```
clear; clc
format short

rutaarch = 'C:\Users\mbaen\POSDOC\MBR\RuidoGEOFmat\';
rutagrab = 'C:\Users\mbaen\POSDOC\MBR\HVRuidoGEOF\';

listest = dir(fullfile(rutaarch));
listest = {listest.name}';
bal = find(ismember(listest,[{'.'};{'..'}])==1);
listest(bal) = [];

%% DATOS INICIALES
senhal = 'noise';
unidad = 'velo';  % DESP VELO ACEL

% Filtro inicial de las señales
w1 = 0; %1;     0=s/filtro
w2 = 0; %255.9; 0=s/filtro

% Factor de esquina para taper de señales
factap = 0.01;

% *****NORMALIZACIÓN*****
% band:   0=ninguna, 2=suma3direcc, 3=SW
% onebit: 1=SI, 0=NO

% SELECCIONAR DATOS
segvent = [500];          % Segundos de las ventanas para inversión
porctrasl = [25];         % Porcentaje de traslape de las ventanas
normalizac = [2 0];       % Normalización: [band,onebit]
tiempoHV = [48*60];       % Tiempo (minutos) para cálculo de cada H/V (Tiempo de registro manipulable
)
ventaleatHV = 0;          % 1=ventanas aleatoria, 0=ventanas continuas
NvBootstrap = 1;          % Número de ventanas para el boostrap
tSTA = 1; %1.35;          % En segundos
tLTA = 60;                % En segundos
Smax = 5;                 % 0=todas las ventanas
Smin = 0.2;
dfnew = 1;

% Si baja el tLTA es más conservador

itertot = length(segvent)*length(porctrasl)*length(normalizac(:,1))*length(tiempoHV);

%% Buscar estación
buscar = listest;
buscar = {'TOME'};        % ¡¡¡ESCOGER ESTACIÓN!!!

%%
figure(100)
p = load('progresivo');
p = p.mycmap;
p = colormap(jet);
close(100)

Ncomb = itertot;
porc = length(p(:,1))/Ncomb;
for i = 1:Ncomb
    col(i,:) = p(ceil(porc*i),:);
end
col = flipud(col);

% tetarot = 0:45:180;
tetarot = 0;
[~,Nbuscar] = ismember(buscar,listest);
for aleat = 1 %:10

    % Ciclo para estaciones
    for ee = 1:length(buscar)
        estac = listest{Nbuscar(ee)};
        fprintf(1,'%d%s%d%s%s\n',ee,'/',length(buscar),' --> ',estac);

        if ~exist(rutagrab,'dir'); mkdir(rutagrab); end
        if ~exist([rutagrab,estac],'dir'); mkdir([rutagrab,estac]); end
        nombgrab = [rutagrab,estac,'\HV_',estac];
        % if exist(nombgrab,'file') ~= 0; continue; end
```

```
        listreg = dir(fullfile(rutaarch,estac,unidad,'*.mat'));
        listreg = {listreg.name}'; %name
        listdias00 = {};
        for i = 1:length(listreg)
            listdias00{i,1} = listreg{i}(1:8);
        end
        listdias = unique(listdias00);

        buscardia = listdias;
        % buscardia = {'20200929';'20200116';'20201129'};
        [~,Nbuscardia] = ismember(buscardia,listdias);
        leyenda = [];
        for dd = 1:length(Nbuscardia)
            k = Nbuscardia(dd);

            nombgrab0 = [nombgrab,'_',listdias{k},'.mat'];
            % if exist(nombgrab0,'file') ~= 0; continue; end

            cont0 = 0;
            ESTR = [];
            vecfechahms = [];
            for i = 1:length(listreg)
                if strcmp(listreg{i}(1:8),listdias{k})
                    cont0 = cont0+1;
                    REG = load([rutaarch,estac,'\',unidad,'\',listreg{i}]);
                    ESTR.EW(:,cont0) = double(REG.EW);
                    ESTR.NS(:,cont0) = double(REG.NS);
                    ESTR.VE(:,cont0) = double(REG.VE);
                    ESTR.dt(cont0,1) = REG.dt;
                    vecfechahms = [vecfechahms;{[listreg{i}(1:8),'_',listreg{i}(9:end-4)]}];
                end
            end
            dt = ESTR.dt(1);
            fmax = 1/(2*dt);
            [Nn,Nhoras] = size(ESTR.EW);
            f0 = [];

            HV = struct('estac',[],'paraadic',[],'clavecomb',[],'Nvent',[],'fcomb',[],'HVmean_comb',[],'NVmean_comb',[],'EVmean_comb',[], …
                'tiempoHV_orig_min',[],'tiempoHV_real_min',[], …
                'f_comb1',[],'HVtot_comb1',[],'HVNSdir_comb1',[],'HVEWdir_comb1',[],'tetarot',[]);

            HV.HVNSdir_comb1 = [];
            HV.HVEWdir_comb1 = [];
            for Nteta = 1:length(tetarot)
                iter = 0;
                ccd = 0;
                Nv = 0;

                % Rotación sismogramas
