

Proyecto de Análisis de Señales y Sistemas Lineales SS5T01 – 2do período de 2010

Simulación de sistema DTMF con Matlab/Octave

El proyecto consiste en probar en Matlab/Octave, un código para la generación de tonos DTMF. El concepto de DTMF lo pueden encontrar en http://es.wikipedia.org/wiki/Marcaci%C3%B3n_por_tonos o mejor, en la versión en inglés que es mucho mas completa: http://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling

Existen muchos ejemplo de este código en internet, preferiblemente podrán utilizar el código que ofrece MatWorks: <http://www.mathworks.com/products/demos/signaltlbx/dtmf/dtmfdemo.html>

Otros links con ejemplos de DTMF en Matlab/Octave:

<http://www.mathworks.com/matlabcentral/fileexchange/724>

<http://vierito.es/wordpress/2009/01/15/llamando-por-telefono-con-matlab/>

http://www.engr.mun.ca/~sircar/project1_files/dtmf.pdf

<http://mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf> << Un tutorial

<http://dev.emcelettronica.com/signal-generation-matlab-example-dtmf-telephony>

<http://www.shawnlankton.com/2003/08/dtmf-generator/>

<http://www.mathworks.com/matlabcentral/fileexchange/11333> << excelente

http://members.tripod.com/dsp_peru/

El trabajo deberá contener:

1. Breve concepto de DTMF, frecuencias, historia y vigencia
2. Código para la generación de los tonos y reproducción en audio. Investigue en los links que arriba se indican, y utilice el código que más le agrade. Para facilitar las cosas, puede guiarse con el código recomendado mas adelante o visite el sitio <http://www.mathworks.com/products/demos/signaltlbx/dtmf/dtmfdemo.html> pero tenga en cuenta que consideraré importante los aportes que usted desarrolle, tales como código alternativo (que puede tomar de la lista de links de arriba), gráficas adicionales que usted decida ofrecer, etc.
3. Gráficas del espectro de cada uno de los números de su teléfono celular en una sola pantalla (una sola hoja). Estas gráficas deben estar respaldadas por el código que las generó, así que será cotejado el código entregado y las gráficas.

A continuación, el código recomendado, traducido y adaptado del original de MatWorks, con comentarios adicionales para explicar la función de los comandos:

```
symbol = {'1','2','3','4','5','6','7','8','9','*','0','#'};
lfg = [697 770 852 941]; % Grupo de Bajas Frecuencias
hfg = [1209 1336 1477]; % Grupo de Altas Frecuencias

% El siguiente codigo genera una matriz de 2x12. Cada fila corresponde a uno
% de los digitos (símbolos) y consta de dos frecuencias (las dos columnas)

f = [];
for c=1:4,
    for r=1:3,
        f = [ f [lfg(c);hfg(r)] ];
    end
end
```

```

f'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Fs = 8000;          % Frecuencia de muestreo de 8 kHz
N = 800;           % Cada todo tendrá una duración de 800ms
t = (0:N-1)/Fs;   % Momentos de las muestras, 800 muestras en Fs
pit = 2*pi*t;

tones = zeros(N,size(f,2));
for toneChoice=1:12,
    % Generar tono
    tones(:,toneChoice) = sum(sin(f(:,toneChoice)*pit));
    % Dibujar la onda de cada tono
    subplot(4,3,toneChoice),plot(t*1e3,tones(:,toneChoice));
                                title(['Simbolo           ',          symbol{toneChoice},':
[' ,num2str(f(1,toneChoice)),',',num2str(f(2,toneChoice)),']'])
    set(gca, 'Xlim', [0 25]);
    ylabel('Amplitud');
    if toneChoice>9, xlabel('Tiempo (ms)'); end
end
set(gcf, 'Color', [1 1 1], 'Position', [1 1 1280 1024])
annotation(gcf,'textbox', 'Position',[0.38 0.96 0.45 0.026],'EdgeColor',[1 1 1],'String', '\bf Cada
tono del teclado en funcion del tiempo');

Nt = 205;
original_f = [lfg(:);hfg(:)] % Frecuencias Originales

% Indices de la DFT. Debido a la discretizacion en el dominio de la frecuencia, es necesario llevar las
frecuencias de la DTMF al punto frecuencial mas cercano.
k = round(original_f/Fs*Nt);
estim_f = round(k*Fs/Nt)      % Frecuencias en las que la DFT sera estimada

tones = tones(1:205,:);

figure,
for toneChoice=1:12,
    % Selecciona la onda de un tono en especifico
    tone=tones(:,toneChoice);
    % Estimar la DFT usando Goertzel (uno de tantos metodos para obtener la DFT)
    ydft(:,toneChoice) = goertzel(tone,k+1); % No se preocupe por estos parametros
    % Dibujar magnitud de la DFT
    subplot(4,3,toneChoice),stem(estim_f,abs(ydft(:,toneChoice)));
                                title(['Simbolo           ',          symbol{toneChoice},':
[' ,num2str(f(1,toneChoice)),',',num2str(f(2,toneChoice)),']'])
    set(gca, 'XTick', estim_f, 'XTickLabel', estim_f, 'Xlim', [650 1550]);
    ylabel('Magnitud del espectro obtenido con la DFT');
    if toneChoice>9, xlabel('Frecuencia (Hz)'); end
end
set(gcf, 'Color', [1 1 1], 'Position', [1 1 1280 1024])
annotation(gcf,'textbox', 'Position',[0.28 0.96 0.45 0.026],...
    'EdgeColor',[1 1 1],...
    'String', '\bf Espectro de frecuencias positivas, estimado utilizando la funcion Goertzel, una
alternativa de FFT');

```

NOTAS IMPORTANTES SOBRE ESTE CÓDIGO:

El núcleo de este proyecto y del código está en la función `goertzel()`, que es una forma de obtener la Transformada de Fourier de una señal discreta. Tal y como se ha explicado varias veces, en un computador, cuando se realiza procesamiento de señales, no se trabaja con la Transformada de Fourier en tiempo continuo, sino en tiempo discreto. Por la teoría que ya hemos completado, sabemos que la transformada de Fourier de una señal periódica en el dominio del tiempo es una señal continua en el dominio de la frecuencia, y esta a su vez no puede ser tratada como tal en el computador, necesitamos discretizarla también. De aquí que la función `goertzel()` (y otras funciones para generar la transformada de una señal discreta, como la importantísima `fft()`) generan una señal discreta, que se conoce como Transformada Discreta de Fourier, materia de estudio del próximo curso de DSP.