

LABORATÓRIO VIRTUAL DE CONTROLE: USO DE TÉCNICAS DE REALIDADE VIRTUAL PARA REPRESENTAÇÃO E CONTROLE DE UM SISTEMA BALL-BEAM

Éder Torres Patrício – eder@ufpa.br

Manoel Ribeiro Filho – mrf@ufpa.br

Orlando Fonseca Silva – orfosi@ufpa.br

Universidade Federal do Pará, Departamento de Engenharia Elétrica e de Computação
Campus Universitário do Guamá

Rua Augusto Corrêa, 01 CEP 66075-110- Caixa postal 479 PABX +55 91 211-2121 - Belém
- Pará - Brasil

Resumo: *Este trabalho visa primordialmente o desenvolvimento de um laboratório virtual como ferramenta de auxílio no ensino de disciplinas de controle. A estratégia de controle utilizada é por alocação de pólos via realimentação de estados. O ambiente tridimensional que representa o sistema dinâmico “Ball-Beam” foi construído utilizando-se a API Java™ 3D. O usuário pode executar o programa utilizando parâmetros da planta e ganhos de realimentação já calculados, porém, se desejar, pode alterar estes valores e então carregá-los no programa, dessa forma, sempre pode projetar um novo controlador e observar o comportamento do sistema em tempo real.*

Palavras-chave: *Realidade virtual, Sistema de controle, Software livre, Educação em informática*

1. INTRODUÇÃO

A interação no processo de ensino-aprendizagem, tornou-se um conceito definitivo e há consenso de que a Realidade Virtual (RV) pode ajudar efetivamente (E. Borges et al, 2002). Neste trabalho pretende utilizar as potencialidades provenientes da RV (Burdea et al, 1994) para auxiliar o ensino e o aprendizado das disciplinas da área de controle. A escolha do sistema dinâmico não-linear “ball-beam” deu-se pois trata-se de um dos problemas clássicos da engenharia de controle (Ogata, 1996). Este sistema tem sido usado para testar técnicas de controle (Faria et al, 1997), (Hauser et al, 1992) e construção de protótipos didáticos (Rothe et al, 1999) para serem usados em laboratórios de ensino e pesquisa.

A estratégia de controle empregada foi por alocação de pólos via realimentação de estados. Dessa forma foi implementado, em Linguagem Java, um simulador que resolve as equações diferenciais do modelo não linear considerando um atuador e sensores ideais e o controlador como sendo contínuo. Os resultados deste simulador se mostraram satisfatórios para os ganhos de realimentação de estados calculados.

Com este trabalho alunos e professores podem observar o processo de controle de um sistema com ganhos previamente calculados, mas também podem alterar parâmetros da planta, desenvolver os cálculos de seus próprios ganhos de realimentação de estados e então incorporá-los diretamente ao sistema, isto é, no mundo tridimensional que representa o

sistema “Ball-Beam”. Nele, os usuários poderão obter resposta visual imediata a qualquer alteração do projeto, proporcionando assim um melhor entendimento do problema.

Usando técnicas de RV construiu-se uma sala onde se colocou uma mesa e sobre esta o sistema “Ball-Beam”. Dessa forma o usuário poderá “andar” pela sala, como se estivesse em um laboratório real. Através de botões e campos de texto poderá interagir com o sistema, ordenando o início do processo de controle com valores padrões ou poderá alterá-los, além disso, pode-se visualizar a plotagem de gráficos das variáveis de estado e ainda rotacionar o ambiente tridimensional, obtendo assim diversos ângulos de visualizações.

2. MODELAGEM DO SISTEMA “BALL - BEAM”

O modelamento do sistema “Ball-Beam” requer uma análise física e matemática do sistema, então, seja a “Figura 1” representativa do sistema “Ball-Beam”, considera-se que a barra pode girar livremente em torno de seu centro quando submetida a um torque e a bola, por sua vez, é livre para rolar ao longo da barra, sem deslizamento (Faria et al, 1997).

