

# Um Controlador Nebuloso Aplicado ao Problema do Pêndulo Invertido

**Marcelo Henrique Essado de Moraes, Amita Muralikrishna**

Programa de Pós-Graduação em Computação Aplicada, CAP/INPE,

12201-970 – São José dos Campos, SP

E-mail: marceloessado@gmail.com, amita@dae.inpe.br

**Ricardo Bravo, Lamartine Nogueira Frutuoso Guimarães**

Instituto de Estudos Avançados – IEAv/CTA

12228-840 – São José dos Campos, SP

E-mail: rbravo@ieav.cta.br, guimarae@ieav.cta.br

## 1. Introdução

A teoria de controles automáticos está cada vez mais integrada a área da computação. Cada vez mais, novos algoritmos e recursos computacionais estão sendo utilizados a fim de realizar os mais diversos tipos de controle. Os sistemas de controle automático são encontrados em abundância em setores da indústria [7], tais como controle de qualidade e fabricação de produtos, linha de montagem automática, controle de ferramentas, tecnologia espacial e de armamento, sistemas de transporte, sistemas de potência, robôs e muitos outros. Dentro deste contexto, o presente trabalho é um estudo da construção de um controlador nebuloso, devido sua característica de ação podendo, inclusive ser descrita de modo qualitativo e facilmente implementada por um conjunto de regras. A teoria de sistemas baseadas em lógica fuzzy proporciona a possibilidade de supervisão inteligente, baseada apenas em informações qualitativas sobre a operação do sistema.

Este trabalho descreve a implementação de um simulador para o pêndulo invertido. O simulador permite utilizar um controlador nebuloso e compará-lo a um controlador convencional que utiliza realimentação de estados. A implementação deste trabalho foi feita baseada no software MATLAB® - The MathWorks, Inc.

## 2. Modelo do Pêndulo Invertido

O Pêndulo Invertido é um processo mecânico absolutamente instável, ou seja, está sujeito a cair em qualquer direção a menos que uma força adequada seja aplicada ao carro de suporte da base do pêndulo, acelerando-o em uma determinada direção.

A Figura 1 ilustra o sistema do pêndulo invertido modelado neste trabalho. Como descrito anteriormente, o problema do pêndulo invertido consiste em equilibrar um bastão (pêndulo) na posição vertical, fazendo com que seu ângulo ( $\theta$ ) esteja em zero graus.

A modelagem deste problema é feita em duas dimensões, desta forma, a liberdade de movimento é paralela ao eixo  $x$  de um plano cartesiano. Já o pêndulo tem um movimento angular sobre o plano  $xOy$  em torno do ponto que é fixado sobre o carrinho. A atuação no sistema para o controle de

equilíbrio vertical do pêndulo, será feita através da aplicação

de uma força  $\vec{F}$  no carrinho. Convencionou-se ainda uma força contrária ao movimento do carrinho como sendo a força de viscosidade do meio.

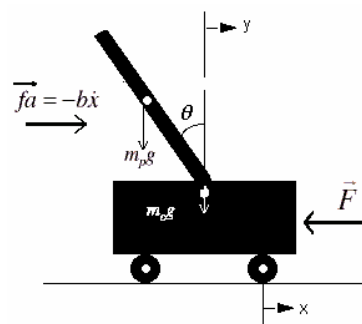


Figura 1: Sistema do pêndulo invertido

A seguir será descrita a modelagem dinâmica do sistema, através de mecânica analítica. Para tanto, serão utilizados os símbolos descritos na Tabela 1.

Símbolo	Descrição	Valor
$l$	Metade do comprimento da haste do pêndulo (centro de massa)	0,3 m
$m_p$	Massa da haste	0,5 Kg
$m_c$	Massa do carrinho	0,5 Kg
$b$	Coefficiente da força viscosa	0,1 Ns/m
$g$	Aceleração da gravidade	9,8 m/s <sup>2</sup>
$q$	Ângulo do pêndulo	
$w$	Velocidade angular do pêndulo	Rad/s
$I$	Momento de inércia do pêndulo	0,006 Kg/m <sup>2</sup>
$x$	Posição do carrinho	m
$t$	Instante de tempo	s
$v$	Velocidade do carrinho	m/s <sup>2</sup>

