

SCILAB, SCICOS E RLTOOL: *SOFTWARES* LIVRES NO ENSINO DE ENGENHARIA ELÉTRICA

ELAINE DE MATTOS SILVA*, JOSÉ PAULO VILELA SOARES DA CUNHA*

**Departamento de Eletrônica e Telecomunicações — Faculdade de Engenharia
Universidade do Estado do Rio de Janeiro — Rua São Francisco Xavier 524, sala 5036A — 20559-900*

Emails: elaine@lee.eng.uerj.br, jpaulo@ieee.org

Abstract— This paper presents some free softwares for Electrical Engineering education. Emphasis is given on the numerical software Scilab and two of its toolboxes: Scicos, a block-diagram editor for simulation of dynamic systems, and Rltool, which is a design and analysis tool for linear control systems. Some features of Scilab are exemplified through the simulation and control of a buck converter. Scilab and the Linux operating system can form free software environments for professional and academic applications.

Keywords— Control education, Free software, Simulation, Linux, Scilab, Numerical methods.

Resumo— São apresentados alguns *softwares livres* adequados ao ensino de Engenharia Elétrica, enfatizando-se o *software* de cálculo numérico Scilab e dois de seus *toolboxes*: Scicos, simulador de sistemas dinâmicos descritos por diagramas, e Rltool, que integra ferramentas para análise e projeto de sistemas de controle lineares. A aplicação do Scilab é descrita através de um exemplo de simulação e controle de um conversor *buck*. O Scilab e o sistema operacional Linux podem formar ambientes de trabalho para aplicações profissionais e acadêmicas.

Palavras-chave— Educação em controle, *Software* livre, Simulação, Linux, Scilab, Cálculo numérico.

1 Introdução

A redução do custo dos computadores e a melhora do seu desempenho permitiram a difusão do seu uso. Esses progressos do *hardware* vêm acompanhados pelo desenvolvimento de *softwares* de alta qualidade e facilidade de uso para diversas aplicações. Particularmente, os computadores têm assumido um papel importante nas salas de aula e é cada vez mais comum o uso de *softwares* de computação numérica e simulação em diversas disciplinas dos cursos de engenharia. No entanto, enquanto o custo do *hardware* vem sendo reduzido, o alto custo de aquisição e manutenção de *softwares* “proprietários” dificulta a sua utilização quando os recursos financeiros são escassos (Pendharkar, 2005), situação comum nas instituições de ensino e pesquisa brasileiras. Isto vem motivando a ampliação do uso acadêmico de *softwares livres* como, por exemplo, o sistema operacional Linux e, na área de engenharia elétrica, o *software* de cálculo numérico Scilab.

Este artigo relata a experiência que está em desenvolvimento na Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro (FEN/UERJ) com o uso de *softwares livres* como ferramentas computacionais no ensino de disciplinas relacionadas a engenharia elétrica e controle.

A organização deste artigo é descrita a seguir. Na Seção 2, discute-se o uso do sistema operacional GNU/Linux e a reação dos alunos do Laboratório de Engenharia Elétrica (LEE) da UERJ frente a este *software*. Após a apresentação dos *softwares* Scilab e o Scicos nas Seções 3 e 4, descreve-se sucintamente algumas de suas aplicações. Na Seção 5, compara-se alguns métodos de simulação de circuitos elétricos. Na Seção 6,

descreve-se o potencial desses *softwares* na área de controle, enfatizando-se a ferramenta Rltool (Pendharkar, 2005).

2 Sistema Operacional GNU/Linux

O sistema operacional GNU/Linux despertou o interesse acadêmico por sua eficiência, confiabilidade e flexibilidade. Destaca-se seu baixo custo de implantação, uma vez que é livre para cópia e distribuição gratuitas segundo a GPL (*General Public License*). A primeira versão do núcleo (*kernel*) do Linux foi lançada em 1991 e, desde então, com a ajuda do projeto GNU que desenvolve programas que compõem a maioria das distribuições atuais, o sistema GNU/Linux (simplesmente denominado *Linux*) tornou-se a principal alternativa aos sistemas operacionais “proprietários” (Tanenbaum, 1995).

