Biblioteca para implementação de controladores utilizando lógica *fuzzy*

Antonio Carlos da Silva Barros
carlos.barros.22@gmail.com

Victor Hugo Costa de Albuquerque
victor120585@yahoo.com.br

Auzuir Ripardo de Alexandria
auzuir@cefetce.br

Resumo

O presente trabalho busca desenvolver uma biblioteca de Lógica *Fuzzy* para implementar o controle em sistemas não-lineares, onde, posteriormente, será utilizada em microcontroladores/microprocessadores para executar diversas formas de controle. A Lógica *Fuzzy* é utilizada, pois oferece uma maior facilidade de trabalho em sistemas de controle, em que nem sempre é possível realizar uma simples modelagem matemática do processo. A plataforma de desenvolvimento utilizada é a C++ *Builder* 6 da *Borland*, tendo em vista a maleabilidade que esta ferramenta oferece na programação orientada a objeto e na programação visual. São desenvolvidas as etapas de inferência de regras, fuzificação e defuzificação que constituem um Controlador *Fuzzy* baseado em Regras. A biblioteca *Fuzzy*, utilizada neste trabalho, é aplicada para realizar o controle do tempo de sinalização de um semáforo de acordo com o fluxo de carro nas avenidas, podendo ser aplicada em outras situações. Na simulação por *software*, o sistema mostra-se eficiente e eficaz para atender as exigências requeridas pelo o usuário. Consequentemente, este sistema está apto para ser implementado em um *hardware*, utilizando um microcontrolador que receberá um algoritmo de dados. Utiliza-se o *MatLab* para solucionar o mesmo problema, com isso validamos a biblioteca *Fuzzy*, pois ambas ferramentas mostram resultados semelhantes. Portanto, aperfeiçoa-se o controle do fluxo de carro e, conseqüentemente, diminui o tráfego. Esta biblioteca, baseada na lógica *Fuzzy*, pode ser aplicada em diversas áreas como, por exemplo, industriais, médicas, agrícolas e outras.

Palavras-chave: *Lógica nebulosa. Controladores fuzzy. Programação orientada a objetos.*

Abstract

The present work searchs to develop a library of Fuzzy Logic to implement the control in nonlinear systems, where, latery, it will be used in microcontrollers/microprocessors to execute several forms of control. The Fuzzy Logic is used, therefore it offers an easiness of work with control systems, and where nor always it is possible to make a simple mathematical modeling of the process. The used platform of development is the C++ Builder 6 Borland, in view of the easiness that this tool offers in the guided programming the object and the visual programming. The stages of inference of rules, fuzificação and defuzificação are developed that constitute Fuzzy Controller based in the Rules. The Fuzzy library, used in this work, is applied to carry through the control of the time of signalling of a traffic light in accordance with the flow of car in the avenues, that can to be applied in other situations. In the simulation by software, the system reveals efficient and efficient to take care of the requirements required for the using. Consequentemente, this system is apt to be implemented in the hardware, using a microcontroller that will receive an algorithm of the data. The MatLab is used to solve problem the same, with this validates the Fuzzy library, therefore both tools show resulted similar. Therefore, the control of the car flow is perfected and, consequently, it diminishes the traffic. This library, based on the Fuzzy logic, can be applied in several areas as, for example, industrials, doctors, agriculturists and others.

Keywords: *Logic fuzzy. Controllers fuzzy. Guided programming the bjects.*

1 Introdução

Sistemas de controle são utilizados na indústria, na automação predial, hospitalar, na agricultura, etc., onde, muitas vezes, são considerados como sistemas simples. Estes sistemas usam, para a automação, recursos geralmente aplicados a

esses casos que são: microcomputadores e placas de obtenção de sinais. Porém, podem-se utilizar outros controladores, que são mais complexos e que exijam uma operação de forma diferenciada, como por exemplo, sistemas de visão computacional e controle não-linear de processos. O mercado exige um grau muito alto de confiabilidade, segurança, precisão e poder de processamento de dados, para que os dispositivos de entradas e saídas dos sistemas possam efetuar suas funções de forma correta e no menor tempo (SILVEIRA, 2002).

