Table of Contents

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

Emulates the fMRI data from a block design experiment of alternating R & L finger tasks in the real data fmriblocks009.nii as detailed on the webpage

```
http://www.mccauslandcenter.sc.edu/crnl/sw/tutorial/html/blockspm.html
```

provided by Chris Rorden in his online fMRI processing tutorial for SPM.

Jason E. Hill demo_4D_simblock.m updated 2 APR 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 warnings ...

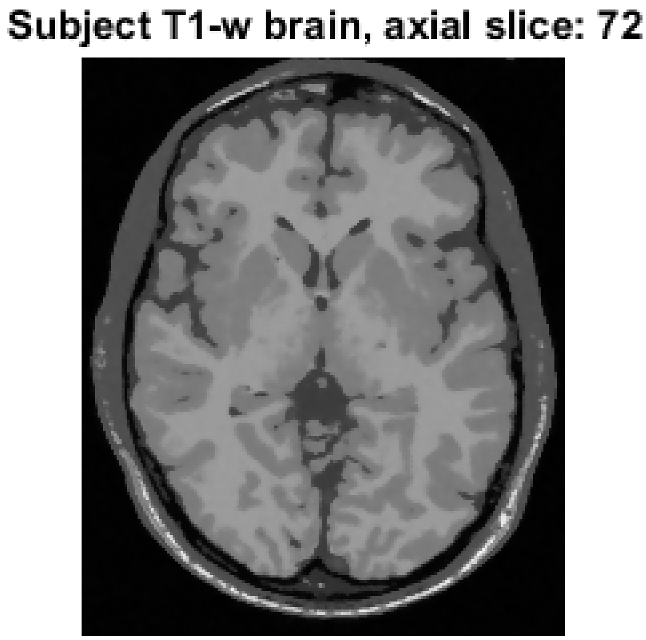
... OpenGl warnings

```
warning('off','MATLAB:opengl:StartupBlacklistedNoSetting');
warning('off', 'MATLAB:hg:AutoSoftwareOpenGL');
% ... finite warning
warning('off', 'MATLAB:FINITE:obsoleteFunction');
% ... NIFTI class warnings 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');
```

Select subject by index (originally there are 20 subjects to choose from)

```
subject_brain = 6;
Now_sss = [5 1 1];
filepathOut = STANCE_genpath(Now_sss);
if ~logical(exist(filepathOut,'file'))
    STANCE_new_session(Now_sss);
end
makeFMRI = true;
```

```
% for reproducibility
s = 0;
%s = []; % allow MATLAB to spontaneously shuffle
if ~isempty(s)
    rng(s);
end

% show MNI volume conformed to BrainWEB dimensions
[V_MNI,Y_MNI] = STANCE_load_volume(filenameMNI);
MNI_dim = V_MNI.dim;
MNI_mat = V_MNI.mat;
origin  = abs(V_MNI.mat(1:3,4))';

[~,I_max] = max(sum(sum(Y_MNI)));
showSlice = I_max(1);

% load the T1w data for subject, for display purposes
[V_T1w,Y_T1w] = STANCE_choose_subject(subject_brain,'T1');

T1w_dim = V_T1w.dim;  % dimensions of T1-w volume
T1w_mat = V_T1w.mat;  % 4x4 homographic matrix relating indeces to
 real-world coordinates
f1 = figure;
subplot(2,1,2)
imshow(imrotate(Y_T1w(:,:,showSlice),90),[]), drawnow;
TITLE = ['Subject T1-w brain, axial slice: ',num2str(showSlice)];
title(TITLE)
truesize
movegui(f1,'north');

% retrieve transfromation matrix mapping MNI152 to subject's native
 space
M = M_array(:,:,subject_brain);

[V_MNI_reg,Y_MNI_reg] = STANCE_register_MNI(V_T1w.fname,M);

dimensions = size(Y_T1w);
origin = round(abs(V_T1w.mat(1:3,4)))';
```



Build activation regions by modelling reported results

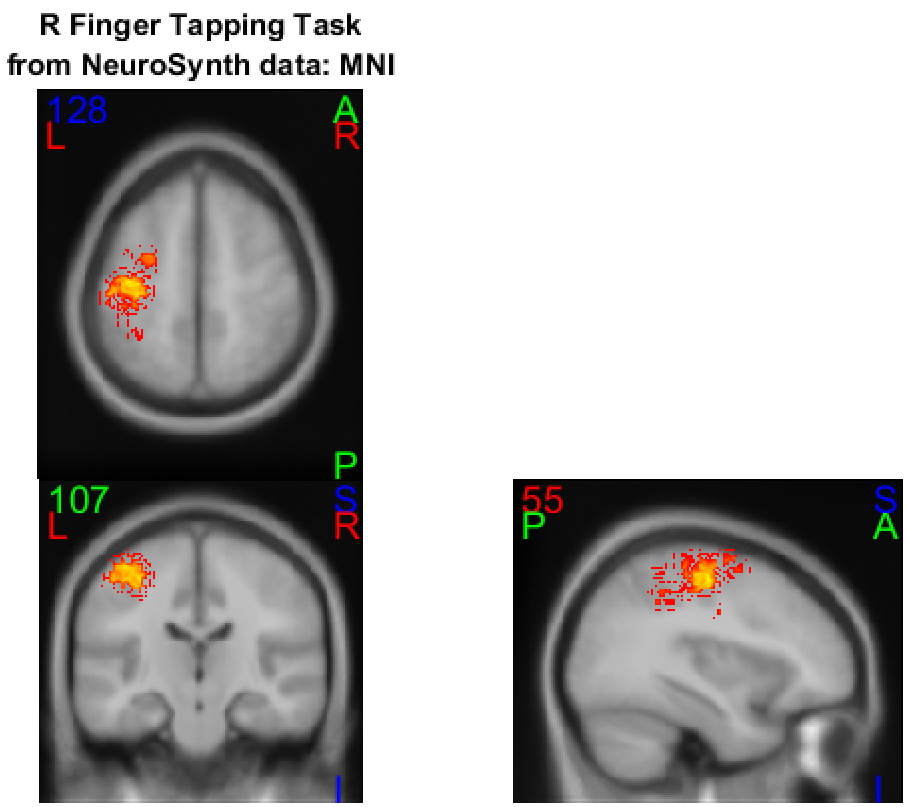
```
% Simulate R/L handed finger tapping task activation and
% block experimental design as detailed on
% Chris Rorden's educational websites:
% http://www.mccauslandcenter.sc.edu/crnl/sw/tutorial/index.html
% and
% http://people.cas.sc.edu/rorden/mricron/peri/index.html

% load activation map from data files
clear task;

uiwait(msgbox('Demo example of a block experimental design of L/R
 finger tasks.','Finger meta-analysis data','modal'));

% load NeuroSynth finger tapping data to construct activation
task_R.name = 'R Finger Tapping';
task_R.activation(1).region    = [STANCEroot,'/activations/finger
 tapping_pFgA_z_FDR_0.01.nii.gz'];
task_R.activation(1).shape     = 'data';     % data derived by forward
 inference
task_R.activation(1).volume    = [];         % the index for 4D data
task_R.activation(1).proportion = 5.0;      % the activation
 thresholds: [lower, saturation]
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task_R.activation(2).region    = [STANCEroot,'/activations/finger
 tapping_pAgF_z_FDR_0.01.nii.gz'];
task_R.activation(2).shape     = 'data';     % data derived by reverse
 inference
task_R.activation(2).proportion = 5.0;     % the activation
 thresholds: [lower, saturation]
task_R.combine{1} = {'OR','all'};
task_R.combine{2} = {'AND','flip'};
task_R.combine{3} = {'mask','L'};
% define signal amplitude
task_R.amplitude = 0.119; % according to the apparent activation level
 seen in fmriblocks009.nii

task_L = task_R;
task_L.name = 'L Finger Tapping';
task_L.combine{3} = {'mask','R'};
task_L.amplitude = 0.91*task_R.amplitude; % according to the apparent
 activation seen in fmriblocks009.nii

if exist('simulations.mat','file')
    load('simulations.mat');
else
    % create simulations struct
end
simulations{Now_sss(1)}.task{1} = task_R;
simulations{Now_sss(1)}.task{2} = task_L;

% left brain component of right-handed task
disp('Defining R finger tapping task activation map...')
task_R.activation(1).map =
 STANCE_load_map(task_R.activation(1).region,V_MNI,5.0,false);
task_R.activation(2).map =
 STANCE_load_map(task_R.activation(2).region,V_MNI,5.0,false);

disp('o Combining finger tapping task activation maps from data
 files...')
task_R.map = STANCE_parse_combine(task_R);

% supress bright artefact on medial surface and inferior area
task_R.map(ceil(0.38*dimensions(1)):end,:,:) =
 0; %0.1*task_R.map(ceil(0.4*dimensions(1)):end,:,:);
task_R.map(:,:,1:90) = 0;

% find the ammount of gray matter volume for the activation map based
 on MNI tissue priors
task_R.GMvolume = STANCE_find_GM_volume(task_R);

% free up working memory (optional)
task_R.activation(1).map = [];
task_R.activation(2).map = [];

TITLE = {'R Finger Tapping Task','from NeuroSynth data: MNI'};
htask2 = STANCE_display_activation_slice(Y_MNI,task_R.map,[],[]);
```

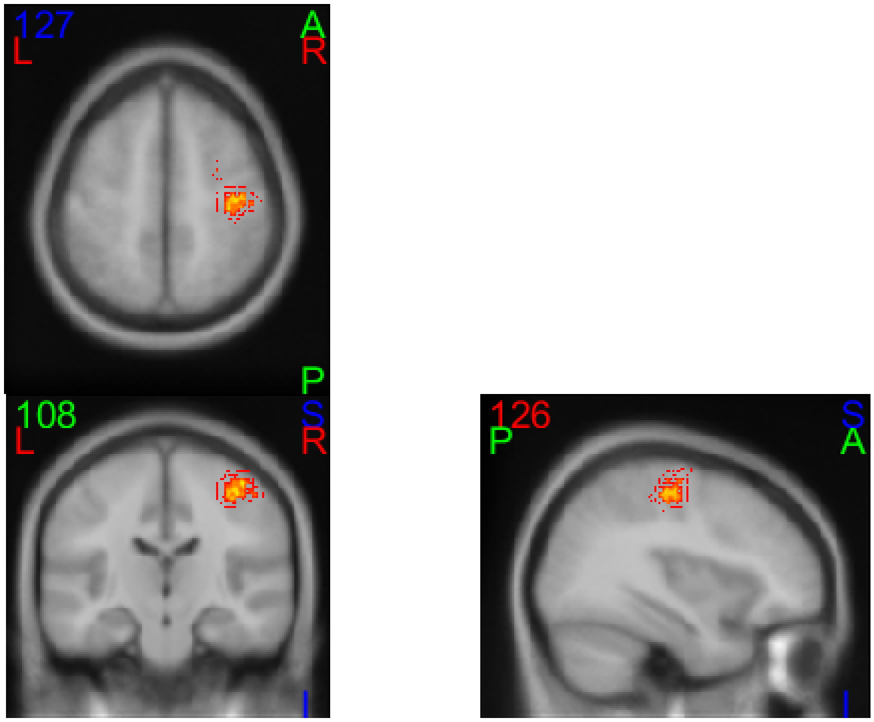
```
title(TITLE)
movegui(htask2,'west');


% right brain component of left-handed task
% NOTE: use fact that L activation is smaller (peak R t-stat ~= 1.5 L
 t-stat)
disp('Defining L finger tapping task activation map...')
task_L.activation(1).map =
 STANCE_load_map(task_L.activation(1).region,V_MNI,7.5,false);
task_L.activation(2).map =
 STANCE_load_map(task_L.activation(2).region,V_MNI,7.5,false);

disp('o Combining finger tapping task activation maps from data
 files...')
task_L.map = STANCE_parse_combine(task_L);

% suppress bright artefact on medial surface and inferior area
task_L.map(ceil(1:floor(0.62*dimensions(1))),:,:) =
 0; %0.1*task_L.map(1:floor(0.6*dimensions(1)):end,:,:);
task_L.map(:,:,1:90) = 0;

% find the ammount of gray matter volume for the activation map based
 on MNI tissue priors
task_L.GMvolume = STANCE_find_GM_volume(task_L);

% free up working memory (optional)
task_L.activation(1).map = [];
task_L.activation(2).map = [];

TITLE = {'L Finger Tapping Task','from NeuroSynth data: MNI'};
htask2 = STANCE_display_activation_slice(Y_MNI,task_L.map,[],[]);
title(TITLE)
movegui(htask2,'east');

% register activation maps to subject brain
[V_finger_tapping_R_reg,Y_finger_tapping_R_reg] =
 STANCE_register_activation(V_T1w.fname,task_R);
[V_finger_tapping_L_reg,Y_finger_tapping_L_reg] =
 STANCE_register_activation(V_T1w.fname,task_L);

% Finger tapping task activation template of the PSM of subject
TITLE = {'R Finger tapping task'; 'activation template in subject'};
htask1sub =
 STANCE_display_activation_slice(Y_T1w,Y_finger_tapping_R_reg,[],[]);
title(TITLE)
movegui(htask1sub,'center');

% make room in memory
if ~strcmp(V_T1w.fname(end-1:end),'gz')
    delete(V_T1w.fname);
end
clear('V_T1w','YT1w');
delete(V_MNI_reg.fname);
```

```
clear('V_MNI_reg','Y_MNI_reg','V_MNI','Y_MNI')

% save activation template
task_R.map = int8(255*task_R.map);
task_L.map = int8(255*task_L.map);
cd([STANCEroot,'/activations'])
save([task_R.name,'.mat'],'task_R')
save([task_L.name,'.mat'],'task_L')
cd(STANCEroot)
task_R.map = [];
task_L.map = [];
```

