

Manual PyMTS

1° Step: Download the file and extract it into a folder named 'PyMTS'.

You cannot change the names of the files and folders inside the folder. They will be used by you to program the experiment.

2° Step: Add all the stimuli that will be used in the experiment to the "stimuli" folder. This includes instructions, model and comparison stimuli, consequences, and so on.

3° Step: Add all the trial block files to the "config" folder (for more information on how to write the 'csv' files, refer to the section "Setting up the trial blocks").

4° Step: Open the "configData.json" file and write on each line. Each line contains a first name in quotes (e.g., "instructions"). You cannot change any of these names. The description of what each of them means and how to fill in each line can be found in the section "Setting up the experiment."

5° Step: Double-click on the "PyMTS.exe" file, enter the participant's name, the experimenter's name, **select the stating block** and click on "enviar". The program should start shortly after.

6° Step: The session data file will be saved in the "data" folder with the name you provided for the participant. We advise you to delete this file from the folder after completing the experimental session **(if you use the same name for a file that already exists in the 'data' folder, the program will reset the first version of the file and you will lose your data).**

Setting up the trial blocks

For each training or testing block, you should create a CSV file. A CSV file can be created using Excel by saving it as "CSV (*.csv)". For example, if you want to present a training block for AB relations, a training block for AC relations, and a testing block for BC and CB relations, you should add three separate CSV files with different names (see example files).

In the folder you received, there is already a template file available for use. You should find ten columns named: "sample", "sample_sound", "comp", "correct_comp", "img_right", "img_wrong", "sound_right", "sound_wrong", "time_right", and "time_wrong". These names **cannot** be changed.

The lines **below** this first line represent the trials (each line is a trial), and the number of lines corresponds to the number of trials in each training/testing block (e.g., if it's a block of 24 trials, the CSV file should have a total of 25 lines - including the first line naming the columns). Each of the columns is explained below:

sample. In this column, you should enter the name of the file that will be presented as the visual model stimulus, including the file type. For example, if it is a '.jpg' file, you should include that information ("A1.jpg" - **without** the quotation marks).

sample_sound. In this column, you should enter the name of the file that will be presented as the auditory model stimulus, including the file type. For example, if it is an '.mp3' file, you should include that information ("sA1.mp3" - **without** the quotation marks). If no stimulus is presented, you can enter the letter "n" (**without** the quotation marks).

comp. In this column, you should enter all the comparison stimuli that will be presented (using the same formatting as the model - adding the file type to the name), separated by commas (e.g., "B1.jpg:B2.jpg:B3.jpg" - Pay attention to this part, you should

separate the names **ONLY** with colon, without adding a space between the colon and the stimulus name, and **without** quotation marks).

correct_comp. In this column, you should enter the comparison stimulus that will be considered the correct stimulus for that trial (e.g., "B1.jpg" - **without** the quotation marks).

img_right. In this column, you should add the image that will be presented in case the person selects correctly (the consequence for a correct response). Remember that PyMTS only accepts images, and the name should include the file type (e.g., "img_right.png" - **without** the quotation marks). If no stimulus is presented, you can enter the letter "n" (**without** the quotation marks).

img_wrong. In this column, you should add the image that will be presented in case the person selects incorrectly (the consequence for an incorrect response). Remember to include the file type (e.g., "img_wrong.png" - **without** the quotation marks). If no stimulus is presented, you can enter the letter "n" (**without** the quotation marks).

sound_right. In this column, you should add the sound that will be played in case the person selects correctly (the consequence for a correct response). Remember to include the file type (e.g., "sound_right.wav" - **without** the quotation marks). The sound should have the duration for which the consequence will be presented. If no stimulus is presented, you can enter the letter "n" (**without** the quotation marks).

sound_wrong. In this column, you should add the sound that will be played in case the person selects incorrectly (the consequence for an incorrect response). Remember to include the file type (e.g., "sound_wrong.wav" - **without** the quotation marks). The sound should have the duration for which the consequence will be presented. If no stimulus is presented, you can enter the letter "n" (**without** the quotation marks).

