Processamento de Sinal (DF-FCUL)

1ª Prática, Set 2014, regime quinzenal

Introdução ao Scilab, modelação e simulação de sistema discretos e contínuos, som (V1.0)

O *Scilab* (http://www.scilab.org/) é um programa de CAM ("Computer-Aided Mathematics") distribuído com uma licença do tipo *open-source* e grátis. O Scilab será utilizado em Processamento de Sinal. O software comercial equiparado é o *Matlab*[1]. Existe uma outra ferramenta grátis e *open-souce* semelhante, o *Octave* (http://www.gnu.org/software/octave/, encontrando-se a versão para *Windows* em http://octave.sourceforge.net/). O Octave tem a ambição de ser um clone do Matlab (em termos de sintaxe). O Octave e o Matlab permitem também realizar as actividades propostas nesta prática (mas o docente não os conhece tão bem). Estas ferramentas podem ser utilizadas em Linux, Mac OS X e noutros sistemas operativos.

Instalação do Scilab

Podem ser instaladas duas versões do Scilab. A versão "oficial" encontra-se em http://www.scilab.org/. A componente gráfica desta versão é escrita em Java, o que a torna lenta. Este problema agrava-se quando se pretende utilizar a interface gráfica de "programação visual" Scicos (http://www.scicoslab.org/), integrada no Scilab, que serve para simular sistemas usando a interligação de blocos discretos e contínuos, lineares e não lineares. É uma ferramenta útil e será utilizada no laboratório.

No site do Scicos pode-se descarregar o *Scicoslab*, uma variante do Scilab/Scicos que usa uma interface gráfica mais antiga mas também mais rápida que o Scilab quando se trata de simular sistemas com blocos.

Dito isto, qualquer dos programas (Scilab ou Scicoslab) poderá ser usado, embora o Scicoslab seja mais rápido no que respeita à componente gráfica. Sugere-se que instale ambos e que escolha o que mais lhe agradar. A qualidade da experiência dependerá em grande medida da velocidade do processador do seu PC!

Objectivos da aula prática

Este laboratório pretende familiarizar os alunos com o Scilab e ensinar várias maneiras de simular sistemas com aquela ferramenta. Para isso sugere-se a **realização das seguintes tarefas:**

- Simulação de um sistema contínuo, quer através da resolução da respectiva equação diferencial, quer usando diagramas de blocos no Scicos em termos do modelo das variáveis de estado (e da transformada de Laplace, quando esta matéria for dada);

- Simulação de um sistema discreto, quer realizando a operação da convolução (escrevendo um programa), quer usando diagramas de blocos no Scicos em termos do modelo das variáveis de estado (MVE) discreto (quer usando a transformada Z, cujos blocos – fracções racionais em z – são suportados pelo Scicos, quando esta matéria for dada);

- Familiarização com as técnicas de base no Scilab: utilização do editor, realização de diagramas de blocos com o Scicos, representação de vectores, apresentação gráfica de sinais, e outras.

- Apresentação de algumas funções de processamento de som no Scilab. Com elas poder-se-á "ouvir" o Teorema da Amostragem.

Os detalhes sobre os comandos do Scilab deverão ser consultados no campo "Help" da barra de menus da janela. No final deste guião serão apontados tutoriais sobre o Scilab e o Scicos.

Simulação de sistemas contínuos

Vai-se simular o sistema, já resolvido numa aula prática, que consiste de um objecto de massa (M = 1 Kg), sujeito a atrito viscoso (B = 7 N.s/m) e fixo por uma mola (K = 12,5 N/m) a um referencial imóvel. O objecto sofre o efeito de uma força externa $F = 2\,U(t)$ N. As condições iniciais são velocidade nula ($v(0) = dx(t)/dt|_{t=0} = 0$ m/s) e posição $x(0) = 0.1$ m. A equação que rege a posição $x(t)$ do objecto é:

$$M\,x''(t) + B\,x'(t) + K\,x(t) = F(t)$$

- Simule a posição da massa no sistema usando directamente a equação diferencial. Escolha parâmetros de simulação adequados às constantes de tempo do sistema. Visualize $x(t)$ e $x'(t)$ até ao desaparecimento da componente transitória daquelas variáveis.

