Table of Contents

Spontaneous and Task-related Activation of Neuronally Correlated Events (STANCE) 1
Turn off warnings... 2
Select subject by index (originally there are 20 subjects to choose from) 2
Build activation regions by modelling reported results 4
Reslice to funtional space according to the fMRI scan protocol 8
Generate the EPI baseline 11
Generate the pristine EPI signal 12
Design the 4D time-series 14
Generate the pristine EPI timeseries with no noise or motion 19
Add spatially varying system noise 20
Add physiological noise times-series 21

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

Illustates the simulation of a 4D fMRI data set from specifying the experimental design and 3D activation maps for corresponding tasks.

author: Dr. Jason E. Hill (post-doc fellow with CNT at TTU) demo_4D_expdesign.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...

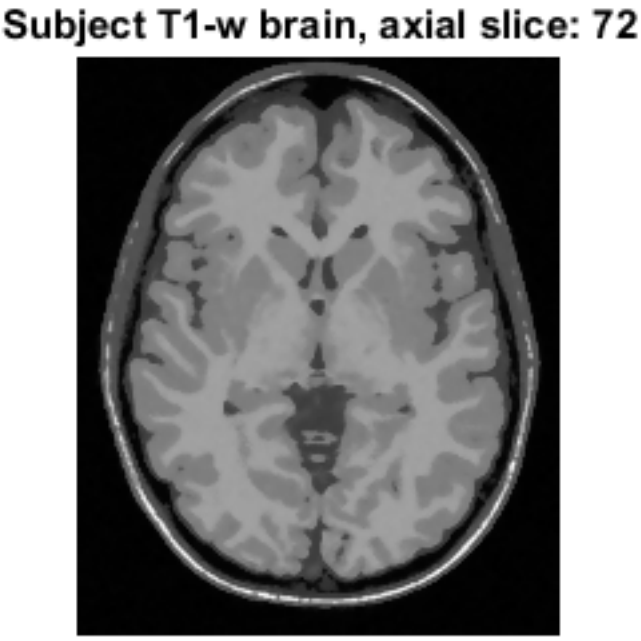
... OpenGl warnings

```
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 = 5;
Now_sss = [4 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);

% figure, imshow(imrotate(Y_MNI(:,:,showSlice),90),[]), drawnow;
% TITLE = ['MNI152 brain, A slice: ',num2str(showSlice)];
% title(TITLE)

% 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,'northwest');

% 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);

% figure, imshow(imrotate(Y_MNI_reg(:,:,showSlice),90),[]), drawnow;
% TITLE = ['MNI152 registered to subject brain, A slice:
 ',num2str(showSlice)];
% title(TITLE)

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



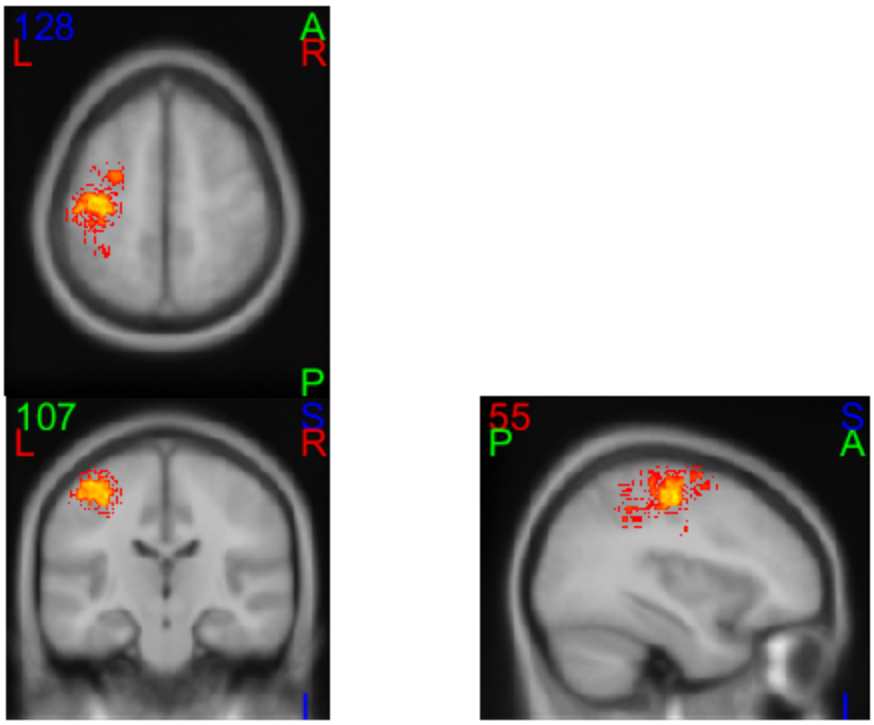
Build activation regions by modelling reported results

```
% load activation map from data files
clear task;

uiwait(msgbox('Demo example of a finger related task data from
 NeuroSynth.','Finger meta-analysis data','modal'));

