Table of Contents

Spontaneous and Task-related Activation of Neuronally Correlated Events (STANCE)

Showcases the modeling of various sources of physiological noise.

author: Dr. Jason E. Hill (post-doc fellow with CNT at TTU) demo_4D_physio updated 1 FEB 2017


```
close all;
clear all; %#ok<CLALL>
currentDir = pwd;
if strcmp(currentDir(end-2:end),'GUI')
    % GUI instance of initialization
    cd ../
    STANCEroot = pwd;
    cd(currentDir)
elseif strcmp(currentDir(end-5:end),'STANCE')
    STANCEroot = pwd;
elseif strcmp(currentDir(end-16:end),'scripts_for_demos')
    cd ../
    STANCEroot = pwd;
else
    hSTANCE = msgbox('Please select the STANCE directory');
    uiwait(hSTANCE);
    currPath = fileparts(mfilename('fullpath'));
    STANCEroot = uigetdir(currPath, 'Add STANCE filepath');
end
cd(STANCEroot)
addpath(genpath(pwd));

% Load STANCE globals ...
if ~exist('STANCE.mat','file')
    STANCE_initialize_STANCE;
    load('STANCE.mat');
else
    load('STANCE.mat');
end
% NOTE: Must add SPM version to filepath prior to usage
addpath(SPMpath);
if exist(spm('Dir'),'dir')
    display('o SPM installation found.')
else
```

```
    warning('SPM installation not found. Please add to MATLAB filepath
 or install.')
    warning('SPM8 installation: http://www.fil.ion.ucl.ac.uk/spm/
software/spm8/')
    exit
end
```

o SPM installation found.

Turn off ...

... OpenGl warning

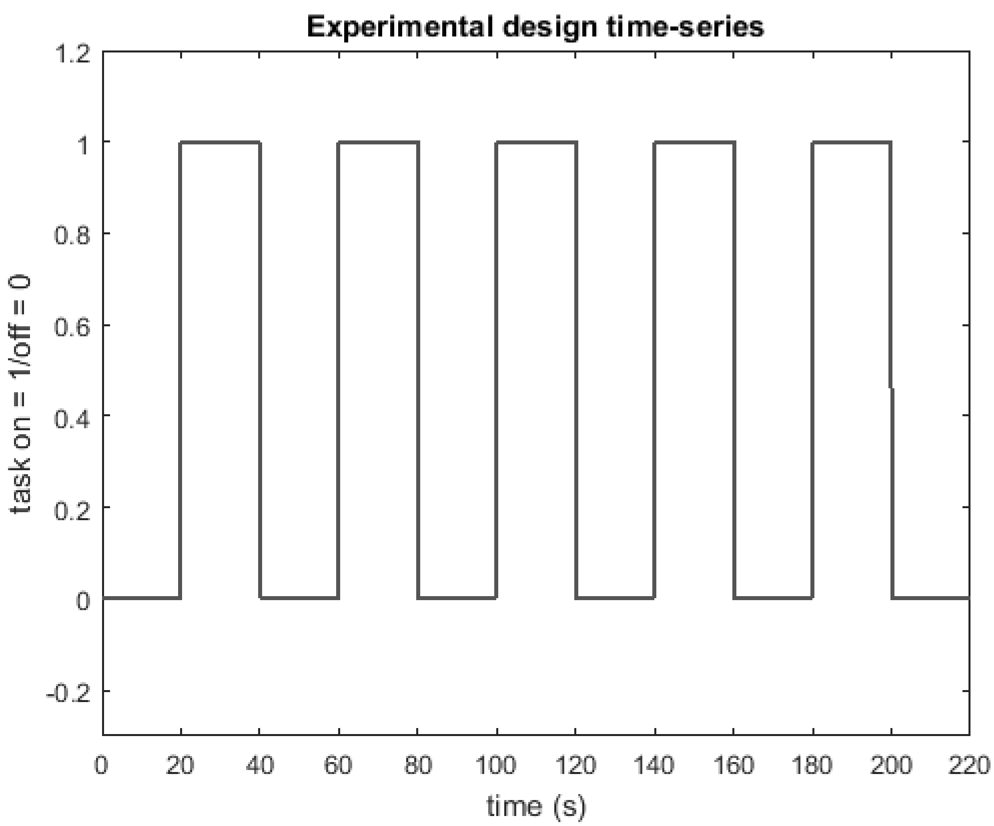
```
warning('off','MATLAB:opengl:StartupBlacklistedNoSetting');
% ... finite warning
warning('off', 'MATLAB:FINITE:obsoleteFunction');
% ... NIFTI class warning when loading SPM mat files
warning('off', 'MATLAB:unknownElementsNowStruc');
warning('off', 'MATLAB:dispatcher:ShadowedMEXExtension');
warning('off', 'MATLAB:pfileOlderThanMfile');
% ... removing files from path
warning('off', 'MATLAB:RMDIR:RemovedFromPath');
warning('off', 'MATLAB:DELETE:FileNotFound');
```

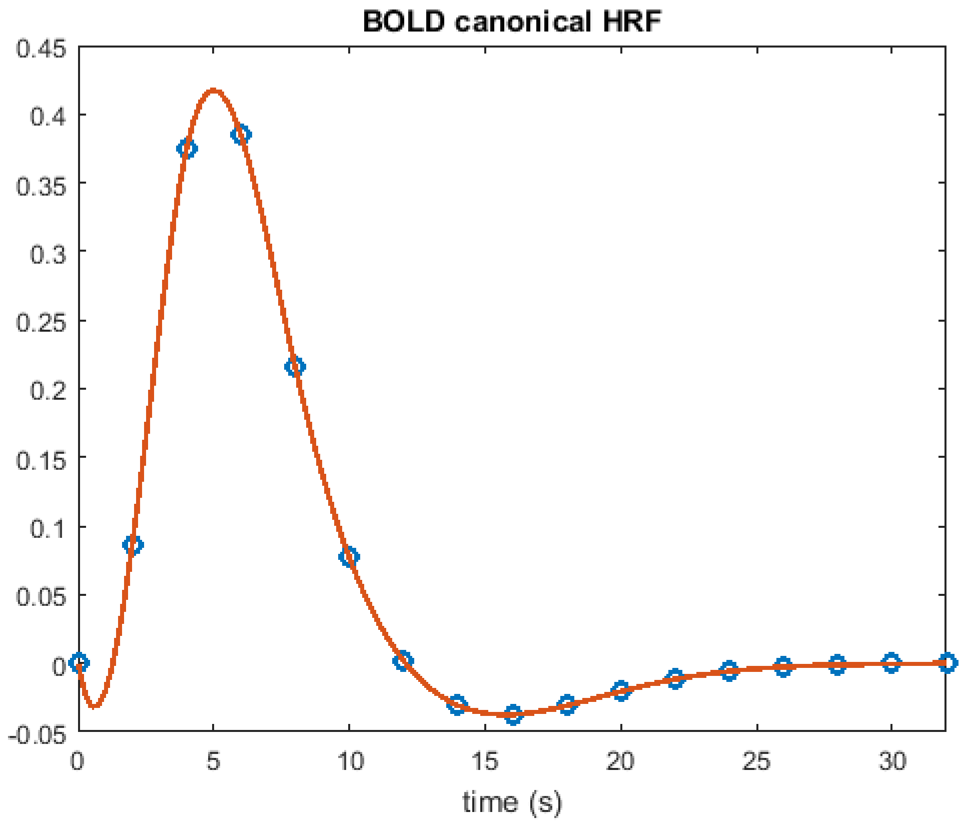
Design 4D time-series

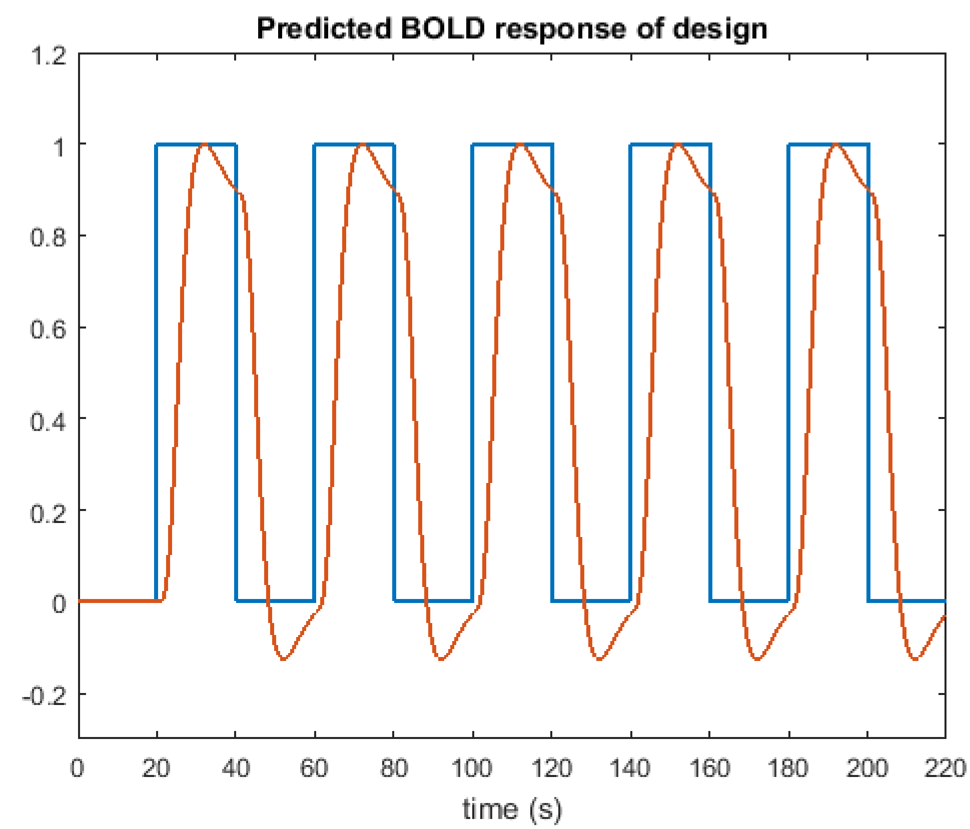
```
Nslices = 43;
TR = 2000;
TRsec = TR/1000;
dt = TRsec/Nslices;
exp_design = STANCE_blocked_design(dt, 20, 20, 20, 220);
Nt = length(exp_design.Data);
Nt_SS = ceil(60/dt);  % add 60 seconds to beginning of physiological
 time series to allow for RR and CR to reach steady state
Nt_75s = ceil(75/dt); % display first 75 seconds of signal
times = [dt*(Nt_SS:-1:0) dt*(1:Nt)]';
NT = (Nt/Nslices);

% for reproducibility
s = 5000;
%s = []; % allow MATLAB to spontaneously shuffle
if ~isempty(s)
    rng(s);
end

h_expdesign = figure;
plot(exp_design,'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([-0.3 1.2])
xlim([0 times(end)])
xlabel('time (s)')
ylabel('task on = 1/off = 0')
title('Experimental design time-series')
movegui(h_expdesign,'northwest');
```

```
% display canonical HRF
hrf = spm_hrf(TRsec);
times2 = 0:0.25:32;
hrf_exact = spline(0:TRsec:32,hrf,times2);
h_hrf = figure;
plot(0:TRsec:32,hrf,'o',times2,hrf_exact,'LineWidth',2.0);
xlim([0,32])
xlabel('time (s)')
title('BOLD canonical HRF')
movegui(h_hrf,'north');

BOLD_ts = STANCE_apply_response_function(dt, exp_design);
baseline_ts = (1-BOLD_ts.Data);
h_predictedBOLD = figure;
plot(times(Nt_SS+2:end),exp_design.Data,times(Nt_SS
+2:end),BOLD_ts.Data,'LineWidth',1.5)
ylim([-0.3 1.2])
xlim([0 times(end)])
xlabel('time (s)')
title('Predicted BOLD response of design')
movegui(h_predictedBOLD,'northeast')

Nphys = Nt+Nt_SS + 1;
```

o Applying the canonical hemodynamic response function to time-series.




BOLD canonical HRF
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
5
10
15
20
25
30
time (s)


Predicted BOLD response of design
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
20
40
60
80
100
120
140
160
180
200
220
time (s)

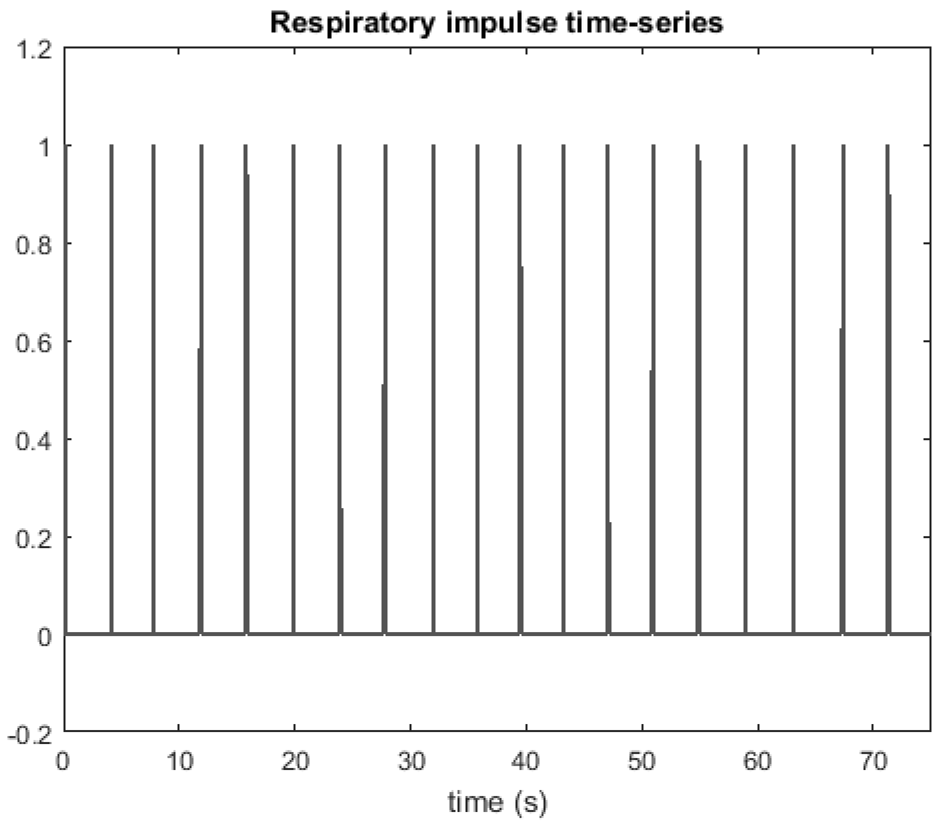
Generate sources of respiration effects on the signal