O Controle de sistemas foi por muito tempo essencialmente baseado em controladores do tipo PID (Proporcional, Integral e Derivativo) (OGATA, 2000). Neste modelo têm-se as malhas de realimentação, em que todos os componentes são acionados de acordo com o valor de realimentação ou *feedback*. Estes sistemas eram basicamente analógicos, compostos por caixas pretas reguladas de acordo com a saída e o erro. A saída do sistema era sempre monitorada para verificar se seus valores estavam dentro dos limites especificados. Segundo Ogata (2000), este modelo usa uma abordagem matemática muito forte, tornando-se muito complexo.

Atualmente, existem outras abordagens para realizar controle em diversos processos industriais, médicos, agrícolas e outros. Com o advento da teoria da Inteligência Artificial, é possível trabalhar com sistemas especialistas, ou seja, usa-se o conhecimento do próprio operador. Geralmente, este conhecimento é baseado em redes neurais artificiais (RNA), algoritmos genéticos, lógica de primeira ordem e lógica *Fuzzy*.

A Lógica *Fuzzy*, também denominada de Lógica Difusa e Lógica Nebulosa, é mais uma alternativa de controle. Ela trata valores que muitas vezes são interpretados e representados somente por especialistas, pois estes valores podem ser imprecisos e/ou incertos. Diferentemente da Lógica Proposicional, que trabalha apenas com valores binários, falso ou verdadeiro (ZADEH, 1965).

O controle usando a Lógica *Fuzzy* é baseado em conhecimentos adquirido por um especialista, ou valores estatísticos de tabelas, como é o caso de series temporais. Estas sentenças formam a base de conhecimento para o agente controlador, formulando-as como regras. Uma das vantagens deste tipo de controlador é que, as entradas podem ser passadas muitas vezes na forma de linguagem natural (Zadeh, 1965).

O matemático Lotfi Zadeh introduz a teoria de conjuntos nebulosos, que é definido no universo de discurso "U" e é caracterizado por uma função de pertinência "μA", a qual mapeia os elementos do conjunto universo "U" para a escala [0 1]. Desta forma, a função de pertinência associa cada elemento "x" pertencente a "U" um número real "μA(x)" no intervalo [0, 1], representando o grau de possibilidade em que o elemento "x" venha a pertencer ao conjunto "A", isto é, o quanto é possível para o elemento "x" pertencer ao conjunto "A" .

Sistemas de controle baseados no conceito de Lógica Nebulosa, desenvolvidos por Lofti Zadeh em meados de 1960, têm sido utilizados com sucesso em diversas áreas, tais como: fabricação de eletrodomésticos, indústria automobilística, sistemas de auxílio à tomada de decisão, controle industrial, ciência e engenharia dos materiais, sistemas hospitalares e outros (SHAW, 1999; ALEXANDRIA, 1994; ANWAR *et al*, 1997; SILER *et al*, 1998)

A Lógica Difusa pode ser utilizada para a implementação de controladores difusos, aplicados nos mais variados tipos de processos. Segundo Ortega (2001), Siler (2005) e Bellman (1970), a utilização de regras difusas e variáveis lingüísticas conferem ao sistema de controle varias vantagens, com, por exemplo:

1) simplificação do modelo do processo;
2) melhor tratamento das imprecisões inerentes aos sensores utilizadas;
3) facilidade na especificação das regras de controle, em linguagem próxima da natural;
4) satisfação de múltiplos objetivos de controle e
5) facilidade de incorporação do conhecimento de especialistas humanos.

Existem também as desvantagens de se trabalhar com este tipo de controle, que são:

1) Necessitam de mais simulação e testes;
2) Não aprendem facilmente;
3) Dificuldades de estabelecer regras corretamente e
4) Não há uma definição matemática precisa.

Porém, os valores das leituras através de sensores dos sinais esperados pelos atuadores do sistema de controle não são difusos, sendo necessário adicionar elementos entre o controlador difuso e o processo a ser controlado. Estes elementos são denominados fuzzificador e defuzzificador, estão posicionados na entrada e saída dos sistemas de controle e, no caso da biblioteca, está sendo realizado por meio de rotinas. São atribuídos nomes iguais a todos os elementos, isto para facilitar

alterações na implementação da biblioteca proposta neste trabalho. Estes elementos são responsáveis por transformar as medidas obtidas dos sensores em conjuntos difusos (fuzzificador), e transformar os conjuntos difusos obtidos na saída do controlador em valores não difusos de controle para o processo (defuzzificador).

