Zero Drift DDM GUI Documentation

Version 3.0

UNFINISHED DRAFT

Wenqing Tu, Oberlin College

October 15, 2025

Abstract

This document provides comprehensive documentation for the Zero Drift DDM GUI, a MATLAB-based tool for simulating and analyzing decision-making processes. The GUI features dynamic reward rate visualization, choice distribution plots, and various customization options. This guide covers installation, usage, and troubleshooting.

Acknowledgements

I would like to thank...

Contents

Chapter 1

Introduction

The Zero Drift Drift-Diffusion Model (DDM) GUI is a versatile and user-friendly interface designed to simulate and analyze decision-making processes under various dynamic reward conditions. This MATLAB-based tool allows users to explore the effects of reward rates, boredom functions, and other key variables on decision outcomes and behavior patterns.

Version 3.0 introduces several enhancements, including improved plotting capabilities, expanded import/export functionalities, and a more intuitive layout to streamline the simulation process. Whether you are a researcher exploring computational neuroscience or a student learning about decision-making models, this GUI provides a robust platform to conduct experiments and visualize results in real time.

This documentation offers a comprehensive guide to installing, navigating, and utilizing the Zero Drift DDM GUI. From a detailed overview of the user interface to examples of typical workflows, we aim to provide everything you need to leverage this tool effectively.

Chapter 2

Installation

2.1. Install MATLAB

1. Visit https://www.mathworks.com to purchase, download, and install MATLAB.
2. Ensure the following toolbox is installed and licensed:
   - **Statistics and Machine Learning Toolbox**

2.2. Download and Extract the GUI

1. Obtain the GUI files from the designated source (e.g., GitHub or an institutional website).
2. If the files are compressed in a `.zip`, extract them to your desired directory.
3. Verify that all required files (e.g., `.m` and `.mlapp` files) are intact.

2.3. Add the GUI Folder to MATLAB Path

1. Open MATLAB.
2. Navigate to the **Home** tab and select **Set Path**.
3. Click **Add Folder** and select the directory containing the GUI files.
4. Click **Save** to finalize the path addition.

2.4. Run the GUI

You can launch the GUI in two ways:

1. Open MATLAB, type the following command in the Command Window, and press **Enter**:

   ```
   ZeroDriftDDM2
   ```

2. Alternatively, locate the `ZeroDriftDDM2.mlapp` file in your file explorer (Finder for

macOS or File Explorer for Windows) and double-click it to open.

  - Note: Before using this method, ensure MATLAB is open and the GUI folder is correctly added to the path to prevent potential bugs.

2.5. Verify Installation

1. Launch the GUI and run a test simulation with default settings.
2. Confirm that plots and interactive elements function as expected.

2.6. Troubleshooting

- If the GUI fails to launch, verify the folder is added to the MATLAB path.
- Ensure all required toolboxes are installed and licensed.
- Consult the **Support** section for additional help.

Chapter 3

Getting Started

Follow these steps to start using the Zero Drift DDM GUI effectively.

3.1. Launching the GUI

1. Open MATLAB and ensure the GUI folder is added to the MATLAB path.
2. Launch the GUI using one of the methods outlined in the **Installation** section.

3.2. Understanding the Interface

The GUI consists of:

- **Main Settings Panel:** Configure simulation parameters (e.g., block number, choice number).
- **Visualization Area:** Displays dynamic plots, decision time distributions, and reward rate distributions.
- **Interactive Elements:** Buttons, sliders, and menus to control simulations.
- **Menu Bar:** Options for exporting figures for saving or further interaction, including features for saving 3D plots.

3.3. Running Your First Simulation

1. With the GUI open, verify the default settings are loaded.
2. Click **Run Simulation** to execute a test simulation.
3. Observe the outputs, including:
    - **Random Walk Plot:** Displays the decision process.
    - **Dynamic Plot:** Shows evolving reward rates and thresholds.
    - **Choice Proportion Plot:** Summarizes decision patterns.

3.4. Saving Results

1. Use the **File** menu to select **Save Results**.
2. Choose a directory and confirm the file is saved as a `.mat`.

3.5. Exploring Advanced Features

- Enable batch simulations using the checkbox in the settings panel.
- Customize parameters like cost models and decision thresholds.
- Use the **Import Data** tab to analyze pre-simulated datasets.

3.6. Tips for Beginners

- Start with simple settings to learn the interface.
- Access the Functionality section for parameter explanations.
- Save configurations frequently.

With these steps, you're ready to explore the Zero Drift DDM GUI. Happy simulating!

Chapter 4

User Interface Overview

4.1. Interactive Elements

4.1.1. Axes

While most axes in the GUI are non-interactive, you can use the options in the upper-right corner for basic interactions. For advanced interaction, use the *Tool* → *Pop-out Figure Menu Bar* to open the figure in MATLAB. (See Section 4.1.6)

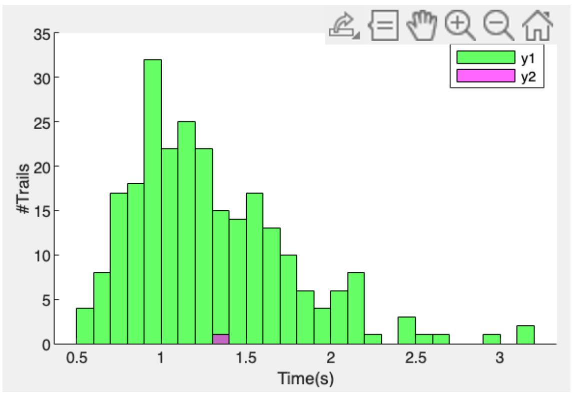


Figure 4.1: Axes Interaction Example.

4.1.2. Button

Press a button to execute the associated action.

Figure 4.2: Example of a Button.

4.1.3. Check Box

Toggle the check box to enable or disable specific features.

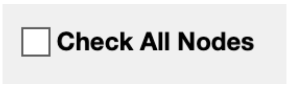


Figure 4.3: Example of a Check Box.