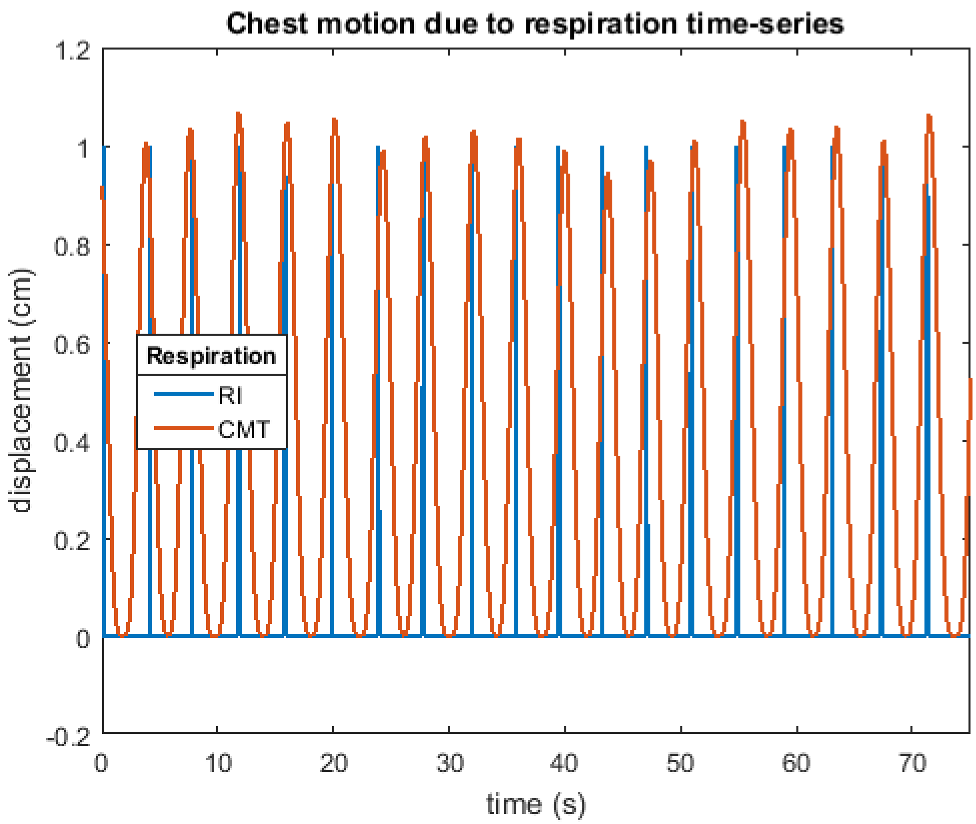
```
uiwait(msgbox('Simulate respiration and its effects on the
 signal.','Respiratory sources'));

% respiratory impulse time-series
RI_ts = make_impulse_timeseries(dt,Nphys,4.0,0.25,s);
h_RI = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+1),RI_ts.Data(Nt_SS+2:Nt_SS+Nt_75s
+1),'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
xlim([0 times(Nt_SS+Nt_75s+1)])
xlabel('time (s)')
title('Respiratory impulse time-series')
movegui(h_RI,'southwest')

% respiratory motion time-series
rng(1);
z_ts = make_pulse_timeseries(dt,Nphys,[],[],[],[],[],[],s+2*Nphys,s
+4*Nphys);
z_ts_Data = squeeze(z_ts.Data);
h_CMT = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+1),RI_ts.Data(Nt_SS+2:Nt_SS+Nt_75s
+1),times(Nt_SS+2:Nt_SS+Nt_75s+1),z_ts_Data(Nt_SS+2:Nt_SS+Nt_75s
+1),'LineWidth', 1.5)
ylim([-0.2 1.2])
xlim([0 times(Nt_SS+Nt_75s+1)])
xlabel('time (s)')
ylabel('displacement (cm)')
title('Chest motion due to respiration time-series')
lgdn_CMT = legend('RI','CMT','Location','best');
title(lgdn_CMT,'Respiration')
movegui(h_CMT,'south')

% respiratory volume time-series
[RVT,FRC,TIV] = STANCE_RVT(dt,Nphys,[],[],[],[],[],[],s+2*Nphys,s
+4*Nphys);
RVT_Data = squeeze(RVT.Data);
h_RVT = figure;
yyaxis left
plot(times(Nt_SS+2:Nt_SS+Nt_75s+1),z_ts_Data(Nt_SS+2:Nt_SS+Nt_75s
+1),times(Nt_SS+2:Nt_SS+Nt_75s+1),RVT.Data(Nt_SS+2:Nt_SS+Nt_75s
+1)/1000.0,'LineWidth', 1.5)
xlim([0 times(Nt_SS+Nt_75s+1)])
ylim([-0.1 2.9])
xlabel('time (s)')
ylabel('chest displacement (cm)')
title('Respiration time-series')
yyaxis right
plot(times(Nt_SS+2:Nt_SS+Nt_75s+1 ),RVT.Data(Nt_SS+2:Nt_SS+Nt_75s
+1 )/1000.0,'LineWidth', 1.5)
```

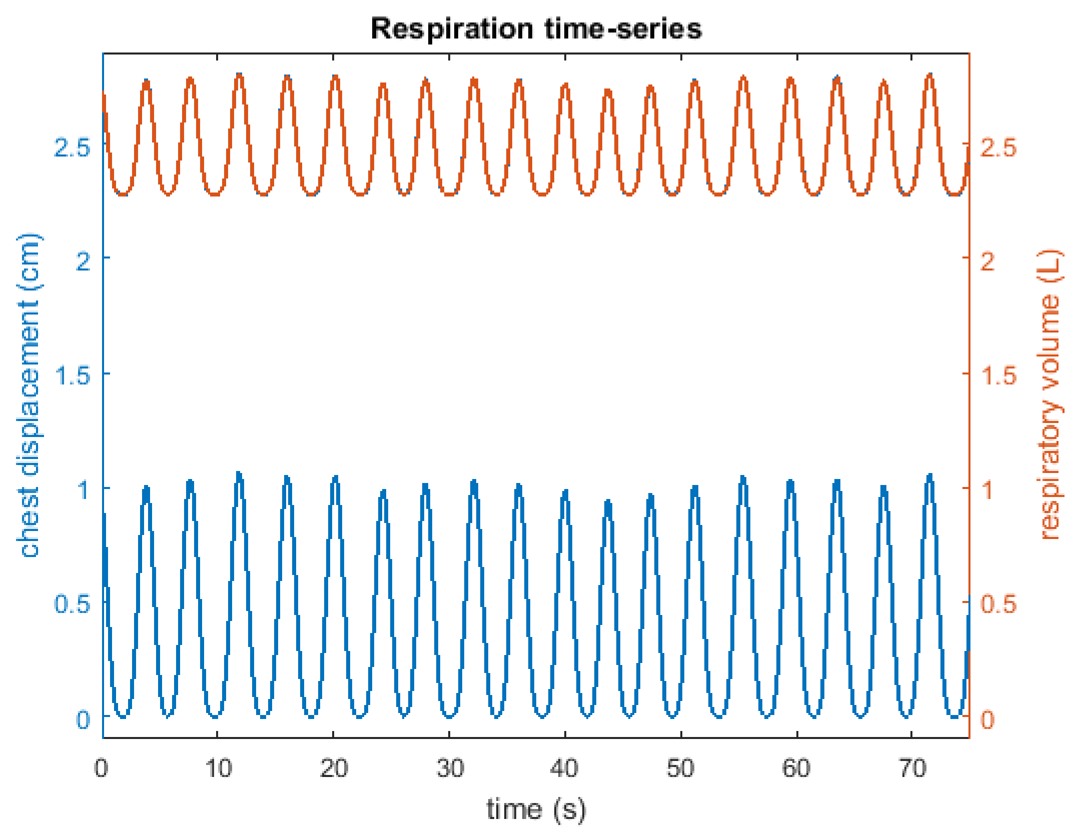
```
ylabel('respiratory volume (L)')
ylim([-0.1 2.9])
movegui(h_RVT,'southeast')

% define RVT baseline to use in modelling HRV
 display('o Constructing RVT baseline to use in modelling HRV.')
RVT_baseline_Data = (RVT_Data - (FRC+0.5*(TIV-FRC)))/(0.5*(TIV-FRC));
RVT_baseline = timeseries(squeeze(RVT_baseline_Data),times);

% define respiratory pulse RP time-series with 0 average and unit
 variance
RP_ts = RVT_baseline;
RP_ts.Data = (RVT_baseline_Data - mean(RVT_baseline_Data(Nt_SS
+2:end)))./std(RVT_baseline_Data(Nt_SS+2:end));
display('Validating that the resulting RP time-series has average = 0;
 variance = 1:')
mean(RP_ts.Data(Nt_SS+2:end))
std(RP_ts.Data(Nt_SS+2:end))
RP_ts_Data = -squeeze(RP_ts.Data); % Increased lung volume increases
 magnetic field inhomogeneity, which decreases T2*.

% define respiratory response RR time-series with 0 average and unit
 variance
RR_ts = STANCE_apply_response_function(dt, RVT_baseline,'RRF');
RR_ts.Data = (RR_ts.Data - mean(RR_ts.Data(Nt_SS+2:end)))./
std(RR_ts.Data(Nt_SS+2:end));
display('Validating that the resulting RR time-series has average = 0;
 variance = 1:')
mean(RR_ts.Data(Nt_SS+2:end))
std(RR_ts.Data(Nt_SS+2:end))
RR_ts_Data = squeeze(RR_ts.Data);

h_RR = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+1 ),RVT_baseline_Data(Nt_SS+2:Nt_SS
+Nt_75s+1 ),times(Nt_SS+2:Nt_SS+Nt_75s+1),RR_ts.Data(Nt_SS+2:Nt_SS
+Nt_75s+1),'LineWidth',1.5)
title('Predicted respiratory response')
xlabel('time (s)')
lgdn = legend('normalized RVT','Respiratory
 Response','Location','best');
title(lgdn,'Respiratory sources')
movegui(h_RR,'center')
```

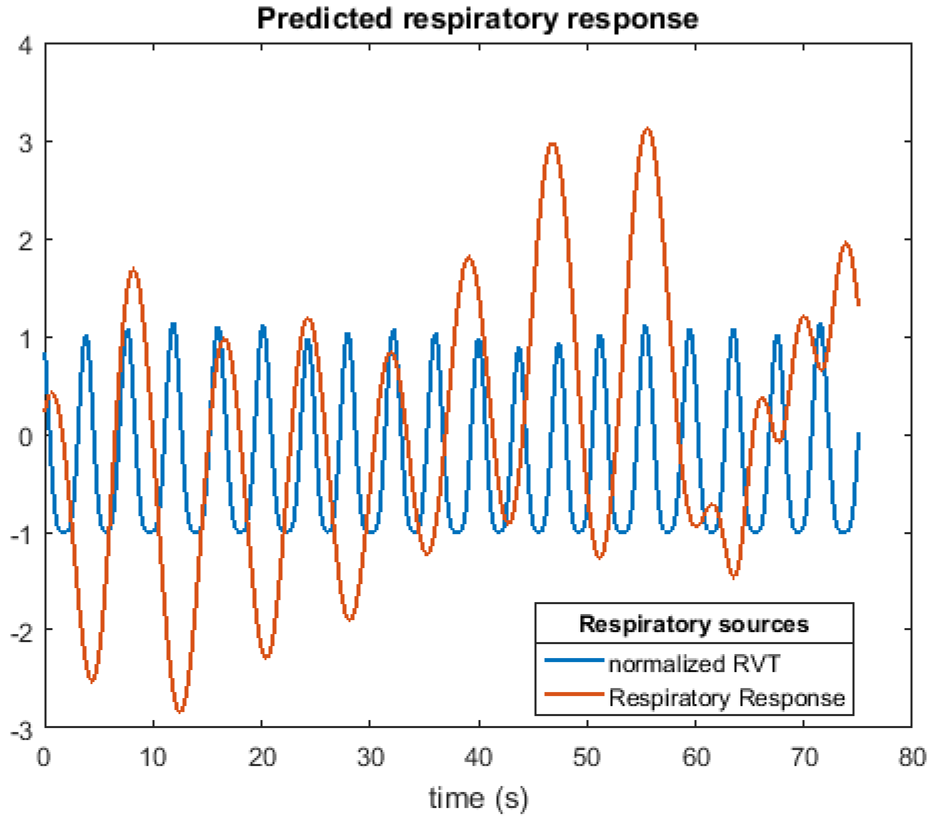
```
o Constructing RVT baseline to use in modelling HRV.
Validating that the resulting RP time-series has average = 0; variance
 = 1:

ans =

  -3.9217e-16


ans =
```

```
    1.0000

o Applying the respiratory response function to time-series.
Validating that the resulting RR time-series has average = 0; variance
 = 1:

ans =

  -1.6056e-14


ans =

    1.0000
```




Chest motion due to respiration time-series
Respiration
RI
CMT
displacement (cm)
time (s)


Respiration time-series
chest displacement (cm)
respiratory volume (L)
time (s)