Com isso, torna-se mais simples ajustar um sistema usando o controlador *Fuzzy*. Diferentemente do controlador PID, em que uma pequena alteração na implementação do sistema poderá exigir um trabalho bastante oneroso de cálculos e alterações mecânicas.

O presente trabalho busca desenvolver uma biblioteca de Lógica *Fuzzy* para implementar o controle em sistemas não-lineares, onde, posteriormente, será utilizada em microcontroladores/microprocessadores podendo executar diversas formas de controle. A plataforma de desenvolvimento utilizada é a C++ *Builder* 6 da *Borland*, tendo em vista a facilidade que esta ferramenta oferece na programação orientada a objeto e, na programação visual, utiliza-se o *MatLab* para solucionar o problema em questão e comparar os resultados obtidos. São desenvolvidas as etapas de inferência de regras, fuzificação e defuzificação que constituem um Controlador *Fuzzy* baseado em Regras.

2 Construindo um controlador *Fuzzy*

São apresentadas a seguir, as etapas de construção de um Controlador Fuzzy baseado em Regras, bem como uma descrição detalhada para cada uma das etapas. Inicialmente é apresentada a declaração das variáveis de entrada e saída, em seguida estabelece-se às regras, os métodos para fuzzicar e defuzzificar a variável de entrada (AHMED, 1996; AMINZADEH *et al*, 1994; ANDERSON *et al*, 1991; BELLMAN *et al*, 1970 BAIDOSOV, 1990; BARROS, 1992, BATTISTON *et al*, 1998 e BELLAMY, 1997).

O exemplo a seguir, mostra o controle da temperatura de um ambiente, em que se utilizam apenas cinco regras, e somente duas variáveis.

Variáveis de entrada e saída

As duas variáveis abordadas são: temperatura do ambiente, que é a variável de entrada com escala [0 30], e a potência que deve ser injetada no ar-condicionado, que é a variável de saída com escala [0 3000]. Os valores das escalas de entrada e saída são valores reais.

No conjunto *Fuzzy* estas variáveis recebem o nome de variáveis lingüísticas, e que são desdobradas da seguinte forma:

- temperatura = {frio, normal, quente}
- potência = {esquentar, zero, esfriar}

As operações no conjunto dos números Fuzzy são mostradas abaixo:

1) Intersecção - E (AND): $A \cap B$

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

2) União – OU (OR): $A \cup B$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

3) Complemento – NÃO (NOT): A’

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

Figura 1: (a) operação de intersecção, (b) operação de união e (c) operação de complementação.

Função de pertinência

Nas funções de pertinência são estabelecidos os limites na escala para cada variável lingüística. Estas funções, geralmente são representadas por funções triangulares, mas também são representadas por trapézios, retângulos e funções gaussianas. O eixo das ordenadas apresenta os valores das escalas de entrada e saída, e o eixo das coordenadas o grau pertinência [μA(x)] de cada elemento que está contido em um conjunto "U".

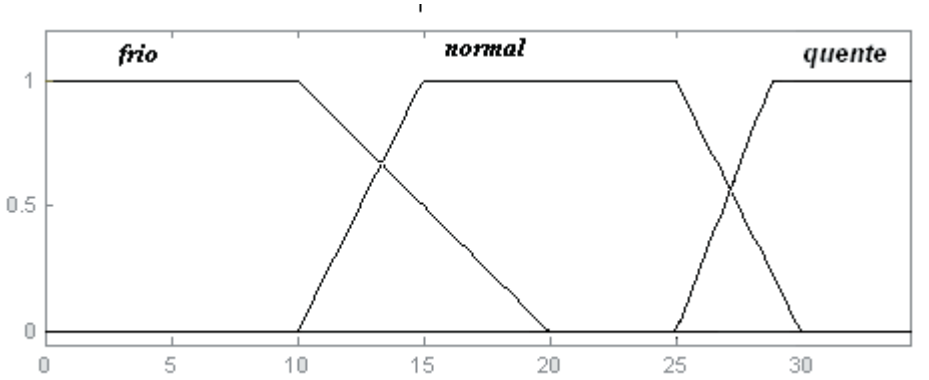

Figura 2: Funções de pertinência para a temperatura.

