

UM SIMULADOR VOLTADO PARA O ENSINO ASSISTIDO POR COMPUTADOR DE CONTROLADORES NEBULOSOS

EMERSON A. D'AQUINO, MAX S. DUTRA

*Laboratório de Robótica, Departamento de Engenharia Mecânica, Universidade Federal do Rio de Janeiro.
Caixa Postal 68503, CEP 21945-970, Rio de Janeiro, RJ, Brazil.
E-mail: maxdutra@ufrj.br*

ALEXANDRE EVSUKOFF

*Instituto Doris Ferraz de Aragon – ILTC
Rua Almirante Teffé, 637, CEP 24030-080, Niterói, RJ, Brazil.
E-mail: evsukoff@cepel.br*

Resumo— Este artigo descreve a implementação de um simulador de pêndulo invertido ou simples, que pode ser utilizado como ferramenta auxiliar em cursos de controle nebuloso. O simulador permite utilizar um controlador nebuloso e compará-lo a um controlador convencional (linear) com realimentação de estado, projetado através da técnica de alocação de pólos. Os resultados de duas simulações são apresentados. O simulador foi escrito em linguagem Java, como um *applet*, podendo ser facilmente incorporado a um documento HTML e disponibilizado na *internet*, em projetos de Educação à Distância.

Abstract— This work describes the implementation of an inverted or simple pendulum simulator that can be used as auxiliary tool in fuzzy control courses. The simulator allows using one fuzzy controller and comparing it to one conventional (linear) controller with state feedback, designed by using poles placement technique. The results of two simulations have been presented. The simulator has been written in Java programming language, as an *applet*. So, it can easily be embedded in an HTML document and made available over internet, in distance learning programs.

Keywords— Distance Learning, Computer-Aided Education, Fuzzy Control, Inverted Pendulum.

1 Introdução

As metodologias de ensino em todas as áreas vêm sofrendo o impacto de rápidas mudanças impulsionadas pelo avanço da informática e das tecnologias de rede. Recentemente, a popularização da *internet* abriu novas e poderosas perspectivas para o ensino assistido por computador e a educação à distância.

No ensino de engenharia, a introdução de computadores nas salas de aula tem permitido o complemento dos ensinamentos teóricos com a realização de trabalhos práticos e simulações numéricas. Entretanto, tais iniciativas focalizam geralmente técnicas convencionais, existindo ainda uma carência de iniciativas voltadas para o ensino de Sistemas de Automação Inteligentes, como, por exemplo, sistemas nebulosos e redes neurais artificiais.

Controladores nebulosos têm sido largamente utilizados em diversas aplicações e representam uma alternativa viável para sistemas demasiadamente complexos, pois não requerem a modelagem matemática do sistema. O pêndulo invertido é uma aplicação clássica da teoria de controle. Assim, este trabalho descreve a implementação de um simulador de pêndulo invertido ou simples, um *applet* escrito em Java, que poderá ser utilizado em projetos de ensino à distância, como ferramenta auxiliar em cursos de controle nebuloso. Este *applet* deverá ser

incorporado a um documento HTML e disponibilizado via *internet*.

O simulador permite visualizar o movimento do sistema, através de animações e gráficos em tempo real. Dois tipos de controlador podem ser utilizados para equilibrar o pêndulo: um controlador nebuloso e um controlador linear com realimentação de estado, projetado através da técnica de alocação de pólos. O simulador permite variar os parâmetros do sistema e de seus controladores, possibilitando diversas experiências diferentes.

Este artigo está organizado da seguinte forma. Na próxima seção, apresenta-se o modelo matemático não-linear do sistema. Na seção 3, discute-se a síntese do controlador linear, referido no restante do texto como “controlador convencional”. Na seção 4, apresenta-se o controlador nebuloso. A seção 5 descreve a interface gráfica do simulador. Na seção 6, são exibidos os resultados de duas simulações e, finalmente, as conclusões e perspectivas são apresentadas na última seção.