% NeuroSynth finger tapping example
task.name = 'Finger Tapping';
task.activation(1).region   = [STANCEroot,'/activations/finger
 tapping_pFgA_z_FDR_0.01.nii.gz'];
task.activation(1).shape    = 'data'; % data derived by forward
 inference
task.activation(1).volume   = [];        % the index for 4D data
task.activation(1).proportion = 5.0;     % the activation thresholds:
 [lower, saturation]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(2).region   = [STANCEroot,'/activations/finger
 tapping_pAgF_z_FDR_0.01.nii.gz'];
task.activation(2).shape    = 'data'; % data derived by reverse
 inference
task.activation(2).proportion = 5.0;     % the activation thresholds:
 [lower, saturation]
```

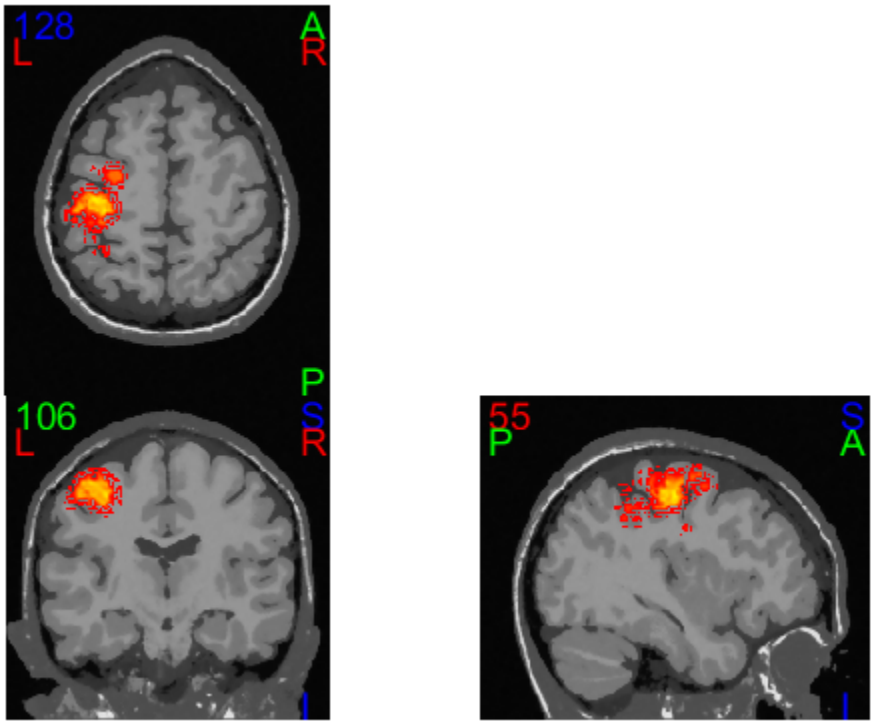
```
task.combine{1} = {'OR','all'};
task.combine{2} = {'AND','flip'};
task.combine{3} = {'mask','L'};
% define signal amplitude
task.amplitude = 0.03; % 3% activation

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

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

disp('o Combining finger tapping task activation maps from data
 files...')
task.map = STANCE_parse_combine(task);
% % explictly this does the following
% task.map = STANCE_combine_maps('OR',task.activation(:).map);
% % combine activation in opposite hemispheres
% task.map = STANCE_combine_maps('AND',task.map,flipud(task.map));
% % restrict to activation due to tasks on the R hand
% task.map(ceil(0.5*dimensions(1)):end,:,:) = 0;

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

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

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

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

% h_task_TS_R =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTS,1);
% title('R Finger Tapping Task from NeuroSynth: S')
%
% h_task_TC =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTC,2);
```

```
% title('R Finger Tapping Task from NeuroSynth: C')

[V_finger_tapping_reg,Y_finger_tapping_reg] =
 STANCE_register_activation(V_T1w.fname,task);

% figure, imshow(imrotate(Y_finger_tapping_reg(:,:,showSliceTA),90),
[]), drawnow;
% title('Finger tapping task activation of the PSM registered')

%h_task_reg =
 STANCE_display_activation_slice(Y_MNI_reg,Y_finger_tapping_reg,[]);
%title('Finger tapping task activation of the PSM registered')

% Finger tapping task activation template of the PSM of subject
TITLE = {'Finger tapping task'; 'activation template in subject'};
htask1sub =
 STANCE_display_activation_slice(Y_T1w,Y_finger_tapping_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')
task.activation(1).map = [];        % clear memory
task.activation(2).map = [];        % clear memory

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

Defining 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...
```

R Finger Tapping Task
from NeuroSynth data: MNI



Finger tapping task
activation template in subject



Reslice to funtional space according to the fMRI scan protocol

```
if makeFMRI == true

% define scan protocol parameters for finger tapping task based on
 those detailed
% in Tables 1. & 3. of "Functional Mapping of Human Sensorimotor
 Cortex with
% 3D BOLD fMRI Correlates Highly With H2150 PET rCBF" in Journal of
 Cerebral Blood Flow and Metabolism
% 16:755-764
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
    scan.voxel.size    = [3 3 3];      % [3.75 3.75 3.75] in original
 experiment
    scan.voxel.matrix  = [64 64 NaN];  % [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] tilt angle
    scan.TR            = 2400; % [ms] repetion time
    scan.TE            = 35;   % [ms] echo time
    scan.ES            = 0.51; % [ms] echo spacing
    scan.FA            = 11;   % [degrees] flip angle
    scan.BW            = 2232; % [Hz/Px]
    scan.order         = 'SD'; % SD = sequential descending order
```

```
    scan.KM0           = 2225; % fit to data with max of 909 at 3T and
 FA = 90 degree  (Siemens 3T ~4000)
    scan.noise_method  = 'percent';
    scan.noise         = 0;    % percent noise relative to peak
    scan.attenuation   = 0;  % coil attenuation factor ~mm^-1
    % 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];

    % 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;

    [~,I_max] = max(sum(sum(Y_reslice(:,:,:,3))));
    showSlice2 = I_max(1);

    % generate T2* baseline volume in functional space
    display('o Generating 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
    [V_finger_tapping_reslice,Y_finger_tapping_reslice] =
 STANCE_reslice_volume(V_finger_tapping_reg,scan,sliceLimits);
    [~,I_max] = max(sum(sum(Y_finger_tapping_reslice)));
    showSlice2TA = I_max(1);
    TITLE = {'Finger tapping task,'; ['functional axial:
',num2str(showSlice2TA)]};
    f3 = figure;
    imshow(imrotate(Y_finger_tapping_reslice(:,:,showSlice2TA),90),[])
    title(TITLE)
    set(f3,'OuterPosition',positionVector2);
    movegui(f3,'northeast')
```

```
    delete(V_finger_tapping_reg.fname);

