

Desenvolvimento de um Programa Aberto de Ensino de Sistemas de Controle

Matheus Pilotto Figueiredo ^{1*}
Mauro André Barbosa Cunha ^{2*}
Glaucius Décio Duarte ^{3*}
Marcelo dos Santos Coutinho ^{4*}
Carlos Mendes Richter ^{5*}
Ricardo Andrade Cava ^{6*}

Resumo: Este artigo apresenta os resultados produzidos no desenvolvimento de um Programa Aberto de Ensino de Sistemas de Controle (PAESC) a ser utilizado como ferramenta auxiliar nas atividades de ensino da disciplina de Sistemas de Controle. Essa disciplina é parte integrante de vários currículos em cursos de engenharia e possui uma característica multidisciplinar, envolvendo, dentre outros, conteúdos de Cálculo, Física, Circuitos Elétricos, Mecanismos, dificultando o aprendizado dos alunos. Com o objetivo primordial de suprir essa dificuldade, o Programa Aberto de Ensino de Sistemas de Controle está sendo desenvolvido no sentido de proporcionar uma ferramenta tecnológica, disponível pela *Web*, para auxiliar o aprendizado dos futuros engenheiros. Para tanto, apresenta gráficos, animações e permite a interação. A ferramenta está sendo construída em linguagem *Java*, e os códigos produzidos são abertos, de modo a permitir que possam ser enviadas contribuições às simulações desenvolvidas.

Palavras-chave: Ensino de Sistemas de Controle. Ferramenta on-line. Gráficos e simulações. Linguagem de programação Java.

Abstract: This paper presents the results obtained by the development of a Control Systems Teaching Open Program to be used as an auxiliary tool in the teaching activities related to Control Systems. This discipline is part of various curricula in engineering and has a multidisciplinary

¹ Acadêmico do Curso de Engenharia Elétrica; ² Doutor em Engenharia Elétrica; ³ Doutor em Informática na Educação; ⁴ Mestre em Engenharia Elétrica; ⁵ Doutor em Engenharia Elétrica; ⁶ Mestre em Ciências da Computação.

* Integrante do Grupo de Pesquisa em Automação e Controle do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense

feature, involving content of calculus, physics, electrical circuits, mechanisms, among others. This feature makes the discipline's contents harder for students to learn. Having as the main purpose to overcome such a difficulty, the Control Systems Open Program has been developed to provide a technological tool, available by the web, to assist the learning of future engineers. It presents graphics, animations and allows interaction. The tool has been built in Java language and the codes produced are open so that other people can send their contributions to the developed simulations.

Key words: Control Systems teaching. On-line Tool. Graphics and simulations. Java programming language.

1. Introdução

A disciplina de Sistemas de Controle é ministrada em diversos cursos, tais como: Engenharia de Controle e Automação, Engenharia Elétrica, Engenharia Mecânica e Engenharia Química. Ela é uma disciplina complexa devido ao fato de relacionar conhecimentos de diversas áreas, tais como: Matemática, Física, Eletricidade, Eletrônica, dentre outras. A literatura nessa área não tem se modificado nos últimos anos com relação aos assuntos da teoria básica, mas sim, com relação à forma de abordagem visando a facilitar o entendimento pelos alunos. Existe um grande número de autores na área da teoria básica de controle, tais como Ogata (2005) e Bishop e Dorf (2001), bastante utilizados em cursos de engenharia no Brasil. É importante citar a utilização de ambientes de simulação nessa área como forma de facilitar o ensino.

Considerando os fatos acima e, também, que os ambientes de simulação existentes não são ambientes específicos de ensino, o desenvolvimento de um programa voltado diretamente às características do ensino desenvolvido, que facilite a visualização de resultados, não só em gráficos, mas com animações dos sistemas estudados, pode auxiliar no aprendizado. Para tanto, o Grupo de Pesquisa em Automação e Controle (GPAC) do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense (IFSul), iniciou o desenvolvimento do Programa Aberto de Ensino de Sistemas de Controle (PAESC), disponível na Web em http://www2.cefetrs.tche.br/~gpac/projetos_paesc.html.

