Portfolio assignment9 EEG Stop-Signal-NoGo exp

Andreas Højlund (linahn@cfin.au.dk)

April 20 2018

Stop-Signal-NoGo

This experiment was an EEG experiment with 3 levels

Deadline

April 30 2018

Stimuli

Participants listened to two different sounds:

“Go”-sound: piano note E (aka. “high note”), 100 ms, 20 ms rise and fall

“Stop”-/“NoGo”-sound: piano note C from the octave below E (i.e., 17 semitones lower than E, aka. “low note”), 100 ms, 20 ms rise and fall

Generally, participants were given max 1500 ms to respond (from the onset of the “Go”-sound). If no response, then a random waiting interval between 0-1000 ms; and if they responded, this waiting interval was varied randomly between 1000-2000.

For “Stop”-trials, ISI (inter-stimulus-interval, aka. “silent gap”) = 100 ms at the outset, which then changed in steps of ±25 ms (first three steps were 200 ms, 100 ms, and 50 ms) depending on whether the participant correctly withheld his/her response (increased) or made a commission error (decreased). This adaptive approach ensured approximately 50% correct responses (i.e., no response) and 50% incorrect responses (i.e., pressing “space”).

Number of stimuli

100 “Stop”-trials (approx. 50% hit/miss, i.e. 50 each)
50 “NoGo”-trials
~375 “Go”-trials (because of range from 0-5 “Go”-trials for each “Stop”-/”NoGo”-trial)

Hence, ~3.75:1 (~375:100) proportion of “Go”- vs. “Stop”-trials
and ~7.5:1 (~375:50) proportion of “Go”- vs. “Stop”-trials

Responses/instructions

Participants were instructed to press “space” to the “Go”-sounds as fast as possible, but, crucially, NOT to respond if they heard the “Stop”-/“NoGo”-sound. Participants were furthermore informed that the “Stop”-/”NoGo”-sounds would occur either immediately after the “Go”-sounds or by themselves, and the participants were instructed to refrain from pressing “space” in both situations.

EEG details

Sampling rate = 1000 Hz
Analog high-pass-filter = 0.1 Hz
Number of channels = 32
Original reference = FCz
EOG channels = HEOG (chan. 28) and VEOG (chan. 32), ref. = FCz

Headerfile = ‘Group[number]_[initials].vhdr’ (this is the inputfile for SPM)
Markerfile = ‘Group[number]_[initials].vmrk’
Datafile = ‘Group[number]_[initials].eeg’

NB! Changing filenames of the raw files is NOT straightforward (i.e., you have to also fix them inside ‘*.vhdr’ and ‘*.vmrk’, which are just plain text files, and then save these files in their original format (i.e., with their ‘.vhdr’ and ‘.vmrk’ suffixes)).

Trigger values:

“Go” (alone) =	`‘S 10’`	“Go” before “Stop” =	`‘S 20’`
“Stop” =	`‘S 11’`	“NoGo” =	`‘S 12’`
any response =	`‘S  1’`		

Tasks

Preprocessing

1. Make a **batch** including the following modules:
   (Start out by setting everything up and testing it on your group’s own dataset, then you can apply the code snippet below and run it on all the participants)
   i. **Conversion** (use default values; when looping over participants later on, “File Name” should be left empty [Edit > Clear Value])
   ii. **Montage** (use the ‘avg_reref.mat’ file available on Blackboard; “Keep other channels” > ‘Yes’ – we want to keep our two EOG-channels)
   iii. **Filter** (Highpass; 0.1 Hz – it says “Cutoff(s)”, but the parenthesis refers to “optionally plural”, not “= seconds”; should be specified in Hz)
   iv. [Optional] **Downsampling** (250 Hz), reducing data (every 4th sample)
   v. [“Optional”] **Filter** (Lowpass: 45 Hz (for you to play with and see how it affects your data)) - NB! Leave the option of a Filter in this particular spot in the batch, then edit the cutoff as you like
   vi. **Epoching** (“How to define trials” > ‘Define trial’):
      a. Time window > [-100 800]
      b. Trial definitions:
         - Make 5 trials using the earlier specified trigger values
         - You decide the Condition labels (but for scripting purposes [evident later], consider sticking to my labels above)
         - Event type > ‘Stimulus’
         - NB! Take notice of the order in which you create your trials ;)
      c. Event padding > 0.4 (in seconds; this allows us to use the responses up to 1000 ms after sound onset if we later want to sort the trials based on correct responses, etc.)