Algumas universidades em países em desenvolvimento como a Índia e o Brasil optaram pelo uso de *softwares livres*. No Brasil observa-se o aumento expressivo do uso desses *softwares* nos cursos de engenharia, porém percebeu-se que na FEN/UERJ ainda não estavam bem difundidos. Para incentivar os alunos ao uso de *softwares livres*, foram criados dois projetos com bolsas para estagiários patrocinadas pela Sub-Reitoria de Graduação da UERJ. No primeiro projeto, instalou-se o Linux nos computadores do LEE e foram elaborados *slides* e uma apostila (Guedes and Silva, 2006), que dão suporte a um curso introdutório ao uso do sistema. Este curso já foi ministrado durante seis meses para alunos de engenharia elétrica. O interesse dos alunos pelo Linux vem sendo verificado pela grande quantidade de inscritos no curso e pela procura de alunos de todas

as modalidades de engenharia. Esses alunos têm se tornado difusores do uso do Linux, outro objetivo deste curso, aumentando o número de usuários deste sistema operacional na FEN/UERJ.

Embora o Linux não tenha sido desenvolvido originalmente para aplicações em *tempo real*, atualmente há versões adaptadas para operar em *tempo real*, e.g., RTAI e RTLinux, sendo a última um *software* comercial. Em (Årzén et al., 2005) descreve-se a aplicação de computadores equipados com sistemas de aquisição de dados e Linux num laboratório para o ensino de controle e sistemas em tempo real. Foram obtidas taxas de amostragem de até 2 kHz (Årzén et al., 2005), satisfatórias para o controle de processos industriais e diversos sistemas eletromecânicos.

3 Scilab

O Scilab é um *software* para cálculo numérico desenvolvido desde 1990 por pesquisadores do INRIA e da ENPC na França e mantido pelo *Scilab Consortium* desde 2003. O Scilab é distribuído gratuitamente através da *Internet* (<http://www.scilab.org>) para diversas plataformas, entre essas o Linux. O Scilab inclui um grande número de bibliotecas (*toolboxes*) que englobam funções gráficas, integração numérica, álgebra linear, otimização e outras. Existem, também, bibliotecas específicas para engenharia, como controle e processamento de sinais. Pode-se expandir o número de funções adicionando-se bibliotecas disponíveis na *Internet*. O Scilab possui uma linguagem de programação própria que permite a criação de programas numéricos (Pires and Rogers, 2002). Atualmente está sendo elaborado o curso “Introdução ao Scilab” para alunos de engenharia elétrica da UERJ. Esse curso é baseado na apostila (Pires, 2004), acrescentando-se tópicos sobre simulação que seguem (Campbell et al., 2005). Os objetivos principais são: (i) rever conceitos de cálculo numérico e simulação para aplicá-los em engenharia elétrica e (ii) apresentar o Scilab tendo-se em vista o curso de engenharia e aplicações profissionais.

4 Scicos

Scicos é uma ferramenta do Scilab que permite a construção de modelos de sistemas dinâmicos através de diagramas de blocos. Diversos blocos estão organizados em grupos (*palettes*) específicos, como os grupos *Sources* (e.g., gerador de eventos, onda senoidal, dente-de-serra), *Sinks* (e.g., osciloscópio, indicador numérico), *Linear* (e.g., integrador, diferenciador, funções de transferência contínuas e discretas) e blocos de elementos elétricos, entre outros. O Scicos possibilita simular modelos onde componentes em tempo contínuo, tempo discreto e orientados por eventos são interconectados,

denominados *sistemas híbridos*. O Scicos ainda possui limitações significativas, que são mais evidentes na construção de diagramas elétricos. No entanto, algumas destas limitações podem ser contornadas, conforme é descrito na Seção 5.3.