4.1.4. Drop Down

Select an option by clicking the drop-down menu.

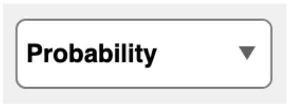


Figure 4.4: Drop-down Menu Example.

4.1.5. Edit Field (Numeric)

Enter a single numeric value within the specified range.

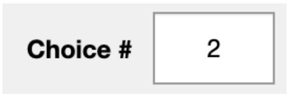


Figure 4.5: Numeric Edit Field Example.

4.1.6. Menu Bar

The GUI features three main menu options: File, Tools, and Help. Note that not all submenus are available in every window.

- **File**:
  - Import existing simulation data or batch files
  - Export figures to a specified directory
- **Tools**:
  - Batch file generator
  - Pop-out figures for separate viewing
  - Axis customization:
    * X-axis: Toggle between W_s and Tau display
    * Y-axis: Switch between linear and logarithmic scales
  - Data smoothing options
- **Help**:
  - Link to this documentation

4.1.7. Slider

Move the slider to adjust its value. For values outside the sliders range, enter them directly into the associated numeric field.

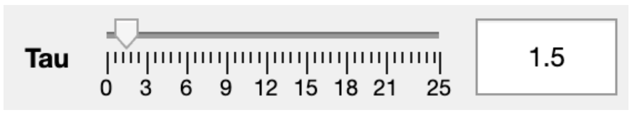


Figure 4.6: Slider Example.

4.1.8. Text Area

Provide input based on the displayed instructions. You may enter multiple values in the text area, separated by commas. Alternatively, you can use a specific convention for entering values, such as:

- zeros(1,20) for a row vector of 20 zeros,
- ones(3,4) for a 3x4 matrix of ones,
- 1 : 20 : 100 for a sequence from 1 to 100 with a step size of 20.

These conventions allow for more flexible input formats that the system can interpret.

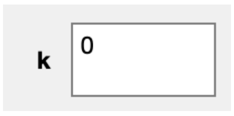


Figure 4.7: Interactive Text Area Example.

4.1.9. Tree (Check Box)

Check or uncheck nodes to customize selections.

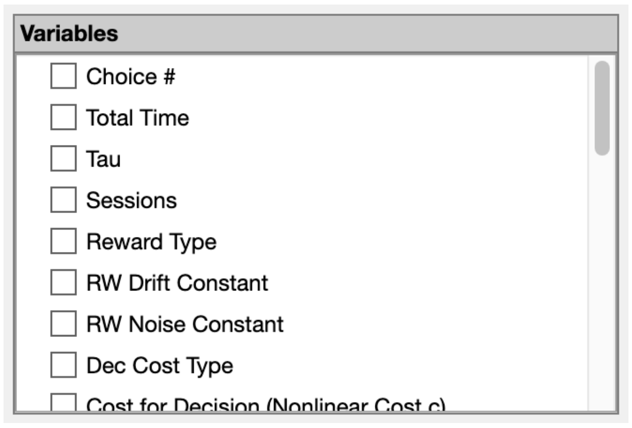


Figure 4.8: Tree with Check Boxes Example.

4.2. Noninteractive Elements

4.2.1. Progress Gauge (Linear)

Displays the progress of the current simulation.

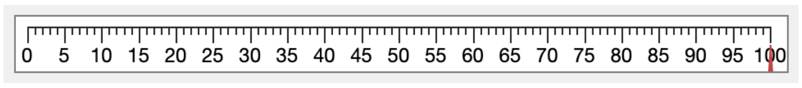


Figure 4.9: Linear Progress Gauge Example.

4.2.2. Lamp

Indicates simulation status:

- **Gray:** Idle state.
- **Green:** Simulation in progress.
- **Red:** Error.

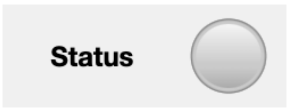


Figure 4.10: Lamp Indicator Example.

4.2.3. Text Area

Displays read-only information such as block time, settings summaries, or total rewards.

Figure 4.11: Non-interactive Text Area Example.

Chapter 5

Functionality

The GUI's complete functionality is organized across multiple specialized windows. To properly access all features, always launch the application through `main.mlapp` and access other windows from within this main interface. The following sections detail each component.

5.1. Main Interface: `main.mlapp`

This primary interface allows users to:

- Configure various simulation parameters
- Adjust plot settings
- Import existing datasets
- Generate files for batch processing

At the top of the interface (shown in Figure 5.1), you'll find three key controls:

- **Plot Button**: Initiates new simulations. Detail see in Section 5.2. ***Important*: Do not use this button for visualizing existing datasets.
- **Mode Selection Dropdown**: Offers two operational modes:
  - *Run Mode*: Single execution with one W_S value through the entire time course.
  - *Stimulation Mode*: Multiple executions with various W_s values, potentially repeating the time course.
- **Quit Button** (grayed out initially): Becomes active only after starting a simulation and remains available until completion.

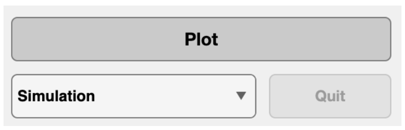


Figure 5.1: Main interface controls showing the Plot button, mode selector, and Quit button.

5.1.1. General Settings

Under the Setting tab, you can adjust values using the sliders or by directly entering desired values into the editable fields with a white background. Fields with a gray background are read-only and are automatically calculated based on other variables.