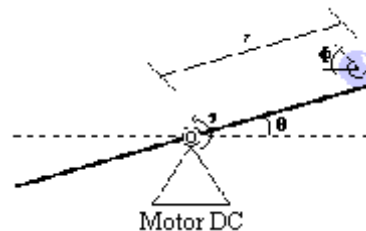


Figura 1 - Sistema “Ball-Beam”

Definindo-se:

r : deslocamento linear da bola [m]

θ : deslocamento angular da barra [rad]

τ : torque de entrada [N.m]

Φ : ângulo de rotação da bola [rad/s]

$\dot{\theta}$: velocidade angular da barra [rad/s]

O modelo pode ser obtido a partir da equação de Lagrange (Houpis et al, 1981), e resulta nas “equações (1)” e “equações (2)”.

$$0 = \left(\frac{J_b}{R_b^2} + m_b \right) \ddot{r} + m_b g \sin \theta - m_b r \dot{\theta}^2 \quad (1)$$

$$\tau = (m_b r^2 + J + J_b) \ddot{\theta} + 2m_b r \dot{r} \dot{\theta} + m_b g r \cos \theta \quad (2)$$

que caracterizam o sistema como não linear. Assim uma das técnicas para linearizá-lo é o uso das matrizes jacobianas (Ogata, 1990).

2.1. Estratégia de Controle por alocação de pólos via realimentação de estados

Consiste em determinar ganhos de realimentação de estados tal que os pólos do sistema em malha fechada assumam valores previamente especificados. Naturalmente, a escolha dos pólos de malha fechada desejada deve ser feita de tal modo que especificações como: sobre sinal, tempo de estabilização etc... sejam satisfeitas.

Esse tipo de controle, só é possível se, o sistema for controlável.

A formulação do problema de controle por realimentação de estados pode ser simplificada da seguinte forma (Astrom et al, 1984), (Franklin et al, 1980):

- a) Estabelecer o modelo matemático linearizado da planta na forma de espaço de estados $\dot{x} = Ax + Bu$
- b) Verificar se o sistema é controlável
- c) Especificar o polinômio característico de malha fechada desejado
- d) Determinar os ganhos de realimentação de estados, por exemplo, usando a fórmula de Akermann[8], ou seja, seguindo (Franklin et al, 1980), especificou-se, $\text{Det}(SI-A+BK)=(s-2)^4$ (todos os pólos de malha fechada em $-3,62$). Os ganhos obtidos a partir da equação anterior usando o software Matlab foram $K=[-0,818 \ -0,404 \ 0,548 \ 0,218]$.

Para verificação do desempenho do sistema em malha fechada, utilizou-se o simulador do Matlab denominado Simulink para implementar as equações dinâmicas não lineares do sistema em conjunto com os valores obtidos para os ganhos de realimentação de estados (Faria et al, 1997). Os resultados encontrados foram satisfatórios, assim, foi elaborado um programa em Java, que realizasse o mesmo papel desse simulador, isto é, que resolvesse as equações do modelo não linear considerando um atuador ideal e o controlador como sendo contínuo.

Os valores de K foram definidos como os ganhos de realimentação de estados padrões, isto é, quando um estudante carregar o software da animação 3D do sistema “Ball-Beam” tais valores serão passados ao simulador que realizará o controle.

3. PROJETO DO SISTEMA

3.1. Arquitetura

A arquitetura do sistema é formada por quatro blocos principais, ver “Figura 2”. No ambiente 3D foi construído uma sala com paredes, janelas, pisos e um sistema de iluminação. Também foi feita a mesa e sobre ela está o sistema “Ball-Beam” que é simulado em malha fechada de acordo com a “equação (1)” e “equação (2)” e os ganhos de realimentação de estados. Observa-se que um protótipo real deste sistema foi construído (Rothe et al, 1999) e faz parte de acervo do Laboratório de Controle e Sistemas – LACOS, da Universidade Federal do Pará. A vantagem do modelo virtual sobre o protótipo é a facilidade da alteração das condições iniciais e principalmente a possibilidade de alterar os parâmetros do sistema, como inércia da bola, massa da bola, raio da bola e inércia da barra, o que exige então um novo projeto dos ganhos de realimentação de estados.

A interface com o usuário trata do problema da interação usuário-computador. O usuário pode navegar pelo ambiente 3D através das setas direcionais do teclado e interagir com o sistema usando uma interface gráfica provida de botões e campos de texto.

