**TDT User Manual**

Table of Contents

[1. Introduction 2](#_Toc466145454)

[1.1 Code Files 2](#_Toc466145455)

[1.2 Backups 2](#_Toc466145456)

[1.3 General Outline of the Task 2](#_Toc466145457)

[2. Running the task 3](#_Toc466145458)

[2.1 Introduction 3](#_Toc466145459)

[2.2 Initial training 4](#_Toc466145460)

[2.3 Threshold Training 6](#_Toc466145461)

[2.4 Threshold Training Analysis 8](#_Toc466145462)

[3. The codes 8](#_Toc466145463)

[4. General remarks 9](#_Toc466145464)

[Appendix A - Psychtoolbox 9](#_Toc466145465)

# Introduction

The TDT (Texture Discrimination Task) has long been used in many ways. This manual is for using the TDT version designed by Yehuda Zur in 2016 using MATLAB environment at Nitzan Censor's lab, School of Psychological Sciences, Tel Aviv University. The requirements for running the task is only MATLAB environment with a Psychtoolbox installed (see appendix A).

## Code Files

The code files needed for running the task are:

* TDT.m
* TDT\_Target.m
* TDT\_Mask.m
* TDT\_Circle.m
* TDT\_Cross.m
* TDT\_Initial\_Training.m
* TDT\_Threshold\_Training.m
* InsideIntro.m
* TDT\_Threshold\_Training\_Analysis.m

Make sure these files are all in your MATLAB path for the codes to work properly.

## Backups

Another important thing to make sure is the backup folders exist. The first backup folder is the local backup folder located in each computer independently, and it should be "C:\Backups". The local backup folder will always be the first folder attempted to save the data in case of a network failure etc. The next backup folder is the network drive backup folder and it should be "Z:\Yehuda\Backups" in the Z network drive in the lab (if for some reason you are using this code outside Nitzan Censor's lab, ignore this folder altogether). This folder contains the backups of all the computers in the lab. If any of these two folders do not exist a warning message will appear in the beginning of the experiment and suggesting an option of canceling the experiment to first create/connect to these folders. The backup files are saved in multiple places along the codes to minimize data lose in case of a major problem. Every single file is saved with a date and time stamp within its name to prevent overwriting of files. Thus, every single backup file ever created will exist until manually deleted. The data is also saved in the current MATLAB folder under the subject name, and this file is the one with the purpose of being used.

## General Outline of the Task

The task has 5 possible screens:

* A cross fixation screen which displays between blocks.
* A circle fixation screen which displays before each trial.
* A target screen which contains a matrix of either vertical or horizontal lines, with 3 diagonally tilted lines positioned in a user-defined place, and a fixation letter in the center (either T or L).
* A mask screen consisting of a matrix of randomly rotated trigonal lines with an F letter in the center.
* A blank screen for receiving the subject's responses for the target screen.

A single 'trial' of the task is a circle fixation screen which disappears when the subject decides, followed by flashes of the target screen and the mask screen. Afterwards a blank screen appears which is the time for the subject's responses. The subject responses must be chronologically for the fixation letter and for the target alignment, and the subject must response twice before continuing to the next trial. A single block is a predetermined number of consecutive trials.

# Running the task

Set your MATLAB folder to the one you want the results to be saved in, since they will be saved in the current folder.

To start any of the parts of the task execute the command "TDT" in the command line. A selection box will open asking to choose one of the 4 functions:

* Introduction
* Initial Training
* Threshold Training
* Threshold Training Analysis

In choosing each of them a GUI will open with parameter tuning for the current subject. After choosing the options for the parameters warning messages will appear when needed, and then the experiment will begin. At the very beginning of each run a single wrong-fixation-letter sound (2 beeps) will sound in to test the sound driver.