- **Choice Number:** Specifies the number of available decision alternatives in the simulation.
- **Total Time:** Defines the duration of each simulation run.
- **Block Number:** Determines the number of distinct reward conditions per simulation. During reward switching, probabilities/magnitudes undergo circular shifting with random shift values, ensuring the new condition never matches the previous one.
- **Tau (τ):** Time constant governing the decay rate of the reward rate estimator.
- **Self-excitation Weight (W_s):** Controls the self-excitation strength for reward rate estimators across the n choices.
- **Effective Time Constant (τ'):** Derived parameter calculated as $\tau' = \tau/(1 - W_s)$, representing the modified time constant accounting for self-excitation.
- **Time Step (dt):** Discrete time increment used for numerical simulation.
- **Sessions:** For batch processing, specifies the number of independent simulations performed with identical parameter settings.
- **Initial Action Rate:** Sets the starting value for the action rate at simulation onset.
- **Initial Reward Rate:** Defines the initial reward rate estimate at the beginning of the simulation.
- **Arbitrary Decision:** This feature allows you to enforce a specific decision at any given time, overriding the system's normal decision-making process. The desired choice is specified in the provided text area, following the format described in Section 4.1.8. To apply the same choice to all designated time points, simply enter a single choice value.

5.1.2. Random Walk

Defines the movement of a simulated dot ($\mathbf{V}_t$) with the following stochastic differential equation:

$$d\mathbf{V}_t = (\lambda \mathbf{V}_t + \rho \mathbf{R}_t)\ dt + \sigma\ d\mathbf{W}_t$$

where $\mu(V_\alpha(n), R_t, c_\mu)$ is the drift function given by:

$$\mu(\mathbf{V}_t, \mathbf{R}_t) = c_\mu \mathbf{R}_t$$

Here:

- $\mathbf{V}_t$ represents the state variable (the position of the decision particle) as a column vector in n dimensions for n choices

- $\mathbf{R}_t$ is the vector of the agent's reward rate estimates for each choice at time t, also a column vector of n rows
- ρ is a scaling constant for the drift, called "Drift Constant" in the app
- $\mathbf{W}_t$ is a Wiener process (standard Brownian motion).

This continuous-time system is simulated in discrete time, with time step size of Δt, as follows:

$$\mathbf{V}_{\text{new}} = \mathbf{V}_{\text{old}} + \Delta t \cdot (\lambda \mathbf{V}_t + \rho \mathbf{R}_t) + \sigma \sqrt{\Delta t}\, N(0,1)$$

where $N(0,1)$ refers to a standard normal random variable

If the dot crosses a threshold, the model treats it as a decision. Thresholds α are calculated using:

$$\alpha(d, \mathbf{R}_t, \alpha_m) = \min\left(\frac{d}{\max(\mathbf{R}_t, 0)}, \alpha_m\right)$$

$$\theta_i(d, \mathbf{R}_{t_i}, \theta_m) = \min\left(\frac{d}{\max(\mathbf{R}_{t_i}, 0)}, \theta_m\right)$$

where:

- d is the threshold scaling parameter.
- $\mathbf{R}_t$ is the reward at time t, with $\max(\mathbf{R}_t, 0)$ ensuring that the reward is non-negative.
- α_m is the maximum allowable threshold.

A choice that is not selected for an extended period results in its reward rate approaching zero, causing its threshold distance to approach infinity. To mitigate this, a maximum threshold distance is set relative to the starting coordinate to prevent instability during reward switches.

5.1.3. Boredom Function

`Reward Type` drop down allows selection of the reward mode:

- **Value 1:** Probability Mode The probability of receiving a reward (magnitude = 1) depends on the Boredom function.
- **Value 3:** Magnitude Mode The decision always yields a reward, but its magnitude is determined by the Boredom function.

The reward probability or magnitude $\mathbf{B}$ is defined as:

$$\mathbf{B}(k, \mathbf{A}(x), b) = \min\left(-k\mathbf{A}_t + b, B_{\min}\right)$$

where the Action Rate $\mathbf{A}_t$ evolves over time according to the following dynamics:

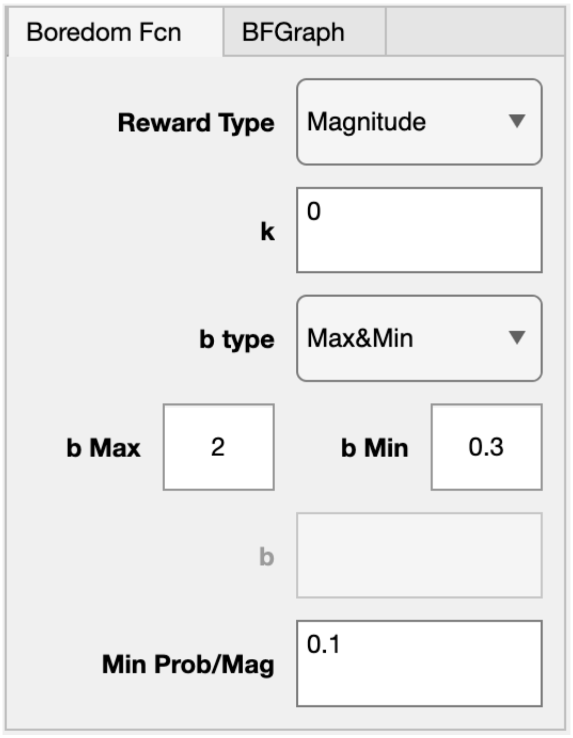
$$\mathbf{A}_{t+\Delta t} = \begin{cases} \dfrac{1}{\tau}(1 - W_s)\left[1 - \mathbf{A}_t \Delta t\right], & \text{if } V_t \geq \alpha, \\[2ex] -\dfrac{1}{\tau}(1 - W_s)\mathbf{A}_t \Delta t, & \text{if } V_t < \alpha. \end{cases}$$

You can enter multiple values for k, individual b, or minimum probability or magnitude values in their respective text fields, separated by commas.

For the `Max & Min` option in the b dropdown, the values of b will be distributed evenly between the minimum and maximum based on the number of choices.

- b **Type 1: Max & Min.** When selecting this option, you can enter the maximum and minimum values for b, and the system will automatically generate evenly distributed b values based on the specified number of choices.
- b **Type 3: Individual values.** In this case, you can manually input individual b values, separated by commas.

The corresponding Boredom Function will be plotted under `BFGraph` tab. You can use this plot to verify and adjust the settings.