Tabela 1: Relação de símbolos de variáveis e constantes

Para a modelagem, utilizou-se a equação de Euler-Lagrange, dada abaixo:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = \sum F_i$$

$$L = T - V$$

$$R = \frac{1}{2} b \dot{x}^2$$

onde,  $L$  é a equação para o sistema dissipativo dada por: (a)  $R$  é o potencial de Raily, que é a força viscosa do meio, (b)  $T$  é a energia cinética e (c)  $V$  que é a energia potencial. Usando o método de Lagrange [6] chega-se a seguinte equação do movimento [1].

$$\frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \mathbf{w} \\ x \\ v \end{pmatrix} = A \begin{pmatrix} \mathbf{q} \\ \mathbf{w} \\ x \\ v \end{pmatrix} \text{vet} \quad (2)$$

onde,

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(m_p l)^2 \text{sen} \mathbf{q} \cos \mathbf{q} \mathbf{w}}{\det} & 0 & \frac{m_p l b \cos \mathbf{q}}{\det} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{(m_p l^2 + I) m_p l \text{sen} \mathbf{q} \mathbf{w}}{\det} & 0 & -\frac{b(m_p l^2 + I)}{\det} \end{pmatrix} \quad (3)$$

$$\text{vet} = \begin{pmatrix} -\frac{m_p l \cos \mathbf{q} F}{\det} + \frac{0}{\det} \\ \frac{(m_p l^2 + I) F - \frac{0}{\det} \text{gsen} \mathbf{q} \cos \mathbf{q}}{\det} \end{pmatrix} \quad (4)$$

$$\det = (m_c + m_p)(m_p l^2 + I) - (m_p l \cos \mathbf{q})^2 \quad (5)$$

e  $F$  é a força restauradora de controle.

### 3. Controlador Nebuloso

O projeto de sistemas de controle fuzzy é baseado em métodos empíricos e é basicamente uma aproximação metódica para tentativa-e-erro. O processo em geral segue os seguintes passos:

- Especificam-se as variáveis de entrada e saída utilizadas
- Define-se o conjunto de regras
- Determina-se o método de defuzzificação
- Testes para verificação do sistema, ajustando os detalhes conforme o propósito inicial

Um controlador fuzzy pode ser representado pela Figura 2, onde, medidas do ambiente (variáveis de entradas) passam

por um processo de fuzzyficação a fim de serem representadas por variáveis fuzzy. Através de um sistema de inferência, a partir de uma base de regras é produzido um valor de saída, representado por uma variável fuzzy de saída. O valor é então defuzzificado a fim de ser representado por um valor preciso a ser fornecido para o ambiente.

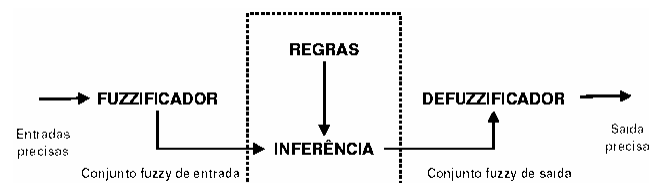


Figura 2: Diagrama de um controlador nebuloso

### 3.1 Variáveis de Entrada e Saída

No projeto de um controlador nebuloso devem ser determinadas as variáveis relevantes para o sistema e que podem ser medidas (entradas) ou controladas (saídas). Uma vez determinado um conjunto satisfatório de variáveis, torna-se necessário o conhecimento dos intervalos razoáveis, dadas as funções de pertencimento para valores que poderão ser assumidos por estas.

O controlador necessita que estas funções de pertencimento para cada variável sejam definidas, o que permite a fuzzificação ou defuzzificação dos valores. Para tanto, as entradas e saídas devem ser relacionadas através de regras de implicação.

As variáveis de entrada que podem ser medidas e as variáveis de saída que podem ser controladas e que são relevantes para os testes do sistema são:

- Erro do ângulo: varia entre  $0$  e  $\pi$
- Derivada da variação do ângulo (velocidade angular): assume-se os mesmos limites de variação do erro do ângulo
- Integral do erro do ângulo
- Erro da posição horizontal: arbitrariamente escolhido entre  $-100$  e  $+100$
- Derivada da posição (velocidade)
- Tempo: não é fuzzificado, relevante somente na discretização
- Força horizontal aplicada (saída)

### 3.2 Funções de Pertencimento

Os processos de fuzzificação e defuzzificação partem da definição de funções de pertencimento para as variáveis de entrada e saída respectivamente.