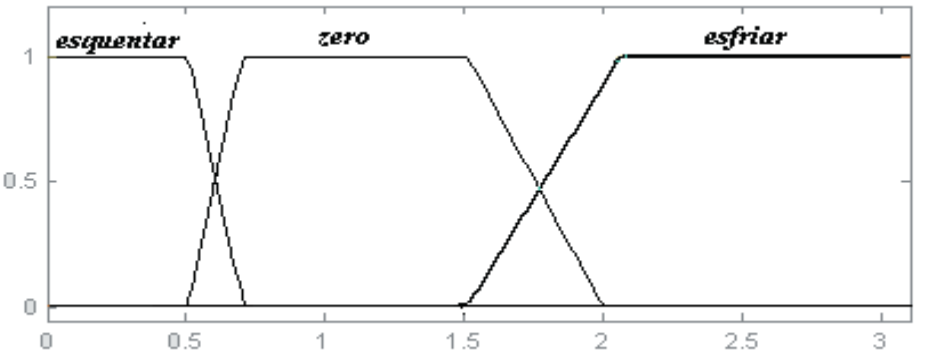

Figura 3: Funções de pertinência para a potência.

Regras (ou base de conhecimento)

Seja um condicionador de ar com processos controlados por uma base de conhecimento de Fuzzy. O sistema deve controlar a temperatura ambiente segundo as seguintes regras:

Se temperatura quente → Aumenta-se a potência

Se temperatura normal → Permanece normal

Se temperatura baixa → dimimuir potência

Fuzzificação das variáveis de entrada

Para transformar (fuzzificar) os valores de entrada do sistema em variáveis lingüísticas, é preciso entrar com os valores de entrada nas escalas de cada variável, observando em que função de pertinência este valor se encontra, e assim retornar o valor μA(x), com o qual ira se trabalhar as regras e operações de AND, OR e implicação. Para obtermos um valor fuzzificado entramos com um valor no eixo das ordenadas, e vejamos em quais funções de pertinência ele toca e então retornamos o valor no eixo das coordenadas. Este método é conhecido como função *singleton*. Onde este valor esta numa escala de 0 a 1.

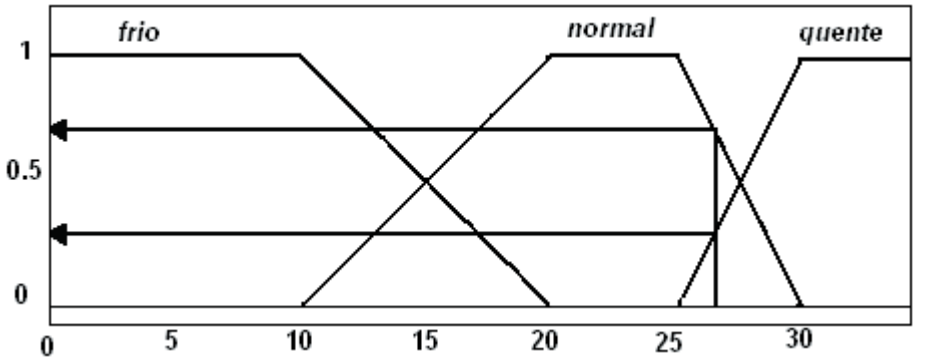

Figura 4: Fuzzificação das variáveis.

Defuzzificação das variáveis de saída

Como todas as operações são realizadas no domínio dos conjuntos *Fuzzy*, é necessário, no final do processo, retornar um valor no conjunto dos números reais. Para isto, podem ser empregados vários métodos como, por exemplo, a media dos máximos (MM), métodos das alturas. Entretanto, o método mais utilizado é o cálculo do centro de massa, O método do Centro de Área (CA) é a técnica de defuzificação mais comumente usada. Ele também é citado na literatura como método do Centro de Gravidade ou do Centróide segundo (Klir *et al*, 1995 e Yen *et al* 1999). Diferentemente do MM, a técnica do Centro de Área para calcular o valor clássico representativo considera toda a distribuição de possibilidade de saída do modelo. O procedimento é similar ao usado para calcular o centro de gravidade em física, se consideramos a função de pertinência μA(x) como a densidade de massa de x. Por outro lado, o método do Centro de Área pode ser compreendido como uma média ponderada, onde μA(x) funciona como o peso do valor x. Se x é discreto, então a defuzificação dos valores do conjunto *F*uzzy A é dada por:

$$yo = \frac{\sum x\mu A(x)x}{\sum x\mu A(x)} \tag{1}$$

Da mesma forma, se x é continuo, então,

$$yo = \frac{\int \mu A(x)xdx}{\int \mu A(x)dx} \tag{2}$$

A Figura 5 explica o método de defuzificação do CA. A Principal desvantagem desse método é o seu custo computacional, principalmente no caso em que x é continuo.