- Repita, usando agora diagramas de blocos do Scicos que permitem representar funções racionais em s (funções de transferência) ou o modelo às variáveis de estado (MVE). Os modelos esm s já foram estudados em circuitos e electrónica e também podem ser associados à transformada de Laplace.

[1] O Matlab (http://www.mathworks.com/) é dispendioso.

Simulação de sistemas discretos

Considere a função de sistema $H(z) = \frac{11\,z^2 - 9.5\,z}{(z-0.5)(z-0.9)}$. Obtenha a correspondente equação às diferenças.

> Calcule e visualize $r[n]$, a resposta do sistema ao degrau $U[n]$, usando: (1) um pequeno programa em Scilab implementando a equação às diferenças, que é uma equação recursiva; (2) a função de convolução do Scilab `convol()` (isto só aprenderá a fazer quando estudar a transformada Z); (3) os diagramas de blocos do Scicos. Ajuste os parâmetros de simulação para valores adequados. Presume-se condições iniciais nulas.

Sinais "chirp"

A frequência da sinusóide $x(t) = \sin(2\pi\, f\, t + \theta)$ é f ou $\omega = 2\pi f$ (frequência angular). Mas qual é a frequência de $y(t) = \sin[\psi(t)]$, onde $\psi(t)$ é uma função qualquer? Possivelmente não saberá a resposta...

De facto a frequência pode ser variável, ou seja $f \equiv f(t)$ e $\omega \equiv \omega(t)$, pois a definição geral desta é

$$\omega(t) = 2\pi f(t) = \frac{d\psi(t)}{dt}$$

pelo, que para uma frequência $f(t)$ pretendida (em Hz), a respectiva sinusóide é criada por

$$\sin(2\pi \int_0^t f(t')dt')$$

Se quisermos uma frequência constante, $f(t) = f_0$, a respectiva sinusóide será calculada porsin$(2\pi \int_0^t f_0 dt') = \sin(2\pi f_0 t)$, confirmando o que se esperava.

A função $\psi(t)$ é a **fase** (em sentido generalizado) da sinusóide e, assim, **a frequência (angular) é a derivada da fase**.

Com este esclarecimento, já se pode definir um sinal *chirp*. Os tipos mais comuns de *chirp* são o chirp linear e o chirp exponencial, cujas frequências são descritas respectivamente por

$$f_\ell(t) = f_0 + kt \qquad f_e(t) = f_0\,\alpha^t$$

Os *chirps* linear (ver a fig. 1) e exponencial serão então dados por

$$x_\ell(t) = \sin[2\pi(f_0{+}kt/2)\,t] \qquad x_e(t) = \sin\left[\frac{2\pi f_0}{\ln(\alpha)}(\alpha^t - 1)\right]$$

Mais adiante o *chirp* linear será utilizado para ilustrar a sobreposição espectral (ou *aliasing*).

> Tarefa: criar (e ouvir) um *chirp* com alguns segundos de duração. Escolha uma frequência de amostragem adequada ao processamento de sinais audio.

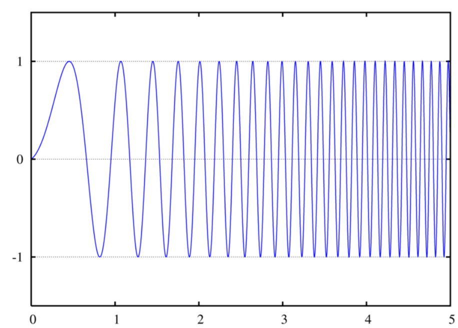

Figure 1: Ilustração de um chirp linear.

Funções de Processamento de Som no Scilab

Estas funções encontram-se agrupadas na pasta do menu *"Help→Sound File Handling"*. As mais interessantes são `sound(y[,fs])`, `playsnd(y,[rate,bits[,command]])`, `loadwave(filename)`, `wavread(wavfile)`, `savewave(filename, x[,rate])`, `wavwrite(y,Fs,wavfile)` e `tv=soundsec(n[,rate])`. Infelizmente o Scilab usa uma notação inconsistente: as variáveis `fs`, `Fs` e `rate` designam todas elas a frequência de amostragem.

