

# Sistema de Controle PID e Nebuloso Aplicado ao Problema do Pêndulo Invertido

Marcelo H. Essado de Moraes  
Programa de Pós-graduação em Computação Aplicada (CAP)  
Aluno de Matéria Isolada  
marceloessado@gmail.com

Amita Muralikrishna  
Programa de Pós-graduação em Computação Aplicada (CAP)  
amita.mk@lac.inpe.br

Lamartine Nogueira Frutuoso Guimarães  
Instituto de Estudos Avançados IEAv/CTA - Divisão de Energia Nuclear  
guimarae@ieav.cta.br

## Resumo

Este artigo tem por finalidade o estudo da construção e comparação de dois controladores: o PID e o Nebuloso. O primeiro por sua funcionalidade, devido sua enorme utilização e simplicidade de implementação. O segundo por proporcionar a possibilidade de supervisão inteligente baseada apenas em informações qualitativas sobre a operação do sistema. Ambos são aplicados ao sistema do pêndulo invertido, o qual reproduz o problema do movimento de equilíbrio de um bastão sobre uma base móvel.

**Palavras-chave:** controle Nebuloso, controle PID, controle automático, inteligência artificial, lógica fuzzy.

## 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, linha de montagem automática, 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 PID e nebuloso aplicado ao problema do pêndulo invertido, cujo objetivo é equilibrar o pêndulo na posição vertical através do ajuste de parâmetros físicos. O sistema pêndulo invertido é constituído por uma base móvel sobre a qual existe uma haste articulada (ver Figura 1). Enfatiza-se que a haste articulada é o pêndulo e através do deslocamento da base pretende-se manter esta na posição vertical.

Utiliza-se neste trabalho um simulador para o pêndulo invertido, o qual permite a validação e comparação entre o controlador nebuloso e o PID. A implementação, de cunho

didático, foi desenvolvida 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 grau.

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

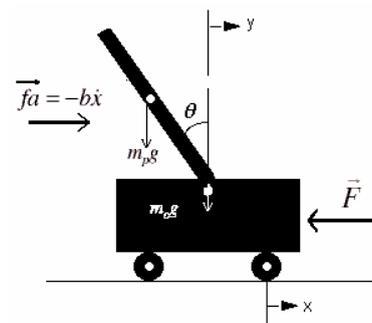


Figura 1. Sistema do pêndulo invertido

A seguir é descrita de forma sucinta a modelagem dinâmica do sistema. Para tanto, serão utilizados os símbolos descritos na Tabela 1.

Tabela 1. Relação de símbolos de variáveis e constantes

<b>Símbolo</b>	<b>Descrição</b>	<b>Valor</b>
$l$	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	Rad
$w$	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>

Para a modelagem, utilizou-se a Mecânica Lagrangeana, chegando a seguinte equação do movimento [3,6]:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \mathbf{w} \\ x \\ v \end{pmatrix} = A \begin{pmatrix} \mathbf{q} \\ \mathbf{w} \\ x \\ v \end{pmatrix} + \text{vet} \quad (1)$$

onde,

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

$$\text{vet} = \begin{pmatrix} -\frac{m_p l \cos q F}{\det} + \frac{(m_c + m_p) m_p g l \text{sen} q}{\det} \\ \frac{(m_p l^2 + I) F}{\det} - \frac{(m_p l)^2 g \text{sen} q \cos q}{\det} \end{pmatrix} \quad (3)$$

$$\det = (m_c + m_p)(m_p l^2 + I) - (m_p l \cos q)^2 \quad (4)$$

e  $F$  é a força restauradora de controle.