vii. **Artefact detection** (use default settings, specify “Threshold channels” > 80); we simply flag as bad all 1 sec segments where the amplitude exceeds 80 µV (think back to what the eye blinks looked like)

viii. **Averaging** (“Robust” > 3; “Compute weights by condition” > ‘Yes’; “Save weights” > ‘No’ – we don’t normally save these weights, but if you want to know more about how this ‘robust averaging procedure’ actually works, you can input ‘Yes’ and inspect)

ix. **Filter** (Lowpass: 45 Hz (or whatever you used in your previous lowpass-step) – SPM advices you to re-apply your lowpass-filter after robust averaging as this procedure may induce high-frequency artefacts (mainly around the edges of the epochs, which is due to the weighting procedure)

x. **Contrast over epochs** (“Weight by replications” > ‘No’):
Make 8 contrasts, replicating the 5 trials you already created, and adding 3 new:

a. NoGo-sound minus Go-sound

b. Stop-sound minus Go-sound

c. Stop-sound minus NoGo-sound
You decide the order, and you decide their labels

xi. **Convert2Images** (“Mode” > ‘scalp x time’; default values otherwise - later on, you may consider also setting “Time window” to e.g. [0 400] in order to narrow down/focus your statistical analysis a bit)

xii. **Smooth** (“FHWM” > [20 20 20] (mm mm ms); “Implicit masking” > ‘Yes’ – this excludes the areas outside the scalp from the analysis)

Group analysis

2. “Use” the following code snippet to get your batch to run on all 10 participants (NB! Don’t use the actual text from below – get the “code_snippets.m” from Blackboard):

```
%% Setting up the batch to run across a loop of subjects

% % Make sure you have SPM added to your path and initialized (you probably
% % already have)
% addpath /YourDrive/YourSPMLocation/ % e.g.
/Users/au183362/Documents/MATLAB/toolboxes/spm12/ %% YOU NEED TO EDIT THIS
%
% % initialize spm (without loading gui)
% spm('defaults', 'eeg');

% set your filepath, folder extension names, batchpath, etc.
filePath = '/Users/au183362/Documents/MATLAB/CogNeuroSci'; % what's your base path? %% YOU
NEED TO EDIT THIS
extRaw = 'raw'; % have you saved your raw files in "raw"?
extScratch = 'scratch'; % have you created a scratch-folder?
batchPath = fullfile(filePath,'scripts/batch_stop_nogo_preproc.mat'); % where have you saved
your batch and under what name?

% initialize the spm_jobman
spm_jobman('initcfg');

% load the relevant batch into the workspace (specified earlier)
load(batchPath)

% set the path for where to save your processed files, and cd there
savePath = fullfile(filePath,extScratch);
cd(savePath)

% create input structure
g = 1:12; % group numbers
ids = {'NM','ER','CM','TRS','AWL','KEP','LTB','FS','SMN','SK','VMP','TP'}; % 2018 initials
```

```
for i = 1:length(g)
    inputs = cell(1,1); % this clears the inputs-structure on every interation
    inputs{1} = fullfile(filePath,extRaw,sprintf('Group%d_%s.vhdr',g(i),ids{i})); % exact
path and filename of each datafile to be processed
    spm_jobman('serial', matlabbatch,'',inputs); % spm_jobman will then run your batch on
your 12 selected files
end
```