Reward Type: Magnitude
k: 0
b type: Max&Min
b Max: 2
b Min: 0.3
b:
Min Prob/Mag: 0.1

Figure 5.2: Sample Boredom Function Configuration. The plotted function will help verify the entered settings.

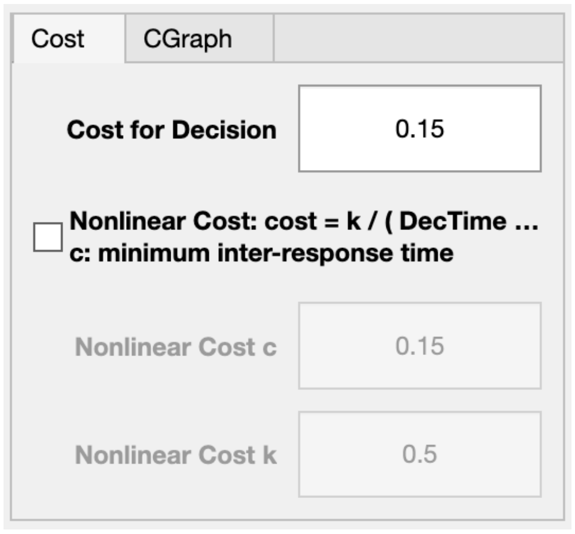
5.1.4. Cost

The cost function can be either linear or nonlinear, as defined by the following equations:

$$\mathcal{C}(T, k_{\mathcal{C}}, c_{\mathcal{C}}) = \begin{cases} \dfrac{c_{\mathcal{C}}}{T}, & \text{(Linear case)} \\[2ex] \dfrac{k_{\mathcal{C}}}{T - c_{\mathcal{C}}}, & \text{(Nonlinear case).} \end{cases}$$

where T is the decision time, and $k_{\mathcal{C}}$, $c_{\mathcal{C}}$ are parameters that shape the cost function.

To switch between the two cost functions, click the corresponding checkbox. The cost function will be plotted under the `CGraph` tab for visual reference.


Cost | CGraph

Cost for Decision: 0.15

☐ Nonlinear Cost: cost = k / (DecTime ...
c: minimum inter-response time

Nonlinear Cost c: 0.15

Nonlinear Cost k: 0.5

Figure 5.3: Sample Cost Function Settings.

5.1.5. Timer

The timer dynamics follow a random walk, which can be described in two cases:

$$d\mathcal{T} = \sigma_{\mathcal{T}} d\bar{W}_t + \begin{cases} \mu_{\mathcal{T}}^c dt, & \text{(non-adapting)} \\ \\ \mu_{\mathcal{T}}(c_{\mathcal{T}}, k_{\mathcal{T}}) dt, & \text{(adapting)} \end{cases}$$

where $\mu_{\mathcal{T}}(c_{\mathcal{T}}, k_{\mathcal{T}})$ is the adaptation function, given by:

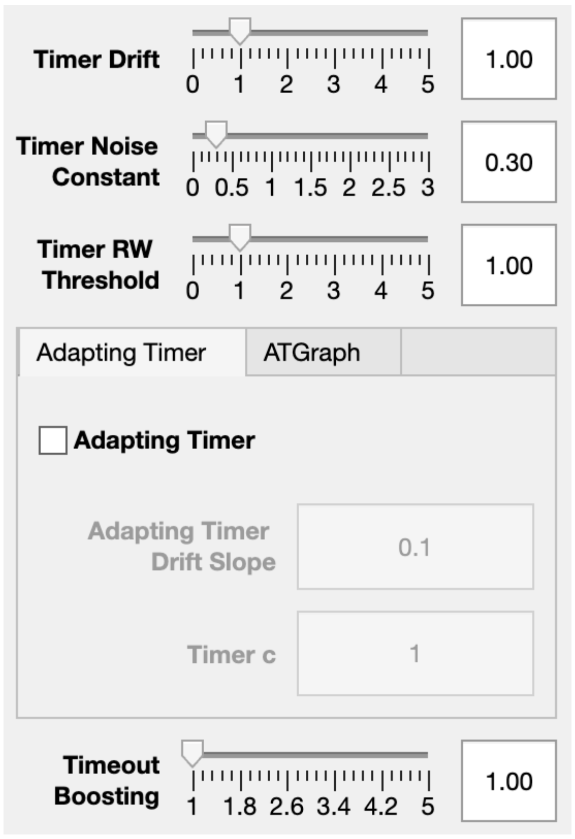
$$\mu_{\mathcal{T}}(c_{\mathcal{T}}, k_{\mathcal{T}}) = \max\left(\frac{1}{c_{\mathcal{T}}} - \frac{1}{k_{\mathcal{T}} \max\left(R_t, 0\right) + c_{\mathcal{T}}}, \mu_{\mathcal{T}_{\min}}\right)$$

This function models the timer's dynamics, allowing it to adapt to the reward function R_t.

In the ****non-adapting**** case, the timer evolves at a constant rate determined by $\mu_{\mathcal{T}}^c$. In the ****adapting**** case, the timer rate depends on the reward function R_t and two additional parameters, $c_{\mathcal{T}}$ and $k_{\mathcal{T}}$, which control the speed of adaptation. This ensures the timer dynamically adjusts to variations in the reward function, providing a more responsive system. The graph of the relationship between sum of reward rate and timer drift is depicted under the `ATGraph` tab.

A ****crossing**** in the timer's random walk results in a ****timeout****, indicating that the decision process has reached its time limit.

The ****timeout boosting**** feature allows the system to shrink the thresholds by scaling them down by a common factor whenever a timeout occurs. This ensures that decisions are made within a reasonable time frame, particularly in the case of prolonged inaction.