In the columns "sample_sound," "img_right," "img_wrong," "sound_right," and "sound_wrong," if you do not want to present any of the stimuli (or if it is a testing block),

you can enter the letter "n" (**without** the quotation marks) instead of the stimulus names. You should **not** delete the columns.

time_right. In this column, you should enter the duration for which the correct consequence will be presented. If no consequence is defined, the screen will remain black for the duration specified in this variable.

time_wrong. In this column, you should enter the duration for which the incorrect consequence will be presented. If no consequence is defined, the screen will remain black for the duration specified in this variable.

You **don't** need to randomize the trials as the program will do it automatically. Just make sure to write all the lines correctly. There's no problem with copying and pasting the lines. You can create one line as an example and then copy it, changing only the necessary information.

Setting up the experiment

The general settings of the experiment are presented in the "configData.json" file. Each line corresponds to a characteristic of the experiment and will be described individually below. The lines are already filled with templates that can be used to guide the programming. Do not change the configuration of the lines and follow the formatting established in the template file.

"screen_color". Here, you should enter the color that will be displayed as the background of the screen. To modify the color, simply change the numbers within the brackets (e.g., black = [0, 0, 0] and white = [255, 255, 255]). This is a variable that will be displayed in all trials.

"ITI". Here, you should enter the duration of the inter-trial interval (ITI) in seconds. If you don't want to present an interval between trials, simply enter the number zero. This is a variable that will be applied to all trials.

"start_block". This is a control variable. It presents the number of the block that will start when the program is initiated. The value zero represents the first block and so on. You don't need to modify this variable because you can select the starting block at the initial screen of the program.

"blocks". In this line, you should enter, within the brackets and separated by commas, the names of the CSV files that are in the "config" folder, in the order they will be presented. For example, if you want to present the blocks "treinoAB," "treinoAC," and "testeEq" in this order, you should write: ["treinoAB", "treinoAC", "testeEq"] (**with** the quotation marks). It is important to pay attention to the number of blocks, as this will be important in the following lines.

"instructions". In this line, you should add the instructions you want to present before each block. For example, if you want to present initial instructions before the first

block ("treinoAB") and before the test block ("testeEq"), you should write: ["inst1.jpg", "n", "inst2.jpg"] (**with** the quotation marks). Here, if you don't want to present an instruction, simply enter the letter "n" (**with** the quotation marks). The instructions should be presented in image format, and you should enter the filename of the image that will be used within the brackets. Remember that this line should have the same number of items as the "blocks" line. For example, if the "blocks" line has three items ("treinoAB", "treinoAC", "testeEq"), the "instructions" line should also have three items ("inst1.jpg", "n", "inst2.jpg").

"comp display". In this line, you should enter the type of MTS used in each of the blocks. If it is a Delayed Matching to Sample (DMTS), you should enter the value that will be presented in seconds. If it is a Simultaneous Matching to Sample (SMTS), you should write "SMTS" (**with** quotation marks). For example, let's say you want to use a 0-second DMTS in "treinoAB" and "treinoAC," and a SMTS in "testeEq," you should write: [0, 0, "SMTS"] (SMTS **with** quotation marks; numbers **without** quotation marks). Make sure to have the same number of items as in the "blocks" line.

"repetitions". In this line, you should write the number of repetitions that each block should have. Zero means that it will have no repetitions. For example, if you want "treinoAB" and "treinoAC" blocks to have a maximum of 3 presentations (maximum of 2 repetitions), and the "testeEq" block to have only one presentation (maximum of 0 repetitions), you should write: [2, 2, 0] (**without** quotation marks). Make sure to have the same number of items as in the "blocks" line.

"criteria". In this line, you should write the criterion to finish the block based on the number of correct trials the participant must achieve. For example, let's say in "treinoAB" and "treinoAC" (both with 24 trials), you want the participant to achieve 100% accuracy in the block, and in "testeEq" (also with 24 trials), you want the participant to achieve a minimum of 91.7% accuracy (equivalent to 22 trials). In that case, you should write: [24, 24,

22] (**without** quotation marks). Make sure to have the same number of items as in the "blocks" line.

