

Experimento Virtual de Controle: Uso de Técnicas de Realidade Virtual para a Construção de um Sistema “Ball-Beam”.

Éder Torres Patrício
Max Rothe-Neves
Manoel Ribeiro Filho
Orlando Fonseca Silva

Departamento de Engenharia Elétrica e Computação - Laboratório de Controle e Sistemas
Universidade Federal do Pará – UFPA
Campus Universitário do Guamá
Rua Augusto Corrêa, 01 CEP 66075-110- Caixa postal 479 PABX +55 91 211-2121 - Belém - Pará - Brasil
eder@ufpa.br mrn@ufpa.br mrf@ufpa.br orfosi@ufpa.br

Abstract: The main purpose of this paper is the development of a virtual experiment to be used as a teaching tool for control disciplines. A linearized mathematical model of the system in the state-space form is used to project a controller using pole location. The three-dimensional ambient which represents the Ball-Beam dynamic system was built using the API Java 3D. The software is loaded with parameters and feedback proportional gains from a real plant, but it's allowed to the user to change those values. It can be used to observe the dynamic of a simulated system before using a controller in a real plant.

Resumo: Este trabalho visa primordialmente o desenvolvimento de um experimento 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 de uma planta real e ganhos de realimentação já calculados ou pode inserir novos valores, para observar a dinâmica do sistema simulado antes de uma aplicação de controle em uma planta real.

1-Introdução

A interação no processo de ensino-aprendizagem tornou-se um conceito definitivo e há consenso de que a Realidade Virtual (RV) pode auxiliar em algumas áreas das ciências [1]. Este trabalho utiliza as potencialidades provenientes da RV [2] com o intuito de auxiliar o processo de ensino e aprendizagem de conceitos relativos a disciplinas da área de controle de sistemas.

Quando um modelo matemático é obtido pela física do processo, geralmente algumas características inerentes a sistemas reais são desprezadas, devido à dificuldade de serem modeladas e por onerarem a ordem do modelo obtido. Com a supressão de tais características, as respostas dos sistemas real e simulado, para um dado controlador, podem ser significativamente diferentes, a ponto de um sistema em malha fechada ser estável e o outro instável. Dessa forma, atividades de simulação desempenham um

papel fundamental em todo o processo de projeto e aplicação de controladores, e tem como objetivo “prever” o comportamento do sistema real para uma situação específica.

O experimento virtual de controle desenvolvido é o do sistema dinâmico não-linear “Ball-Beam”. A escolha por este sistema justifica-se por dois motivos: 1) trata-se de um dos problemas clássicos da engenharia de controle [3] e 2) existe uma planta real deste sistema, construída com recursos próprios, no Laboratório de Controle de Sistemas (LACOS) da Universidade Federal do Pará (UFPA), que tem sido utilizada para testar técnicas de controle em atividades de ensino e pesquisa. [4, 10, 5]. O uso do experimento virtual deverá auxiliar o projeto de novos controladores.

A estratégia de controle empregada nessa primeira versão de experimento é por alocação de pólos via realimentação de estados. Implementado em Linguagem Java, o simulador resolve as equações diferenciais do modelo que contém uma

entrada e quatro variáveis de estados (o atuador e os sensores para a leitura das variáveis de estados foram considerados ideais). O período de amostragem utilizado foi de 1 ms, sendo suficientemente pequeno para considerar o sistema contínuo.

Com este trabalho, alunos e professores podem observar o processo de controle do sistema para uma determinada condição inicial e com ganhos previamente calculados, ou podem alterar esses valores à sua vontade. Dessa forma, obtém-se uma 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 pedir 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”

A modelagem 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 [4].

