

Groove iEEG

Experimental paradigm to study wanting to move and pleasure feelings while people listen to short musical patterns. The experiment is currently optimized to perform intracranial EEG. Levels of syncopation and harmony complexity are manipulated.

Rationale for the experiment

In the experiment, we present musical patterns to the participants lasting 10s each. The patterns consist of a repeated chord played with different rhythms on top of a metrical background (hi-hat). The rhythms have two levels of complexity (defined as syncopation) and the chords have also two levels of complexity (defined as consonance/dissonance). In previous experiments, it has been shown that participants like more and want to move more to patterns with medium rhythm complexity (i.e. not too predictable and not too complex). This effect is enhanced when chords have medium complexity (i.e. not too consonant and not too dissonant). These findings provide an empirical basis to the idea that there is a "sweet spot" in music appreciation and groove, where music with medium levels of complexity is preferred. The experimental paradigm here intends to further investigate this with iEEG.

Experimental paradigm

There are two blocks in the experiment. In one of them, on each trial participants hear the musical pattern and then rate how much they wanted to move (1 = not at all, 5 = very much). In the other block, they hear the same excerpts but instead have to rate how much they liked it. Listeners have up to 7 seconds to provide ratings. Blocks are counterbalanced. Two practice trials are played at the start of each block. There are 48 trials per block. Each level of complexity (medium or high) for either rhythm or harmony has 24 trials per block and 48 in the whole experiment. For the interaction between rhythm and harmony, there are 24 trials per design cell (e.g. harmony-low/rhythm-medium) in the whole experiment and 12 per block. The paradigm takes around 20-25 minutes. There is a break in the middle of each block and a pause between blocks.

The experiment is implemented in either Psychopy 3 (versions after 2021) or Psychopy 3.1.2 and can be run with the script "scripts/groove_harmony_iEEG.py" for Psychopy versions after 2021, or "scripts/groove_harmony_iEEG_3.1.2.py" for Psychopy 3.1.2 (this is the version in Knight-lab laptops). A Spanish version of the task can be run with the script "scripts/groove_harmony_iEEG_spanish.py" (only in Psychopy > 2021). Psychopy versions beyond 2021 are needed to send triggers. If instead Psychopy 3.1.2 is used, sound onsets need to be recorded in a different way (e.g., a photodiode) and set up in the system.

The stimuli are found under the "stimuli" directory. A list and metadata for the stimuli used in this experiment are found in "stimuli/stim_list.csv". This list is loaded and randomized in the Psychopy script. Names and data for other stimuli not included in this experiment are also found in "stimuli/Stim_Names.xlsx". Logfiles are saved under "logs".

A short online musical expertise questionnaire needs to be filled out before the task in the following links:

In English:

<https://survey.au.dk/LinkCollector?key=DZ73351SL632>

In Spanish:

<https://survey.au.dk/LinkCollector?key=WFV7599FUJCK>

A physical copy of the questionnaires can also be used.

Software requirements:

Linux and Psychopy 3 (versions above 2021 if triggers are needed or v3.1.2 if sound onsets are recorded otherwise. Knight-lab laptops support only v3.1.2).

Peripherals needed:

- Keyboard for participant responses
- Photodiode to record sound/screen onsets. This is crucial if information about sound onsets is not obtained otherwise.
- Either headphones or loudspeakers to present the stimuli.

Instructions for running the experiment

1. Open the link for the online questionnaire provided above and ask the participant to fill it.
2. To run the task, open Psychopy 3.1.2 (alternatively 2021.2.3 if appropriate):
 - Open the bash terminal and write 'conda activate david' (works for Knight-lab laptops)
 - Then type 'psychopy'
 - Alternatively, you can run the task directly from the command line (only for knight-lab laptops):
 - Open the bash terminal and write "conda activate david"
 - Navigate (cd) to the scripts folder and type "groove_harmony_iEEG_3.1.2.py".
 - Skip step 3 and go to step 4
 - For non-Knight-lab laptops, Psychopy may be installed in different places. Ask around.
3. In the coder, open the script "scripts/ groove_harmony_iEEG_3.1.2.py" (or alternative versions). Click on the "run" button and write the subject code in the pop-up window.
4. In the pop-up window, write the subject code. Leave the other two fields blank to run the task with default settings. Otherwise, if you want to change block order or skip the practice, then follow steps 6 and 7 below.
5. Write which block order to run. If blank, block order is chosen randomly. If "1", it runs first the liking block. If 2, it runs the wanting to move block first. Leave blank unless you want to start with a specific block, e.g., if the experiment crashed in the middle.
6. Write whether practice should be included (1 or blank) or not (0). Normally, practice should be included, but if the experiment crashes and you want to start from the main task, then write 0.
7. Press ok to start the experiment.
8. Once the experiment finishes, make sure to obtain the log files (custom and default).

Tips and caveats

Make sure that the participant understands the task well. Prompt them to ask as many questions as they like. Make sure they use the numeric keys in the keyboard to answer.

If the system requires triggers to keep track of stimulus timing, then make sure that the parallel port address of your machine is properly set in the script “triggers.py”. If your system does not record triggers, please make sure to record sound onsets otherwise (e. g. recording the sound signal, setting a photodiode). Make sure the photodiode fully covers the white square in the lower left corner of the screen.

Contact Details

David Quiroga

510-3656529

dquiroga@berkeley.edu or dquiroga@clin.au.dk