2 Modelo Matemático

O sistema, apresentado na Fig. 1, constitui-se de um pêndulo montado sobre um carrinho que se desloca ao longo de um trilho. O pêndulo possui massa m , concentrada na extremidade da haste rígida, sem massa, de comprimento l . O ângulo entre o pêndulo e a vertical é denominado θ . O carrinho possui massa m_c , e sua posição ao longo do trilho é dada pela coordenada x .

Na Fig. 1, são apresentados os diagramas de corpo livre do carrinho e do pêndulo. Para o carrinho foram consideradas apenas as forças que agem na direção horizontal. A força F é exercida pelo controlador sobre o sistema. A força F_a corresponde à força de atrito. A força H é exercida pelo pêndulo sobre o carrinho. No diagrama de corpo livre do pêndulo, temos: seu peso mg , a força V , exercida pelo carrinho sobre o pêndulo na direção vertical, e o torque de atrito T_a . As equações de movimento do sistema pêndulo-carrinho são dadas por:

$$\ddot{\theta} = \frac{g \sin \theta + \frac{T_a}{ml} - \cos \theta \left(\frac{F + F_a + ml \dot{\theta}^2 \sin \theta}{m + m_c} \right)}{l \left(1 - \frac{m \cos^2 \theta}{m + m_c} \right)} \quad (1)$$

$$\ddot{x} = \frac{F + F_a - ml(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta)}{m + m_c} \quad (2)$$

onde g é a aceleração da gravidade. O atrito será modelado como:

$$F_a = -c_f \dot{x}, \quad T_a = -c_t \dot{\theta} \quad (3)$$

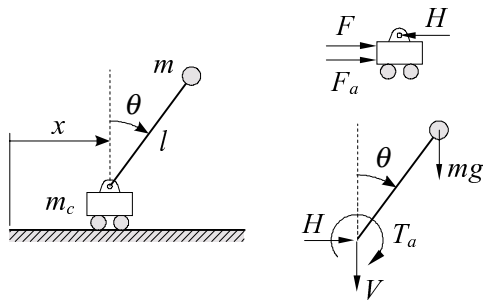


Figura 1. Representação esquemática do sistema pêndulo-carrinho e os diagramas de corpo livre para o carrinho e o pêndulo.

Para se obterem soluções aproximadas para as equações de movimento (1) e (2), utilizou-se o método de Runge-Kutta (4ª. ordem), um método de passo simples cujo erro local é proporcional a h^5 , onde h é o incremento utilizado (Boyce e Di Prima, 1994; Ruggiero e Lopes, 1988).

3 Controlador Convencional

Dois tipos de controlador podem ser utilizados nas simulações com pêndulo invertido: um controlador nebuloso e um controlador linear. A finalidade deste último é servir como referência para a avaliação do desempenho do primeiro. O controlador linear é realimentado com o estado do sistema e projetado através da técnica convencional de alocação de pólos, sendo, por esta razão, referido neste artigo como “controlador convencional”. O objetivo do controle é equilibrar o pêndulo na posição vertical invertida ($\theta = 0$), mantendo-se o carrinho na posição $x = 0$.

O projeto do controlador convencional requer a linearização, em torno de $\theta = 0$, do modelo dado pelas equações de (1) a (3). Portanto, por hipótese, $\sin \theta \cong \theta$, $\cos \theta \cong 1$ e $\dot{\theta} \sin \theta \cong 0$. Definindo-se as

variáveis de estado: $x_1 = \theta$, $x_2 = \dot{\theta}$, $x_3 = x$ e $x_4 = \dot{x}$, e o sinal de controle $u = F$, obtém-se o modelo linear:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (4)$$

onde \mathbf{x} é o vetor de estado e as matrizes \mathbf{A} e \mathbf{B} são dadas por:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(m+m_c)g}{m_c l} & -\frac{c_t}{m m_c l^2} & 0 & \frac{c_f}{m_c l} \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{m_c} & \frac{c_t}{m_c l} & 0 & -\frac{c_f}{m_c} \end{bmatrix} \quad (5)$$

$$\mathbf{B} = \begin{bmatrix} 0 & -\frac{1}{m_c l} & 0 & \frac{1}{m_c} \end{bmatrix}^T \quad (6)$$

