software Livre.

Como principal diferença entre as duas, pode-se destacar que no caso da licença GPL, é exigido que os trabalhos derivados do software em questão sejam licenciados também sob a licença GPL, enquanto que a licença BSD³, que é bastante simplificada, requer apenas o reconhecimento dos autores e outras pequenas restrições. Em termos práticos, isso implica em que sob uma licença BSD o software pode ser usado tanto em outros projetos de Software Livre como no desenvolvimento de Softwares Proprietários, enquanto que no caso da GPL, a sua utilização pode ser feita apenas em Softwares Livres.

O conceito de *copyleft* é o mecanismo legal que garante que obras derivadas de obras livres sejam também livres. Neste contexto, licenças no estilo GPL são também chamadas de *licenças copyleft*, ou seja, não permitem que redistribuidores incluam restrições adicionais àquelas que já existem, enquanto que licenças no estilo BSD, que não incluem a característica *copyleft*, são também denominadas *licenças permissivas*.

A licença Apache 2.0⁴ - criada pela Apache Software Foundation, e também incluída na categoria das *licenças permissivas*, difere um pouco da licença BSD, devido aos seguintes aspectos: por tratar de forma explícita da questão da patente de software; pela possibilidade da licença Apache ser adotada por referência, ou seja, não exige a inclusão do texto da licença no código fonte; ou ainda, pela possibilidade de inclusão de um arquivo NOTICE como parte da distribuição, cujo conteúdo deve ser apenas informativo, e que precisa ser mantido quando o software for redistribuído.

A GNU Lesser General Public Licence – LGPL, por sua vez, corresponde a um meio termo entre as *licenças copyleft* como a GPL e as *licenças permissivas*, tais como a BSD. A LGPL pertence a categoria das *licenças de copyleft fraco*, também conhecidas como *licenças recíprocas parciais*. Em linhas gerais, essas licenças determinam que modificações feitas em um software sob esta licença sejam disponibilizadas sob a mesma licença, no entanto, quando essas modificações forem usadas como parte de outro projeto de software, este projeto não

¹ A descrição da licença pode ser obtida em: <http://www.gnu.org/licenses/gpl.html>

² Também denominada *BSD new* ou *BSD de 3 cláusulas*

³ Vide versão completa da licença em: <http://opensource.org/licenses/BSD-3-Clause>

⁴ A versão completa da licença pode ser obtida em: <http://www.apache.org/licenses/LICENSE-2.0>

precisa, obrigatoriamente, ser distribuído sob a mesma licença. Ou seja, pode ser distribuído inclusive como Software Proprietário.

Uma discussão mais detalhada sobre o assunto pode ser encontrada em Free Software Foundation (2012), Smith & Free Software Foundation (2007), Campos (2006) ou Leite (2009).

No que se refere a existência de interface gráfica para auxiliar o processo de construção da modelagem fuzzy, pode-se observar que esse recurso está disponível em 8 das 18 ferramentas pesquisadas (vide figura 2).

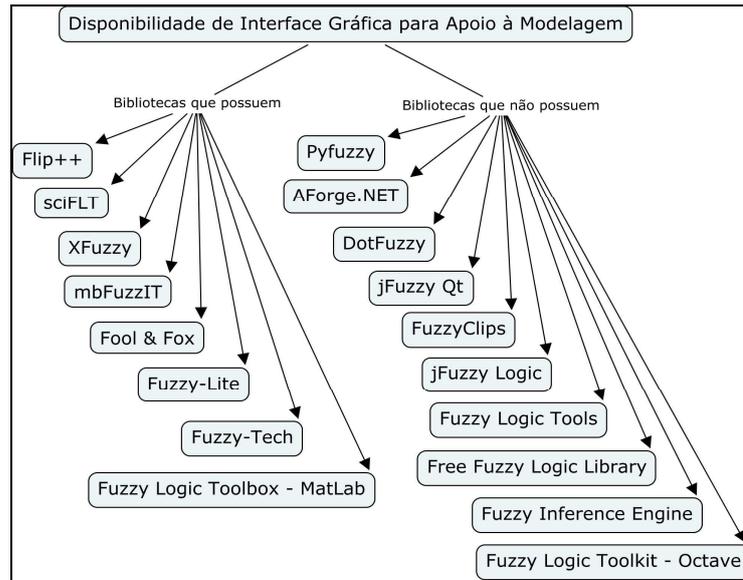


Figura 2: Ferramentas pesquisadas organizadas quanto à existência de interface para modelagem

Em relação à linguagem de programação empregada, observa-se uma certa predominância de c/c++ ou Java, com a maioria das implementações e pode-se perceber também o início da utilização de linguagens menos tradicionais na área de computação científica, tais como c# e python, como pode ser visto na figura 3.