### 3. Controlador PID

O controlador PID é bastante difundido na indústria de modo geral. Este tipo de controlador é projetado para um determinado ponto de operação (*set point*) e normalmente apresenta bom desempenho em condições operacionais

semelhantes àquela do projeto, porém, quando ocorrem mudanças maiores no ponto de operação do sistema, o controlador pode não mais apresentar um desempenho satisfatório diante da nova condição operacional. Desta forma, para manter um bom desempenho numa ampla faixa de operação é necessário que, diante de uma mudança no ponto de operação, o controlador possa ter seus parâmetros automaticamente ajustados para o novo estado do sistema adaptando-se a essas alterações.

PID é uma abreviatura referente as primeiras letras das palavras proporcional, integral e diferencial, assim subentendido: proporcional a um erro, proporcional à integral no tempo deste erro e proporcional à derivada no tempo deste erro.

A equação fundamental do controlador PID pode ser escrita da seguinte forma:

$$u(t) = k_p e(t) + k_p k_i \int_0^{t_f} e(t) dt + k_p k_d \frac{de(t)}{dt} \quad (5)$$

onde,

$u(t)$ : é o sinal de controle,

$e(t)$ : é o erro que vem do sistema,

$k_p$ : o ganho proporcional,

$k_d$ : o ganho derivativo e

$k_i$ : o ganho integral.

Pela simples inspeção da equação acima, percebe-se que a primeira parcela do lado direito da mesma está relacionada com o controle proporcional, a segunda com o integral e a terceira com o diferencial.

No caso do pêndulo invertido suponha-se que seja realizada uma medida do ângulo  $\theta$  que o pêndulo faz com a vertical e que o erro seja produzido com relação a um valor de demanda  $q_D$ , fixo pelo usuário. Este erro é dado por:

$$e(t) = (q(t) - q_D) \quad (6)$$

Considerando que para a integral, tem-se:

$$i(t) = \int_0^t e(t) dt \quad (7)$$

e tomando a sua derivada, tem-se:

$$\frac{di(t)}{dt} = e(t), \text{ ou } \frac{di(t)}{dt} = (q(t) - q_D) \quad (8)$$

Para a parte derivativa tem-se:

$$\frac{de(t)}{dt} = \frac{d(\mathbf{q}(t) - \mathbf{q}_D)}{dt} = \mathbf{q} = \mathbf{w} \quad (9)$$

Assim, a equação fundamental pode ser escrita da forma:

$$u(t) = k_p e(t) + k_p k_i i(t) + k_p k_d w(t) \quad (10)$$

No caso particular do pêndulo a ser equilibrado  $\mathbf{q}_D = 0rd$ , a força restauradora é a força de controle do sistema, assim representada:  $F(t) = u(t)$ .

A inclusão da equação de controle na formulação do pêndulo implica em integrar a equação (8). Isto pode ser conseguido modificando as equações (1) (2) e (3).

$$\frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \mathbf{w} \\ x \\ v \\ i \end{pmatrix} = A_c \begin{pmatrix} \mathbf{q} \\ \mathbf{w} \\ x \\ v \\ i \end{pmatrix} + \mathit{vet}_c \quad (11)$$

$$A_c = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{(m_p l)^2 \cos(\mathbf{q})}{\det} & 0 & \frac{m_p l \cos(\mathbf{q})}{\det} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{(m_p l^2 + I) \cos(\mathbf{q})}{\det} & 0 & \frac{b(m_p l^2 + I)}{\det} & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (12)$$

$$\mathit{vet}_c = \begin{pmatrix} 0 \\ -\frac{m_p l \cos(\mathbf{q}) F}{\det} + \frac{(m_c + m_p) m_p g \sin(\mathbf{q})}{\det} \\ 0 \\ \frac{m_p l^2 + I}{\det} F - \frac{(m_p + l^2) g \sin(\mathbf{q}) \cos(\mathbf{q})}{\det} \\ -\mathbf{q}_D \end{pmatrix} \quad (13)$$

As equações acima são respectivamente: a equação da dinâmica do pêndulo invertido com o controle PID, a matriz do sistema e o vetor de entradas externas.

## 4. Controlador fuzzy

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 entrada) 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.