"pos_sample". In this line, you should specify the coordinates for the position where the sample stimulus will be presented. The position [0, 0] represents the center of the screen, where the first item represents the X-axis, and the second item represents the Y-axis. For example, if you want to present the sample stimulus at the center of the X-axis but slightly above the center on the Y-axis, you can use something like: [0, 200] (**without** quotation marks). Feel free to adjust these values and test different positions as desired.

"pos_comps". In this line, you should specify the coordinates for the position where the comparison stimuli will be presented. For each comparison stimulus that will be presented, you need to add a position on the screen (following the same format of placing the two numbers inside brackets [X, Y]). For example, if you want to present two comparison stimuli on the left and right sides of the bottom of the screen, you can write: [[-350, -250], [350, -250]] (**without** quotation marks). Each position should be placed inside brackets, within the main brackets, and separated by commas. Feel free to adjust these values and test different positions as desired.

"stimulus_size". In this line, you should specify the size of the stimuli that will be used as the model and comparison. We recommend that you standardize the proportion of all stimuli used (e.g., 1:1). In this case, you should also enclose the values for the X and Y axes within brackets (e.g., [250, 250] - **without** quotation marks). Feel free to adjust these values and test the size you prefer.

"consequence_size". In this line, you should specify the size of the stimuli that will be used as consequences. We recommend that you standardize the proportion of all stimuli used as consequences (e.g., 4:3). In this case, you should also enclose the values for the X

and Y axes within brackets (e.g., [1800, 900] - **without** quotation marks). Feel free to adjust these values and test the size you prefer.

"instructions_size". In this line, you should specify the size of the stimuli that will be used as instructions. We recommend that you standardize the proportion of all stimuli used as instructions (e.g., 4:3). In this case, you should also enclose the values for the X and Y axes within brackets (e.g., [1200, 900] - **without** quotation marks). Feel free to adjust these values and test the size you prefer.

"end_text". In this line, you should enter the filename of the stimulus that will be presented when the person finishes the experiment or reaches the maximum number of repetitions you defined in the "repetitions" line. This file should be located within the "stimuli" folder (e.g., "end_text.png" - **with** quotation marks)

The visual stimuli such as cues and consequences can be created using programs like PowerPoint and saving the file as an image. All lines in the "configData.json" file should end with a comma (see the provided template file), except for the last line ("end_text"). You should not change the names of the lines, the indexing (the position and spacing of the lines as paragraphs) or remove the set symbols "{ }".

If you have any questions, please send an email to alceuregaco@ufscar.br and stay updated on the software's developments on the PyMTS GitHub page:

<https://github.com/AlceuRegaco/PyMTS>

Data generated by the program.

Participant. The name or code of the participant.

Experimenter. The name or code of the researcher.

Date. The date of the procedure.

Total_Trial. The total number of trials in the experiment.

Block. The name of the block to which that line belongs.

Block_Trial. The total number of trials in the block.

Accuracy. Whether the participant answered correctly (1) or incorrectly (0) in that trial.

Total_Correct. The accumulated number of correct responses in the block (this value resets when the participant repeats or reaches the end of the block).

Sample. The visual sample stimulus.

Sample_Sound. The auditory sample stimulus.

Comps. The comparison stimuli presented in their respective positions (e.g., if in the "pos_comp" line of the "configData.json" file you specified two spaces for presenting the sample stimulus – [[Position X], [Position Y]] – and the information saved is ['B1.png', 'B2.png'], it means that B1 was presented at Position X and B2 at Position Y).

Selected_Comp. The comparison stimulus selected by the participant.

Time_Click_Sample. The time until the participant clicks on the visual sample stimulus.

Time_Click_Sample. The time until the participant selects the comparison stimulus.

Time_Trial. The total time of the trial.