Timer Drift 1.00

Timer Noise Constant 0.30

Timer RW Threshold 1.00

Adapting Timer | ATGraph

☐ Adapting Timer

Adapting Timer Drift Slope 0.1

Timer c 1

Timeout Boosting 1.00

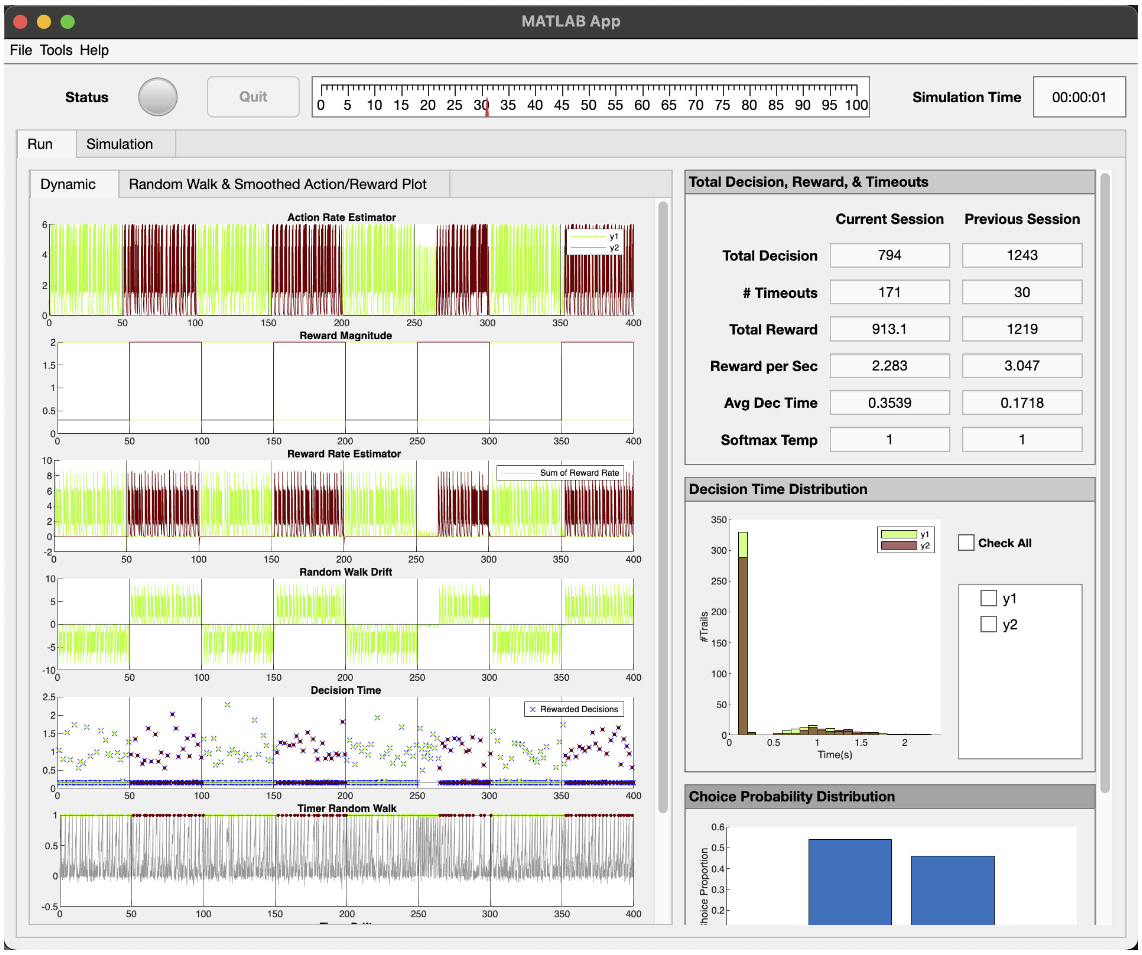
Figure 5.4: Sample Timer Settings.

5.2. Running new Run/ Simulation and Plot

To initiate a new simulation, click the `Plot` button located at the top of the main interface. This action will open a new window showing the plotted results.

On the top of the new window (shown in Figure 5.5), you will find a status lamp that indicates the current state of the simulation (See Section 4.2.2 for details). The `Quit` button will become active once the simulation starts and will remain available until the simulation is complete. Progress gauges will display the completion status of the current session and the overall batch process. Simulation time will be shown in the upper-right corner.

Below the status indicator, you will find two tabs: `Run` and `Simulation`. This GUI won't automatically adjust to the tab you are on when you click the `Plot` button. Please make sure you redirected to the correct tab to view the results you want.



Simulation Time 00:00:01

Total Decision, Reward, & Timeouts

	Current Session	Previous Session
Total Decision	794	1243
# Timeouts	171	30
Total Reward	913.1	1219
Reward per Sec	2.283	3.047
Avg Dec Time	0.3539	0.1718
Softmax Temp	1	1

Figure 5.5: Sample New Run Window.

5.3. Importing Datasets for Visualization

To import a simulated dataset or utilize existing datasets for plotting, navigate to the `File` → `Import Datasets` tab located in the upper-left corner of the GUI. This will direct you to open new windows for data import, where you can manage datasets for your visualization tasks.

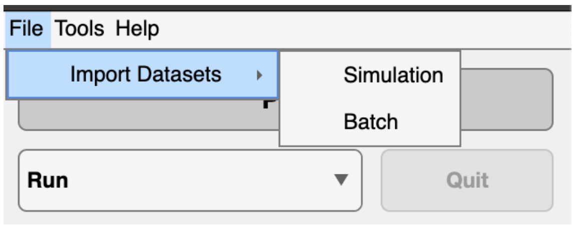
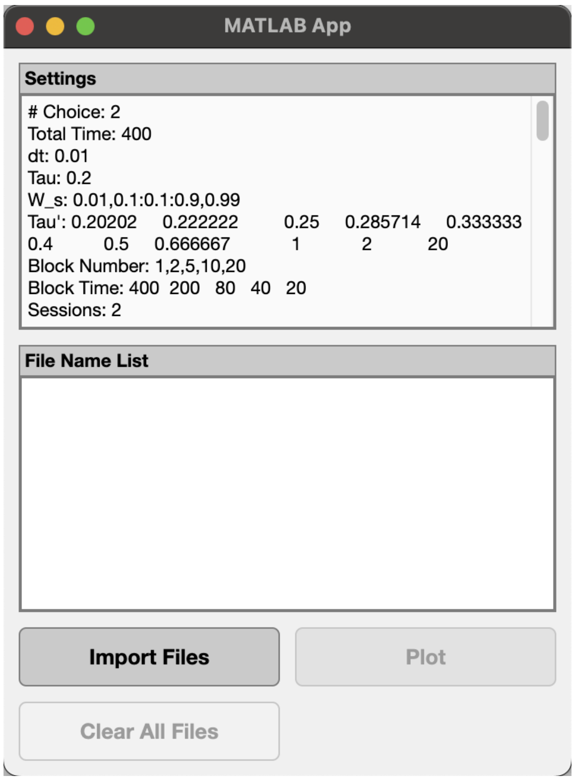