    % mask with gray matter mask
    [Y_finger_tapping_reslice,Y_GM] =
 STANCE_GM_mask(Y_finger_tapping_reslice,task.GMvolume,Now_sss);
    %figure,
 imshow(imrotate(Y_finger_tapping_reslice(:,:,showSlice2TA),90),[]);

    subjectActivation3D{1} = Y_finger_tapping_reslice;

    cd(STANCE_genpath(Now_sss))

 save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subjectActivation3
    cd(STANCEroot)

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);
    Y_finger_tapping_reslice = squeeze(subjectActivation3D{1});

    % generate T2* baseline volume in functional space
    display('o Generating 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_reslice,Y_GM] =
 STANCE_GM_mask(Y_finger_tapping_reslice,task.GMvolume,Now_sss);
    %figure,
 imshow(imrotate(Y_finger_tapping_reslice(:,:,showSlice2TA),90),[]);
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_reslice)));
showSlice2TA = I_max(1);

% add activation to T2* baseline
disp('Adding activation to the T2* baseline ...')
```

```
[V_T2star_Map_Act,Y_T2star_Map_Act] =
 STANCE_add_activation(V_T2star_Map.fname,Y_finger_tapping_reslice,scan.TE,task.am
TITLE = {'T2* map w/ BOLD activation,', ['Axial slice:
 ',num2str(showSlice2TA)]};
f4 = figure;
imshow(imrotate(Y_T2star_Map_Act(:,:,showSlice2TA),90),[])
title(TITLE)
set(f4,'OuterPosition',positionVector2);
movegui(f4,'east')
% figure, imshow(imrotate(Y_T2star_Map_Act(:,:,showSlice2TA),90)-
imrotate(Y_T2star_Map(:,:,showSlice2TA),90),[]);
% TITLE = ['T2* map activation - baseline, A
 slice:',num2str(showSlice2TA)];
% title(TITLE)
```

o Generating T2 baseline volume in functional space.*
Adding activation to the T2 baseline ...*

Generate the EPI baseline

```
disp('Generating the EPI baseline ...')
[V_EPI0,Y_EPI0] =
 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_EPI0(:).*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')
```

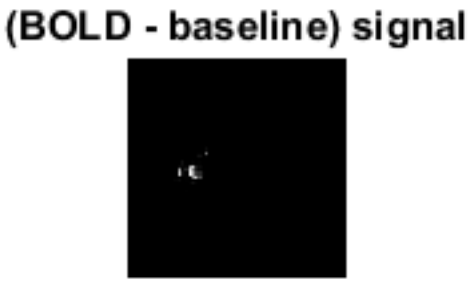
Generating the EPI baseline ...
o Writing C:\spm\STANCE/fMRI/study004/subject0001/session006\EPI_BOLD_0001_006.nii
The maximum intensity of the simulated signal: 253.9648

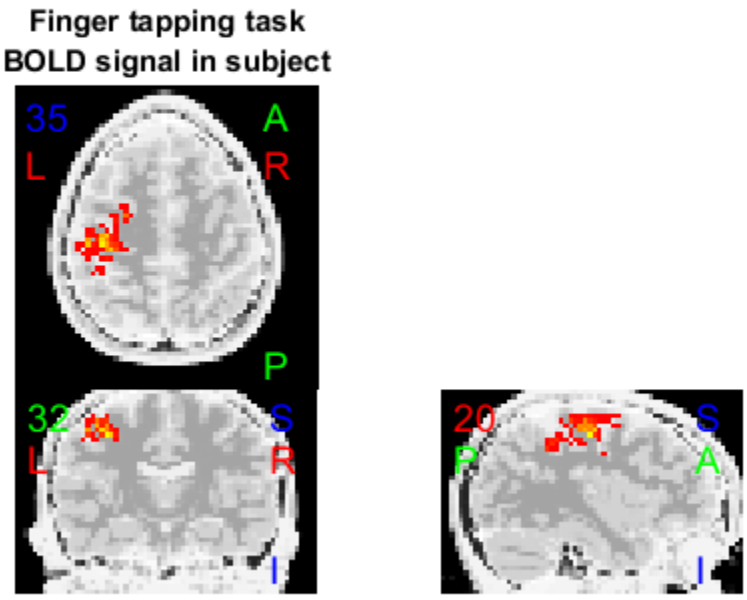
Generate the pristine EPI signal

```
disp('Generating the pristine EPI signal ...')
```

```
Now_sss = [4 1 2];
STANCE_new_session(4,1,2,true);

% exact EPI signal, no noise, no attenuation
[V_EPI,Y_EPI] =
 STANCE_EPI_signal(fn_fuzzy_reslice,Y_T2star_Map_Act,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 = {'Finger tapping task', 'BOLD signal in subject'};
htask1subfun =
 STANCE_display_activation_slice(Y_EPI,Y_finger_tapping_reslice,[],
[]);
title(TITLE)
movegui(htask1subfun,'center');