Para o controlador convencional, a lei de controle é expressa por $u = -\mathbf{K}\mathbf{x}$. Portanto, a Eq. (4) torna-se:

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(t) \quad (7)$$

que tem como solução:

$$\mathbf{x}(t) = e^{(\mathbf{A} - \mathbf{B}\mathbf{K})(t-t_0)} \mathbf{x}(t_0) \quad (8)$$

Se os autovalores da matriz $\mathbf{A} - \mathbf{B}\mathbf{K}$, denominados pólos do controlador, estiverem situados à esquerda do eixo imaginário do plano complexo, então $\mathbf{x}(t)$ se aproximará de $\mathbf{0}$ à medida que t tenda para infinito, para qualquer $\mathbf{x}(t_0) \neq \mathbf{0}$. O projeto do controlador convencional consiste, portanto, na determinação da matriz \mathbf{K} tal que $\mathbf{A} - \mathbf{B}\mathbf{K}$ possua autovalores adequados.

Sejam $\lambda_1, \lambda_2, \lambda_3$ e λ_4 autovalores desejáveis para a matriz $\mathbf{A} - \mathbf{B}\mathbf{K}$, escolhidos segundo algum critério de desempenho adotado pelo usuário, como, por exemplo, o do regulador quadrático linear (Ogata, 1996). Então, o polinômio característico desejado será:

$$(s - \lambda_1)(s - \lambda_2)(s - \lambda_3)(s - \lambda_4) = s^4 + \alpha_1 s^3 + \alpha_2 s^2 + \alpha_3 s + \alpha_4 \quad (9)$$

Sejam ainda a_1, a_2, a_3 e a_4 os coeficientes do polinômio $|s\mathbf{I} - \mathbf{A}| = s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4$. Se o sistema for controlável, ou seja, se o determinante de sua matriz de controlabilidade \mathbf{M} , expressa por:

$$\mathbf{M} = [\mathbf{B} : \mathbf{A}\mathbf{B} : \mathbf{A}^2\mathbf{B} : \mathbf{A}^3\mathbf{B}] \quad (10)$$

for diferente de zero, então a matriz \mathbf{K} é dada por (Ogata, 1996):

$$\mathbf{K} = [\alpha_4 - a_4 : \alpha_3 - a_3 : \alpha_2 - a_2 : \alpha_1 - a_1] \mathbf{T}^{-1} \quad (11)$$

onde:

$$\mathbf{T} = \mathbf{M} \begin{bmatrix} a_3 & a_2 & a_1 & 1 \\ a_2 & a_1 & 1 & 0 \\ a_1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

4 Controlador Nebuloso

O desenvolvimento de controladores a partir da técnica de alocação de pólos requer a modelagem matemática do sistema a ser controlado, além da linearização das equações obtidas, como mostra a seção anterior. Controladores nebulosos podem ser desenvolvidos a partir de regras de controle lingüísticas fornecidas por um especialista com base em sua experiência e/ou intuição.

O controlador nebuloso desenvolvido atinge as metas de equilíbrio angular do pêndulo e de posicionamento do carrinho através de dois controladores nebulosos independentes, denominados, respectivamente, controlador angular e controlador de posição. Suas saídas são somadas para o controle final, como mostra a Fig. 2.