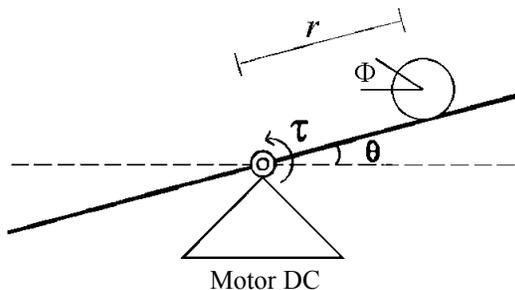


Figura-1 - Sistema “Ball-Beam”

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 [6], e resulta nas equações 1 e 2,

$$0 = \left(\frac{J_b}{R_b^2} + mb \right) \ddot{r} + mbg \sin \theta - mbr\ddot{\theta}^2 \quad \text{Eq (1)}$$

$$\tau = (mbr^2 + J + J_b) \ddot{\theta} + 2mbr\dot{\theta} + mbgr \cos \theta \quad \text{Eq (2)}$$

que caracterizam o sistema como não-linear. Assim uma das técnicas para linearizá-lo é o uso da matriz jacobiana [7].

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, entre outras sejam satisfeitas.

Esse tipo de controle só é possível se o sistema for controlável, como é o caso do sistema “Ball-Beam”.

A formulação do problema de controle por realimentação de estados para esse sistema pode ser simplificada da seguinte forma [8, 9]:

- a) Achar o modelo matemático linearizado da planta na forma de espaço de estados, $\dot{X}(t) = AX(t) + Bu(t)$.
- b) Especificar o polinômio característico de malha fechada desejado.
- c) Determinar os ganhos de realimentação de estados, por exemplo, usando a fórmula de Ackermann [8].

Segundo a referência [10], especificou-se todos os pólos de malha fechada em -2 . Os ganhos obtidos para a ação de controle do tipo $u(t) = -KX(t)$, usando o software Matlab foram $K = [-0.536 \ -0.091 \ 0.48 \ 0.16]$.

Para verificação do desempenho do sistema em malha fechada, utilizou-se o Simulink para implementar as equações dinâmicas do sistema em conjunto com os valores obtidos para os ganhos de realimentação de estados [4]. 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.

Os valores de K foram definidos como os ganhos padrões, isto é, quando um usuário carregar o programa 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 construída uma sala com paredes, janelas, pisos e um sistema de iluminação. Também foi feita a mesa e sobre ela encontra-se o sistema “Ball-Beam”, que é simulado em malha fechada de acordo com as equações (1) e (2) e os ganhos de realimentação de estados.

Qualquer alteração das condições iniciais ou dos parâmetros do sistema, como inércia da bola, massa da bola, raio da bola e inércia da barra, pode ser realizada. Com a alteração, um novo projeto de controlador deve ser realizado pelo usuário e os novos ganhos de realimentação de estados inseridos no programa.

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 das equações (1) e (2) com a realimentação de estados, enviando os resultados para o **Ambiente 3D** que se modificará a cada iteração das amostras. 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 amostras.

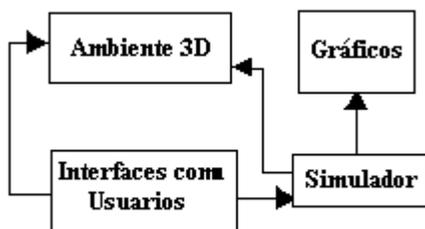


Figura-2: Arquitetura do sistema

3.2- Implementação Computacional

Todo o sistema foi implementado em linguagem Java [10] que é uma plataforma de programação livre e traz consigo a característica

de portabilidade, isto é, Java usa a idéia “escreva uma vez, execute em qualquer lugar” o que é um recurso importante para qualquer aplicação.

O **Ambiente 3D** foi implementado utilizando a API (Application Programming Interface) Java 3D [12, 13] 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 [14].

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, além dos ganhos de realimentação, que podem ser alterados pelo usuário 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.