## Introduction

The introduction is an endless loop of trials in which the target and mask screens do not flash. Instead the screens pass manually. The explanation is for the mouse as a response device and the (keyboard) in parentheses. Each screen is passed with the click of a mouse button (space bar), only the circle fixation is passed with the mouse wheel (space bar). In the blank screen after the mask screen the subject must response twice: once for the fixation letter and once for the target alignment. When responding to the fixation letter wrong, 2 beeps sound immediately after the first response. After both responses, the next trial will start with the circle fixation screen. To exit the introduction the esc key need be pressed at any circle fixation screen. No output data is saved in the introduction.

Parameter choosing screen:

All default values in this screen were chosen according to previous studies. In general, one should not change them and only insert the subject's information.

* The response device is either mouse or keyboard.
* The response keys are the keyboard keys for responding to the target screen. They must be provided when the response device is the keyboard and no key can appear in both. The number of keys mustn’t be equal.
* The screen is the screen used for the experiment. The choices are the available screens at the lab. This choice is for tuning the gamma correction value. A message will appear before the beginning of the experiment stating the stats of the chosen screen asking to make sure this is the correct screen and the brightness is adjusted.
* Number of lines is simply the number of background lines in the target screen (number of rows and number of columns).
* The relative size of the screen is a number between 0 and 1 defining the relative area of the screen on which the screens are displayed. This parameter is useful when screens are different sizes or the subjects sit at different distances from the screen and a specific visual angle is required.
* Screen height, width and refresh rate: this options are when the required parameters are different from the screen's current settings. If the settings chosen are not supported by the screen a warning message will appear asking whether to continue with the current supported settings or to abort. These values can remain empty when one wishes to preserve the current screen settings.
* Orientation is either '-' for horizontal background lines or '|' for vertical background lines.
* Target offset is the position of the center of the target within the matrix of background lines. The offset is positive to the right and down, and negative to the left and up, while the origin (zero position) is the very center, where the fixation letter is located. Since the target is 3 lines, the minimum offset is ±2 and the maximum is 1 less than the farthest line.

## Initial training

The initial training is the next step after the introduction and its goal is to familiar the subject with the timing of the task. In this training the subject is required to response to a certain percentage of the trials correctly, out of a certain number of trials. The subject has a maximum number of tries to do so. All trials have the same SOA timing (the time between the appearance of the target screen and the appearance of the mask screen). Operating the task is the same as in the introduction, only all the screens pass automatically (except for the circle fixation which passes with the pressing of the mouse wheel or the space bar).

At the beginning of the experiment a welcome screen displays. During this message, it is possible to press the esc key to exit the experiment without starting it. This option is directed for situations where the experimenter did not enter the correct parameters and wants to start over before continuing.

At the end of each try a message appears, which is directed to both the experimenter and the subject. When the message states "Press the mouse (or space bar) to continue (some number)", this means the subject failed to get the predetermined percentage. The number in the parenthesis is the number of correct fixation responses followed by the number of correct alignment responses. For example, (107) means 10 correct fixation responses and 7 correct alignment responses. During this message pressing the esc key will exit the training while saving everything in an output file. When the message states "Training finished." It means the maximum number of tries finished and the subject failed altogether. When the message states "Well done!" It means the subject succeeded in the last try. In each of these two messages the experimenter should press the esc key to exit the training, and an output file containing all the data will be saved.

During any circle fixation screen the experimenter can press the esc key to enter a within-training introduction. The introduction is identical to the one explained in 2.1, only when the experimenter exits the introduction the initial training continues from the spot in which it stopped. The amount of times the experimenter can exit to an introduction session is unlimited, and all the exists are recorded.

Parameter choosing screen:

All default values in this screen were chosen according to previous studies. In general, one should not change them and only insert the subject's information. The parameters explained here are only the ones special to the initial training. The rest of the parameters were explained in the introduction experiment section (2.1).

