May 2024

Real-time neurofeedback: methods, practices and application

Workshop practice material

The MATLAB code in this workshop uses the real-time output data files (with the extension '.rtp') generated by the **neurofeedback (NF)** software (called TBV) to calculate and present feedback with Psychtoolbox inside the MRI scanner.

All the scripts used here are from an fMRI NF study that was used to train/teach meditation/mindfulness in beginners. All the task instructions hence involve instructions to meditate. Since this is a simulation exercise, the scripts will reuse the output files generated earlier during the meditation NF study.

All the template scripts for this exercise are in *Code/Template_Code_simulation/* in the MAC or Windows folder. All the data are in *Data/.* All the scripts have detailed comments explaining different aspects of the code. All comments are green in colour, and are followed by a % sign in the script.

Upon completing this hands-on practical session, you will have learnt to manipulate NF source and NF interface with Psychtoolbox in MATLAB, using real-time NF outputs generated by the NF software.

Note: This workshop is designed to be beginner-friendly to suit those with minimal experience in coding/programming. In case you find any part of this exercise challenging, please ask for help. To know more about any of the MATLAB functions used in any part of the script, type *help <function_name>* in the MATLAB Command Window (e.g., *help corr*). You can also find help in *Code/Working_Code_simulation* which contains all the working scripts (with answers

to the non-optional coding questions) that are ready to be run. These scripts do not any need modifications to work.

Getting ready

Download the workshop material from HERE. Unzip the downloaded folder. Open *Terminal* if using Mac or *Command Window* if using Windows.

Type the following two commands:

> *cd* <path to the unzipped folder's path>
>
> <path to MATLAB folder>*MATLAB_R20XXx.app/bin/matlab*
>
> (if using Mac)
>
> OR
>
> "<path to MATLAB folder>*R20XXx\bin\matlab.exe"

In *MAC/data/NeuroFeedback_Run_Two* subfolder, you can open *NeuroFeedback_RunTwo-3.rtp.* In this file, the first number represents the number of brain regions (2 in this case) used by TBV to calculate mean fMRI signals. The 2nd and 3rd value represent the corresponding mean **percent signal change (PSC)** of the respective brain regions. All the code in this workshop will be using these PSC values for calculating feedback scores and displaying on the screen in different formats.

In MATLAB, go inside the MAC folder if you are using a MAC computer. If using a Windows computer, go inside the Windows folder. You will need to run all your MATLAB scripts from inside this folder in MATLAB.

Activity 1

In MATLAB, open *NeuroFeedback_simulation1_intermittent_ROI_tpl.m*

This script is used to calculate PSC of brain region 1, while controlling for the PSC from brain region 2 (reflecting physiological confound), and display feedback scores based on the resulting mean PSCs via a simple feedback bar display, intermittently after every meditation task block. Higher the PSC, higher the feedback score and vice-versa.

a) In *NeuroFeedback_simulation1_intermittent_ROI_tpl.m,* make your edits under the *Needs Change* section. Make the following changes:

```
TR = 1.0;

%Note: Folder/File names here should not have any numbers because it interferes
%with the rt_load_BOLD function

feedback_dir = [pwd '/Data/NeuroFeedback_Run_Two'];
feedback_file_name = 'NeuroFeedback_RunTwo';
run_no = 2;
```

After making the changes, run the script. Psychtoolbox will go through the paradigm, and display feedback intermittently after each meditation block. You can even try to follow the instructions on meditation yourself! This will give you a first-hand glimpse into what participants experienced during the meditation NF study inside the MRI scanner.

b) You can explore further, by changing the TR value, or even the file and folder names (to use data from other runs in the */Data* folder).

c) What happens when you decrease the TR? What happens when you increase it? Think about the implications of having a shorter vs. longer TR for neurofeedback.