cd(STANCE_genpath)
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits')
cd(STANCEroot)
```

Generating the pristine EPI signal ...
Erasing files in C:\spm\STANCE/fMRI/study004/subject0001/session002.
o Writing C:\spm\STANCE/fMRI/study004/subject0001/
session002\EPI_BOLD_0001_002.nii
The maximum intensity of the simulated signal: 253.9648



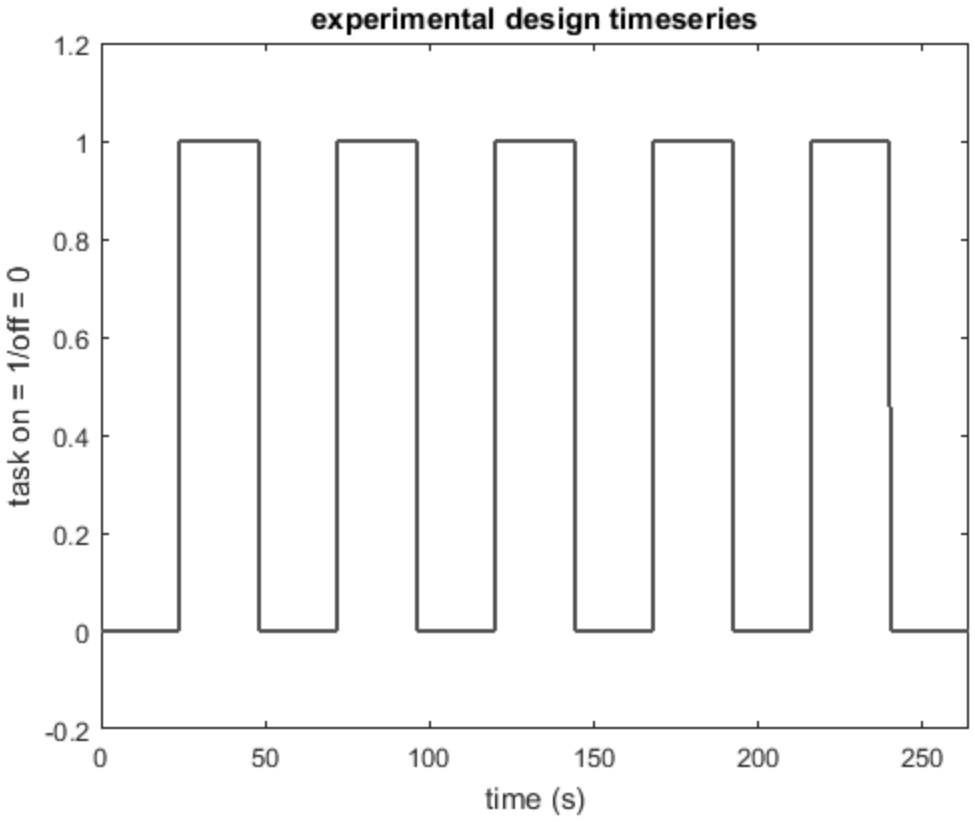


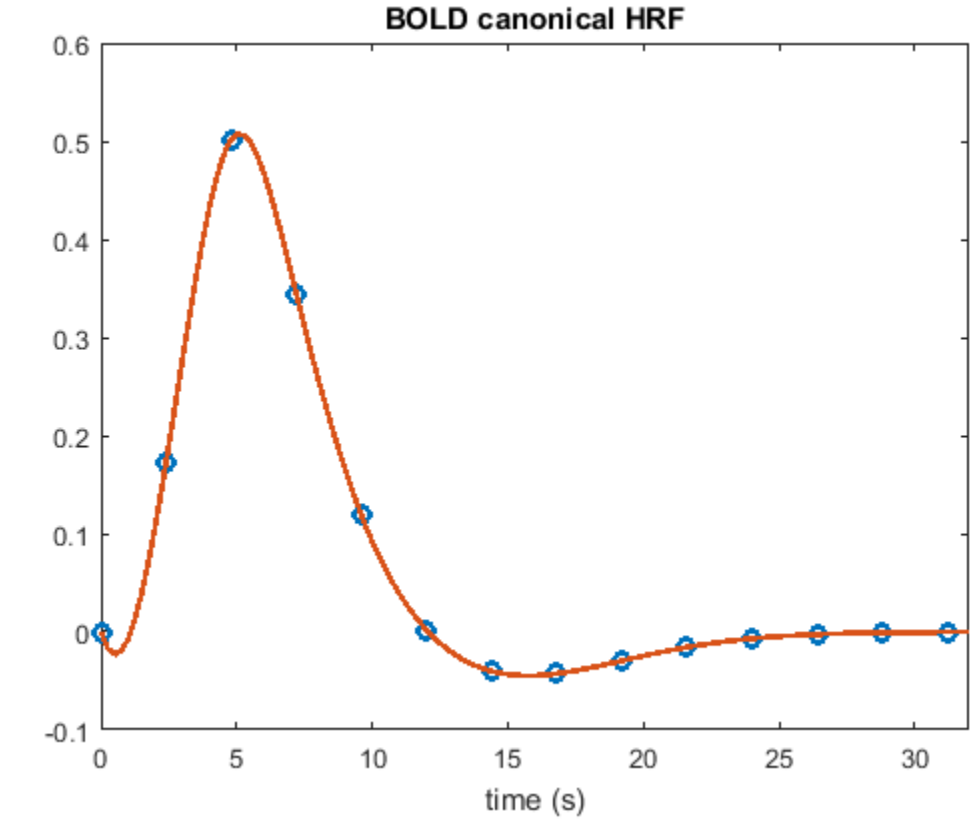
Design the 4D time-series

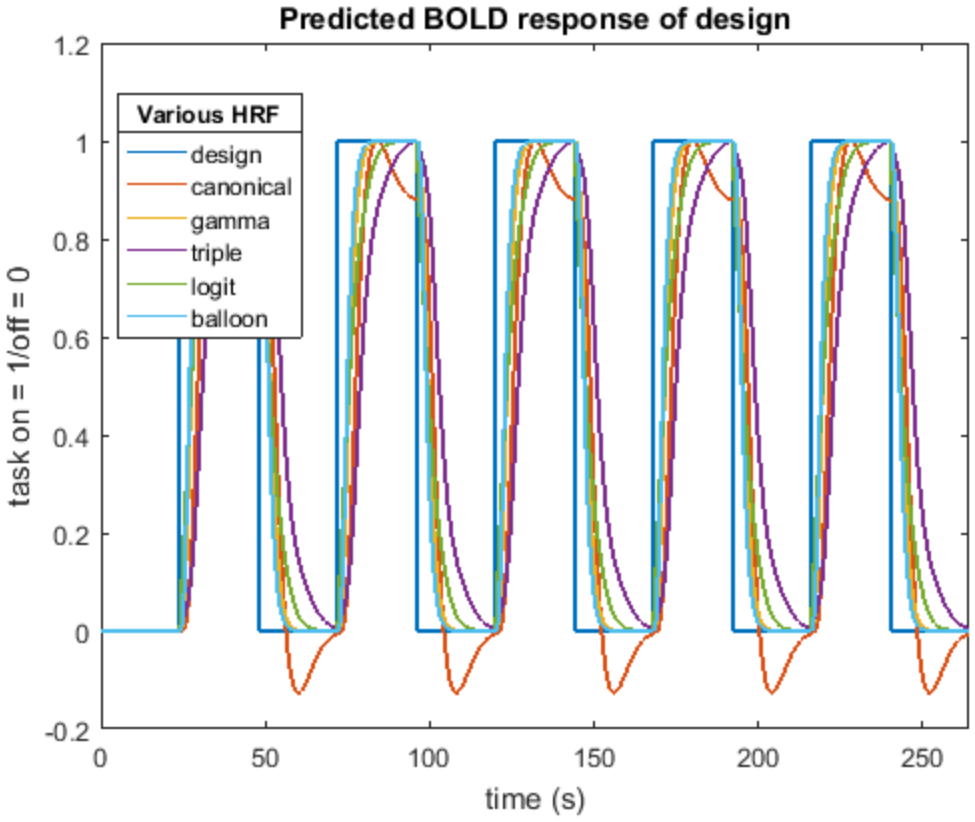
```
disp('Constructing the experiment design unto the 4D time-series ...')
uiwait(msgbox('Constructing experiment design and response.','4D time-
series'));