Generate sources of heart beat effects on the signal

```
uiwait(msgbox('Simulate the heart beat and its effects on the
 signal.','Cardiac sources'));
% cardiac event impulse time-series
display('o Constructing cardiac event impulse time-series from an IPFM
 model.')

TI = 1.0; % second
m0 = dt./TI;
% Reproduce figure

% inverse frequency power: 1/|f|^\alpha noise; here alpha = 1 is pink
 noise.
alpha = 1;
NumChannels = 1;
hpink =
 dsp.ColoredNoise('InverseFrequencyPower',1,'NumChannels',NumChannels,...
    'SamplesPerFrame',Nphys);
rng(s+6*Nphys)
m_chaotic = step(hpink)';
m_chaotic_max = max([max(m_chaotic),-min(m_chaotic)]);
m_chaotic = (m0/4) + (m0/8)*(5/m_chaotic_max)*m_chaotic;
```

```
ms_data = zeros(1,Nphys);
ms_data = ms_data + m0 + m_chaotic;
ms = timeseries(ms_data,times);
ms.TimeInfo.Units = 'seconds';
ms = setuniformtime(ms,'StartTime',dt);
ms = setuniformtime(ms,'Interval',dt);

HR_data = ms_data*TI*60./dt;

h_HRT = figure;
plot(times(Nt_SS+2:Nt_SS+1+Nt_75s),HR_data(Nt_SS+2:Nt_SS
+1+Nt_75s),'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([50 100])
xlim([0 times(Nt_SS+2+Nt_75s)])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('Chaotic Heart Rate timeseries')
movegui(h_HRT,'northwest')

%[CEI_ts,CEI_Nk] = make_event_timeseries(ms,TI);
[CEI_ts,~] = make_event_timeseries(ms,TI);
CEI_ts_Data = squeeze(CEI_ts.Data);
h_CEIT = figure;
plot(times(Nt_SS+2:Nt_SS+2+Nt_75s),CEI_ts_Data(Nt_SS+2:Nt_SS
+2+Nt_75s),'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
xlim([0 times(Nt_SS+2+Nt_75s)])
xlabel('time (s)')
title('Cardiac impulse timeseries')
movegui(h_CEIT,'north')


% One tone harmonic modulation example in Fig. 1 in
% "Improved Heart Rate Variability Signal Analysis from the Beat
 Occurrence Times According to the IPFM Model"
% by Javier Mateo and Pablo Laguna published in
% IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 47, NO. 8, AUGUST
 2000
uiwait(msgbox('Simulate the heart beat from a one tone harmonic
 modulation.','One tone example'));

display('o Constructing cardiac event impulse time-series from an IPFM
 model.')

TI = 1.0; % 1 beat/second
m0 = dt./TI;
f1 = 0.1; %Hz
m_1tone = (m0*0.4)*cos(2*pi*f1*times);

%ms_data = zeros(1,Nt);
ms_data = m0 + m_1tone;
ms = timeseries(ms_data,times);
ms.TimeInfo.Units = 'seconds';
```

```
ms = setuniformtime(ms,'StartTime',dt);
ms = setuniformtime(ms,'Interval',dt);

HR_data = ms_data*TI*60./dt;

Nt_20s = 20/dt;

h_IHR = figure;
plot(times(Nt_SS+2:Nt_SS+2+Nt_20s),HR_data(Nt_SS+2:Nt_SS
+2+Nt_20s),'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([30 90])
xlim([0 times(Nt_SS+2+Nt_20s)])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('Instantaneous Heart Rate timeseries [beats/min]')
movegui(h_IHR,'west')

%[CEI_ts,CEI_Nk] = make_event_timeseries(ms,TI);
[CEI_ts,~] = make_event_timeseries(ms,TI);
CEI_ts_Data = squeeze(CEI_ts.Data);
h_CIT = figure;
plot(times(Nt_SS+2:Nt_SS+2+Nt_20s),CEI_ts_Data(Nt_SS+2:Nt_SS
+2+Nt_20s),'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
xlim([0 times(Nt_SS+2+Nt_20s)])
xlabel('time (s)')
title('Cardiac impulse timeseries')
movegui(h_CIT,'center')

%figure, periodogram(HR_data)
fs = 1/dt; % sampling frequency [Hz]
L_data = length(HR_data);
freqs = (fs/2)*(-1:2/L_data:1-2/L_data);
Rhalf = ceil(L_data/2)+1:L_data;
HR_Data_dB = 20*log(abs(fftshift(fft(HR_data)))/sqrt(L_data*fs));
h_estPSD = figure;
plot(freqs((Rhalf)),HR_Data_dB(Rhalf))
xlabel('Frequency (Hz)')
ylabel('Power Density (dB/Hz)')
title('Estimated Power Spectral Density')
axis([0 fs/16 min(HR_Data_dB) max(HR_Data_dB(Rhalf))])
movegui(h_estPSD,'east')


% Two tone harmonic modulation example in section V.A. in
% "Improved Heart Rate Variability Signal Analysis from the Beat
 Occurrence Times According to the IPFM Model"
% by Javier Mateo and Pablo Laguna published in
% IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 47, NO. 8, AUGUST
 2000
uiwait(msgbox('Simulate the heart beat from a two tone harmonic
 modulation.','Two tone example'));
```

```
display('o Constructing cardiac event impulse time-series from an IPFM
 model.')

TI = 1.0; % 1 beat/second
m0 = dt./TI;
f1 = 0.1; %Hz
f2 = 0.251; %Hz
m_2tone = m0*(0.1*cos(2*pi*f1*times)+0.1*cos(2*pi*f2*times));

%ms_data = zeros(1,Nt);
ms_data = m0 + m_2tone;
ms = timeseries(ms_data,times);
ms.TimeInfo.Units = 'seconds';
ms = setuniformtime(ms,'StartTime',dt);
ms = setuniformtime(ms,'Interval',dt);

HR_data = ms_data*TI*60./dt;

h_IHR2 = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),HR_data(Nt_SS+2:Nt_SS+Nt_75s
+2),'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([30 90])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('Instantaneous Heart Rate time-series [beats/min]')
movegui(h_IHR2,'southwest');

%[CEI_ts,CEI_Nk] = make_event_timeseries(ms,TI);
[CEI_ts,~] = make_event_timeseries(ms,TI);
CEI_ts_Data = squeeze(CEI_ts.Data);
h_CIT2 = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),CEI_ts_Data(Nt_SS+2:Nt_SS+Nt_75s
+2),'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
title('Cardiac impulse time-series')
movegui(h_CIT2,'south');

%figure, periodogram(HR_data)
L_data = length(HR_data);
freqs = (fs/2)*(-1:2/L_data:1-2/L_data);
Rhalf = ceil(L_data/2)+1:L_data;
HR_Data_dB = 20*log(abs(fftshift(fft(HR_data)))/sqrt(L_data*fs));
h_estPSD2 = figure;
plot(freqs((Rhalf)),HR_Data_dB(Rhalf))
xlabel('Frequency (Hz)')
ylabel('Power Density (dB/Hz)')
title('Estimated Power Spectral Density')
axis([0 fs/16 min(HR_Data_dB) max(HR_Data_dB(Rhalf))])
movegui(h_estPSD2,'southeast')
```

```
% Three tone harmonic modulation example in in the well-known HRV
 spectral regions of very low
% frequency (VLF), low frequency (LF) and high frequency (HF) in
 section 4.1 of
% "Spectral analysis of heart rate variability using the integral
 pulse frequency modulation model"
% by I. P. Mitov, published in Med. Biol. Eng. Comput., 2001, 39, 1-7.
uiwait(msgbox('Simulate the heart beat from a three tone harmonic
 modulation.','Three tone example'));

display('o Constructing cardiac event impulse time-series from an IPFM
 model.')

TI = 1.05; % beat/second
m0 = dt./TI;
f1 = 0.02; %Hz
f2 = 0.09; %Hz
f3 = 0.21; %Hz
m = 0.12;
m_3tone =
 m0*m*(cos(2*pi*f1*times)+cos(2*pi*f2*times)+(2/3)*cos(2*pi*f3*times));

%ms_data = zeros(1,Nphys);
ms_data = m0 + m_3tone;
ms = timeseries(ms_data,times);
ms.TimeInfo.Units = 'seconds';
ms = setuniformtime(ms,'StartTime',dt);
ms = setuniformtime(ms,'Interval',dt);

HR_data = ms_data*TI*60./dt;

h_IHR3 = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),HR_data(Nt_SS+2:Nt_SS+Nt_75s
+2),'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([30 90])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('Instantaneous Heart Rate timeseries [beats/min]')
movegui(h_IHR3,'northwest');

[CEI_ts,~] = make_event_timeseries(ms,TI);
CEI_ts_Data = squeeze(CEI_ts.Data);
h_CIT3 = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),CEI_ts_Data(Nt_SS+2:Nt_SS+Nt_75s
+2),'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
title('Cardiac impulse timeseries')
movegui(h_CIT3,'north');

h_IHR_CIT3 = figure;
```

```
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),40*CEI_ts_Data(Nt_SS+2:Nt_SS
+Nt_75s+2),times(Nt_SS+2:Nt_SS+Nt_75s+2),HR_data(Nt_SS+2:Nt_SS+Nt_75s
+2),'LineWidth', 1.25)
ylim([30 90])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('HR and its corresponding cardiac impulse [beats/min]')
lgdn2 = legend('HB','HR','Location','best');
title(lgdn2,'Cardiac sources')
movegui(h_IHR_CIT3,'south');

%figure, periodogram(HR_data)
L_data = length(HR_data);
freqs = (fs/2)*(-1:2/L_data:1-2/L_data);
Rhalf = ceil(L_data/2)+1:L_data;
HR_Data_dB = 20*log(abs(fftshift(fft(HR_data)))/sqrt(L_data*fs));
h_estPSD3 = figure;
plot(freqs((Rhalf)),HR_Data_dB(Rhalf))
xlabel('Frequency (Hz)')
ylabel('Power Density (dB/Hz)')
title('Estimated Power Spectral Density')
axis([0 fs/16 min(HR_Data_dB) max(HR_Data_dB(Rhalf))])
movegui(h_estPSD3,'northeast')

uiwait(msgbox('Model heart-rate variability with realistic
 quasi-periodic functions of breathing and blood pressure
 variations.','Realistic model'));
% model heart-rate variability with realistic quasi-periodic functions
% from physiological sources of breathing and blood pressure
 variations
display('o Constructing cardiac event impulse time-series from a IPFM
 model.')

TI = 1.05; % beat/second
m0 = dt./TI;

f1 = 0.02; %Hz
VLF = make_pulse_timeseries(dt,Nphys,1/f1,10,2,[],[],[],s+6*Nphys,s
+8*Nphys);
VLF_Data = squeeze(VLF.Data);
VLF_Data = 2.0*(VLF_Data - 0.5);

f2 = 0.1; %Hz
Mayer_wave = make_pulse_timeseries(dt,Nphys+Nt_SS,1/f2,2,2,[],[],[],s
+10*Nphys,s+12*Nphys);
Mayer_wave_Data = squeeze(Mayer_wave.Data);
Mayer_wave_Data = 2.0*(Mayer_wave_Data(1:Nphys) - 0.5);

%f3 = 0.21; %Hz
m = 0.12;
m_phys = m0*m*(VLF_Data+Mayer_wave_Data+(2/3)*RVT_baseline_Data);

%ms_data = zeros(1,Nphys);
```

```
ms_data = m0 + m_phys;
ms = timeseries(ms_data,times);
ms.TimeInfo.Units = 'seconds';
ms = setuniformtime(ms,'StartTime',dt);
ms = setuniformtime(ms,'Interval',dt);

HR_data = ms_data*TI*60./dt;

h_IHR4 = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),HR_data(Nt_SS+2:Nt_SS+Nt_75s
+2),'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([30 90])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('Instantaneous Heart Rate timeseries [beats/min]')
movegui(h_IHR4,'west')

[CEI_ts,CEI_Nk] = make_event_timeseries(ms,TI);
CEI_ts_Data = squeeze(CEI_ts.Data);
h_CIT4 = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),CEI_ts_Data(Nt_SS+2:Nt_SS+Nt_75s
+2),'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
title('Cardiac impulse timeseries')
movegui(h_CIT4,'center')

h_IHR_CIT4 = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_75s+2),40*CEI_ts_Data(Nt_SS+2:Nt_SS
+Nt_75s+2),times(Nt_SS+2:Nt_SS+Nt_75s+2),HR_data(Nt_SS+2:Nt_SS+Nt_75s
+2),'LineWidth', 1.25)
ylim([30 90])
xlim([0 times(Nt_SS+Nt_75s+2)])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('HR and its corresponding cardiac impulse [beats/min]')
lgdn3 = legend('HB','HR','Location','best');
title(lgdn3,'Cardiac sources')
movegui(h_IHR_CIT4,'south');

%figure, periodogram(HR_data)
L_data = length(HR_data);
freqs = (fs/2)*(-1:2/L_data:1-2/L_data);
Rhalf = ceil(L_data/2)+1:L_data;
HR_Data_dB = 20*log(abs(fftshift(fft(HR_data)))/sqrt(L_data*fs));
h_PSD4 = figure;
plot(freqs((Rhalf)),HR_Data_dB(Rhalf))
xlabel('Frequency (Hz)')
ylabel('Power Density (dB/Hz)')
title('Estimated Power Spectral Density')
axis([0 fs/16 min(HR_Data_dB) HR_Data_dB(Rhalf(1))])
movegui(h_PSD4,'east');
```

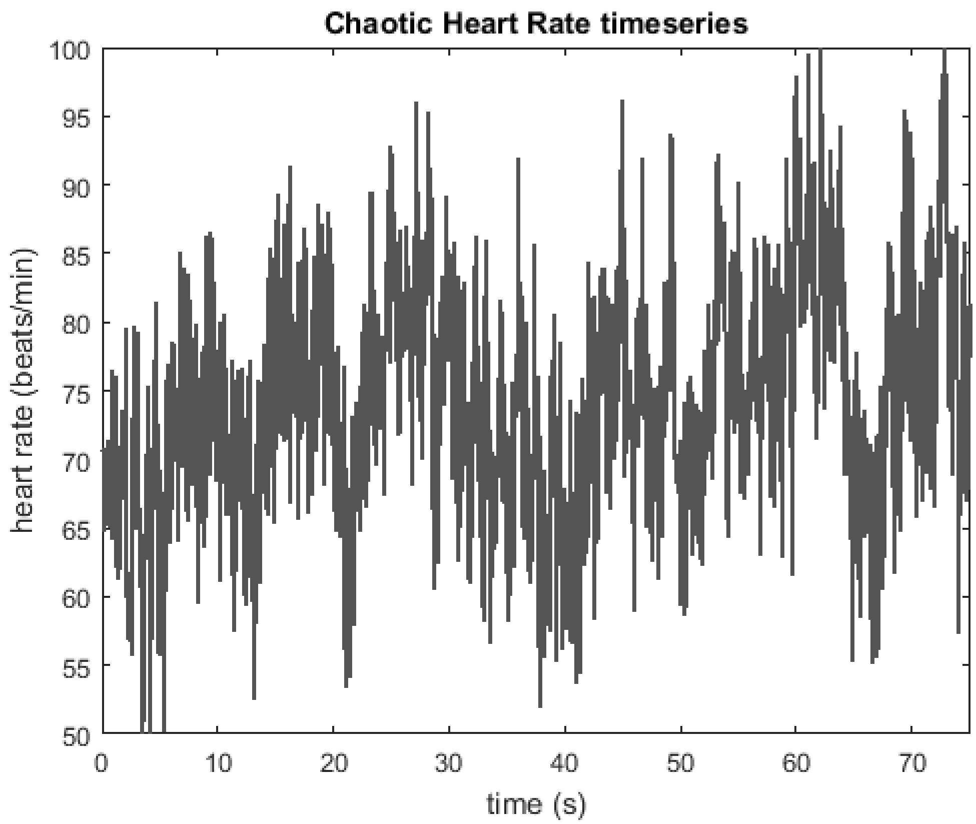
```
% generate realistic cardiac signals:
uiwait(msgbox('Generate realistic cardiac signals.','Realistic
 cardiac'));

display('o Constructing the HR time-series.')
HR_ts = timeseries(squeeze(HR_data),times);

display('o Constructing the CR time-series.')
CR_ts = STANCE_apply_response_function(dt, HR_ts, 'CRF');
CR_ts.Data = (CR_ts.Data - mean(CR_ts.Data(Nt_SS+2:end)))./
std(CR_ts.Data(Nt_SS+2:end));
display('Validating that the resulting CR time-series has average = 0;
 variance = 1:')
mean(CR_ts.Data(Nt_SS+2:end))
std(CR_ts.Data(Nt_SS+2:end))
CR_ts_Data = squeeze(CR_ts.Data);

h_predictedCR = figure;
yyaxis left
plot(times(Nt_SS+2:Nt_SS+Nt_75s+1 ),HR_data(Nt_SS+2:Nt_SS+Nt_75s
+1 ),'LineWidth',1.5)
xlim([0 times(Nt_SS+Nt_75s+1 )])
xlabel('time (s)')
ylabel('heart rate (beats/min)')
title('Predicted cardiac response')
yyaxis right
plot(times(Nt_SS+2:Nt_SS+Nt_75s+1 ),CR_ts.Data(Nt_SS+2:Nt_SS+Nt_75s
+1 ),'LineWidth',1.5)
ylabel('normalized cardiac response')
movegui(h_predictedCR,'southwest');

% pulse wave velocity
display('o Constructing the PWV time-series.')
v_ts = STANCE_PWV_timeseries(CEI_ts,[],[],[],[],s+14*Nphys);
v_ts_Data = squeeze(v_ts.Data);
h_PWV = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_20s+1),10*CEI_ts_Data(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),v_ts_Data(Nt_SS+2:Nt_SS+Nt_20s
+1),'LineWidth', 1.5)
ylim([-0.2 25])
xlim([0 times(Nt_SS+Nt_20s+1)])
xlabel('time (s)')
ylabel('velocity (mm/s)')
title('Pulse wave velocity time-series')
lgdn_PWV = legend('HB','PWV','Location','best');
title(lgdn_PWV,'Cardiac pulse')
movegui(h_PWV,'southeast');

% Cardiac pulse
display('o Constructing the CP time-series.')
CP_ts = v_ts;
```

