Appendix A

Matlab code for topology case 3

```
clearvars; close all; clc

%% data

% system data:
LCL_L1o = 1.2; % H
LCL_R1o = 0.0; % ohm
LCL_L2o = 0.641; % H
LCL_R2o = 0.0; % ohm
LCL_Co = 1.491e-7; % F
LCL_Rco = 0; % ohm
Tr1_Lo = 51.568e-3; % H

Cable33_Lo = 18.181e-3; % H
Cable33_Ro = 0.372; % ohm
Cable33_C1o = 5.709e-8; % F
Cable33_C2o = Cable33_C1o; % F
Tr2_Lo = 38.676e-3; % H

Cable150_Lo = 1e-3; % H
Cable150_Ro = 0.056; % ohm
Cable150_C1o = 0.52e-6; % F
Cable150_C2o = Cable150_C1o; % F
Tr3_Lo = 19.338e-3; % H
Tuned_Co = 5.658e-6; % F
PhReact_Lo = 19.3e-3; % H

%% ------- settings -------
f = 50;
w = 2*pi*f;
res = 0.01;
H = res:res:30;
%                Zmodels f.sweep HMA     Stability-bode
calculate = [   true     true    true     true];

colormodels = [56  18  77;
            196 141 227;
            242 0   52]/255;

fprintf('--------------- CASE 3 (4 WT); resolution: %f --------------- \n', res);

ymax = 50e3;
view = [0,H(length(H)),0,ymax];
% view = [9 14 0 100]; % zoom
```

```
xlab = 'Harmonics Order';
ylab = 'Impedance [Ohm]';

%% convertion to equivalent circuit at V=150kV
Vout = 150e3;

Vlow = 150e3;
Vmid = 150e3;
Vhigh = 150e3;
LCL_L1 = ind_equiv(LCL_L1o,f,Vlow,Vout);
LCL_R1 = res_equiv(LCL_R1o,Vlow,Vout);
LCL_L2 = ind_equiv(LCL_L2o,f,Vlow,Vout);
LCL_R2 = res_equiv(LCL_R2o,Vlow,Vout);
LCL_C  = cap_equiv(LCL_Co,f,Vlow,Vout);
LCL_Rc = res_equiv(LCL_Rco,Vlow,Vout);
Tr1_L  = ind_equiv(Tr1_Lo,f,Vlow,Vout);

Cable33_L  = ind_equiv(Cable33_Lo,f,Vmid,Vout);
Cable33_R  = res_equiv(Cable33_Ro,Vmid,Vout);
Cable33_C1 = cap_equiv(Cable33_C1o,f,Vmid,Vout);
Cable33_C2 = cap_equiv(Cable33_C2o,f,Vmid,Vout);
Tr2_L = ind_equiv(Tr2_Lo,f,Vmid,Vout);

Cable150_L = ind_equiv(Cable150_Lo,f,Vhigh,Vout);
Cable150_R = res_equiv(Cable150_Ro,Vhigh,Vout);
Cable150_C1 = cap_equiv(Cable150_C1o,f,Vhigh,Vout);
Cable150_C2 = cap_equiv(Cable150_C2o,f,Vhigh,Vout);
Tr3_L = ind_equiv(Tr3_Lo,f,Vhigh,Vout);
Tuned_C = cap_equiv(Tuned_Co,f,Vhigh,Vout);
PhReact_L = ind_equiv(PhReact_Lo,f,Vhigh,Vout);

%% loading converters models
if calculate(1)==true
    S_WTconverter = 100e6;
    [Zwt_p, Zwt_n, ~,~, Zhvdc_p, Zhvdc_n, ~,~] = convertersImpedanceModel(H,S_WTconverter);
    fprintf('Converters models: WTs and HVDC loaded! \n');

    Hf = H*f;
    figure(1); hold on;

    subplot(2,2,1)
    pos = semilogx(Hf,mag2db(abs(Zwt_p)), 'LineWidth', 1); hold on
    neg = semilogx(Hf,mag2db(abs(Zwt_n)), 'r', 'LineWidth', 1); hold off
    legend([pos,neg], 'positive', 'negative','Location','SouthEast');
    title('Impedance of aggregated WT converter (100MW)');
    xlabel('Frequency [Hz]');
    ylabel('Impedance magnitude [dB]');
    V = axis;
    axis([Hf(1), Hf(end), V(3), V(4)]);
    clear pos neg V
    grid on

    subplot(2,2,2)
    pos = semilogx(Hf,mag2db(abs(Zhvdc_p)), 'LineWidth', 1); hold on
    neg = semilogx(Hf,mag2db(abs(Zhvdc_n)), 'r', 'LineWidth', 1); hold off
    legend([pos,neg], 'positive', 'negative','Location','SouthEast');
    title('Impedance of HVDC converter');
    xlabel('Frequency [Hz]');
    ylabel('Impedance magnitude [dB]');