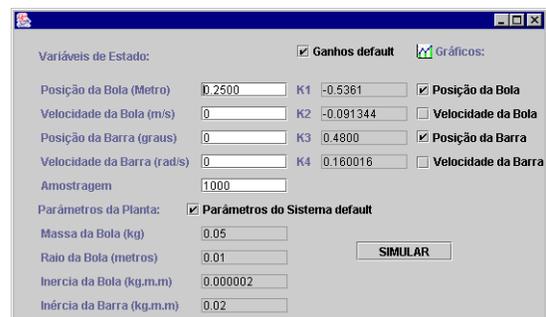


Figura-3: Janela inicial

Após o usuário ter feito suas configurações ele clicará no botão **Simular** quando então aparecerá o ambiente virtual tridimensional juntamente com os quatro botões de controle na parte superior do ambiente virtual, como mostra a Figura-4, onde a bola está na posição 0.25 metros da origem.



Figura-4: Ambiente virtual tridimensional

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-5 ilustra o movimento da barra e da bola durante o processo de controle.

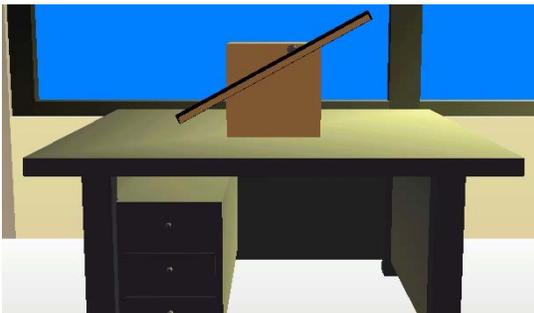


Figura-5: Sistema durante o controle

Depois do término do processo de controle a esfera encontra-se no centro da barra e esta fica na posição horizontal, como mostra a Figura-6.

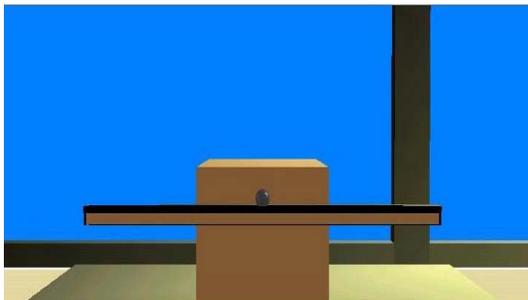


Figura-6: Sistema após o controle

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 que gostaria de observar. A Figura-7 mostra o gráfico da posição da esfera em relação ao número de amostras.



Figura-7: Posição da bola versus Amostras

O usuário pode navegar pela sala através das teclas direcionais. A Figura-8 mostra outra visão do ambiente após o usuário ter se deslocado pela sala.



Figura-8: Outra visão 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-9. Sendo assim, o usuário pode escolher qual angulação visual observará o sistema.

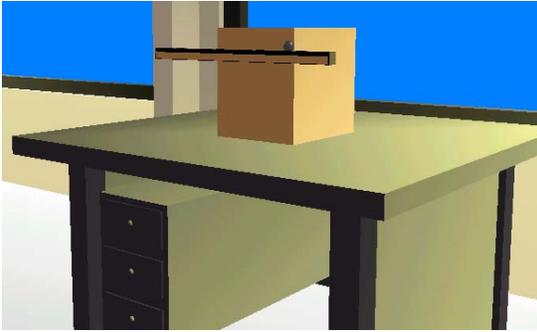


Figura-9: Ambiente rotacionado

O botão restaurar é responsável por reiniciar as variáveis de estados com os valores configurados pelo usuário, isto é, após uma simulação, utilizando esse botão o usuário pode preparar o sistema para outra simulação não precisando fechar e abrir o programa novamente. A Figura-10 mostra o ambiente virtual com a bola localizada a -0.25 metros da origem. Após o disparo do botão “Iniciar Controle” o sistema inicia o processo de controle, a Figura-11 mostra o sistema durante o controle. Após o controle o sistema encontra-se como mostra a Figura-6.



