**README Instrumental Motivation Task**

Table of Contents

[Introduction 2](#_Toc27577610)

[Script description IMT\_main.m 2](#_Toc27577611)

[Part 1: Settings 3](#_Toc27577612)

[Part 2: Timings 5](#_Toc27577613)

[Overview other code sections 6](#_Toc27577615)

[Part 3: Input from console 6](#_Toc27577616)

[Part 4: fMRI settings 6](#_Toc27577617)

[Part 5: load required files 6](#_Toc27577618)

[Part 6: Psychtoolbox and screens 6](#_Toc27577619)

[Part 7: general image settings 6](#_Toc27577620)

[Part 8: WoF specific settings 7](#_Toc27577621)

[Part 9: VAS specific settings 7](#_Toc27577622)

[Part 10: Training 7](#_Toc27577623)

[Part 11: Input device unrelated variables 7](#_Toc27577624)

[Part 12: Input device unrelated variables 7](#_Toc27577625)

[Part 13: Input device unrelated variables 7](#_Toc27577626)

[Part 14: Text instructions 7](#_Toc27577627)

[Part 15: start fMRI procedure 7](#_Toc27577628)

[Part 16: actual trials 7](#_Toc27577629)

[Part 17: end of trial 8](#_Toc27577630)

[Part 18: save data 8](#_Toc27577631)

[Output description 9](#_Toc27577632)

[Supporting files 9](#_Toc27577633)

[Condition files 9](#_Toc27577634)

[Folder structure 10](#_Toc27577635)

[Instructions 11](#_Toc27577636)

[Jitters 11](#_Toc27577637)

[Training 11](#_Toc27577638)

[VAS/PANAS 11](#_Toc27577639)

[WoF 11](#_Toc27577640)

[References 11](#_Toc27577641)

Current version: IMT Version 1.0

# **Introduction**

The Instrumental Motivation Task is a cost-benefit paradigm that allows participants to work for food and monetary reward though effort in the form grip force. The output from this task allows for the estimation of effort vigor and maintenance. It is a task that can be used for investigating motivational processes.

The current version not only allows for grip force, it also enables the following major features:

*PET/fMRI Compatibility*

The grip force device is MRI compatible. The main task does not require visual input and therefore does not require a mirror.

Implementation of the Instrumental Motivation Task

The Script has to be run first in the Training mode (**runINDEX = 1), and for a second time, in the Experiment mode (runINDEX = 2).**

**The ‘Training’ Mode is comprised of 3 sections.**

1. **Calibration Phase:**

This sets the minimum and maximum effort via the GripForce-Device which is used to calculate effort and points during later trials.

1. **Cue Association Training**

The participant learns the tone-reward associations as they are repeatedly paired. This is repeated until a set performance across two consecutive query blocks is reached for the four different association.

|  |  |
| --- | --- |
| **Cue Learning** | **Cue Query** |
| A block of **learning** trials in which the tone (CS) is followed by the reward picture (US). The CS and US are presented, such that the CS precedes the US 1.5 seconds and CS and US are shown together for another 1.5 seconds. | A block of **query** trials in which the tones |

1. **Training of experimental task as it will be during the ‘Experiment’ Mode:**The task as it will be during the experiment is practiced (see Figure 1; cue phase, signal tone, bidding phase, and feedback). In addition, participants receive feedback whether their trial was valid (i.e., they pressed *after* *the* *end* of the signal tone) or invalid (i.e., they pressed *before* *the end* of the signal tone (threshold of set force is exceeded)).

**The ‘Experiment’ Mode reminds the participant of the learned tone-cue associations and follows for each trial the following order:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cue Phase (Tone): An auditory cue that was previously learned to be associated with a reward is resented. | Short signal tone (*Beep*1. ) is played to signal that the Bidding is about to start. | Bidding Phase: Participants can indicate via the GripForce-Device how much effort they are willing to exert for this given reward. | Jitter (jitter feedback start) | Feedback Phase: Based on the cumulative exerted effort (relative to their maximal possible Force) participants receive auditory feedback on the amount of reward points (cent or kcal points) they have won. | Jitter (inter-trial time) |
| Musiknote mit einfarbiger Füllung | Volumen mit einfarbiger Füllung |  |  | Chatblase Silhouette    You won **34 kcal-points**. |  |

Figure 1. Structure of one trial of the instrumental motivation task.