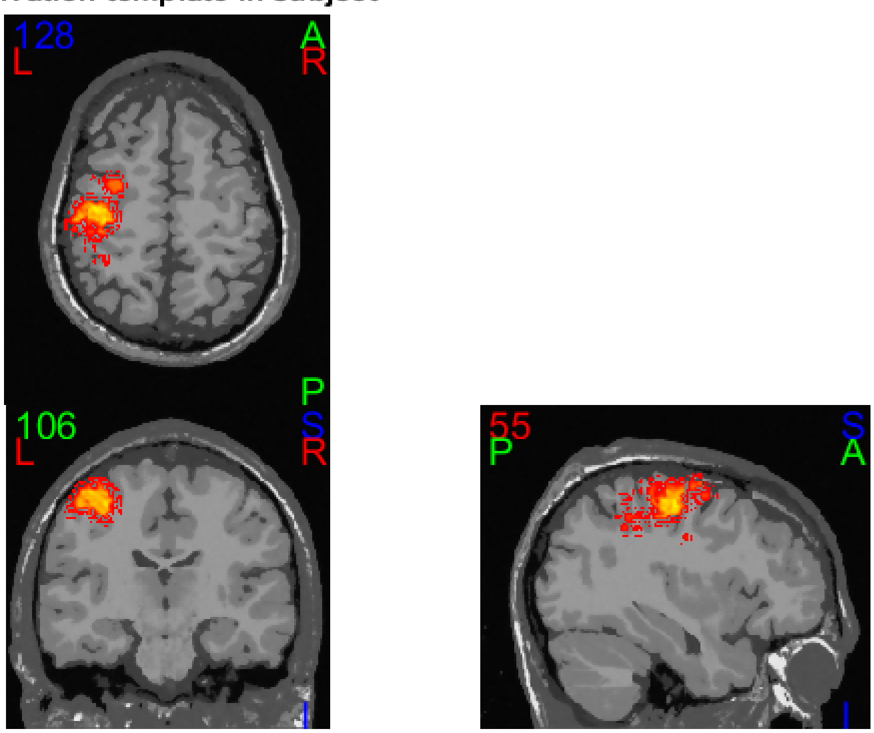
```
Defining R finger tapping task activation map...
o Loading map from data in C:\spm\STANCE/activations/finger
 tapping_pFgA_z_FDR_0.01.nii.gz.
o Loading map from data in C:\spm\STANCE/activations/finger
 tapping_pAgF_z_FDR_0.01.nii.gz.
o Combining finger tapping task activation maps from data files...
Defining L finger tapping task activation map...
o Loading map from data in C:\spm\STANCE/activations/finger
 tapping_pFgA_z_FDR_0.01.nii.gz.
o Loading map from data in C:\spm\STANCE/activations/finger
 tapping_pAgF_z_FDR_0.01.nii.gz.
o Combining finger tapping task activation maps from data files...
```



**L Finger Tapping Task
from NeuroSynth data: MNI**



**R Finger tapping task
activation template in subject**



Reslice volume according to fMRI scan protocol specifications

```
disp('Reslicing volumes to functional space...')
if makeFMRI == true

choiceFlag = 0;
cd(STANCE_genpath(Now_sss,2))
if logical(exist([STANCE_genpath(Now_sss,2),'/
STANCEsubject.mat'],'file'))
cd(STANCE_genpath(Now_sss))
if logical(exist([STANCE_genpath(Now_sss),'/STANCEscan.mat'],'file'))
    % Construct a questdlg with two options
    choice = questdlg('Previous scan found for this subject and
 session, load info?', ...
 'Scan found', ...
 'Yes','No','No');
    % Handle response
    switch choice
        case 'Yes'
            load('STANCEscan.mat')
            cd(STANCE_genpath(Now_sss,2))
            load('STANCEsubject.mat')
            choiceFlag = 1;
        case 'No'
            STANCE_new_session(Now_sss);
    end
end
end
cd(STANCEroot)

if ~choiceFlag
    % define scan protocol parameters (these are based on those at
 http://www.mccauslandcenter.sc.edu/crnl/sw/tutorial/index.html
    scan.voxel.size    = [3 3 3.6];
    scan.voxel.matrix  = [64 64 36];  % [64 50 24]; in original
 experiment
    scan.voxel.spacing = [0 0 0.2*scan.voxel.size(3)]; % assume 20% Z
 spacing
    scan.tiltAngle      = 0;    % degrees
    scan.NV             = 302;  % number of volumes
    scan.TR             = 1920; % [ms]
    scan.TE             = 35;   % [ms]
    scan.ES             = 0.51; % [ms] echo spacing
    scan.FA             = 11;   % [degrees]
    scan.BW             = 2232; % [Hz/Px]
    scan.order          = 'SA'; % SA = sequential ascending order
    scan.KM0            = (4.5315*2225); % fit to data with typical
 value of 1151 in CSF and 960 in full GM
    scan.noise_method  = 'percent';
    scan.noise          = 0;    % percent noise relative to peak
    scan.attenuation    = 0;    % coil attenuation factor ~mm^-1
```

```
    scan.acceleration  = 1.0;  % no slice-wise acceleration
    scan.interpolation = 2.0; % used GRAPPA to interpolate acquisition
 32->64 lines
    % NOTE: also have that FWHM ~4.5 mm

    simulations{Now_sss(1)}.scan = scan;
    save('simulations.mat','simulations')

    % load tissue fuzzy memberships in subject's native space
    [V_fuzzy,~] = STANCE_choose_subject(subject_brain,'fuzzy',true);
    fn_tissue = [V_fuzzy(1).fname,'.gz'];

    % generate the tissue fuzzy memberships in functional space
    [V_reslice,Y_reslice] = STANCE_reslice_tissue(fn_tissue,scan,[],
[],false,Now_sss); % change the last arg to 'true' to show figures
    sliceLimits =
 [V_reslice(1).sliceLimitLower,V_reslice(1).sliceLimitUpper];
    [~,I_max] = max(sum(sum(Y_reslice(:,:,:,3))));
    showSlice2 = I_max(1);
    % figure, imshow(imrotate(Y_reslice(:,:,showSlice2,3),90),[]);
    % TITLE = ['Reslice tissue priors - gray matter, A
 slice:',num2str(showSlice2TA)];
    % title(TITLE)
    fn_fuzzy_reslice = V_reslice(1).fname;

    % generate T2* baseline volume in functional space
    [V_T2star_Map,Y_T2star_Map] =
 STANCE_make_parameter_map(fn_fuzzy_reslice,'T2star');
    scrsz = get(groot,'ScreenSize');
    positionVector2 = [scrsz(3)/2.5 scrsz(4)/2.5 scrsz(3)/5
 scrsz(4)/3];
    % f2 = figure;
    % imshow(imrotate(Y_T2star_Map(:,:,showSlice2),90),[])
    % title('T2* baseline volume')
    % set(f2,'OuterPosition',positionVector2);
    % movegui(f2,'north')

    % project activation map on to functional space
    [~,Y_finger_tapping_R_reslice] =
 STANCE_reslice_volume(V_finger_tapping_R_reg,scan,sliceLimits);
    [~,I_max] = max(sum(sum(Y_finger_tapping_R_reslice)));
    showSlice2TA = I_max(1);
    % TITLE = {'R Finger tapping task,'; ['functional axial:
 ',num2str(showSlice2TA)]};
    % f3R = figure;
    %
 imshow(imrotate(Y_finger_tapping_R_reslice(:,:,showSlice2TA),90),[])
    % title(TITLE)
    % set(f3R,'OuterPosition',positionVector2);
    % movegui(f3R,'northeast')
    delete(V_finger_tapping_R_reg.fname);
    clear V_finger_tapping_R_reg Y_finger_tapping_R_reg;

    % project activation map on to functional space
```

```
    [~,Y_finger_tapping_L_reslice] =
STANCE_reslice_volume(V_finger_tapping_L_reg,scan,sliceLimits);
    [~,I_max] = max(sum(sum(Y_finger_tapping_L_reslice)));
    showSlice2TA = I_max(1);
    % TITLE = {'L Finger tapping task,'; ['functional axial:
',num2str(showSlice2TA)]};
    % f3L = figure;
    %
imshow(imrotate(Y_finger_tapping_L_reslice(:,:,showSlice2TA),90),[])
    % title(TITLE)
    % set(f3L,'OuterPosition',positionVector2);
    % movegui(f3L,'southeast')
    delete(V_finger_tapping_L_reg.fname);
    clear V_finger_tapping_L_reg Y_finger_tapping_L_reg;

    % mask with gray matter mask
    [Y_finger_tapping_R_reslice,Y_GM] =
STANCE_GM_mask(Y_finger_tapping_R_reslice,task_R.GMvolume,Now_sss);
    [Y_finger_tapping_L_reslice,~]    =
STANCE_GM_mask(Y_finger_tapping_L_reslice,task_L.GMvolume,Now_sss);

    subjectActivation3D{1} = Y_finger_tapping_R_reslice;
    subjectActivation3D{2} = Y_finger_tapping_L_reslice;

else
    % load the tissue fuzzy memberships in functional space
    [V_reslice,Y_reslice] = STANCE_load_volume(fn_fuzzy_reslice);
    [~,I_max] = max(sum(sum(Y_reslice(:,:,:,3))));
    showSlice2 = I_max(1);
    % figure, imshow(imrotate(Y_reslice(:,:,showSlice2,3),90),[]);
    % TITLE = ['Reslice tissue priors - gray matter, A
slice:',num2str(showSlice2TA)];
    % title(TITLE)
    fn_fuzzy_reslice = V_reslice(1).fname;

    Y_finger_tapping_R_reslice = subjectActivation3D{1};
    Y_finger_tapping_L_reslice = subjectActivation3D{2};

    % generate T2* baseline volume in functional space
    [V_T2star_Map,Y_T2star_Map] =
STANCE_make_parameter_map(fn_fuzzy_reslice,'T2star');
    scrsz = get(groot,'ScreenSize');
    positionVector2 = [scrsz(3)/2.5 scrsz(4)/2.5 scrsz(3)/5
scrsz(4)/3];
    % f2 = figure;
    % imshow(imrotate(Y_T2star_Map(:,:,showSlice2),90),[])
    % title('T2* baseline volume')
    % set(f2,'OuterPosition',positionVector2);
    % movegui(f2,'north')

    % mask with gray matter mask
    [Y_finger_tapping_R_reslice,Y_GM] =
STANCE_GM_mask(Y_finger_tapping_R_reslice,task_R.GMvolume,Now_sss);
```

```
    [Y_finger_tapping_L_reslice,~]    =
 STANCE_GM_mask(Y_finger_tapping_L_reslice,task_L.GMvolume,Now_sss);
end

% save all of the elements common to the subject
cd(STANCE_genpath(Now_sss,2))
save('STANCEsubject.mat','Now_sss','subject_brain','fn_tissue','fn_fuzzy_reslice',
cd(STANCEroot)

clear subjectActivation3D;

[~,I_max] = max(sum(sum(Y_finger_tapping_R_reslice)));
showSlice2TA = I_max(1);
```

```
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session001.
Warning: Directory already exists.
Reslicing gray matter fuzzy membership labels...
Reslicing background fuzzy membership labels...
Reslicing CSF fuzzy membership labels...
Reslicing white matter fuzzy membership labels...
Reslicing fat fuzzy membership labels...
Reslicing muscle fuzzy membership labels...
Reslicing skin fuzzy membership labels...
Reslicing skull fuzzy membership labels...
Reslicing blood vessels fuzzy membership labels...
Reslicing connective tissue fuzzy membership labels...
Reslicing dura matter fuzzy membership labels...
Reslicing bone marrow fuzzy membership labels...
```

Adding activation to T2* baseline

```
disp('Add activation to T2* baseline...')
[V_T2star_Map_Act_R,Y_T2star_Map_Act_R] =
 STANCE_add_activation(V_T2star_Map.fname,Y_finger_tapping_R_reslice,scan.TE,task_
[V_T2star_Map_Act_L,Y_T2star_Map_Act_L] =
 STANCE_add_activation(V_T2star_Map.fname,Y_finger_tapping_L_reslice,scan.TE,task_
% TITLE = {'T2* map w/ BOLD activation R,', ['Axial slice:
 ',num2str(showSlice2TA)]};
% f4 = figure;
% imshow(imrotate(Y_T2star_Map_Act_R(:,:,showSlice2TA),90),[])
% title(TITLE)
% set(f4,'OuterPosition',positionVector2);
% movegui(f4,'east')
```

```
Add activation to T2* baseline...
```

Generate the EPI baseline

```
disp('Creating baseline EPI signal ...')
[~,Y_EPI] = STANCE_EPI_signal(fn_fuzzy_reslice,Y_T2star_Map,scan);
% figure, imshow(imrotate(Y_EPI0(:,:,showSlice2TA),90),[]);
% TITLE = {'Gray matter priors,', ['axial
 slice:',num2str(showSlice2TA)]};
```

```
% f5 = figure;
% imshow(imrotate(Y_GM(:,:,showSlice2TA),90),[])
% title(TITLE)
% set(f5,'OuterPosition',positionVector2);
% movegui(f5,'southeast')

maxS = max(Y_EPI(:).*Y_GM(:));
% TITLE = {'Baseline signal volume,',['Axial slice:
 ',num2str(showSlice2TA)]};
% f6 = figure;
% imshow(imrotate(Y_EPI0(:,:,showSlice2TA),90),[0,maxS])
% title(TITLE)
% set(f6,'OuterPosition',positionVector2);
% movegui(f6,'south')

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits')
cd(STANCEroot)
```