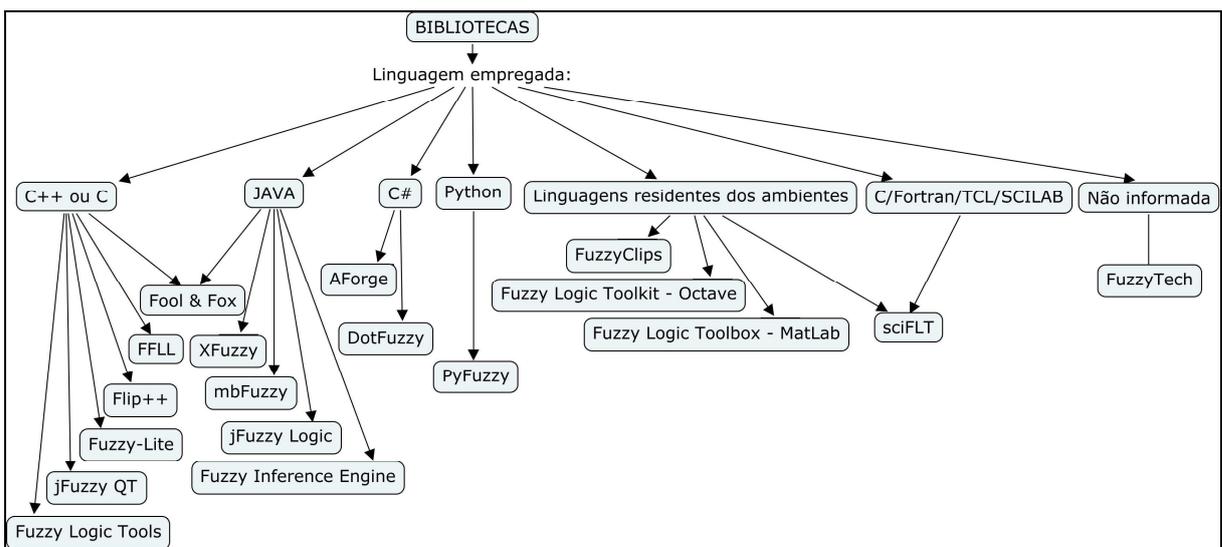


Figura 3 - Ferramentas organizadas quanto à linguagem de desenvolvimento

No que tange a compatibilidade com o padrão para FCL definido pela IEC, International Electrotechnical Commission(1997), pode-se observar sua utilização em 6 das ferramentas pesquisadas, porém algumas apenas carregam e executam arquivos com essa

sintaxe, mas não geram os arquivos com a modelagem dentro do padrão. Outro destaque corresponde ao grupo de quatro ferramentas que empregam a sintaxe da FCL definida na Toolbox Fuzzy do Matlab. O XFuzzy por sua vez emprega uma linguagem específica denominada XFL3 conforme descrito em Moreno-Velo et al (2001a). As demais ferramentas não discriminam de forma explícita o uso de uma linguagem de controle fuzzy. Na figura 4 pode-se observar a distribuição das bibliotecas considerando o emprego da FCL.

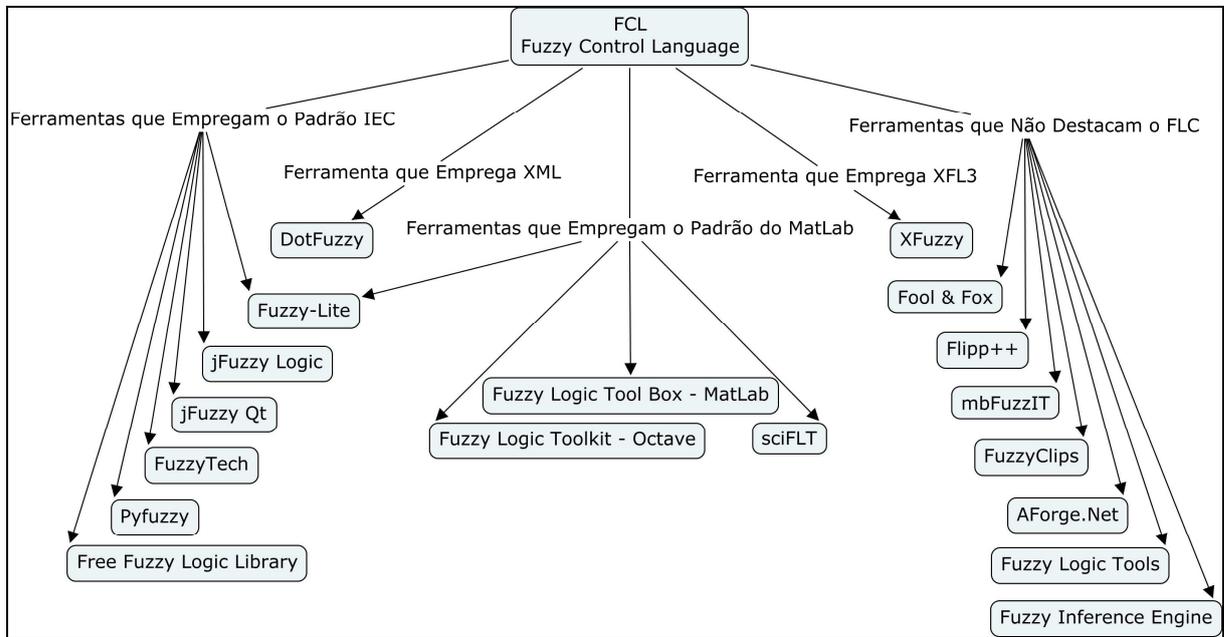


Figura 4 - Ferramentas organizadas quanto ao padrão de FCL

No quadro que se segue são listadas apenas as informações mais relevantes sobre cada uma das ferramentas. Vale destacar que os softwares foram listados em ordem alfabética e não ordenados segundo sua importância.