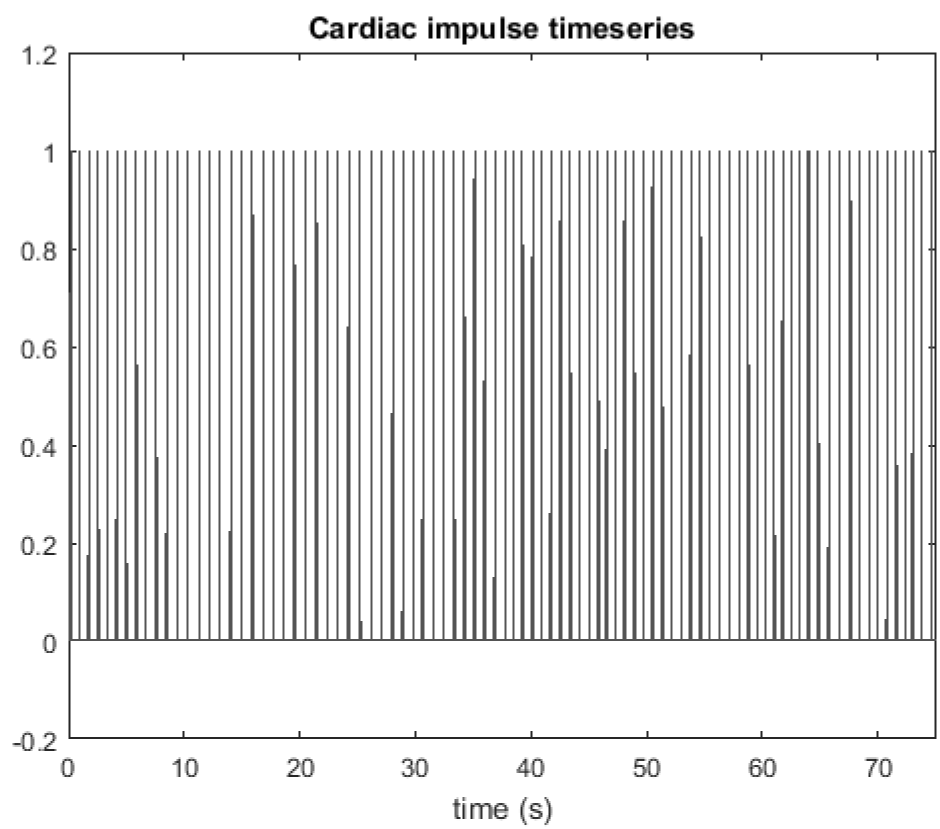
```
CP_ts.Data = (CP_ts.Data - mean(CP_ts.Data(Nt_SS+2:end)))./
std(CP_ts.Data(Nt_SS+2:end));
display('Validating that the resulting CP time-series has average = 0;
 variance = 1:')
mean(CP_ts.Data(Nt_SS+2:end))
std(CP_ts.Data(Nt_SS+2:end))
CP_ts_Data = -squeeze(CP_ts.Data);  % cardiac pulse causes the signal
 S_0 to DECREASE, but T2* to INCREASE via the CRF (HgO_2 pulsations)

Rsquared                = [[0.046, 0.022, 0.042, 0.013, 0.00325,
 0.022];... % CSF Row: [RP RR CP CR BP InterCRP]
                           [0.04,   0.021, 0.03,  0.012, 0.0025,
 0.017];... % GM Row
                           [0.042,  0.020, 0.032, 0.011, 0.00275,
 0.020];... % WM Row
                           [0.037,  0.017, 0.079, 0.009, 0.00225,
 0.040];... % BS Row
                           [0.033,  0.02,  0.25,  0.01,  0.0025,
 0.02]];    % BV Row
CSF_Rsquared = Rsquared(1,:);
GM_Rsquared  = Rsquared(2,:);
WM_Rsquared  = Rsquared(3,:);
BS_Rsquared  = Rsquared(4,:);
BV_Rsquared  = Rsquared(5,:);
%ave_Rsquared = mean(Rsquared);


% The interaction of the CP and RP
display('o Constructing the InterCRP time-series.')
% Define the interaction of the cardiac and respiratory pulses
 (InterCRP) time-series
display('o Generating the interaction of the cardiac and respiratory
 pulses (InterCRP) time-series.')
% model as a product of a weighted average of respiratory sources with
 cardiac sources
% ... in gray matter
InterCRP_GM_ts = RVT;
InterCRP_GM_ts.Data = (GM_Rsquared(1)*RP_ts_Data
+GM_Rsquared(2)*RR_ts_Data).*(GM_Rsquared(3)*CP_ts_Data +
 GM_Rsquared(4).*CR_ts_Data + GM_Rsquared(5)*Mayer_wave_Data);
InterCRP_GM_mean = mean(InterCRP_GM_ts.Data(Nt_SS+2:end));
InterCRP_GM_std  = std(InterCRP_GM_ts.Data(Nt_SS+2:end));
InterCRP_GM_ts.Data = (InterCRP_GM_ts.Data - InterCRP_GM_mean)./
InterCRP_GM_std;
InterCRP_GM_ts_Data = squeeze(InterCRP_GM_ts.Data);
% ... in white matter
InterCRP_WM_ts = RVT;
InterCRP_WM_ts.Data = (WM_Rsquared(1)*RP_ts_Data
+WM_Rsquared(2)*RR_ts_Data).*(WM_Rsquared(3)*CP_ts_Data +
 WM_Rsquared(4).*CR_ts_Data + WM_Rsquared(5)*Mayer_wave_Data);
InterCRP_WM_mean    = mean(InterCRP_WM_ts.Data(Nt_SS+2:end));
InterCRP_WM_std      = std(InterCRP_WM_ts.Data(Nt_SS+2:end));
InterCRP_WM_ts.Data = (InterCRP_WM_ts.Data - InterCRP_WM_mean)./
InterCRP_WM_std;
```

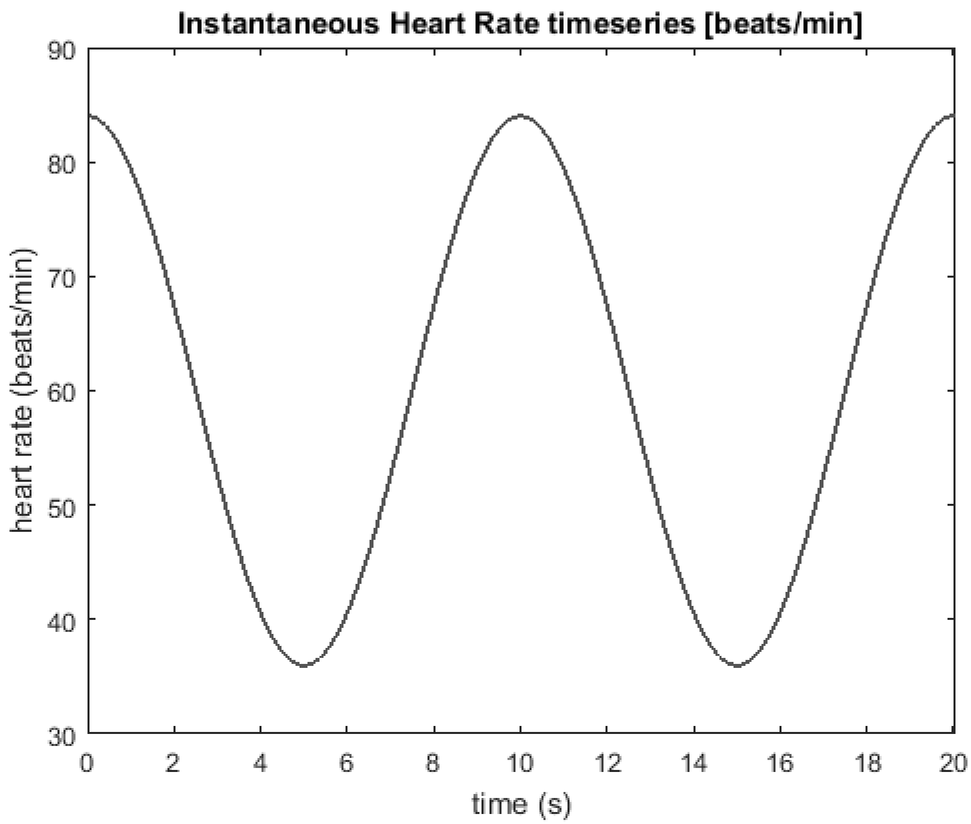
```
InterCRP_WM_ts_Data = squeeze(InterCRP_WM_ts.Data);
% ... in cerebrospinal fluid
InterCRP_CSF_ts = RVT;
InterCRP_CSF_ts.Data = (CSF_Rsquared(1)*RP_ts_Data
+CSF_Rsquared(2)*RR_ts_Data).*(CSF_Rsquared(3)*CP_ts_Data +
 CSF_Rsquared(4).*CR_ts_Data + CSF_Rsquared(5)*Mayer_wave_Data);
InterCRP_CSF_mean    = mean(InterCRP_CSF_ts.Data(Nt_SS+2:end));
InterCRP_CSF_std     = std(InterCRP_CSF_ts.Data(Nt_SS+2:end));
InterCRP_CSF_ts.Data = (InterCRP_CSF_ts.Data - InterCRP_CSF_mean)./
InterCRP_CSF_std;
InterCRP_CSF_ts_Data = squeeze(InterCRP_CSF_ts.Data);
% ... in blood vessels
InterCRP_BV_ts = RVT;
InterCRP_BV_ts.Data = (BV_Rsquared(1)*RP_ts_Data
+BV_Rsquared(2)*RR_ts_Data).*(BV_Rsquared(3)*CP_ts_Data +
 BV_Rsquared(4).*CR_ts_Data + BV_Rsquared(5)*Mayer_wave_Data);
InterCRP_BV_mean    = mean(InterCRP_BV_ts.Data(Nt_SS+2:end));
InterCRP_BV_std     = std(InterCRP_BV_ts.Data(Nt_SS+2:end));
InterCRP_BV_ts.Data = (InterCRP_BV_ts.Data - InterCRP_BV_mean)./
InterCRP_BV_std;
InterCRP_BV_ts_Data = squeeze(InterCRP_BV_ts.Data);
% ... in brain stem
InterCRP_BS_ts = RVT;
InterCRP_BS_ts.Data = (BS_Rsquared(1)*RP_ts_Data
+BS_Rsquared(2)*RR_ts_Data).*(BS_Rsquared(3)*CP_ts_Data +
 BS_Rsquared(4).*CR_ts_Data + BS_Rsquared(5)*Mayer_wave_Data);
InterCRP_BS_mean    = mean(InterCRP_BS_ts.Data(Nt_SS+2:end));
InterCRP_BS_std     = std(InterCRP_BS_ts.Data(Nt_SS+2:end));
InterCRP_BS_ts.Data = (InterCRP_BS_ts.Data - InterCRP_BS_mean)./
InterCRP_BS_std;
InterCRP_BS_ts_Data = squeeze(InterCRP_BS_ts.Data);

% show all the main sources of physiological noise
uiwait(msgbox('Show all the main sources of physiological effects on
 the signal.','Show all sources'));
h_allsources = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_20s+1),CR_ts_Data(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),RR_ts_Data(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),Mayer_wave_Data(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),InterCRP_GM_ts_Data(Nt_SS+2:Nt_SS
+Nt_20s+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),RP_ts_Data(Nt_SS+2:Nt_SS
+Nt_20s+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),CP_ts_Data(Nt_SS+2:Nt_SS
+Nt_20s+1),'LineWidth', 1.5)
ylim([-4.5 4.5])
xlim([0 times(Nt_SS+Nt_20s+1)])
xlabel('time (s)')
title('Normalized physiological noise by all sources')
lgdn4 = legend('CR','RR','BP','InterCRP','RP','CP','Location','best');
title(lgdn4,'Noise source')
movegui(h_allsources,'northwest');

% show sources of physiological noise over coarse of experiment
h_allsources = figure;
```

