

```

function [SCMS,f,ord_ax,stft,steps,t,do]=Angular_Temporal_STFT(x,fs,speed,step,win)

% speed - rotational speed [Hz]
% step - angular step for moving window [2pi]
% win - temporal window length [s]

% steps [2pi] - angular increments
% do - resulting number of samples-per-rotation [samples/2pi]

x=x(:);
N=length(x);
dt = 1/fs;

fi=2*pi*cumsum(speed)*dt;
fi=fi-fi(1); % Phase starts from zero

win=round(win*fs); % window is now in samples
do=2*pi/step;

% envelope
xre = abs(x+1i*hilbert(x));

% filtered envelope
ds = (diff(speed)/dt)./speed(1:end-1);
LU = max(ds);

xref = dig_filter_LP_JU(xre,fs,LU);

% normalization of amplitude
x = x./xref';

steps=0:step:fi(end);
steps_ind=zeros(1,length(steps));

    stft=zeros(win/2,length(steps));

    h=waitbar(0,'please wait...');

for i=1:length(steps)

    [temp,ind]=min(abs(fi-steps(i)));
    steps_ind(i)=ind;

if steps_ind(i)+win>N
    xi=x(steps_ind(i):end).*hanning(length(x(steps_ind(i):end)));
else
    xi=x(steps_ind(i):steps_ind(i)+win-1).*hanning(win);
end
    fft_xi=fft(xi,win)/(0.5*win);
    stft(:,i)=fft_xi(1:end/2);
    waitbar(i/length(steps),h);
end
close(h);

```

```
df=fs/win;
f=linspace(df/2,fs/2-df/2,win/2);
t=(steps_ind-1)/fs;

SCMS=(abs(fft(abs(stft')))); % synchronized cyclic modulation spectrum
SCMS=SCMS(1:end/2,:);
SCMS(1,:) = 0;
ord_ax=linspace(0,do/2,length(steps)/2);
```