Para o controlador em questão são usadas funções triangulares e trapezoidais. Para a força horizontal foram definidas cinco gradações para a função de pertencimento. O intervalo foi dividido simetricamente em cinco gradações; negativo, pouco negativo, zero, pouco positivo e positivo. Ainda, para efeitos de testes, foram consideradas outras duas gradações complementares, a negativa e positiva, que cobrem toda a área negativa e positiva, respectivamente.

O intervalo escolhido para o erro do é dividido simetricamente em três intervalos e mais dois correspondentes aos extremos, conforme explicado para força.

No caso da derivada do erro do ângulo foram assumidos os mesmos limites, mas as funções de pertencimento do negativo e positivo foram tomadas como trapézios com sua parte triangular concentrada em torno do zero. A grande área na qual a função satura-se no 1 é devido a não se desejar grandes velocidades angulares.

Para a posição foram criadas apenas duas gradações, representando o negativo e positivo, no universo de discurso de -100 a +100.

Já a derivada e integral da posição não entraram em nenhuma regra e por isso não foram criadas gradações para tais medições. A integral do erro do ângulo teve universo de discurso assumido igual ao da derivada do erro do ângulo, o que é boa escolha para valores de integral não divergente. As figuras abaixo relacionadas definem as funções de pertencimento citadas

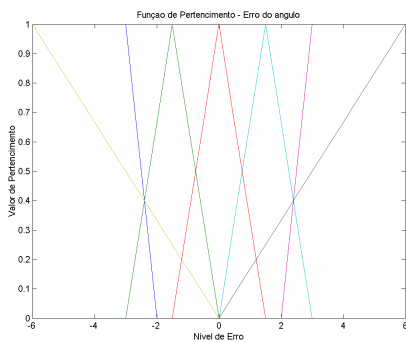


Figura 3: Função de Pertencimento do erro do ângulo

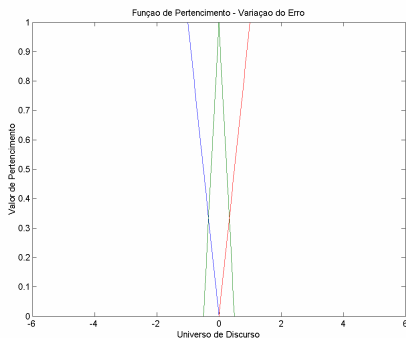


Figura 4: Função de Pertencimento para a derivada do erro do ângulo

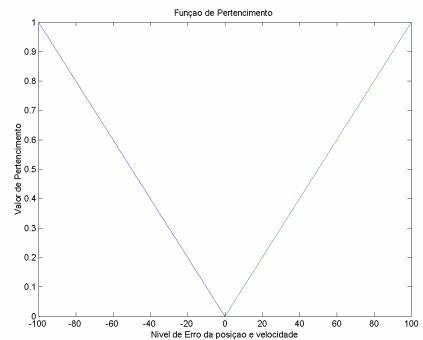


Figura 6: Função de Pertencimento para o erro da posição e velocidade do carrinho

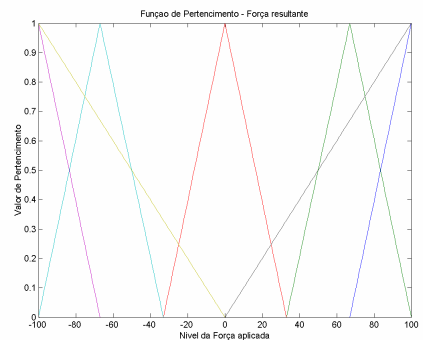


Figura 5: Função de Pertencimento para força horizontal

### 3.3 Regra de Implicação

O operador de implicação nada mais é do que a interpretação das regras, fazendo-se necessário uma avaliação, tanto do lado esquerdo (LHS), quanto do lado direito (RHS), na qual a resposta à entrada é a própria implicação. A escolha da regra de implicação é um parâmetro importante, do ponto de vista que é nessa fase que as variáveis linguísticas são concebidas, conseqüentemente a interpretação *AND*, *OR*, e *ELSE*. O operador de implicação utilizado foi o Produto Larsen, do qual avalia o produto dos valores de pertencimento obtidos e pela interpretação do *ELSE* das regras pelo processo de inferência *OR* [9], respondeu perfeitamente ao problema do pêndulo invertido. A equação abaixo mostra o produto dos valores de pertencimento:

$$f_p[m_A(x), m_B(y)] \equiv m_A(x) \bullet m_B(y) \quad (6)$$

onde,  $f_p[m_A(x), m_B(y)]$  é a relação de implicação entre as regras aplicadas ao operador Larsen.

De maneira didática a Tabela 2 descreve as regras referentes aos parâmetros citados no item anterior.

Regra	Descrição
1	if erroAngulo is <b>Muito Negativo</b> and derivadaAngulo is <b>Negativo</b> then força is <b>Muito Baixa</b>

2	<i>if</i> erroAngulo <i>is</i> <b>Muito Negativo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Zero</b> <i>then</i> força <i>is</i> <b>Muito Baixa</b>
3	<i>if</i> erroAngulo <i>is</i> <b>Muito Negativo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Positivo</b> <i>then</i> força <i>is</i> <b>Baixa</b>
4	<i>if</i> erroAngulo <i>is</i> <b>Pouco Negativo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Negativo</b> <i>then</i> força <i>is</i> <b>Baixa</b>
5	<i>if</i> erroAngulo <i>is</i> <b>Pouco Negativo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Zero</b> <i>then</i> força <i>is</i> <b>Baixa</b>
6	<i>if</i> erroAngulo <i>is</i> <b>Pouco Negativo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Positivo</b> <i>then</i> força <i>is</i> <b>Zero</b>
7	<i>if</i> erroAngulo <i>is</i> <b>Zero</b> <i>and</i> derivadaAngulo <i>is</i> <b>Negativo</b> <i>then</i> força <i>is</i> <b>Zero</b>
8	<i>if</i> erroAngulo <i>is</i> <b>Zero</b> <i>and</i> derivadaAngulo <i>is</i> <b>Zero</b> <i>then</i> força <i>is</i> <b>Zero</b>
9	<i>if</i> erroAngulo <i>is</i> <b>Zero</b> <i>and</i> derivadaAngulo <i>is</i> <b>Positivo</b> <i>then</i> força <i>is</i> <b>Zero</b>
10	<i>if</i> erroAngulo <i>is</i> <b>Pouco Positivo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Negativo</b> <i>then</i> força <i>is</i> <b>Zero</b>
11	<i>if</i> erroAngulo <i>is</i> <b>Pouco Positivo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Zero</b> <i>then</i> força <i>is</i> <b>Alta</b>
12	<i>if</i> erroAngulo <i>is</i> <b>Pouco Positivo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Positivo</b> <i>then</i> força <i>is</i> <b>Alta</b>
13	<i>if</i> erroAngulo <i>is</i> <b>Muito Positivo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Negativo</b> <i>then</i> força <i>is</i> <b>Alta</b>
14	<i>if</i> erroAngulo <i>is</i> <b>Muito Positivo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Zero</b> <i>then</i> força <i>is</i> <b>Muito Alta</b>
15	<i>if</i> erroAngulo <i>is</i> <b>Muito Positivo</b> <i>and</i> derivadaAngulo <i>is</i> <b>Negativo</b> <i>then</i> força <i>is</i> <b>Muito Alta</b>
16	<i>if</i> erroAngulo <i>is</i> <b>erroNegativo</b> <i>then</i> força <i>is</i> <b>forcaBaixa</b>
17	<i>if</i> erroAngulo <i>is</i> <b>erroPositivo</b> <i>then</i> força <i>is</i> <b>forcaAlta</b>
18	<i>If</i> velocidade <i>is</i> <b>Negativa</b> <i>then</i> força <i>is</i> <b>Alta</b>
19	<i>If</i> velocidade <i>is</i> <b>Positiva</b> <i>then</i> força <i>is</i> <b>Baixa</b>
20	<i>If</i> posição <i>is</i> <b>Negativa</b> <i>then</i> força <i>is</i> <b>Alta</b>
21	<i>If</i> posição <i>is</i> <b>Positiva</b> <i>then</i> força <i>is</i> <b>Baixa</b>