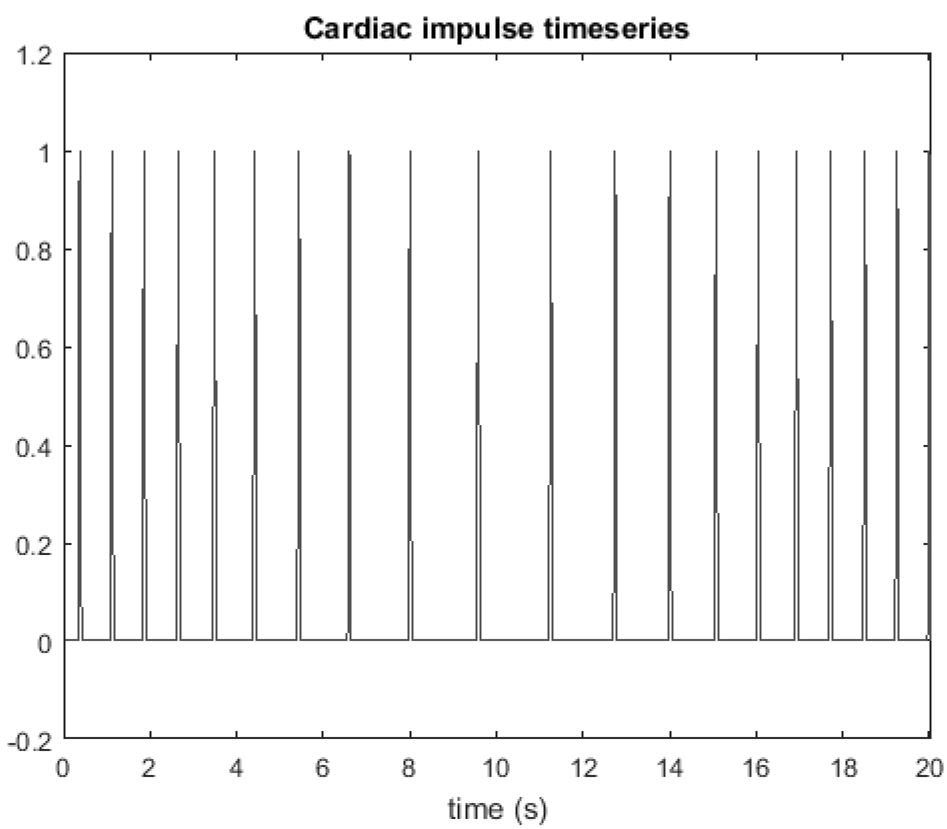
```
plot(times(Nt_SS+2:end),CR_ts_Data(Nt_SS+2:end),times(Nt_SS
+2:end),RR_ts_Data(Nt_SS+2:end),times(Nt_SS
+2:end),Mayer_wave_Data(Nt_SS+2:end),'LineWidth', 1.5)
ylim([-4.5 4.5])
xlim([0 times(end)])
xlabel('time (s)')
title('Normalized physiological noise by LF sources')
lgdn5 = legend('CR','RR','BP','Location','best');
title(lgdn5,'Noise source')
movegui(h_allsources,'northeast');
```

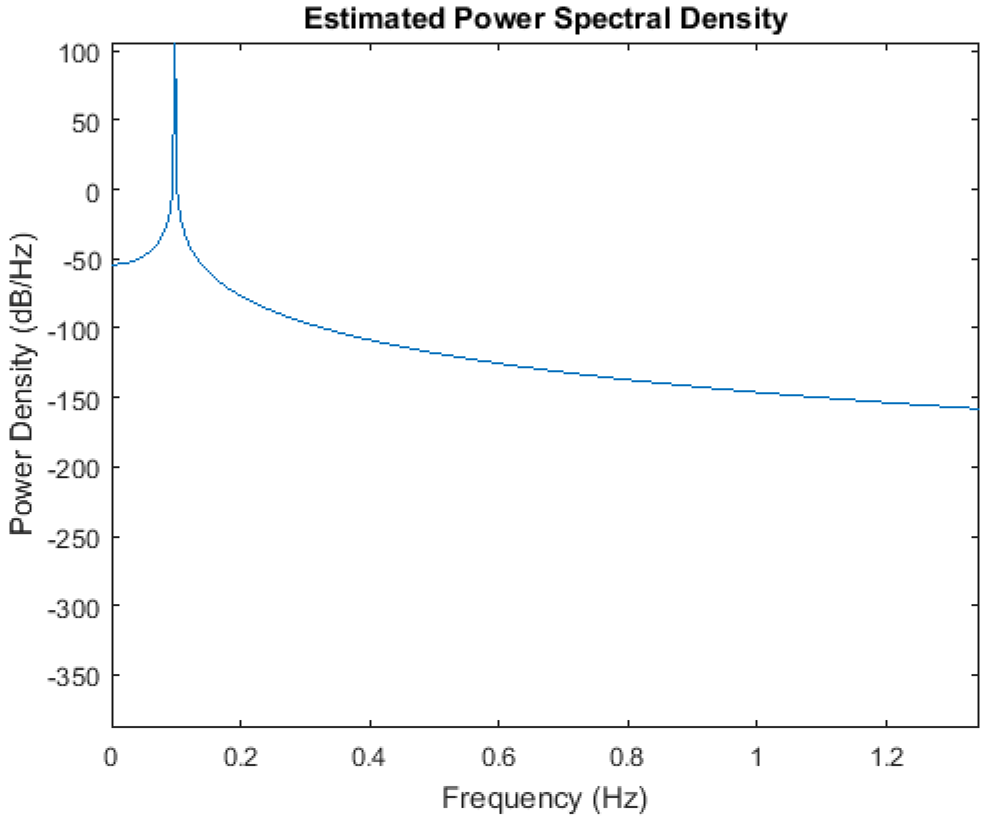
```
o Constructing cardiac event impulse time-series from an IPFM model.
o Constructing cardiac event impulse time-series from an IPFM model.
o Constructing cardiac event impulse time-series from an IPFM model.
o Constructing cardiac event impulse time-series from an IPFM model.
o Constructing cardiac event impulse time-series from a IPFM model.
o Constructing the HR time-series.
o Constructing the CR time-series.
o Applying the cardiac response function to time-series.
Validating that the resulting CR time-series has average = 0; variance
 = 1:

ans =

   4.3796e-16


ans =

    1.0000

o Constructing the PWV time-series.
o Constructing the CP time-series.
Validating that the resulting CP time-series has average = 0; variance
 = 1:

ans =

   3.1675e-14


ans =

    1.0000

o Constructing the InterCRP time-series.
o Generating the interaction of the cardiac and respiratory pulses
 (InterCRP) time-series.
```

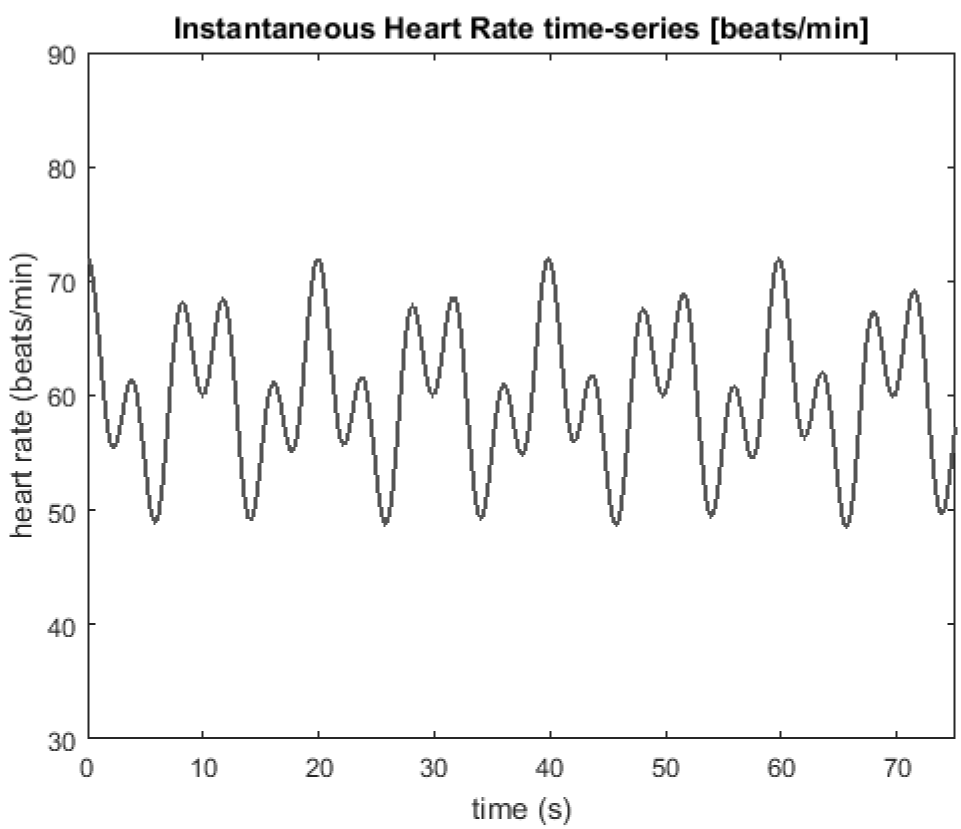

Chaotic Heart Rate timeseries
heart rate (beats/min)
100
95
90
85
80
75
70
65
60
55
50
0
10
20
30
40
50
60
70
time (s)

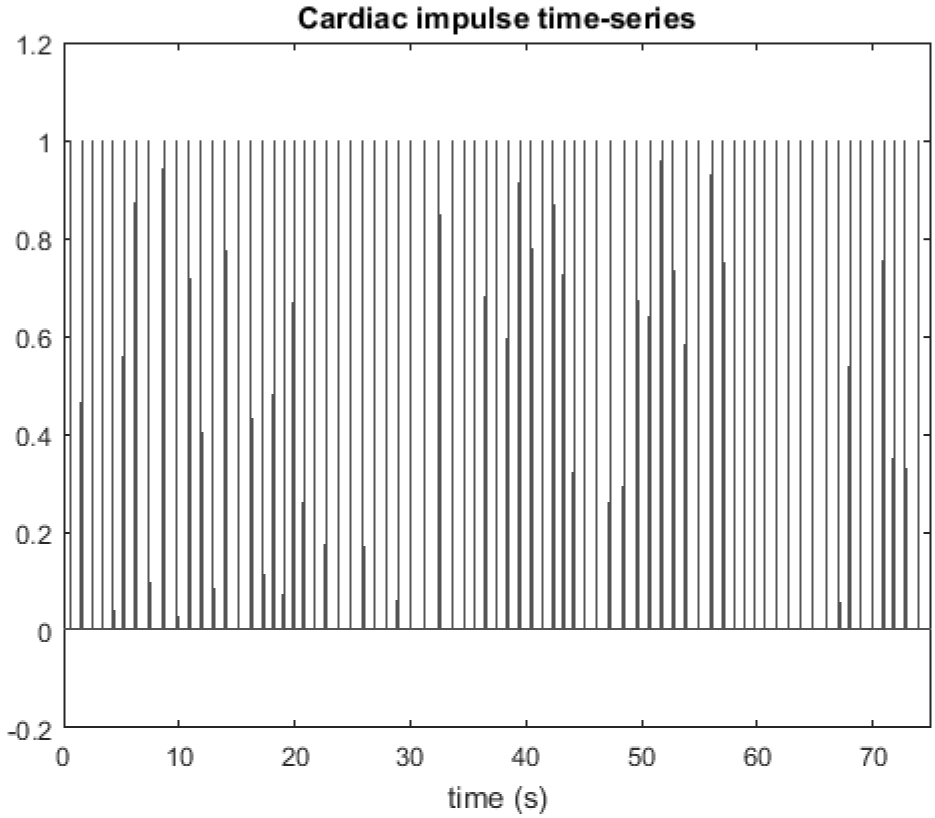

Cardiac impulse timeseries
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
10
20
30
40
50
60
70
time (s)

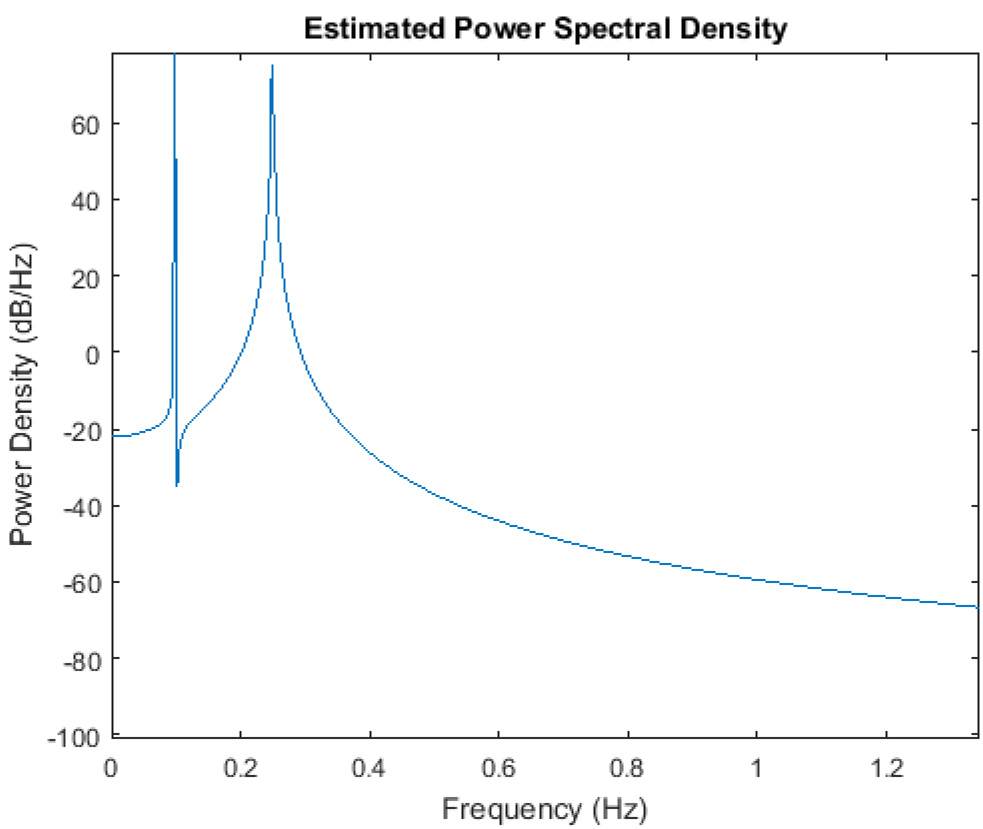

Instantaneous Heart Rate timeseries [beats/min]
heart rate (beats/min)
time (s)
90
80
70
60
50
40
30
0
2
4
6
8
10
12
14
16
18
20


Cardiac impulse timeseries
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
2
4
6
8
10
12
14
16
18
20
time (s)