O bloco “Simulador” trata da resolução da “equação (1)” e “equação (2)” com a realimentação de estados, enviando os resultados para o “Ambiente 3D” que se modificará a cada iteração das amostragens. O bloco de “Gráficos” é responsável por receber as variáveis de estados provenientes do bloco “Simulador” e então plotar esses dados em relação ao número de amostragens.

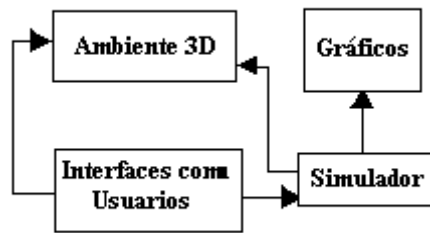


Figura 2- Arquitetura do sistema

3.2. Implementação Computacional

Todo o sistema foi implementado em linguagem Java™ (Sun, 2004) que é uma plataforma de programação livre e traz consigo a característica da portabilidade, isto é, Java™ usa a idéia “escreva uma vez, execute em qualquer lugar” o que é um recurso importante para qualquer aplicação. O bloco “Ambiente 3D” foi implementado utilizando a API Java™ 3D (Selman, 2002), (Bouvier, 2002) que se apresenta bastante difundida no âmbito do desenvolvimento de Sistemas de Realidade Virtual. Referindo-se ao bloco “Interfaces com Usuários”, a interface gráfica foi desenvolvida usando o pacote swing presente nas GUI (Graphical User Interface) do Java™. Sendo que a parte de navegação do usuário através do ambiente tridimensional foi feita usando a classe KeyNavigator da API Java™ 3D. O bloco “Simulador” foi implementado usando-se métodos da classe Math do Java™. O bloco de “Gráficos” foi desenvolvido sobre a API de geração de gráficos JFreeChart (JfreeChart, 2004).

4. RESULTADOS

Quando o programa é executado aparece a janela inicial, mostrada na “Figura 3”. Nesta janela aparecem os valores padrões como posição e velocidade inicial da esfera sobre a barra e posição e velocidade angular inicial da barra transversal, que podem ser alterados pelo usuário. Além disso, ele poderá usar os valores padrões para os parâmetros do sistema e os ganhos de realimentação ou modifica-los através dos CheckBox’s “Ganhos e Parâmetros do Sistema default”. O usuário poderá ver até quatro gráficos (posição e velocidade da esfera, posição e velocidade da barra), que são escolhidos através dos CheckBox’s correspondentes.

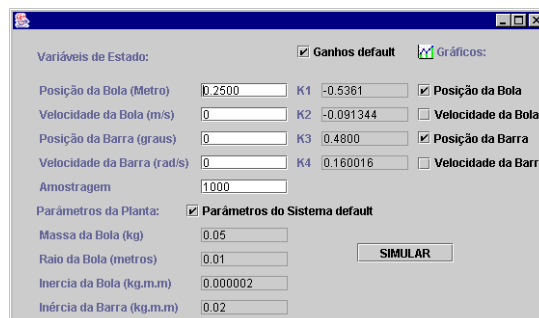


Figura 3- Janela inicial

Após o usuário ter feito suas configurações ele clicará no botão “Simular” quando então aparecerá o ambiente virtual tridimensional juntamente com alguns botões de controle, como mostra a “Figura 4”.

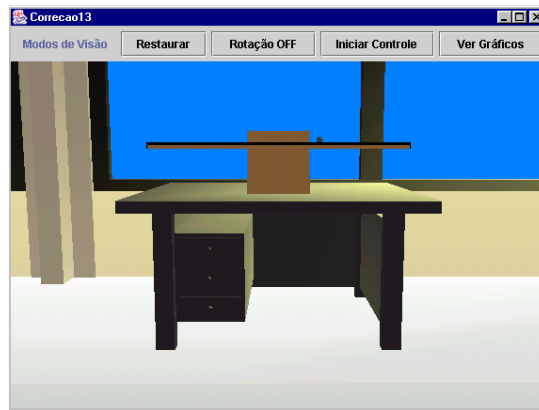


Figura 4- Ambiente virtual tridimensional

O usuário pode “andar” pela sala através das teclas direcionais. A “Figura 5” mostra outra visão do ambiente após o usuário ter se deslocado pela sala.