Figure 5.6: Click on this tab to go to saving & importing page.

5.3.1. Import Simulation

To import existing simulation from a dataset, nativate to the `File` → `Import Datasets` → `Simulation` tab. No setting adjusting can be done in this window. The upper `Settings` text area will show all the setting copy from your `main.mlapp` settings when you initially open this window. These will be the criteria for filtering datasets for import. Only the dataset comtaining the exactly same settings will be importted. Simply click on the `Import Files` button and select a directory to complete import. Note that this won't go through the file under subfolder.

When there's some files successfully imported, both `Clear All Files` and `Plot` button will be available. You may click on `Clear All Files` button to clear all the datasets you imported. Clicking on `Plot` button will lead to open another window for plotting using existing datasets. See Section *** for more detail.



```
# Choice: 2
Total Time: 400
dt: 0.01
Tau: 0.2
W_s: 0.01,0.1:0.1:0.9,0.99
Tau': 0.20202    0.222222    0.25    0.285714    0.333333
0.4    0.5    0.666667    1    2    20
Block Number: 1,2,5,10,20
Block Time: 400  200  80  40  20
Sessions: 2
```

Figure 5.7: Window for importing simulation.

5.3.2. Import Batch

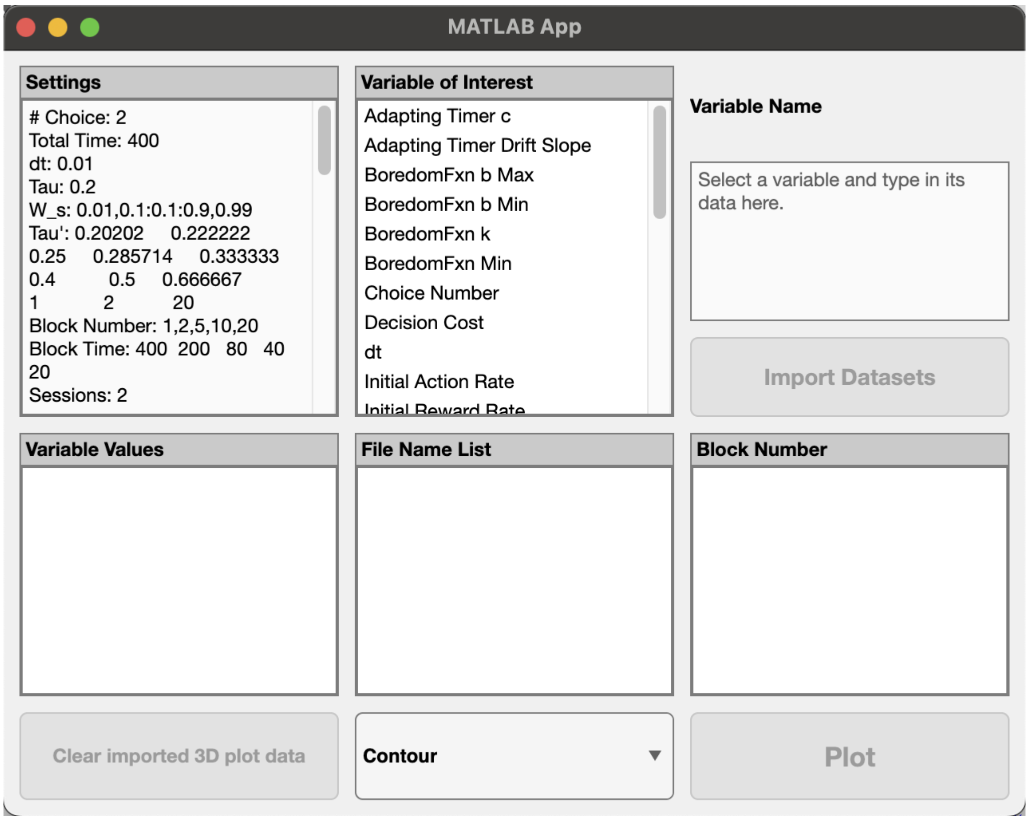
To import existing batch (multiple W_s, multiple values for another parameter, with multiple sessions for each condition) from a dataset, nativate to the `File` → `Import Datasets` → `Simulation` tab.

Similar as Section 5.3.1, you cannot edit individual parameter here but please go back to your `main.mlapp` to edit the settings. Here, you may select one parameter from the upper middle list box for having multiple values. Specific values need to be entered on the right side. Click on `Import Datasets` button below to select a directory for completing import similar to Section 5.3.1.

If at least one file for each values are imported, you will be able to check the each variable and their corresponding file name at the lower left and middle list boxes. You may also clear all the files through the `Clear imported 3D plot data` button on the bottom left.

Before you click on the `Plot` button, check the block number you want to plot, and you may switch the plotting type between `Contour` and `Waterfall` through the drop down.