Tabela 1: Quadro resumo das ferramentas Fuzzy

Ferramenta	Site do Projeto	Distribuição ⁵	Linguagem ⁶	FCL ⁷	Última Versão ⁸
AForge.NET	http://www.aforgenet.com	GNU LGPL	C#	NO	2012
DotFuzzy	http://www.havana7.com/dotfuzzy/default.aspx	GNU LGPL	C#	XML	2009
Flip++	http://www.dbai.tuwien.ac.at/proj/StarFLIP/	GNU GPL	C++	NO	2005
Fool & Fox	http://rhaug.de/fool/	GNU GPL	C/Java	NO	2002

⁵ Informa a modalidade de licença para a distribuição do software.

⁶ Linguagem de programação na qual a ferramenta foi desenvolvida.

⁷ Informa se a ferramenta explicita o uso de uma FCL e qual o padrão usado. A legenda “NO” significa que não foi observado;

⁸ Data referente a ultima alteração no repositório do controle de versionamento, quando este é disponível, caso contrário refere-se à última atualização informada pelo autor.

Free Fuzzy Logic Library	http://ffll.sourceforge.net	BSD 3	C/C++	IEC	2003
Fuzzy Inference Engine	http://people.clarkson.edu/~esazonov/FuzzyEngine.htm	GNU GPL	Java	NO	2005
Fuzzy Logic Tools	http://uhu.es/antonio.barragan/category/temas/fuzzy-logic-tools	GNU GPL	C++	NO	2012
Fuzzy Logic Tool Box - MatLab [©]	http://www.mathworks.com/products/fuzzylogic/	Software Proprietário	Linguagem residente do ambiente MatLab.	Própria	2012
FuzzyClips	http://www.ortech-engr.com/fuzzy/fzyclips.html	Software Proprietário ⁹ .	CLIPS	NO	2004
Fuzzy-Lite	http://www.fuzzylite.com	Apache 2.0	C++	IEC ou padrão MatLab	2011
FuzzyTech [©]	http://www.fuzzytech.com/	Software Proprietário	Não Informado	IEC	2012
jFuzzy Logic	http://jfuzzylogic.sourceforge.net/html/index.html	GNU LGPL	Java	IEC	2012
jFuzzy Qt	http://jfuzzyqt.sourceforge.net/	GNU GPL	C++	IEC	2011
mbFuzzIT	http://mbfuzzit.sourceforge.net/en/mbfuzzit_software.html	GNU GPL	Java	NO	2005
Fuzzy Logic Toolkit - Octave	http://octave.sourceforge.net/index.html	GNU GPL	Linguagem residente do ambiente Octave.	padrão MatLab	2012
Pyfuzzy	http://pyfuzzy.sourceforge.net/	GNU LGPL	Python	IEC	2009
sciFLT	http://www.geocities.ws/jaime_urzua/sciFLT/sciflt.html	GNU GPL	C/Fortran/TC L/SCILAB	padrão MatLab	2004
XFuzzy	http://www2.imse-cnm.csic.es/Xfuzzy/	GNU GPL até a versão 3.0 e BSD 3 na versão 3.3	Java	XFL3	2012

Fonte: levantamento realizado pelos autores

[©] MATLAB é marca registrada da The MathWorks, Inc.

⁹ Originalmente era liberado gratuitamente para uso educacional e de pesquisa.

[©] fuzzyTech é marca registrada da Inform Software Corp.

5. CARACTERÍSTICAS DAS PRINCIPAIS FERRAMENTAS

A seguir é apresentada uma descrição mais completa das ferramentas que se destacaram pela solidez da solução ou pelo conjunto de recursos que apresentaram. Para facilitar a apresentação do levantamento, as bibliotecas foram separadas em dois grandes grupos: aquelas distribuídas como Software Livre e as distribuídas como Software Proprietário. Na seção que se segue são discutidos os aspectos principais do primeiro grupo.

Uma discussão mais ampla e detalhada do levantamento como um todo pode ser obtida em Pontes et al(2013).

5.1. FERRAMENTAS DISTRIBUÍDAS COMO SOFTWARE LIVRE

5.1.1. AForge.NET

Consiste de uma poderosa framework feita para desenvolvedores e pesquisadores nos campos de Computação Visual e Inteligência Artificial, abrangendo Processamento de Imagem, Redes Neurais, Algoritmos Genéticos e Lógica Fuzzy entre outros. Além dos demais recursos oferecidos, a framework possui uma biblioteca de lógica fuzzy - AForge.Fuzzy, que consiste de uma série de classes que permitem desde o uso de simples conjuntos fuzzy e variáveis linguísticas até a construção de sistemas de inferência fuzzy completos.