5 Simulações com Scilab e Scicos

O Scilab integra um grande número de ferramentas para análise e simulação de modelos de diversos tipos. Para ilustrá-las, utiliza-se como exemplo o conversor *buck* apresentado na Fig. 1.a, inspirado em (Mohan et al., 1995, Seção 4-6). Tem-

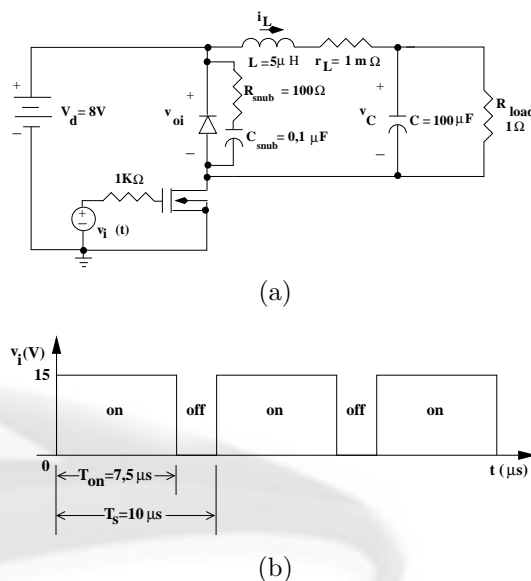


Figura 1: (a) Conversor *buck*. (b) Sinal de controle da chave.

se como objetivo primário avaliar o desempenho transitório desse conversor e, finalmente, o projeto de um controlador para a tensão de saída (v_C). A modelagem será realizada através de três abordagens distintas. Uma destas abordagens foi implementada no Scilab e duas no Scicos, possibilitando a comparação de suas aplicações e limitações. Nas duas primeiras abordagens, o MOSFET, o diodo e a fonte de tensão V_d são substituídos por um sistema de chaveamento controlado ideal, representado pelo gerador de tensão retangular $v_{oi}(t)$ na Fig. 2, c.f. (Mohan et al., 1995, Fig. 4-6).

5.1 Simulação do circuito simplificado

No circuito simplificado (Fig. 2), o sinal de entrada é a tensão $v_{oi}(t)$. Definindo-se como variáveis de estado a corrente no indutor (i_L) e a tensão no capacitor (v_C), obtém-se as equações de estado deste circuito (Mohan et al., 1995, Seção 4-5-1):

$$\frac{di_L(t)}{dt} = \frac{1}{L}v_{oi}(t) - \frac{r_L}{L}i_L(t) - \frac{1}{L}v_C(t), \quad (1)$$

$$\frac{dv_C(t)}{dt} = \frac{1}{C}i_L(t) - \frac{1}{RC}v_C(t), \quad (2)$$

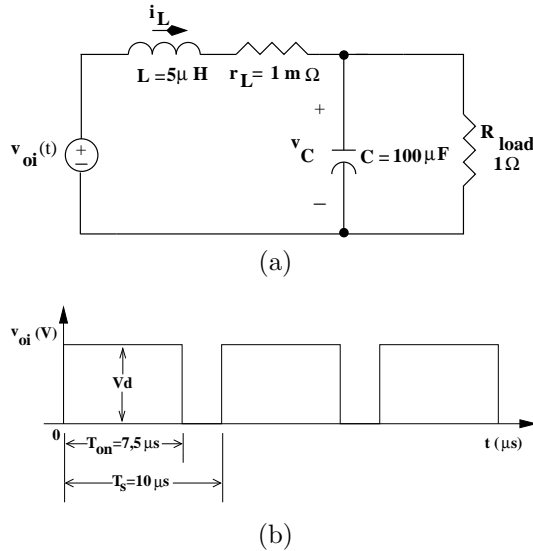


Figura 2: (a) Circuito simplificado. (b) Sinal $v_{oi}(t)$.

que podem ser reescritas na forma matricial:

$$\frac{dx(t)}{dt} = Ax(t) + bv_{oi}(t), \quad (3)$$

onde $x := [i_L, v_C]^T$ é o vetor de estado e

$$A = \begin{bmatrix} -\frac{r_L}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}, \quad b = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}. \quad (4)$$

Esta equação diferencial pode ser resolvida numericamente pelo método trapezoidal de integração codificado em (Mohan et al., 1995, Seção 4-7), utilizando-se comandos elementares que podem ser realizados no Scilab. Uma alternativa mais simples é o algoritmo para o Scilab apresentado na Tabela 1, que utiliza o comando `ode` para resolver equações diferenciais ordinárias.

