# Anexo_II_A_Parametros Simulacion

August 27, 2018

**ANEXO II. A) Parámetros de la simulación de datos fMRI.**
**José Pedro Manzano Patrón**
**Lenguaje: Matlab**

Este anexo solo contiene el fichero de configuración de los parámetros para la simulación de datos de fMRI con SimTB [http://mialab.mrn.org/software/simtb/documentation.html]. El experimento simulado consiste en:

- Un paradigma de tarea por bloques con una sola tarea de activación (ON-OFF –> TC_block_n = 2;)
- La tarea dura 20 TRs y el intervalo entre tareas (ISI) dura 30 TRs.
- El tiempo de adquisición TR = 2 seg.
- Se simulan 24 componentes independientes (sources).
- Se suponen datos pre-procesados, por lo que se han excluido:
  - Componente 6 (Sinus signal dropout), CSF (ICs 14 y 15) y sustancia blanca (ICs 16,17). Las componentes de CSF serán modificadas en el siguiente anexo por una señal sinusoidal y la misma señal al cuadrado (por lo que habrá 26 ICs en total)
- Las componentes 4, 5 (frontal), 18 (Dorsal Atention Network), 27 y 28 (sistema auditivo) se activan levemente (amplitud = 0.5) al escuchar la orden de mover o no.
- Las componentes 22 y 23 (regiones motoras) están fuertemente relacionadas (amplitud por encima de 0.7) con la tarea de mover. Presentan así correlación alta con las componentes 4, 5, 18, 27 y 28 en esos casos. La componente 7 (precuneo) se coactiva con 22 y 23, pero es mucho mas ruidosa.
- La componente 8 (DMN) está inversamente relacionada con la tarea –> En estado de reposo se activa y en los momentos de actividad o de recibir ordenes está inversamente correlacionada.
- Las componentes 7 (ruidosa), 22 y 23 (componentes motoras) están correlacionadas entre sí y se activan solo en algunos casos (cuando tienen que pulsar el boton).
- La generación de las TC se realiza a partir de la convolución del diseño por tareas con la HRF. Ver apartado de %%TC GENERATION para conocer los detalles.
- Por simplificación para el análisis se ha supuesto:

- Todas las ICs aparecen en todos los sujetos.
  - Las tareas siguen el mismo orden en todos los sujetos.
  - Mismo tejido (sustancia gris) y baseline para todos los sujetos y componentes.

- Para que la simulación sea más similar a datos fMRI reales:
  - Se ha añadido ruido (rician noise) sin spikes de CSF en todas las componentes.
  - Se han añadido artefactos por head-motion.
  - El CNR está distribuido uniformemente de 0.65 a 2 entre los sujetos.

Guarde el siguiente fichero con el nombre que desee, pero no olvide modificarlo cuando vaya a realizar la simulación. Para ejecutar la simulación y ver los resultados, acuda al **ANEXO I. B) SIMULACIÓN DATOS fMRI**