A ferramenta possui farta documentação que vem junto com o instalador e os arquivos fonte da framework, além de diversos artigos sobre aplicações desenvolvidas, como pode ser visto em Caversan (2009).

A framework não possui uma ferramenta gráfica para ajudar na construção da modelagem fuzzy, porém possui classes que podem ser usadas no Visual Studio da Microsoft para gerar telas do tipo “What You See What You Get”.

5.1.2. FUZZY LOGIC TOOLS

É uma framework que permite armazenar, analisar e projetar sistemas fuzzy genéricos do tipo Takagi-Sugeno, com múltiplas entradas e múltiplas saídas, sem restrições. Segundo seus autores foi compilado e testado em plataformas Linux e Windows.

O software é resultado de uma pesquisa de doutorado defendida em 2009 na Universidade de Huelva (PINA, 2009). Possui várias funções de pertinência implementadas, tais como triangular, trapezoidal e gaussiana. Possui um manual de referencia bem elaborado e que pode ser obtido no site do projeto (PINA & MÁRQUEZ, 2011).

5.1.3. FUZZY-LITE

É um sistema de inferência fuzzy, multiplataforma que funciona principalmente como uma biblioteca, sem nenhuma dependência além do Standard Template Library que vem incluso nas bibliotecas padrões de C++.

Possui uma GUI - Graphic User Interface, que permite visualizar e criar sistemas fuzzy. Esta interface gráfica é baseada na framework Qt¹⁰ e facilita o uso da ferramenta.

A biblioteca dispõe dentre outros recursos, de diversas funções de pertinência tais como: triangular, trapezoidal, discreta, gaussiana, sigmoide e s-shape; além de uma série de operadores lógicos para serem usados como T-Normas e S-Normas. Em termos de método de defuzzificação tem implementado o algoritmo do centroide, menor dos máximos, média dos

¹⁰ Qt é uma aplicação multiplataforma, distribuída como software livre que consiste de uma framework com APIs (Application Programming Interface) para programação em C++ que permite rápida criação de interfaces para usuários. Para maiores informações vide site do projeto: <http://qt-project.org/>

máximos e maior dos máximos. Trabalha com os modificadores “qualquer”, “não”, “extremamente”, “frequentemente” e “muito”. Atualmente trabalha com regras de Mamdani, porém já está previsto a inclusão de regras do tipo Takagi-sugeno. Possui ainda um recurso interessante, que é a importação/exportação de arquivos com modelagem no padrão do MATLAB (*.fis) ou no padrão FCL da IEC.

O site do projeto possui uma boa documentação sobre o uso da biblioteca, além de um grupo no googleGroups para discutir e tirar dúvidas sobre a mesma. Possui também um vídeo demonstrativo de uma aplicação utilizando a biblioteca.

5.1.4. JFUZZY LOGIC

Pacote que disponibiliza uma série de classes para construção de sistemas de inferência fuzzy e implementa uma linguagem de controle fuzzy de acordo com a especificação definida pela IEC. A ferramenta possui classes para várias funções de pertinência tais como: triangular, trapezoidal, singleton e gaussiana dentre outras e ainda permite a criação de função de pertinência customizada pelo usuário, além de método de implicação pelo mínimo e pelo produto e métodos de agregação empregando soma limitada, máximo e etc.

Quanto aos métodos de defuzzificação, disponibiliza o centróide e a média dos máximos, dentre outros.

A biblioteca não possui uma interface que facilite a criação da modelagem fuzzy, mas permite carregar um arquivo FCL e transformá-lo em objetos Java em memória. A partir dessa modelagem “traduzida” para classes Java é possível executar o motor de inferência com essa modelagem. É importante destacar que o JFuzzy Logic não cria o arquivo FCL.

Junto com a ferramenta é disponibilizado um demo que exemplifica o uso da FCL para o mesmo exemplo da definição de gorjeta em função da qualidade do serviço e da comida que é discutido com bastante detalhe no tutorial da toolbox fuzzy do MATLAB (MATHWORKS,2006).

Além da hierarquia de classes e métodos no formato javaDoc¹¹ em HTML, uma explanação detalhada da ferramenta também pode ser obtida em Cingolani & Alcalá-Fdez (2012).

5.1.5. FUZZY LOGIC TOOLKIT - OCTAVE

É uma toolbox fuzzy compatível com MATLAB desenvolvida para o ambiente Octave. Octave é um Software Livre voltado para computação científica e alternativo ao ambiente MATLAB. Possui uma comunidade bastante ativa e uma série de pacotes disponíveis. O pacote Fuzzy Logic Toolkit possui um subconjunto dos comandos encontrados na toolbox fuzzy do MATLAB.