O sinal de entrada $v_{oi}(t)$ é gerado pela modulação por largura de pulso (*pulse width modulation* — PWM) resultante da comparação de um sinal dente-de-serra $v_{st}(t)$ de período T_s com a tensão de controle $v_{control} = 0,75 V$, conforme a Fig. 3. Quando $v_{control} > v_{st}(t)$, então $v_{oi}(t) = V_d$ (8 V), caso contrário $v_{oi}(t) = 0 V$.

5.2 Simulação orientada pelo diagrama de blocos

Neste exemplo foi simulado o circuito simplificado da Fig. 2 mas, desta vez, foi usado o Scicos para criar o diagrama de blocos da Fig. 4 que representa as equações de estado (1)–(2). A modulação PWM foi realizada conforme descrito na seção anterior. Para melhor organização, o modulador foi implementado no superbloco (*superblock*) detalhado na Fig. 5 e incluído na Fig. 4. Esta implementação leva em consideração que a amplitude do sinal dente-de-serra gerado pelo Scicos é proporcional ao período, o que foi contornado

Tabela 1: Simulação usando o comando `ode`.

```
// Variáveis de estado x1:=iL e x2:=vC.
// Condições iniciais:
t0=0; // Tempo inicial (s).
x0=[4.0 ; 5.5]; // Estado inicial (A,V).
// Parâmetros do modelo:
Vd=8; // Tensão de alimentação (V).
C=100e-6; // Capacitância (F).
L=5e-6; // Indutância (H).
rL=1e-3; // Resistência do indutor (ohms).
R=1; // Resistência de carga (ohms).
// Matrizes A e b da equação de estado:
A=[-rL/L -1/L; 1/C -1/(C*R)];
b=[1/L; 0];
// Parâmetros do tempo e onda retangular:
fs=100e3; // Frequência (Hz).
Ts=1/fs; // Período (s).
tmax=50*Ts; // Duração da simulação (s).
deltat=Ts/50; // Passo de integração (s).
Vcontrol=0.75; // Constante comparada à dente-
// de-serra para gerar a retangular.
t = [t0:deltat:tmax]; // Gera o tempo.
// Definição da função a ser integrada (dx/dt):
function dxdt=f(t,x)
vst=t/Ts-fix(t/Ts); // Onda dente-de-serra.
if Vcontrol>vst then // Modulação PWM.
voi=Vd;
else
voi=0;
end;
dxdt=A*x+b*voi; // Equação de estado.
endfunction
// Resolve a equação diferencial
// via Runge-Kutta RKF45:
x = ode("rkf",x0,t0,t,f);
// Desenha o gráfico da solução:
clf();
plot(t,x);
```

multiplicando-se $v_{control}$ pelo período antes de compará-lo ao sinal dente-de-serra.

Os resultados desta simulação apresentados na Fig. 6 são praticamente os mesmos que foram obtidos pelo uso do comando `ode` na Seção 5.1 com as mesmas condições iniciais ($i_L(0) = 4 A$ e $v_C(0) = 5,5 V$). A comparação entre as simulações realizadas através do modelo representado pelo diagrama de blocos na Fig. 4 e pelo algoritmo na Tabela 1 permite ao aluno selecionar aquela mais adequada às simulações que realizará futuramente.