    V = axis;
    axis([Hf(1), Hf(end), V(3), V(4)]);
    clear pos neg V
    grid on
    subplot(2,2,3)
    pos = semilogx(Hf,rad2deg(angle(Zwt_p)), 'LineWidth', 1); hold on
    neg = semilogx(Hf,rad2deg(angle(Zwt_n)), 'r', 'LineWidth', 1); hold off
    legend([pos,neg], 'positive', 'negative')
    title('Impedance of aggregated WT converter (100MW)');
    xlabel('Frequency [Hz]');
    ylabel('Impedance angle [deg.]');
    V = axis;
    axis([Hf(1), Hf(end), V(3), V(4)]);
    clear pos neg V
    grid on

    subplot(2,2,4)
    pos = semilogx(Hf,rad2deg(angle(Zhvdc_p)), 'LineWidth', 1); hold on
    neg = semilogx(Hf,rad2deg(angle(Zhvdc_n)), 'r', 'LineWidth', 1); hold off
    legend([pos,neg], 'positive', 'negative')
    title('Impedance of HVDC converter');
    xlabel('Frequency [Hz]');
    ylabel('Impedance angle [deg.]');
    V = axis;
    axis([Hf(1), Hf(end), V(3), V(4)]);
    clear pos neg V
    grid on

    hold off
end

%% Impedance - 4 legs (seen from middle LCL point)
if calculate(2)==true
    fprintf('\n\n-------- Frequency Sweep --------\n\n')
    ZabsM3 = zeros(length(H),3);
    for model = 1:3;  % 1-VS; 2-CS (WT conv.),3-Z(s)
        for hh=1:length(H)
            h=H(hh);
            s = 1i*h*w;

            if (model==1 || model==2)
                Z1 = imp_parallel(s*PhReact_L, 1/(s*Tuned_C)) + s*Tr3_L;
            elseif model==3
                Z1 = Zhvdc_p(hh) + s*Tr3_L;
            end

            if model==1
                Z1B = imp_parallel(LCL_R1+s*LCL_L1,1/(s*LCL_C))+LCL_R2+s*LCL_L2+s*Tr1_L;
            elseif model==2
                Z1B = imp_parallel(10e8,1/(s*LCL_C))+LCL_R2+s*LCL_L2+s*Tr1_L;
            elseif model==3
                Z1B = imp_parallel(Zwt_p(hh),1/(s*LCL_C))+LCL_R2+s*LCL_L2+s*Tr1_L;
            end

            Z2B = imp_parallel(Z1B, 1/(s*Cable33_C1)) + Cable33_R+s*Cable33_L;
            Z3B = imp_parallel(Z2B, 1/(s*Cable33_C2)) + s*Tr2_L;

            Z1C = imp_parallel(Z3B, Z3B);
            Z2C = imp_parallel(Z1C, 1/(s*Cable150_C1)) + Cable150_R+s*Cable150_L;
            Z3C = imp_parallel(Z2C, 1/(s*Cable150_C2));

            Z1A = imp_parallel(Z1,Z3C);
            Z2 = imp_parallel(Z1A, 1/(s*Cable150_C2)) + Cable150_R+s*Cable150_L;
            Z3 = imp_parallel(Z2, 1/(s*Cable150_C1));

            Z4 = imp_parallel(Z3B, Z3) + s*Tr2_L;
            Z5 = imp_parallel(Z4, 1/(s*Cable33_C2)) + Cable33_R+s*Cable33_L;
            Z6 = imp_parallel(Z5, 1/(s*Cable33_C1)) + s*Tr1_L + s*LCL_L2+LCL_R2;
            Z7 = imp_parallel(Z6, 1/(s*LCL_C));
            if model==1
                Z8 = LCL_R1+s*LCL_L1;
            elseif model==2
                Z8 = 10e8;
            elseif model==3
                Z8 = Zwt_p(hh);
            end
            Z = imp_parallel(Z7, Z8);

            Zabs = abs(Z);
            ZabsM3(hh,model) = Zabs;
        end
    end
    clear hh h Z1 Z2 Z3 Z4 Z5 Z6 Z7 Z8 Z1B Z2B Z3B Z1A Z1C Z2C Z3C Z Zabs

    fig2 = figure(2);
    fig2.Position = [292 180 759 489];
    model1 = plot(H,ZabsM3(:,1), 'LineWidth', 1,...