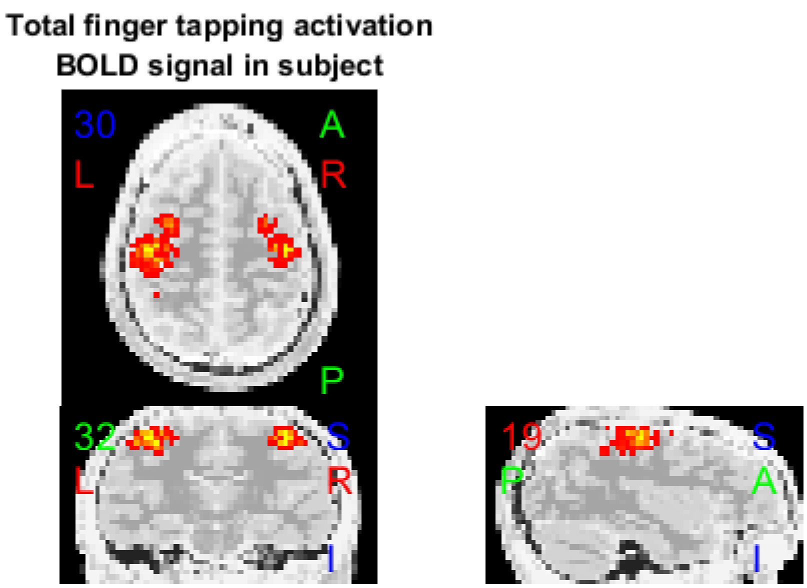
```
Creating baseline EPI signal ...
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 1145.8984
```

Generate the pristine EPI signal for right handed finger tapping

```
disp('Creating pristine EPI data for RH task...')

Now_sss = [5 1 2];
STANCE_new_session(5,1,2,true);

% exact EPI signal, no noise, no attenuation
[V_EPI,Y_EPI] =
 STANCE_EPI_signal(fn_fuzzy_reslice,Y_T2star_Map_Act_R,scan);
maxS = max(Y_EPI(:).*Y_GM(:));
% TITLE = {'Exact BOLD signal,',['Axial slice:
 ',num2str(showSlice2TA)]};
% f7 = figure;
% imshow(imrotate(Y_EPI(:,:,showSlice2TA),90),[0,maxS])
% title(TITLE)
% set(f7,'OuterPosition',positionVector2);
% movegui(f7,'southwest')

% f8 = figure;
% imshow(imrotate(Y_EPI(:,:,showSlice2TA),90)-
imrotate(Y_EPI0(:,:,showSlice2TA),90),[])
% title('(BOLD - baseline) signal')
% set(f8,'OuterPosition',positionVector2);
% movegui(f8,'west')
```

```
% Finger tapping task activation of the PSM in subject
TITLE = {'Total finger tapping activation', 'BOLD signal in subject'};
htask1subfun =
 STANCE_display_activation_slice(Y_EPI,Y_finger_tapping_R_reslice
+Y_finger_tapping_L_reslice,[],[]);
title(TITLE)
movegui(htask1subfun,'center');

clear V_finger_tapping_L_reslice Y_finger_tapping_L_reslice V_finger_tapping_R_res

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits')
cd(STANCEroot)
```

Creating pristine EPI data for RH task...
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session002.
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session002\EPI_BOLD_0001_002.nii
The maximum intensity of the simulated signal: 1145.8983



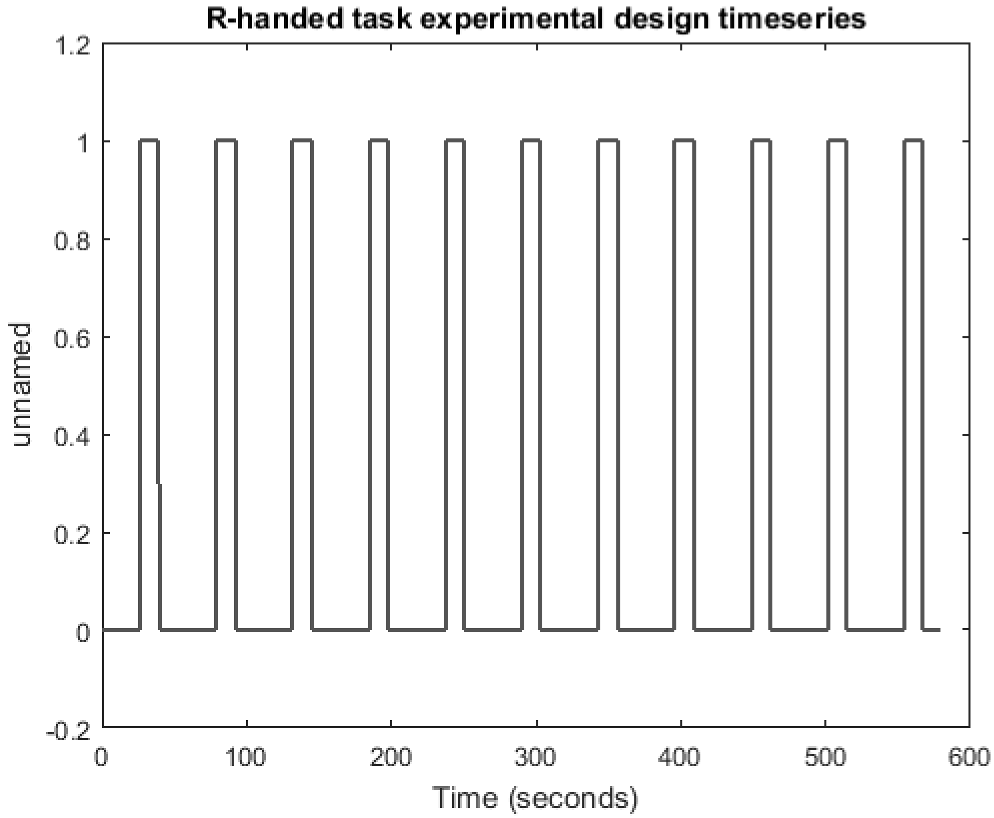
Design the 4D time-series

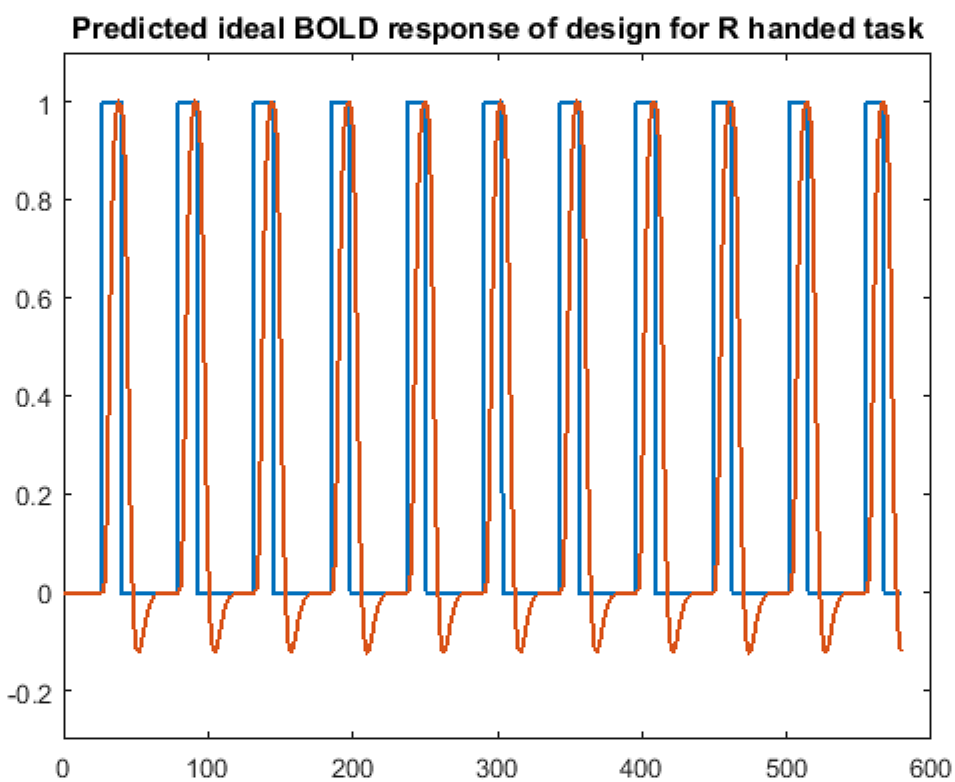
```
disp('Constructing the experiment design unto the 4D time-series ...')
uiwait(msgbox('Constructing experiment design and response.','4D time-
series'));
```

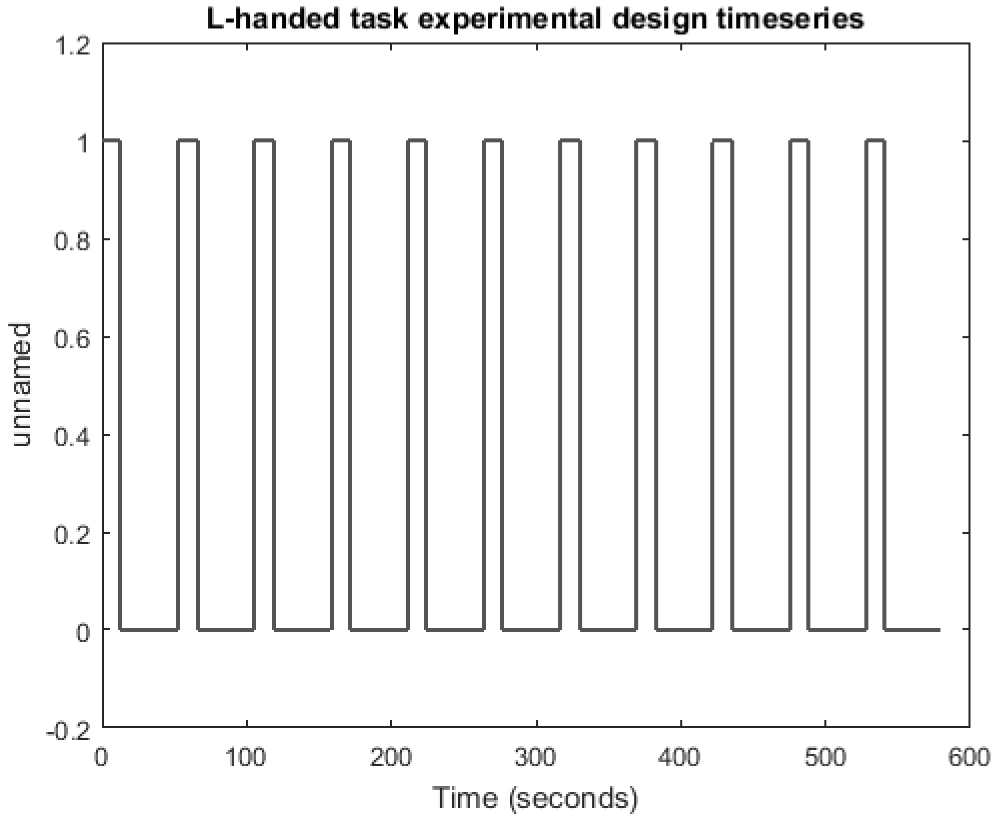
```
Now_sss = [5 1 3];
STANCE_new_session(5,1,3,true);

tic
Nslices = size(Y_T2star_Map,3);
TRsec = scan.TR/1000;
dt = TRsec/Nslices;
scantime = TRsec*scan.NV;
% start R-handed task on times: 26.4,79.2,132,187.8, ... [s]
exp_design_R = STANCE_blocked_design(dt, 26.4, 13, (2*13.4+13),
 scantime, true);
% start L-handed task on times: 0,52.8,105.6,158.4, ... [s]
exp_design_L = STANCE_blocked_design(dt, 0, 13, (2*13.4+13), scantime,
 false);
Nt = length(exp_design_R.Data);
times = dt*(1:Nt)';
NT = Nt/Nslices;


% save all of the elements common to the study
cd(STANCE_genpath(Now_sss,1))
save('STANCEstudy.mat','task_R','task_L','exp_design_R','exp_design_L')
cd(STANCEroot)


h_exp_designR = figure;
plot(exp_design_R,'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
title('R-handed task experimental design timeseries')
movegui(h_exp_designR,'west');
h_exp_designL = figure;
plot(exp_design_L,'LineWidth', 1.5,'Color',[0.33,0.33,0.33])
ylim([-0.2 1.2])
title('L-handed task experimental design timeseries')
movegui(h_exp_designL,'east');

% display ideal BOLD HRF response to the experimental design
hrf = spm_hrf(TRsec);
times2 = 0:0.25:32;
hrf_exact = spline(0:TRsec:32,hrf,times2);
% figure,
% h_hrf = plot(0:TRsec:32,hrf,'o',times2,hrf_exact,'LineWidth',2.0);
% xlim([0,32])
% title('BOLD canonical HRF')

toc

tic

BOLD_R_ts = STANCE_apply_response_function(dt,exp_design_R);
BOLD_L_ts = STANCE_apply_response_function(dt,exp_design_L);
baseline_ts = (1-BOLD_R_ts.Data-BOLD_L_ts.Data);
```