                teta = tetarot(Nteta);
                costeta = cosd(teta);
                sinteta = sind(teta);
                EW = ESTR.EW*costeta+ESTR.NS*sinteta;  %longitudinal
                NS =-ESTR.EW*sinteta+ESTR.NS*costeta;  %transversal
                VE = ESTR.VE;

                fprintf(1,'\t%d%s%d%s%d\n',Nteta,'/',length(tetarot),' --> teta=',teta);

                %% ***************************************************
                % CICLO LONGITUD DE VENTANAS
                % ***************************************************
                for vv = 1:length(segvent)
                    ptosvent = round(segvent(vv)/dt);
                    if rem(ptosvent,2) ~= 0; ptosvent = ptosvent-1; end
                    Nespec = 1*ptosvent;
                    NQ = Nespec/2+1;
                    df = 1/(Nespec*dt);

                    if df > dfnew
                        NQ = fmax/dfnew+1;
                        Nespec = (NQ-1)*2;
                    end

                    frec = linspace(0,fmax,NQ).';
                    frec = round(frec*1000000)/1000000;
                    flim1 = frec(1); %0.01
```

```
                    flim2 = fmax; %20;
                    ini = find(frec>=flim1,1,'first');
                    fin = find(frec>=flim2,1,'first');
                    f = frec(ini:fin);
                    Nfrecred = fin-ini+1;

                    %% ***************************************************
                    % CICLO TRASLAPE DE VENTANAS
                    % ***************************************************
                    wincleantot = [];
                    for tt = 1:length(porctrasl)

                        % VENTANEO
                        Ntras = floor(porctrasl(tt)/100*ptosvent);
                        iv = (1:ptosvent-Ntras:Nn).';
                        fv = iv+ptosvent-1;
                        cambmax = [];
                        cambmax = find(fv>Nn);
                        if ~isempty(cambmax); fv(cambmax) = Nn; end
                        tvent = [iv,fv,fv-iv+ones(length(iv),1)];
                        elim = find(and(fv==Nn,tvent(:,3)<ptosvent));
                        % if length(elim) >= 2; tvent(elim(2:end),:) = []; iv(elim(2:end)) = []; fv(
elim(2:end)) = []; end
                        tvent(elim,:) = []; iv(elim) = []; fv(elim) = [];
                        M = length(iv);

                        % ELIMINA LAS VENTANAS MÁS ENERGÉTICAS DE LA SEÑAL EN SEGUNDOS
                        if Smax == 0
                            wincleantot = ones(M,1);
                        else
                            % wincleantot = zeros(M,1);
                            [wincleanEW,STALTAEW] = picossig6(EW,dt,iv,fv,tSTA,tLTA,Smax,Smin);
                            [wincleanNS,STALTANS] = picossig6(NS,dt,iv,fv,tSTA,tLTA,Smax,Smin);
                            [wincleanVE,STALTAVE] = picossig6(VE,dt,iv,fv,tSTA,tLTA,Smax,Smin);
                            wincleantot = wincleanEW.*wincleanNS.*wincleanVE;
                        end
                        Nvent = sum(sum(wincleantot));

                        %% Figuras para revisión 1
                        % figure(300)
                        % set(gcf,'Position',get(0,'Screensize'))
                        % fig = tiledlayout(3,2,'TileSpacing','tight','Padding','tight');
                        %
                        % t = (0:dt:(length(NS)-1)*dt).';
                        %
                        % nexttile(1)
                        % plot(t,STALTANS,'k'); hold on; grid on
                        % line([t(1) t(end)],[Smax Smax],'color','r','linestyle','--','linewidth',2)
                        % set(gca,'YTick',0:1:3)
                        % set(gca,'XTickLabel',[])
                        % ylabel('NS','fontname','Times New Roman','fontSize',14)
                        % ylim([0 11])
                        % set(gca,'fontname','Times New Roman','fontSize',14)
                        %
                        % nexttile(3)