In [ ]:
```
%-----------------------------------------------------------------------------
% To create the simulation parameter structure:
% >> sP = simtb_create_sP('experiment_params_block', M, nC);
%    Simulation can be executed with any number of subjects, M, or components, nC,
%    though nC should be >= 4 given task modulation amplitudes (see Lines 46-49).
% To run the simulation:
% >> simtb_main(sP)
%-----------------------------------------------------------------------------

%% OUTPUT PARAMETERS
%-----------------------------------------------------------------------------
% Directory to save simulation parameters and output
out_path = '/Users/hose/Desktop/TFM_TECI/simulated_data/';
% Prefix for saving output
prefix = 'block';
% FLAG to write data in NIFTI format rather than matlab
saveNII_FLAG = 0;
% Option to display output and create figures throughout the simulations
verbose_display = 1;
%-----------------------------------------------------------------------------

%% RANDOMIZATION
%-----------------------------------------------------------------------------
%seed = round(sum(100*clock));  % randomizes parameter values
seed = 3571;                    % choose seed for repeatable simulation
simtb_rand_seed(seed);          % set the seed
%-----------------------------------------------------------------------------

%% SIMULATION DIMENSIONS
%-----------------------------------------------------------------------------
M  = 5;   % number of subjects
% nC is the number of components defined below, nC = length(SM_source_ID);
nV = 148; % number of voxels; dataset will have [nV x nV] voxels.
nT = 250; % number of time points
```

```matlab
TR = 2;   % repetition time

% number of different connectivity states
%nStates = 2;

% probability of unique events
%pU = 0.5;

% amplitude of unique events (relative to module-specific events)
%aU = .5;

% probability of state specific events
%pState = .5;

%Module membership for each state
%ModMem = zeros(nC,nStates);

% Number of event types (i.e., number of different modules)
%nE = 3;


%% SPATIAL SOURCES
%-------------------------------------------------------------------------------
% Choose the sources. To launch a stand-alone GUI:
% >> simtb_pickSM
SM_source_ID = [ 2  3  4  5  7  8  9 10 11    ...
                12 13 18 19 20 21 22 23 24 ...
                25 26 27 28 29 30]; % all but (1, 6, 14, 15, 16, 17)

% Sources 1 (general non-brain), 6 (Sinus signal dropout), 14,15 (CSF) and 16, 17 (WM) a
% as if the signal is already pre-processed

nC = length(SM_source_ID);  % number of components

% LABEL COMPONENTS
%%
%nonbrain  = find(SM_source_ID == 1);
s2                          = find(SM_source_ID == 2);
s3                          = find(SM_source_ID == 3);

% Frontal: 1 second temporal delay from bilateral frontal
comp_F1     = find(SM_source_ID ==  4);
comp_F2     = find(SM_source_ID ==  5);

% Medial Frontal: has lower baseline intensity (signal dropout)
%comp_MF     = find(SM_source_ID ==  6);

% Precuneus: activation only to targets
```

```
comp_P      = find(SM_source_ID ==  7);

% DMN: negative activation to task events
comp_DMN    = find(SM_source_ID ==  8);

%
s9                   = find(SM_source_ID ==  9);
s10                  = find(SM_source_ID ==  10);
s11                  = find(SM_source_ID ==  11);
s12                  = find(SM_source_ID ==  12);
s13                  = find(SM_source_ID ==  13);

% Dorsal Attention Network: activation to novels more than targets
comp_DAN   = find(SM_source_ID == 18);


s19                   = find(SM_source_ID == 19);
s20                   = find(SM_source_ID == 20);
s21              = find(SM_source_ID == 21);

% (Sensori)Motor: activation to targets and novels (weakly)
comp_M1    = find(SM_source_ID == 22);
comp_M2    = find(SM_source_ID == 23);

% Bilateral frontal: positive activation to for targets and novels
comp_BF    = find(SM_source_ID == 24);

s25                    = find(SM_source_ID == 25);
s26                    = find(SM_source_ID == 26);

% Here, we label components or component groups that may be used later
% Auditory: strong positive activation for all task events
comp_AUD1  = find(SM_source_ID == 27);
comp_AUD2  = find(SM_source_ID == 28);

% Hippocampus: activation only to novels
comp_H1    = find(SM_source_ID == 29);
comp_H2    = find(SM_source_ID == 30);


% compile list of all defined components of interest
complist = [s2 s3 s9 s10 s11 s12 s13 s19 s20 s21 s25 s26 ...
                        comp_AUD1 comp_AUD2 comp_DMN comp_BF  comp_F1 comp_F2 ...
            comp_P    comp_DAN  comp_H1  comp_H2  comp_M1 comp_M2 ...
            ];

%-------------------------------------------------------------------------------
%% COMPONENT PRESENCE
```

```
%---------------------------------------------------------------------------
% [M x nC] matrix for component presence: 1 if included, 0 otherwise
% For components not of interest there is a 90% chance of component inclusion.
%SM_present = (rand(M,nC) < 0.9);  Some not present in some subjects
SM_present = ones(M,nC);    % all sources present in all subjects

% Components of interest (complist) are included for all subjects.
SM_present(:,complist) = ones(M,length(complist));


%---------------------------------------------------------------------------
%% SPATIAL VARIABILITY
%---------------------------------------------------------------------------
% Variability related to differences in spatial location and shape.
SM_translate_x = 0.1*randn(M,nC);  % Translation in x, mean 0, SD 0.1 voxels.
SM_translate_y = 0.1*randn(M,nC);  % Translation in y, mean 0, SD 0.1 voxels.