tic
Nslices = size(Y_T2star_Map,3);
TRsec = scan.TR/1000;
dt = TRsec/Nslices;
exp_design = STANCE_blocked_design(dt, 24, 24, 24, 24*11);
```

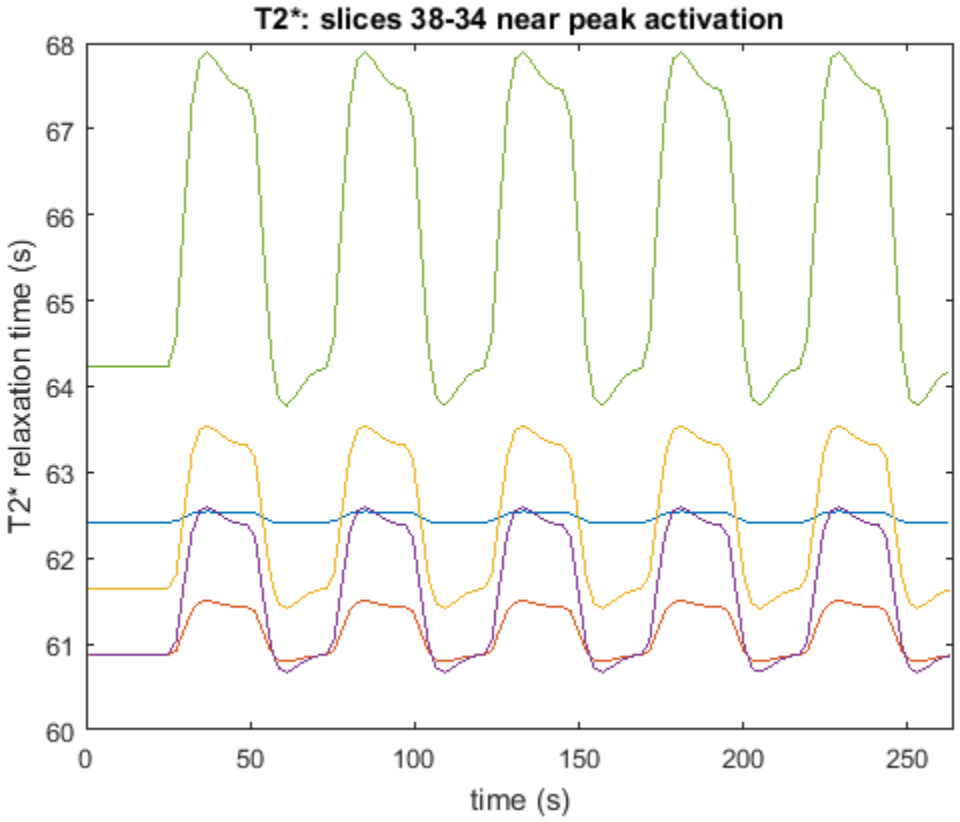
```
Nt = length(exp_design.Data);
times = dt*(1:Nt)';
NT = (Nt/Nslices);

h_expdesign = figure;
plot(exp_design,'LineWidth', 1.5,'Color',[0.33,0.33,0.33]);
ylim([-0.2 1.2])
xlim([0 times(end)])
xlabel('time (s)')
ylabel('task on = 1/off = 0')
title('experimental design timeseries')
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,'northeast');

toc

% save all of the elements common to the study
cd(STANCE_genpath(Now_sss,1))
save('STANCEstudy.mat','task','exp_design')
cd(STANCEroot)


tic

% apply canonical HRF to experimental design
BOLD_ts = STANCE_apply_response_function(dt,exp_design);
baseline_ts = (1-BOLD_ts.Data);

BOLD_gamma_ts = STANCE_apply_response_function(dt,exp_design,'gamma');
BOLD_triple_ts =
 STANCE_apply_response_function(dt,exp_design,'triple');
BOLD_logit_ts = STANCE_apply_response_function(dt,exp_design,'logit');
BOLD_balloon_ts =
 STANCE_apply_response_function(dt,exp_design,'balloon');

h_predictedBOLD = figure;
plot(times,exp_design.Data,times,BOLD_ts.Data,times,BOLD_gamma_ts.Data,times,BOLD_
ylim([-0.2 1.2])
xlim([0 times(end)])
xlabel('time (s)')
ylabel('task on = 1/off = 0')
title('Predicted BOLD response of design')
lgdBOLD =
 legend('design','canonical','gamma','triple','logit','balloon','Location','best')
```