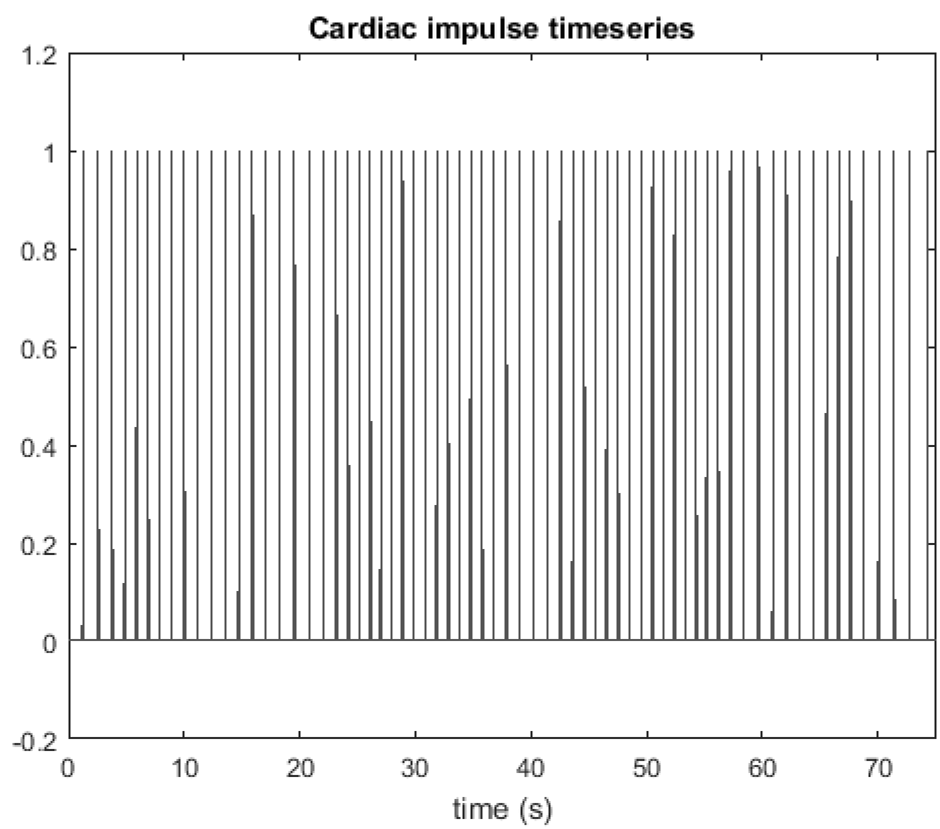

Estimated Power Spectral Density
Power Density (dB/Hz)
Frequency (Hz)
100
50
0
-50
-100
-150
-200
-250
-300
-350
0
0.2
0.4
0.6
0.8
1
1.2

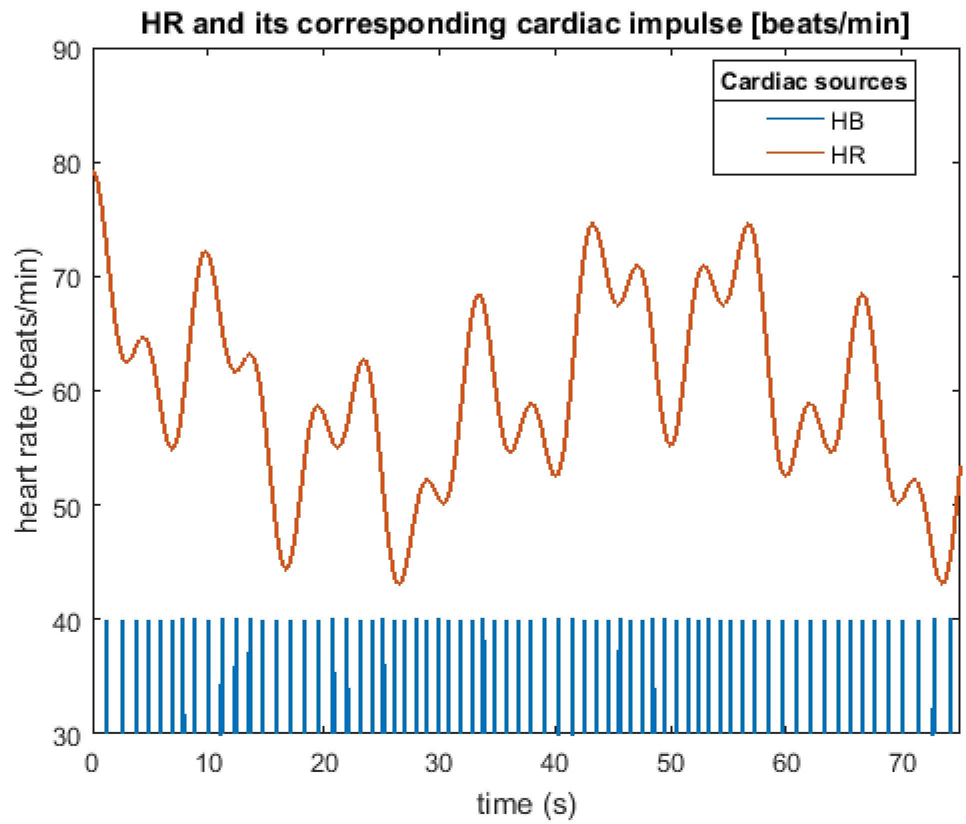

Instantaneous Heart Rate time-series [beats/min]
heart rate (beats/min)
time (s)
90
80
70
60
50
40
30
0
10
20
30
40
50
60
70

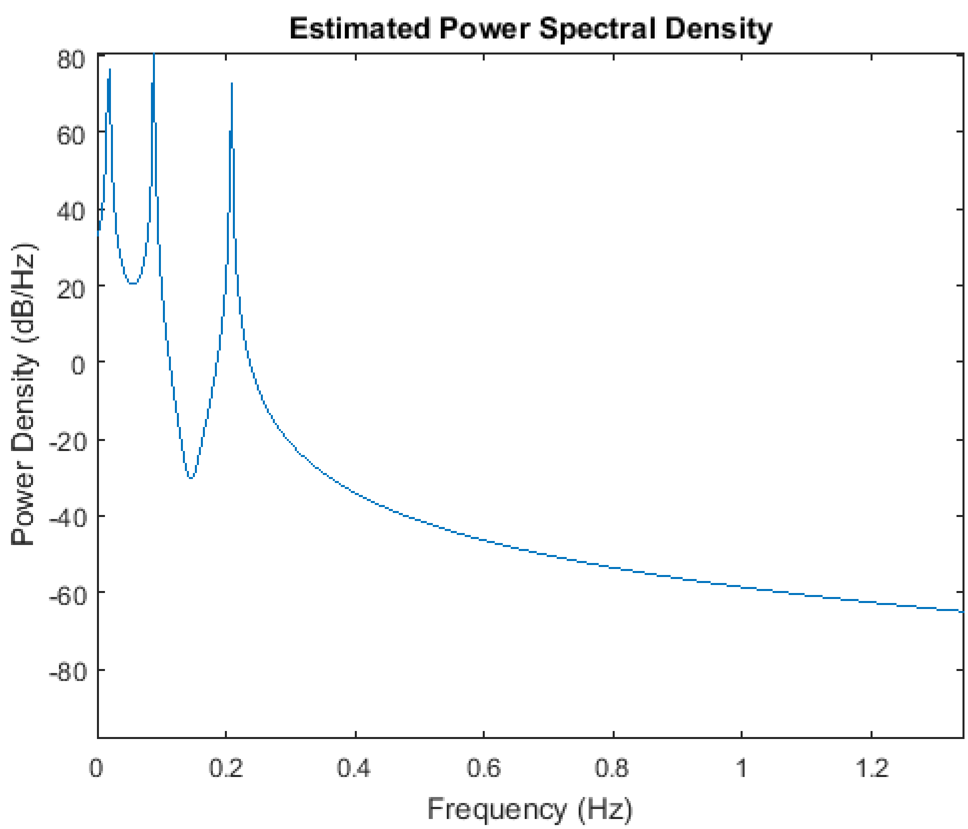

Cardiac impulse time-series
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
10
20
30
40
50
60
70
time (s)

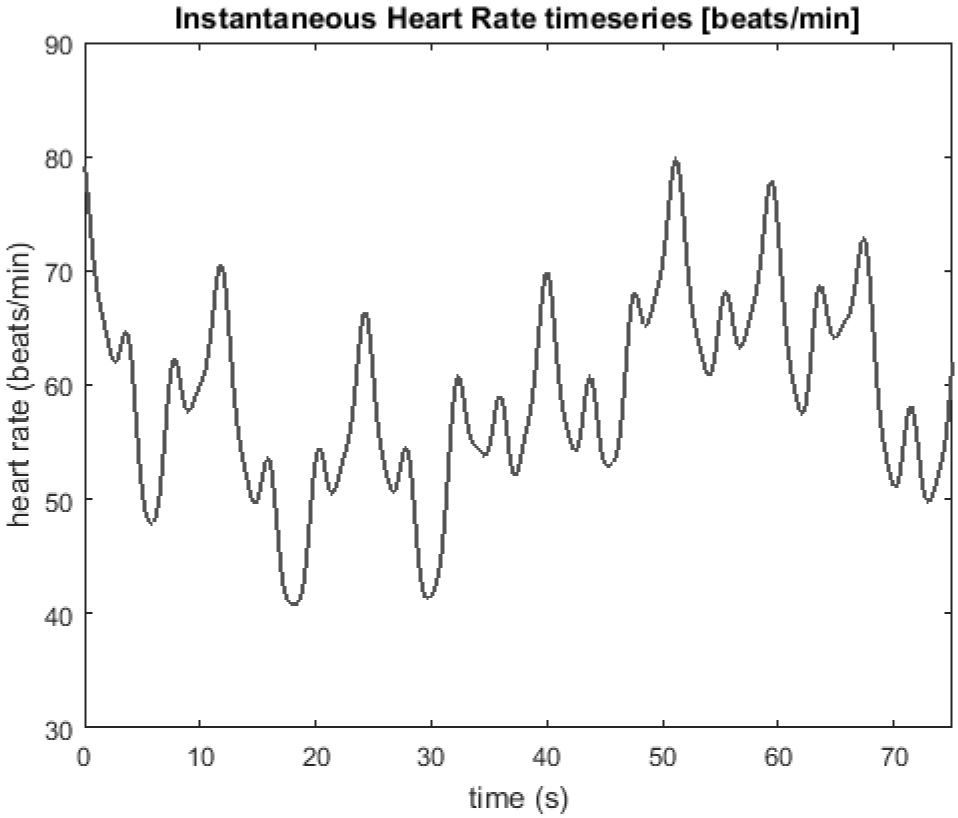

Estimated Power Spectral Density
Power Density (dB/Hz)
60
40
20
0
-20
-40
-60
-80
-100
0
0.2
0.4
0.6
0.8
1
1.2
Frequency (Hz)

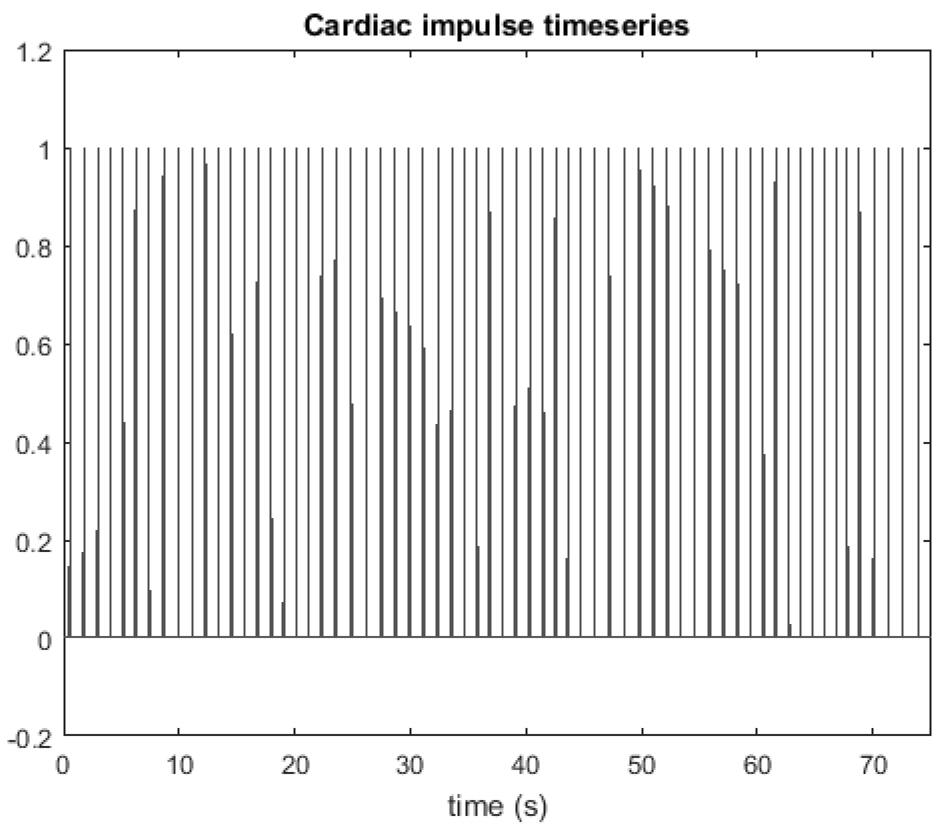

Instantaneous Heart Rate timeseries [beats/min]
heart rate (beats/min)
time (s)

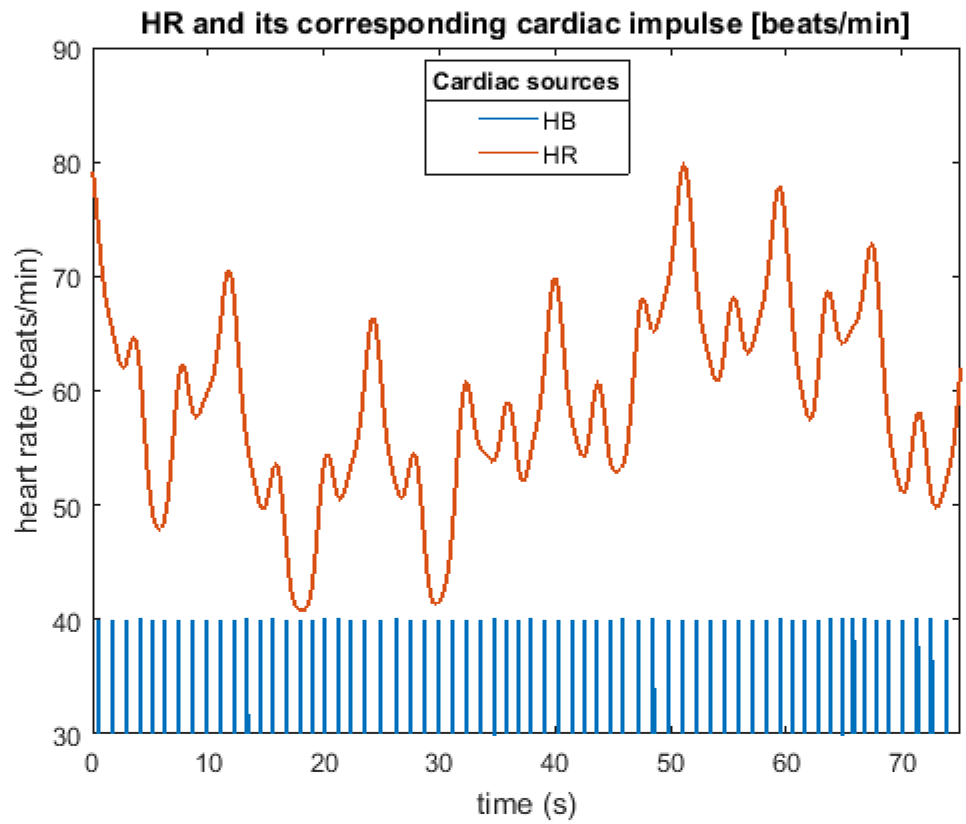

Cardiac impulse timeseries
time (s)