# **Script description Prep\_scanner\_volume**

This Script allows the experimenter to test the volume via the headphones in the scanner calling the prep\_IMT\_volume script which plays 3 different sound effects (Beep – Cue Sound similar to IMT – and synthezised sound). The Sounds are played until the experimenter confirm that the volume is set correctly.

# **Script description IMT\_main.m**

In principle, there are two parts in which settings need to be adapted to customize the IMT: the **settings section** and the **timing section** (Part 1 and Part 2). There are some other points that need to be taken into consideration, that are not part of these sections:

* Participant IDs are filled up with 0 at the front of the number, until the ID consists of a total of 6 numbers. This operation is done in *Part 3: input from console.*
* All supporting files and images are loaded into PTB in *Part 5: load required files* and should be adapted if files with different naming schemes are to be loaded (e.g. other images should be loaded).
* Text instructions that contain variable values (e.g., personalized food reward through the bidding phase) are, for a large part, not in the language file, and need to be changed within *Part 14: text instructions.* Similarly, other hard coded texts are present in section 17 and at the beginning of section 16 (16.01: breaks).

Before Running IMT.

## *Part 1: Settings*

This part has to be set in the Create\_SettingStructs.m. The code has to be run before starting the experiment and the IMT\_main loads the settings during the experiment directly.

The following will detail all settings and that can be used to customize the IMT script to ones needs. Settings are elaborated on in order of appearance in the script.

|  |  |
| --- | --- |
| **General Settings** | |
| settings.do\_fullscreen = 1; | 0 gives a small screen useful for debugging or testing. 1 Gives a full screen, which is automatically the second monitor (if one is connected) |
| settings.do\_fmri = 1; | Includes MR trigger and timing recordings if set to 1. |
| settings.do\_GFD = 1; | Uses grip force device as input device. No Input device established if set to 0. |
| settings.lang\_de = 1; | 1 sets the language of the instructions to German, 0 sets the language of the instructions to English. |
| settings.clckforce = 30000; | Force that needs to be exerted to click through the instructions before maximum/minimum calibration is performed. |
| **Settings specific for IMT** | |
| settings.value\_money | Cents earned with 100 money points |
| settings.value\_food | Kcal earned with 100 food points |
| Settings.calibration\_trials | Amount of training trials used to estimate the maximum/minimum effort. 3 for grip force. |
| settings.cue\_conditioning\_trials | Amount of cue conditioning learning trials (i.e., sound and image are paired in a CS-US fashion) for one learning block. Blocks are repeated until performance on query trials is |
| settings.cue\_conditioning\_test\_trials |  |
| settings.train\_trials |  |
| settings.trials |  |
| settings.cond\_performance |  |
| subj.version |  |
| subj.study = 'TUE008'; | (Short) indicator of project. Is used to load/save files with appropriate project names. |
| subj.runINDEX = '1'; | 1 for training the IMT. 2 for the actual experiment. |