                        % plot(t,STALTAEW,'k'); hold on; grid on
                        % line([t(1) t(end)],[Smax Smax],'color','r','linestyle','--','linewidth',2)
                        % set(gca,'YTick',0:1:3)
                        % set(gca,'XTickLabel',[])
                        % ylabel('EW','fontname','Times New Roman','fontSize',14)
                        % ylim([0 11])
                        % set(gca,'fontname','Times New Roman','fontSize',14)
                        %
                        % nexttile(5)
                        % plot(t,STALTAVE,'k'); hold on; grid on
                        % line([t(1) t(end)],[Smax Smax],'color','r','linestyle','--','linewidth',2)
                        % set(gca,'YTick',0:1:3)
                        % ylabel('VE','fontname','Times New Roman','fontSize',14)
                        % xlabel('Time (s)','fontname','Times New Roman','fontSize',14)
                        % ylim([0 11])
                        % set(gca,'fontname','Times New Roman','fontSize',14)
                        % h1 = plot(0,0,'k');
                        % h2 = plot(0,0,'--r','linewidth',2);
                        % % lg = legend([h1 h2],'STA/LTA',['(STA/LTA)max = ',num2str(Smax)]);
                        % % set(lg,'location','south outside','fontname','Times New Roman','fontSize
',14)
```

```
                        %% Figuras para revisión 2
                        % figure(201)
                        % subplot(2,1,1)
                        % plot(t,STALTAVE,'k'); hold on %; grid on
                        % line([t(1) t(end)],[Smax Smax],'color','r','linestyle','--','linewidth',2)
                        % line([t(1) t(end)],[Smin Smin],'color','r','linestyle',':','linewidth',2)
                        % set(gca,'YTick',0:1:3)
                        % ylabel('STA/LTA ratio','fontname','Times New Roman','fontSize',14)
                        % xlim([0 t(end)])
                        % ylim([0 3.5])
                        % set(gca,'XTickLabel',[])
                        % set(gca,'fontname','Times New Roman','fontSize',14)
                        % h1 = plot(0,0,'k');
                        % h2 = plot(0,0,'--r','linewidth',2);
                        % h3 = plot(0,0,':r','linewidth',2);
                        % lg = legend([h1 h2 h3],'STA/LTA',['(STA/LTA)max = ',num2str(Smax)],['(STA/
LTA)min = ',num2str(Smin)]);
                        % set(lg,'location','northwest','fontname','Times New Roman','fontSize',14)

                        % DIVISIÓN DE LA SEÑAL EN VENTANAS DE TIEMPO
                        Ndias = length(vecfechahms);
                        Nvini = Ndias*M; %Nvini = M;
                        EWv = (zeros(ptosvent,Nvini)); %single
                        NSv = (zeros(ptosvent,Nvini)); %single
                        VEv = (zeros(ptosvent,Nvini)); %single
                        fechahmsvent = cell(Nvini,1);
                        cont = 0;
                        for j = 1:Ndias
                            for kk = 1:M
                                cont = cont+1;
                                EWv(:,cont) = EW(iv(kk):fv(kk),j);
                                NSv(:,cont) = NS(iv(kk):fv(kk),j);
                                VEv(:,cont) = VE(iv(kk):fv(kk),j);
                                fechahmsvent(cont,1) = {[vecfechahms{j},'_',num2str(kk)]};
                            end
                        end
                        wincleantotlinea = reshape(wincleantot,[1,Nvini]);
                        ventok = find(wincleantotlinea~=0);

                        % VENTANAS EFECTIVAS
                        EWv = EWv(:,ventok);
                        NSv = NSv(:,ventok);
                        VEv = VEv(:,ventok);
                        fechahmsvent = fechahmsvent(ventok);

                        ventNOefectiv = unique([find(sum(abs(EWv))==0) find(sum(abs(NSv))==0) find(s
um(abs(VEv))==0)]);
                        ventefectiv = 1:Nvent;