Figura-10: Bola localizada a -0.25 metros

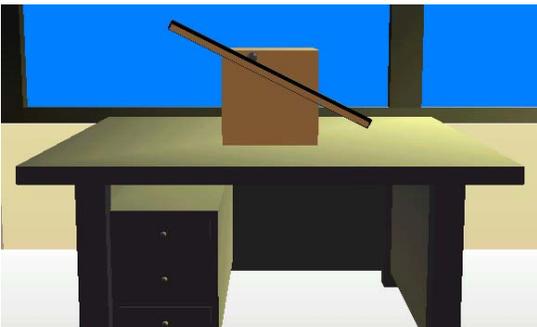


Figura-11: Sistema durante o controle

Como principal vantagem da utilização desse experimento em conjunto com um sistema real, é que os respectivos resultados, incluindo a visualização das dinâmicas, podem ser comparadas. A busca por modelos matemáticos mais representativos se torna mais eficiente com a utilização de tais ferramentas, possibilitando conseqüentemente, uma melhor confiança no projeto de controladores. A Figura-12 ilustra a resposta dos sistemas real e simulado, para o problema de regulação, para uma posição inicial da bola igual a 7 cm e pólos de malha fechada em $S = -7$.

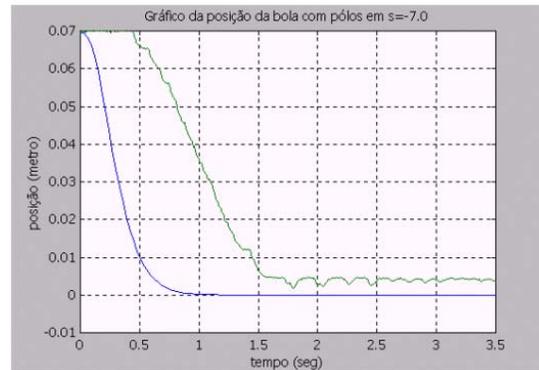


Figura-12: Gráfico comparativo das respostas do sistema real x simulado

5- Conclusão

Neste trabalho apresentou-se o projeto, a implementação e os resultados de um experimento virtual de controle, especificamente desenvolvido para 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 se alunos ou professores desejarem alterar o projeto do controlador, isso é feito alterando as variáveis de estado inicial, como posição ou velocidade da bola, posição ou velocidade da barra, taxa de amostragens e parâmetros da planta (inércia da barra e bola, massa e raio da bola), dessa forma também será necessário que o usuário recalcule os ganhos de realimentação de estados, evidenciando-se que para cada execução do programa um novo sistema “Ball-Beam” pode ser projetado.

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

- [1] – 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.
- [2] –Burdea, Grigore e Coiffet, Philippe – Virtual Reality Technology, John Wiley & New York, NY, 1994.
- [3] – Ogata, K., (1996). Projeto de Sistemas Lineares de Controle com Matlab. Prentice - Hall do Brasil, Rio de Janeiro.
- [4] – 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 der Conclusão de Curso, DEEC-CT-UFPA, dezembro de 1997.
- [5] – 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.
- [6] –Houpis, D’Azzo-Análisa e Projeto de Sistemas de Controle Lineares-2a edição, McGraw-Hill, Inc, 1981.
- [7] – Ogata, Katsuhiko-Engenharia de Controle Moderno-2ª edição, PHB-Prentice Hall do Brasil, 1990.
- [8] – Astrom, Karl J., Wittenmark, Bjorn-Computer Controlled Systems – Theory and design – 2a edição, Prentice Hall, Inc., Englewood Cliffs, N.J 07632, 1984.
- [9] – Franklin, Gene F., Powell, J. David – Digital Control of Dynamic Systems – 1a edição, Addison – Wesley Publishing Company, 1980.
- [10] – 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.
- [11] – Sun Microsystem. Online(2004): site oficial <http://www.sun.com/>
- [12] –D. Selman, Java 3D Programming, Manning Publications CO. 2002.
- [13] –BOUVIER, D. J., “Getting Started with the Java 3D API”, Online: disponível na Internet via <http://java.sun.com/products/java-media/3d/collateral>
- [14] – JfreeChart class library,(2004) Online: na Internet via <http://www.jrefinery.com>