3. Grand-average across all 10 participants with the following code snippet (NB! Still don't use the actual text from below – get the "code_snippets.m" from Blackboard):

```
%% Grand mean setup (+ low-pass)

extScratch = 'scratch'; % have you created a scratch-folder?
% extScratch = 'scratch/success'; % in which scratch-folder are your grand-mean inputs?

g = 1:12; % group names
% ids =
ids = {'NM','ER','CM','TRS','AWL','KEP','LTB','FS','SMN','SK','VMP','TP'};

for i = length(g):-1:1
    P{i,1} =
fullfile(filePath,extScratch,sprintf('wfmaefdfMspmeeg_Group%d_%s.mat',g(i),ids{i})); % MAKE
SURE THIS MATCHES YOUR FILENAMES
end
% grandmean
S = [];
S.D = char(P);
S.outfile = fullfile(filePath,extScratch,'GMavg_stop_nogo.mat'); % WHAT SHOULD YOUR
GRANDMEAN DATAFILE BE CALLED?
S.weighted = 0;   % 0 = proper 2nd level; 1 = only pseudo 2nd level (weights average by
number of replications in inputs)
D = spm_eeg_grandmean(S);

S = [];
S.D = fullfile(filePath,extScratch, D.fname);
S.band = 'low';
S.type = 'butterworth';
S.order = 5;
S.dir = 'twopass';
S.freq = 30;
D = spm_eeg_filter(S);  % THIS WILL APPEND AN "f"-PREFIX TO THE FILENAME YOU JUST SPECIFIED
IN THE PREVIOUS STEP
```

Successful and unsuccessful Stop-responses

4. Make an identical batch to your previous batch, only this should start with the module called **Artefact detection** (and the input should be left blank)

    i. Only change to this batch should be in the **Contrast over epochs** module where you will now specify 9 contrasts:

        a. replicate the 5 conditions you already created

        b. add 1 new condition: the StopError condition (see the relevant code snippet below for how this condition was created)

        c. add 3 actual contrasts:

            - Stop minus StopError
            - Stop minus Go
            - StopErro rminus Go

5. Now **copy** all your epoched files into a subfolder called "success" (hint: use the prefixes to identify the epoched files)
6. Use the code snippet for 'Splitting into successful and unsuccessful Stop-trials in code_snippets.m:

```
%% Splitting into succesful and unsuccesful Stop-trials

% % Make sure you have SPM added to your path and initialized (you probably
% % already have)
% addpath /YourDrive/YourSPMLocation/ % e.g.
/Users/au183362/Documents/MATLAB/toolboxes/spm12/ %% YOU NEED TO EDIT THIS
%
% % initialize spm (without loading gui)
% spm('defaults', 'eeg');

% set your filepath, folder extension names, batchpath, etc.
filePath = '/Users/au183362/Documents/MATLAB/CogNeuroSci'; % what's your base path? %% YOU
NEED TO EDIT THIS
extRaw = 'raw'; % have you saved your raw files in "raw"?
extScratch = 'scratch/success'; % have you created a scratch-folder?
batchPath = fullfile(filePath,'scripts/batch_stop_nogo_artefact_succ.mat'); % where have you
saved your batch and under what name?

% initialize the spm_jobman
spm_jobman('initcfg');

% set the path for where to save your processed files (which should be different from our
base-scratch-folder), and cd there
savePath = fullfile(filePath, extScratch);
cd(savePath)

% create input structure
g = 1:12; % group numbers
ids = {'NM','ER','CM','TRS','AWL','KEP','LTB','FS','SMN','SK','VMP','TP'}; % 2018 initials

for i = 1:length(g)
    loadPrefix = 'efdfMspmeeg'; % specify the prefix for the epoched files
    D = spm_eeg_load(fullfile(filePath,extScratch,sprintf('%s_Group%d_%s.mat',loadPrefix,
g(i), ids{i}))); % loading the already epoched file
    listConds = conditions(D)'; % list of all the epoched trials
    idxGoStop = strmatch('Go-Stop', listConds); % trial-indices of all the Go-Stop-trials
    idxStop = strmatch('Stop', listConds); % trial-indices of all the Stop-trials
    idxStopAll = idxStop; % creating a copy of our idxStop-list
    idxStop(idxStop==ntrials(D)) = []; % in case the very last trial is a Stop-trial, we
have to discard it for now (because we use "+1" in the next line to identify "Response"s)
    idxGoStopError = strmatch('Response', listConds(idxGoStop+1)); % identifying all trials
where there's a response right after the Go-Stop sound (aka. a pseudo-error)
    idxStopError = strmatch('Response', listConds(idxStop+1)); % identifying all trials
where there's a response right after the Go-Stop sound (aka. an error)
    D = conditions(D, idxStop(idxStopError), 'StopError'); % renaming all Stop trials where
there was a response AFTER the stop-signal
    D = conditions(D, idxGoStop(idxGoStopError), 'Go-StopError'); % renaming all Go-Stop
trials where there was a response BEFORE the stop-signal
    D = conditions(D, idxGoStop(idxGoStopError)+2, 'Stop-PreError'); % renaming all Stop
trials where there was a response BEFORE the stop-signal
    save(D);

    load(batchPath)
    inputs = cell(1,1); % this clears the inputs-structure on every interation
    inputs{1} = fullfile(filePath,extScratch,fname(D)); % exact path and filename of each
datafile to be processed
    spm_jobman('serial', matlabbatch,'',inputs); % spm_jobman will then run your batch on
your 12 selected files
end
```

