Departament de Disseny i Programació de Sistemes Electrònics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

La transformada discreta de Fourier

Espectre d'un senyal PWM abans i despres de filtrar

Jordi Bonet i Dalmau

8 de gener de 2014

En una primera part, usarem la transformada discreta de Fourier (DFT) per tal de calcular l'espectre d'un senyal, x. Després usarem aquest senyal per a fer una modulació d'amplada de pols, *Pulse Width Modulaton* (PWM), i calcularem l'espectre del senyal modulat, x_{PWM}. A partir d'aquest espectre decidirem la freqüència de tall f_c d'un filtre de primer ordre per tal que en filtrar x_{PWM} el senyal obtingut y sigui semblant al senyal original x. Finalment implementarem aquest filtratge amb Octave tant en el domini temporal, usant *filter.m*, com en el domini freqüencial, usant *fft.m*, i compararem els temps de càlcul.

1 L'espectre d'un senyal de veu

A continuació calcularem la DFT d'un senyal x obtingut com la suma d'un senyal de veu x_{veu}, mostrejat a $f_s = 48\,\text{kHz}$, i un senyal sinusoïdal x_{sin} de freqüència f_{sin} i amplitud A_{sin}. El resultat de la DFT X el dividirem per el nombre de mostres N. Aquesta normalització ens permet interpretar la DFT com la sèrie de Fourier de la repetició d'x. Com a resultat, en la part de l'espectre propera a $\pm f_{sin}$ trobarem uns coeficients de valor $A_{sin}/2$.

El següent codi en Octave [1]. us pot servir d'ajut.

```
%% Espectre d'un senyal de veu
% senyal de veu
load ALIMENTACIO.mat, % carrega x_veu i fs=48e3
% base de temps mostreig
N=length(x_veu);
t=[0:N-1]'/fs;
% senyal sinusoidal
A_sin=0.05;
f_sin=600;
x_sin=A_sin*cos(2*pi*f_sin*t);
% suma
x=x_veu+x_sin;

% calcul de l'espectre
[X,F]=f_TF(x,fs); %funcio implementada

% representacio de l'espectre
plot(F,abs(X))
% stem(F,abs(X))
```

[1] Caldrà que tingueu instal·lat el paquet octave-signal (que té dependència dels paquets octave-general i octave-control). Podeu usar $ sudo aptitude install octave-signal

La implementació de la funció que fa la DFT, normalitza i desplaça, i calcula el vector de freqüència en el domini analògic és:

```
function [X,F]=f_TF(x,fs)

N=length(x);
% calculo la TF
X1=fft(x);
% escalo
X2=X1/N;
% shift
X=fftshift(X2);

% frequencia corresponent a cada coeficient
F=0*X;
F(1:end)=[-ceil((N-1)/2):floor((N-1)/2)]*fs/N;
```

Observeu el resultat a la *Figura* 1. Es pot comprovar com el senyal apareix representat entre $-24\,\text{kHz}$ i $24\,\text{kHz}$, la meitat de la freqüència de mostreig. Podeu veure un zoom de les freqüències inferiors a $1\,\text{kHz}$ a la *Figura* 2. S'observen a $600\,\text{Hz}$ un coeficient de valor $0.05/2$ corresponent al senyal sinusoïdal x_{sin}. També és interessant observar com l'espectre de veu es concentra en determinades freqüències que estan relacionades harmònicament. En concret observem un pic a $f_1 \simeq 133\,\text{Hz}$ i al seu segon $\simeq 267\,\text{Hz}$ i tercer harmònic $\simeq 400\,\text{Hz}$, i un altre pic a $f_2 \simeq 340\,\text{Hz}$ i al seu segon harmònic $\simeq 680\,\text{Hz}$. El valor d'aquestes freqüències, anomenades formants, està relacionat amb les freqüències de ressonància que generem per a produir els sons, bàsicament les vocals. Els homes tenen els formants a freqüències inferiors al de les dones i els nens. Podeu trobar més informació a *Formant*.