5.3 Simulação orientada pelo diagrama elétrico

O Scicos possui blocos elétricos e hidráulicos que utilizam a linguagem para modelagem, orientada a objeto, denominada Modelica (Campbell et al., 2005, Capítulo 14). Os conjuntos desses blocos ainda estão em desenvolvimento, o que dificulta a simulação de circuitos elétricos. Ao desenhar o circuito da Fig. 1 no Scicos, percebeu-se

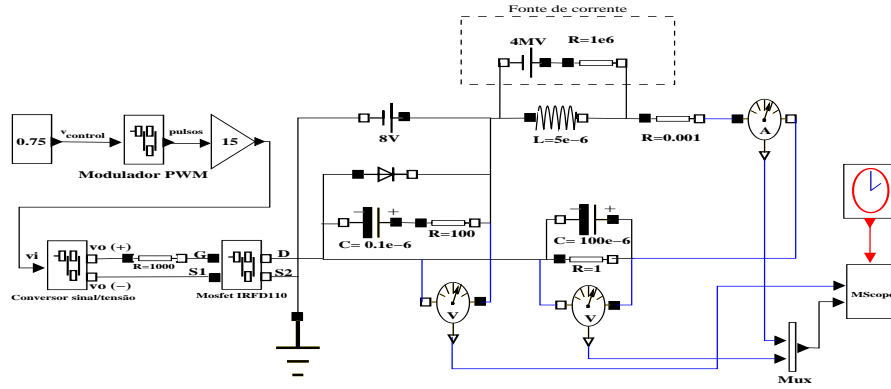


Figura 7: Diagrama elétrico do circuito completo.

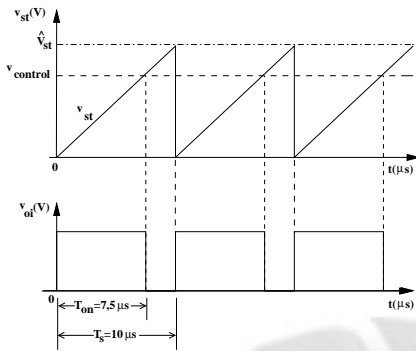


Figura 3: Modulação da largura de pulso para a geração do sinal $v_{oi}(t)$ através da comparação da dente-de-serra $v_{st}(t)$ com $v_{control}$.

que não é permitido definir a corrente inicial no indutor, embora seja possível definir diretamente a tensão inicial no capacitor. Para contornar este problema adicionou-se uma fonte de corrente em paralelo com o indutor, que é formada pela fonte de tensão em série com uma resistência suficientemente grande ($i_L(0) = 4 A = 4 MV/1 M\Omega$, vide destaque na Fig. 7). Além disso, foi necessário modelar o MOSFET utilizando-se parâmetros extraídos do seu manual. O modelo apresentado na Fig. 8 combina a característica estática $i_D(v_{GS}, v_{DS})$ descrita em (Mohan et al., 1995, Seção 22-3) com a dinâmica imposta pelas capacitâncias C_{gs} , C_{gd} e C_{ds} . A simulação realizada através do diagrama elétrico do conversor *buck* na Fig. 7 codificada no Scicos permite a obtenção dos resultados apresentados na Fig. 9. Comparando estes resultados com aqueles obtidos com o circuito simplificado na Fig. 6, o aluno pode perceber as diferenças entre o chaveamento ideal e o obtido com dispositivos de comutação reais, neste caso o MOSFET e o diodo *freewheeling*.

6 Aplicações em Controle

O Scilab possui diversos comandos para a análise, projeto e simulação de sistemas de controle através de técnicas baseadas no espaço de estado, na

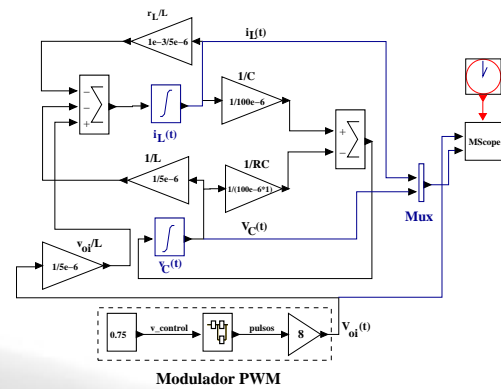


Figura 4: Diagrama que representa a equação de estado do circuito elétrico simplificado.