*Part 2: Timings*

|  |  |
| --- | --- |
| **Determined during the Create\_Settings script for the whole study, loaded in the IMT main script.** | |
| timings.cue\_length = 3 | Length (in seconds) of the cue phase (tone) |
| timings.signaltone\_length = 0.3 | Length (in seconds) of the signal tone (beep) |
| timings.effort\_ length = 3 | Length (in seconds) of the bidding phase (input via GFD)) |
| timings.break\_length = 10 | Length (in seconds) of a single break |
| timings.feedback\_length= 4 | Length (in seconds) of the feedback (in total) |
| timings.number\_breaks = 1 | Number of breaks during the entire paradigm |
| timings.calibration\_length = 10 | Length (in seconds) of a calibration trial |
| timings.cue\_response\_wait = 4 | Length (in seconds) of the response during a query trial during the training |
| timings.avrg\_jttr\_feedback='2' | The average jitter time for the start of the feedback (time between bidding phase and feedback phase). Needs to be a string as it is used to load the correct jitter file. |
| timings. avrg\_jttr\_itt='2'; | The average jitter time for the start of the next trial (time between feedback phase and cue phase). Needs to be a string as it is used to load the correct jitter file. |
| timings. max\_jttr\_feedback  = '12' | The largest value the feedback jitter can take. Needs to be a string as it is used to load the correct jitter file. |
| timings.max\_jttr\_itt = '12' | The largest value the inter-trial-interval jitter can take. Needs to be a string as it is used to load the correct jitter file. |
| timings.break\_trials = 37 | Number of trials after which a break is happening |

# **Overview code sections IMT\_main.m**

## *Part 3: Input from console*

This part of the code has the following functions (in order):

1. Request subject and session information from experimenter (
   * Run: training vs. experiment
   * ID: participant ID
   * Session: session ID.
2. Convert this information to the appropriate formats (e.g. num2str, abbreviations etc.). Participants ID are filled with zeros at the front of the number to make a complete, 6 number ID.
3. Get operating system.
4. Cursor is hidden

## *Part 4: fMRI settings*

This part of the code specifies the triggers and other MR-specific variables.

## *Part 5: Psychtoolbox and screens*

This part sets all the PTB and screen settings, such as the colors that are used in the paradigm.

## *Part 6: general Stimulus settings*

This part sets the parameters for all the different element on the screen, that apply to both grip force and frequency tasks, such as the position of text, the position of the tube, the size of the ball etc.

## *Part 7: load required files*

This part of the code loads all the different supporting files. These are in order:

1. Condition files
2. Jitter files. These are only loaded correctly if the avrg\_jttr and max\_jttr are correctly filled (as strings!) in in *Section 2: Timings.* See ‘supporting files’ for more information on how to create jitters that are supported by the EAT\_main.m script.
3. Images. If any other images need to be loaded, the names of the images need to be adapted in this section.

## *Part 8: Calibration*

Grip Force Device Calibration to determine minimum and maximum effort.

## *Part 11: Input device unrelated variables*

Contains all the variables that are needed to run the EAT for both frequency-based and grip force-based input.

## *Part 12: Input device related variables*

**12.1:** Variables that are required when the frequency version is run, but not when the grip force version is run.

**12.2:** Variables that are required when the grip force version is run, but not when the frequency version is run.

## *Part 12: Cue Conditioning and Cue Performance Check*

**Runs the CueCond\_IMT script.**

## *Part 14: Text instructions*

Contains all the instructions that will be shown before the start of either the ‘real’ training, or the experiment. As of now, texts that have variable values in them are not loaded directly, so some of the texts need to be adapted within this section, if need be. This is something that will be updated in later versions.

## *Part 15: start fMRI procedure*

Listen for trigger and start getting MR-related timing values.

## *Part 16: actual trials*

**16.01:** starts break at the appropriate times. The text used for the breaks can be changed in this section.

**16.02:** update all the trial settings before trial start (conditions,.. ).

**16.03:** Cue phase (Sound + Image)

16.04: Instrumental Phase (bidding): Input query grip force device.

16.05: Feedback Preparation (win calculation)

16.06: Auditory Feedback

16.07: Save Backup Data at end of every trial

## *Part 17: end of trial*

Update the Maximum effort exerted if it is in the training phase. Prepare final feedback screen with wins. Show final feedback screen(s) with wins. Final feedback text can be adapted in this section. Screen is closed through mouse click.

## *Part 18: save data*

All data is finally saved here in the appropriate folders.