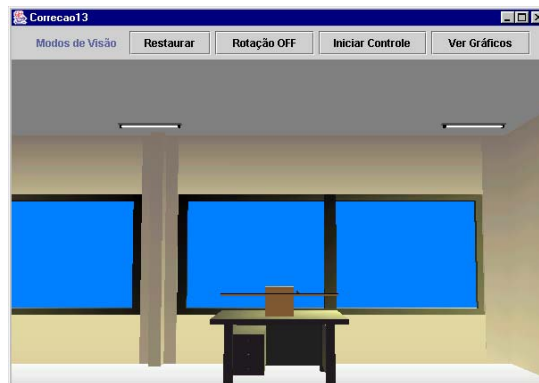


Figura 5- Outra visão do ambiente virtual

Existem quatro botões e um rótulo presentes na parte superior do ambiente virtual. O botão “Rotação” permite que a cena gire ao redor da origem cartesiana do ambiente 3D que se localiza no centro do modelo “ball-beam”, sobre a mesa, como mostra a “Figura 6”. Sendo assim, o usuário pode escolher qual angulação visual observará o sistema.



Figura 6- Ambiente rotacionado

O botão restaurar é responsável por reiniciar as variáveis de estados com seus valores padrões, isto é, após uma simulação, utilizando esse botão o usuário pode preparar o projeto para outra simulação não precisando fechar e abrir o programa novamente, voltando a cena da “Figura 4”. O botão “Iniciar Controle” é responsável pelo início da animação 3D do sistema “Ball-Beam”, a partir daí, a barra, em forma de calha, faz um movimento de gangorra, visando não deixar a bola cair por uma de suas extremidades, lembrando que a interação entre barra e bola é realizada a partir do modelo matemático do sistema. A “Figura 7” mostra o movimento da barra e da bola durante o processo de controle.

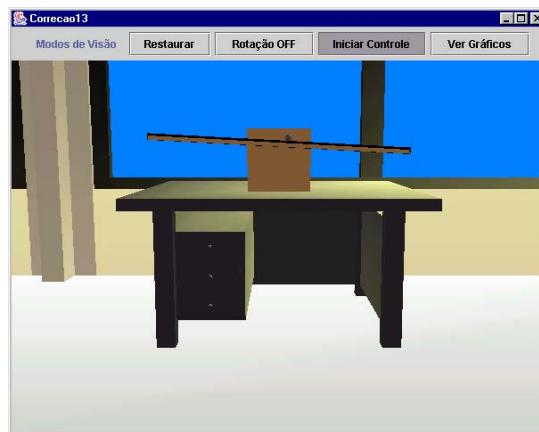


Figura 7- Sistema durante o controle

Depois que terminar o processo de controle a esfera encontra-se no centro da barra e esta fica na posição horizontal, como mostrado na “Figura 8”.

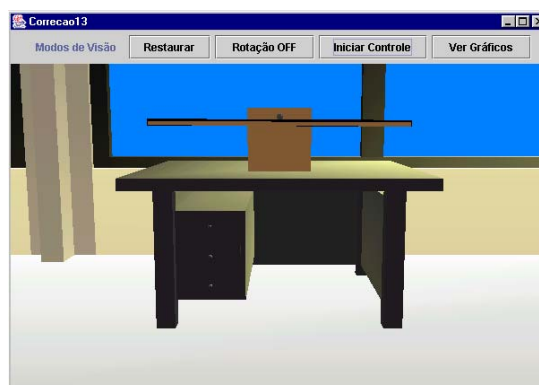


Figura 8- Sistema após o controle

O botão “Ver Gráficos” é utilizado pelo usuário para observar os gráficos das variáveis de estado, lembrando que o usuário, na janela inicial (“Figura 3”), escolheu quais os gráficos das variáveis de estado gostaria de observar. A “Figura 9” mostra o gráfico da posição da esfera em relação ao número de amostras.