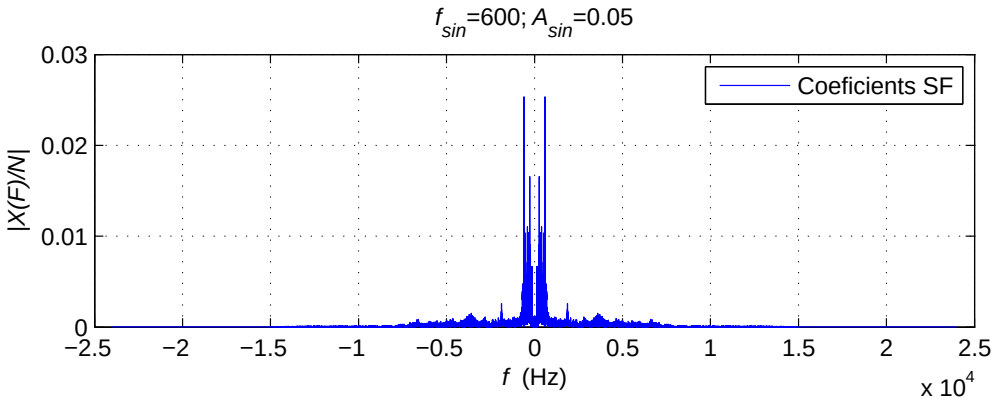

Figura 1: Espectre del senyal x

Tasca 1. Relacioneu la resolució freqüencial de l'espectre de la *Figura* 1, separació entre coeficients, i relacioneu-la amb la durada del senyal.

Tasca 2. Enregistreu el so de diverses vocals i observeu a quines freqüències es troben els formants. Seleccioneu la durada del senyal per a tenir una resolució freqüencial d'1 Hz.

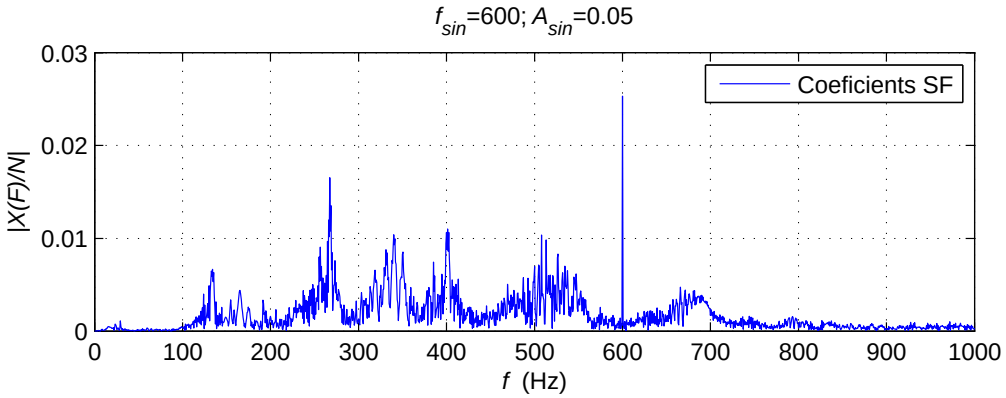

Figura 2: S'observa amplitud $A_{sin}/2$ a f_{sin}

2 Espectre d'un senyal PWM

En sessions anteriors hem usat el mòdul PWM del microcontrolador per implementar un convertidor digital analògic d'un senyal x digital. Aquest senyal modulat x_{PWM} s'ha de filtrar amb un pas-baix per tal de quedar-nos amb el senyal modulador x, ara analògic. Aquest procés de filtratge es veu facilitat si la freqüència del senyal PWM f_{PWM} és molt més gran que l'amplada de banda del senyal x. Si $x = x_{sin}$ de la secció anterior, aquesta amplada de banda és f_{sin}. Si $x = x_{veu}$ de la secció anterior, aquesta amplada de banda, segons la *Figura* 1, es pot considerar de 5 kHz si seguim un criteri conservador, però es podria considerar també una amplada de banda d' 1 kHz si tolerem que la veu pateixi una forta distorsió, sense que això impliqui perdre intel·ligibilitat.

A ATENEA trobareu penjat l'arxiu *.m en Octave que us permetrà calcular l'espectre d'un senyal PWM. És important comentar les següents línies:

```
% senyal modulador
x=0*x_veu+1*x_sin;
```

Aquí podeu triar si usar només el senyal de veu, només el sinus (configuració per defecte) o bé una combinació de tots dos.

```
% senyal PWM
fpwm=1*fs;
k=64; % mostres per periode de PWM
DR=5; % rang dinamic de 0 a DR
```

Amb el primer paràmetre fixem la freqüència del senyal PWM que ha de ser major que la de mostreig del senyal de veu f_s. El codi és molt més ràpid si f_{PWM} és múltiple d'f_s. El paràmetre k indica el nombre de mostres que usarem per a definir un període del senyal PWM i equival a la resolució del senyal PWM. En el nostre cas podem assignar 64 valors diferents a l'amplada del senyal PWM. Per tant tenim un mòdul PWM de 6 bits. Si volem simular el mòdul PWM del microcontrolador de l'Arduino, de 8 bits, caldrà assignar $k = 256$. Un valor de k petit generarà soroll de quantificació. Finalment, el paràmetre DR defineix el valor que pren el senyal PWM quan està actiu (es considera que quan està inactiu val zero) i pren el valor de 5 V, tensió d'alimentació del microcontrolador de l'Arduino.