# **Output description**

The most important data structure is **output.data\_mat.** The variable names can be found in the struct **output.variable\_labels\_data\_mat**. The meanings of the labels (and thus also the meaning of the columns) are described below:

|  |  |
| --- | --- |
| **Label (sorted in order of appearance)** | **Description** |
| ID | ID |
| Session | Session |
| Run\_Index | Run Index (1: Training, 2: Experiment) |
| Trial | Trial number |
| Reward\_Money | Condition Reward type (0 = food, 1 = money) |
| Reward\_Mag | Condition Reward magnitude (1 = Low, 10 = High) |
| Maximum\_Effort | Max Effort (according to initial calibration) |
| Minimum\_Effort | Min Effort (according to initial calibration) |
| Time\_Abs | Absolute time of each trial |
| Force\_Abs | Absolute Force during each trial |
| Time\_Trial | Time relative to trial start (before Signal tone) |
| Force\_rel | Relative Force (i.e., relative to Max and Min Force as determined during calibration) (before Signal tone) |
| Time\_Trial\_aftBeep | Time relative to end of Signal tone |
| Force\_rel\_aftBeep | Relative Force (i.e., relative to Max and Min Force as determined during calibration) after Signal tone |
| AUC | Effort across 3s of the trial: Area under the Curve (AUC is calculated starting after the Signal tone). |
| AUC\_normed | Effort across 3s of the trial: Normalised Area under the Curve (AUC\_normed is calculated starting after the Signal tone). |
| Points\_won | Final points won for each trial. |
| Invalid\_effort | Stores whether a trial is counted as invalid (0 = valid, 1 = invalid) according to exceeding a relative effort threshold during the signal tone phase. |

Subj

|  |  |
| --- | --- |
| Version | Version of the Code |
| Study | Study Tag as character e.g., ‘TUE008’ |
| runINDEX | Run Index whether a training or experiment was conducted as character (‘1’: Training; ‘2’: Experiment) |
| subjectID | Subject identifier as character e.g., ‘000005’ |
| sessionID | Session number as character e.g., ‘000005’ |
| date\_start | Date and start time |
| Date | Date and start time formatted |
| Id | Subject identifier as number |
| Sess | Session number as number |
| runLABEL | Labels the runINDEX in language |
| Date\_end | Date and end time |
| Length\_exp | Date and end time formatted |

Output.win saves the final wins at the end of the IMT in coin points (sum\_coins) and corresponding money (money; in Euro) as well as final cookie points (sum\_cookies) and corresponding kcal (kcal). Output.win.per\_trial saves for each trial (columns) the points (1st row) and the respective reward type (2nd row; 0 = food, 1 = money) and reward magnitude (3rd row; 1 = low, 10 = high).

Output.rating.type saves the type of Input device that was used. Currently only implemented the Grip force device (‘bidding\_GFD’).

Saved during Training (specifically, calibration phase) under **effort**

|  |  |
| --- | --- |
| collectMax.maxEffort | Max Effort (minimum value of GFD) for 3 calibration trials |
| collectMax.values\_per\_trial | The rows:   1. Subject ID 2. 1:length Effort vector 3. Time referenced to calibration trial start 4. Absolute Force over time |
| collectMax.minEffort | Min Effort (maximum value of GFD) for 3 calibration trials |

Input\_device.mat

|  |  |
| --- | --- |
| Input\_device.maxEffort | Final Max Effort value |
| Input\_device.minEffort | Final Min Effort value |