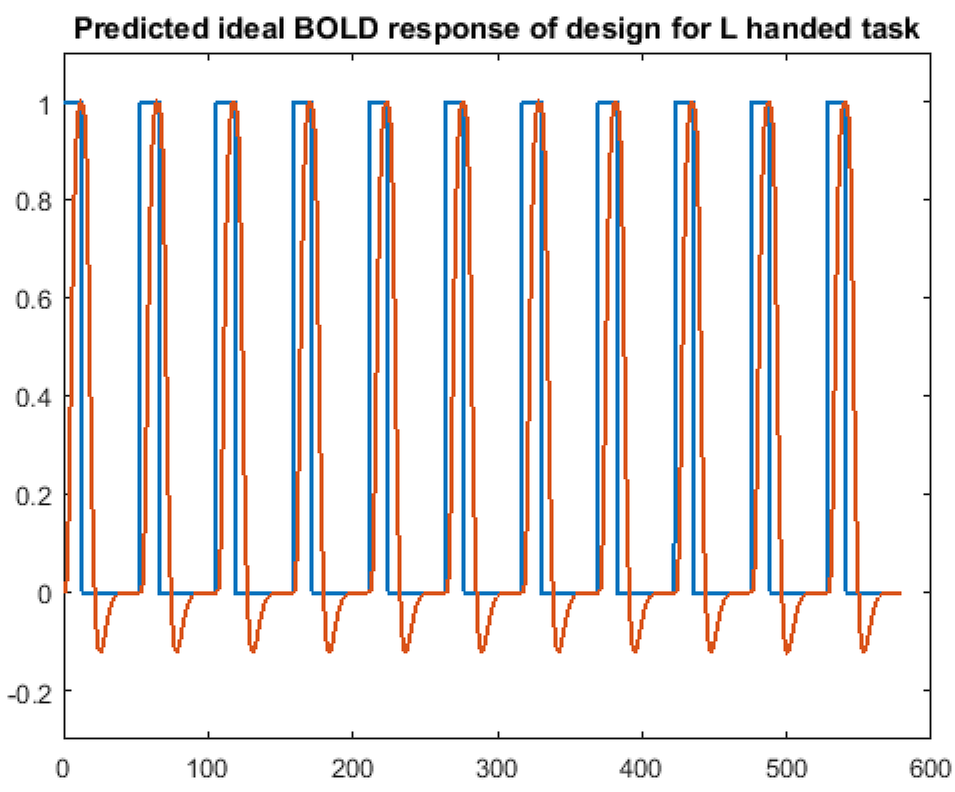
```
h_exp_designR2 = figure;
plot(times,exp_design_R.Data,times,BOLD_R_ts.Data,'LineWidth',1.5)
ylim([-0.3 1.1])
title('Predicted ideal BOLD response of design for R handed task')
movegui(h_exp_designR2,'west');
h_exp_designL2 = figure;
plot(times,exp_design_L.Data,times,BOLD_L_ts.Data,'LineWidth',1.5)
ylim([-0.3 1.1])
title('Predicted ideal BOLD response of design for L handed task')
movegui(h_exp_designL2,'east');


% apply subject character: on-task percentage
on_task_fraction = 0.99;
on_task = (rand(Nt,1)> (1-on_task_fraction));
subject_exp_design_R = exp_design_R;
subject_exp_design_R.Data = on_task.*subject_exp_design_R.Data;
subject_exp_design_L = exp_design_L;
subject_exp_design_L.Data = on_task.*subject_exp_design_L.Data;

clear exp_design_R exp_design_L;
```

```
Constructing the experiment design unto the 4D time-series ...
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session003.
Elapsed time is 1.994069 seconds.
o Applying the canonical hemodynamic response function to time-series.
o Applying the canonical hemodynamic response function to time-series.
```


R-handed task experimental design timeseries
unnamed
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600
Time (seconds)


Predicted ideal BOLD response of design for R handed task
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600


L-handed task experimental design timeseries
unnamed
Time (seconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600


Predicted ideal BOLD response of design for L handed task
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600

Construct the HRF maps to model HRF variability (proof-of-principle)

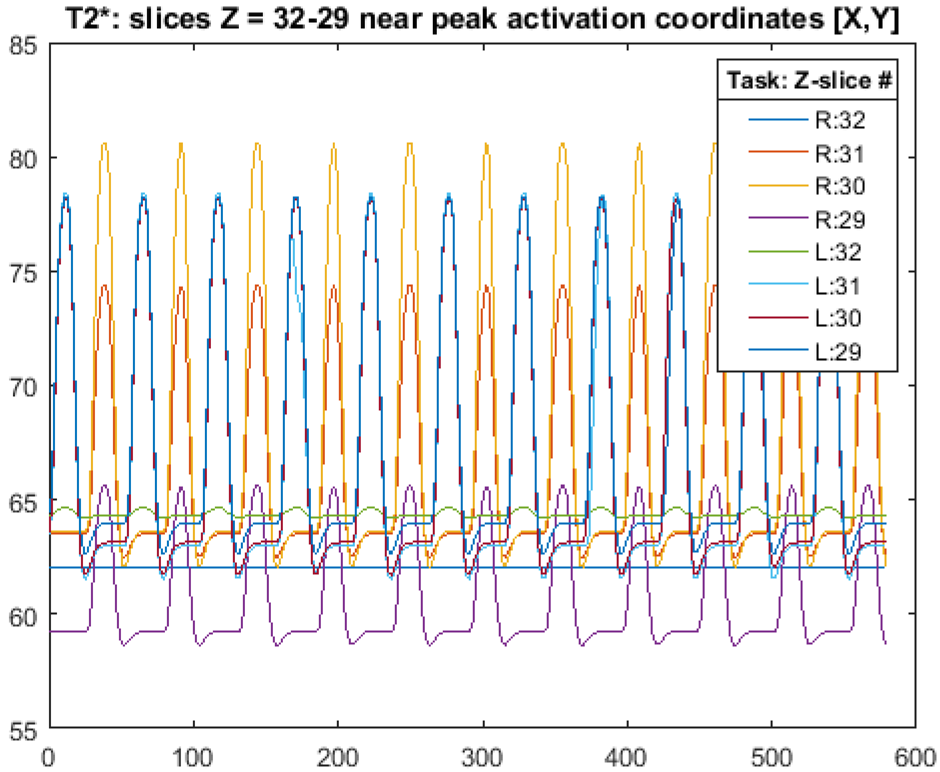
```
disp('Imposing HRF variability ...')
uiwait(msgbox('Constructing spatially varying HRF.','HRF
 variability'));

altas_fname = [STANCEroot,'/MNI/aal.nii.gz'];
[V_atlas,Y_atlas] = STANCE_load_volume(altas_fname);
% assign default (canonical) values for visual cortex
HRF_param.alpha1 = 6.0;
HRF_param.alpha2 = 16.0;
HRF_param.beta1  = 1.0;
HRF_param.beta2  = 1.0;
HRF_param.c      = (1/6.0);

% define HRF variability in regions near ROI based upon observed FWHM
% as detailed in Figure 2 for the FWHM map (TR = 2.5s @ 2T) in
% "Sensitivity of the resting state hemodynamic response function
% estimation to autonomic nervous system fluctuations"
% by Guo-Rong Wua & Daniele Marinazzo, who were accepted by
% Phil. Trans. R. Soc. A. (in press)
FWHMfactor = gamFWHM(HRF_param.alpha1);
beta1_map = 0.*Y_atlas + FWHMfactor/6.5; % default amplitude (visual
 cortex)
beta1_map(Y_atlas==0)  = 1.0;            % to "zero" out the effect of
 the beta exponent use 1.0 instead of 0.0
beta1_map(Y_atlas==1)  = FWHMfactor/6.3; % left precentral gyrus
 (primary motor cortex)
beta1_map(Y_atlas==2)  = FWHMfactor/6.3; % right precentral gyrus
 (primary motor cortex)
beta1_map(Y_atlas==7)  = FWHMfactor/6.4; % left frontal mid
beta1_map(Y_atlas==8)  = FWHMfactor/6.4; % right frontal mid
beta1_map(Y_atlas==19) = FWHMfactor/6.2; % left supplementary motor
 area
beta1_map(Y_atlas==20) = FWHMfactor/6.2; % right supplementary motor
 area
beta1_map(Y_atlas==57) = FWHMfactor/6.3; % left postcentral
beta1_map(Y_atlas==58) = FWHMfactor/6.3; % right postcentral
beta1_max = max(max(max(beta1_map)));
beta1_map = smooth3(beta1_map,'gaussian',7);
beta1_map = uint8(255*(beta1_map/beta1_max));

V_HRF_beta1 = V_atlas;
V_HRF_beta1.fname = [STANCE_genpath(Now_sss,1),'/HRF_beta1.nii'];
V_HRF_beta1 = spm_create_vol(V_HRF_beta1);
V_HRF_beta1 = spm_write_vol(V_HRF_beta1,beta1_map);

HRF_param.beta1 = [];
HRF_param.beta1.max = beta1_max;

% project beta1 map on to functional space
```

```
[V_HRF_beta1_reslice,Y_HRF_beta1_reslice] =
 STANCE_reslice_volume(V_HRF_beta1,scan,sliceLimits);
beta1_max = max(max(max(Y_HRF_beta1_reslice)));
Y_HRF_beta1_reslice = uint8(255*(Y_HRF_beta1_reslice/beta1_max));
Y_HRF_beta1_reslice(Y_HRF_beta1_reslice<26) = 255;
V_HRF_beta1_reslice =
 spm_write_vol(V_HRF_beta1_reslice,Y_HRF_beta1_reslice);

HRF_param.beta1.map = V_HRF_beta1_reslice.fname;
h_beta1 = figure;
imshow(255*rot90(Y_HRF_beta1_reslice(:,:,30)/beta1_max));
title('FWHM variation in the HRF near motor cortex')
movegui(h_beta1,'north');

clear('V_HRF_beta1', 'beta1_map', 'V_HRF_beta1_reslice','Y_HRF_beta1_reslice','V_a

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subject_exp_design_
cd(STANCEroot)

% calculate BOLD response with HRF variability
BOLD_R_ts =
 STANCE_apply_response_function(dt,subject_exp_design_R,'canonical',HRF_param);
BOLD_L_ts =
 STANCE_apply_response_function(dt,subject_exp_design_L,'canonical',HRF_param);
baseline_ts = (1-BOLD_R_ts.Data-BOLD_L_ts.Data);

% generate T2* baseline
T2star_4D =
 zeros(size(Y_T2star_Map,1),size(Y_T2star_Map,2),Nslices,NT);

sliceOrder = scan.order;
sliceTiming = make_slice_timing(sliceOrder,Nslices);

for t = 1:Nt
    ti = ceil(t/Nslices);
    STi = mod(t,Nslices);
    if STi == 0
        STi = Nslices;
    end
    T2star_4D(:,:,sliceTiming(STi),ti) = ...

 Y_T2star_Map(:,:,sliceTiming(STi)).*baseline_ts(:,:,sliceTiming(STi),ti) ...
        +
 Y_T2star_Map_Act_R(:,:,sliceTiming(STi)).*BOLD_R_ts.Data(:,:,sliceTiming(STi),ti)
        +
 Y_T2star_Map_Act_L(:,:,sliceTiming(STi)).*BOLD_L_ts.Data(:,:,sliceTiming(STi),ti)
end

Times = 1:TRsec:NT*TRsec;
% peak activation for R task near: X = 18, Y  = 32, Z = 30
% peak activation for L task near: X = 47, Y  = 32, Z = 30
h_T2star = figure;
```

```
plot(Times,squeeze(T2star_4D(18,32,32,:)),Times,squeeze(T2star_4D(18,32,31,:)),...

 Times,squeeze(T2star_4D(18,32,30,:)),Times,squeeze(T2star_4D(18,32,29,:)),...

 Times,squeeze(T2star_4D(46,32,32,:)),Times,squeeze(T2star_4D(46,32,31,:)),...

 Times,squeeze(T2star_4D(46,32,30,:)),Times,squeeze(T2star_4D(46,32,29,:)));
title('T2*: slices Z = 32-29 near peak activation coordinates [X,Y]')
lgd =
 legend('R:32','R:31','R:30','R:29','L:32','L:31','L:30','L:29','Location','best')
title(lgd,'Task: Z-slice #')
movegui(h_T2star,'northeast');
```