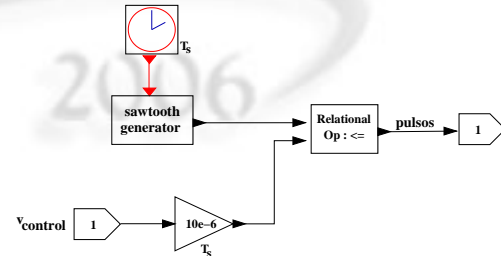


Figura 5: Modulador PWM.

resposta em frequência ou no diagrama do lugar das raízes, dentre outras. Para facilitar a análise e o projeto de sistemas de controle lineares com uma entrada e uma saída, foi desenvolvido o *toolbox* Rltool que integra os comandos do Scilab através de uma interface gráfica (Pendharkar, 2005). Dentre as características do Rltool pode-se citar:

- facilita a verificação da influência dos parâmetros do controlador no lugar das raízes e no desempenho do sistema de controle;
- permite a adição e remoção de pólos e zeros em uma função de transferência usando-se o *mouse*;
- possibilita visualizar simultaneamente o diagrama do lugar das raízes, as respostas no domínio do tempo e da frequência;

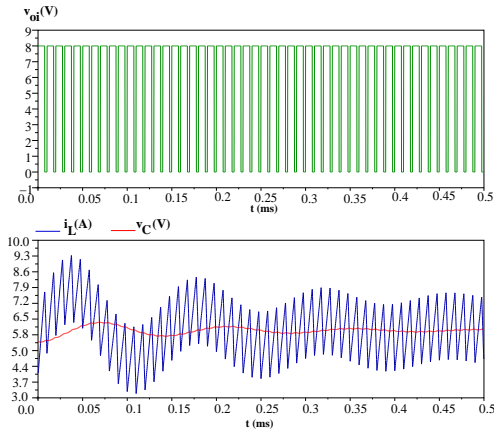


Figura 6: Resultados da simulação do circuito simplificado na Fig. 2.

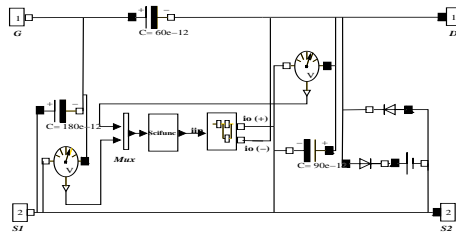


Figura 8: Modelo do MOSFET utilizado na Fig. 7.

- simula a resposta dinâmica da malha fechada para sinais de referência arbitrários.

Originalmente os diagramas de resposta em frequência (e.g., Nyquist, Bode) gerados pelo Scilab utilizam *hertz* como unidade de frequência. Uma vez que a unidade *radianos por segundo* é mais conveniente ao projeto de controladores, o Prof. José Paulo adaptou as rotinas disponíveis no Scilab para esta unidade. Estas rotinas foram incluídas no Rltool, o que é um exemplo da cooperação que o *software livre* possibilita.

Para ilustrar o projeto de controladores auxiliado pelo Scilab, considera-se um regulador para

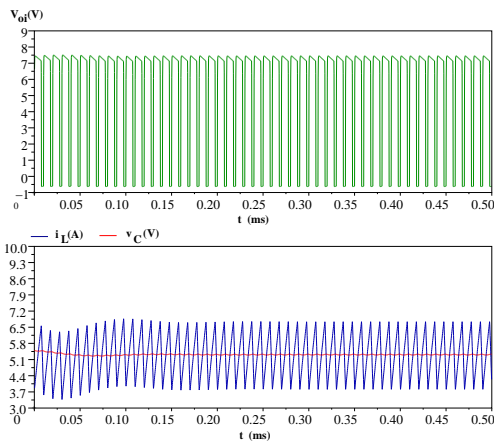


Figura 9: Resultados da simulação do circuito elétrico completo na Fig. 7.