O Toolkit permite a construção, modificação e execução de sistemas fuzzy a partir de comandos de linha dentro do ambiente Octave. Produz ainda saídas gráficas para as funções de pertinência e para a superfície resultante da modelagem fuzzy. Dentre os recursos oferecidos destaca-se uma boa variedade de funções de pertinência, diversos operadores para T-Norma e S-Norma tais como produto algébrico, soma algébrica, soma limitada, produto drástico entre outros. Trabalha com modelo de Mamdani e de Sugeno. Não possui ainda uma ferramenta gráfica para apoio a atividade de modelagem, porém a autora cita este desenvolvimento para as próximas versões (MARKOWSKY & SEGEE, 2011).

¹¹ Javadoc - É um utilitário fornecido pela Sun Microsystems junto ao JDK para gerar documentação de códigos Java em formato HTML a partir de comentários feitos no próprio código fonte Java.



A biblioteca foi desenvolvida na linguagem de alto nível residente do ambiente Octave a qual é bastante compatível com a linguagem do MATLAB, o que permite inclusive que os scripts gerados para um dos softwares funcione também no ambiente alternativo. A toolkit exige o ambiente Octave em versão maior ou igual a 3.2.4 para seu funcionamento.

Em termos de documentação a ferramenta possui um guia de referência em HTML disponível no site do projeto.

5.1.6. PYFUZZY – PYTHON FUZZY PACKAGE

É uma framework que permite trabalhar com conjuntos fuzzy e processá-los com operações da lógica fuzzy.

Não tem interface facilitadora para a montagem da modelagem, mas pode ler arquivos no formato FCL conforme estabelecido pela IEC. Está na versão 0.1.0 de outubro de 2009 e tem uma quantidade considerável de downloads atuais.

No site do projeto podemos encontrar uma extensa documentação, com diagramas, versões atualizadas em PDF e HTML, exemplos de uso e demonstrações.

A ferramenta tem implementadas várias funções de pertinência tais como, triangular, trapezoidal, singleton e poligonais, além dos métodos de defuzzificação pelo centroide, máximo a esquerda e máximo a direita. Possui ainda diversos operadores fuzzy, como por exemplo, max, min, produto algébrico e soma algébrica, dentre outros.

5.1.7. SCIFLT

Consiste de uma toolbox para fuzzy desenvolvida para funcionar no escopo do Scilab, que é um software livre voltado para computação científica e de engenharia desenvolvido pelo INRIA- Institut de Recherche en Informatique et en Automatique.

O Scilab é uma alternativa Open Source ao ambiente MATLAB, que é um software proprietário. Permite inclusive importar arquivos de modelagem gerados no MATLAB.

Entre seus principais recursos pode-se destacar: diversas funções de pertinência implementadas, editor gráfico, implementação de Takagi-Sugeno e Mamdani, métodos de implicação empregando mínimo e produto e diversos métodos de defuzzificação.

Em termos de documentação, no site principal aparecem telas relativas à utilização da toolbox e os principais recursos da ferramenta são discutidos em Chuan, Zengqi & Ling (2004).

5.1.8. XFUZZY

O software foi desenvolvido pelo Instituto de Microeletrônica de Sevilha e consiste de um ambiente de desenvolvimento que possibilita a construção de sistemas de inferência fuzzy. É integrado por um conjunto de ferramentas que atuam para facilitar as diferentes etapas do processo de modelagem.

O ambiente constrói a modelagem fuzzy baseada na linguagem de especificação XFL3, possui várias funções de pertinência, e tem implementados diversos operadores para t-normas, s-normas, implicação e defuzzificação. Também permite o uso de modificadores linguísticos definidos livremente pelo usuário, além de possuir recursos visuais para facilitar a edição dos componentes do sistema e geração de gráficos bidimensionais e tridimensionais. Já em sua versão 3.0 foram incorporados algoritmos para facilitar a obtenção de regras na ferramenta para aprendizado supervisionado. Outro recurso interessante é a possibilidade de construção de

sistemas hierárquicos, onde ocorre a saída de um subsistema fuzzy gerando dados de entrada para outro subsistema.

XFuzzy possui diversos módulos que podem ser executados como programas independentes e que funcionam de forma integrada no escopo de uma interface gráfica que facilita o processo de modelagem. Além do projeto disponibilizar um manual bem elaborado, como pode ser visto em IMSE-CNM (2012) ou Arbex et al(2011), é possível encontrar vários artigos com discussão sobre a utilização da biblioteca no próprio site da ferramenta como, por exemplo, Baturone et al(2007), Moreno-Velo et al(2003) ou Moreno-Velo et al (2001b).

5.2. FERRAMENTAS DISTRIBUÍDAS COMO SOFTWARE PROPRIETÁRIO

5.2.1. FUZZY LOGIC TOOL BOX - MATLAB®

Consiste de uma biblioteca com uma série de funções que permitem criar e editar sistemas de inferência fuzzy no escopo do ambiente MATLAB. Possui uma poderosa interface gráfica que auxilia na modelagem dos sistemas de forma bastante amigável.

As funções da toolbox são escritas na linguagem residente do ambiente matemático em arquivos com extensão *.m* e podem ser customizadas ou estendidas de acordo com a necessidade do usuário.