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

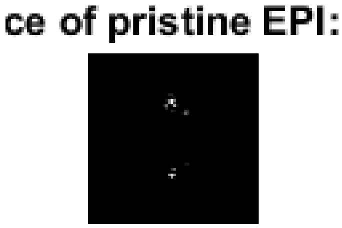
on in the HRF near

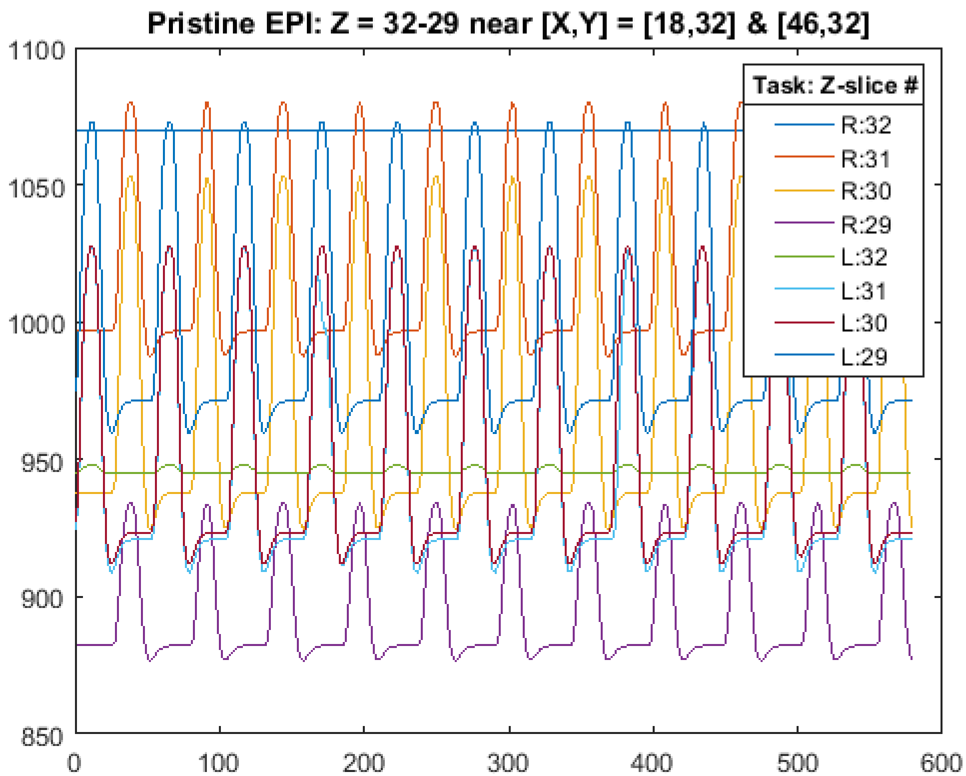


Generate the pristine EPI timeseries with no noise or motion

```
uiwait(msgbox('Generating pristine EPI 4D signal.','Pristine 4D
 data'));

display('o Generating pristine EPI 4D signal.')

[~,Y_EPI4D] = STANCE_EPI_signal(fn_fuzzy_reslice,T2star_4D,scan);

EPIvar = squeeze(var(Y_EPI4D,0,4));
h_EPIvar = figure;
imshow(EPIvar(:,:,30),[])
title('Variance of pristine EPI: Z = 30')
movegui(h_EPIvar,'center');
% NOTE: peak activation on LHS (RH task) observed at [18,32,30], var =
 110.1
%       peak activation on RHS (LH task) observed at [46,32,30], var =
 38.58

Times = 1:TRsec:NT*TRsec;
h_EPI0 = figure;
plot(Times,squeeze(Y_EPI4D(18,32,32,:)),Times,squeeze(Y_EPI4D(18,32,31,:)),...
```

```
 Times,squeeze(Y_EPI4D(18,32,30,:)),Times,squeeze(Y_EPI4D(18,32,29,:)),...

 Times,squeeze(Y_EPI4D(46,32,32,:)),Times,squeeze(Y_EPI4D(46,32,31,:)),...

 Times,squeeze(Y_EPI4D(46,32,30,:)),Times,squeeze(Y_EPI4D(46,32,29,:)));
title('Pristine EPI: Z = 32-29 near [X,Y] = [18,32] & [46,32]')
lgd =
 legend('R:32','R:31','R:30','R:29','L:32','L:31','L:30','L:29','Location','best')
title(lgd,'Task: Z-slice #')
movegui(h_EPI0,'east');

disp(['The temporal standard deviation of voxel [18,32,32]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,32,:))))]);
disp(['The temporal standard deviation of voxel [18,32,30]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,30,:))))]);
disp(['The temporal standard deviation of voxel [46,32,30]:
 ',num2str(std(squeeze(Y_EPI4D(46,32,30,:))))]);

disp(['Fractions of CSF, GM, and WM in voxel [18,32,32]:
 ',num2str([Y_reslice(18,32,32,2) Y_reslice(18,32,32,3)
 Y_reslice(18,32,32,4)])]);
disp(['Fractions of CSF, GM, and WM in voxel [18,32,30]:
 ',num2str([Y_reslice(18,32,30,2) Y_reslice(18,32,30,3)
 Y_reslice(18,32,30,4)])]);
disp(['Fractions of CSF, GM, and WM in voxel [46,32,30]:
 ',num2str([Y_reslice(46,32,30,2) Y_reslice(46,32,30,3)
 Y_reslice(46,32,30,4)]  )]);

clear('V_reslice','Y_reslice');

o Generating pristine EPI 4D signal.
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session003\EPI_BOLD_0001_003.nii
The maximum intensity of the simulated signal: 1145.8984
The temporal standard deviation of voxel [18,32,32]: 0.0038775
The temporal standard deviation of voxel [18,32,30]: 43.6924
The temporal standard deviation of voxel [46,32,30]: 40.0214
Fractions of CSF, GM, and WM in voxel [18,32,32]: 0.94464    0.050824
 0.00061645
Fractions of CSF, GM, and WM in voxel [18,32,30]: 0.029082      0.9036
    0.063405
Fractions of CSF, GM, and WM in voxel [46,32,30]: 0.00080444
 0.8806     0.11468
```

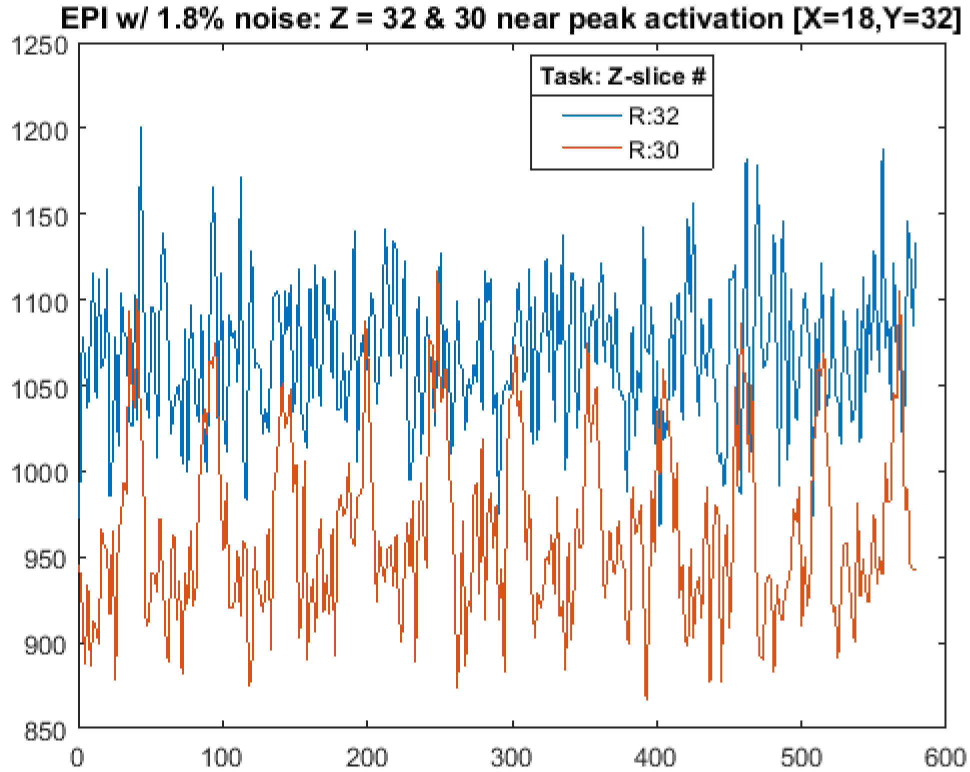




Add spatially varying system noise

```
uiwait(msgbox('Generating EPI 4D signal with system noise that varies
 with tissue type.','4D data + noise'));

display('Generating EPI 4D signal with system noise.')

Now_sss = [5 1 4];
STANCE_new_session(5,1,4,true);

% from data observe sigma ~= 1.8% in GM (apart from drift) see GM
 control: std = 22.6/ peak amp ~ 1250 ~= 1.8%
scan.noise              = 1.8;    % percent noise relative to peak in GM
 (convenient number chosen for further calculations)
```

```
noiseMap = STANCE_make_noise_map(fn_fuzzy_reslice,2,2);

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subject_exp_design_
cd(STANCEroot)

% EPI timeseries with noise, no attenuation
[~,Y_EPI4D] =
 STANCE_EPI_signal(fn_fuzzy_reslice,T2star_4D,scan,noiseMap,[],[],[],
[],s+2*Nt);

toc

Times = 1:TRsec:NT*TRsec;
h_EPIn = figure;
plot(Times,squeeze(Y_EPI4D(18,32,32,:)),Times,squeeze(Y_EPI4D(18,32,30,:)));
title('EPI w/ 1.8% noise: Z = 32 & 30 near peak activation
 [X=18,Y=32]')
lgdn = legend('R:32','R:30','Location','best');
title(lgdn,'Task: Z-slice #')
movegui(h_EPIn,'southeast');

disp(['The temporal standard deviation of voxel [18,32,32]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,32,:))))]);
disp(['The temporal standard deviation of voxel [18,32,30]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,30,:))))]);

Generating EPI 4D signal with system noise.
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session004.
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session004\EPI_BOLD_0001_004.nii
The maximum intensity of the simulated signal: 1403.728
Elapsed time is 907.116677 seconds.
The temporal standard deviation of voxel [18,32,32]: 42.2802
The temporal standard deviation of voxel [18,32,30]: 52.2635
```



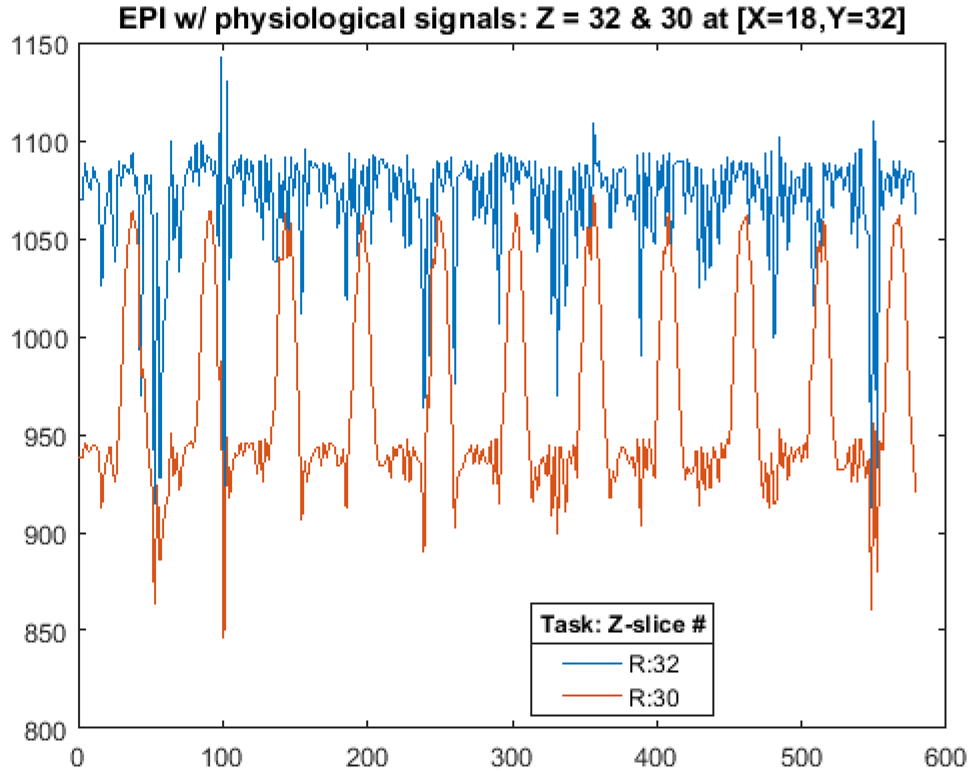
Generate and add physiological noise times-series

```
uiwait(msgbox('Generating EPI 4D signal with only physiological noise
 added.','4D data + physio'));

disp('Adding physiological noise to the times-series ...')

Now_sss = [5 1 5];
STANCE_new_session(5,1,5,true);

load('motion_parameters.mat')
% P(1)  - x translation
% P(2)  - y translation
% P(3)  - z translation
% P(4)  - x rotation about - {pitch} (radians)
% P(5)  - y rotation about - {roll}  (radians)
% P(6)  - z rotation about - {yaw}   (radians)
y = rpfmriblocks009(:,2);

[RTIave, RTIsigma] = STANCE_estimate_RTI(y,scan.TR/1000);

disp(['From y-motion estimate the RTI and its standard deviation to
 be: ',num2str([RTIave, RTIsigma])]);
```

```
% define physiological noise parameters
physio.weight                  = 75.0;    % [kg] weight
physio.lambdas                 = [0.009,0.006,0.02,0.05]; % lambda
 values for major tissue types according to the tSNR model
physio.respiratory.TI          = RTIave;   % [s]  average respiratory
 time interval
physio.respiratory.sigma       = RTIsigma; % [s]  standard deviation
 of "
physio.respiratory.A_z         = 1.0;      % [cm] average chest motion
 height
physio.respiratory.A_z_sigma   = 0.005;    % [cm] standard deviation
 of chest motion height
physio.respiratory.seed.impulse= s;        % the random number
 generator seed for respiratory impulse
physio.respiratory.seed.dz     = s;        % the random number
 generator seed for chest motion
physio.cardiac.TI              = 1.05;     % [s]  heart beat time
 interval
physio.cardiac.IPFM.freqs      = [0.02,0.1,NaN];    % [Hz] frequencies
 for Integral Pulse Frequency Modulation Model (IPFM)
physio.cardiac.IPFM.sigmas     = [0.2,0.2,NaN];     % standard
 deviations of rates in terms of fractional value of (1/f) for IPFM
physio.cardiac.IPFM.amplitudes = [1.0, 1.0, (2/3)]; % the sinusoid
 amplitudes  for IPFM
physio.cardiac.IPFM.seeds      = [[s+4*Nt,s+6*Nt], [s+8*Nt,s+10*Nt],
 NaN]; % the random number generator seeds for IPFM
physio.cardiac.PWV.seed        = s+12*Nt;  % the random number
 generator seed for the PWV simulator

% proof-of-principle: add lag-times to cardiac time-series
lags = 0*Y_GM;
[X,Y,Z] = meshgrid(1:size(Y_GM,1),1:size(Y_GM,2),1:size(Y_GM,3));
lags = round((X + Y + Z)/10); % using units of dt so max lag time
 ~0.87 s
physio.cardiac.lags = lags;

[physio_4D,~,~] =
 STANCE_physio_4D(fn_fuzzy_reslice,length(subject_exp_design_R.Data),scan,physio);

% EPI timeseries, with physiological noise (with lags included), no
 system (thermal) or attenuation
scan.noise            = 0.0;    % percent noise relative to peak

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subject_exp_design_
cd(STANCEroot)

[~,Y_EPI4D] = STANCE_EPI_signal(fn_fuzzy_reslice,T2star_4D,scan,
[],physio_4D);
h_EPIp = figure;
plot(Times,squeeze(Y_EPI4D(18,32,32,:)),Times,squeeze(Y_EPI4D(18,32,30,:)));
```

```
title('EPI w/ physiological signals: Z = 32 & 30 at [X=18,Y=32]')
lgdn = legend('R:32','R:30','Location','best');
title(lgdn,'Task: Z-slice #')
movegui(h_EPIp,'south');

disp(['The temporal standard deviation of voxel [18,32,32]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,32,:))))]);
disp(['The temporal standard deviation of voxel [18,32,30]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,30,:))))]);
```

```
Adding physiological noise to the times-series ...
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session005.
From y-motion estimate the RTI and its standard deviation to be:
 4.4256     0.18434
o Generating the respiratory volume time-series (RVT).
o Generating the respiratory pulse (RP) time-series.
o Generating the respiratory response (RR) time-series.
o Applying the respiratory response function to time-series.
o Generating the VLF heart-rate modulation time-series.
o Generating Mayer wave blood pressure (BP) variation time-series.
o Generating the instantaneous heart rate (HR) time-series from a
 generalized IPFM model.
o Generating the cardiac response (CR) time-series from HR.
o Applying the cardiac response function to time-series.
o Generating the cardiac event impulse (CEI) time-series.
o Generating the pulse wave velocity (PWV) time-series.
o Generating the cardiac pulse (CP) time-series.
o Generating the interaction of the cardiac and respiratory pulses
 (InterCRP) time-series.
o Generating the physiological noise per tissue type time-series.
o Adding lag times ...
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session005\EPI_BOLD_0001_005.nii
The maximum intensity of the simulated signal: 1152.8636
The temporal standard deviation of voxel [18,32,32]: 31.3686
The temporal standard deviation of voxel [18,32,30]: 47.6215
```

