Processamento de Sinal (DF-FCUL)

Novembro de 2014

Filtros IIR (V1.0)

Há duas grandes classes de filtros digitais: os filtros IIR ("Infinite Impulse Response") e os filtros FIR ("Finite Impulse Response"). Nesta aula são focados os filtros IIR, cuja função de transferência $H(z)$ é escrita, na forma geral, como:

$$H(z) = \frac{Y(z)}{U(z)} = \frac{\sum_{k=0}^{M} b_k\, z^{-k}}{1 + \sum_{k=1}^{N} a_k\, z^{-k}} \tag{1}$$

(onde se convencionou que $a_0 = 1$). A implementação do filtro IIR assenta na correspondente equação às diferenças (entrada u, saída y)

$$y[n] = \sum_{k=0}^{M} b_k u[n-k] - \sum_{k=1}^{N} a_k y[n-k] \tag{2}$$

O cálculo dos parâmetros a_k e b_k dos filtros IIR é habitualmente feito a partir de um *protótipo analógico*, $H_a(s)$, projectado pelas técnicas já estudadas nas disciplinas de electrónica, ao qual se aplica uma transformação genérica $s \to f(z)$ para obter $H(z)$, ou seja, $H(z) = H_a[f(z)]$. As três transformações mais utilizadas na prática são:

- a *aproximação da derivada*, em que se define

$$s = \frac{1 - z^{-1}}{T}$$

- a *transformação bilinear*, em que

$$s = \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}}$$

- a *invariância ao impulso* que, quando $H_a(s)$ pode ser decomposta em fracções parciais apenas com pólos simples, corresponde ao seguinte mapeamento do domínio contínuo (em s) para o domínio discreto (em z):

$$H_a(s) = \sum_{k=1}^{N} \frac{A_k}{s - p_k} \quad \to \quad H(z) = T \sum_{k=1}^{N} \frac{A_k}{1 - e^{p_k T} z^{-1}}$$

  onde T é o período de amostragem, e T multiplica o somatório em $H(z)$ para compensar o erro de ganho inerente à transformação. Existem expressões semelhantes à anterior para mapear automaticamente, do domínio s para o domínio z, fracções em cujo denominador se encontram polinómios do 2º grau em s, ou que têm pólos múltiplos. (Note que *na invariância ao impulso não se aplicou directamente uma transformação $s \to f(z)$*, sendo esta transformação diferente, no conceito, das transformação bilinear e aproximação da derivada pela diferença regressiva.)

Finalmente, recorda-se que **sempre que é usada a *transformação bilinear* há que fazer um *pre-warping*** da frequência do filtro analógico utilizado como protótipo. Sendo $s = j\Omega$ e $z = e^{j\omega}$ (onde $\omega = 2\pi f/f_s$ é a frequência angular digital normalizada) respectivamente as frequências complexas analógica e digital, a relação entre Ω e ω na transformação bilinear é:

$$\Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}\right) \qquad \text{ou} \qquad \omega = 2\arctan\left(\frac{\Omega T}{2}\right)$$

pelo que se, por exemplo, quisermos realizar um filtro passa-baixo digital com frequência de corte ω_C, a frequência (não normalizada) de corte do protótipo analógico onde o projecto assentará, Ω_C, deverá ser calculada com as anteriores expressões. **A especificação inicial da(s) frequência(s) do filtro é feita no domínio digital** (por exemplo, sendo a frequência de amostragem $f_S = 10$ kHz, e pretendendo-se realizar um filtro passa-baixo com $f_C = 3$ kHz, a frequência de corte do protótipo analógico será $\Omega_C = 2 f_S \tan(\pi f_C / f_S)$.)

A transformação bilinear é a técnica de projecto de filtros IIR mais popular.

Objectivos da aula prática

Nesta aula pretende-se ensinar os alunos a projectar filtros digitais do tipo IIR, de ordem moderada, usando como auxiliar de projecto a ferramenta computacional disponível, o Scilab. As capacidades do Scilab na visualização de gráficos ajudarão na verificação da correcção dos projectos. Os alunos podem comparar as diversas técnicas de mapeamento fazendo implementações de um mesmo protótipo analógico com cada uma delas. Deve ser recordado que o *aliasing* está sempre presente na implementação dos filtros digitais e, por isso, deverão estar atentos a este aspecto.

Projecto de filtros analógicos

As técnicas de *projecto de filtros analógicos* consistem, em termos genéricos, dos seguintes procedimentos: (i) analisar o problema real e estabelecer as especificações (ou requisitos) de projecto do filtro; (ii) escolher uma família de filtros, ou aproximação (Butterworth, Chebyshev, Bessel,...), com propriedades adequadas à satisfação dos requisitos; (iii) determinar a ordem do filtro e escrever a sua função passa-baixo normalizada a partir de tabelas (de polinómios, ou de pólos, veja as figuras neste documento); (iv) aplicar a transformação da frequência normalizada (no passa-baixo) para a frequência real do filtro a projectar (passa-alto, passa-banda ou rejeita-banda), obtendo-se assim a função de transferência real $H_a(s)$; (v) finalmente, implementar o filtro com um circuito electrónico apropriado.

No presente trabalho o último passo é obliterado, sendo **substituído pela implementação com filtro IIR digital**, projectado com as técnicas mencionadas.

No final do documento encontram-se algumas tabelas com filtros Butterworth passa-baixo normalizados a Ω_C'=1.