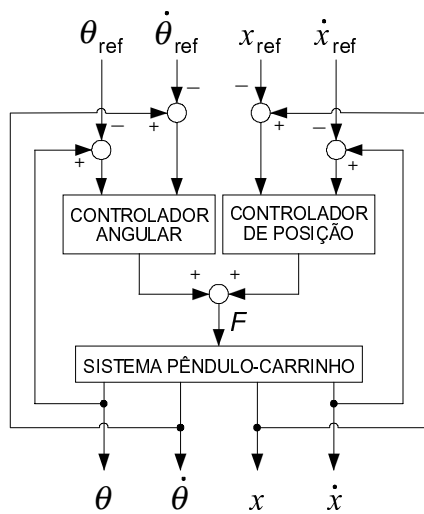


Figura 2. Representação esquemática do controlador nebuloso.

O controlador angular é responsável pela estabilização do pêndulo, seja na posição vertical com a haste voltada para cima (pêndulo invertido), seja na posição vertical com a haste voltada para baixo (pêndulo simples). Suas entradas são os erros associados ao ângulo e à velocidade angular do pêndulo, $e_\theta = \theta - \theta_{ref}$ e $\dot{e}_\theta = \dot{\theta} - \dot{\theta}_{ref}$. Os valores de referência θ_{ref} e $\dot{\theta}_{ref}$ são nulos. A saída deste controlador é a força a ser aplicada sobre o carrinho.

O controlador de posição é responsável pelo posicionamento do carrinho. Suas entradas são os erros associados à posição e à velocidade do carrinho, $e_x = x - x_{ref}$ e $\dot{e}_x = \dot{x} - \dot{x}_{ref}$. Os valores de referência x_{ref} e \dot{x}_{ref} dependem da meta de controle para a posição do carrinho. Duas metas foram implementadas: manter o carrinho em repouso numa posição arbitrária ao longo do trilho; e deslocar o carrinho de uma posição inicial até uma posição final, ambas arbitrárias, com velocidade constante.

Os sistemas TSK, também conhecidos como sistemas Takagi-Sugeno (Jang *et al.*, 1997; Passino e Yurkovich, 1998; Takagi e Sugeno, 1985), são sistemas nebulosos que possuem como regra típica:

Se \hat{x} é A e \hat{y} é B então $\hat{z} = f(\hat{x}, \hat{y})$

onde A e B são conjuntos nebulosos e $f(\hat{x}, \hat{y})$ é uma função *pr*iquantificadora. Se K é uma constante, o sistema é denominado *TSK* em *roc*. O objetivo de simplificar a implementação computacional de controladores nebulosos, com regras angulares e de posição são sistemas TSK de ordem zero, com função de pertinência triangular. Portanto, possumo seguinte base de regras:

Se ε é A_i e $\dot{\varepsilon}$ é B_j então $z_k = c_k$
 onde $i = 1, \dots, n_A$; $j = 1, \dots, n_B$; $k = 1, \dots, n_R$; ε e $\dot{\varepsilon}$ referem-se a e_θ e \dot{e}_θ , no caso do controlador angular, e a e_x e \dot{e}_x , no caso do controlador de posição. As constantes c_k são as saídas das regras; n_A e n_B correspondem ao número de conjuntos nebulosos nas discretizações (partições) nebulosas associadas às variáveis de entrada ε e $\dot{\varepsilon}$, respectivamente; e n_R é o número de regras nebulosas. A saída z é dada por:

$$z = \frac{\sum_{k=1}^{n_R} w_k z_k}{\sum_{k=1}^{n_R} w_k} \quad (13)$$

onde $w_k = T(\mu_{A_i}(\varepsilon), \mu_{B_j}(\dot{\varepsilon}))$ representa o valor de ativação da regra k , $\mu_{A_i}(\varepsilon)$ e $\mu_{B_j}(\dot{\varepsilon})$ são as funções de pertinência dos conjuntos nebulosos A_i e B_j , e o operador $T(\cdot)$ é uma t-norma. Neste trabalho, foram implementadas duas opções de t-norma: mínimo e produto.