```
title(lgdBOLD,'Various HRF')
movegui(h_predictedBOLD,'southwest');

clear BOLD_gamma_ts  BOLD_triple_ts BOLD_logit_ts BOLD_balloon_ts;

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(t) +
 Y_T2star_Map_Act(:,:,sliceTiming(STi))*BOLD_ts.Data(t);
end

Times = 1:TRsec:NT*TRsec;
figure,
plot(Times,squeeze(T2star_4D(20,32,38,:)),Times,squeeze(T2star_4D(20,32,37,:)),Tim
xlim([0 times(end)])
xlabel('time (s)')
ylabel('T2* relaxation time (s)')
title('T2*: slices 38-34 near peak activation')

Now_sss = [4 1 3];
STANCE_new_session(4,1,3,true);

Constructing the experiment design unto the 4D time-series ...
Elapsed time is 0.552010 seconds.
o Applying the canonical hemodynamic response function to time-series.
o Applying the Gamma distribution response function to time-series.
o Applying the triple Gamma response function to time-series.
o Applying the triple logit response function to time-series.
Warning: Default parameter values are used.
Erasing files in C:\spm\STANCE/fMRI/study004/subject0001/session003.
```


experimental design timeseries
task on = 1/off = 0
time (s)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
50
100
150
200
250


BOLD canonical HRF
time (s)
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
5
10
15
20
25
30

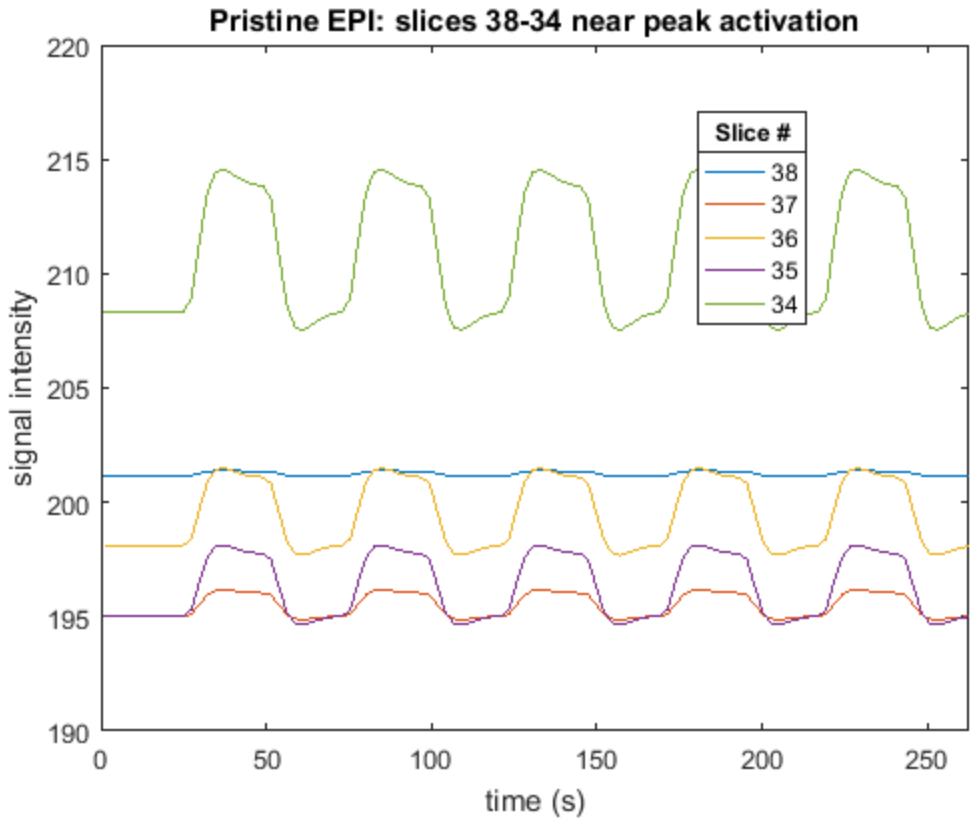

Predicted BOLD response of design
Various HRF
design
canonical
gamma
triple
logit
balloon
task on = 1/off = 0
time (s)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
50
100
150
200
250


T2*: slices 38-34 near peak activation
T2* relaxation time (s)
time (s)
68
67
66
65
64
63
62
61
60
0
50
100
150
200
250

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.')

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

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

% from figure 11 peak activation near: X = 20, Y  = 32, Z = 34

Times = 1:TRsec:NT*TRsec;
h_EPI0 = figure;
plot(Times,squeeze(Y_EPI4D(20,32,38,:)),Times,squeeze(Y_EPI4D(20,32,37,:)),Times,s
title('Pristine EPI: slices 38-34 near peak activation')
lgd = legend('38','37','36','35','34','Location','best');
xlabel('time (s)')
ylabel('signal intensity')
axis([0 Times(end) 190 220])
title(lgd,'Slice #')
movegui(h_EPI0,'center');
```

```
o Generating pristine EPI 4D signal.
o Writing C:\spm\STANCE/fMRI/study004/subject0001/
session003\EPI_BOLD_0001_003.nii
The maximum intensity of the simulated signal: 253.9648
```