O PAESC é apresentado na sequência. Para tanto, este trabalho está dividido como segue. Na seção 2, apresentam-se as características do PAESC. Na seção 3, aborda-se a linguagem de programação utilizada. Na seção 4, discutem-se os programas desenvolvidos e, na seção 5, apresenta-se a conclusão do trabalho.

2. Características do sistema

O PAESC está sendo desenvolvido com as seguintes concepções:

- utilização de programas livres;
- códigos fontes disponíveis aos usuários;
- desenvolvimento de simulações interativas e didáticas;
- possibilidade de rodar as simulações na página ou de baixar os arquivos;
- possibilidade dos usuários enviarem sugestões.

3. Linguagem de programação

Com o objetivo de atender às características listadas na seção 2 e, também, com base nos conhecimentos prévios dos autores deste trabalho, alguns testes foram feitos com duas linguagens de programação, a saber: *Java* e *C++*.

Esses testes mostraram que as implementações em *C++* tiveram um menor tempo de execução quando comparadas às implementações em *Java*. Isto já era esperado, devido ao fato de *Java* ser uma linguagem interpretada. Contudo, conseguiu-se verificar que para os propósitos do PAESC esta diferença de tempo não é significativa. Além disso, têm sido desenvolvidas tecnologias como a compilação *Just In Time (JIT)* que melhoram o desempenho dos programas *Java* em tempo de execução, algo que não foi levado em conta nesses primeiros testes.

Após uma análise de diversos fatores, levando-se em conta a portabilidade dos programas, optou-se pela linguagem *Java*.

No site da empresa que criou a linguagem *Java* (SUNMICROSYSTEMS, 2009a), pode-se fazer o *download* gratuito de todas as ferramentas necessárias para o desenvolvimento de programas nessa linguagem, tais como: um instalador do *Java Standard Edition Development Kit (JDK)* e do *NetBeans*.

O *JDK* contém elementos como um compilador que transforma código fonte *Java* para *bytecode* (arquivos *.class*). A *Java Virtual Machine (JVM)*, que também integra o *JDK*, é responsável pela execução dos arquivos *.class*. O *NetBeans* é um ambiente de desenvolvimento integrado (*Integrated Development Environment - IDE*) que serve para escrever códigos fonte com mais facilidade e rapidez. Deve-se, também, citar que a utilização da documentação da *Application Programming Interface (API)* (SUNMICROSYSTEMS, 2009B) proporcionou uma utilização mais eficiente da coleção de classes que a API contém.

A literatura para linguagem *Java* e para soluções de problemas é bastante vasta. Neste trabalho, dentre outras, utilizou-se Sun Microsystems (2009C), Deitel e Deitel (2005) e Grupo de Usuários Java (2009).

Visando à anexação em páginas *HTML* (RAMALHO, 1997) e consequente execução a partir da *internet*, optou-se pela utilização de *applets Java*. Assim, exige-se como único requisito do usuário ter a *JVM* para a plataforma em utilização, mas é bem provável que o computador já a possua, pois os próprios *browsers* procuram atualizações nesse sentido continuamente.

O fato de utilizar a *Internet* permite integrar textos com teoria às respectivas simulações por meio de *links*, ativáveis pelo usuário se assim o desejar.

Os programas utilizados no projeto, bem como os produzidos, são todos gratuitos, permitindo o desenvolvimento do projeto por meio de estudantes ou programadores interessados.

Embora ainda não utilizado na versão atual, é interessante relatar que se estudou a *Virtual Reality Modeling Language (VRML)* (MANSSOUR, 2009) para possível utilização na continuação do PAESC.

Na próxima seção, apresentam-se os primeiros programas desenvolvidos.

4. Programas desenvolvidos

Os programas desenvolvidos objetivaram a simulação de sistemas dinâmicos. Para isso, após a obtenção das equações diferenciais, as mesmas foram discretizadas (Franklin et al., 1994) para serem implementadas em *Java*.

Ressalta-se que a classe implementada para plotar os gráficos é dinâmica e permite operações interativas através da mudança dos parâmetros da função, ou mesmo, por meio da utilização de ferramentas do gráfico, como a de ampliação ou a de marcação de pontos.