5 Descrição da Interface do Simulador

O simulador foi escrito em Java, sob a forma de um *applet* (Cadenhead, 1998). A linguagem Java foi escolhida por ser independente de plataforma, por ser uma linguagem de programação orientada a objetos e porque *applets* escritos em Java podem facilmente ser incorporados a documentos HTML, facilitando sua difusão através da *internet*.

A interface gráfica do simulador é composta por quatro painéis, os quais podem ser acessados através do *menu de escolha* situado na parte superior da tela. A Fig. 3 mostra o painel "Principal". Neste painel há um *display* onde, durante as simulações, é exibida uma animação correspondente ao movimento realizado pelo sistema pêndulo-carrinho, em tempo real. O estado do sistema é indicado à direita deste *display*. É possível alterar os parâmetros do sistema (m, m_c, l, c_f e c_d), as condições iniciais e o incremento h do método de Runge-Kutta. Há duas opções para o ângulo do pêndulo: θ , próprio para simulações de pêndulo invertido, e $\phi = \pi - \theta$ adequado para o pêndulo simples.

É possível também alterar o controlador que atua sobre o sistema, há três opções: *nenhum*, onde nenhum controlador atua sobre o sistema, *controlador nebuloso* e *controlador convencional*. É sempre permitido ao usuário atuar sobre o sistema pêndulo-carrinho através do teclado. As setas à

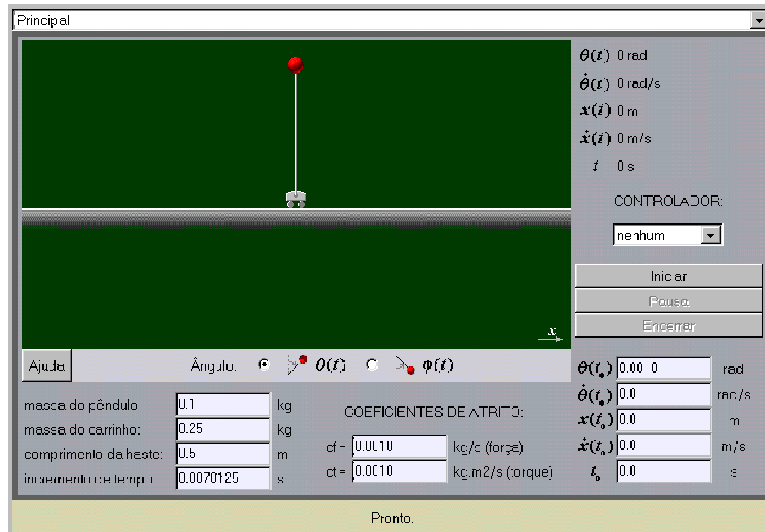


Figura 3. Painel “Principal”.

direita e à esquerda controlam o sinal da força aplicada, positivo e negativo, respectivamente, enquanto que as setas para cima e para baixo determinam o módulo da força, aumentando-o e diminuindo-o, respectivamente. A força é aplicada enquanto as setas à direita ou à esquerda estiverem pressionadas, e seu módulo permanece constante até que o usuário o altere. A força “exercida” pelo usuário é sempre somada à força exercida pelo controlador do sistema.

O painel “Gráficos”, apresentado na Fig. 4, permite a geração, em tempo real, das curvas $\theta(t)$ ou $\phi(t)$, $\dot{\theta}(t)$ ou $\dot{\phi}(t)$, $x(t)$ e $\dot{x}(t)$, utilizando-se controlador nebuloso, o controlador convencional ou nenhum controlador. Podem ser geradas ainda as curvas $F(t)$ e $Pot(t)$, onde $Pot(t)$ é a potência instantânea fornecida ao sistema pêndulo-carrinho pelo acionado. Curvas associadas a controladores diferentes podem ser sobrepostas num mesmo gráfico para fins de comparação. O intervalo de tempo observado nos gráficos pode ser alterado. Este painel também permite ao usuário atuar sobre o

sistema através do teclado, do mesmo modo que no painel “Principal”.

