function Spec = CPS\_W(y,x,alpha,nfft,Noverlap,Window,opt,P)

% Spec = CPS\_W(y,x,alpha,nfft,Noverlap,Window,opt)

% Welch's estimate of the (Cross) Cyclic Power Spectrum of signals y

% and x at cyclic frequency alpha:

% opt = 'sym' : symmetric version E{Y(f+alpha/2)X\*(f-alpha/2)}/W

% opt = 'asym': asymmetric version E{Y(f)X\*(f-alpha)}/W

% (f and alpha are normalized by the sampling frequency and constrained to take values between 0 and 1)

% x and y are divided into K overlapping blocks (Noverlap taps), each of which is

% detrended, windowed and zero-padded to length nfft. Input arguments nfft, Noverlap, and Window

% are as in function 'PSD' or 'PWELCH' of Matlab. Denoting by Nwind the window length, it is recommended to use

% nfft = 2\*NWind and Noverlap = 2/3\*Nwind with a hanning window or Noverlap = 1/2\*Nwind with a half-sine window.

% Note: use analytic signal to avoid correlation between + and - frequencies

%

% -------

% Outputs

% -------

% Spec is a structure organized as follows:

% Spec.S = Cyclic Power Spectrum vector

% Spec.f = vector of normalized frequencies

% Spec.K = number of blocks

% Spec.Var\_Reduc = Variance Reduction factor

%

% Spec = CPS\_W(y,x,...,P) where P is a scalar between 0 and 1,

% also returns Spec.CI the P\*100% confidence interval for Spec.S

% (requires function 'chi2inv' of the statistical toolbox of Matlab)

%

% --------------------------

% Reference: J. Antoni, "Cyclic Spectral Analysis in Practice", Mechanical Systems and Signal Processing, Volume 21, Issue 2, February 2007, Pages 597-630.

% --------------------------

% Author: J. Antoni

% Last Revision: 12-2014

% --------------------------

if length(Window) == 1

Window = hanning(Window);

end

Window = Window(:);

n = length(x); % Number of data points

nwind = length(Window); % length of window

% check inputs

if (alpha>1||alpha<0),error('alpha must be in [0,1] !!'),end

if nwind <= Noverlap,error('Window length must be > Noverlap');end

if nfft < nwind,error('Window length must be <= nfft');end

if nargin > 7 && (P>=1 || P<=0),error('P must be in ]0,1[ !!'),end

y = y(:);

x = x(:);

K = fix((n-Noverlap)/(nwind-Noverlap)); % Number of windows

% compute CPS

index = 1:nwind;

f = (0:nfft-1)/nfft;

t = (0:n-1)';

CPS = 0;

if strcmp(opt,'sym') == 1

y = y.\*exp(-1i\*pi\*alpha\*t);

x = x.\*exp(1i\*pi\*alpha\*t);

else

x = x.\*exp(2i\*pi\*alpha\*t);

end

for i=1:K

xw = Window.\*x(index);

yw = Window.\*y(index);

Yw1 = fft(yw,nfft); % Yw(f+a/2) or Yw(f)

Xw2 = fft(xw,nfft); % Xw(f-a/2) or Xw(f-a)

CPS = Yw1.\*conj(Xw2) + CPS;

index = index + (nwind - Noverlap);

end

% normalize

KMU = K\*norm(Window)^2; % Normalizing scale factor ==> asymptotically unbiased

CPS = CPS/KMU;

% variance reduction factor

Window = Window(:)/norm(Window);

Delta = nwind - Noverlap;

R2w = xcorr(Window);

k = nwind+Delta:Delta:min(2\*nwind-1,nwind+Delta\*(K-1));

if length(k) >1

Var\_Reduc = R2w(nwind)^2/K + 2/K\*(1-(1:length(k))/K)\*(R2w(k).^2);

else

Var\_Reduc = R2w(nwind)^2/K;

end

% confiance interval

if nargin > 7

v = 2/Var\_Reduc;

alpha = 1 - P;

if alpha == 0 % Sa ~ Chi2

temp = 1./chi2inv([1-alpha/2 alpha/2],round(v));

CI = v\*CPS\*temp;

else % Sa ~ Normal

Sy = CPS\_W(y,y,0,nfft,Noverlap,Window,opt);

Sx = CPS\_W(x,x,0,nfft,Noverlap,Window,opt);