* Subject name/number could be whatever you want, and the output data will be saved in a file under that name/number. If you enter a name/number which already exists in the current folder a message will warn you from overwriting.
* The load button will load the subject's information from the previous session, if such exists. The session number will increase by 1 and the number of tries will be the number of tries left after the last session. If for any reason the session number already exists it will be overwritten and a warning message will appear.
* The day number, group number, sleep hours, age and gender are parameters which are not used in the experiment. They are only saved in the output file in case of anyone wants to investigate it in the future.
* The number of trials is per each try.
* The number of tries are the maximum times allowed for the subject to go through the initial training. When an existing subject is loaded, the number is automatically updated as the number of tries left. If none are left, the number is 0 and should be redefined.
* The SOA is the time (in milliseconds) between the appearance of the target screen and the appearance of the mask screen. It is the same time for all trials in the initial training.
* The target time is the time (in milliseconds) the target screen appears (within the SOA time)
* The percentages thresholds are simply the percentages the subject must succeed (together in one try) to complete the initial training.

Output file

The output file will be saved in the current MATLAB folder under the name "TDT\_Subject\_[subject name/number]\_Results.mat". The file will contain a MATLAB struct names InitialTraining. Inside the struct will be 4 other structs:

* "choose": a struct containing all the parameters chosen through the GUI per each session. This struct is loaded when starting an initial training session with an existing subject.
* "Parameters": a struct containing all parameters not chosen through the GUI per each session. The parameters are explained in detail in the code (each one at the point at which it is created). There also is a struct within the Parameters struct named "System" which contains information about the system and MATLAB on which the session was conducted.
* "Times": all timing parameters of the experiment per session. All values are in seconds, and are also explained in details inside the code where they are created.
* "Output": information regarding the performance of the subject in each session. The struct contains the following fields:
  + "Tries": number of tries conducted in the session.
  + "Displays": the fixation letter ('T' or 'L') and target orientation ('-' or '|') per each trial. Each column represents a trial and the 3rd dimension (page) represents the different tries. The columns are chronological.
  + "Responses": the responses of the subject. This field is arranged the same as the "Displays" field, so it is easy to compare the responses to the actual displays.
  + "Response Times": the timings of the responses. It is arranged the same as the "Displays" and "Responses" fields. Each number is the time (in seconds) passes from the disappearing of the mask screen until the specific response was clicked.
  + "Finished": either 'Yes' or 'No'. An indicator for whether the subject finished the session with success or not (or the experimenter exited during).
  + "SuccessPercentages": the percentages of correct responses per try for fixations and for alignments separately.
  + "TotalTrainingTime": the time of the total session.
  + "TrainingDateTime": a date and time stamp of when the session was performed.

If for any reason there is a blank line for a session in the structs that could have resulted from not entering the session number correctly (For example, after completing only one session entering session number 3, which will result in an empty 2nd line between the 1st and the 3rd lines, in all structs).

## Threshold Training

The threshold training is the task itself. It is outlined the same as the initial training with a few differences. There are no tries, but blocks. The number of blocks is predetermined regardless of the subject's performance. The SOA timing is not consisted, but chosen randomly for each trial along 14 optional timings: [340,300,260,240,220,200,180,160,140,120,100,80,60,40] (all in milliseconds). The number of trials in each block will always be a multiple of 14 so all SOA timings will appear the same amount of times. The 1st session is a "pre-training" session designed to familiar the subject with the short timings of the task compared to those of the initial training. Therefore, the default values in the GUI are always for this pre-training session which is a single block of 14 trials, one for each SOA, without re-training condition. When loading an existing subject with only one session (which should be the pre-training session) the GUI default parameters change to those of a normal session: 9 blocks of 28 trials each (2 for each SOA in each block) with re-training condition. When a subject who has done more than one session is loaded, his/her parameters are loaded and the re-training condition is turned off. Operating the task is the same as in the initial training. At the end of each block pressing the esc key will exit the session while saving all the data.

At the beginning of the experiment a welcome screen displays. During this message, it is possible to press the esc key to exit the experiment without starting it. This option is directed for situations where the experimenter did not enter the correct parameters and wants to start over before continuing.