A toolbox disponibiliza várias funções de pertinência além de permitir a construção de funções de pertinência customizadas pelo usuário. Suporta “and”, “or” e “not” nas regras definidas e aceita regras do tipo Mamdani ou Sugeno. Possui vários métodos de defuzzificação implementados e oferece recursos para aprendizado adaptativo neurofuzzy.

O uso dos recursos relativos à ferramenta do MATLAB para a construção de interface gráfica denominada GUIDE – Graphical User Interface Development Environment, permite o desenvolvimento de aplicativos com janelas, botões e menus bastante amigáveis conforme pode ser visto, por exemplo, em Arruda (2006) e Arruda et al(2007) que discute uma série de simuladores fuzzy para geração de planos de produção alternativos.

O ambiente de desenvolvimento possui uma vasta documentação disponível, não apenas em termos de manuais bastante detalhados e didáticos, como também vários livros e textos acessíveis. Para maiores detalhes sobre o ambiente MATLAB em geral consultar Matsumoto(2001) ou Hanselman e Littlefield(2003).

Adicionalmente alguns autores fornecem em seus livros o código fonte para algoritmos fuzzy na linguagem residente do MATLAB, tais como Jang,Sun e Mizutani (1997), Hines (1997) e Aguiar e Oliveira Jr (1999).

5.2.2. FUZZYTECH®

É uma poderosa toolbox voltada para lógica fuzzy com interface gráfica para definição dos seus componentes. permite edição das regras de diferentes maneiras, como por exemplo, sob a forma de matriz ou como numa planilha. Permite análise das características do sistema em construção através do plot de seus componentes em gráficos 2D ou 3D.

É uma ferramenta proprietária que foi desenvolvida pela INFORM em 1987 para uso interno e em 1990 introduzida no mercado europeu. A empresa atua também em consultoria na área de lógica fuzzy e foi fundada em 1969 pelo professor Zimmermann, um renomado pesquisador no assunto.

Existem várias edições do produto, com diferentes níveis de recurso sendo oferecidos, mas em linhas gerais podemos citar: implementação de várias funções de pertinência,

operadores de mínimo e máximo para agregação dos inputs de cada regra e diversos métodos para defuzzificação, tais como centro da área e centro dos máximos. Possui assistentes de projeto que facilitam todo o processo de modelagem. Todas as edições da ferramenta são compatíveis com o padrão IEC-1131-7.

Além da boa documentação, de vários artigos sobre lógica fuzzy e assuntos correlatos disponíveis no site do produto (vide por exemplo, Fonseca et al (1997)), é possível fazer download de uma versão demo do sistema, a partir de cadastramento no site do produto. A ferramenta possui também help e um assistente para criação de projetos bastante amigável.

6. CONCLUSÃO

Conforme descrito inicialmente, a motivação original para realização deste levantamento foi a de selecionar uma ferramenta fuzzy para ser implementada no escopo do sistema de planejamento que estava sendo projetado. Devido às especificidades do desenvolvimento do sistema GesPlan, optou-se pela utilização da biblioteca XFuzzy dentre os softwares pesquisados, de forma a possibilitar a implantação dos módulos que empregam a abordagem baseada em SIFs.

Os aspectos mais relevantes na orientação da escolha foram as seguintes características apresentadas pela biblioteca: ser desenvolvida na linguagem Java; ser software livre; possuir uma interface gráfica para auxílio ao processo de modelagem; possuir uma grande variedade de operadores fuzzy e demais recursos necessários na construção da modelagem já implementados; ser fartamente documentada e mantida atualizada.

Como orientação geral para a seleção de bibliotecas para novos desenvolvimentos pode-se dizer que um aspecto que certamente é determinante na seleção refere-se à linguagem de programação empregada. É possível identificar, por exemplo, uma preponderância das linguagens C e Java seguida pela utilização das linguagens residentes de ambientes de computação científica mais genéricos tais como MATLAB e Octave, por exemplo.

No que tange a disponibilidade de ferramenta gráfica para auxílio à modelagem pode-se observar sua ocorrência em 8 das bibliotecas pesquisadas, porém com uma grande diferença em termos dos recursos que oferecem e ao nível de maturidade das mesmas. Neste quesito merece destaque a robustez das funções oferecidas na toolbox fuzzy do MATLAB, no FuzzyTech e no Xfuzzy.

Outro aspecto que merece atenção do desenvolvedor se refere ao padrão da FCL empregado. Observou-se o uso do padrão FCL da IEC em 6 das ferramentas, e do padrão usado no MatLab em 4 das ferramentas, além do uso da linguagem XFL3 no XFuzzy.

Em relação à forma de distribuição do software observou-se uma grande participação de Softwares Livres com 15 bibliotecas dentre as pesquisadas enquanto que 3 são distribuídas como Software Proprietário.

Vale salientar ainda, que é preciso levar em consideração na escolha a forma como será distribuída a solução empregando a biblioteca selecionada, uma vez que o tipo de licença da biblioteca usada, pode ser determinante na definição da licença do aplicativo final. Por exemplo, se a ferramenta selecionada tiver a licença GPL, necessariamente o aplicativo no qual será usada deverá seguir a mesma licença.