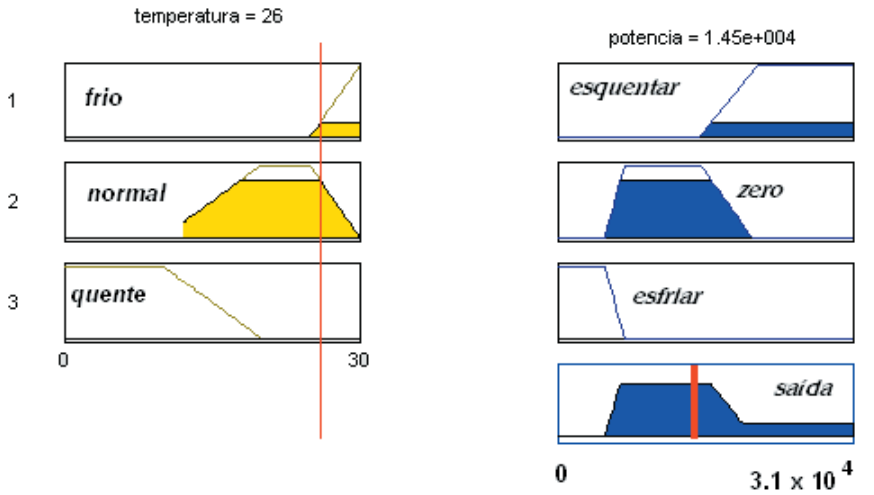

Figura 5: Exemplo do método de defuzzificação CA.

Simulação do problema

O problema pode ser verificado usando qualquer *software* de controle *Fuzzy*. Para este exemplo, utiliza-se a biblioteca de classes implementada em C++ *Builder* 6 da *Borland*, proposta neste trabalho.

A elaboração deste exemplo é apenas a titulo de entendimento, da forma de construção de um controlador Fuzzy.

3 Aplicação da biblioteca *Fuzzy* no controle do tempo de semáforos no cruzamento de duas avenidas.

A seguir é mostrada a aplicação realizada pela biblioteca. As declarações das variáveis, base de conhecimento, funções de pertinência e fase de teste. Todas essas etapas são descritas na aplicação desenvolvida para testar as respostas da biblioteca. Realizando o controle do tempo de semáforos no cruzamento de duas avenidas.

O código é desenvolvido no propósito de ser embarcado em sistemas microprocessados. É devido a este fato, que são feitas algumas restrições na maneira de se declarar as variáveis e funções de pertinência.

O algoritmo para a aplicação dessa biblioteca, segue os seguintes estágios:

Estágio 1: são definidas as escalas das variáveis de entrada e saída.

• Variáveis de entrada: [0, 20]

Densidade de tráfego da avenida A→ Da.
Densidade de tráfego da avenida B → Db
Código: // declaração das variáveis *F*uzzy de entrada
var_ *F*uzzy dA_P, dA_MEP, dA_ME, dA_MEG, dA_G;
var_ *F*uzzy dB_P, dB_MEP, dB_ME, dB_MEG, dB_G;

• Variáveis de saída: [10, 50]

Tempo de abertura do sinal verde em segundos A → Ta
Tempo de abertura do sinal verde em segundos B → Tb
Código: // declaração das variáveis *F*uzzy de saída
var_ *F*uzzy tA_P, tA_MEP, tA_ME, tA_MEG, tA_G;
var_ *F*uzzy tB_P, tB_MEP, tB_ME, tB_MEG, tB_G;
var_ *F*uzzy tA, tB;

Então, para este problema, temos duas variáveis lingüísticas, que estão representadas no conjunto dos números *F*uzzy da seguinte forma:

• densidade de tráfego nas avenidas = { P, MD, A, GR} e
• tempo de abertura do sinal verde em segundos = { P, MD, A, GR}.

Estágio 2: as funções de pertinência têm formas triangulares, conforme apresentado nas Figuras 6 e 7.