                        ventefectiv(ventNOefectiv) = [];
                        Nv = length(ventefectiv);
                        EWv = EWv(:,ventefectiv);
                        NSv = NSv(:,ventefectiv);
                        VEv = VEv(:,ventefectiv);
                        fechahmsvent = fechahmsvent(ventefectiv);

                        if Nv == 0; continue; end

                        % REMUEVE LA MEDIA POR VENTANAS Y APLICA TAPER
                        EWv = EWv-mean(EWv);
                        NSv = NSv-mean(NSv);
                        VEv = VEv-mean(VEv);
                        tap = repmat(tukeywin(ptosvent,factap),1,length(NSv(1,:)));
                        EWv = EWv.*tap;
                        NSv = NSv.*tap;
                        VEv = VEv.*tap;

                        %% Figuras para revisión 1
                        % figure(300)
                        % set(gcf,'Position',get(0,'Screensize'))
                        % % fig = tiledlayout(3,1,'TileSpacing','tight','Padding','tight');
                        % % title(fig,estac,'fontname','Times New Roman','fontSize',14,'fontweight',
'bold','interpreter','none');
                        %
                        % t = (0:dt:(length(NS)-1)*dt).';
                        %
                        % d = find(wincleantot~=0);
                        % NSm = NS;
```

```
% EWm = EW;
% VEm = VE;
% ml = max([max(abs(NSm)) max(abs(EWm)) max(abs(VEm))]);
% % ml = 1;
%
% nexttile(2)
% plot(t,NSm/ml,'k'); hold on; grid on
% ylabel('NS','fontname','Times New Roman','fontSize',14)
% xlim([t(1) t(end)])
% ylim([-1 1])
% % set(gca,'YTick',[-0.02,0,0.02])
% set(gca,'XTickLabel',[])
% set(gca,'fontname','Times New Roman','fontSize',14)
%
% nexttile(4)
% plot(t,EWm/ml,'k'); hold on; grid on
% ylabel('EW','fontname','Times New Roman','fontSize',14)
% xlim([t(1) t(end)])
% ylim([-1 1])
% % set(gca,'YTick',[-0.02,0,0.02])
% set(gca,'XTickLabel',[])
% set(gca,'fontname','Times New Roman','fontSize',14)
%
% nexttile(6)
% plot(t,VEm/ml,'k'); hold on; grid on
% ylabel('VE','fontname','Times New Roman','fontSize',14)
% xlabel('Time (s)','fontname','Times New Roman','fontSize',14)
% xlim([t(1) t(end)])
% ylim([-1 1])
% % set(gca,'YTick',[-0.02,0,0.02])
% set(gca,'fontname','Times New Roman','fontSize',14)
%
% for q = 1:length(wincleantot)
%
%     if ismember(q,d)
%         NSd = NSm(iv(q):fv(q));
%         EWd = EWm(iv(q):fv(q));
%         VEd = VEm(iv(q):fv(q));
%
%         nexttile(2)
%         fill([t(iv(q)),t(fv(q)),t(fv(q)),t(iv(q)),t(iv(q))], ...
%             [-1,-1,1,1,-1],'b','edgecolor','b','facealpha',0.5) ; hold on
%
%         nexttile(4)
%         fill([t(iv(q)),t(fv(q)),t(fv(q)),t(iv(q)),t(iv(q))], ...
%             [-1,-1,1,1,-1],'b','edgecolor','b','facealpha',0.5) ; hold on
%
%         nexttile(6)
%         fill([t(iv(q)),t(fv(q)),t(fv(q)),t(iv(q)),t(iv(q))], ...
%             [-1,-1,1,1,-1],'b','edgecolor','b','facealpha',0.5) ; hold on
%         % plot(t(iv(q):fv(q)),NSd/ml,'c',t(iv(q):fv(q)), ...
%         %     EWd/ml+1,'c',t(iv(q):fv(q)),VEd/ml+2,'c'); hold on; grid on
%         % text(t(iv(q)),0.03,num2str(q),'fontname','Times New Roman','font
Size',14)
%         set(gcf,'color','white')
%         % h1 = plot(0,0,'k');
%         % h2 = fill([0,0,0,0,0],[0,0,0,0,0],'b','edgecolor','r','facealpha
',0.1);
%         % lg = legend([h1 h2],'Corrected signals','Windows selected');
%         % set(lg,'location','south outside','fontname','Times New Roman','
fontSize',14)
%     end
% end
% close(300)

%% Figuras para revisión 2
% figure(201)
% t = (0:dt:(length(NS)-1)*dt).';
% d = find(wincleanVE~=0);
% ml = max([max(abs(NS)) max(abs(EW)) max(abs(VE))]);
%
% subplot(2,1,2)
% plot(t,VEm,'k'); hold on %; grid on
% ylabel('Velocity (cm/s)','fontname','Times New Roman','fontSize',14)
% xlabel('Time (s)','fontname','Times New Roman','fontSize',14)
% ylim([-max(abs(VEm)) max(abs(VEm))])
% set(gca,'fontname','Times New Roman','fontSize',14)
```

```
                        % for q = d.'