Training Only

**Output.cue\_conditioning** with corresponding labels in output.variable\_labels\_cue\_conditioning

|  |  |
| --- | --- |
| ID | subject ID |
| Session | session |
| Run | run index (can only be 1, training)  (1 – training)  (2 – experiment) |
| Trial | Trial number (Trial will start from 1 with every Set). One Set will have trials: cue\_conditioning\_trials plus cue\_conditioning\_test\_trials  Thus for absolute trials repetition of Cue Conditioning Sets has to be taken into account. |
| CueCond\_Repeat | Counter of Cue Condition Repeats. Every repeat has number of trials as specifies in settings cue\_conditioning\_trials plus cue\_conditioning\_test\_trials |
| Reward\_Money | Reward Type, 0 = Food, 1= Money |
| Reward\_Magnitude | Reward Magnitude, 1 = Low, 10 = High. |
| Trial\_Query | Whether a Trial in the Cue Condition Srt is a Query Trial or not. 0 = Cue learning. 1 = Query |
| Query\_Mag | If a Trial is a Query Trial, what does it query? 0 = Reward Type (Food vs. Money), 1 = Reward Magnitude (Low vs. High) |
| Test\_Correct | Whether answer to query was evaluated as correct (1) or incorrect (0). Not, not pressing a key within a specified time window is evaluated as incorrect. |
| RT | Reaction Time for pressing a key. |
| Key pressed | Indicates whether a key was pressed within the response window (0 = no, 1 = key pressed) |

**Timestamps**

|  |  |
| --- | --- |
| From Calibration | |
| timestamps.calibration.exp\_on | Start Calibration |
| timestamps.calibration.exp\_off | End Calibration |
| From Cue Conditioning | |
| timestamps.conditioning.exp\_on | Start Cue Conditioning |
| Timestamps.conditioning.learning.start\_CS |  |
| Timestamps.conditioning.learning.start\_US |  |
| Timestamps.conditioning.learning.end\_CS\_US |  |
| timestamps.conditioning.query.start\_CS |  |
| timestamps.conditioning.query.end\_CS |  |
| timestamps.conditioning.query.start\_response |  |
| timestamps.conditioning.query.response |  |
| timestamps.conditioning.query.end\_response |  |
| timestamps.conditioning.query.start\_feedback |  |
| timestamps.conditioning.query.end\_feedback |  |
| timestamps.conditioning.query.response |  |
| timestamps.conditioning.exp\_off | End Cue Conditioning |
| From full trials (so Training and Experiment) | |
| timestamps.IMT.exp\_on | Start TRIALS |
| timestamps.IMT.start\_break | Start of Break |
| timestamps.IMT.end\_break | End of Break |
| timestamps.IMT.start\_CS | Start of Auditory tone signaling the reward type and condition |
| timestamps.IMT.stop\_CS | Stop of of Auditory tone signaling the reward type and condition |
| timestamps.IMT.start\_effort | Start of Bidding (GFD) recording (attention: participant is not yet instructed to bid but we record for early responses. |
| timestamps.IMT.start\_effort\_beep | Start Signal tone (beep) |
| timestamps.IMT.end\_effort\_beep | Stop Signal tone (beep), with this participant is instructed to start bidding. |
| timestamps.IMT.end\_effort | End of Bidding recording. |
| timestamps.start\_jitter\_effort | Start of Jitter between Effort and Feedback Phase |
| timestamps.end\_jitter\_effort | End of Jitter between Effort and Feedback Phase |
| timestamps.IMT.start\_feedback | Time when Reading out Feedback starts |
| timestamps.IMT.endvoice\_feedback | Time Feedback Voice reading out Points finished |
| timestamps.IMT.end\_feedback | Time Feedback finishes |
| timestamps.start\_iit | Start of Jitter between Feedback and next Cue Phase |
| timestamps.end\_itt | End of Jitter between Feedback and next Cue Phase |
| timestamps.IMT.exp\_end | End TRIALS |
| Additional Information saved: Jitter are in total the same for every participant, but shuffled for the trials in the IMT\_main script. Therefore, the exact vectors with the jitter information as well as total trial duration are saved. This can be used to compare the ideal timing with actual timing (e.g., due to hardware differences). | |
| timings.shuffled\_feedback\_jitter | vector with length (in seconds) of every trial’s feedback jitter. |
| timings.shuffled\_itt\_jitter | vector with length (in seconds) of every trial’s inter trial time jitter. |
| Timings.duration\_trial\_theoretical = [0.09, 0.23,..] | vector with length (in seconds) for every trial duration according to theoretical (i.e., ideal) calculation.  duration\_trial(each\_trial) = timings.cue\_length + beepLengthSecs + timings.effort\_length + 1 + feedback\_jitter(each\_trial) + timings.feedback\_length + 1 + itt\_jitter(each\_trial); |