Tabela 2: Regras descritas pelas funções de pertencimento

### 3.4 Métodos de defuzzificação

Os métodos de defuzzificação utilizados neste trabalho consistem em encontrar o centro de massa do conjunto fuzzy de saída, conhecido como Centro de Áreas (*Center of Área – COA*) e Centro de Somas (*Center of Sums – COS*). O método de COA generalizado para valores discretos é definido como:

$$u = \frac{\sum_{i=1}^N m_{SAIDA}(u_i)}{\sum_{i=1}^N m_{SAIDA}(u_i)} \quad (7)$$

onde,  $u_i$  é a resultante dos valores do universo de discurso para  $i = \{1, 2, 3, \dots, N\}$ .

O método de COS generalizado para valores discretos assume a seguinte equação:

$$u = \frac{\sum_{i=1}^N u_i \sum_{k=1}^n m_{B_k}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n m_{B_k}(u_i)} \quad (8)$$

onde,  $m_{B_k}(u_i)$  é a função de pertencimento, no ponto  $u_i$  do universo de discurso, resultante da  $k$ -ésima regra ativada.

## 4. Resultados

Os resultados obtidos têm por finalidade avaliar o desempenho, bem como a robustez de um controlador nebuloso, onde, através da variação de parâmetros na implementação o objetivo do trabalho fora atingido: equilibrar o pêndulo para  $q = 0$  (ver Figura 1) e estabilizar o carrinho numa posição desejada, comparando-o ainda com um método de controle clássico, o controlador PID (Proporcional Integral e Derivativo). Para a realização dos testes foi utilizado um Pentium II, com capacidade de processamento de 800 MHz e 264 MB.

Através das regras da Tabela 2 e o método de implicação citado, a Figura 6 mostra a resposta do controlador através do método de defuzzificação COS, com todas as variáveis de estado chegando a zero para 200 segundos de simulação e um tempo computacional de 668 segundos. Com o método de defuzzificação COA aplicado aos parâmetros anteriores chega-se a valores próximos aos obtidos pelo método COS, porém com uma menor precisão, como mostra a Figura 7. O sistema ainda apresentou maiores oscilações, chegando a um gasto computacional de 3943 segundos, deixando explícito uma dificuldade maior da computação dos dados.

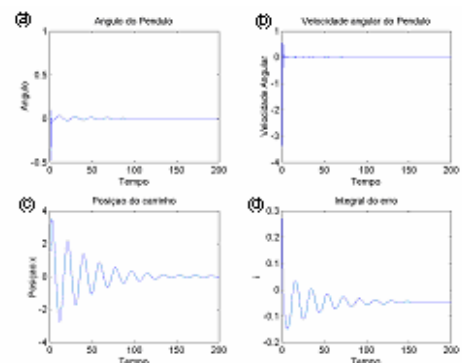


Figura 6: Larsen e COS. (a) ângulo do pêndulo; (b) velocidade angular; (c) posição do carrinho; (d) integral do erro

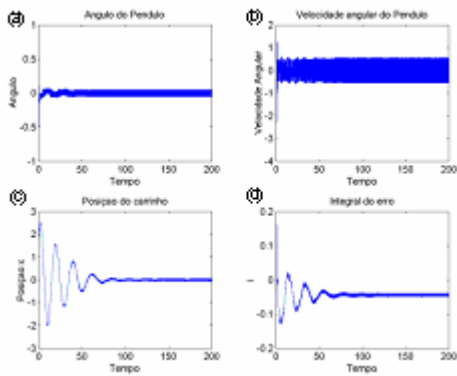


Figura 7: Larsen e COA. (a) ângulo do pêndulo; (b) velocidade angular; (c) posição do carrinho; (d) integral do erro

#### 4.1 Comparação com PID

Pôde-se observar que o Produto Larsen, principalmente o método de defuzzificação COS teve um resultado satisfatório, tanto para fins de simulação, como de comparação, obtendo êxito na resposta do sistema, ou seja, não só na velocidade e posição do ângulo, como também a posição do carrinho e sua velocidade se mantiveram estáveis, assumindo valores nulos (zeros).