Clicking on `Plot` button will also lead to open another window for plotting using existing datasets. See Section *** for more detail.



```
# Choice: 2
Total Time: 400
dt: 0.01
Tau: 0.2
W_s: 0.01,0.1:0.1:0.9,0.99
Tau': 0.20202    0.222222
0.25    0.285714    0.333333
0.4        0.5    0.666667
1          2         20
Block Number: 1,2,5,10,20
Block Time: 400  200   80   40
20
Sessions: 2
```

Figure 5.8: Window for importing simulation.

5.4. Multiple Settings

Since the GUI cannot terminate an ongoing simulation and may encounter unexpected errors, the Multiple Settings section allows you to create a `.mat` file for multiple simulations. You can then use the `ZeroDriftDDM_MultipleSettings.m` script to import the simulated `.mat` file and run the simulations in batches over an extended period.

To create a batch, navigate to the `Tools` → `Batch file generator` tab located in the upper-left corner of the GUI. This will open a new window for generating the settings file.

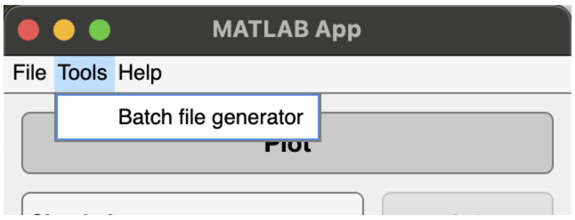
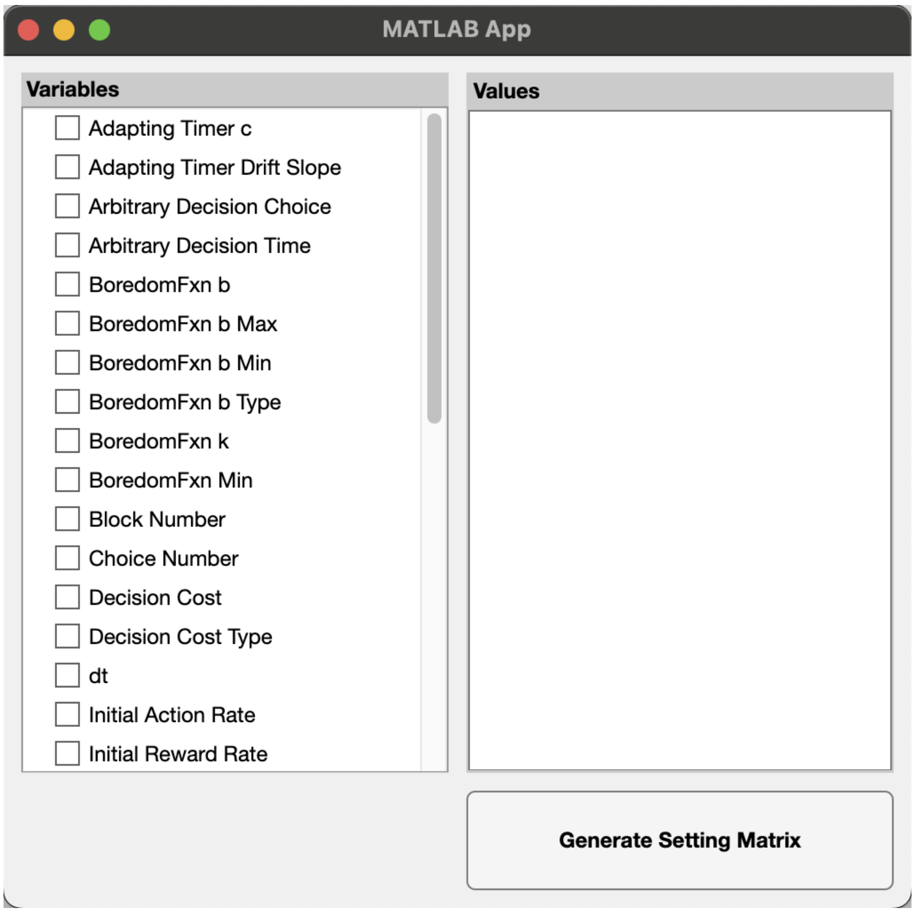


Figure 5.9: Batch File Generator Window.

Here you need to specify multiple values for the selected variables. All unspecified variables will retain their values as set in the general settings.



- [ ] Adapting Timer c
- [ ] Adapting Timer Drift Slope
- [ ] Arbitrary Decision Choice
- [ ] Arbitrary Decision Time
- [ ] BoredomFxn b
- [ ] BoredomFxn b Max
- [ ] BoredomFxn b Min
- [ ] BoredomFxn b Type
- [ ] BoredomFxn k
- [ ] BoredomFxn Min
- [ ] Block Number
- [ ] Choice Number
- [ ] Decision Cost
- [ ] Decision Cost Type
- [ ] dt
- [ ] Initial Action Rate
- [ ] Initial Reward Rate

Figure 5.10: Multiple Settings Panel

For most variables, enter the values on the line following the commented variable name in the values section (e.g., % Choice #). Separate different values with commas.

For the Boredom Function parameters k, b, and y minimum values, you can either enter a single value per condition or multiple sets of values. Each set of values should be placed on a separate line. Within each set, values should be separated by commas.

- Ensure that if you enter individual values, the number of values within each set matches the number of choices.

Once you have entered all the values, click the "**Generate Setting Matrix**" button to create the `.mat` file. The GUI will prompt you to choose the file path and name. Click Save to complete this step.

Next, use MATLAB to navigate to the folder where `ZeroDriftDDM_MultipleSettings.m` is located. Ensure that the `.mat` file you just created is also saved in the same folder.

To start running the batch, enter the following command in the MATLAB command line:

```
ZeroDriftDDM_MultipleSettings(filename.mat)
```

Replace `filename.mat` with the actual name of the settings file you created.

The `ZeroDriftDDM_MultipleSettings.m` script will display progress updates in the command line during execution, allowing you to monitor the status of the process. The progress will include the elapsed time, the current progress percentage, and the estimated time remaining.