Na sequência, descrevem-se algumas simulações já incluídas no PAESC:

- **Funções do circuito RC** - apresenta três gráficos, um de tensão de entrada, um de tensão no capacitor e outro de tensão no resistor (Figura 1). Para esse trabalho, foram criados alguns novos métodos para as classes existentes para fazer a associação de funções de forma que quando uma principal terminasse de ser plotada as outras associadas dependentes dela fossem plotadas também.

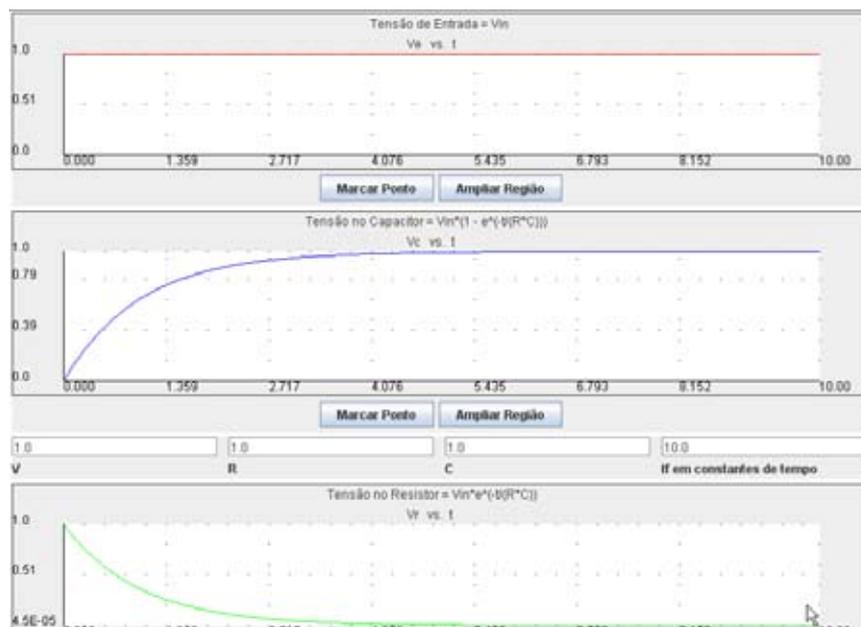


Figura 1 - Gráficos de um circuito RC

- **Simulação de um circuito RC** - consiste em uma simulação em que se pode modificar o estado de uma chave em um desenho de um circuito RC e simular como ficaria a tensão no capacitor com o passar do tempo, abrindo-se e fechando-se a chave (Figura 2). Para esse trabalho, foi criada a classe "GraficoComTempo" que chama o método *repaint()* do gráfico com certo período de tempo e modifica um *flag* indicando até que ponto ele deve ser plotado, dando a impressão de que o gráfico é algo como um osciloscópio durante a simulação.

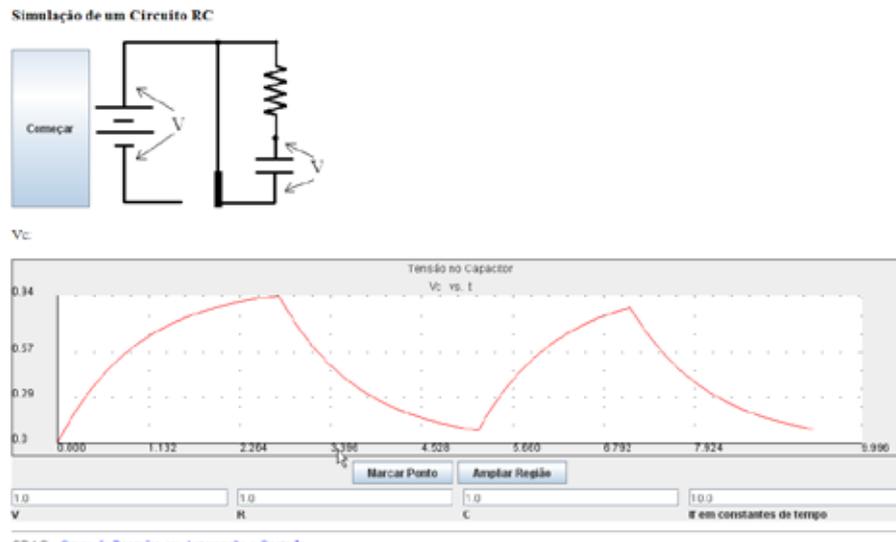


Figura 2 - Simulação de um circuito RC

- Identificação de um Sistema de Primeira Ordem** - exercício em que o estudante gera aleatoriamente um gráfico de um sistema de primeira ordem e, em seguida, deve analisar o gráfico para descobrir quais são os valores corretos para os parâmetros “Yreg” e “Tau” (Figura 3). Para este programa, a classe “Gráfico” foi incrementada para mostrar os valores de abscissa e ordenada dos pontos conforme o mouse é arrastado sobre o gráfico.