**Timestamps\_relative**

**Identical in structure to timings, but here not system time is recorded, instead the relative timings to the start of the section (i.e., start of calibration; start of cue learning; start of IMT) are calculated.**

**MR Timings**

**The relative timings to the start of the MRI trigger are calculated for the IMT as well as the durations.**

|  |  |
| --- | --- |
| MR\_timings.dummy\_volumes | Number of dummy volumes as specified in the settings |
| MR\_timings.on\_trigger\_loop | Time Start Searching for Trigger |
| MR\_timings.trigger.all | Time of all Triggers (until Dummy volumes reached) |
| MR\_timings.trigger.fin | Time when the final Trigger was found |
| MR\_timings.onsets.break | Time between Final Trigger and Break Start |
| MR\_timings.durations.break | Duration break |
| MR\_timings.onsets.condition\_preview\_reward | Time between Final Trigger and Start of Reward Tone |
| MR\_timings.durations.condition\_preview\_reward | Duration Reward Tone Preview as |
| MR\_timings.onsets.effort | Time between Final Trigger and Start of Bidding Measurement |
| MR\_timings.durations.effort | Duration entire bidding measurement (from end of cue, via signal beep, to end of bidding phase) |
| MR\_timings.onsets.effort\_Beep | Time between Final Trigger and Start of **Signal tone** that indicates official bidding phase |
| MR\_timings.durations.effort\_Beep | Duration **Signal tone** |
| MR\_timings.onsets.effort\_aftBeep | Time between Final Trigger and Start of actual **bidding phase.** |
| MR\_timings.durations.effort\_aftBeep | Duration of actual **bidding phase.** |
| MR\_timings.onsets.jitter\_effort | Time between Final Trigger and Start of Jitter between Effort and Feedback |
| MR\_timings.durations.jitter\_effort | Duration of Jitter between Effort and Feedback |
| MR\_timings.onsets.feedback | Time between Final Trigger and Start of Feedback |
| MR\_timings.durations.feedback\_voice | Duration Feedback that is read out |
| MR\_timings.durations.feedback\_end | Duration Feedback phase in total |
| MR\_timings.onsets\_itt | Time between Final Trigger and Start of Jitter between Trials |
| MR\_timings.durations.itt | Duration for Jitter between Trials |

# **Supporting files**

## *Condition files*

Condition files are .mat files that currently follow the following naming convention, which is as follows for training files:

**EATTrain\_cond\_(subj.study)\_(PPID)\_S(sessID)\_R1.mat**

And as follows for “real” experimental trials

**EATExp\_cond\_(subj.study)\_(PPID)\_S(sessID)\_R1.mat**

In which subj.study is the abbreviation for the study (which can be adapted in *Part 1: settings*) PPID is the participant ID that is inserted in the console, and sessID is the session ID that is inserted in the console.

Condition files need four columns for them to be useful. The row index indicates the trial. The meaning of these columns is as follows

1. Difficulty of the trial
2. Type of reward (0 for food, 1 for money)
3. Magnitude of reward (1 for low, 10 for high, corresponds to value of a second of effort)
4. Uncertainty (0 for certain, 1 for uncertain)

Even if uncertainty is not desired, this fourth column should be filled with zeros to prevent the code from crashing. Multiple example scripts that create condition files can be found in the directory. They all start with: “Create\_Cond” followed by the number of trials and whether uncertainty is included or not. E.g. **Create\_Cond\_48T\_Cert.mat** creates condition files with 48 trials, but without uncertainty. **Condition files should be made before starting the experimental phase.** The naming of new scripts that create condition files is irrelevant to the functioning of EAT\_main.m.