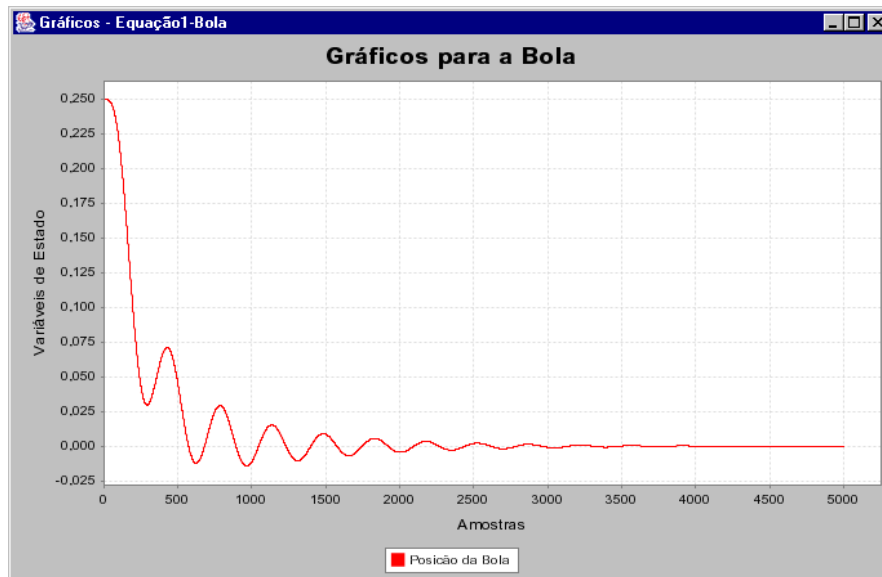


Figura 9- Posição da esfera versus Amostras

A “Figura 10” mostra a variação da posição da barra em relação ao numero de amostras.

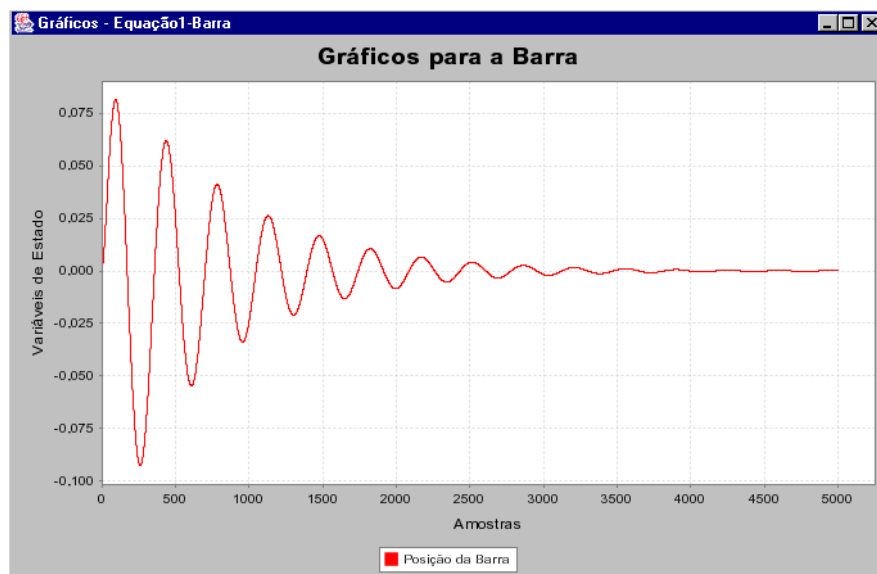


Figura 10- Posição da barra versus amostras

5. CONCLUSÃO

Neste trabalho apresentou-se o projeto, a implementação e os resultados de um laboratório virtual (Lopes et al, 2002) de controle, especificamente desenvolvido para o sistema dinâmico “Ball-Beam”. A construção de um ambiente virtual tridimensional representando o sistema “Ball-Beam” faz parte do interesse de desenvolver um laboratório virtual para disciplinas de Controle que contará também com outras plantas, como o “Pêndulo Invertido”, que já está em desenvolvimento, tendo como ponto de partida as mesmas características interativas desse trabalho.