SM_theta       = 1.0*randn(M,nC);  % Rotation, mean 0, SD 1 degree.
%                 Note that each 'activation blob' is rotated independently.
SM_spread = 1+0.03*randn(M,nC); % Spread < 1 is contraction, spread > 1 is expansion.


%---------------------------------------------------------------------------
%% TC GENERATION
%---------------------------------------------------------------------------
% Choose the model for TC generation.  To see defined models:
% >> simtb_countTCmodels

TC_source_type = ones(1,nC);    % % Types of model generations of TCs. 1 = convolution u
TC_source_params = cell(M,nC);  % initialize the cell structure
% Use the same HRF for all subjects and relevant components
P(1) = 8;    % delay of response (relative to onset)
P(2) = 16;   % delay of undershoot (relative to onset)
P(3) = 1;    % dispersion of response
P(4) = 1;    % dispersion of undershoot
P(5) = 8;    % ratio of response to undershoot
P(6) = 6;    % onset (seconds)
P(7) = 24;   % length of kernel (seconds)
[TC_source_params{:}] = deal(P);

% Implement 1 second onset delay for components comp_F1 and comp_F2, for instance
P(6) = P(6) + 1;  % delay by 1s
[TC_source_params{:,[comp_F1 comp_F2]}] = deal(P);

%---------------------------------------------------------------------------
%% EXPERIMENT DESIGN
%---------------------------------------------------------------------------
% BLOCKS
TC_block_n = 2;          % Number of blocks [set = 0 for no block design]
```

```
TC_block_same_FLAG = 1;  % 1 = block structure same for all subjects
                         % 0 = block order will be randomized
TC_block_length = 20;    % length of each block (in samples)
TC_block_ISI    = 30;    % length of OFF inter-stimulus-intervals (in samples)
TC_block_amp    = zeros(nC, TC_block_n); % initialize [nC x TC_block_n] matrix

% task-state 1: OFF
TC_block_amp([comp_AUD1 comp_AUD2],             1) = 1.0; % moderate task-modulation
TC_block_amp([comp_BF comp_F1 comp_F2 comp_DAN], 1) = 0.7; % mild
TC_block_amp([comp_DMN],                        1) =-0.5; % negative weak
% task-state 2: ON
TC_block_amp([comp_AUD1 comp_AUD2],             2) = 1.2; % strong
TC_block_amp([comp_BF comp_F1 comp_F2],         2) = 1.0; % moderate
TC_block_amp([comp_DAN],                        2) = 0.8; % mild
TC_block_amp([comp_P],                          2) = 0.5; % weak
TC_block_amp([comp_M1 comp_M2],                 2) = 1.0; % moderate
TC_block_amp([comp_DMN],                        2) =-0.5; % negative weak

%-------------------------------------------------------------------------------
%% UNIQUE EVENTS
%-------------------------------------------------------------------------------
TC_unique_FLAG = 1; % 1 = include unique events
TC_unique_prob = 0.2*ones(1,nC); % [1 x nC] prob of unique event at each TR

TC_unique_amp  = ones(M,nC);     % [M x nC] matrix of amplitude of unique events
TC_unique_amp(:,[comp_AUD1 comp_AUD2])            = 0.35;
TC_unique_amp(:,[comp_BF comp_F1 comp_F2])        = 0.3;
TC_unique_amp(:,[comp_DAN])                       = 0.5;
TC_unique_amp(:,[comp_P])                         = 0.5;
TC_unique_amp(:,[comp_M1 comp_M2])                = 0.2;
TC_unique_amp(:,[comp_H1 comp_H2])                = 0.4;
TC_unique_amp(:,[comp_DMN])                       = 0.3;

%-------------------------------------------------------------------------------
%% DATASET BASELINE
%-------------------------------------------------------------------------------
% [1 x M] vector of baseline signal intensity for each subject
D_baseline = 800*ones(1,M); % [1 x M] vector of baseline signal intensity

%-------------------------------------------------------------------------------
%% TISSUE TYPES
%-------------------------------------------------------------------------------
% FLAG to include different tissue types (distinct baselines in the data)
D_TT_FLAG = 0;                     % if 0, baseline intensity is constant
D_TT_level = [1.15, 0.8, 1, 1.2]; % TT fractional intensities
% To see/modify definitions for tissue profiles:
% >> edit simtb_SMsource.m
```

```
%-------------------------------------------------------------------------------
%% PEAK-TO-PEAK PERCENT SIGNAL CHANGE
%-------------------------------------------------------------------------------
D_pSC = 3 + 0.25*randn(M,nC);   % [M x nC] matrix of percent signal changes
% To make statistical moments of data look more like real data
%D_pSC(:,comp_CSF1) = 1.2*D_pSC(:,comp_CSF1);
%D_pSC(:,comp_CSF2) = 1.2*D_pSC(:,comp_CSF2);
%D_pSC(:,comp_WM1)  = 0.5*D_pSC(:,comp_WM1);
%D_pSC(:,comp_WM2)  = 0.5*D_pSC(:,comp_WM2);

%-------------------------------------------------------------------------------
%% NOISE
%-------------------------------------------------------------------------------
D_noise_FLAG = 1;               % FLAG to add rician noise to the data
% [1 x M] vector of contrast-to-noise ratio for each subject
% CNR is distributed as uniform between 0.65 and 2.0 across subjects.
minCNR = 0.65;  maxCNR = 2;
D_CNR = rand(1,M)*(maxCNR-minCNR) + minCNR;

%-------------------------------------------------------------------------------
%% MOTION
%-------------------------------------------------------------------------------
D_motion_FLAG = 0;              % 1=motion, 0=no motion
D_motion_TRANSmax = 0.02;       % max translation, proportion of entire image
D_motion_ROTmax = 5;            % max rotation, in degrees
D_motion_deviates = ones(M,3);  % proportion of max each subject moves
D_motion_deviates(1,:) = 0.5;   % Subject 1 moves half as much
%-------------------------------------------------------------------------------
% END of parameter definitions
```