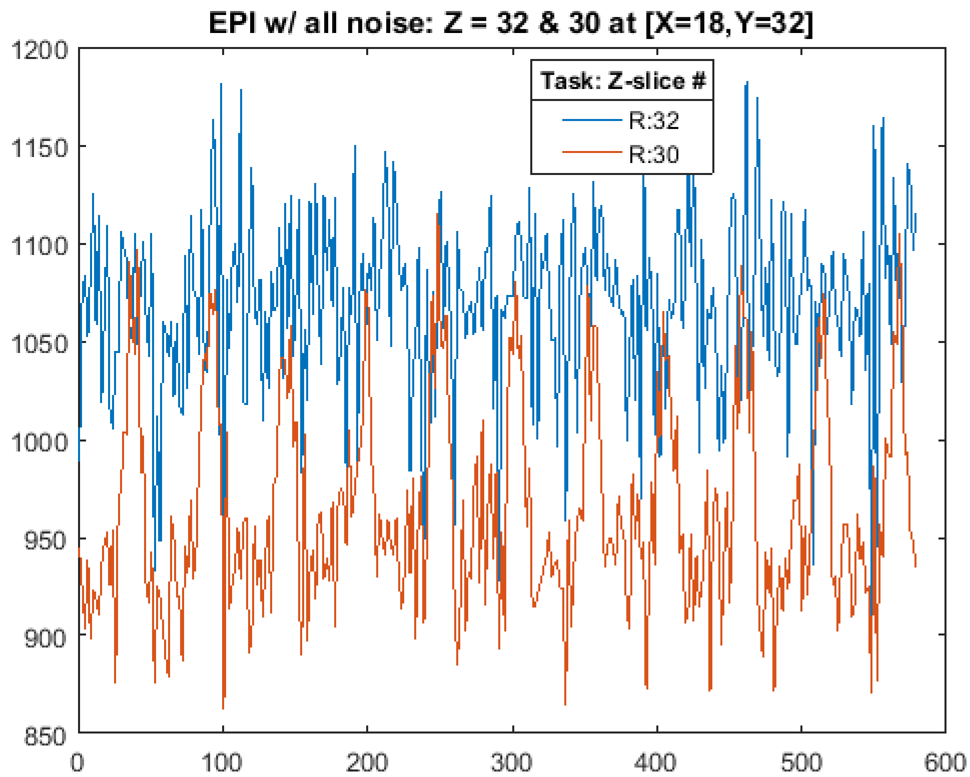


EPI timeseries, with physiological noise and system (thermal) noise (no attenuation)

```
uiwait(msgbox('Generating EPI 4D signal with all noise added.','4D
 data + all noise'));

display('Generating EPI 4D signal with all noise.')

Now_sss = [5 1 6];
STANCE_new_session(5,1,6,true);

% NOTE: according to the tSNR model with lambda = 0.9% in GM, then the
% residual system noise = sqrt(0.018^2 - ~0.01^2) ~ 0.015 = 1.5%
scan.noise            = 1.5;    % percent noise relative to peak

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subject_exp_design_
cd(STANCEroot)

[~,Y_EPI4D] =
 STANCE_EPI_signal(fn_fuzzy_reslice,T2star_4D,scan,noiseMap,physio_4D,
[],[],[],s+2*Nt);
```

```
h_EPIalln = figure;
plot(Times,squeeze(Y_EPI4D(18,32,32,:)),Times,squeeze(Y_EPI4D(18,32,30,:)));
title('EPI w/ all noise: Z = 32 & 30 at [X=18,Y=32]')
lgdn = legend('R:32','R:30','Location','best');
title(lgdn,'Task: Z-slice #')
movegui(h_EPIalln,'southwest');

disp(['The temporal standard deviation of voxel [18,32,32]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,32,:))))]);
disp(['The temporal standard deviation of voxel [18,32,30]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,30,:))))]);
```

```
Generating EPI 4D signal with all noise.
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session006.
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session006\EPI_BOLD_0001_006.nii
The maximum intensity of the simulated signal: 1352.1358
The temporal standard deviation of voxel [18,32,32]: 47.284
The temporal standard deviation of voxel [18,32,30]: 52.7164
```



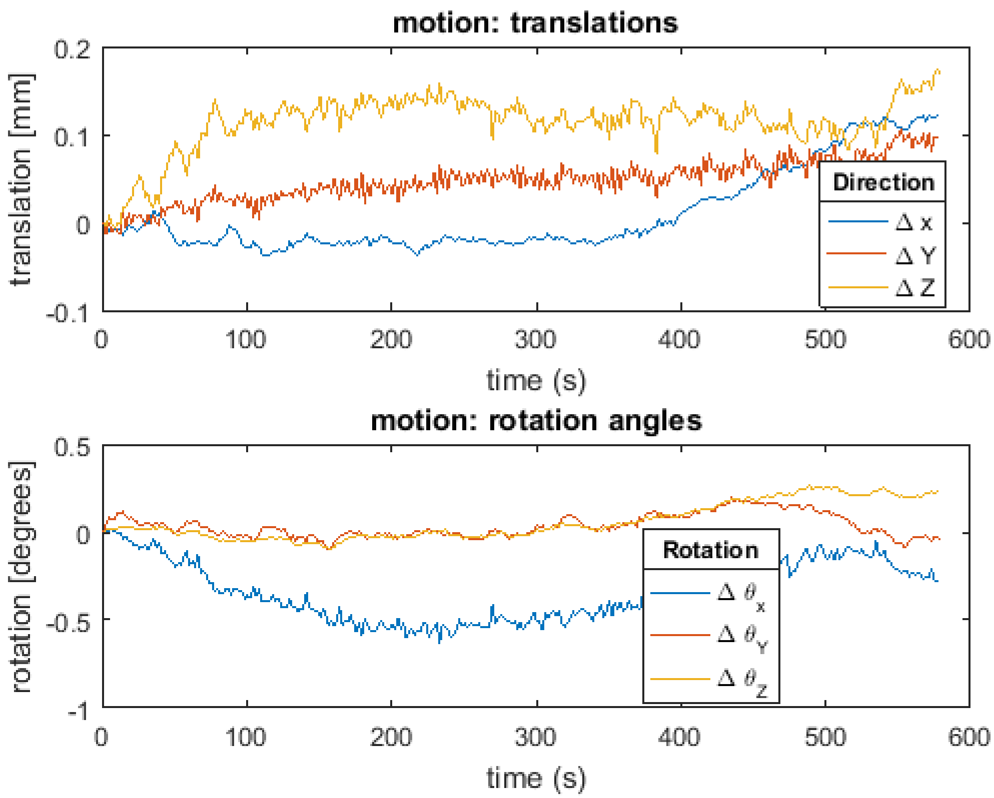
Load and add motion to EPI signal

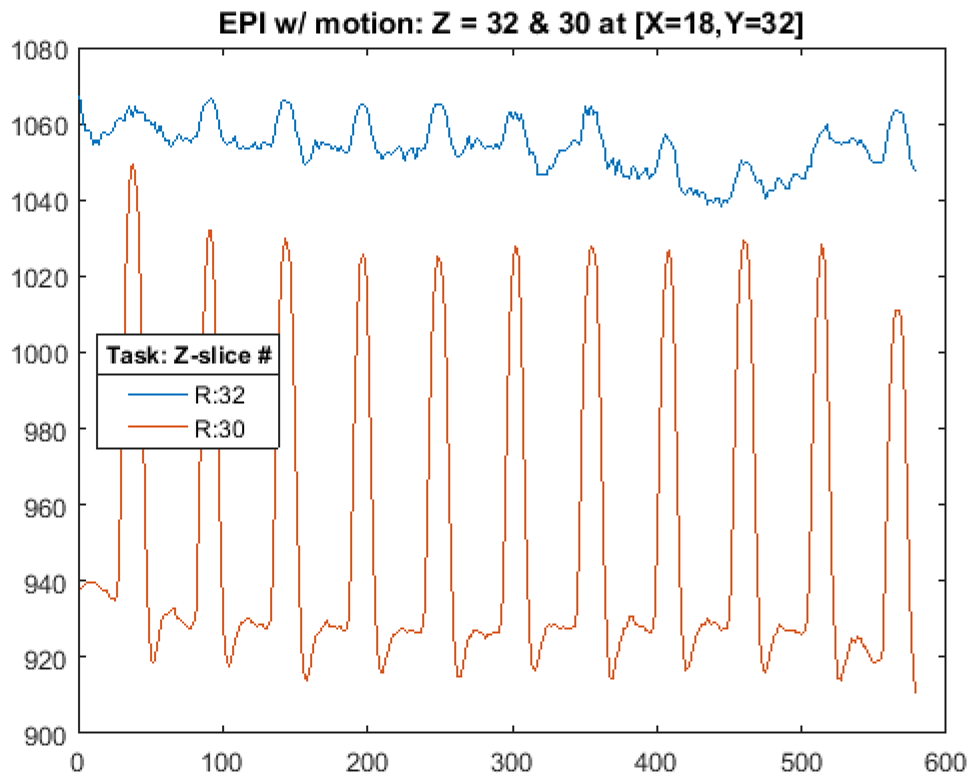
```
uiwait(msgbox('Generating EPI 4D signal with only motion added.','4D
 data + motion'));

display('Generating EPI 4D signal with motion added.')
```

```
Now_sss = [5 1 7];
STANCE_new_session(5,1,7,true);

motion = rpfmriblocks009;
% x_translation = rpfmriblocks009(:,1)/scan.voxel.size(1);
% y_translation = rpfmriblocks009(:,2)/scan.voxel.size(2);
% z_translation = rpfmriblocks009(:,3)/scan.voxel.size(3);
% pitch_rotation = rpfmriblocks009(:,4)*180.0/pi;
% roll_rotation  = rpfmriblocks009(:,5)*180.0/pi;
% yaw_rotation   = rpfmriblocks009(:,6)*180.0/pi;
for i = 1:3
    motion(:,i) = motion(:,i)/scan.voxel.size(i);
end
motion(:,4:6) = motion(:,4:6)*180.0/pi;

h_motion = figure;
subplot(2,1,1)
plot(Times,squeeze(motion(:,1)),Times,squeeze(motion(:,2)),Times,squeeze(motion(:,
title('motion: translations')
xlabel('time (s)')
ylabel('translation [mm]')
lgdnD = legend('\Delta x','\Delta Y','\Delta Z','Location','best');
title(lgdnD,'Direction')
subplot(2,1,2)
plot(Times,squeeze(motion(:,4)),Times,squeeze(motion(:,5)),Times,squeeze(motion(:,
title('motion: rotation angles')
xlabel('time (s)')
ylabel('rotation [degrees]')
lgdnR = legend('\Delta \theta_x','\Delta \theta_Y','\Delta
 \theta_Z','Location','best');
title(lgdnR,'Rotation')

scan.noise            = 0.0;    % percent noise relative to peak

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subject_exp_design_
cd(STANCEroot)

tic
[V_EPI4D,Y_EPI4D] = STANCE_EPI_signal(fn_fuzzy_reslice,T2star_4D,scan,
[],[],motion);
toc

h_EPIm = figure;
plot(Times,squeeze(Y_EPI4D(18,32,32,:)),Times,squeeze(Y_EPI4D(18,32,30,:)));
title('EPI w/ motion: Z = 32 & 30 at [X=18,Y=32]')
lgdn = legend('R:32','R:30','Location','best');
title(lgdn,'Task: Z-slice #')
movegui(h_EPIm,'west');

disp(['The temporal variance of voxel [18,32,32]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,32,:))))]);
```

```
disp(['The temporal variance of voxel [18,32,30]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,30,:))))]);
```

```
Generating EPI 4D signal with motion added.
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session007.
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session007\EPI_BOLD_0001_007.nii
The maximum intensity of the simulated signal: 1145.7194
Elapsed time is 80.940760 seconds.
The temporal variance of voxel [18,32,32]: 6.5081
The temporal variance of voxel [18,32,30]: 38.177
```