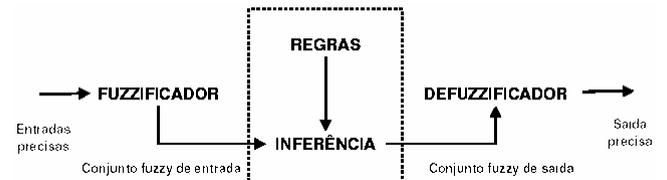


Figura 2. Diagrama de um controlador nebuloso

### 4.1. 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 de intervalos razoáveis, definindo-se assim as funções de pertencimento para valores que poderão ser assumidos por estas.

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

As variáveis de entrada e saída que são relevantes para os testes do sistema são:

- Erro do ângulo,
- Derivada da variação do ângulo (velocidade angular),
- Integral do erro do ângulo,
- Erro da posição horizontal,
- Derivada da posição (velocidade),
- Tempo: não é fuzzyficado, relevante somente na discretização,
- Força horizontal aplicada (saída).

### 4.2. Funções de pertencimento

Os processos de fuzzyficação 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, conforme mostra a Figura 3. 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.

A Figura 4 mostra o intervalo escolhido para o erro do ângulo, 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 mesmo limites, mas as funções de pertencimento do negativo e positivo foram tomadas como trapézios com sua parte triangular concentrada em torno do zero, como mostra a Figura 5. 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, mostrado na Figura 6.

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.