        'Color',[colormodels(1,1), colormodels(1,2), colormodels(1,3)]); hold on
    model2 = plot(H,ZabsM3(:,2), 'LineWidth', 1,...
        'Color',[colormodels(2,1), colormodels(2,2), colormodels(2,3)]);
    model3 = plot(H,ZabsM3(:,3), 'LineWidth', 1,...
        'Color',[colormodels(3,1), colormodels(3,2), colormodels(3,3)]);
    axis(view)
    title('Frequency sweep - seen from LCL filter middle bus');
    xlabel(xlab);
    ylabel(ylab);
    legend([model1,model2,model3], 'VS model','CS-WT model', 'Z(s) model');
    grid on
    hold off
    clear model1 model2 model3;

    fig3 = figure(3);
    fig3.Position = [292 180 759 489];
    model1 = semilogy(H,ZabsM3(:,1), 'LineWidth', 1,...
        'Color',[colormodels(1,1), colormodels(1,2), colormodels(1,3)]); hold on
    model2 = semilogy(H,ZabsM3(:,2), 'LineWidth', 1,...
        'Color',[colormodels(2,1), colormodels(2,2), colormodels(2,3)]);
    model3 = semilogy(H,ZabsM3(:,3), 'LineWidth', 1,...
        'Color',[colormodels(3,1), colormodels(3,2), colormodels(3,3)]);
    title('Frequency sweep - seen from LCL filter middle bus');
    xlabel(xlab);
    ylabel(strcat(ylab,' (log axis)'));
    legend([model1,model2,model3], 'VS model','CS-WT model', 'Z(s) model');
    grid on
    hold off
    clear model1 model2 model3;

    [zz1,ww1] = findpeaks(ZabsM3(:,1));
    [zz2,ww2] = findpeaks(ZabsM3(:,2));
    [zz3,ww3] = findpeaks(ZabsM3(:,3));
    fprintf('\nVS model:')
    fprintf('\n- for h.order: %f, peak impedance: %f',[ww1'*res; zz1'])
    fprintf('\nCS model:')
    fprintf('\n- for h.order: %f, peak impedance: %f',[ww2'*res; zz2'])
    fprintf('\nZ(s) model:')
    fprintf('\n- for h.order: %f, peak impedance: %f',[ww3'*res; zz3'])

    clear zz1 ww1 zz2 ww2 zz3 ww3
end
```

```
%% harmonics modal analysis - model 1 (no conv. models)
if calculate(3)==true
    fprintf('\n\n-------- HMA - without converter models --------\n\n')
    % admittance matrix
    n_bus = 20;
    n_h = length(H);
    ZmodalMaxMSave = zeros(n_h,3);
    modelStr = {'VS', 'CS', 'Z(s)'};
    Saver = {[] [] []};
    for model=1:3
        fprintf(strcat('\n ---- HMA: \t',modelStr{model}, ' model ----\n'));
        ZmodalMaxTable = zeros(n_h,4); % harm order, mode of max imp, modal imp abs, angle
        ZmodalMaxM = zeros(n_h,1);
        ZmodalAllM = zeros(n_h,n_bus); % all modes (impedances)
        PFmodalM = zeros(n_h,n_bus); % PFs for each bus for each harmonic
        eM = zeros(n_bus,n_h);

        for hh = 1:length(H)
            h = H(hh);
            s = 1i*h*w;

            if (model==1 || model==2)
                y1_1 = 1/(s*PhReact_L) + s*Tuned_C + 1/(s*Tr3_L);
            elseif model==3
                y1_1 = 1/(Zhvdc_p(hh)) + 1/(s*Tr3_L);
            end
            y2_2 = 1/(s*Tr3_L) + 2*s*Cable150_C2 + 2*1/(Cable150_R+s*Cable150_L);