Observa-se no levantamento realizado uma boa variedade de bibliotecas disponíveis considerando os mais variados portes e diferentes perfis de utilização. Levando em consideração a análise disponibilizada nesta pesquisa, cabe ao desenvolvedor então, diante das características específicas da aplicação que pretende implementar, definir qual a ferramenta mais adequada ao seu problema.

7. REFERÊNCIAS

- AGUIAR, H. & OLIVEIRA JR., *Lógica Difusa – Aspectos Práticos e Aplicações*. Rio de Janeiro: Editora Interciência, 1999. p. 190.
- ARBEX, M. A.; ARRUDA, D. M.; PONTES, R. M.; PONTES, F. A. & ARAUJO, A. M. B. *XFuzzy 3.0 Manual – Versão em português*. Nota Técnica – INT, fev. 2011, 45 p.
- ARRUDA, D. M. *Modelagem do planejamento mestre da produção através do emprego de regras nebulosas*. 2006. 196 p. Tese (Doutorado em Engenharia de Produção) – Universidade Federal do Rio de Janeiro, 2006
- ARRUDA, D. M.; COSENZA, C. A. N.; MARTINS, R. C. & PONTES, R. M. “Proposta de um sistema de inferência nebulosa para geração e avaliação de planos de produção alternativos”, In: XXVII Encontro Nacional de Engenharia de Produção, 2007, 9 p.
- ARRUDA, D. M.; FERNANDES, E.; COSENZA, C. A. N.; PONTES, R. M. & PONTES, F. A. “Desenvolvimento de um protótipo de sistema inteligente voltado para o planejamento da produção”, In: XXXII Encontro Nacional de Engenharia de Produção, 2012, 14 p.
- BÁRDOSSY, A., DUCKSTEIN, L., *Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological and Engineering Systems*. USA: CRC Press, 1995. p. 232.
- BATURONE, I.; MORENO-VELO, F. J.; SANCHEZ-SOLANO, S.; BARRIGA, A.; BROX, P.; GERSNOVIEZ, A. & BROX, M. “Using XFuzzy environment for the whole design of fuzzy systems”, In: *proc. IEEE International Conference on Fuzzy Systems*, pp., London, july. 2007, disponível em: <<http://www.imse-cnm.csic.es/online/2003/FUZZIEEE2003.FMV.pdf>>. Acesso em 21 fev. 2011.
- CAMPOS, A. *O que é software livre*. BR-Linux. Florianópolis, março de 2006. Disponível em <<http://br-linux.org/linux/faq-softwarelivre>>. Acesso em 2 de abril de 2013.
- CAVERSAN, F. L. *Fuzzy Computing: Basic Concepts*. 2009. Disponível em <http://www.aforgenet.com/articles/fuzzy_computing_basics/>. Acesso em 17 de jun de 2013.
- CINGOLANI, P & ALCALÁ-FDEZ, J. *jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation*. FUZZ-IEEE 2012: 1-8 p.
- CHUAN F.; ZENGQI, S. & LING, S. "Design and implementation of Scilab fuzzy logic toolbox", *Computer Aided Control Systems Design*, 2004 IEEE International Symposium on , vol., no., pp.147-151, 4-4 Sept. 2004
- COX, E., *The Fuzzy Systems Handbook, a Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems*. USA: Academic Press, 1994. 626 p.
- FONSECA, J.; AFONSO, J.; MARTINS, J. & COUTO, C. *Avaliação da aplicação de técnicas de lógica fuzzy no controle de máquinas elétricas*. Depto. de Eletrônica. Universidade de Minho. Braga, Portugal. 1997. 7 p.
- FREE SOFTWARE FOUNDATION. *What is free software*, 2012. Disponível em: <<http://www.gnu.org/philosophy/free-sw.en.html>>. Acesso em 2 de abril de 2013.
- HANSELMAN, D. & LITTLEFIELD, B., *MatLab 6- Curso Completo*, Trad. Cláudia Sant’Ana Martins; revisão: Alberto Saa, Francisco A. M. Gomes e M. Aparecida Diniz Ehrhardt. São Paulo: Pearson – Prentice Hall, 2003, 676 p. Tradução de: *Mastering MatLab 6 – A Comprehensive Tutorial and Reference*,
- HINES, J. W. *Fuzzy and Neural Approaches in Engineering – MatLab Supplement*. USA: John Wiley and Sons, Inc., 1997, 218 p.
- IMSE-CNM , *XFuzzy. Fuzzy Logic Design Tools*, Espanha – Sevilla: Instituto de Microeletrônica de Sevilla, 2003, 54 p. disponível em: <http://www2.imse-cnm.csic.es/Xfuzzy/Xfuzzy_3.0/Xfuzzy3.0_en.pdf>. Acesso em 15 fev. 2011.
- INTERNATIONAL ELECTROTECHNICAL COMMISSION, IEC 1131 – *Programmable Controllers. Part 7 – Fuzzy Control Programming* , Committee Draft CD 1.0, 1997. Disponível em: <<http://www.fuzzytech.com/binaries/ieccd1.pdf>>. Acesso em 2 ago. 2011.
- JANG, J. R.; SUN, C. & MIZUTANI, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. USA: Prentice-Hall. Inc., 1997.
- KLIR, G. J. & B., YUAN, *Fuzzy Sets and Fuzzy Logic – Theory and Applications*, USA: Prentice Hall, 1995, 574 p.