Código: // tA_P.fper = new fp(0,10,20,0,60,1);

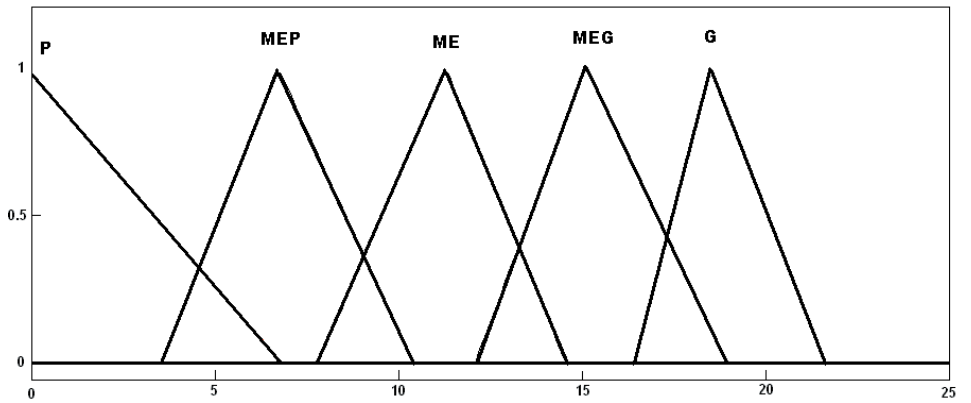

Figura 6: Funções de pertinência para a variável lingüística entrada.

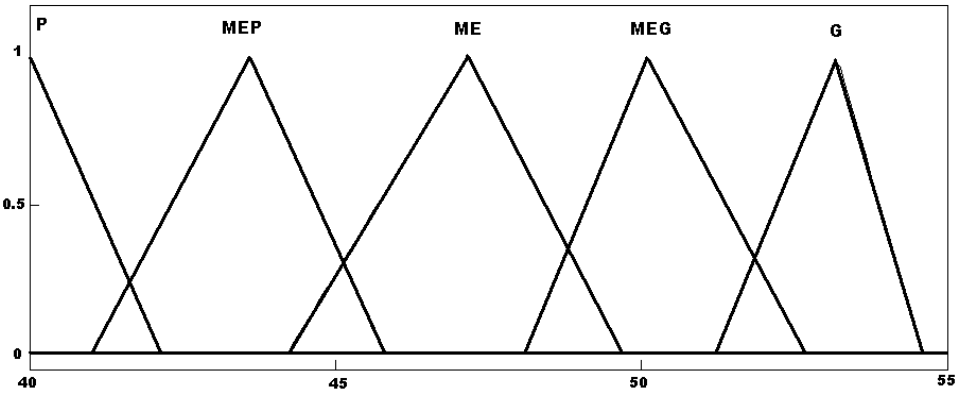

Figura 7: Funções de pertinência para a variável lingüística saída.

Estágio 3:as regras elaboradas para o controle são definas nas Tabelas 1 e 2.
Código: //Regra_tA1 = Regra_tA1.implica(Regra_tA1.e(dA_P, dB_P), tA_P);

Tabela 1: Conjunto de regras de inferência para Avenida A

Da	**Db**				
	P	MEP	ME	MEG	G
P	P	P	P	P	P
MEP	MEP	MEP	MEP	MEP	MEP
ME	MEG	MEG	ME	MEP	MEP
MEG	MEG	MEG	MEG	ME	ME
G	G	G	MEG	ME	ME

Tabela 2: Conjunto de regras de inferência para Avenida B

Da	***Db***				
	P	MEP	ME	MEG	G
P	P	MEP	MEG	MEG	G
MEP	P	MEP	MEG	MEG	G
ME	P	MEP	ME	MEG	MEG
MEG	P	MEP	MEP	ME	ME
G	P	MEP	MEP	ME	ME

Estágio 4: as Figuras 7 e 8 mostram a interface do sistema proposto desenvolvido no ambiente C++ *Builder* da *Borland* e no *MatLab*, respectivamente.