O painel “Controlador Nebuloso” apresenta três sub-painéis, exibidos nas figuras 5, 6 e 7, todos acessíveis através de um menu de escolha. Os sub-painéis “Controle Angular” e “Controle de Posição do Carrinho” permitem definir os parâmetros associados aos controladores angular e de posição, respectivamente. Tais parâmetros envolvem: o ângulo do pêndulo, θ ou ϕ ; a t-norma a ser empregada; o número de conjuntos nebulosos, bem como as posições dos centros das funções de pertinência, utilizadas nas discretizações (partições) nebulosas das variáveis de entrada; e as constantes presentes nos consequentes das regras.

O sub-painel “Meta de Controle para a Posição do Carrinho” permite estabelecer a meta de controle para o controlador de posição. Duas metas estão disponíveis: manter o carrinho em repouso numa posição arbitrária ao longo do trilho; e deslocar o carrinho de uma posição inicial até uma posição final, ambas arbitrárias, com velocidade constante.

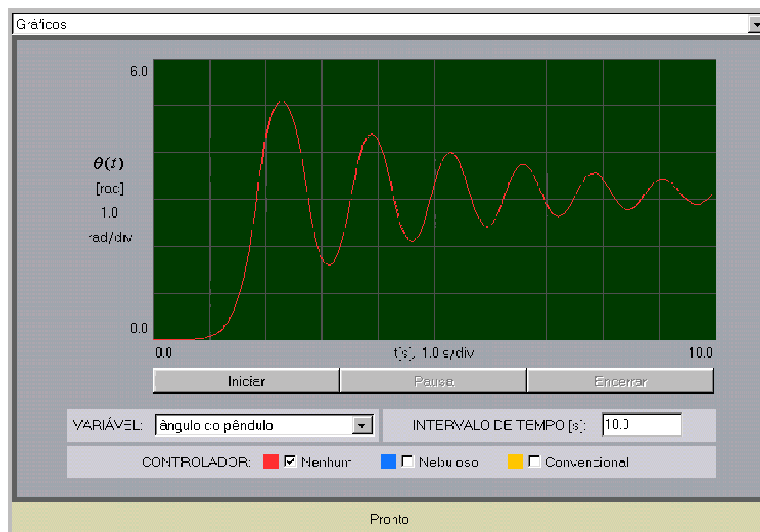


Figura 4. Painel “Gráficos”.



Figura 5. Sub-painel “Controle Angular”.

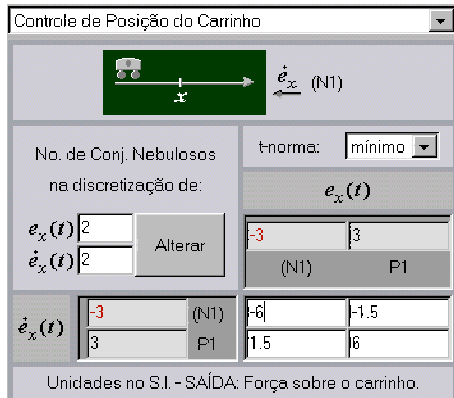


Figura 6. Sub-painel “Controle de Posição do Carrinho”.

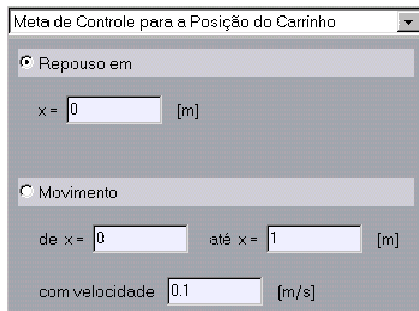


Figura 7. Sub-painel “Meta de Controle para a Posição do Carrinho”.