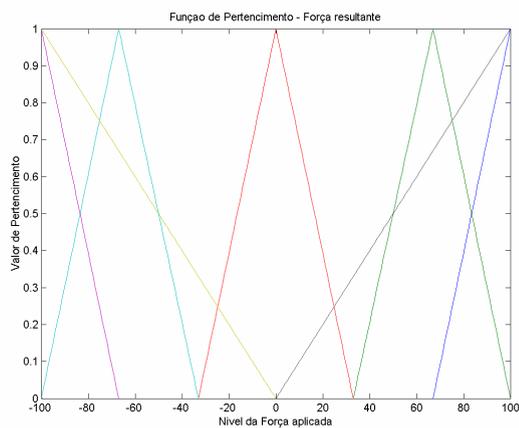


Figura 3. Função de Pertencimento para força horizontal

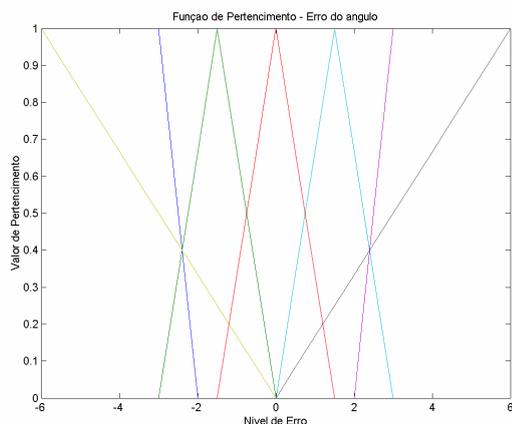


Figura 4. Função de Pertencimento do erro do ângulo

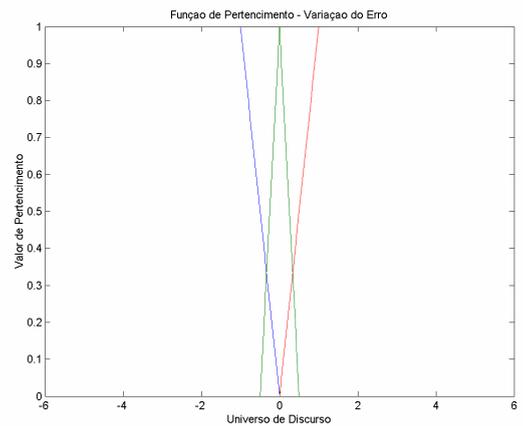


Figura 5. Função de Pertencimento para a derivada do erro do ângulo

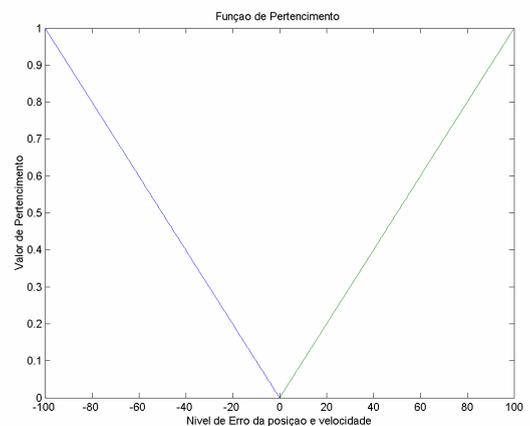


Figura 6. Função de Pertencimento para o erro da posição e velocidade do carrinho

### 4.3. Regra de implicação

O operador de implicação nada mais é do que a interpretação das regras, onde a escolha da regra de implicação é um parâmetro importante, do ponto de vista que é nessa fase que as variáveis lingüísticas são concebidas, conseqüentemente a interpretação *AND*, *OR*, e *ELSE*. O operador de implicação utilizado foi o Produto Larsen, respondendo perfeitamente ao problema proposto. A equação abaixo mostra o produto dos valores de pertencimento:

$$f_p[m_A(x), m_B(y)] \equiv m_A(x) \bullet m_B(y) \quad (14)$$

onde,  $f_p[m_A(x), m_B(y)]$  é a relação de implicação entre as regras aplicadas ao operador Larsen, do qual avalia o produto dos valores de pertencimento obtidos pela interpretação do *ELSE* das regras pelo processo de inferência *OR*, denotando uma união das relações de implicação para cada regra individualmente [9].

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

Tabela 2. Regras descritas pelas funções de pertencimento

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

#### 4.4 Métodos de defuzzificação

Os métodos de defuzzificação utilizados neste trabalho consistem em encontrar o centro de massa do conjunto *fuzzy* de saída, conhecidos como Centro de Áreas (*Center of Área – COA*) e Centro de Somas (*Center of Sums – COS*). A diferença básica entre estes dois métodos é que o COA leva em consideração a sobreposição das áreas resultantes das funções de pertencimento associadas a cada regra. Enquanto que o COS considera a soma de todas essas áreas, levando em conta a intersecção das mesmas duas ou mais vezes, ou seja, esta é uma maneira indireta de se atribuir pesos as componentes da média calculada.

O método de COA generalizado para valores discretos é definido como:

$$u = \frac{\sum_{i=1}^N u_i m_{SAIDA}(u_i)}{\sum_{i=1}^N m_{SAIDA}(u_i)} \quad (15)$$

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:

$$u = \frac{\sum_{i=1}^N u_i \cdot \sum_{k=1}^n m_{B_k}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n m_{B_k}(u_i)} \quad (16)$$

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.

## 5. Resultados

Os resultados obtidos têm por finalidade avaliar o desempenho de um controlador Nebuloso e compará-lo com o controlador PID. Para a realização dos testes foi utilizado um Pentium III, com capacidade de processamento de 800 MHz e 264 MB.

Através das regras descritas na Tabela 2 e o método de implicação citado, a Figura 7 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 computacionais 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 8. O sistema ainda apresentou maiores oscilações, chegando a um gasto computacional de 3943 segundos, deixando explícito uma dificuldade maior na computação dos dados.

Nos testes realizados com o controle PID [6], através de várias simulações após obter um bom conjunto de ganhos, o ângulo do pêndulo atingiu o valor zero e a velocidade do carrinho um valor muito próximo de zero, como mostra a Figura 9. Os valores para o PID dão-se com um tempo de simulação de 1000 segundos, ainda com um deslocamento do carrinho superior ao do controlador *fuzzy*.