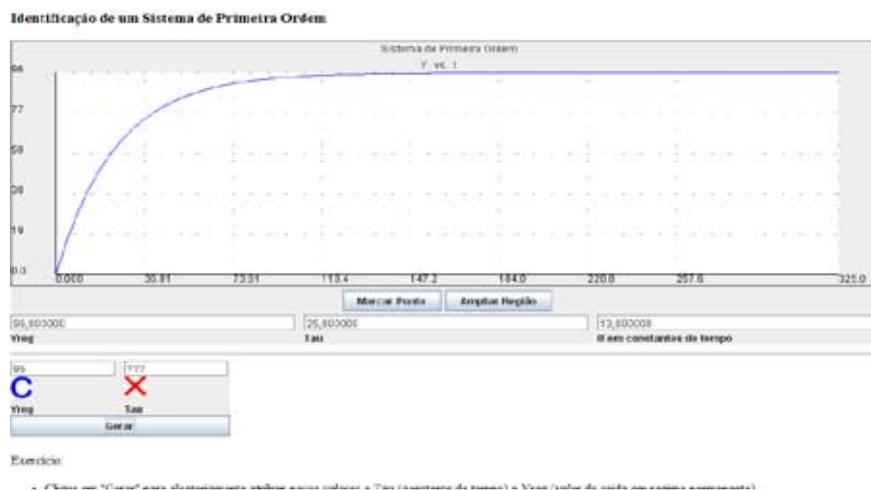


Figura 3 - Identificação de um sistema de primeira ordem

- **Identificação de um Sistema de Segunda Ordem** - consiste em um exercício semelhante ao de Identificação de um Sistema de Primeira Ordem. A diferença é que há mais parâmetros a serem descobertos. Para este programa, foi desenvolvida uma caixa de parâmetros com imagens, que permite que o estudante preencha campos com os valores que considera corretos e que estes sejam corrigidos pelo programa, mostrando uma imagem de certo (C) ou errado (X).

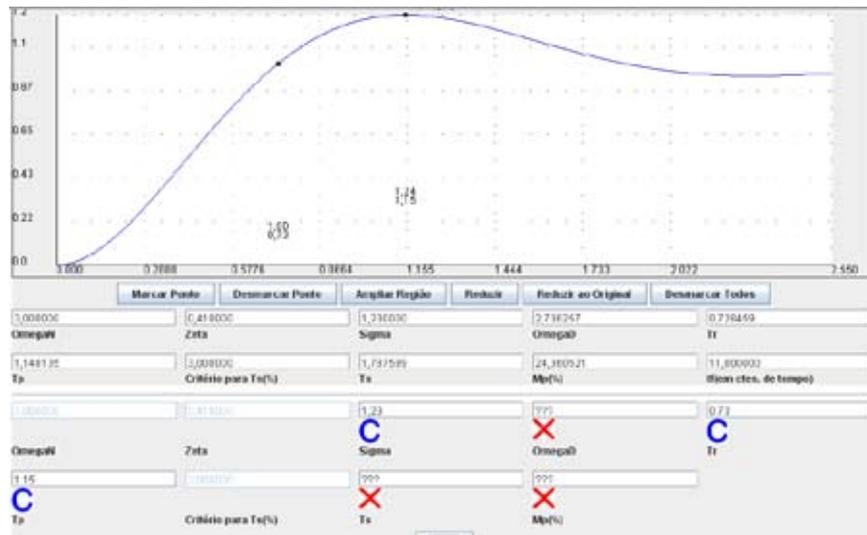


Figura 4 – Identificação de um sistema de segunda ordem

- **Controle PID de Velocidade** - consiste em um conjunto de três gráficos: Velocidade, Erro e Sinal de Controle. Pode-se selecionar as ações Proporcional, Integral e Derivativo para o sinal de controle independentemente. Isso permite a constatação do funcionamento dos controladores clássicos de forma conjunta ou isolada.