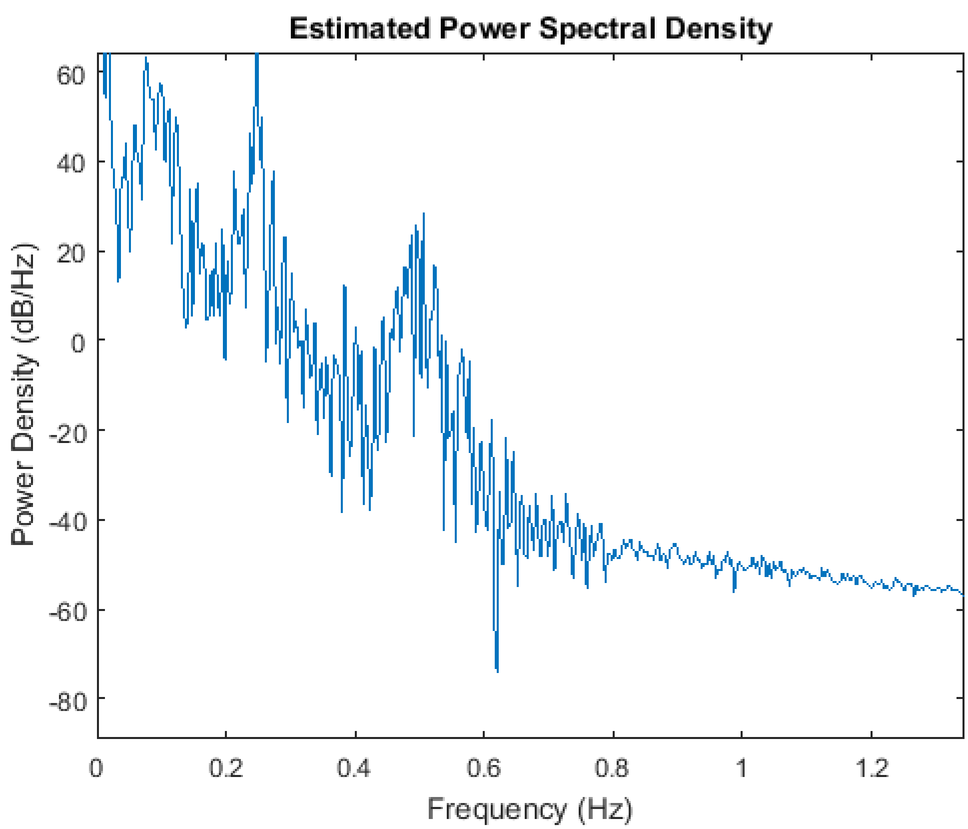

HR and its corresponding cardiac impulse [beats/min]
Cardiac sources
HB
HR
heart rate (beats/min)
time (s)
90
80
70
60
50
40
30
0
10
20
30
40
50
60
70

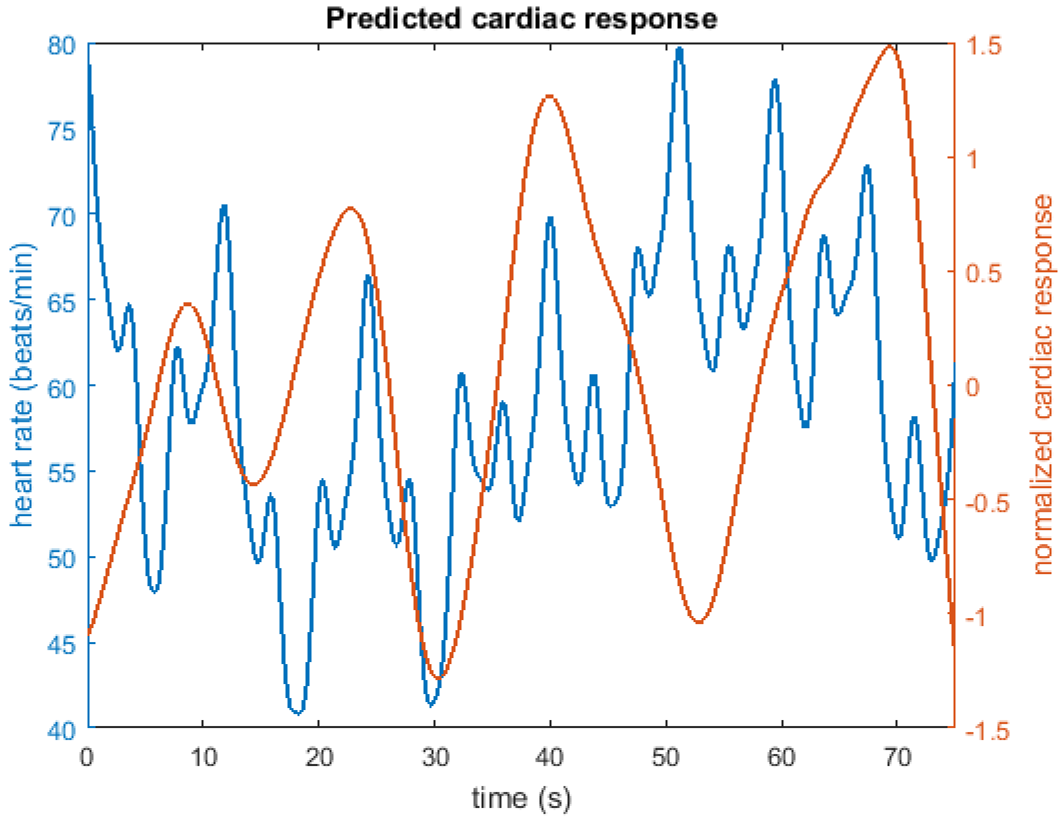

Estimated Power Spectral Density
Power Density (dB/Hz)
Frequency (Hz)
80
60
40
20
0
-20
-40
-60
-80
0
0.2
0.4
0.6
0.8
1
1.2


Instantaneous Heart Rate timeseries [beats/min]
heart rate (beats/min)
time (s)
90
80
70
60
50
40
30
0
10
20
30
40
50
60
70

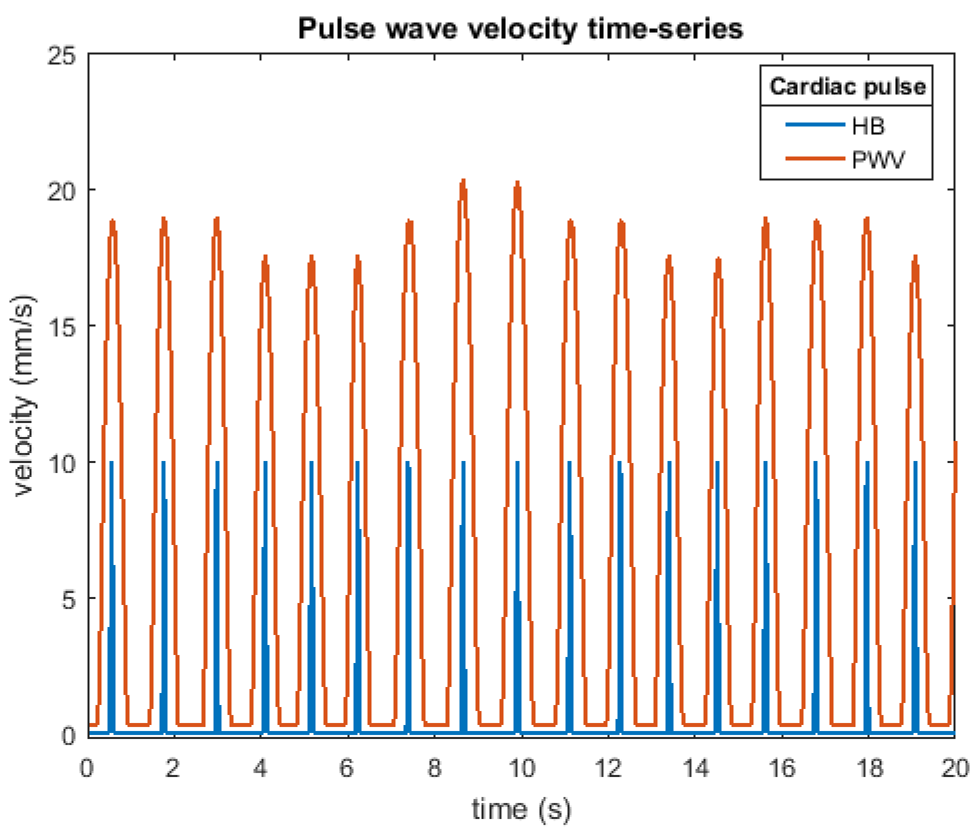

Cardiac impulse timeseries
time (s)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
10
20
30
40
50
60
70


HR and its corresponding cardiac impulse [beats/min]
Cardiac sources
HB
HR
heart rate (beats/min)
time (s)
90
80
70
60
50
40
30
0
10
20
30
40
50
60
70

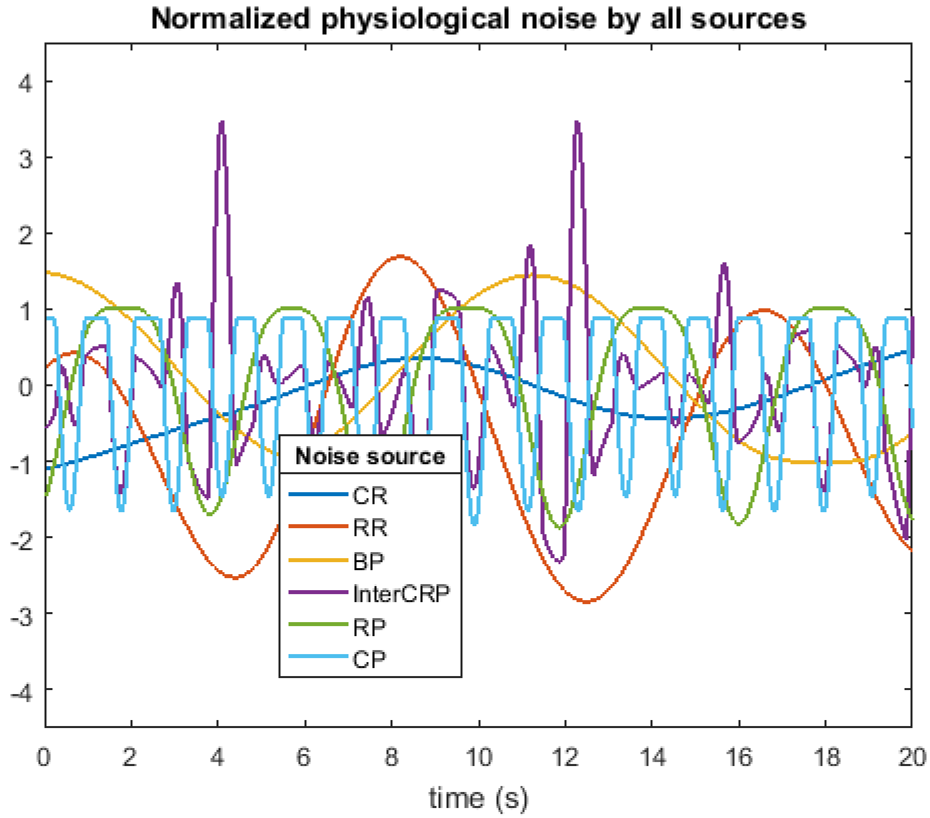

Estimated Power Spectral Density
Power Density (dB/Hz)
Frequency (Hz)
60
40
20
0
-20
-40
-60
-80
0
0.2
0.4
0.6
0.8
1
1.2

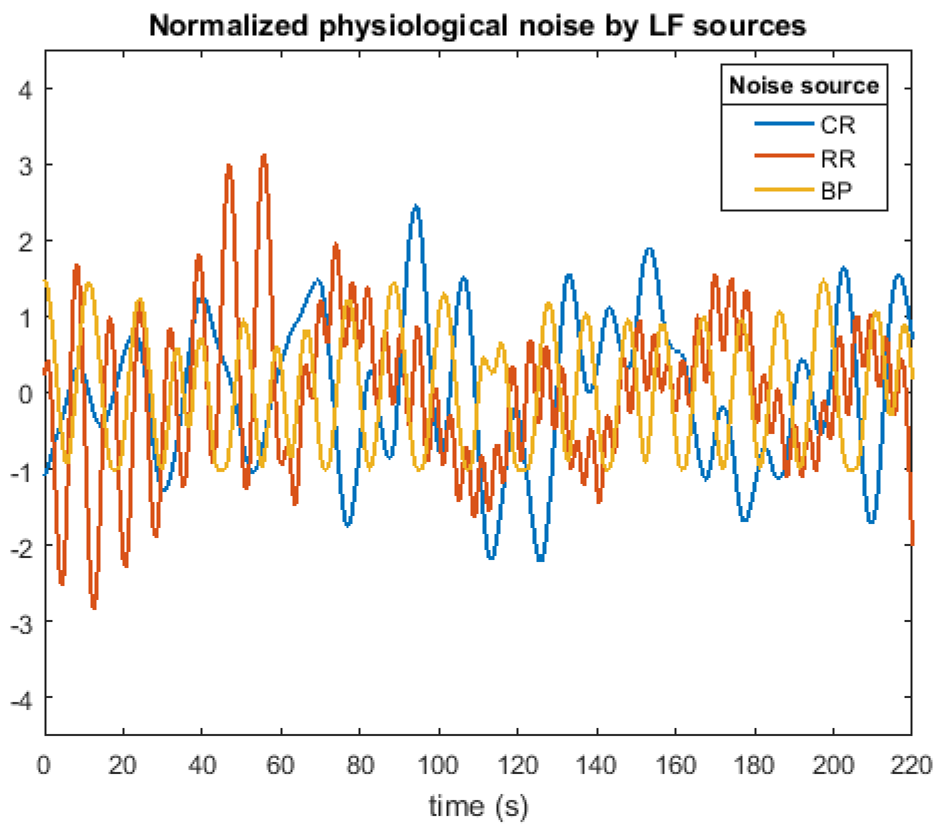


Predicted cardiac response
heart rate (beats/min)
normalized cardiac response
time (s)
80
75
70
65
60
55
50
45
40
1.5
1
0.5
0
-0.5
-1
-1.5
0
10
20
30
40
50
60
70



Pulse wave velocity time-series
Cardiac pulse
HB
PWV
velocity (mm/s)
time (s)
25
20
15
10
5
0
0
2
4
6
8
10
12
14
16
18
20