At the end of the session there will be a message saying the session has finished. Pressing the esc key will exit that message and end the session. After exiting the experiment a question dialog will ask whether to perform the Weibull analysis or not. This analysis is usually done for each full block finished. If for any reason the analysis was not done or should be done again, use the Threshold Training Analysis code. The Weibull analysis graph is also saved as a MATLAB figure in the current folder under the name 'Weibull\_Graph\_Subject\_[subject's name/number]\_Session\_Number\_[number of session]', and backups of the figure will be saved as well.

There is no option for entering a within-session introduction in the threshold training like in the initial training.

Parameter choosing screen:

All default values in this screen were chosen according to previous studies. In general, one should not change them and only insert the subject's information. The parameters explained here are only the ones special to the threshold training. The rest of the parameters were explained in the introduction experiment section (2.1) and the initial training section (2.2).

* The number of trials is the number of times each SOA is presented in each block. That means the number of actual trials conducted in each block will be 14 times the number chosen here.
* The number of blocks has no limits and is usually 9.
* The re-training condition is a condition when chosen adds the possibility to exit the threshold training when the subject does not perform well enough. When chosen, if at any of the first 3 blocks there were more than one mistake in the longest 6 SOA's regarding alignment, the session ends. That is, for the SOA's [340,300,260] each one appears twice in each block. If out of the 6 responses for the target alignment for these SOA's in some block there were 2 or more mistakes, the session ends. At the end of each block a message states to press the space bar to continue. If the re-training condition has been selected and achieved, a message will appear stating "Block finished, please call the instructor". At that point only the esc key will work and will exit the task. The experimenter should understand at that point that the subject has met the re-training condition and should go back to initial training.

Output file

The output file will be saved in the current MATLAB folder under the name "TDT\_Subject\_[subject name/number]\_Results.mat". The file will contain a MATLAB struct names ThresholdTraining. Inside the struct will be 5 other structs:

* "choose", "Parameters" and "Times": identical to the structs in the InitialTraining struct, containing the same fields, all are explained in the code.
* "Output": information regarding the performance of the subject in each session. The struct contains the fields explained in the InitialTraining output before. We explain here the ones special to the ThresholdTraining struct:
  + "SOAs": the SOA timings (in milliseconds) in chronological order. Each row represents a different block, and each column a different trial.
  + "Successes": a binary array representing all responses, while 0 is a wrong response and 1 is a correct response. There are two values for each trial, top row for fixation letter and bottom row for target alignment. The columns are for the different trials in chronological order and the 3rd dimension is for the different blocks.
  + "SortedSuccesses": the same as successes only re-arranged so the columns are not chronological but sorted in the descending order of the SOAs. That is the first two columns in each block are for the 340 SOA and the last two are for the 40 SOA etc.
  + "SOASuccess": an array of the relative successes of each SOA within each block arranged in the way the PSY program arranges the results. Each row is for a different SOA. The left column for fixation and the right column for alignment. This array is the one upon which the Weibull analysis calculates its results.
* "WeibullAnalysis": The results of the Weibull analysis for each block. The analysis returns 4 values, and all are saved as fields in this struct.

If for any reason there is a blank line for a session in the structs that could have resulted from not entering the session number correctly (For example, after completing only one session entering session number 3, which will result in an empty 2nd line between the 1st and the 3rd lines, in all structs).

## Threshold Training Analysis

This code is designed to calculate and save a Weibull Analysis for a threshold training results file. To use the code, make sure the current MATLAB folder contains the output file. The code will ask for a subject name/number and for a session number. The name/number must be identical to the one in the output file's name. If the session already has a Weibull analysis, it will be overwritten. The analysis results will be saved within the output file in the struct ThresholdTraining.WeibullAnalysis, and a backup will be saved as well. The Weibull analysis graph is also saved as a MATLAB figure in the current folder under the name 'Weibull\_Graph\_Subject\_[subject's name/number]\_Session\_Number\_[number of session]', and backups of the figure will be saved as well.