                        %     if q == d(1)
                        %         yampl = 5*max([max(abs(NSm(iv(q):fv(q)))) ...
                        %             max(abs(EWm(iv(q):fv(q)))) max(abs(VEm(iv(q):fv(q))))]);
                        %     end
                        %     fill([t(iv(q)),t(fv(q)),t(fv(q)),t(iv(q)),t(iv(q))], ...
                        %         [-yampl,-yampl,yampl,yampl,-yampl],'b','edgecolor','r','facealpha'
,0.3) ; hold on
                        % end
                        % xlim([0 t(end)])
                        % ylim([-yampl yampl])
                        % set(gcf,'color','white')
                        % h1 = plot(0,0,'k');
                        % h2 = fill([0,0,0,0,0],[0,0,0,0,0],'b','edgecolor','r','facealpha',0.1);
                        % lg = legend([h1 h2],'Corrected signals','Windows selected');
                        % set(lg,'location','northwest','fontname','Times New Roman','fontSize',14)
                        % % cla

                        %% ************************************************
                        % CICLO DE NORMALIZACIÓN
                        % ************************************************
                        for norm = 1:length(normalizac(:,1))
                            band = normalizac(norm,1);
                            onebit = normalizac(norm,2);

                            % Carpeta y archivo para grabar resultados
                            nombcomb = [senhal,'-',unidad,'-','filt',num2str(w1),'to',num2str(w2), ...
                                'Hz-band',num2str(band),'-onebit',num2str(onebit),'-tv', ...
                                num2str(round(segvent(vv)*10000)/10000), ...
                                's-Overlap',num2str(porctrasl(tt)),'%'];
                            nombcomb = strrep(nombcomb,' ','');
                            nombcomb = strrep(nombcomb,'-onebit0','');
                            nombcomb = strrep(nombcomb,'onebit1','1bit');
                            nombcomb = strrep(nombcomb,'band0','raw');
                            nombcomb = strrep(nombcomb,'band1','Eunit');
                            nombcomb = strrep(nombcomb,'band2','SW3dir');
                            nombcomb = strrep(nombcomb,'band3','SW');
                            % % ---------
                            % carpeta = [rutagrab,estac];
                            % % if ~exist(carpeta,'dir'); mkdir(carpeta); end
                            % nombarch = [carpeta,'\',estac,'-',nombcomb,'.mat'];
                            % nombarch = strrep(nombarch,'-','_');
                            % nombarch = strrep(nombarch,'%','trasl');
                            % % if exist(nombarch,'file') ~= 0; continue; end
                            % % ----------------------------------------

                            % NORMALIZACIÓN
                            [fNSventnorm,fVEventnorm,fEWventnorm,~,~,~,~,~] = ...
                                F_normalizacionfrec(NSv,VEv,EWv, ...
                                Nespec,band,onebit,dt,factap);

                            % promNS_EW = abs((fNSventnorm+fEWventnorm)/2);
                            fNSvent = abs(fNSventnorm(ini:fin,:));
                            fEWvent = abs(fEWventnorm(ini:fin,:));
                            fVEvent = abs(fVEventnorm(ini:fin,:));
                            fHHvent = abs(sqrt((fNSventnorm(ini:fin,:).^2+fEWventnorm(ini:fin,:).^2)
/2));

                            fNSventnorm = [];
                            fEWventnorm = [];
                            fVEventnorm = [];

                            %% ************************************************
                            % CICLO DE TIEMPOS PARA CÁLCULO DE H/V
                            % ************************************************
                            for nh = 1:length(tiempoHV)
                                iter = iter+1;
                                fprintf(1,'\t%s%s%s%s%d%s%d\n',estac,'_',listdias{k},' --> iter ',iter,'/
',itertot);
                                suav = 0;   %0=no; 1=sí

                                [HVtot,NVmean,EVmean,NventHV,vini, ...
                                    tiempoHVnuevo,numHV,HVvent] = ...
                                    F_HVruido(f,fNSvent,fEWvent,fVEvent, ...
                                    fHHvent,segvent(vv),porctrasl(tt), ...