Activity 2

In MATLAB, open *NeuroFeedback_simulation2_intermittent_FC_tpl.m*

This script is used to calculate **functional connectivity (FC)** between brain region 1 and brain region 2 during each meditation task block, and display feedback scores based on the FC values via a simple feedback bar display, intermittently after every meditation task block. Higher the FC, higher the feedback score and vice-versa.

a) In *NeuroFeedback_simulation2_intermittent_FC_tpl.m,* make changes under the *Needs Change* section similar to Activity 1. Set the TR to 0.8. Change the file and folder to point to data from run 3.

b) Scroll down towards the end of the script to the *calculate_feedback* function. You will edit parts of this function to calculate FC and estimate the feedback score based on the FC value.

Calculate FC using the corr() function in MATLAB.

```
%BOLD PSC values from target ROI 1
med_vals_roi1 = ROI_vals(first_tr-baseline_lag_dur+1:last_tr-baseline_lag_dur+1,1);

%BOLD PSC values from target ROI 2
med_vals_roi2 = ROI_vals(first_tr-baseline_lag_dur+1:last_tr-baseline_lag_dur+1,2);

%Calculating functional connectivity between roi1 and roi2
[FC_val,~] =
```

c) Now transform the *FC_val* to a feedback value, such that higher FC produces a higher feedback value. Note that the feedback bar has 20 levels. So you need to transform the FC values (ranging between -1 and 1) to a feedback score ranging from 0 to 20. The feedback score needs to be stored in the *curr_feedback* variable.

```
%Calculating functional connectivity between roi1 and roi2
[FC_val,~] =

%Feedback
%Higher feedback value for higher FC value or vice-versa (as desired)
curr_feedback =
```

Once you have made the changes, run the script. The feedback you see now would be from FC instead of PSC.

d) If you rerun the script with run 2, you will observe how the scores are different compared to Activity 1 with the PSC.

e) You can explore further by inverting the feedback-FC relationship, i.e., showing a higher feedback score for a lower FC value. This will be relevant for tasks that aim to train participants to decrease their FC between 2 specific brain regions.

f) Think about what you would need in the calculation, if you wanted to control for physiological confound before or while estimating FC.

Do you think the method of FC calculation presented here would work for paradigms with continuous feedback display (where the feedback display is updated every TR)? Why or why not?

Activity 3

In MATLAB, open *NeuroFeedback_simulation3_continuous_bar_ROI_tpl.m*

This script is used to calculate PSC of brain region 1, while controlling for the PSC from brain region 2 (reflecting physiological confound), and display feedback scores based on the resulting PSCs via a simple feedback bar display, that changes after every TR continuously during the meditation task block. Lower the PSC, higher the feedback score and vice-versa.

a) In *NeuroFeedback_simulation3_continuous_bar_ROI_tpl.m,* make changes under the *Needs Change* section similar to Activities 1 & 2. Set the TR to 0.8. Change the file and folder to point to data from run 4.

b) Scroll down towards the end of the script to the *calculate_feedback* function. You will edit parts of this function to calculate feedback for continuous real-time display.

You would need to update *current_psc* to reflect the PSC value of the latest TR, instead of the mean value from the whole block like in Activity 1. *resid_BOLD* contains the processed PSC signal of all TRs.

```
%Cumulative GLM - Regressing out detrended and demeaned confound from ROI so far
[beta,~,stats] = glmfit(all_conf_vals-mean(all_conf_vals),all_vals);
resid_BOLD = stats.resid + beta(1);

current_psc =
current_conf = 0;
```

c) Now transform the *current_psc* value to a feedback score ranging between 0 and 20. Here, we desire to show a higher score when the PSC value is lower (as we are training participants to deactivate the target brain region). Note that you need to use the *PSC_thresh* variable to calculate *curr_feedback*, since this *PSC_thresh* decides the range of *current_psc*. Unlike FC which has a fixed range of -1 to 1, PSC ranges can be altered as required. For this script, the default threshold set is -2 (i.e., we want feedback scores to increase for PSCs from 0 to -2).

```
%Feedback value:
%Higher negative feedback value implies greater deactivation
%Convert negative feedback value to positive feedback value in the
%bar
% 0 and +ve PSC = feedback value of 1
% -ve PSC = feedback value above 1

curr_feedback =  % deactivation
```