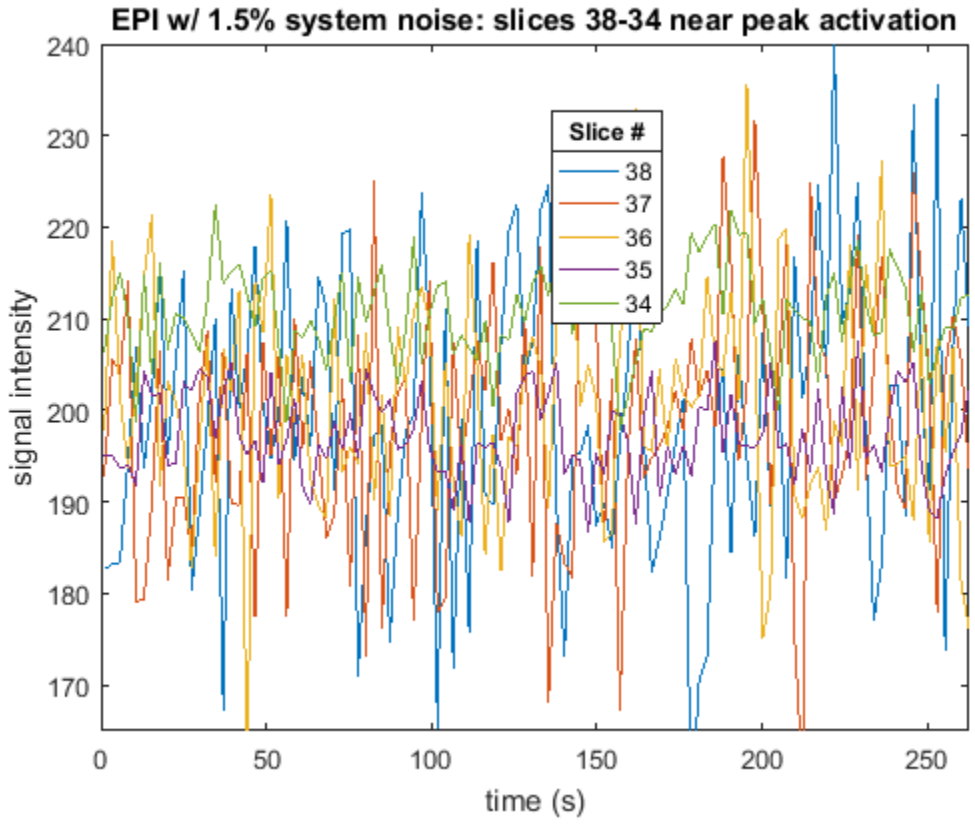


Add spatially varying system noise

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

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

scan.noise            = 1.5;    % percent noise relative to peak
noiseMap = STANCE_make_noise_map(fn_fuzzy_reslice,2,4);

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

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

toc

% from figure 11 peak activation near: X = 20, Y  = 32, Z = 34
```

```
Times = 1:TRsec:NT*TRsec;
h_EPIn = figure;
plot(Times,squeeze(Y_EPI4D(20,32,38,:)),Times,squeeze(Y_EPI4D(20,32,37,:)),Times,s
title('EPI w/ 1.5% system noise: slices 38-34 near peak activation')
xlabel('time (s)')
ylabel('signal intensity')
axis([0 Times(end) 165 240])
lgdn = legend('38','37','36','35','34','Location','best');
title(lgdn,'Slice #')
movegui(h_EPIn,'northwest');
```

```
Erasing files in C:\spm\STANCE/fMRI/study004/subject0001/session004.
o Writing C:\spm\STANCE/fMRI/study004/subject0001/
session004\EPI_BOLD_0001_004.nii
The maximum intensity of the simulated signal: 372.9436
Elapsed time is 492.719500 seconds.
```



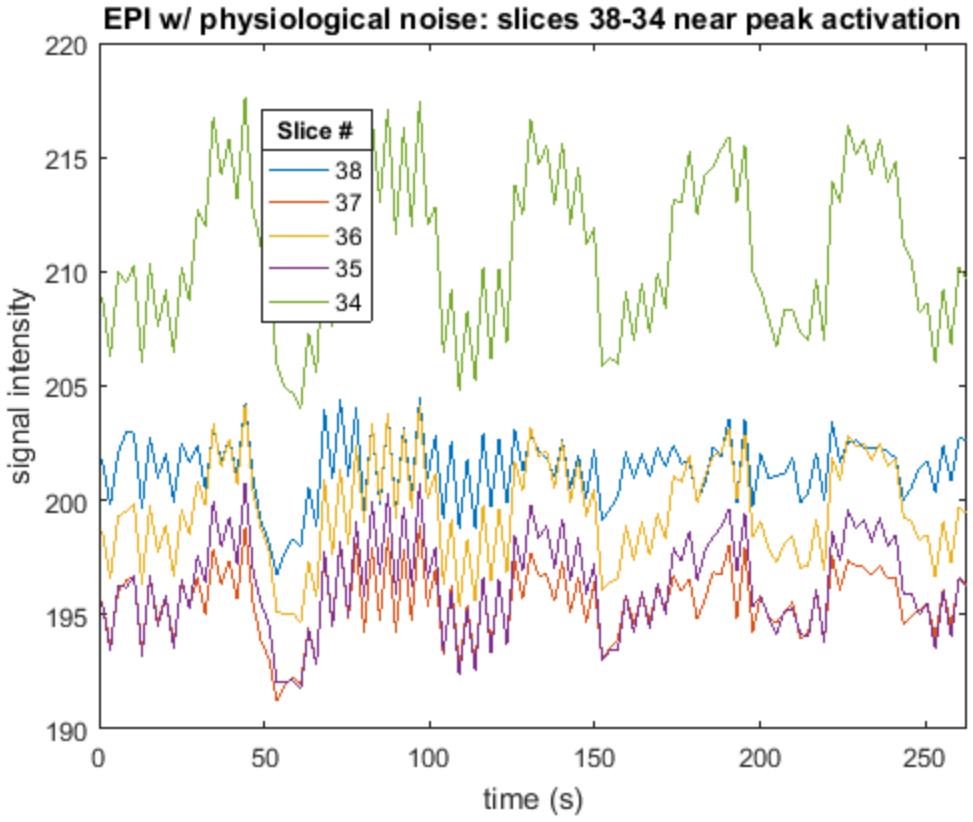
Add physiological noise times-series

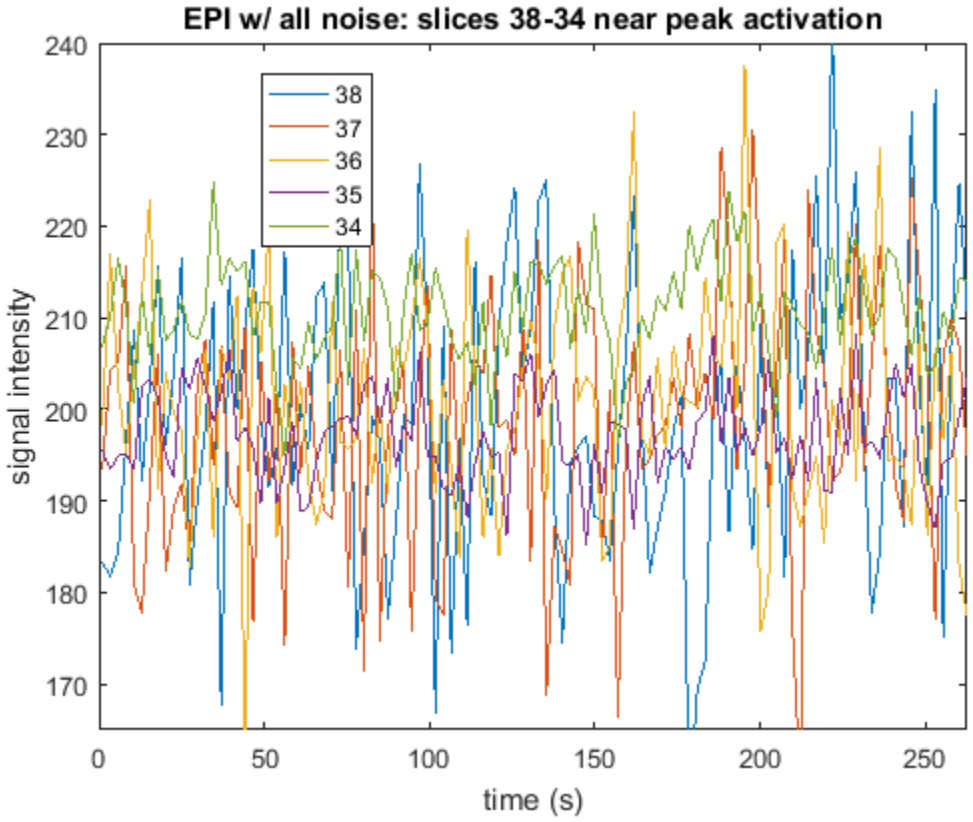
```
uiwait(msgbox('Generating EPI 4D signal with only physiological noise
 added.','4D data + physio'));