Por defeito, se não for indicado um diferente valor, `fs` é considerada no Scilab como sendo 22050 Hz (metade dos 44.1 KHz usados na gravação dos CDs audio comuns).

Neste trabalho, presuma e use $fs = 8$ KHz a não ser que lhe seja recomendado um outro valor.

Algumas das funções atrás mencionadas têm a mesma funcionalidade (e.g. `savewave()` e `wavwrite()`). Aquelas que iremos utilizar mais vezes são `sound()`, que reproduz o som, e `soundsec()`, que gera[2] um vector `tv` preenchido com instantes de tempo correspondentes a n segundos reais amostrados à frequência fs.

Eco

O eco não é mais do que um sinal invertido, atrasado e atenuado. As figuras 2, 3 e 4 ilustram o eco em termos de processamento de sinal (presumindo apenas uma refexão).

[2] A função `soundsec()` tem um *bug* que define instantes errados quando a frequência de amostragem é pequena.

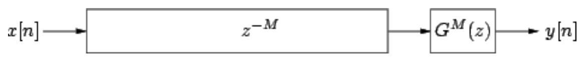

Figure 2: Propagação com atraso e atenuação.

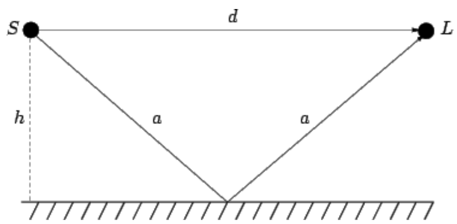

Figure 3: Esquema da propagação com eco.

Com fs = 8 KHz e supondo um "reflector de som" localizado a 100 m de distância, simule o eco. Crie um sinal com 10 s de duração em que apenas os primeiros 5% das amostras são uma sinusóide de 200 Hz e os restantes 95% são nulos. Lembre-se que a velocidade de propagação do som no ar é aproximadamente 340 m/s. Teste várias atenuações g (veja a fig. 4). Pode testar também reflectores múltiplos. Em suma, use a sua imaginação para simular e explorar o modelo.

Sobreposição espectral (*aliasing*)

Projecte a constante k de um *chirp* linear, com frequência inicial $f_0 = 200$ Hz, de forma que a frequência seja 1 KHz em $t = 1$ s. Crie sinais com 1, 5 e 10 seg. de duração e reproduza-os. Comente o que ouve à luz do *Teorema da Amostragem* (ou *Teorema de Nyquist*). Repita a audição do sinal com 10 segundos de duração após substituir o k que calculou anteriormente por um décimo do seu valor.

Recursos para consulta

Caso tenha tempo instale e dê uma vista de olhos ao Scilab ou ao Scicoslab, fazendo a pesquisa do respectivo "Help" que está bem organizado. Alguns recursos de consulta úteis são:

- Tutorial do Scicos em http://www.scicos.org/TUTORIAL/tutorial.html e em pdf aqui: http://tic.javeriana.edu.co/apps/Manuales/Scilab/Scicos/Tutorial-scicos.pdf.

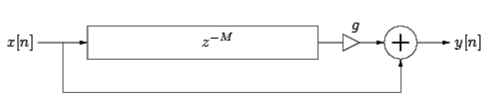

Figure 4: Diagrama de blocos da propagação com eco.

- Capítulo de um livro sobre o Scilab (veja a referência mais adiante) dedicado ao Scicos http://www.scicos.org/book/ScilabScicosBook-Ch7.pdf.
- Tutorial sobre o Scilab: http://comptlsci.anu.edu.au/Numerical-Methods/tutorial-all.pdf.
- Descrição das "entranhas" do Scicos: http://www.scicos.org/p.pdf.
- Ligação do Scicos ao projecto digital (i.e. Booleano, com registos, flip-flops, etc...) aqui http://scicoshdl.sourceforge.net/newhttpd/doc/ug_scicos_hdl
- Livro sobre simulação com o Scilab/Scicos http://www.scicos.org/book.html.
- Manual do Scilab (2300 páginas, 6,5 Mbytes): http://www.scilab.org/download/5.1.1/manual_scilab-5.1.1_en_US.pdf.