            y3_3 = 1/(Cable150_R+s*Cable150_L) + s*Cable150_C1 + 2*1/(s*Tr2_L);

            y4_4 = 1/(s*Tr2_L) + s*Cable33_C2 + 1/(Cable33_R+s*Cable33_L);
            y5_5 = 1/(Cable33_R+s*Cable33_L) + s*Cable33_C2 + 1/(s*Tr1_L);
            y6_6 = 1/(s*Tr1_L) + 1/(LCL_R2+s*LCL_L2);
            if model==1
                y7_7 = 1/(LCL_R2+s*LCL_L2) + s*LCL_C+LCL_Rc + 1/(LCL_R1+s*LCL_L1);
            elseif model==2
                y7_7 = 1/(LCL_R2+s*LCL_L2) + s*LCL_C+LCL_Rc;
            elseif model==3
                y7_7 = 1/(LCL_R2+s*LCL_L2) + s*LCL_C+LCL_Rc + 1/(Zwt_p(hh));
            end
            y8_8 = y4_4;
            y9_9 = y5_5;
            y10_10 = y6_6;
            y11_11 = y7_7;

            y12_12 = y3_3;

            y13_13 = y4_4;
            y14_14 = y5_5;
            y15_15 = y6_6;
            y16_16 = y7_7;

            y17_17 = y4_4;
            y18_18 = y5_5;
            y19_19 = y6_6;
            y20_20 = y7_7;

            y1_2 = 1/(s*Tr3_L);

            y2_3 = 1/(Cable150_R+s*Cable150_L);
            y2_12 = y2_3;

            y3_4 = 1/(s*Tr2_L);
```

```
y3_8 = y3_4;
y12_13 = y3_4;
y12_17 = y3_4;

y4_5 = 1/(Cable33_R+s*Cable33_L);
y5_6 = 1/(s*Tr1_L);
y6_7 = 1/(LCL_R2+s*LCL_L2);
y8_9 = y4_5;
y9_10 = y5_6;
y10_11 = y6_7;
y13_14 = y4_5;
y14_15 = y5_6;
y15_16 = y6_7;
y17_18 = y4_5;
y18_19 = y5_6;
y19_20 = y6_7;

Y = [y1_1 -y1_2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
    -y1_2 y2_2 -y2_3 0 0 0 0 0 0 0 0 -y2_12 0 0 0 0 0 0 0 0;...
    0 -y2_3 y3_3 -y3_4 0 0 0 -y3_8 0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 -y3_4 y4_4 -y4_5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 -y4_5 y5_5 -y5_6 0 0 0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 -y5_6 y6_6 -y6_7 0 0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 -y6_7 y7_7 0 0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 -y3_8 0 0 0 0 y8_8 -y8_9 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 -y8_9 y9_9 -y9_10 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 -y9_10 y10_10 -y10_11 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 -y10_11 y11_11 0 0 0 0 0 0 0 0 0;...
    0 -y2_12 0 0 0 0 0 0 0 0 0 y12_12 -y12_13 0 0 0 -y12_17 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 -y12_13 y13_13 -y13_14 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0 -y13_14 y14_14 -y14_15 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0 0 -y14_15 y15_15 -y15_16 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 -y15_16 y16_16 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 -y12_17 0 0 0 0 y17_17 -y17_18 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -y17_18 y18_18 -y18_19 0;...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -y18_19 y19_19 -y19_20;...
