ANEXO B: Código para el cálculo del factor de calidad de atenuación Qc.

Programa principal

```
%Universidad Nacional de Colombia
%Programa_Calculo del parámetro Qc
%Ordoñez-Potes, Sanchéz, J.2013-----------------------------------------------
% Llama los archivos
A=dir(fullfile('D:\Documents\TESIS\Registro sísmicos\ASCII FORMAT\waveforms\Programa\*.out'));
list={A.name}'; % Enlista los archivos con su respectivo nombre
for k=1:1
[enca,val] = hdrload(A(k,1).name);% abre los archivos y los organiza
for i=1: length (val)
R(i,k)=val(i); % carga matriz con los registros para cada componente
end
disp(A(k,1).name)
end
s=R; %matriz final con las tres componentes
format short
k=1;

% Tiempo de muestreo
Fs=100; % Frecuencia
n=length(R(:,k));%Leer el numero de datos
T=(0:n-1)/Fs; %Tiempo de muestreo
t=T'; % Transpuesta del vector tiempo
clear k

% tiempo de la onda Coda
for ii=1:1
tp(ii)=input('Ingrese el valor del tiempo de arribo P');
ttp=fix(Fs*tp(ii));
ts(ii)=input('Ingrese el valor del tiempo de arribo S');
tts=fix(Fs*ts(ii));
Co(ii)=2*ts(ii); %Tiempo de la onda coda
ttc=fix(Fs*Co(ii));

% Grafica de la señal con los tiempos de arribo de las ondas P y S y Coda
%Marcadores
for i=1:2000
marp(i,:)=i; % Marca el tiempo de arribo de la onda P
mars(i,:)=2*i; % Marca el tiempo de arribo de la onda S
marc(i,:)=i; % Marca el inicio del tiempo de análisis de la onda Coda
end

jj=1;

%Calculo de Qc para cada componente
%%Utiliza un filtro de Butterworth de cuarto orden para filtrar la señal

fc=[1.5,3,6,10,15]; %Frecuencia central
banda = [1,2,2,4,4,8,8,12,12,18]; %Ancho de banda
for i=1:length(T); % ventana de tiempo de 12 s, Fs=100
    if (T(i))==Co(ii);
        for l=fix(Co(ii)*Fs):fix((Co(ii)*Fs+12*Fs))
          mtff(l)= T(l);
          Frr(l)=s(l,jj);
```

```
        end
    end
end


% análizada
for j=1:1
    mtfv(:,j)=mtff(Co(ii)*Fs:(Co(ii)*Fs+12*Fs)); % Vector tiempo de la Onda Coda sin ceros
    Frv(:,j)=Frr(Co(ii)*Fs:(Co(ii)*Fs+12*Fs)); % Señal de la Onda Coda sin ceros
    mtf=mtfv(:,j);% Matriz de los tiempos de señales en la ventana de tiempo
    Fr=Frv(:,j);% Matriz de las señales en la ventana de tiempo
end
clear j
clear k

% Filtrado de las señales
for i=1:length(fc)
    nfr=i;
    signal=singalfil(Fr,banda,Fs,nfr); % Filtro de la señal
    [y,T]=envelope(signal,Fs); %Envolvente de la señal, Transformada de Hilbert
    hy=log(y); %algortimo natural de la envolvente
    ss(:,i)=signal;
    yy(:,i)=y;

% Calculo de ventanas de tiempo cada 1s de Qc
  for kk=1:12
      ttfw{i,kk}=mtf(1:(kk*Fs));
      hhyw{i,kk}=hy(1:(kk*Fs));
      pw{i,kk}=polyfit(ttfw{i,kk},hhyw{i,kk},1); %???