The existing creation files require the following input to create compatible files:

* subj.studyID: The abbreviation for the project in which the task is used (e.g. TUE002)
* subj.study\_part\_ID: The session ID for which the condition files need to be made.
* start\_range: The first participant ID for which a condition file should be made.
* id\_range: The last participant ID for which a condition file should be made.

If these values are correctly inserted, all existing scripts should make condition files that are compatible with our naming conventions.

## *Folder structure*

The IMT will automatically create:

* A folder named **Backup** in which temporary and backup files are stored during the experiment.
* A folder named **Data** in which the final data files are stored.

The IMT further requires the following sub-folders to function:

* A folder named **conditions** with condition files per subject (created using Create\_Conditions.m).
* A folder named **jitters** which contains all the jitter files used in the experiment (Created during Create\_SettingsStruct).

**All images** need to be placed in the same folder as EAT\_main.m to be loaded. They **do not** go into a separate images folder. The images can be downloaded with the most current directory.

## *Instructions*

All the instructions that have numbers in them that update according to values that are updated within the paradigm (e.g. seconds) are still hardcoded within EAT\_main.m. This is an issue that is planned to be resolved in the next update. All instructions that do not take input during the paradigm are taken from instruction scripts *instructions\_german.*  
In these instruction scripts, the instructions are saved in the structure *instr.* Throughout the script, the DrawFormattedText function calls on the different strings in this structure to print out the instructions on the screen. Therefore, any new script should follow the names of the strings used in the existing language files.

## *Jitters*

New jitter files can be made in the end of the Create\_SettingsStruct.m that will call the DelayJitter function and will be saved in the *jitters* folder. This function takes 3 arguments:

1. Mu\_jitter: the average value for the jitter
2. Max\_delay: the longest length for the jitter
3. N\_trials: the number of trials for which the jitter is made

## *Training*

The training file is a .m file (TrainEAT\_main.m) that is called by the main file, to run the evaluation of the maximum effort. The first part of the script runs this estimation trials, the second part of the script runs a *bidding phase.* In both phases the structure of the code is very similar to EAT\_main.m, using the same procedures for ball movement unless otherwise indicated in the description below.

### Estimation trials

These are 2 or 3 trials (depending on the settings, see *Part 1: Settings*.) in which the participant is prompted to exert either maximum or minimum effort (just holding the device). In the case of 2 trials, we recommend using both to estimate the maximum. In the case of 3 trials the middle trial can be used for minimum effort estimation. This is only necessary if the device has an arbitrary minimum, like the grip force device that we are using to operate the paradigm. The execution of this section is like the execution of the main script, the main difference being that the bar in the tube moves upward, to indicate the maximum effort reached, rather than the minimum effort required to earn points. If the bidding phase is turned off, this part of the script will directly flow back to the main script to continue with the rest of the training, using the same code as the actual experimental run.

Relevant output is collected in the ‘collectMax’ structure, and includes the minEffort and maxEffort, used to determine the ball specifications in the following ‘official’ trials. It also contains the effort data from each of the estimation trials.

# **References**

Meyniel, F., Safra, L., & Pessiglione, M. (2014). How the brain decides when to work and when to rest: dissociation of implicit-reactive from explicit-predictive computational processes. PLoS Comput Biol, 10(4), e1003584. doi:10.1371/journal.pcbi.1003584 Meyniel, F., Sergent, C., Rigoux, L., Daunizeau, J., & Pessiglione, M. (2013). Neurocomputational account of how the human brain decides when to have a break. Proc Natl Acad Sci U S A, 110(7), 2641-2646. doi:10.1073/pnas.1211925110

Neuser, M. P., Teckentrup, V., Kühnel, A., Hallschmid, M., Walter, M., & Kroemer, N. B. (2019). Vagus nerve stimulation increases vigor to work for rewards. bioRxiv, 789982.