a tensão de saída do conversor *buck* (v_C). O sinal de controle é a tensão de entrada do modulador PWM ($0 \leq v_{control}(t) \leq 1V$). Para realizar o projeto, utiliza-se a tensão média num ciclo gerada pela chave, dada por (Mohan et al., 1995, Seção 7-3)

$$\bar{v}_{oi} = \frac{V_d}{\hat{V}_{st}} v_{control}, \quad (5)$$

onde $\hat{V}_{st} = 1V$ é a tensão de pico da dente-de-serra. Então, computa-se a função de transferência do modelo dada por

$$g_p(s) := \frac{v_C(s)}{v_{control}(s)} = C(sI - A)^{-1}b \frac{V_d}{\hat{V}_{st}}, \quad (6)$$

com $C = [0, 1]$, através do comando do Scilab:

```
-->gp=C*inv(%s*eye(2,2)-A)*b*Vd
gp =
-----
1.600E+10
-----
2
2.002E+09 + 10200s + s
```

Esta função de transferência pode ser utilizada pelo Rltool no projeto do controlador. Aqui se utiliza a ação integral e o cancelamento dos pólos estáveis da planta, resultando na função de transferência do controlador

$$g_c(s) := \frac{v_{control}(s)}{e(s)} = k \frac{[g_p(s)]^{-1}}{s(s + 10^4)}, \quad (7)$$

onde $e = V_{ref} - v_C$ é o sinal de erro, V_{ref} é a tensão de referência e k é o ganho do controlador. A Fig. 10 exibe as janelas do Rltool: a janela de ajuste do ganho, o diagrama do lugar das raízes e a janela com a curva de Bode, a curva de sensibilidade e a resposta ao degrau. O ganho $k = 2,5 \times 10^7$ foi ajustado para se obter uma resposta ao degrau rápida e sem *overshoot*, o que foi facilitado pela boa integração entre os diversos diagramas no Rltool.

Para completar o projeto do regulador, o seu desempenho pode ser avaliado em simulações realizadas pelo Scicos com o diagrama na Fig. 7, ao qual é adicionado o controlador (7). A Fig. 11 apresenta os resultados obtidos com $V_{ref} = 5,0V$ e condições iniciais nulas. Esta simulação é bastante realista pois este modelo não-linear inclui as chaves do conversor *buck*. Assim, comprova-se o bom transitório da tensão regulada (v_C), que neste exemplo é similar à resposta ao degrau do modelo linear que foi obtida pelo Rltool na Fig. 10.

7 Conclusões

Descreveu-se como o Linux, Scilab, Scicos e Rltool podem ser usados no ensino e no projeto de sistemas eletrônicos e de controle. Além de possuírem as vantagens inerentes a *softwares livres*, essas ferramentas se integram bem num único ambiente de

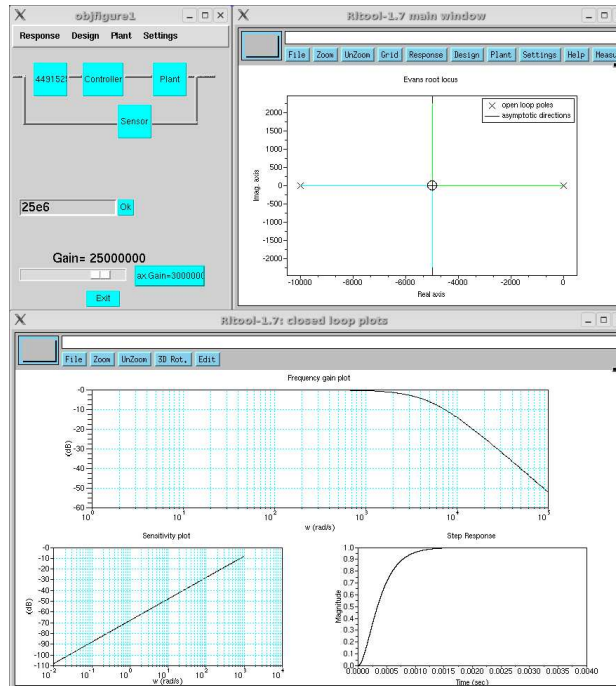


Figura 10: Janelas do Rltool com o resultado final do projeto do regulador de tensão. Os pólos do sistema em malha fechada estão posicionados no centro do diagrama do lugar das raízes, janela superior direita.