Tasca 3. Guardeu tots els arxius del zip penjat a ATENEA en un directori local. Executeu *pds_pwm_spectrum2.m* amb els paràmetres per defecte. Concentreu-vos en observar la distri-

bució de l'espectre del senyal x_{PWM} quan $x = x_{sin}$. Modifiqueu f_{sin}, f_{PWM} i k i observeu els canvis.

Tasca 4. Ara concentreu-vos en observar la distribució de l'espectre del senyal x_{PWM} quan $x = x_{veu}$. Modifiqueu f_{PWM} i k i observeu els canvis. Es possible que us calgui comentar el canvi d'eixos $axis$ que apareix darrera els $plots$.

3 Filtratge del senyal PWM

Finalment volem filtrar el senyal x_{PWM}. Aquest filtratge el farem seguint tres estratègies diferents:

Tasca 5. La primera consisteix en **simular** com es comportaria un **filtre analògic**. En aquest cas intentem reproduir l'efecte de filtrar amb un filtre RC de 1r ordre, amb freqüència de tall f_c, com l'utilitzat anteriorment al laboratori. El filtratge es realitza en el domini freqüencial, multiplicant cada component de la DFT del senyal x_{PWM} pel valor del filtre analògic H_a a la freqüència de cada component. A continuació es realitza la DFT inversa i s'obté la sortida y_{DFTa}. Observeu que aquest filtratge requereix una DFT, una multiplicació i una DFT inversa. Observeu el temps de càlcul requerit per a fer aquest filtratge *filtrat_DFTa_Nxk*. Quantes mostres intervenen en cada DFT? Observeu que a l'inici del fitxer fixem el nombre de mostres d'x_{veu} a una potència de 2. Com varia el temps de filtratge en variar el nombre de mostres d'x_{veu}?

Tasca 6. La segona estratègia consisteix en **implementar** un **filtre digital** que reprodueixi el comportament del filtre analògic. El filtratge es realitza en el domini freqüencial, multiplicant cada component de la DFT del senyal x_{PWM} pel valor del filtre digital H_d a la freqüència de cada component. A continuació es realitza la DFT inversa i s'obté la sortida y_{DFT}. Aquest filtratge també requereix una DFT, una multiplicació i una DFT inversa. Observeu el temps de càlcul requerit per a fer aquest filtratge *filtrat_DFT_Nxk*. Quantes mostres intervenen en cada DFT? Observeu que a l'inici del fitxer fixem el nombre de mostres d'x_{veu} a una potència de 2. Com varia el temps de filtratge en variar el nombre de mostres d'x_{veu}?

Tasca 7. La tercera estratègia segona també consisteix en **implementar** un **filtre digital** que reprodueixi el comportament del filtre analògic. Aquest cop, però, el filtratge es realitza en el domini temporal, reproduint allò que faria un sistema a temps real, per exemple una FPGA. Usem la funció $filter.m$ per a obtenir la sortida y_{filter}. Observeu el temps de càlcul requerit per a fer aquest filtratge *filtratge_filter*. Observeu que a l'inici del fitxer fixem el nombre de mostres d'x_{veu} a una potència de 2. Com varia el temps de filtratge en variar el nombre de mostres d'x_{veu}?

Tasca 8. Comenteu les diferències entre les dues primeres estratègies, basades en el domini freqüencial: diferències en el temps de càlcul, diferències entre H_a i H_d, diferències en visualitzar i escoltar els senyals temporals x, y_{DFTa} i y_{DFT}.

Tasca 9. Comenteu les diferències entre les estratègies, basades en el domini freqüencial, i la darrera, basada en el domini temporal: diferències en el temps de càlcul, i diferències en visualitzar i escoltar els senyals temporals x, y_{DFT} i y_{filter}.

Tasca 10. Verifique el soroll de quantificació generat en usar una resolució de PWM baixa, per exemple de 3 bits amb $k = 8$.

Tasca 11. Finalment, trieu els paràmetres adequats per tal que no s'observi cap diferència en escoltar el senyal x i qualsevol de les sortides y. Escolliu k prou gran, una f_{PWM} prou gran, i un filtre amb f_c i n tals que atenuï la part de l'espectre que es troba al voltant de f_{PWM} sense distorsionar la part de l'espectre que es troba a l'origen, corresponent al senyal original x. Alerta amb el temps de simulació!