Once you have made the changes, run the script. You will now experience a paradigm where the feedback is continuously changing.

d) You can explore further by inverting the feedback-PSC relationship, i.e., showing a higher feedback score for a higher PSC value (training for activation of target region). You may also explore by changing the PSC threshold under the *Optional to change* section at the top of the script. Higher thresholds make the task harder for participants by scaling their scores down.

e) You can explore by reducing vs. increasing the TR in this script. You will notice how the feedback updates are quicker with a shorter TR. When you decrease the TR, you can increase the values of *block_dur_TR, rest_dur_TR and cue_dur_TR* under *Optional to change* to ensure that the overall paradigm still runs for a reasonable duration. Very short TRs would essentially mimic the experience of an EEG neurofeedback paradigm (which has a high sampling rate). Think about the implications of having a short TR in a continuous NF paradigm in terms of experimental power.

f) Did you find it hard to meditate and focus on your breath with the continuous NF display, compared to the intermittent NF display? How can the continuous paradigm be made more user-friendly for meditation tasks?

g) Think about how participants would always need to remember about the hemodynamic lag (of 7 s) while undergoing continuous NF training.

Activity 4

In MATLAB, open *NeuroFeedback_simulation4_continuous_image_ROI_tpl.m*

This script is used to calculate PSC of brain region 1, while controlling for the PSC from brain region 2 (reflecting physiological confound), and display feedback scores based on the resulting PSCs via a more complex feedback display. In this feedback display, the brightness of a relaxing image changes after every TR continuously during the meditation task block. Lower the PSC, higher the feedback score and hence brighter the image, and vice-versa. In other words, the relaxing image becomes clearer and easier to see when your meditative focus is higher.

a) In *NeuroFeedback_simulation4_continuous_image_ROI_tpl.m,* make changes under the *Needs Change* section similar to the previous Activities. Set the TR to 0.8. Change the file and folder to point to data from run 7.

Once you have made these changes, run the script. You will now experience a continuous adaptive NF paradigm where the brightness of a relaxing image will change based on the feedback score.

b) Scroll down to find the function *DrawFeedbackImage* (line no. 459). How would you edit this function to invert the relationship between image brightness and feedback score? Note that *feedback_num* is a feedback score between 0.1 and 1. In the current script, higher value leads to more brightness. You need to change something such that a lower value can lead to more brightness, and vice-versa.

```
function feedback_num = DrawFeedbackImage()
% Updates feedback screen based on latest score

%OUTPUT
%feedback_num - feedback value (between 0.1 and 1)

global window texture

%rect_num is the number of rectangles to color corresponding to the current feedback value
feedback_num = calculate_feedback();

%Changing the image opacity based on current feedback value
%Screen('DrawTexture', windowPointer, texturePointer [,sourceRect] [,destinationRect] [,rotationAngle]
Screen('DrawTexture',window,texture,[],[],[],[],feedback_num);
Screen('Flip',window);


end
```

c) Once you have made the change, run the script and observe if and how the image behaviour has been inverted.

d) **(Optional exercise)** Now you can change the functionality of the feedback display, by making it increase the redness of the image, instead of brightness. For this, you would need to make changes to *curr_feedback* in the *calculate_feedback* function, such that *curr_feedback* ranges between 0 and 255 (usual range of RGB values). Additionally, adapt the *if-else* condition (shown below) accordingly to reflect this range.

```
if curr_feedback<0.1
    curr_feedback=0.1;
elseif curr_feedback>1
    curr_feedback=1;
end

end
```