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -y19_20 y20_20];

e = eig(Y); % eigenvalues
eM(:,hh) = e;
[T,A] = eig(Y); % T - rigth eigenvector matrix
L = inv(T); % L - left eigenvector matrix

ZmodalAll = abs(inv(A));
for zz = 1:n_bus
    ZmodalAllM(hh,zz) = ZmodalAll(zz,zz);
end

[lambdaMin,mode] = min(abs(e));
ZmodalMax = 1/lambdaMin;
em = e(mode);
ang = rad2deg(angle(em));

ZmodalMaxTable(hh,1) = h;
ZmodalMaxTable(hh,2) = mode;
ZmodalMaxTable(hh,3) = ZmodalMax;
ZmodalMaxTable(hh,4) = ang;

ZmodalMaxM(hh) = ZmodalMax;

for b=1:n_bus
    PFmodalM(hh,b) = abs(L(mode,b)*T(b,mode));
end
```

```
end
fig4 = figure(4);
fig4.Position = [244 13 847 700];
subplot(3,1,model)
plot(H,ZmodalAllM);
axis(view);
str = strcat('HMA (all modes):',modelStr{model},' model');
title(str);
xlabel(xlab);
ylabel(ylab);
legend(num2str([1:n_bus]'),'Location','NorthEast');
grid on

fig5 = figure(5);
fig5.Position = [244 13 847 700];
subplot(3,1,model)
plot(H,ZmodalMaxM, 'Color',[colormodels(model,1) colormodels(model,2) colormodels(model
axis(view)
str = strcat('HMA (only max. modes):',modelStr{model},' model');
title(str);
xlabel(xlab);
ylabel(ylab);
grid on

[Z_peak,h_crit_idx] = findpeaks(ZmodalMaxTable(:,3));
h_crit = h_crit_idx * res;

fprintf('harmonic order - critical mode - modal impedance(abs) - angle\n');
ZmodalHcrit = ZmodalMaxTable(h_crit_idx,:)

fprintf('harmonic order - participation factors for all buses\n');
PFmodalHcrit = PFmodalM(h_crit_idx,:)
sum(PFmodalHcrit,2)
Saver{model} = PFmodalHcrit;

fprintf('--> Eigenvalues of critical frequencies\n');
eMcrit = eM(:,h_crit_idx);
abs(eMcrit.^-1)

fprintf('greates participation factors:\n');
for ff=1:length(PFmodalHcrit(:,1))
    M = PFmodalHcrit(ff,:);
    mx = max(M);
    tmx = find(M==mx);
    omx1 = 0.99999*mx;
    omx2 = 1.00001*mx;
    omx = setdiff(intersect(find(M>omx1),find(M<omx2)),tmx);

    fprintf('For harmonic: %f, bus: %s has greatest PF=%f.\n',...
        ZmodalHcrit(ff,1), num2str(tmx), mx);
    if ~isempty(omx)
        fprintf('\t(also same PF at buses: ');
        fprintf('%s', num2str(omx));
        fprintf(')\n');
    end
end

top_modes = unique(ZmodalMaxTable(h_crit_idx,2));
top_modes_impedances = ZmodalAllM(:,top_modes);

fig6 = figure(6);
fig6.Position = [244 13 847 700];
```

```
        subplot(3,1,model)
        plot(H,top_modes_impedances);
        str = strcat('HMA (critical modes only):',modelStr{model},' model');
        title(str);
        legend(num2str(top_modes));
        xlabel(xlab);
        ylabel(ylab);
        grid on

        ZmodalMaxMSave(:,model) = ZmodalMaxM;
    end
end

%% Stability
nyqview = 1.5;
Hstab = 0.2:res:40; % order
Hfstab = Hstab*f; % Hz

if calculate(4)==true
    fprintf('\n\n-------- Stability of the system --------\n')
    [Zwt_p, Zwt_n, ~, ~, Zhvdc_p, Zhvdc_n, ~, ~] = convertersImpedanceModel(Hstab,S_WTconverter);

    %% BODE
    ZspM = zeros(1,length(Hstab));
    ZsnM = zeros(1,length(Hstab));
    Ztr3M = zeros(1,length(Hstab));
    for hh = 1:length(Hstab)
        h = Hstab(hh);
        s = 1i*h*w;

        Z1p = imp_parallel(Zwt_p(hh),1/(s*LCL_C)) + ...
