```
function [lags, crosscorr, max_crosscorr_lag]=amp_crosscorr(eeg1,eeg2,samp_freq,low_freq,high_freq)

% amp_crosscorr filters two eeg signals between a specified frequency band,
% calculates the crosscorrelation of the amplitude envelope of the filtered signals
% and returns the crosscorrelation as an output.

% USAGE:    [lags, crosscorr, max_crosscorr_lag]=amp_crosscorr(eeg1,eeg2,samp_freq,low_freq,high_freq)

%INPUTS:
%   eeg1-vector containing local field potential from brain area 1
%   eeg2-vector containing local field potential from brain area 2
%   samp_freq-sampling frequency, in Hz, of eeg1 and eeg2
%   low_freq-low cut off, in Hz, of the band pass filter that will be applied to eeg1 and eeg2
%   high_freq-high cut off, in Hz, of the band pass filter that will be applied to eeg1 and eeg2


%OUTPUTS:
%   lags-vector contaning lags from -100 ms to +100 ms, over which the
%   crosscorrelation was done
%   crosscorr-vector with the crosscorrelation of the amplitude of eeg1 eeg2
%   after being filtered between low_freq and high_freq
%   max_crosscorr_lag-lag at which the crosscorrelation peaks. Negative
%   max_crosscorr_lag indicates that eeg1 is leading eeg2.

% check inputs
if nargin ~=5
    error('ERROR in amp_crosscorr. There must be 5 inputs. - USAGE: [lags, crosscorr, max_crosscorr_lag]= amp_crosscorr(eeg1,eeg2,samp_freq,low_freq,high_freq);')
end

if nargout ~=3
    error('ERROR in amp_crosscorr. There must be 3 outputs. - USAGE: [lags, crosscorr, max_crosscorr_lag]=amp_crosscorr(eeg1,eeg2,samp_freq,low_freq,high_freq);')
end

%check consistency of data

if length(eeg1)~= length(eeg2);
    error('ERROR in amp_crosscorr. eeg1 and eeg2 must be vectors of the same size;')
end

s=size(eeg1);
if min(s)~=1
error('ERROR in amp_crosscorr. eeg1 and eeg2 must be one-dimensional vectors')
end

s=size(eeg2);
if min(s)~=1
error('ERROR in amp_crosscorr. eeg1 and eeg2 must be one-dimensional vectors')
end

order = round(samp_freq); %determines the order of the filter used
if mod(order,2)~= 0
    order = order-1;
end

Nyquist=floor(samp_freq/2);%determines nyquist frequency

MyFilt=fir1(order,[low_freq high_freq]/Nyquist); %creates filter
filtered1 = Filter0(MyFilt,eeg1); %filters eeg1 between low_freq and high_freq
filtered2 = Filter0(MyFilt,eeg2); %filters eeg2 between low_freq and high_freq



filt_hilb1 = hilbert(filtered1); %calculates the Hilbert transform of eeg1
amp1 = abs(filt_hilb1);%calculates the instantaneous amplitude of eeg1 filtered between low_freq and high_freq
amp1=amp1-mean(amp1); %removes mean of the signal because the DC component of a signal does not change the correlation
filt_hilb2 = hilbert(filtered2);%calculates the Hilbert transform of eeg2
amp2 = abs(filt_hilb2);%calculates the instantaneous amplitude of eeg2 filtered between low_freq and high_freq
 amp2=amp2-mean(amp2);

[crosscorr,lags]=xcorr(amp1, amp2,round(samp_freq/10),'coeff'); %calculates crosscorrelations between amplitude vectors
lags=(lags./samp_freq)*1000; %converts lags to miliseconds
g=find(crosscorr==max(crosscorr));%identifies index where the crosscorrelation peaks
max_crosscorr_lag=lags(g);%identifies the lag at which the crosscorrelation peaks


figure('color',[1 1 1])
plot(lags, crosscorr,'color',[0 0 1],'linewidth',2),hold on %plots crosscorrelations
plot(lags(g),crosscorr(g),'rp','markerfacecolor',[1 0 0],'markersize',10)%plots marker at the peak of the cross correlation
plot([0 0],[1.05*max(crosscorr) 0.95*min(crosscorr)],'color',[0 0 0],'linestyle',':', 'linewidth',2) %plots dashed line at zero lag
set(gca,'xtick',[-100 -50 0 50 100])
axis tight, box off, xlim([-101 100])
xlabel('Lag (ms)','fontsize',14)
ylabel('Crosscorrelation','fontsize',14)
```