Then you would need to make changes in the *DrawFeedbackImage* function. Here, you would need to set the *modulateColor* option instead of the currently active *globalAlpha* option (see image below where the comment shows the function's template usage). The *modulateColor* option would be a vector of 3 values for Red-Green-Blue (RGB) (e.g., [200 10 10]), while the *globalAlpha* value could be fixed at 0.8 or 1 instead of *feedback_num*. How should you incorporate the *feedback_num* into the *modulateColor* setting?

```
%Changing the image opacity based on current feedback value
%Screen('DrawTexture', windowPointer, texturePointer [,sourceRect] [,destinationRect] [,rotationAngle] [, filterMode] [, globalAlpha] [, modulateColor]);
Screen('DrawTexture',window,texture,[],[],[],[],feedback_num);
```

Run the script after making the changes, and observe how the feedback interface now has a different impact on your meditation task performance.

e) Think about the implications of a complex feedback interface vs. a simple feedback interface in terms of user-friendliness and learning impact.

Activity 5 (non-coding)

Design a simple neurofeedback experiment for a problem statement you have in mind, from your field of interest/expertise.

What would the goal of your NF study be?

Identify the elements you would wish to incorporate for each of the following components in your NF design (refer to slides again for options and descriptions of each component):

1) **Source of neurofeedback** (what would you feed back to participants?)
2) **Control for confounding aspects** (what kind of controls would you include – control group, control for physiological confounds, etc.?)
3) **Feedback interface** (would the feedback be continuous, intermittent, complex, adaptive and/or simple?)
4) **Instructions to participants** (what would the task be?)
5) **Measurement of outcomes** (what are your key outcome measures?)

Activity 6 (Advanced and optional)

Adapt *NeuroFeedback_simulation4_continuous_image_ROI_tpl.m* to change the image fed back based on whether the ROI is activated (PSC is +ve) or deactivated (PSC is -ve). For example, for positive PSC, display and change the brightness of image 1 (higher PSC = higher brightness), and for negative PSC, display and change the brightness of image 2 (more negative PSC = higher brightness).

For this, you would need to download a new image off the internet, rename the image as required and save it in the *Data/* folder along with the existing *relaxing_image.jpeg*. You would then need to make necessary changes in the *calculate_feedback* and *DrawFeedbackImage* functions. You also need to uncomment and edit the following commented lines (for *img2* and *texture2*) shown below to incorporate the additional image:

```
%% Needs Change

TR =

%Note: Folder/File names here should not have any numbe
%with the rt_load_BOLD function

feedback_dir = [pwd '/Data/NeuroFeedback_'];
feedback_file_name = 'NeuroFeedback_';
run_no =

%Loading image
img = imread([pwd '/Data/relaxing_image.jpeg']);
%img2 = imread([pwd '/Data/image2.jpeg']);
```

```
120
121
122 -    % Getting image representation
123      texture = Screen('MakeTexture', window, img);
124      %texture2 =  Screen('MakeTexture', window, img2);
```

Ensure to incorporate the necessary modifications in *calculate_feedback* and *DrawFeedbackImage.* Run the script and observe the differences.

Activity 7 (Advanced and optional)

Adapt *NeuroFeedback_simulation4_continuous_image_ROI_tpl.m* to change the image feedback based on changing thresholds. For instance, if the PSC crosses an initial threshold of -1, change the image (image 2) and increase the threshold to -1.5. If the PSC further crosses this threshold of -1.5, increase it further and change the image (image 3). In all cases, the brightness of the image should continue to indicate the PSC change within the respective threshold.

Similarly, you can think of other ways to modify the provided template codes, and be creative with how you calculate and present your feedback.

Final Note: *Working_code_realtime_AdditionalResource/* contains the actual script that was used in the real-time meditation NF study (this is in Windows format), which was recently concluded. The script contains many comments explaining different parts of the code. This script will NOT work in simulation, as it was designed to work for actual NF in real-time at the MRI centre. In case you plan to develop an NF experiment of your own and run it in the MRI centre (MBCIU) with the existing NF setup, you can use this MATLAB code as a template to design your feedback presentation with Psychtoolbox.