São também indicadas as transformações de frequência no domínio analógico.

Trabalho prático

Em aulas anteriores foi mencionado o filtro digital passa-baixo de 1ª ordem projectado com a técnica da aproximação da derivada. Agora vai-se lidar com filtros "a sério"... Considere $T^{-1} = f_S = 10$ KHz.

1. Projecte um *filtro passa-baixo digital* IIR de 3ª ou 4ª ordem para separar duas frequências, uma baixa e outra alta, por exemplo 113 Hz e 1277 Hz (estas frequências estão relacionadas com a marcação DTMF usada nos telefones). Especifique um valor razoável para a frequência de corte do filtro. Use a transformação bilinear no projecto, tendo o cuidado de fazer o *pre-warping*. Observe os resultados com gráficos.

2. De seguida, usando um protótipo de 2ª ordem, crie um filtro passa-banda analógico (de 4ª ordem) e a partir dele projecte o seu irmão digital, também com a transformação bilinear. O objectivo é realçar a frequência central de um sinal constituído pela soma de 3 sinusóides de frequências diferentes (e.g. 113 Hz, 451 Hz e 1277 Hz).

3. Outro uso que pode dar a um passa-banda é na realização de um **multiplicador de frequência**: filtre uma onda quadrada de forma a deixar passar apenas uma das harmónicas superiores (defina a frequência central do filtro de acordo com a harmónica que quer deixar passar).

Use o Scilab nos cálculos auxiliares (e.g. para calcular zeros de polinómios) e na manipulação simbólica de polinómios para implementar as transformações. Se pesquisar o *"help"* do Scilab vai encontrar uma série de funções dedicadas ao projecto de filtros analógicos e digitais.
Aquelas especificações do filtro não são rígidas, isto é, pode ser criativo e variar as frequências dos sinais e da amostragem, o tipo e ordem do filtro, etc...
Realize o filtro com uma sequência de iterações (isto é, com um ciclo "for") que implementa a equação às diferenças correspondente. Pondere a sua realização com a convolução.

Anexo

Na fig. 1 encontra-se uma tabela com coeficientes (a_n) de filtros de Butterworth passa-baixo normalizados. A correspondente função de transferência, para uma ordem n, é

$$H_a(S,n) = \frac{1}{S^n + a_{n-1}S^{n-1} + ... + a_1 S + 1} \qquad (3)$$

Uma alternativa útil são os polinómios já factorizados, como se mostra na fig. 2. Neste caso é mais simples a implementação do filtro em cascata ou em paralelo.

n	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
2	1.414214								
3	2.000000	2.000000							
4	2.613126	3.414214	2.613126						
5	3.236068	5.236068	5.236068	3.236068					
6	3.863703	7.464102	9.141620	7.464102	3.863703				
7	4.493959	10.097835	14.591794	14.591794	10.097835	4.493959			
8	5.125831	13.137071	21.846151	25.688356	21.846151	13.137071	5.125831		
9	5.758770	16.581719	31.163437	41.986386	41.986386	31.163437	16.581719	5.758770	
10	6.392453	20.431729	42.802061	64.882396	74.233429	64.882396	42.802061	20.431729	6.392453

Fig. 1: Tabela com coeficientes a_n de filtros de Butterworth passa-baixo normalizados (eq. 3).

n	Factors of Polynomial $B_n(s)$
1	$(s+1)$
2	$s^2 + 1.4142s + 1$
3	$(s+1)(s^2+s+1)$
4	$(s^2 + 0.7654s + 1)(s^2 + 1.8478s + 1)$
5	$(s+1)(s^2 + 0.6180s + 1)(s^2 + 1.6180s + 1)$
6	$(s^2 + 0.5176s + 1)(s^2 + 1.4142s + 1)(s^2 + 1.9319s + 1)$
7	$(s+1)(s^2 + 0.4450s + 1)(s^2 + 1.2470s + 1)(s^2 + 1.8019s + 1)$
8	$(s^2 + 0.3902s + 1)(s^2 + 1.1111s + 1)(s^2 + 1.6629s + 1)(s^2 + 1.9616s + 1)$

Fig. 2: Tabela de filtros de Butterworth passa-baixo normalizados (aqui a frequência normalizada analógica é o s minúsculo, contrariamente ao texto em que se usa o S maiúsculo), factorizados em polinómios do primeiro e do segundo grau.

As transformações da *frequência analógica normalizada* $S = j\Omega'$ para a frequência analógica não normalizada, $s = j\Omega$, nos casos passa-baixo (LP), passa-alto (HP), passa-banda (BP) e rejeita-banda (BR), onde $\Omega_0 = \sqrt{\Omega_1 \Omega_2}$, são:

$$S = s/\Omega_C \quad (LP) \qquad\qquad S = \Omega_C/s \quad (HP)$$

$$S = \frac{\Omega_0}{\Omega_2 - \Omega_1}\left(\frac{\Omega_0}{s} + \frac{s}{\Omega_0}\right) = \frac{s^2 + \Omega_0^2}{(\Omega_2 - \Omega_1)s} \quad (BP)$$

$$S = \frac{(\Omega_2 - \Omega_1)s}{s^2 + \Omega_0^2} \quad (BR)$$

Alguns comandos úteis no projecto de filtros IIR (consulte detalhes no *"help"* do Scilab, começando pelo item "Signal" na "Signal Processing Toolbox" do *"help"* do Scilab):
`poly() numer() denom() analpf() eqiir() frmag() iir()`