      poliw{i,kk}=regrepol(ttfw{i,kk},hhyw{i,kk},pw{i,kk});
      desv(i,kk)=std(poliw{i,kk});
      nl(i,kk)=length (hhyw{i,kk}); % longitud del vector seleccionado por la ventana
      rsw(i,kk)=qu_error(ttfw{i,kk},hhyw{i,kk},nl(i,kk)); %Error cuadrado
      pendiw{i,kk}=pw{i,kk};  %Valor de la pendiente
      pendiente=pw{i,kk};
      ppendi(i,kk)=pendiente(1,1);
  end
end
 % Calculo de Qc
 for i=1:length(fc)
    ad(i)= (-fc(i)*(ts(ii)-tp(ii)))/2;
    qc(i,:)=ad(i)./ppendi(i,:);
    qcdesvf(i,:)=-ad(i)./desv(i,:);
 end

  %Escogencia qc
for i=1:length(fc)
     for n=2:12
    if rsw(i,n-1)<rsw(i,n)&& ppendi(i,n)<0
       Qc(ii,i)=qc(i,n);
       qcdesv(ii,i)=qcdesvf(i,n);
    else
    end
      end
end
```

```
jj=jj+1;
end

disp(Qc);
disp (qcdesv);

%Gráfica la señal
figure;
plot(t(1:Co(ii)*Fs+12*Fs),R(1:Co(ii)*Fs+12*Fs,1),t(ttp),marp,'r',t(tts),marp,'r',t(ttc),marp,'g'), grid
%Grafica la señal
title(['FIGURA 1: Señal original y tiempos de arribo de las ondas P, S y Coda
Tp=',num2str(tp),'Ts=',num2str(ts),'Tc=',num2str(Co(ii))])
xlabel('Tiempo (s)')
ylabel('Amplitud')

figure;
plot(mtf,Fr);
title('FIGURA 2:Ventana de tiempo 12seg')
xlabel('Tiempo (s)')
ylabel('Amplitud')

for i=1:length(fc)
figure;
plot(mtf,ss(:,i),mtf,yy(:,i),'r'), grid %Grafica la señal
title(['FIGURA', num2str(i) ,': Señal con Filtro de', num2str(fc(i)),' Hz', 'y, Envol. suavizada de
Hilbert'])
xlabel('Tiempo (t)')
ylabel('Amplitud')
end
figure;
plot (fc,Qc, 'bx')
```

Funciones

```
% Coeficientes para filtrar
function [signal]=singalfil(Fr,banda,Fs,nfr)
 i=nfr;
 f1(i)= round(banda(i))/(0.5*Fs);
 f2(i)= round(banda(2*i))/(0.5*Fs);
 f3(i)=f1(i)-f1(i)*0.6;
 f4(i)=f2(i)+f2(i)*0.6;
Rp=3;
Rs=30;
Wpp = [f1 f2]; %Pasabanda
Wss= [f3 f4]; %stopbanda
[b,a]=filtrom(f1(i),f2(i),f3(i),f4(i),Rp,Rs);
signal= filter(b,a,Fr);
end

function [b,a]=filtrom(f1,f2,f3,f4,Rp,Rs)
%Calcula los coeficentes del filtro butterworth
%f1: Frecuencia limite inferior del filtro pasabanda
%f2: Frecuencia limite superior del filtro pasabanda
```

```
%f3: Frecuencia limite inferior de las bandas eliminadas
%f4: Frecuencia limite superior de las bandas eliminadas
%Rp: Atneuacion en decibelio en pasabanda
%Rs: Atenuacion en decibelios en el stopbada

Wpp = [f1 f2]; %Pasabanda
Wss = [f3 f4]; %stopbanda
[Nn,Wn] = buttord(Wpp,Wss,Rp,Rs);
[b,a] = butter(Nn,[f1 f2]);


function [y,T]=envelope(signal,Fs)
%http://www.mathworks.com/matlabcentral/fileexchange/37545-envelope-detctio
%n-for-signal-analysis/content/envelope.m
clc;
close all;

%Dominio de frecuencia
Nf=length(signal); %Longitud del vector resultante
x=abs(hilbert(signal));
X=fft(x,Nf); % Transformada de Fourier
s=sqrt( X.*conj(X))/ Nf; % Valor real de la transfor. de Fourier
T=Fs*(1:Nf/2)/ Nf; %Tiempo de muestreo 2

%Envelope Detection based on Hilbert Transform and then FFT
analy=hilbert(signal);
y=abs(analy);
N=2*2048;T=Nf/Fs;
sig_f=abs(fft(y(1:Nf)',Nf));
sig_n=sig_f/(norm(sig_f));
freq_s=(0:Nf-1)/T;


%Calculo de logaritmo natural para cada segundo.
function[hy,ttf]=alnatural(y,mtf,Fs)
for i=1:12
   for m=Co*Fs:(Co*Fs+i*Fs)
   hy(m)=log(y(m));
   ttf(m)=mtf(m*Fs);
end
end
```