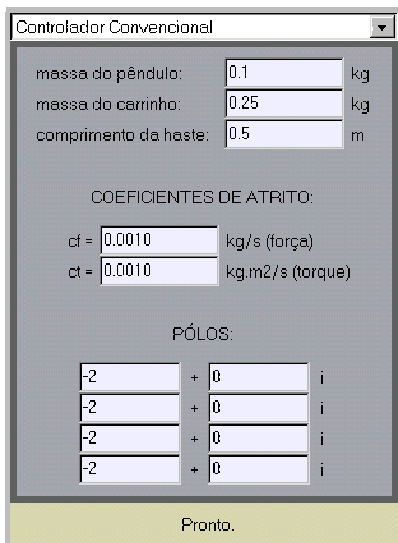


Figura 8. Painel “Controlador Convencional”.

O painel “Controlador Convencional”, apresentado na Fig. 8, permite estabelecer os parâmetros ligados ao controlador convencional. Os parâmetros do sistema pêndulo-carrinho: m , m_c , l , c_f e c_t são utilizados somente no cálculo da matriz \mathbf{K} , dada pela Eq. (11).

6 Resultados

Nesta seção, são apresentados os resultados de duas simulações. Os parâmetros associados aos controladores nebuloso e convencional são apresentados nas figuras de 5 a 8. A meta de controle para ambos os controladores é estabilizar o pêndulo na posição vertical invertida ($\theta = 0$) e manter o carrinho em $x = 0$. Em ambas as simulações, foram adotados os seguintes parâmetros: $m = 0,1 \text{ kg}$; $m_c = 0,25 \text{ kg}$; $l = 0,5 \text{ m}$; $c_f = 0,001 \text{ kg/s}$ e $c_t = 0,001 \text{ kg} \cdot \text{m}^2/\text{s}$. Na primeira simulação, utilizou-se $\theta(0) = 0,1 \text{ rad}$ e na segunda $\theta(0) = 0,949 \text{ rad}$. As demais condições iniciais (em $t = 0$) foram nulas. O incremento de tempo h , do método de Runge-Kutta, foi de $0,0078125 \text{ s}$ (2^{-7} s). Os pólos do controlador convencional foram escolhidos de modo que a potência de acionamento fosse, aproximadamente, a mesma para os controladores convencional e nebuloso ao longo da primeira simulação. Os resultados das simulações são apresentados nas figuras de 9 a 14. As curvas em azul estão associadas ao controlador nebuloso, as curvas em amarelo ao controlador convencional.

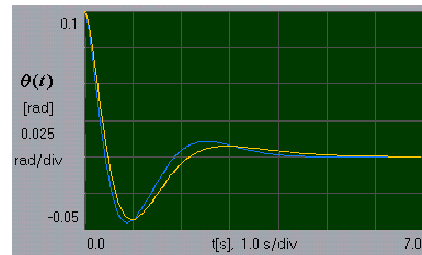


Figura 9. Ângulo do pêndulo na 1ª. simulação.

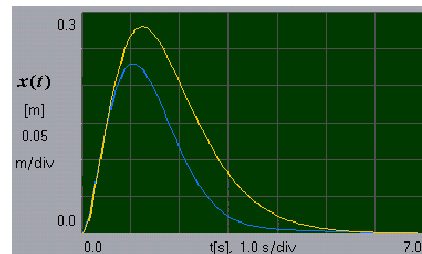


Figura 10. Posição do carrinho na 1ª. simulação.

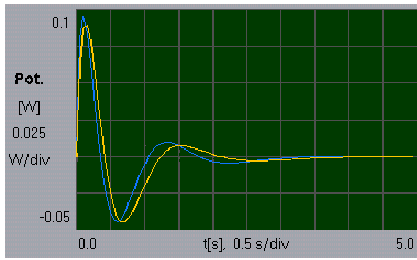


Figura 11. Potência de acionamento na 1ª. simulação.