Var\_Sa = Sy.S.\*Sx.S/v;

temp = sqrt(2)\*erfinv(2\*P-1);

CI = CPS\*[1 1] + temp\*sqrt(Var\_Sa(:))\*[1 -1];

end

end

% set up output parameters

if nargout == 0

figure,newplot;

plot(f,10\*log10(abs(CPS))),grid on

xlabel('[Hz]'),title('Spectral Correlation Density (dB)')

else

Spec.S = CPS;

Spec.f = f;

Spec.K = K;

Spec.Var\_Reduc = Var\_Reduc;

if nargin > 7

Spec.CI = CI;

end

end

function Coh = SCoh\_W(y,x,alpha,nfft,Noverlap,Window,opt,P)

% Coh = SCoh\_W(y,x,alpha,nfft,Noverlap,Window,opt,P)

% Welch's estimate of the Cyclic Spectral Coherence of signals y and x

% at cyclic frequency alpha (normalized by sampling frequency):

% opt = 'sym' : symmetric version E{Y(f+alpha/2)X\*(f-alpha/2)}/sqrt(E|Y(f+alpha/2)|¬≤E|X\*(f-alpha/2)|¬≤)

% opt = 'asym': asymmetric version E{Y(f)X\*(f-alpha)}/sqrt(E|Y(f)|¬≤E|X\*(f-alpha)|¬≤)

% (f and alpha are normalized by the sampling frequency and constrained to take values between 0 and 1)

% x and y are divided into K overlapping blocks (Noverlap taps), each of which is

% detrended, windowed and zero-padded to length nfft. Input arguments nfft, Noverlap, and Window

% are as in function 'PSD' or 'PWELCH' of Matlab. Denoting by Nwind the window length, it is recommended to use

% nfft = 2\*NWind and Noverlap = 2/3\*Nwind with a hanning window or Noverlap = 1/2\*Nwind with a half-sine window.

% Note: use analytic signal to avoid correlation between + and - frequencies

%

% SCoh\_W calls function 'CPS\_W'.

%

% -------

% Outputs

% -------

% Coh is a structure organized as follows:

% Coh.C = Cyclic Spectral Coherence

% Coh.Syx = Cross Cyclic Spectral Density

% Coh.Sy and Coh.Sx = Cyclic Spectral Densities of signals y and x %cared about

% Coh.f = vector of frequencies

% Coh.K = number of blocks

% Coh.Var\_Reduc = Variance Reduction factor

% Coh = CPS\_W(y,x,...,P) where P is a scalar between 0 and 1,

% also returns Coh.thres the P\*100% significance level

% for testing against H0: |Coh.S|^2 = 0

% (requires function 'chi2inv' of the statistical toolbox of Matlab)

%

% --------------------------

% Reference: J. Antoni, "Cyclic Spectral Analysis in Practice", Mechanical Systems and Signal Processing, Volume 21, Issue 2, February 2007, Pages 597-630.

% --------------------------

% Author: J. Antoni

% Last Revision: 12-2014

% --------------------------

if length(Window) == 1

Window = hanning(Window);

end

Window = Window(:);

n = length(x);

nwind = length(Window);

% check inputs

if (alpha>1)||(alpha<0),error('alpha must be in [0,1] !!'),end

if nwind <= Noverlap,error('Window length must be > Noverlap');end

if nfft < nwind,error('Window length must be <= nfft');end

if (nargin>7) && (P>=1 || P<=0),error('P must be in ]0,1[ !!'),end

y = y(:);

x = x(:);

t = (0:n-1)';

if strcmp(opt,'sym') == 1

y = y.\*exp(-1i\*pi\*alpha\*t);

x = x.\*exp(1i\*pi\*alpha\*t);

else

x = x.\*exp(2i\*pi\*alpha\*t);

end

%key steps in cyclic spectral analysis

Syx = CPS\_W(y,x,0,nfft,Noverlap,Window,opt);

Sy = CPS\_W(y,y,0,nfft,Noverlap,Window,opt);

Sx = CPS\_W(x,x,0,nfft,Noverlap,Window,opt);

Coh.K = Sx.K;

Coh.f = Sx.f;

Coh.Syx = Syx.S;

Coh.Sy = Sy.S;

Coh.Sx = Sx.S;

