Exemples bàsics de Processament Digital de Senyal amb Scilab

Scilab és un software de càlcul científic que conté eines per al tractament de senyals (representació gràfica, càlcul de transformades de Fourier i moltes més). Es pot considerar un "clon" de Matlab, el paquet comercial de software matemàtic més utilitzat arreu del mon, amb l'avantatge d'ésser gratuit (software lliure). Scilab es pot instal.lar en els sistemes operatius més habituals (Windows, Linux, MacOS). Per a més informació sobre Scilab i la seva instal.lació: `http://www.scilab.org/`.

En aquest document mostram alguns exemples de processament bàsic de senyals de veu amb Scilab. Els senyals de veu (o música) utilitzats es troben en format .wav i s'han obtingut d'Internet (`http://www.thefreesite.com/Free_Sounds/Free_WAVs/` i `http://www.moviewavs.com/`). També és possible crear fitxers .wav amb la gravadora de sons de Windows (al menú Todos los programas/Accesorios/Entretenimiento).

Començam per executar Scilab. S'obre una finestra (consola Scilab) en la qual escriurem les comandes. Obrim un senyal de veu emmagatzemat a l'ordinador:

```
-->[y,Fs,bits]=wavread("G:\pdsTelem\curs_09_10\practicaScilab\jedi3.wav");
```

Els valors del senyal queden emmagatzemats en l'array y. $y[1]$ conté la primera mostra del senyal, $y[2]$ la segona i així successivament. El nombre total de mostres (longitud de l'array) es pot calcular fent

```
-->Nt=length(y)
 Nt  =

    77295.
```

F_s és la freqüència de mostreig del senyal (Hz). $F_s = 1/T$, on T és el periode de mostreig:

```
-->Fs
 Fs  =

    11025.
```

El senyal es pot escoltar per comprovar que s'ha carregat correctament i també representar gràficament:

```
-->playsnd(y, Fs);

-->scf(0), plot2d3(y);
```

A continuació seleccionam una part del senyal original, la compresa entre les mostres 1 i 30000. Reproduïm el bocí de senyal, el representam gràficament i el guardam (afegint el sufix "short" al nom original).

```
-->y2=y(1:30000);

-->playsnd(y2, Fs);

-->scf(1), plot2d3(y2);

-->wavwrite(y2, Fs, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short.wav");

-->N=length(y2)
 N  =

    30000.
```

Suposem que volem afegir una música de fons al senyal de veu. Per fer això obrim un nou fitxer .wav que contengui la música desitjada i ens asseguram que la frequència de mostreig sigui la mateixa que la del senyal inicial. A continuació seleccionam un bocí de la mateixa longitud que el senyal inicial i cream un nou senyal resultant de la suma dels dos senyals (perquè la música no soni amb un volum més baix que la veu reduïm el valor del senyal de música en un factor 2). Finalment escoltam el resultat final i guardam el resultat.

```
-->[x,Fsx,bits]=wavread("G:\pdsTelem\curs_09_10\practicaScilab\imperial.wav");

-->Fsx
 Fsx  =

    11025.

-->scf(2), plot2d3(x);

-->x2=x(6500+1:6500+N);

-->xy=0.5*x2+y2;

-->playsnd(xy, Fs);

-->wavwrite(xy, Fs, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_imperial.wav");
```

També podem comprovar l'efecte de reproduir un senyal amb una freqüència de mostreig diferent de l'original:

```
-->playsnd(y2, 2*Fs);
-->playsnd(y2, Fs/2);

-->wavwrite(y2, 2*Fs, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_fast.wav");
-->wavwrite(y2, Fs/2, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_slow.wav");
```

Observam el canvi de freqüència i de durada dels senyals resultants.

Una eina molt útil per a l'anàlisi de senyals és la FFT (Fast Fourier Transform), una implementació ràpida de la transformada de Fourier discreta (DFT) i que ha contribuït de manera decisiva al desenvolupament del Processament Digital de Senyals dels darrers 40 anys. En tots cas, a nivell teòric no hi ha cap diferència entre la FFT i la DFT. Recordem algunes propietats de la transformada de Fourier de senyals discrets que ens permetran entendre els resultats que obtindrem:

- La DFT d'un senyal discret (periòdic) $x[n]$ de N mostres (periode N) és un nou senyal discret $X[k]$ de N mostres.
- $X[k]$ és un senyal periòdic de periode N.
- La transformada de Fourier (en temps discret) d'un senyal discret (no periòdic) de N mostres és un senyal continu $X(\omega)$.
- $X(\omega)$ és un senyal periòdic de periode 2π.
- La relació entre $X[k]$ i $X(\omega)$ és: $X[k] = \frac{1}{N}X(\frac{2\pi}{N}k)$, per a $k = 0, 1, \cdots, N-1$. És a dir, $X[k]$ s'obté per discretització de $X(\omega)$, agafant N mostres del periode principal.
- El senyal discret $x[n]$ es considera ben mostrejat si el periode de mostreig T cumpleix la condició de Nyquist: $1/T \geq 2B$, on B és la freqüència màxima del senyal continu original $x_a(t)$. Això equival a dir $B \leq F_s/2$, és a dir, que la freqüència màxima del senyal continu és inferior a la meitat de la freqüència de mostreig ($F_s = 1/T$). En aquest cas no hi ha aliasing i el senyal original es pot recuperar a partir de les mostres.
- Per a un senyal ben mostrejat $X_a(\Omega) = TX(\omega)$ entre $\omega = -\pi$ i $\omega = \pi$.
- La relació entre les freqüències discretes ω i les freqüències contínues Ω és $\Omega T = \omega$, on T és el periode de mostreig. Això significa que $NX[k]$ és el valor de la component espectral de freqüència angular $\Omega = \frac{2\pi}{NT}k = \frac{2\pi F_s}{N}k$ del senyal continu original (on F_s és la freqüència de mostreig). En freqüència (no angular): $F = \frac{k}{NT} = k\frac{F_s}{N}$.

- La transformada de Fourier $X(\omega)$ d'un senyal $x[n]$ real té les següents propietats de simetria: $|X(-\omega)| = |X(\omega)|$, fase$(X(-\omega))$ = $-$fase$(X(\omega))$.

A continacíó calculam i representam la FFT del senyal de veu:

```
-->y2FFT=fft(y2);
-->scf(3), plot2d3(abs(y2FFT));
```

Observem la simetria de l'espectre, degut a que el senyal de veu pren valors reals.

Ara comprovam l'efecte d'eliminar les components espectrals d'una banda de freqüències. Per exemple eliminam la banda compresa entre els indexos $k_1 = 5000$ i $k_2 = 10000$. Aquestes indexos corresponen a les freqüències continues $\frac{(k_1-1)\cdot Fs}{N} = 1837.1325$Hz i $\frac{(k_2-1)\cdot Fs}{N} = 3674.6325$Hz. (El factor $k-1$ s'emplea perquè amb la notació utilitzada en Scilab pels arrays la mostra k té index $k+1$). Observem que per conservar la simetria de la FFT també s'han d'eliminar les freqüencies compreses entre els indexos $N+2-k_2 = 20002$ i $N+2-k_1 = 25002$.

```
-->y3FFT=y2FFT;
-->k1=5000;
-->k2=10000;
-->Fs*(k1-1)/N
 ans  =

    1837.1325

-->Fs*(k2-1)/N
 ans  =

    3674.6325

-->y3FFT(k1:k2)=0;
-->y3FFT((N+2-k2):(N+2-k1))=0;
-->scf(4), plot2d3(abs(y3FFT));

-->y3=real(ifft(y3FFT));
-->scf(5), plot2d3(y3);
-->playsnd(y3, Fs);
-->wavwrite(y3, Fs, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_bandaElim.wav");
```

A continuació comprovam l'efecte d'amplificar les components espectrals d'una banda de freqüències (en particular multiplicam per 2 els valors originals). Per exemple amplificam la banda compresa entre els indexos $k_1 = 5000$ i $k_2 = 10000$.

```
-->y4FFT=y2FFT;
-->k1=5000;
-->k2=10000;
-->Fs*(k1-1)/N
 ans  =

    1837.1325

-->Fs*(k2-1)/N
 ans  =

    3674.6325

-->y4FFT(k1:k2)=2*y4FFT(k1:k2);
-->y4FFT((N+2-k2):(N+2-k1))=2*y4FFT((N+2-k2):(N+2-k1));
```

```
-->scf(6), plot2d3(abs(y4FFT));

-->y4=real(ifft(y4FFT));
-->scf(7), plot2d3(y4);
-->playsnd(y4, Fs);
-->wavwrite(y4, Fs, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_bandaAmpl.wav");
```