# The codes

The codes basically use the Psychtoolbox command 'Flip' to change between screens in high timing accuracy. The screens are created as matrices the size of the screen resolution, that is every entry is the grey-level of a single pixel (0-255). The function codes TDT\_Target, TDT\_Mask, TDT\_Circle, and TDT\_Cross return the matrices according to the parameters they receive as inputs. The task codes are divided into sections as follows:

Get parameters through GUI: this section is responsible for the GUI in each task. The first lines define the details of the screens in the lab, please make sure you update these values in all codes whenever needed, so the colors the code produce will be as accurate as possible. The next lines define the default values for the parameters in the GUI in a struct called 'choice', which is later saved as 'choose' in the output file. The GUI itself is a while loop, which repeats itself if the values in the GUI are not perfectly legal for running the task. Every loop the subfunction 'choosedialog', which is written at the bottom of each code, is called upon. The subfunction defines the GUI graphically.

Parameter tuning: this section defines all the parameters of the experiment which are not in the GUI. Those parameters were left out from the GUI since it was decided that they are not frequently changed as the ones in the GUI. All of them are changeable though in this section as well. Each one has a short explanation of its purpose and a default value updated to a date written in the beginning of the section. Please make sure to update the date and default values when changing any of these parameters permanently. Some of the geometry-related parameters are explained in more detail in the functions TDT\_Target, TDT\_Mask, TDT\_Circle, and TDT\_Cross where they are being used.

Set up the experiment: setting up parameters which are not changeable, since they are calculated from other parameters mainly. Also, this section sets up and opens the onscreen window which will be used throughout the whole task. Nothing in this section should be changed without being 100% sure of the change.

Run experiment: the task itself, written as a loop for each try/block, and an inner loop for each trial. These loops are not a short code, and are not so trivial to understand. The most difficult part to understand is probably the part which receives the subject's responses. Nothing in this section should be changed without being 100% sure of the change.

End the experiment/Training: after finishing the task, this section closes the onscreen window, organizes the output file to save with all its contents and saves it. In the threshold training code, this section also calculates and saves the Weibull analysis. Nothing in this section should be changed without being 100% sure of the change.

# General remarks

If the code ever crashes do not panic, it happened not once in the past. Simply restart the MATLAB or even the whole computer and it should work. It is also recommended to run the task once (even up to the welcome screen and then exiting with pressing the esc key) right before a subject begins, to reduce the chance of crashing. If you're working in Nitzan Censor's lab and using the network drive Z, make sure you are connected before opening MATLAB.

The recommended thing for beginners who want to familiar themselves with the task is to first run the introduction task while changing the parameters in the GUI and in the 'Parameter tuning' section (just make sure to change the values back to default when you finish experimenting). After understanding the geometry of the task and the purpose of the parameters move on to do the same in the initial training or threshold training to familiar yourself with the timing of the task.

If you wish to change anything within the code other than in the first two sections, firstly make sure you're 100% sure of what you're doing. To fully understand the code before changing it, it is recommended to first read the function codes TDT\_Target, TDT\_Mask, TDT\_Circle, and TDT\_Cross to understand how they work. After that go to the 'run experiment' section and read the section from the inside loop to the outside (the inner trial loop, and then the outer try/block loop). The rest of the sections are less important and easier to understand.

I tried to make the code as documented and organized as possible. If there are any questions, feel free to contact me at yehuda.zur.89@gmail.com.

# Appendix A - Psychtoolbox

To install Psychtoolbox go to <http://psychtoolbox.org/download> and follow the instructions for your operating system. The codes were written and tested using Windows 7 operation system, with Psychtoolbox version 3.0.12 and 64-Bit GStreamer 1.0 version 1.7.91.

If you want to investigate and study the Psychtoolbox commands, go to: <http://docs.psychtoolbox.org/Psychtoolbox> where all the commands are documented and explained.