disp('Adding physiological noise to the times-series ...')
Now_sss = [4 1 5];
STANCE_new_session(4,1,5,true);
```

```
% 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           = 4.8;   % [s]  average respiratory
 time interval (NOTE: made a bit longer to better illustrate with this
 TR)
physio.respiratory.sigma        = 0.25;  % [s]  the standard deviation
 of "
physio.respiratory.A_z          = 1.0;   % [cm] average chest motion
 height
physio.respiratory.A_z_sigma    = 0.005; % [cm] the standard deviation
 of chest motion height
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];     % variation 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
% Reference (tSNR model): G. Krüger & G. H. Glover,
%                         "Physiological noise in oxygenation-
sensitive magnetic resonance imaging"
%                         Magn Reson Med. 2001 Oct;46(4):631-7.

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

% EPI timeseries, with physiological noise (just sigmas, no 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','exp_design','physio
cd(STANCEroot)

[~,Y_EPI4D] = STANCE_EPI_signal(fn_fuzzy_reslice,T2star_4D,scan,
[],physio_4D);

h_EPIp = figure;
plot(Times,squeeze(Y_EPI4D(20,32,38,:)),Times,squeeze(Y_EPI4D(20,32,37,:)),Times,s
title('EPI w/ physiological noise: slices 38-34 near peak activation')
xlabel('time (s)')
ylabel('signal intensity')
axis([0 Times(end) 190 220])
lgdn = legend('38','37','36','35','34','Location','best');
title(lgdn,'Slice #')
movegui(h_EPIp,'northeast');
```

```
% EPI timeseries, with physiological noise (just sigmas, no lags
% included), and system (thermal) noise (no attenuation)

Now_sss = [4 1 6];
STANCE_new_session(4,1,6,true);

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','exp_design','noiseM
cd(STANCEroot)

[V_EPI4D,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(20,32,38,:)),Times,squeeze(Y_EPI4D(20,32,37,:)),Times,s
title('EPI w/ all noise: slices 38-34 near peak activation')
xlabel('time (s)')
ylabel('signal intensity')
axis([0 Times(end) 165 240])
lgdn = legend('38','37','36','35','34','Location','best');
movegui(h_EPIalln,'south');

clear task;
```

```
Adding physiological noise to the times-series ...
Erasing files in C:\spm\STANCE/fMRI/study004/subject0001/session005.
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 Writing C:\spm\STANCE/fMRI/study004/subject0001/
session005\EPI_BOLD_0001_005.nii
The maximum intensity of the simulated signal: 254.0037
Erasing files in C:\spm\STANCE/fMRI/study004/subject0001/session006.
o Writing C:\spm\STANCE/fMRI/study004/subject0001/
session006\EPI_BOLD_0001_006.nii
The maximum intensity of the simulated signal: 372.9679
```


EPI w/ physiological noise: slices 38-34 near peak activation
Slice #
38
37
36
35
34
signal intensity
time (s)
220
215
210
205
200
195
190
0
50
100
150
200
250


EPI w/ all noise: slices 38-34 near peak activation
38
37
36
35
34
signal intensity
time (s)
240
230
220
210
200
190
180
170
0
50
100
150
200
250

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

Published with MATLAB® R2016a