7. Finally, re-use the grand-mean from above, only change

```
extScratch = 'scratch'; % have you created a scratch-folder?
% extScratch = 'scratch/success'; % in which scratch-folder are your grand-mean inputs?
```

to

```
% extScratch = 'scratch'; % have you created a scratch-folder?
extScratch = 'scratch/success'; % in which scratch-folder are your grand-mean inputs?
```

Voluntary extra tasks

Stats

8. Make a batch with the following 3 modules:
   i. **Factorial design specification:**
      a. “Directory” > leave blank
      b. “Design” > “One-sample t-test” > “Scans” > leave blank

ii. **Model estimation** (defaults)

iii. **Contrast manager** (“Contrast Sessions”):

   a. T-contrast > “Name” > ‘diff-t’; “Weights vector” > 1

   b. T-contrast > “Name” > ‘diff-t-neg’; “Weights vector” > -1

   c. F-contrast > “Name” > ‘diff-f’; “Weights vector” > 1

   d. “Delete existing contrasts” > ‘Yes’

9. Use the following code snippet (from Blackboard) to run your stats-batch on the Stop-vs-StopError previously defined difference wave (you can also run it on any of the other contrasts that we’ve defined earlier)

```
%% STATS

% initialize the spm_jobman
spm_jobman('initcfg');

% load the relevant batch into the workspace
batchPath = fullfile(filePath, 'scripts/batch_stop_nogo_stats.mat');
load(batchPath)

% conditions
% conds = {'Stop-vs-Go', 'NoGo-vs-Go', 'Stop-vs-NoGo'}; % HAVE YOU NAMED YOUR CONTRASTS
EXACTLY AS I HAVE? OTHERWISE, ADJUST THESE STRINGS
conds = {'Stop-vs-StopError'}; % HAVE YOU NAMED YOUR CONTRASTS EXACTLY AS I HAVE? OTHERWISE,
ADJUST THIS/THESE STRING(S)

for i = 1:length(conds)
    savePath = fullfile(filePath, extStats, conds{i});
    if ~exist(savePath, 'dir')
        mkdir(savePath)
    end
    inputs{1} = cellstr(savePath);

    [inputs{2}, ~] = spm_select('ExtFPListRec', fullfile(filePath, extScratch),
sprintf('(^scondition_%s.nii)', conds{i}));
    inputs{2} = cellstr(inputs{2});

    spm_jobman('serial', matlabbatch, '', inputs{:});
end
```

i. Click on “Results” in the GUI in order to inspect your effects:

   a. Select your contrast (e.g., ‘diff-t’); “apply masking” > ‘none’; “p value” > ‘none’ (or ‘FWE’); “threshold” > 0.001 (or 0.05); “extent” > 50 (or 200); “Data Type: …” > ‘Scalp-Time’

Reporting

I. Create and save a plot of your group’s participant’s ERP to the Go-, Stop-, and NoGo-sounds from channel Fz (I suggest using Display… > M/EEG in the GUI, and when you have the figure open: File > Save As…, and then choose the pdf-option).

II. Do the same for the lowpass-filtered group-average and create and save a plot of the three difference waves (aka. “contrasts”) at Cz.

III. Save a plot for the Stop and StopError conditions at Cz for the non-lowpass-filtered group-average of the successful/unsuccessful data

IV. Voluntary: Report the stats output coordinate table (including the graphics) of your preferred comparison (default: Stop-vs-StopError) and contrast (e.g., diff-f).

V. Collect all material and submit as a single file (pdf or html) to Blackboard.