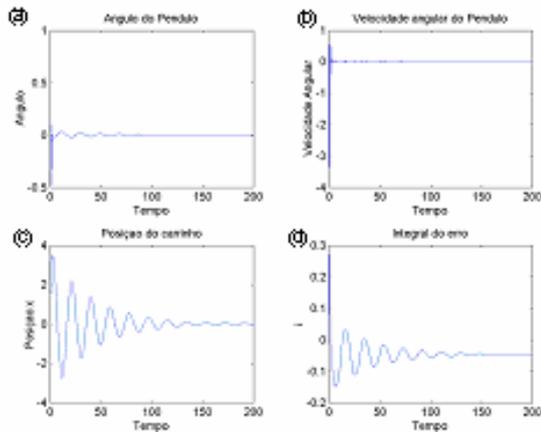


Figura 7. Larsen e COS. (a) ângulo do pêndulo; (b) velocidade angular; (c) posição do carrinho; (d) integral do erro

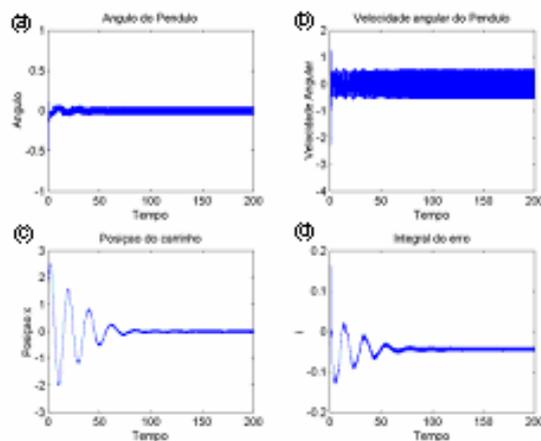


Figura 8. Larsen e COA. (a) ângulo do pêndulo; (b) velocidade angular; (c) posição do carrinho; (d) integral do erro

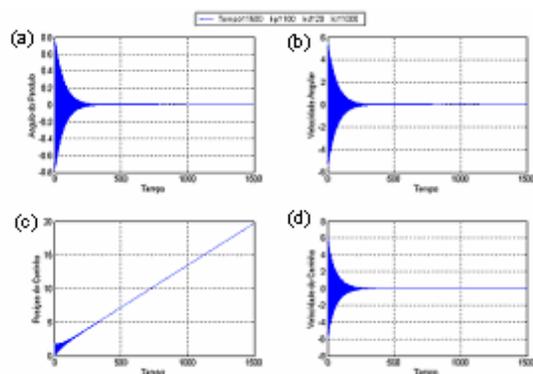


Figura 9. Resposta do controle PID, (a) ângulo do pêndulo, (b) velocidade angular, (c) posição do carrinho, (d) velocidade do carrinho.

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).

## 6. 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. Todos os experimentos foram realizados utilizando a formulação matemática da dinâmica de comportamento do pêndulo descritas na sessão 2.

O desenvolvimento de controladores para sistemas dinâmicos não-lineares compreende um desafio. 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*.

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

## 7. Referências

- [1] D. M. Etter, Engineering problem solving with MATLAB, Editora Prentice-Hall, 1993.
- [2] J. W. Hines, Matlab Supplement to Fuzzy and Neural Approaches in Engineering, Editora Jhon Wiley and Sons, New York, 1997.
- [3] L. T. Hoffman, Controlador Nebuloso para o Pêndulo Invertido, Projeto Final, INPE, 2003.
- [4] B. C. Kuo, Sistemas de Controle Automático, Editora Prentice-Hall do Brasil, 1982.
- [5] J. W. Leech, Mecânica Analítica, traduzido por Carlos Campos de Oliveira, Universidade do Estado da Guanabara, Rio de Janeiro, 1971.
- [6] M. H. E. Morais, A. Muralikrishna, R. Bravo, Controladores PID e Nebuloso para o Pêndulo Invertido, Projeto Final, INPE, 2004.
- [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.