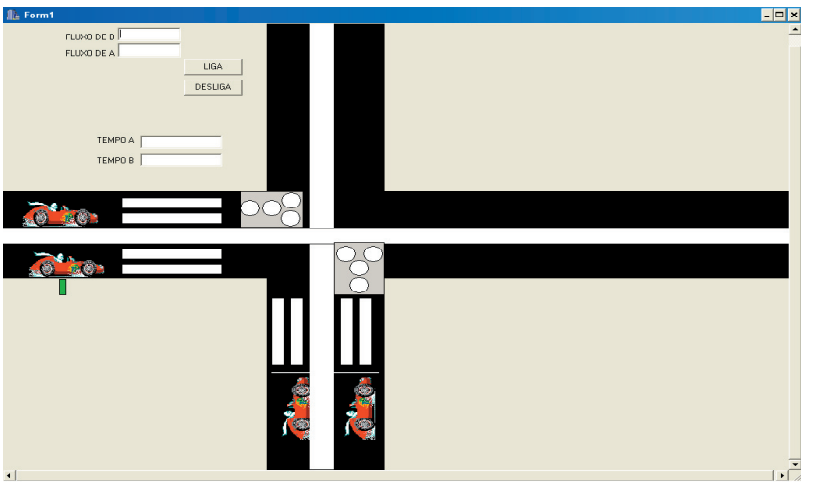

Figura 7: Ambiente de teste.

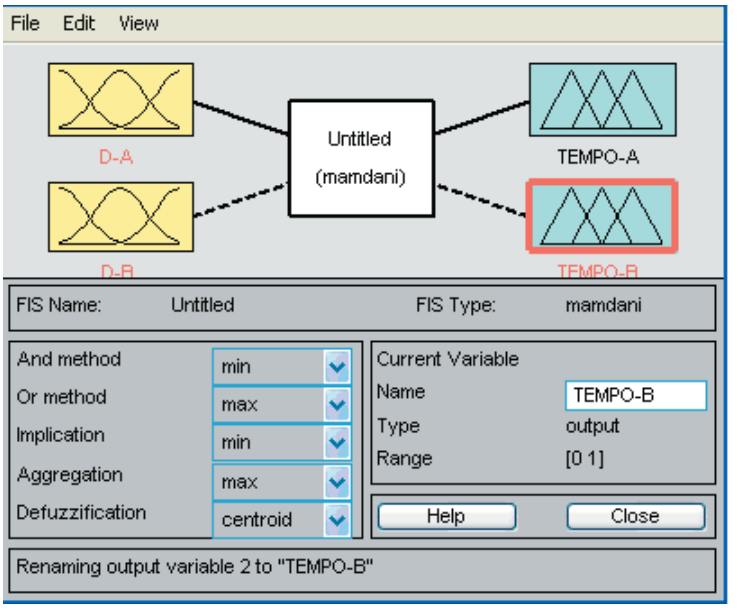

Figura 8: Ambiente de teste no *MatLab* 7.5.

4 Resultados e discussão

Com base nos resultados obtidos através dos testes realizados utilizando o Modelo Madami, pode-se afirmar que o algoritmo desenvolvido é capaz de revelar concordâncias entre os resultados apresentados, quando comparado com outros *softwares* comercias como, por exemplo, o *Toolbox* do *Matlab* e *FUZZYTECH*.

O ambiente de programação C++ *Builder* 6 da *Borland* mostra-se eficiente e eficaz para o controle de sistemas que utilizam algoritmos baseados na Lógica Fuzy, pois oferece respostas rápidas e precisas para a aplicação desejada.

O tempo de abertura dos semáforos é adequado para as condições de trafego simuladas, bem como para as regras estabelecidas, que realiza o controle de trafego das avenidas de forma correta. Desta forma, consegue-se reduz o tráfego de veículos em horários de *rush*.

5 Conclusão

Conclui-se, portanto, que a biblioteca *F*uzzy pode ser utilizada normalmente para executar o controle de sistemas, pois além de versátil, a biblioteca apresenta resultados precisos em um curto espaço de tempo. Além do mais, ela oferece aos estudantes de engenharia e engenheiros, que realizam o controle de sistemas, mais uma opção, que é a Lógica Fuzzy

A locação das variáveis, das regras e das funções de pertinência no escopo do algoritmo de programação, são facilmente implementadas no C++ *Builder*.

A biblioteca pode ser usada para realizar qualquer controle de sistemas que não pode ser modelado de uma maneira simples, onde é necessária uma abordagem de uma lógica diferente da proposicional.

Referências

AHMED, E. Fuzzy cellular automata models in immunology. *Journal of Statistical Physics*, London, v. 85, n°. (1/2), p. AHMED, E. Fuzzy cellular automata models in immunology. Journal of Statistical Physics, London, v. 85, n.1/2, p. 291-293, 1996.