Normalized physiological noise by all sources
Noise source
CR
RR
BP
InterCRP
RP
CP
time (s)


Normalized physiological noise by LF sources
Noise source
CR
RR
BP
time (s)

Combine all sources of physiological noise per tissue type

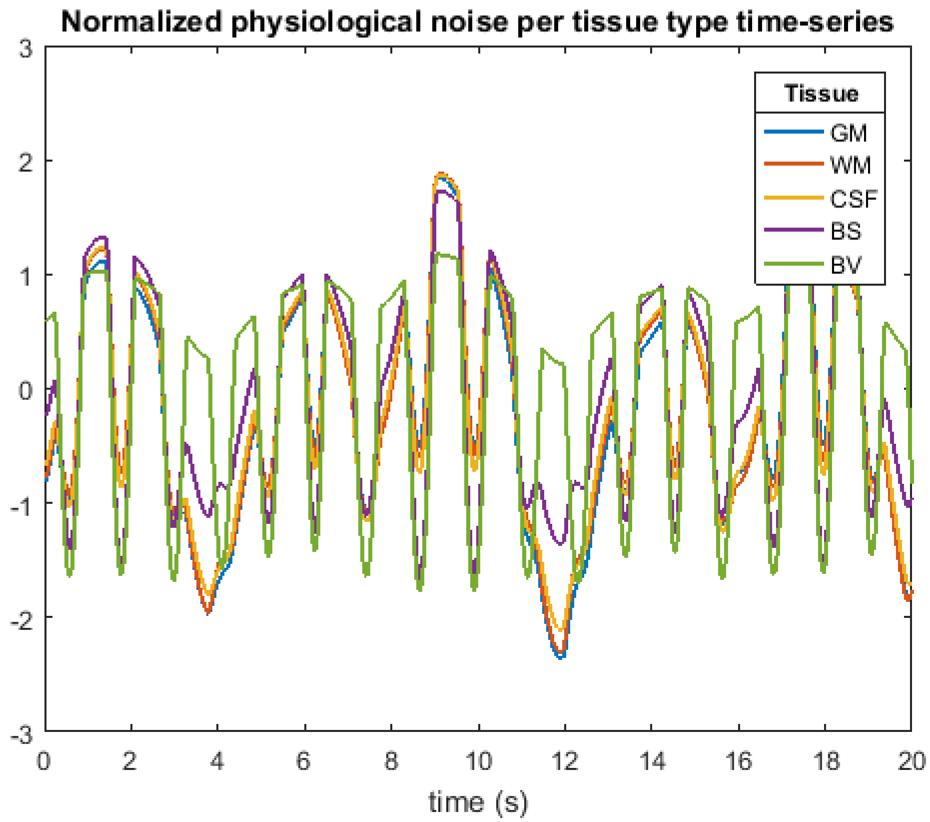
```
uiwait(msgbox('Finally combine all physiological sources for each
 tissue type.','Combine sources'));

% Physiological signal in the gray matter (GM)
display('o Constructing the physiological signal in the GM.')
GM_phys = RVT;
GM_phys.Data = 0.04*RP_ts_Data + 0.03*CP_ts_Data +
 0.017*InterCRP_GM_ts_Data + 0.021*RR_ts_Data + 0.012*CR_ts_Data;
GM_phys.Data = (GM_phys.Data - mean(GM_phys.Data(Nt_SS+2:end)))./
std(GM_phys.Data(Nt_SS+2:end));
display('Validating that the physiological time-series in the GM has
 average = 0; variance = 1:')
mean(GM_phys.Data(Nt_SS+2:end))
std(GM_phys.Data(Nt_SS+2:end))
GM_phys_noise = squeeze(GM_phys.Data);

% Physiological signal in the white matter (WM)
display('o Constructing the physiological signal in the WM.')
WM_phys = RVT;
WM_phys.Data = 0.042*RP_ts_Data + 0.032*CP_ts_Data +
 0.020*InterCRP_WM_ts_Data + 0.020*RR_ts_Data + 0.011*CR_ts_Data +
 0.00275*Mayer_wave_Data;
WM_phys.Data = (WM_phys.Data - mean(WM_phys.Data(Nt_SS+2:end)))./
std(WM_phys.Data(Nt_SS+2:end));
display('Validating that the physiological time-series in the WM has
 average = 0; variance = 1:')
mean(WM_phys.Data(Nt_SS+2:end))
std(WM_phys.Data(Nt_SS+2:end))
WM_phys_noise = squeeze(WM_phys.Data);

% Physiological signal in the cerebrospinal fluid (CSF)
display('o Constructing the physiological signal in the CSF.')
CSF_phys = RVT;
CSF_phys.Data = 0.046*RP_ts_Data + 0.042*CP_ts_Data +
 0.022*InterCRP_CSF_ts_Data + 0.022*RR_ts_Data + 0.013*CR_ts_Data +
 0.00325*Mayer_wave_Data;
CSF_phys.Data = (CSF_phys.Data - mean(CSF_phys.Data(Nt_SS+2:end)))./
std(CSF_phys.Data(Nt_SS+2:end));
display('Validating that the physiological time-series in the CSF has
 average = 0; variance = 1:')
mean(CSF_phys.Data(Nt_SS+2:end))
std(CSF_phys.Data(Nt_SS+2:end))
CSF_phys_noise = squeeze(CSF_phys.Data);

% Physiological signal in the brain stem (BS)
display('o Constructing the physiological signal in the BS.')
BS_phys = RVT;
```

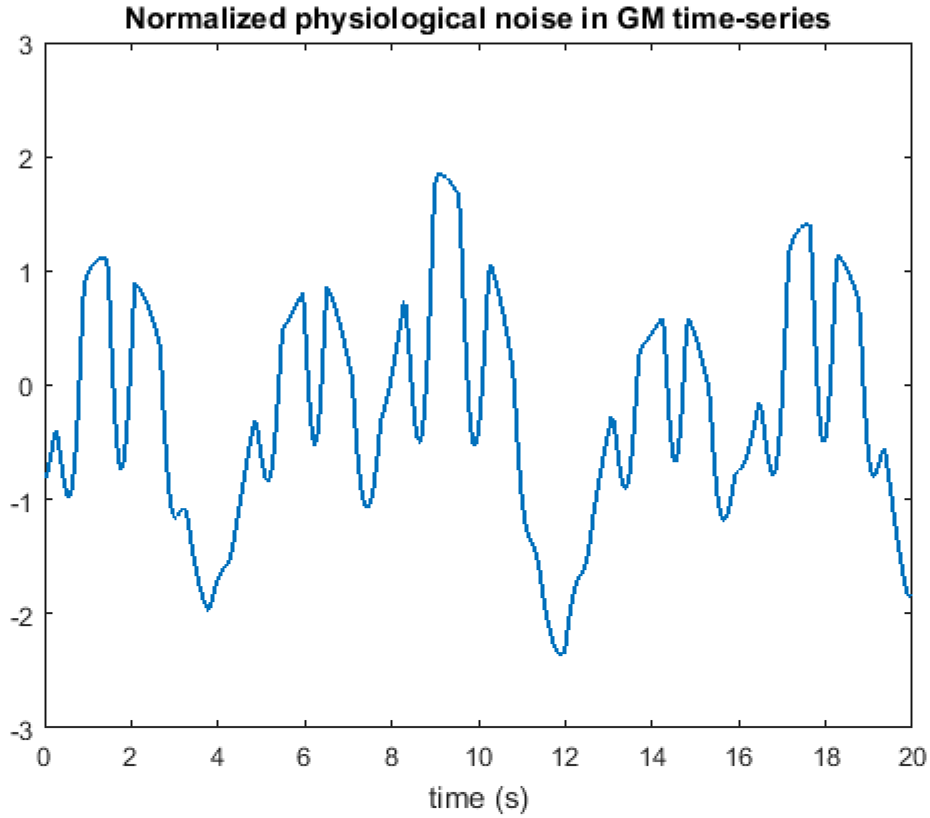
```
BS_phys.Data = 0.037*RP_ts_Data + 0.079*CP_ts_Data +
 0.040*InterCRP_BS_ts_Data + 0.017*RR_ts_Data + 0.009*CR_ts_Data +
 0.00225*Mayer_wave_Data;
BS_phys.Data = (BS_phys.Data - mean(BS_phys.Data(Nt_SS+2:end)))./
std(BS_phys.Data(Nt_SS+2:end));
display('Validating that the physiological time-series in the BS has
 average = 0; variance = 1:')
mean(BS_phys.Data(Nt_SS+2:end))
std(BS_phys.Data(Nt_SS+2:end))
BS_phys_noise = squeeze(BS_phys.Data);

% Physiological signal in the blood vessels (BV):
% estimate for blood vessel from average values plus that the blood
 flow
% velocity in a blood vessel is ~70 times that of capillaries in the
 GM
% so the CP std is expected to be ~8 times as much
display('o Constructing the physiological signal in the BV.')
BV_phys = RVT;
BV_phys.Data = 0.033*RP_ts_Data + 0.25*CP_ts_Data +
 0.02*InterCRP_BV_ts_Data + 0.02*RR_ts_Data + 0.01*CR_ts_Data +
 0.0025*Mayer_wave_Data;
BV_phys.Data = (BV_phys.Data - mean(BV_phys.Data(Nt_SS+2:end)))./
std(BV_phys.Data(Nt_SS+2:end));
display('Validating that the physiological time-series in the BV has
 average = 0; variance = 1:')
mean(BV_phys.Data(Nt_SS+2:end))
std(BV_phys.Data(Nt_SS+2:end))
BV_phys_noise = squeeze(BV_phys.Data);


% physiological noise vs tissue type
h_physiotissue = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_20s+1),GM_phys_noise(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),WM_phys_noise(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),CSF_phys_noise(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),BS_phys_noise(Nt_SS+2:Nt_SS+Nt_20s
+1),times(Nt_SS+2:Nt_SS+Nt_20s+1),BV_phys_noise(Nt_SS+2:Nt_SS+Nt_20s
+1),'LineWidth', 1.5)
ylim([-3.0 3.0])
xlim([0 times(Nt_SS+Nt_20s+1)])
xlabel('time (s)')
title('Normalized physiological noise per tissue type time-series')
lgdn5 = legend('GM','WM','CSF','BS','BV');
title(lgdn5,'Tissue')
movegui(h_physiotissue,'southeast');

% physiological noise in gray matter
h_physioGM = figure;
plot(times(Nt_SS+2:Nt_SS+Nt_20s+1),GM_phys_noise(Nt_SS+2:Nt_SS+Nt_20s
+1),'LineWidth', 1.5)
ylim([-3.0 3.0])
xlim([0 times(Nt_SS+Nt_20s+1)])
xlabel('time (s)')
```

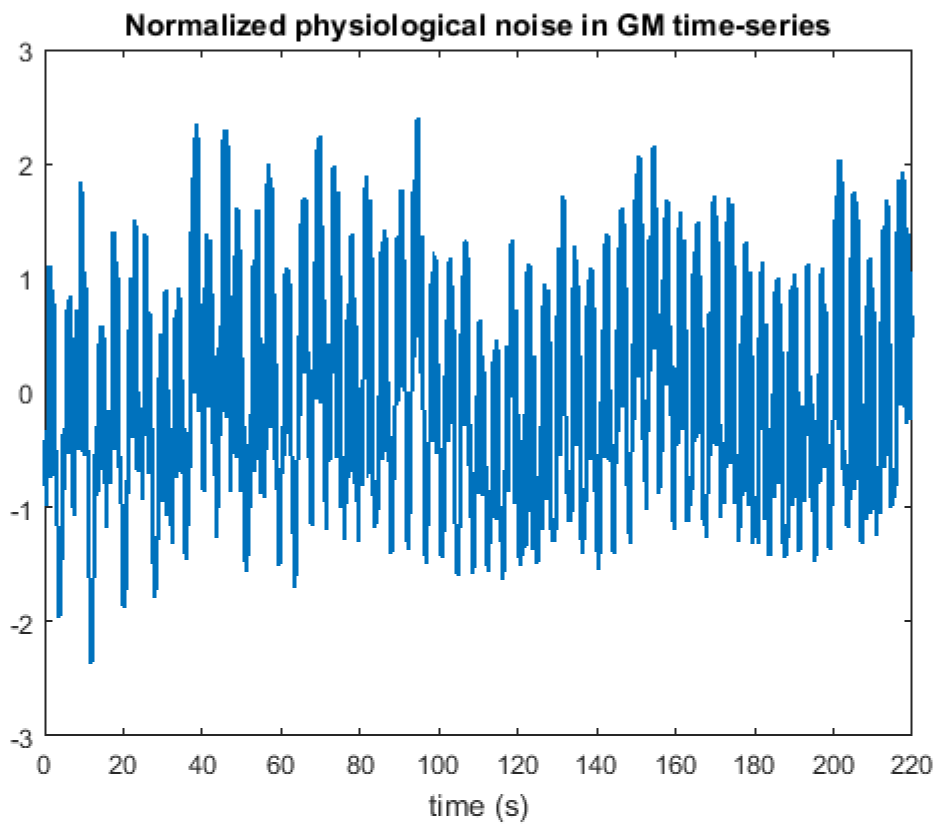
```
title('Normalized physiological noise in GM time-series')
movegui(h_physioGM,'southwest');

% physiological noise in gray matter
h_physioGM2 = figure;
plot(times(Nt_SS+2:end),GM_phys_noise(Nt_SS+2:end),'LineWidth', 1.5)
ylim([-3.0 3.0])
xlim([0 times(end)])
xlabel('time (s)')
title('Normalized physiological noise in GM time-series')
movegui(h_physioGM2,'north');

cd(currentDir)
```

```
o Constructing the physiological signal in the GM.
Validating that the physiological time-series in the GM has average =
 0; variance = 1:

ans =

  -5.5323e-17


ans =

    1.0000

o Constructing the physiological signal in the WM.
Validating that the physiological time-series in the WM has average =
 0; variance = 1:

ans =

   1.4787e-18


ans =

    1.0000

o Constructing the physiological signal in the CSF.
Validating that the physiological time-series in the CSF has average =
 0; variance = 1:

ans =

   5.0488e-17


ans =

     1

o Constructing the physiological signal in the BS.
```

```
Validating that the physiological time-series in the BS has average =
 0; variance = 1:

ans =

  -2.2416e-17


ans =

    1.0000

o Constructing the physiological signal in the BV.
Validating that the physiological time-series in the BV has average =
 0; variance = 1:

ans =

  -2.5115e-18


ans =

    1.0000
```




Normalized physiological noise in GM time-series
3
2
1
0
-1
-2
-3
0
2
4
6
8
10
12
14
16
18
20
time (s)


Normalized physiological noise in GM time-series
3
2
1
0
-1
-2
-3
0
20
40
60
80
100
120
140
160
180
200
220
time (s)

Published with MATLAB® R2016a