É importante salientar que é possível, aos alunos ou professores realizarem a alteração nos parâmetros da planta: inércia da barra, inércia da bola, massa da bola e raio da bola, sendo

dessa forma, também necessário que o usuário recalcule os ganhos de realimentação de estados, ou seja, neste caso para cada execução do programa tem-se um novo sistema “Ball-Beam” e um novo projeto do controlador, além de permitir alterações nas variáveis de estado inicial: posição ou velocidade linear da bola e posição ou velocidade angular da barra bem como a taxa de amostragens.

Pretende-se montar um CD, contendo as máquinas virtuais do Java e do Java 3D, para as plataformas Windows e Linux, assim como o programa desenvolvido. Desta maneira o programa poderá ser facilmente instalado nessas duas plataformas. Como os programas desenvolvidos são implementados na linguagem Java, estes poderão rodar em browser's ou navegadores como Internet Explorer (Windows) e Mozilla (Linux).

6. REFERÊNCIAS BIBLIOGRÁFICAS

Astrom, Karl J., Wittenmark, Bjorn-Computer Controlled Systems - Theory and design - 2a edição, Prentice Hall, Inc., Englewood Cliffs, N.J 07632, 1984.

Bouvier, D. J. Getting Started with the Java 3D API, Online: disponível na Internet via <http://java.sun.com/products/java-media/3d/collateral> , 2002.

Burdea, Grigore e Coiffet, Philippe. Virtual Reality Technology, John Wiley & New York, NY, 1994.

D. Selman. Java 3D Programming, Manning Publications CO. 2002.

E. Borges, A Cardoso, e E. Lamounier Jr. Investigando técnicas interativas para aprimorar o uso de realidade virtual no ensino de geometria espacial. Anais do V Symposium on Virtual Reality, SVR 2002, pp.78-89, Fortaleza, outubro de 2002.

Faria, C. G. e Moscoso, M.N.A. Simulação Híbrida de um Sistema Dinâmico do tipo Ball-Beam com a Implementação de um Controlador Digital em Tempo Real, Trabalho de Conclusão de Curso, DEEC-CT-UFPA, dezembro de 1997.

Franklin, Gene F., Powell, J. David. Digital Control of Dynamic Systems - 1a edição, Addison - Wesley Publishing Company, 1980.

Houpis, D'Azzo-Análisa e Projeto de Sistemas de Controle Lineares-2a edição, McGraw-Hill, Inc, 1981.

J. Hauser, S. Sastry, and P. Kokotovic. Nonlinear control via approximate input-output linearization: The ball and beam example - IEEE Trans. Automat. Contr, vol. 37, pp. 392-398, 1992.

JfreeChart class library, Online: na Internet via <http://www.jrefinery.com> , 2004.

Lopes, A. S.; Oliveira, J. C.; Santos, M. P. L. Laboratórios Virtuais: uma nova alternativa na preparação dos alunos para as aulas experimentais. In: XXX Congresso Brasileiro de Ensino de Engenharia, 2002, Piracicaba. Anais. Piracicaba: COBENGE/2002, 2002, p. 1-10.

Ogata, Katsuhiro. Engenharia de Controle Moderno-2a edição, PHB-Prentice Hall do Brasil, 1990.

Ogata, Katsuhiro. Projeto de Sistemas Lineares de Controle com Matlab. Prentice - Hall do Brasil, Rio de Janeiro, 1996.

Rothe-Neves, M. Construção de protótipo para laboratório de ensino e pesquisa: sistema ball-beam, Trabalho de Conclusão de Curso, DEEC-CT-UFPA, dezembro de 1999.

Sun Microsystem. Online: site oficial <http://www.sun.com/>, 2004.

VIRTUAL LABORATORY OF CONTROL: USE OF TECHNIQUES OF VIRTUAL REALITY FOR REPRESENTATION AND CONTROL OF A BALL-BEAM SYSTEM

***Abstract:** This work primordially aims at the development of a virtual laboratory as tool of helping in the education of disciplines of control. The used strategy of control is for allocation of polar regions via feedback of states. The three-dimensional environment that represents the dynamic system "Ball-Beam" was constructed using the API (Application Programming Interface) Java™ 3D. The user can execute the program using parameters of the plant and profits of feedback already calculated, however, if it desire, it can modify those values and then to load them in the program, this way it can always project a new controller and observes the behavior of the system in real time.*

***Key-words:** Virtual reality, Control system, Free software's, Computing education*