Una altra modificació que podem fer és assignar un valor constant a totes les components espectrals d'una banda de freqüències. Per exemple assignam el valor 100 a la banda compresa entre els indexos $k_1 = 4000$ i $k_2 = 5000$. Aquestes indexos corresponen a les freqüències continues $\frac{(k_1-1)\cdot Fs}{N} = 1469.6325$Hz i $\frac{(k_2-1)\cdot Fs}{N} = 1837.1325$Hz. Observem que per conservar la simetria de la FFT també s'han d'amplificar les freqüencies compreses entre els indexos $N + 2 - k_2 = 25002$ i $N + 2 - k_1 = 26002$.

```
-->y5FFT=y2FFT;
-->k1=4000;
-->k2=5000;
-->Fs*(k1-1)/N
 ans  =

    1470.

-->Fs*(k2-1)/N
 ans  =

    1837.5

-->y5FFT(k1:k2)=100;
-->y5FFT((N+2-k2):(N+2-k1))=100;
-->scf(8), plot2d3(abs(y5FFT));

-->y5=real(ifft(y5FFT));
-->scf(9), plot2d3(y5);
-->playsnd(y5, Fs);
-->wavwrite(y5, Fs, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_bandaAdd.wav");
```

Finalment estudiam l'efecte de submostrejar (o delmar) el senyal original, és a dir, utilitzar un nombre menor de mostres per representar-lo. Per exemple podem prendre 1 de cada 2 mostres del senyal:

```
-->y6=y2(1:2:N);
-->scf(10), plot2d3(y6);
-->playsnd(y6, Fs/2);
-->wavwrite(y6, Fs/2, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_delmat2.wav");
```

Com a conseqüència del submostreig el nou senyal ocupa la meitat que el senyal original (compressió), no obstant això apreciam uns certa distorsió respecte a l'original quan escoltam el senyal.

Per entendre perquè s'ha produit la distorsió hem de pensar que un submostreig de factor 2 és equivalent a un mostreig del senyal continu original amb un periode $2T$ (per aquest motiu s'ha utilitzat una freqüència $F_s/2$ a l'hora de reproduir el senyal). De manera que en l'espectre del senyal resultant les repeticions de l'espectre original es troben separades $\frac{2\pi}{2T} = \frac{2\pi F_s}{2}$ en lloc de $\frac{2\pi}{T} = 2\pi F_s$ i això produeix un efecte d'**aliasing**. Podem observar l'efecte si representam la DFT del senyal submostrejat:

```
-->y6FFT=fft(y6);
-->scf(11), plot2d3(abs(y6FFT));
```

Una manera d'evitar l'efecte de l'aliasing és limitar la freqüència màxima del senyal original a $F_s/4$. D'aquesta manera asseguram que les repeticions (separades una distància $F_s/2$ no es solaparan. Per limitar la freqüència màxima a $F_s/4$ posam a zero (eliminam) les components espectrals de freqüències

més altes (aquesta operació rep el nom de **filtratge anti-aliasing**. Això modifica el senyal original però fa que el resultat del submostreig no sigui tan dolent que quan no es fa el filtratge anti-aliasing (malgrat que en el cas de l'exemple no s'aprecia massa la diferència).

```
-->y7FFT=y2FFT;
-->k1=1+N/4;
-->k2=1+N/2;
-->Fs*(k1-1)/N
 ans  =

    2756.25
-->Fs*(k2-1)/N
 ans  =

    5512.5
-->y7FFT(k1:k2)=0;
-->y7FFT((N+2-k2):(N+2-k1))=0;
-->scf(12), plot2d3(abs(y7FFT));
-->y7=real(ifft(y7FFT));
-->y8=y7(1:2:N);
-->playsnd(y8, Fs/2);
-->y8FFT=fft(y8);
-->scf(13), plot2d3(abs(y8FFT));
-->wavwrite(y8, Fs/2, "G:\pdsTelem\curs_09_10\practicaScilab\jedi3_short_delmat2F.wav");
```

Per últim comentar que l'efecte de l'aliasing degut al submostreig és més evident en imatges digitals que en senyals de veu. Les següents imatges mostres es resultat de submostrejar una imatge original sense i amb filtre anti-aliasing.

Figura 1: Imatge original (a dalt) i dues versions sub-mostrejades (a baix) (s'ha prés un pixel de cada quatre). El sub-mostreig equival a utilitzar un període de mostreig 4 vegades superior a l'original. La conseqüència és l'aliasing que es pot observar en la imatge de inferior esquerra. L'aplicació d'un filtre anti-aliasing evita aquest fenòmen (imatge inferior dreta).