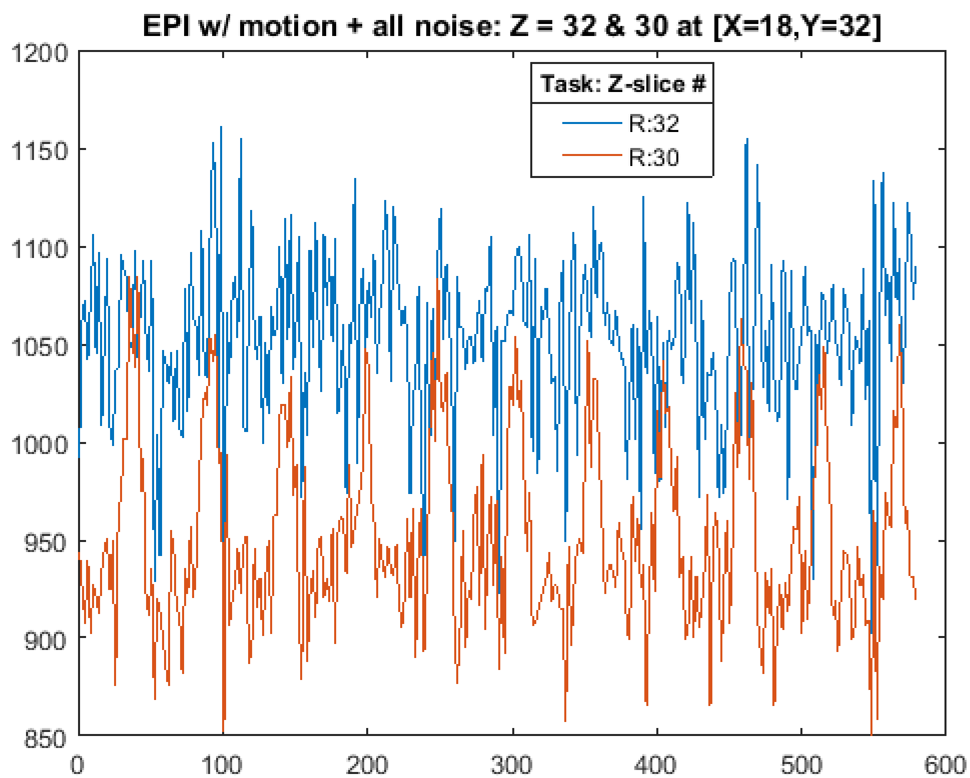


Generate EPI with motion and all types of noise

```
uiwait(msgbox('Generating EPI 4D signal with motion and all noise
 added.','4D data + motion + all noise'));

display('Generating EPI 4D signal with motion and all noise.')

Now_sss = [5 1 8];
STANCE_new_session(5,1,8,true);

% Since from the motion only time-series above:
 std(Y_EPI4D(18,32,32,:))/1250 = 0.52%
% Then out of an apparent residual system noise of 1.5%,
% the "true" system noise can be estimated to be ~1.4%

scan.noise            = 1.4;    % percent noise relative to peak

% save all of the elements common to the session's scan
cd(STANCE_genpath(Now_sss))
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subject_exp_design_
cd(STANCEroot)

tic
```

```
[~,Y_EPI4D] =
 STANCE_EPI_signal(fn_fuzzy_reslice,T2star_4D,scan,noiseMap,physio_4D,motion,
[],[],s+2*Nt);
toc

h_EPIm = figure;
plot(Times,squeeze(Y_EPI4D(18,32,32,:)),Times,squeeze(Y_EPI4D(18,32,30,:)));
title('EPI w/ motion + all noise: Z = 32 & 30 at [X=18,Y=32]')
lgdn = legend('R:32','R:30','Location','best');
title(lgdn,'Task: Z-slice #')
movegui(h_EPIm,'northwest');

disp(['The temporal standard deviation of voxel [18,32,32]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,32,:))))]);
disp(['The temporal standard deviation of voxel [18,32,30]:
 ',num2str(std(squeeze(Y_EPI4D(18,32,30,:))))]);
```

```
Generating EPI 4D signal with motion and all noise.
Erasing files in C:\spm\STANCE/fMRI/study005/subject0001/session008.
o Writing C:\spm\STANCE/fMRI/study005/subject0001/
session008\EPI_BOLD_0001_008.nii
The maximum intensity of the simulated signal: 1331.739
Elapsed time is 52.131146 seconds.
The temporal standard deviation of voxel [18,32,32]: 44.6188
The temporal standard deviation of voxel [18,32,30]: 47.1908
```



Compare simulated data with real data

```
uiwait(msgbox('Comparing simulated 4D data with real data.','Compare
 4D data'));

disp('------------------------------------')
disp('Comparing simulated with real data.')
disp('------------------------------------')

% NOTE: In real data peak activation is at
% L side activation is observed at [19,27,30] with baseline ~660 and
 max 769.4: 109.4
% R side activation is observed at [46,32,30] with baseline ~530 and
 max 638.7: 108.7
% A control gray matter voxel with no activation due to the task at
 [38,7,19]

load('simblock.mat')
LHSpeakSim = squeeze(Y_EPI4D(18,32,30,:));
RHSpeakSim = squeeze(Y_EPI4D(46,32,30,:));

h_RHS = figure;
subplot(3,1,1)
plot(Times,RHSpeak)
xlim([0 Times(end)])
ylim([350 750])
xlabel('time (s)')
ylabel('intensity')
title('Peak activation time-series on the RHS (real data)')
subplot(3,1,2)
plot(Times,RHSpeakSim)
xlim([0 Times(end)])
ylim([750 1150])
xlabel('time (s)')
ylabel('intensity')
title('Peak activation time-series on the RHS (simulated)')
subplot(3,1,3)
plot(Times,GMcontrol)
xlim([0 Times(end)])
ylim([750 1150])
xlabel('time (s)')
ylabel('intensity')
title('GM control time-series w/ no activation (real data)')
movegui(h_RHS,'northeast');

disp(['Mean and standard deviation of real data (RHS peak activation):
 ',num2str([mean(RHSpeak) std(RHSpeak)])]);
disp(['Mean and standard deviation of simulated data (RHS peak
 activation): ',num2str([mean(RHSpeakSim) std(RHSpeakSim)])]);
disp(['Mean and standard deviation of real data (GM control):
 ',num2str([mean(GMcontrol) std(GMcontrol)])]);
```

```
h_LHS = figure;
subplot(3,1,1)
plot(Times,LHSpeak)
xlim([0 Times(end)])
ylim([500 850])
xlabel('time (s)')
ylabel('intensity')
title('Peak activation time-series on the LHS (real data)')
subplot(3,1,2)
plot(Times,LHSpeakSim)
ylim([800 1150])
xlim([0 Times(end)])
xlabel('time (s)')
ylabel('intensity')
title('Peak activation time-series on the LHS (simulated)')
subplot(3,1,3)
plot(Times,GMcontrol)
xlim([0 Times(end)])
ylim([800 1150])
xlabel('time (s)')
ylabel('intensity')
title('GM control time-series w/ no activation (real data)')
movegui(h_LHS,'southeast');

% statistic of left-hand side peak activation (Right-hand task)
disp(['Mean and standard deviation of real data (LHS peak activation):
 ',num2str([mean(LHSpeak) std(LHSpeak)])]);
disp(['Mean and standard deviation of simulated data (LHS peak
 activation): ',num2str([mean(LHSpeakSim) std(LHSpeakSim)])]);
disp(['Mean and standard deviation of real data (GM control):
 ',num2str([mean(GMcontrol) std(GMcontrol)])]);


LHSpeak_block_ave = (LHSpeak(1:27) + LHSpeak(27+(1:27)) +
 LHSpeak(55+(1:27)) ...
                   + LHSpeak(82+(1:27)) + LHSpeak(110+(1:27)) +
 LHSpeak(137+(1:27)) ...
                   + LHSpeak(165+(1:27)) + LHSpeak(192+(1:27)) +
 LHSpeak(220+(1:27))...
                   + LHSpeak(247+(1:27))+ LHSpeak(275+(1:27)))/11.0;
LHSpeakSim_block_ave = (LHSpeakSim(1:27) + LHSpeakSim(27+(1:27)) +
 LHSpeakSim(55+(1:27)) ...
                   + LHSpeakSim(82+(1:27)) + LHSpeakSim(110+(1:27)) +
 LHSpeakSim(137+(1:27)) ...
                   + LHSpeakSim(165+(1:27)) + LHSpeakSim(192+(1:27)) +
 LHSpeakSim(220+(1:27))...
                   + LHSpeakSim(247+(1:27)) +
 LHSpeakSim(275+(1:27)))/11.0;
RHSpeak_block_ave = (RHSpeak(1:27) + RHSpeak(27+(1:27)) +
 RHSpeak(55+(1:27)) ...
                   + RHSpeak(82+(1:27)) + RHSpeak(110+(1:27)) +
 RHSpeak(137+(1:27))...
                   + RHSpeak(165+(1:27)) + RHSpeak(192+(1:27)) +
 RHSpeak(220+(1:27)) ...
```

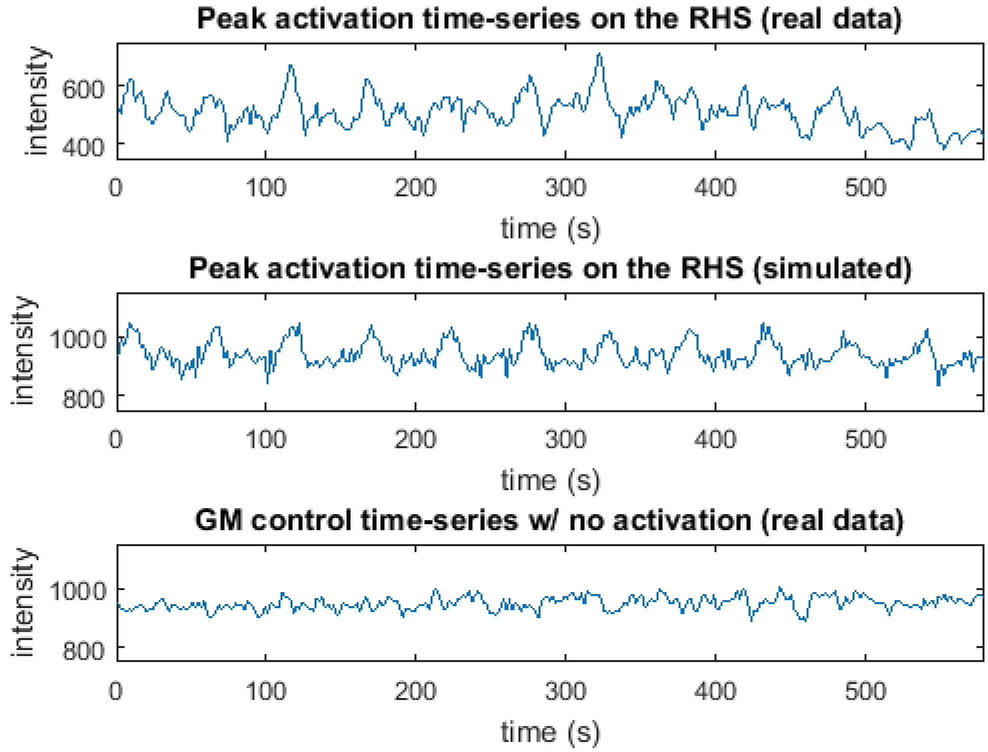
```
                     + RHSpeak(247+(1:27)) + RHSpeak(275+(1:27)))/11.0;
RHSpeakSim_block_ave = (RHSpeakSim(1:27) + RHSpeakSim(27+(1:27)) +
 RHSpeakSim(55+(1:27)) ...
                     + RHSpeakSim(82+(1:27)) + RHSpeakSim(110+(1:27)) +
 RHSpeakSim(137+(1:27))...
                     + RHSpeakSim(165+(1:27)) + RHSpeakSim(192+(1:27)) +
 RHSpeakSim(220+(1:27)) ...
                     + RHSpeakSim(247+(1:27)) +
 RHSpeakSim(275+(1:27)))/11.0;
GMcontrol_block_ave = (GMcontrol(1:27) + GMcontrol(27+(1:27)) +
 GMcontrol(55+(1:27)) ...
                     + GMcontrol(82+(1:27)) + GMcontrol(110+(1:27)) +
 GMcontrol(137+(1:27))...
                     + GMcontrol(165+(1:27)) + GMcontrol(192+(1:27)) +
 GMcontrol(220+(1:27)) ...
                     + GMcontrol(247+(1:27)) +
 GMcontrol(275+(1:27)))/11.0;


RHSpeak_block_extent = max(RHSpeak_block_ave) -
 min(RHSpeak_block_ave);
disp(['Difference between max and min, RHS peak activation(real data):
 ',num2str(RHSpeak_block_extent)])
RHSpeakSim_block_extent = max(RHSpeakSim_block_ave) -
 min(RHSpeakSim_block_ave);
disp(['Difference between max and min, RHS peak activation(simulated):
 ',num2str(RHSpeakSim_block_extent)])
RHSpeak_block_diff = (RHSpeak_block_ave(15:26) -
 RHSpeak_block_ave(2:13))/12.0;
RHSpeak_block_diff_sum = -sum(RHSpeak_block_diff);
disp(['Mean difference between 1st & 2nd half of block (RHS peak in
 real data): ',num2str(RHSpeak_block_diff_sum )])
RHSpeakSim_block_diff = (RHSpeakSim_block_ave(15:26) -
 RHSpeakSim_block_ave(2:13))/12.0;
RHSpeakSim_block_diff_sum = -sum(RHSpeakSim_block_diff);
disp(['Mean difference between 1st & 2nd half of block (RHS peak in
 simulated): ',num2str(RHSpeakSim_block_diff_sum )])

RHSbaseline_offset = (mean(RHSpeak_block_ave(15:26))-
min(RHSpeak_block_ave(15:26)))-(mean(RHSpeakSim_block_ave(15:26))-
min(RHSpeakSim_block_ave(15:26)));

RHSpeak_block_height = max(RHSpeak_block_ave) -
 mean(RHSpeak_block_ave(15:26));
disp(['Difference between max and baseline, RHS peak activation(real
 data): ',num2str(RHSpeak_block_height)])
RHSpeakSim_block_height = max(RHSpeakSim_block_ave) -
 mean(RHSpeakSim_block_ave(15:26));
disp(['Difference between max and baseline, RHS peak
 activation(simulated): ',num2str(RHSpeakSim_block_height)])

LHSpeak_block_extent = max(LHSpeak_block_ave) -
 min(LHSpeak_block_ave);
```