            LCL_R2 + s*(LCL_L2+Tr1_L);
        Z2p = imp_parallel(Z1p,1/(s*Cable33_C1))+Cable33_R+s*Cable33_L;
        Z3p = imp_parallel(Z2p,1/(s*Cable33_C2))+s*Tr2_L;
        Z3pB = imp_parallel(Z3p,Z3p); % incl. second branch
        Z4p = imp_parallel(Z3pB,1/(s*Cable150_C1))+Cable150_R+s*Cable150_L;
        Z5p = imp_parallel(Z4p,1/(s*Cable150_C2));

        Zsp = imp_parallel(Z5p,Z5p);
        ZspM(hh) = Zsp;

        Z1n = imp_parallel(Zwt_n(hh),1/(s*LCL_C)) + ...
            LCL_R2 + s*(LCL_L2+Tr1_L);
        Z2n = imp_parallel(Z1n,1/(s*Cable33_C1))+Cable33_R+s*Cable33_L;
        Z3n = imp_parallel(Z2n,1/(s*Cable33_C2))+s*Tr2_L;
        Z3nB = imp_parallel(Z3n,Z3n); % incl. second branch
        Z4n = imp_parallel(Z3nB,1/(s*Cable150_C1))+Cable150_R+s*Cable150_L;
        Z5n = imp_parallel(Z4n,1/(s*Cable150_C2));
        Zsn = imp_parallel(Z5n,Z5n);
        ZsnM(hh) = Zsn;

        Ztr3M(hh) = s*Tr3_L;
    end
    clear Z1p Z2p Z3p Z4p Zsp Z1n Z2n Z3n Z4n Zsn Z3pB Z3nB

    Zlp = Zhvdc_p + Ztr3M;
    Zln = Zhvdc_n + Ztr3M;

    fig7 = figure(7);
    fig7.Position = [244 115 847 576];
    subplot(2,1,1)
    pos = semilogx(Hfstab,mag2db(abs(ZspM)), 'b','LineWidth', 1); hold on
    neg = semilogx(Hfstab,mag2db(abs(ZsnM)), 'b--', 'LineWidth', 1);
```

```
grdpos = semilogx(Hfstab,mag2db(abs(Zlp)), 'r', 'LineWidth', 1);
grdneg = semilogx(Hfstab,mag2db(abs(Zln)), 'r--', 'LineWidth', 1); hold off
grid on
    title('Bode diagram of frequency dependent impedance of "source" and "grid"');
V = axis;
axis([10, 2000, V(3), V(4)]);
legend([pos,neg,grdpos,grdneg], 'pos. source', 'neg. source',...
    'pos. grid', 'neg. grid', 'Location', 'EastOutside');
xlabel('Frequency [Hz]');
ylabel('Magnitude [dB]');
clear pos neg grdpos grdneg V

subplot(2,1,2)
pos = semilogx(Hfstab,rad2deg(angle(ZspM)), 'b','LineWidth', 1); hold on
neg = semilogx(Hfstab,rad2deg(angle(ZsnM)), 'b--', 'LineWidth', 1);
grdpos = semilogx(Hfstab,rad2deg(angle(Zlp)), 'r', 'LineWidth', 1);
grdneg = semilogx(Hfstab,rad2deg(angle(Zln)), 'r--', 'LineWidth', 1); hold off
grid on
V = axis;
axis([10, 2000, V(3), V(4)]);
legend([pos,neg,grdpos,grdneg], 'pos. source', 'neg. source',...