                                    tiempoHV(nh),suav,ventaleatHV,NvBootstrap);
```

```
                                tiempoHVnuevo_str = num2str(round(tiempoHVnuevo*100)/100);
                                clavecomb = ['CD-HV',tiempoHVnuevo_str,'hr','-',nombcomb,'-Nw',num2
str(NventHV(1)),'-NwBS',num2str(numHV)];

                                if teta == 0
                                    ccd = ccd+1;
                                    HV.clavecomb{ccd} = clavecomb;
                                    HV.Nvent{ccd} = NventHV(1);
                                    HV.fcomb{ccd} = f;
                                    HV.HVmean_comb{ccd} = HVtot;
                                    HV.NVmean_comb{ccd} = NVmean;
                                    HV.EVmean_comb{ccd} = EVmean;
                                    HV.tiempoHV_orig_min{ccd} = tiempoHV(nh);
                                    HV.tiempoHV_real_min{ccd} = tiempoHVnuevo*60;
                                end

                                if teta == 0 && iter == 1
                                    HV.estac = estac;
                                    HV.paraadic.diajuliano = listdias{k};
                                    HV.paraadic.fechahms = vecfechahms;
                                    HV.paraadic.ventaleatHV = ventaleatHV;
                                    HV.paraadic.NvBootstrap = NvBootstrap;
                                    HV.paraadic.tSTA = tSTA;
                                    HV.paraadic.tLTA = tLTA;
                                    HV.paraadic.Smax = Smax;
                                    HV.paraadic.Smin = Smin;
                                    if df > dfnew
                                        HV.paraadic.df = dfnew;
                                    else
                                        HV.paraadic.df = df;
                                    end
                                    HV.f_comb1 = HV.fcomb{1};
                                    HV.HVtot_comb1 = HV.HVmean_comb{1};
                                    HV.tetarot = tetarot;
                                end

                                if iter == 1
                                    HV.HVNSdir_comb1 = [HV.HVNSdir_comb1;NVmean.'];
                                    HV.HVEWdir_comb1 = [HV.HVEWdir_comb1;EVmean.'];
                                end
                            end
                            NSv = [];
                            VEv = [];
                            EWv = [];
                        end
                    end
                end

                %% Figura
                if teta == 0
                    h = figure(201);
                    for ic = 1:length(HV.clavecomb)
                        % HVmeansmooth = suavmatr(HV.HVmean_comb{ic},HV.fcomb{ic},0,24); %length(HV.
HVmean_comb{ic})*0.1
                        semilogx(HV.fcomb{ic},HV.HVmean_comb{ic},'linewidth',1.5); hold on; grid on
                        % plot(HV.fcomb{ic},HVmeansmooth,'linewidth',1.5); hold on %,'color',col(ic,
:) hold on; grid on
                        % semilogx(HV.fcomb{ic},HV.NVmean_comb{ic},'linewidth',1.5); hold on; grid o
n
                        % semilogx(HV.fcomb{ic},HV.EVmean_comb{ic},'linewidth',1.5); hold on; grid o
n
                    end
                    if isempty(ic); ic = 0; end
                    leyenda = [leyenda;HV.clavecomb];
                    leg = legend(leyenda);
                    title(['HVSR ',estac],'fontname','Liberation Serif','fontSize',12,'interpreter','none')
                    set(gca,'fontname','Liberation Serif','fontSize',12)
                    set(gcf,'color','white')
                    xlabel('Frecuencia (Hz)','fontname','Liberation Serif','fontSize',12)
                    ylabel('Amplitud H/V','fontname','Liberation Serif','fontSize',12)
                    maxyplot = get(gca,'ytick');
                    xlim([min(f) max(f)]); %[0.01 10]
                    % set(gca,'xtick',[0,1:1:10]) %[0.1,1:1:10]
                    drawnow

                    % print(gcf,nombgrab0(1:end-4),'-dpng','-r600')
                    % close(h)
```

```
            end
        end
        save(nombgrab0,'HV','-v7.3');

        % % Figura HV direccional
        % contourf(HV.fcomb{1},HV.tetarot,HV.HVNSdir_comb1); shading interp
        % xlabel('Frecuencia (Hz)','fontname','Liberation Serif','fontSize',12)
        % ylabel('Ángulo de rotación (deg)','fontname','Liberation Serif','fontSize',12)
        % view([0 0 1])
        % xlim([0.01 HV.fcomb{1}(end)])
        % ylim([0 180])
        % colormap(jet)
        %
        % legtetaevec = num2str(HV.tetarot.');
        % legend(legtetaevec,'interpreter','none')
    end
  end
end
```