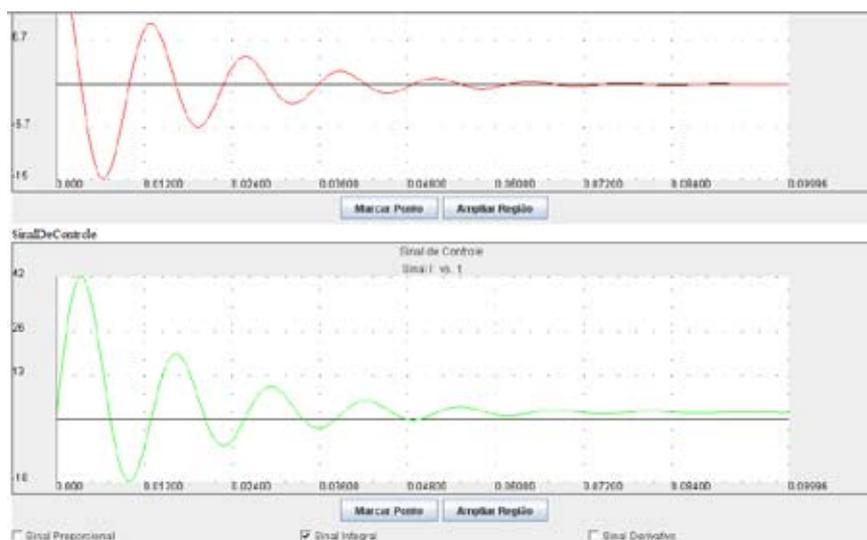


Figura 5 – Controle de velocidade

- **Controle de Posição** - contém um *applet* representando um sistema para teste de controle de posição com um ponteiro para posição de referência e outro cuja posição é controlada. Há três gráficos, um da posição, outro do erro e outro do sinal de controle, e é possível alterar parâmetros tanto do sistema físico como do sistema de controle.

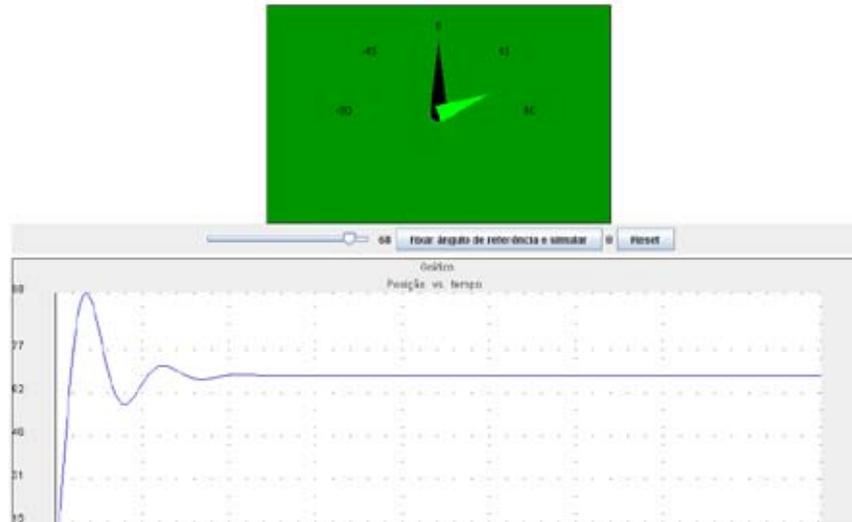


Figura 6 – Controle de posição

- **Simulação de um Sistema Massa-Mola-Amortecedor** – apresenta o gráfico de um sistema massa-mola-amortecedor (Figura 7) e utiliza uma equação diferencial discretizada para representar o sistema. A variação da posição do bloco é simulada por meio de uma animação em que o bloco se move ocupando as posições mostradas no gráfico conforme passa o tempo. Ao modificar algum parâmetro, o elemento correspondente pisca na simulação por alguns segundos para que o estudante perceba essa correspondência mais facilmente.