    'pos. grid', 'neg. grid', 'Location', 'EastOutside');
xlabel('Frequency [Hz]');
ylabel('Angle [deg]');
clear pos neg grdpos grdneg V

%% Intersection of curves and stability margin
eps = 0.1;
% points at less than 1.5 order exluded
fundamental_idx = find(Hstab==1);
start_idx = fundamental_idx + 0.5/res;
Hstab_idx = 1:length(Hstab);

fprintf('\n --- Intersections of grid positive with source pos and neg ---');
% Find point of intersection - Grid/Load positive
y1p = abs(ZspM);
y1n = abs(ZsnM);
y2 = abs(Zlp);
distance_p = abs(y1p-y2);
distance_n = abs(y1n-y2);
idx_p = find(distance_p < eps);
idx_n = find(distance_n < eps);

idx_filtered_p = intersect(idx_p,Hstab_idx(start_idx:end));
idx_filtered_n = intersect(idx_n,Hstab_idx(start_idx:end));

px_pp = Hfstab(idx_filtered_p);
py_pp = y1p(idx_filtered_p);
px_np = Hfstab(idx_filtered_n);
py_np = y1n(idx_filtered_n);

% phase margins = 180-fi
fi1_p = angle(ZspM);
fi1_n = angle(ZsnM);
fi2 = angle(Zlp);
fi_diff_p = abs(rad2deg(fi1_p-fi2));
fi_diff_n = abs(rad2deg(fi1_n-fi2));
ph_margin_p = 180 - fi_diff_p(idx_filtered_p);
ph_margin_n = 180 - fi_diff_n(idx_filtered_n);
order_filtered_p = Hstab(idx_filtered_p);
order_filtered_n = Hstab(idx_filtered_n);
distance_p = distance_p';
distance_n = distance_n';
```

```
    MarginsP = [order_filtered_p' [50*order_filtered_p]'...
        ph_margin_p' distance_p(idx_filtered_p)]
    MarginsN = [order_filtered_n' [50*order_filtered_n]'...
        ph_margin_n' distance_n(idx_filtered_n)]
    clear idx_filtered_p idx_filtered_n

    fprintf('\n --- Intersections of grid negative with source pos and neg ---');
    % Find point of intersection - Grid/Load negative
    y1p = abs(ZspM);
    y1n = abs(ZsnM);
    y2 = abs(Zln);
    distance_p = abs(y1p-y2);
    distance_n = abs(y1n-y2);
    idx_p = find(distance_p < eps);
    idx_n = find(distance_n < eps);

    idx_filtered_p = intersect(idx_p,Hstab_idx(start_idx:end));
    idx_filtered_n = intersect(idx_n,Hstab_idx(start_idx:end));

    px_pn = Hfstab(idx_filtered_p);
    py_pn = y1p(idx_filtered_p);
    px_nn = Hfstab(idx_filtered_n);
    py_nn = y1n(idx_filtered_n);

    % phase margins = 180-fi
    fi1_p = angle(ZspM);
    fi1_n = angle(ZsnM);
    fi2 = angle(Zln);
    fi_diff_p = abs(rad2deg(fi1_p-fi2));
    fi_diff_n = abs(rad2deg(fi1_n-fi2));
    ph_margin_p = 180 - fi_diff_p(idx_filtered_p);
    ph_margin_n = 180 - fi_diff_n(idx_filtered_n);
    order_filtered_p = Hstab(idx_filtered_p);
    order_filtered_n = Hstab(idx_filtered_n);
    distance_p = distance_p';
    distance_n = distance_n';
    MarginsP = [order_filtered_p' [50*order_filtered_p]'...
        ph_margin_p' distance_p(idx_filtered_p)]
    MarginsN = [order_filtered_n' [50*order_filtered_n]'...
        ph_margin_n' distance_n(idx_filtered_n)]

    figure(7)
    subplot(2,1,1);
    hold on
    plot(px_pp, mag2db(py_pp), 'o', 'MarkerSize', 10, 'Color', [0.8 0.1 0.5]);
    plot(px_np, mag2db(py_np), 'o', 'MarkerSize', 10, 'Color', [0.8 0.1 0.5]);
    plot(px_pn, mag2db(py_pn), 'o', 'MarkerSize', 10, 'Color', [0.8 0.1 0.5]);
    plot(px_nn, mag2db(py_nn), 'o', 'MarkerSize', 10, 'Color', [0.8 0.1 0.5]);
    hold off

end

%% Save variables
FS_ZM3 = ZabsM3; % FS data
HMA_ZmaxM3 = ZmodalMaxMSave; % HMA max modes only data
save('results/fromCase3.mat', 'FS_ZM3', 'HMA_ZmaxM3');
```