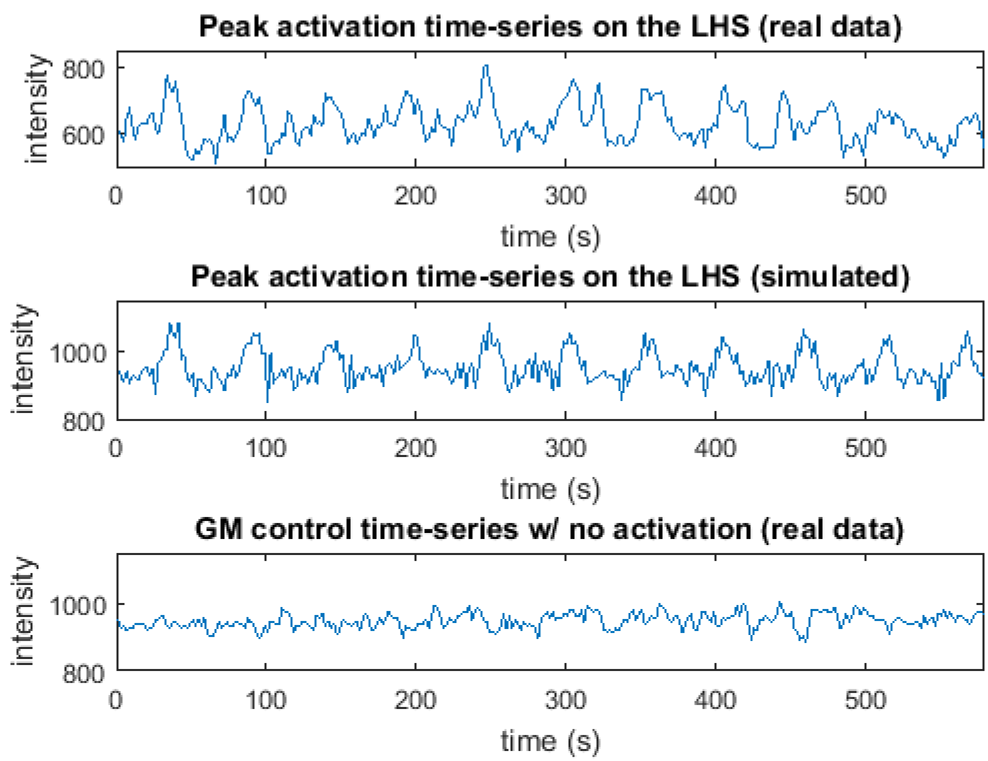
```
disp(['Difference between max and min, LHS peak activation(real data):
 ',num2str(LHSpeak_block_extent)])
LHSpeakSim_block_extent = max(LHSpeakSim_block_ave) -
 min(LHSpeakSim_block_ave);
disp(['Difference between max and min, LHS peak activation(simulated):
 ',num2str(LHSpeakSim_block_extent)])
LHSpeak_block_diff = (LHSpeak_block_ave(15:26) -
 LHSpeak_block_ave(2:13))/12.0;
LHSpeak_block_diff_sum = sum(LHSpeak_block_diff);
disp(['Mean difference between 2nd & 1st half of block (LHS peak in
 real data): ',num2str(LHSpeak_block_diff_sum )])
LHSpeakSim_block_diff = (LHSpeakSim_block_ave(15:26) -
 LHSpeakSim_block_ave(2:13))/12.0;
LHSpeakSim_block_diff_sum = sum(LHSpeakSim_block_diff);
disp(['Mean difference between 2nd & 1st half of block (LHS peak in
 simulated): ',num2str(LHSpeakSim_block_diff_sum)])

LHSbaseline_offset = (mean(LHSpeak_block_ave(2:13))-
min(LHSpeak_block_ave(2:13)))-(mean(LHSpeakSim_block_ave(2:13))-
min(LHSpeakSim_block_ave(2:13)));

LHSpeak_block_height = max(LHSpeak_block_ave) -
 mean(LHSpeak_block_ave(2:13));
disp(['Difference between max and baseline, LHS peak activation(real
 data): ',num2str(LHSpeak_block_height)])
LHSpeakSim_block_height = max(LHSpeakSim_block_ave) -
 mean(LHSpeakSim_block_ave(2:13));
disp(['Difference between max and baseline, LHS peak
 activation(simulated): ',num2str(LHSpeakSim_block_height)])

GMcontrol_block_extent = max(GMcontrol_block_ave) -
 min(GMcontrol_block_ave);
disp(['Difference between max and min, GM control (real data):
 ',num2str(GMcontrol_block_extent)])
GMcontrol_block_diff = (GMcontrol_block_ave(15:26) -
 GMcontrol_block_ave(2:13))/12.0;
GMcontrol_block_diff_sum = -sum(GMcontrol_block_diff);
disp(['Mean difference between 1st & 2nd half of block (GM control in
 real data): ',num2str(GMcontrol_block_diff_sum )])


% show plots of time-series averaged over all 11 blocks
h_RHS_ave = figure;
subplot(1,3,1)
plot(Times(1:27),RHSpeak_block_ave - min(RHSpeak_block_ave))
xlim([0 Times(27)])
ylim([0 120])
xlabel('time (s)')
ylabel('intensity')
title('RHS block (real data)')
subplot(1,3,2)
plot(Times(1:27),RHSpeakSim_block_ave - min(RHSpeakSim_block_ave) +
 RHSbaseline_offset)
xlim([0 Times(27)])
```

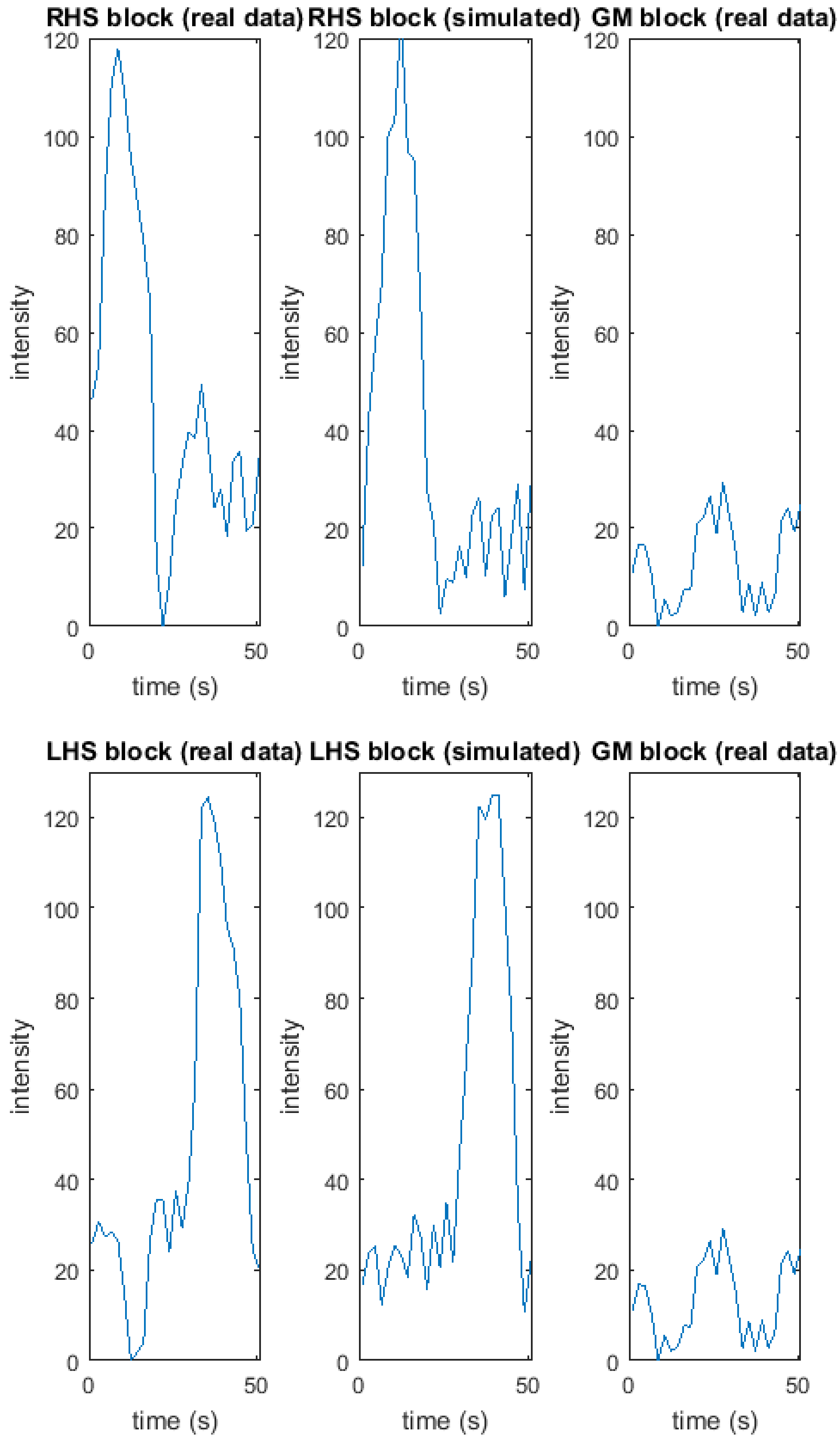
```
ylim([0 120])
xlabel('time (s)')
ylabel('intensity')
title('RHS block (simulated)')
subplot(1,3,3)
plot(Times(1:27),GMcontrol_block_ave - min(GMcontrol_block_ave))
xlim([0 Times(27)])
ylim([0 120])
xlabel('time (s)')
ylabel('intensity')
title('GM block (real data)')
movegui(h_RHS_ave,'southwest');

% show plots of time-series averaged over all 11 blocks
h_LHS_ave = figure;
subplot(1,3,1)
plot(Times(1:27),LHSpeak_block_ave - min(LHSpeak_block_ave))
xlim([0 Times(27)])
ylim([0 130])
xlabel('time (s)')
ylabel('intensity')
title('LHS block (real data)')
subplot(1,3,2)
plot(Times(1:27),LHSpeakSim_block_ave - min(LHSpeakSim_block_ave) +
 LHSbaseline_offset)
xlim([0 Times(27)])
ylim([0 130])
xlabel('time (s)')
ylabel('intensity')
title('LHS block (simulated)')
subplot(1,3,3)
plot(Times(1:27),GMcontrol_block_ave - min(GMcontrol_block_ave))
xlim([0 Times(27)])
ylim([0 130])
xlabel('time (s)')
ylabel('intensity')
title('GM block (real data)')
movegui(h_LHS_ave,'northwest');
```

```
------------------------------------
Comparing simulated with real data.
------------------------------------
Mean and standard deviation of real data (RHS peak activation):
 510.0232      57.30697
Mean and standard deviation of simulated data (RHS peak activation):
 938.81      42.45682
Mean and standard deviation of real data (GM control): 948.947
 22.59135
Mean and standard deviation of real data (LHS peak activation):
 629.0265       56.9497
Mean and standard deviation of simulated data (LHS peak activation):
 950.5359      47.19079
Mean and standard deviation of real data (GM control): 948.947
 22.59135
```

```
Difference between max and min, RHS peak activation(real data):
 117.9091
Difference between max and min, RHS peak activation(simulated):
 121.6364
Mean difference between 1st & 2nd half of block (RHS peak in real
 data): 37.5152
Mean difference between 1st & 2nd half of block (RHS peak in
 simulated): 50.3905
Difference between max and baseline, RHS peak activation(real data):
 86.3258
Difference between max and baseline, RHS peak activation(simulated):
 107.2905
Difference between max and min, LHS peak activation(real data):
 124.4545
Difference between max and min, LHS peak activation(simulated):
 114.2728
Mean difference between 2nd & 1st half of block (LHS peak in real
 data): 57.4394
Mean difference between 2nd & 1st half of block (LHS peak in
 simulated): 54.971
Difference between max and baseline, LHS peak activation(real data):
 103.1742
Difference between max and baseline, LHS peak activation(simulated):
 102.0499
Difference between max and min, GM control (real data): 29.2727
Mean difference between 1st & 2nd half of block (GM control in real
 data): -2.0227
```


Peak activation time-series on the RHS (real data)
intensity
600
400
0
100
200
300
400
500
time (s)
Peak activation time-series on the RHS (simulated)
intensity
1000
800
0
100
200
300
400
500
time (s)
GM control time-series w/ no activation (real data)
intensity
1000
800
0
100
200
300
400
500
time (s)


Peak activation time-series on the LHS (real data)
intensity
800
600
0
100
200
300
400
500
time (s)
Peak activation time-series on the LHS (simulated)
intensity
1000
800
0
100
200
300
400
500
time (s)
GM control time-series w/ no activation (real data)
intensity
1000
800
0
100
200
300
400
500
time (s)


RHS block (real data)
RHS block (simulated)
GM block (real data)
LHS block (real data)
LHS block (simulated)
GM block (real data)
intensity
time (s)
0
20
40
60
80
100
120
0
50

Save results, free up memory and return

```
clear task_R;
clear task_L;

else
    save('simulations.mat','simulations')
end
cd(currentDir)
```

```
Reslicing volumes to functional space...
```

Published with MATLAB® R2016a