ALEXANDRIA, A. R. Desenvolvimento de um controlador PID com sintonia fuzzy. In: ENCONTRO DE PESQUISA E PÓS-GRADUAÇÃO DO CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO CEARÁ, 3., Fortaleza, 2003. *Anais*... Fortaleza, CEFET-CE, 2003. 1CD-ROM.

AMINZADEH, F.; JAMSHIDI, M. *Soft computing:* fuzzy logic, neural networks and distributed artificial intelligence. New York: Prentice Hall, 1994.

ANDERSON, R. M.; MAY, R. M. *Infectious diseases of humans:* dinamics and control. Oxford: Oxford University Press, 1991.

ANWAR, T.; IGOR, B. Fuzzy classification system for psychiatry. In: IFSA WORLD CONGRESS, 7., Prague, 1997 *Proceedings*... Prague: IFSA, 1997. 1 CD-ROM.

BAIDOSOV, V. A. Fuzzy diferential inclusions. *PMM Journal of Applied Mathematics and Mechanics*, New York, v. 54, n. 1, p. 8-13, 1990.

BARROS, L. C. Modelos determinísticos com parâmetros subjetivos. 1992. 217 f. Dissertação (Mestrado em Matemática)-Instituto de Matemática, Estatística e Ciência da Computação, Universidade de Campinas, Campinas, SP, 1992.

BATTISTON, F. M. et al. Fuzzy controlled feedback applied to a combined scanning tunneling and force microscope. *Applied Physics Letters*, Argonne, v. 72, n. 1, p. 25-27, 1998.

BELLAMY, J. E. Medical diagnosis, diagnostic spaces and fuzzy systems. *Journal of American Veterinarian Medicine Association*, Schaumburg, v. 210, n. 3, p. 390-396, 1997.

BELLMAN, R.; ZADEH, L. A. Decision-making in a fuzzy environment. *Management Science*, Evanston , v. 17, p. 141-154, 1970.

KLIR, G.; YUAN, B. *Fuzzy sets and fuzzy logic:* theory and applications. New York: Prentice Hall, 1995.

OGATA, K. *Engenharia de controle moderno.* 3. ed. Rio de Janeiro: LTC, 2000.

ORTEGA, N. R. S. *Aplicação da teoria de conjuntos fuzzy a problemas da biomedicina.* 2001. 342 f.

Tese (Doutorado em Física)-Instituto de Física, Universidade de São Paulo, São Paulo 2001.

SHAW, I. S.; SIMÕES, M. G. *Controle e modelagem fuzzy.* São Paulo: Edgard Blücher, 1999.

SILER, W.; BUCLEY, J. J. *Fuzzy expert systems and fuzzy reasoning.* New York: John Wiley and Sons, 2005.

SILVEIRA, P. R. *Automação e controle discreto.* São Paulo: Érica, 2002.

YEN, J.; LANGARI, R. *Fuzzy logic:* intelligence, control, and information. New York: Prentice Hall, 1999.

ZADEH, L.A. Fuzzy sets. *Information and Control*, Berkley, v. l, n. 8, p. 338-353, 1965.

SOBRE OS AUTORES

Antonio Carlos da Silva Barros
Tecnólogo em Mecatrônica-Mecânica pelo o Centro Federal de Educação Tecnológica do Ceará – CEFET-CE Mestrando em Engenharia de Teleinformática pela Universidade Federal do Ceará – UFC, na área de Inteligência Artificial e Sistemas de Controle.

Victor Hugo Costa de Albuquerque
Tecnólogo em Mecatrônica-Mecânica pelo Centro Federal de Educação Tecnológica do Ceará – CEFET-CE. Mestrando em Engenharia de Teleinformática pela Universidade Federal do Ceará – UFC, na área de Processamento digital de imagens e reconhecimento de padrões.

Auzuir Ripardo de Alexandria
Engenheiro Elétrico e Bacharel em Ciências da Computação pela Universidade Federal da Paraíba - UFPB, mestre e doutorando na Universidade Federal do Ceará - UFC no curso de Engenharia de Teleinformática, na área de Processamento digital de imagens. Professor Titular da área da indústria do Centro Federal de Educação Tecnológica do Ceará – CEFET-CE.