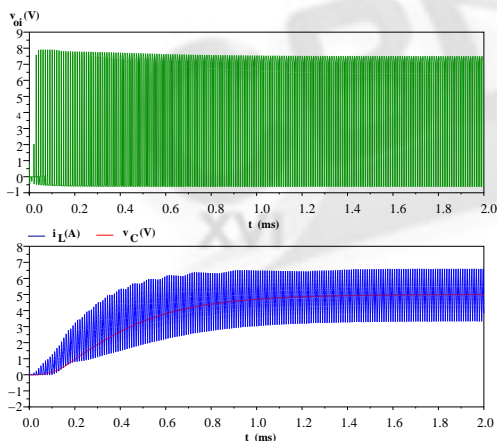


Figura 11: Resultados da simulação do conversor *buck* na Fig. 7 com o regulador de tensão (7).

trabalho, que pode auxiliar o desenvolvimento de sistemas eletrônicos e de controle, como foi apresentado através do exemplo de projeto e simulação de um regulador para a tensão gerada por um conversor *buck*. No entanto, mostrou-se que o simulador Scicos ainda apresenta deficiências significativas, embora seja muito útil ao aluno iniciante. Aqueles acostumados com o Matlab e o simulador Simulink podem sentir dificuldades com os poucos blocos no Scicos e a escassez de *toolboxes* no Scilab. Estes problemas podem ser reduzidos pois o código do Scilab é *aberto*, o que facilita seu desenvolvimento por voluntários. Por outro lado, o Scilab apresenta vantagens em relação ao Matlab como, por exemplo, a manipulação mais amigável

de polinômios, frações racionais e matrizes racionais.

Não é demais comentar que as figuras utilizadas neste texto foram editadas com Xfig, *software livre* para a edição de gráficos vetoriais. As figuras geradas pelo Scilab podem ser armazenadas no formato do Xfig, para então serem editadas e convertidas para diversos formatos gráficos.

Agradecimentos

Este trabalho foi parcialmente financiado pela Faperj e pelo Cetreina/SR-1/UERJ.

Referências

- Årzén, K.-E., Blomdell, A. and Wittenmark, B. (2005). Laboratories and real-time computing, *IEEE Contr. Sys. Mag.* **25**(1): 30–34.
- Campbell, S. L., Chancelier, J.-P. and Nikoukhah, R. (2005). *Modeling and Simulation in Scilab/Scicos*, Springer-Verlag.
- Guedes, R. M. and Silva, E. M. (2006). Introdução ao uso do Linux. Disponível em <http://www.lee.eng.uerj.br/~elaine>.
- Mohan, N., Undeland, T. M. and Robbins, W. P. (1995). *Power Electronics: converters, applications, and design*, 2nd edn, John Wiley & Sons.
- Pendharkar, I. (2005). Rltool for Scilab, *IEEE Contr. Sys. Mag.* **25**(1): 23–25.
- Pires, P. S. M. (2004). Introdução ao Scilab. Disponível em <http://www.dca.ufrn.br/~pmotta>.
- Pires, P. S. M. and Rogers, D. A. (2002). Free/open source software: an alternative for engineering students, *Proc. 32nd ASEE/IEEE Frontiers in Education Conf.*, Boston, MA, pp. T3G-7–T3G-11.
- Tanenbaum, A. S. (1995). *Sistemas Operacionais Modernos*, Prentice-Hall do Brasil, Rio de Janeiro.