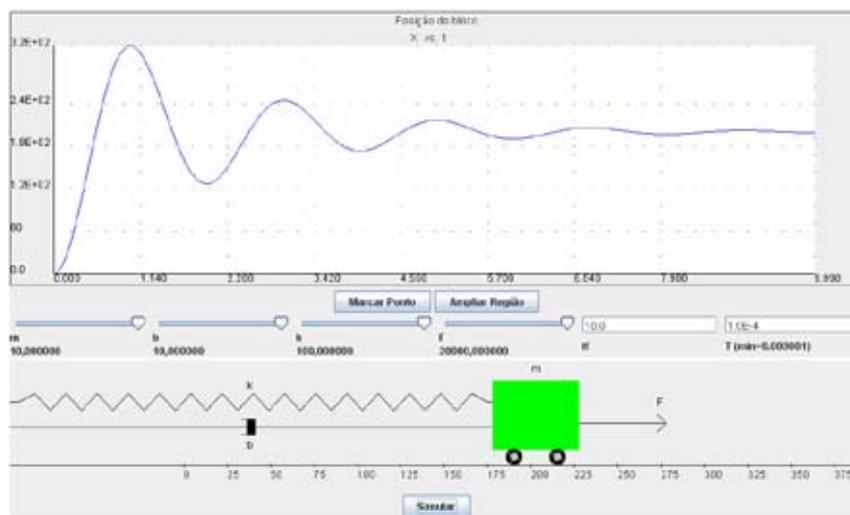


Figura 7 – Simulação de um sistema massa-mola amortecedor

Alguns desses programas já foram utilizados na disciplina de Sistemas de Controle no Curso Técnico de Eletrônica no IF-Sul (CUNHA, 2005).

5. Conclusões

Este trabalho tratou da implementação das primeiras simulações para o desenvolvimento do Programa Aberto de Ensino de Sistemas de Controle (PAESC). Discutiram-se as características buscadas pelo PAESC, a linguagem de programação utilizada e os programas implementados.

Como os programas são desenvolvidos em *software* livre, há possibilidade de expansão do programa com a colaboração de programadores e estudantes interessados em criar novas classes ou simulações, melhorar as atuais, ou mesmo sugerir novas idéias ou modificações.

Devido à dimensão do projeto idealizado, pretende-se dar continuidade ao mesmo em trabalhos futuros.

Referências

- CUNHA, Mauro André Barbosa. **Apostila de Sistemas de Controle: Curso Técnico de Eletrônica - CEFET**. Pelotas: [S.e.], 2005.
- DEITEL, Harvey M.; DEITEL, Paul J. Java: **Como programar**. 6.ed. São Paulo: Prentice-Hall, 2005.
- DORF, Richard C.; BISHOP, Robert H. **Sistemas de Controle Modernos**. Rio de Janeiro: LTC, 2001.
- FRANKLIN, Gene F.; POWELL, J. David; EMAMI-NAEINI, Abbas. **Feedback Control of Dynamic Systems**. 5.ed. [S.l.]: Prentice Hall, 2005.
- Grupo de Usuários Java. Disponível em: <<http://www.guj.com.br/home.index.logic>> Acesso em: jun. 2009.
- MANSSOUR, Isabel Harb. **Introdução à VRML 2.0**. Disponível em: <<http://www.inf.pucrs.br/~manssour/VRML/index.html>> Acesso em: jun. de 2009.
- OGATA, Katsuhiko. **Engenharia de Controle Moderno. 4.ed.** São Paulo: Pearson-Prentice Hall, 2003.
- RAMALHO, José A. **HTML avançado**. São Paulo: Makron Books, 1997.
- SUNMICROSYSTEMS. **JDK 6 Update 14 with NetBeans 6.7**. Disponível em: <<http://java.sun.com/javase/downloads/index.jsp>> Acesso em: jun. 2009a.
- SUNMICROSYSTEMS. **Java™ 2 Platform Standard Edition 5.0 API Specification**. Disponível em: <<http://java.sun.com/j2se/1.5.0/docs/api/>> Acesso em: jun. 2009b.
- SUNMICROSYSTEMS. **The Java Tutorials**. Disponível em: <<http://java.sun.com/docs/books/tutorial/>> Acesso em: jun. 2009c.