LEITE, L. A. F. *Análise de licenças de software*. Relatório Técnico – Grupo de Estudos em Iteração – Escola Politécnica da USP, abr. 2009, 9 p. Disponível em: < http://lts-i.pcs.usp.br/xgov/pub/anexos_xgov/@0031%20LEITE%20Analise%20das%20licencas%20de%20software.pdf> . Acesso em 2 abr 2013.

MARKOWSKY, L. & SEGEE, B. "The Octave Fuzzy Logic Toolkit", *Open-Source Software for Scientific Computation (OSSC), 2011 International Workshop on* , pp.118-125, Oct. 2011. Disponível em < <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6184706&isnumber=6184685> > Acesso em 22 abr 2013

MATHWORKS, *Fuzzy Logic Toolbox – For use with MatLab - User's Guide – Version 2, USA – MA*:The MathWorks, Inc, 2006, 289 p.

MATSUMOTO, E. Y., *MatLab6 – Fundamentos de Programação*, 2ª edição, São Paulo: Editora Érica Ltda, 314 p., 2001

MORENO-VELO, F. J.; SANCHEZ-SOLANO, S.; BARRIGA, A.; BATURONE, I. & LÓPEZ, D. R. *An specification language for fuzzy systems*, Mathware & soft computing, pp. 1-15, 2001a, Vol. VIII, Num. 3, disponível em:< <http://upcommons.upc.edu/revistes/handle/2099/3608>>. Acesso em 15 de março de 2011.

MORENO-VELO, F. J.; SANCHEZ-SOLANO, S.; BARRIGA, A.; BATURONE, I. & LÓPEZ, D. R. "XFuzzy 3.0: A development environment for Fuzzy Systems", *In: Proc. International Conference in Fuzzy Logic and Technology*, pp. 93-96, Leicester, Sep. 2001b, disponível em <<http://www.imse-cnm.csic.es/online/2001/EUSFLAT2001.FMV.pdf>>. Acesso em 21 fev 2011.

MORENO-VELO, F. J.; SANCHEZ-SOLANO, S.; BARRIGA, A.; BATURONE, I. & LÓPEZ, D. R. "Rapid design of fuzzy systems with XFuzzy", *In: Proc. International Conference in Fuzzy Logic and Technology*, pp. 342-347, St. Louis, may. 2003, disponível em: <<http://www.imse-cnm.csic.es/online/2003/FUZZIEEE2003.FMV.pdf>>. Acesso em 21 fev. 2011.

PIÑA, A. J. B. *Síntesis de Sistemas de Control Borroso Estables por Diseño*, 2009. Tese de Doutorado. Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática. Universidad de Huelva, Spain. 308p. Disponível em: < http://dl.dropboxusercontent.com/u/1412345/Tesis-S%C3%ADntesis_de_Sistemas_de_Control_Borroso_Estables_por_Dise%C3%B1o.pdf> Acesso em 17 abr. 2013.

PIÑA, A. J. B. & MÁRQUEZ, J. M. A. *Fuzzy Logic Tools - Reference Manual v1.0*. CEA Intelligent Control Group - Universidad de Huelva, Spain. 2011. 239p. Disponível em: < http://uhu.es/antonio.barragan/FLTHelp/Fuzzy_Logic_Tools_Reference_Manual_v1.0.pdf> Acesso em 17 abr. 2013.

PONTES, F. A., ARRUDA, D. M., ABUD, G.M.D., PONTES, R. M., BARCELLOS, B. F. O., *Avaliação de ferramentas voltadas para lógica fuzzy*. Nota Técnica – INT, jul. 2013, 88 p.

ROSS, T. J. *Fuzzy logic with engineering applications*, USA: McGraw-Hill, 1995, p. 600.

SILVA, E. L. & MENEZES, E. M. *Metodologia da pesquisa e elaboração de dissertação*. Florianópolis: UFSC, 2005

SMITH, B., FREE SOFTWARE FOUNDATION, *A quick guide to GPLv3*, 2007, 6 p. Disponível em: < <http://www.gnu.org/philosophy/free-sw.en.html>>. Acesso em 2 de abril de 2013.

ZADEH, L. A. "Fuzzy sets", *Information and Control* , Vol. 8, p. 338-353, 1965.

_____, "Fuzzy Logic", *IEEE Computer Mag.*, p. 83-93, abr. 1988.

_____, "Fuzzy Logic", *Communications of the ACM* , Vol. 37 , N° 3, p. 77-84, mar. 1994.