Coh.C = Syx.S./sqrt(Sy.S.\*Sx.S);

% variance reduction factor

Window = Window(:)/norm(Window);

Delta = nwind - Noverlap;

R2w = xcorr(Window);

k = nwind+Delta:Delta:min(2\*nwind-1,nwind+Delta\*(Coh.K-1));

if length(k) >1

Coh.Var\_Reduc = R2w(nwind)^2/Coh.K + 2/Coh.K\*(1-(1:length(k))/Coh.K)\*(R2w(k).^2);

else

Coh.Var\_Reduc = R2w(nwind)^2/Coh.K;

end

% threshold on square magnitude at P significance level under H0

if nargin > 7

Coh.thres = chi2inv(1-P,2)\*Coh.Var\_Reduc/2;

end

% set up output parameters

if nargout == 0

figure,newplot;

subplot(211),plot(Coh.f(1:nfft/2+1),abs(Coh.C(1:nfft/2+1)).^2), grid on

if nargin > 7,

hold on,plot(Coh.f(1:nfft/2+1),Coh.thres,':r'),

title(['Spectral Coherence (squared magnitude) and threshold at ',num2str(100\*P),'% level of significance'])

else

title('Spectral Coherence (squared magnitude)')

end

subplot(212),plot(Coh.f(1:nfft/2+1),angle(Coh.C(1:nfft/2+1))), grid on

xlabel('[Hz]'),title('Phase')

end

% Illustrates usage of 'SCoh\_W' on a synthetic signal

% ---------------------------------------------------

% This example illustrates the detection of a cyclostationary signal deeply

% buried in stationary noise; its theoretical spectral content is in band

% [.15 .25] and its cyclic frequency is alpha = .00125.

%

% --------------------------

% Author: J. Antoni

% Last Revision: 12-2014

% --------------------------

load signal.mat % get x (data provided in the pack)

L = length(x); % signal length

Nw = 256; % window length

Nv = fix(2/3\*Nw); % block overlap

nfft = 2\*Nw; % FFT length

da = 1/L; % cyclic frequency resolution

a1 = 51; % first cyclic freq. bin to scan (i.e. cyclic freq. a1\*da)

a2 = 200; % last cyclic freq. bin to scan (i.e. cyclic freq. a2\*da)

% Loop over cyclic frequencies

C = zeros(nfft,a2-a1+1);

S = zeros(nfft,a2-a1+1);

Q = ~strcmp(which('chi2inv'),'')==1; % check if function 'chi2inv' is available

for k = a1:a2;

%if Q == 1

Coh = SCoh\_W(x,x,k/L,nfft,Nv,Nw,'sym',.01);

%else

% Coh = SCoh\_W(x,x,k/L,nfft,Nv,Nw,'sym');

end

C(:,k-a1+1) = Coh.C;

S(:,k-a1+1) = Coh.Syx;

end

% Plot results

Fs = 1; % sampling frequency in Hz

alpha = Fs\*(a1:a2)\*da;

f = Fs\*Coh.f(1:nfft/2);

figure

imagesc(alpha,f,abs(S(1:nfft/2,:))),

colormap(jet),colorbar,axis xy,title('Cyclic Spectral Density'),

xlabel('cyclic frequency \alpha [Hz]'),ylabel('spectral frequency f [Hz]')

figure

imagesc(alpha,f,abs(C(1:nfft/2,:))),

colormap(jet),colorbar,axis xy,title('Cyclic Spectral Coherence'),

xlabel('cyclic frequency \alpha [Hz]'),ylabel('spectral frequency f [Hz]')

figure

if Q == 1

plot(f,abs(C(1:nfft/2,74:76)).^2,f,Coh.thres\*ones(1,nfft/2),'m:')

title('Squared magnitude Cyclic Spectral Coherence and its 1% level of significance'),

else

plot(f,abs(C(1:nfft/2,74:76)).^2)

title('Squared magnitude Cyclic Spectral Coherence'),

end

xlabel('cyclic frequency \alpha [Hz]'),

legend(['\alpha = ',num2str((73+a1)\*da\*Fs),'Hz'],['\alpha = ',num2str((74+a1)\*da\*Fs),'Hz'],['\alpha = ',num2str((75+a1)\*da\*Fs),'Hz'])

disp(['Variance reduction factor = ',num2str(Coh.Var\_Reduc)])