Nos testes realizados com o controle PID [1], através de várias simulações utilizando os mesmos parâmetros de controle e também baseados na tentativa-e-erro, obteve-se um bom conjunto de ganhos do qual o ângulo do pêndulo assumia o valor zero e a velocidade do carrinho um valor muito próximo de zero, como mostra a Figura 8, porém, tais resultados se mostram inferiores aqueles conseguidos com o controle fuzzy, inclusive os valores para o PID deu-se com um tempo de simulação de 1000 segundos, ainda com um deslocamento do carrinho superior ao do controlador fuzzy, ficando clara a distinção entre as duas técnicas de controle.

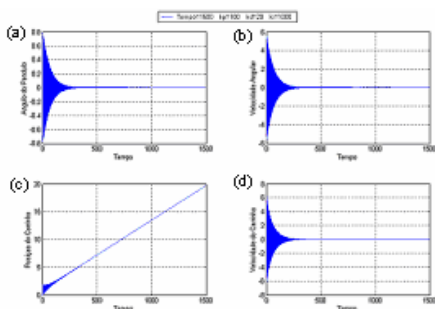


Figura 8: Resposta do controle PID. (a) ângulo do pêndulo, (b) velocidade angular, (c) posição do carrinho, (d) velocidade do carrinho.

#### 5. Conclusão

Neste trabalho foram apresentados resultados obtidos através da avaliação do controlador fuzzy para o problema do pêndulo invertido, comparando-o com os resultados do controlador PID [1]. Todos os experimentos foram realizados utilizando uma formulação matemática da dinâmica de comportamento do pêndulo descritas na sessão 2.

O desenvolvimento de controladores para sistemas dinâmicos não-lineares compreende um desafio [7]. Quando procura-se controlar tal sistema, a especificação de um controlador Fuzzy leva uma vantagem considerável, pois tem uma abordagem mais intuitiva, baseada em variáveis e regras lingüísticas. Todavia quando a complexidade do problema aumenta, sua dimensão cresce proporcionalmente, dificultando, de certa forma a definição de regras fuzzy.

As maiores dificuldades encontradas estão relacionadas com a estimação dos parâmetros dos controles, devido a dimensão do sistema. Sendo assim, sugere-se o uso de técnicas de estimação de parâmetros ou abordagem de problemas inversos, bem como a aplicação de mais variáveis de estado no caso do PID.

Este sistema, da maneira que se apresenta pode ser aplicado a outros domínios de conhecimento, desde que sejam modelados com quatro entradas e uma saída.

#### 6. Referências

- [1] M. H. E. Morais, A. Muralikrishna, R. Bravo, Controladores PID e Nebuloso para o Pêndulo Invertido, Projeto Final, INPE, 2004.
- [2] D. M. Etter, Engineering problem solving with MATLAB, Editora Prentice-Hall, 1993.
- [3] J. W. Hines, Matlab Supplement to Fuzzy and Neural Approaches in Engineering, Editora Jhon Wiley and Sons, New York, 1997.
- [4] L. T. Hoffman, Controlador Nebuloso para o Pêndulo Invertido, Projeto Final, INPE, 2003.
- [5] B. C. Kuo, Sistemas de Controle Automático, Editora Prentice-Hall do Brasil, 1982.
- [6] J. W. Leech, Mecânica Analítica, traduzido por Carlos Campos de Oliveira, Universidade do Estado da Guanabara, Rio de Janeiro, 1971.
- [7] K. Ogata, Modern Control Engineering, University of Minnesota, Prentice-Hall, Inc, 1970.
- [8] R. J. Stonier, A. J. Stacy, Learning Fuzzy Laws for the Inverted Pendulum, Central Qld University, Australia, 1998.
- [9] L.H. Tsoukalas, R. E. Uhrig, Fuzzy and Neural Approaches in Engineering, Editora Jhon Wiley and Sons, Inc, New York, 1997.