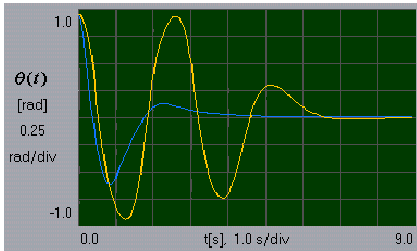


Figura 12. Ângulo do pêndulo na 2ª. simulação.

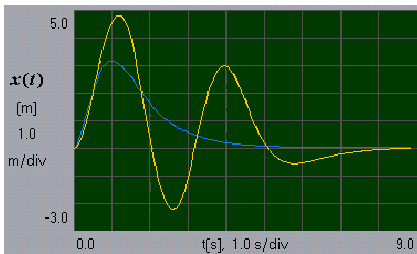


Figura 13. Posição do carrinho na 2ª. simulação.

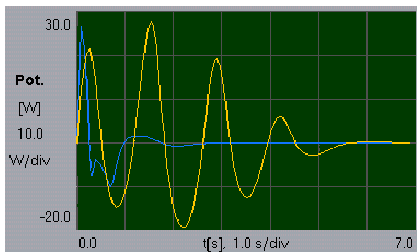


Figura 14. Potência de acionamento na 2ª. simulação.

As figuras de 9 a 14 ilustram o que o simulador pode realizar em termos de geração de curvas, o que permite analisar o desempenho de cada controlador ou mesmo compará-los. Pode-se observar, por exemplo, que o controlador nebuloso utilizado, que apresenta apenas oito regras, possui desempenho equivalente ao do controlador convencional, nas condições mencionadas para a primeira simulação. As figuras de 12 a 14 ilustram como o desempenho de cada controlador é afetado pelo comportamento não-linear do sistema pêndulo-carrinho.

7 Conclusão

Este trabalho apresenta a implementação de um simulador de pêndulo invertido ou simples voltado para o ensino, assistido por computador, de

controladores nebulosos. Foram implementados dois controladores: um linear, com realimentação de estado, projetado através da técnica de alocação de pólos e um controlador nebuloso formado por dois controladores, um responsável pelo controle angular do pêndulo e o outro pelo posicionamento do carrinho, ambos implementados como sistemas TSK de ordem zero. O controlador linear não tem como objetivo obter desempenho ótimo, servindo apenas como base para a comparação de resultados obtidos com o controlador nebuloso.

O simulador é um *applet*, escrito em Java, o que facilita sua difusão através da *internet*. O principal objetivo do simulador é fornecer suporte para o ensino de técnicas de Automação Inteligente, entretanto, pode ser estendido para incorporar outras técnicas mais avançadas de controle convencional, como controle robusto e controle adaptativo.

Na continuação deste trabalho, está prevista a construção de uma bancada real para que os alunos possam confrontar os resultados obtidos em simulação com os resultados reais.

Referências Bibliográficas

- Boyce, W.E. e R.C. Di Prima (1994). Métodos Numéricos. *Equações Diferenciais Elementares e Problemas de Valores de Contorno*. 5ª. ed., Guanabara Koogan, Rio de Janeiro.
- Cadenhead, R. (1998). *Aprenda em 24 horas Java 1.1*. Campus, Rio de Janeiro.
- Jang, J.R., C. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing*. Prentice Hall, Upper Saddle River, U.S.A.
- Ogata, K. (1996). *Projeto de sistemas lineares de controle com Matlab*. Prentice-Hall do Brasil, Rio de Janeiro.
- Passino, K.M. & S. Yurkovich (1998). *Fuzzy Control*. Addison-Wesley, Menlo Park, U.S.A.
- Ruggiero, M.A.G. e V.L.R. Lopes (1988). Soluções Numéricas de Equações Diferenciais Ordinárias. *Cálculo Numérico Aspectos Teóricos e Computacionais*. McGraw-Hill, São Paulo.
- Takagi, T. & M. Sugeno (1985). Fuzzy identification of Systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* (jan./feb.), Vol. SMC-15, nº. 1, pp 116-132.