Note: The estimated time remaining may not be accurate for simulations with longer total times or a larger number of total dots in later stages of execution.

```
Elapsed time is 662.488123 seconds.
Progress: 11.11% | Elapsed Time: 00:11:02.52 | Estimated Time Remaining: 01:28:20.13
```

Figure 5.11: Example Progress Command Line.

5.5. Batch Plot - Histogram

The histogram panel, located under the "`Batch`" → "`Histogram`" tab, enables you to analyze the distribution of sessions for each W_s and the selected value of variable of interest. The GUI also displays the mean of the distribution, providing a clear view of its shape and characteristics. This feature helps in assessing variability and trends within the dataset.

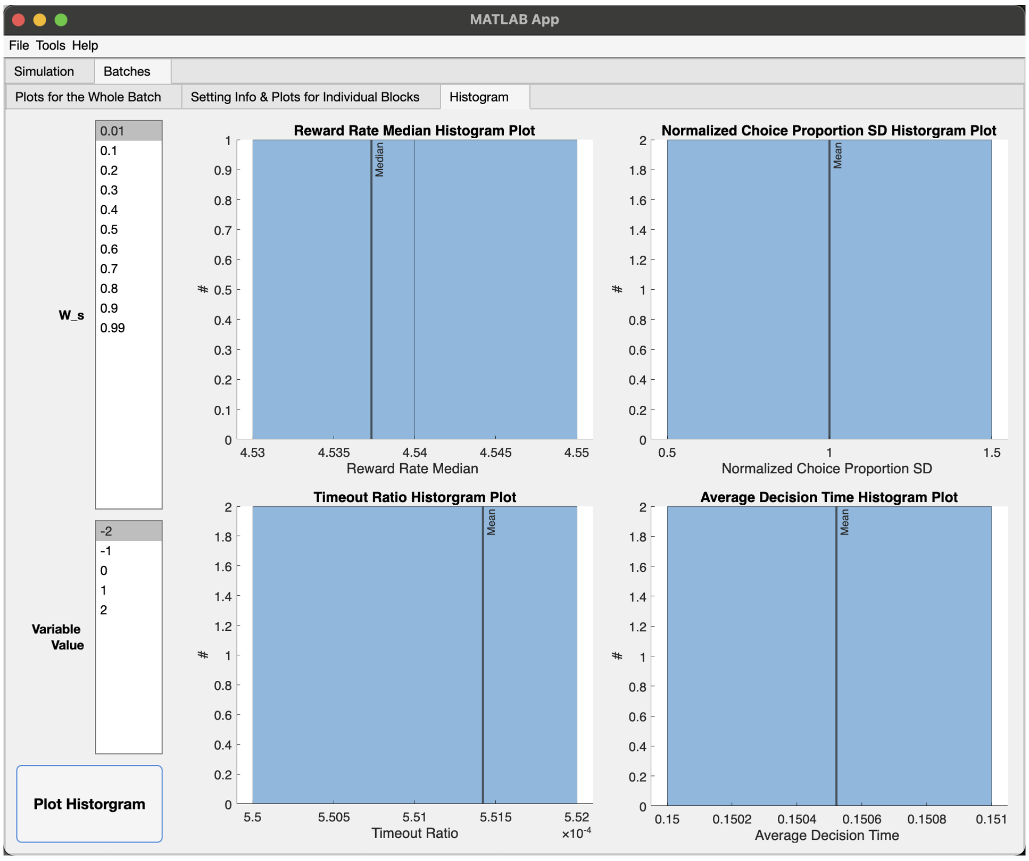

MATLAB App
File Tools Help
Simulation
Batches
Plots for the Whole Batch
Setting Info & Plots for Individual Blocks
Histogram
W_s
0.01
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.99
Variable Value
-2
-1
0
1
2
Plot Historgram
Reward Rate Median Histogram Plot
Median
Reward Rate Median
Normalized Choice Proportion SD Historgram Plot
Mean
Normalized Choice Proportion SD
Timeout Ratio Historgram Plot
Timeout Ratio
Average Decision Time Histogram Plot
Average Decision Time
#

Figure 5.12: Example of 3D Histogram Plots.

5.6. Menu Bar

5.6.1. Pop-out Figure

The menu bar allows you to select a figure of interest. By clicking on the desired figure, the GUI will open it in a separate MATLAB figure window, enabling further interaction and exploration. If the plot does not display correctly or appears distorted, resizing the figure window should resolve the issue. **Note:** The colorbar will not be included in the new figure window.

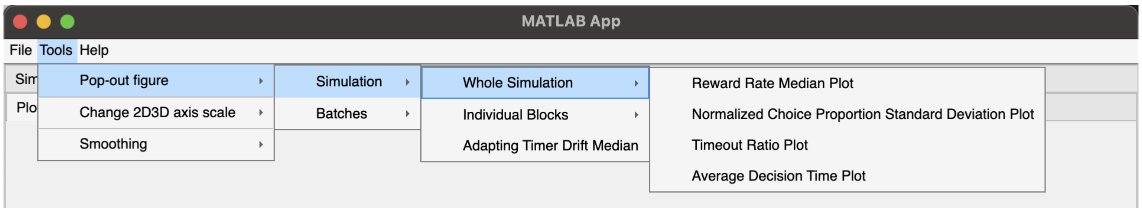


Figure 5.13: Pop-out Figure Menu Bar.

5.6.2. Save Run Results

To save the results of a single simulation run, use the keyboard shortcut **Command + S** (macOS) or **Ctrl + S** (Windows). Alternatively, navigate to **`File`** → **`Export`** → **`Run`** in the menu bar.

When initiated, the GUI prompts for a destination directory. Upon selection, the system automatically creates a timestamped folder and saves all results within it, including a `.mat` file containing the simulation settings and numerical results.

Figures are exported in both `.fig` (interactive) and `.png` (static) formats to support different use cases.

Important notes:

- Smoothed action rate and reward rate plots are not saved, as smoothing is applied only for visualization purposes. To preserve smoothed plots, use the popout window function and save them manually.
- The raw data used for plotting is not included in the export. Only the final figures and numerical results are saved.

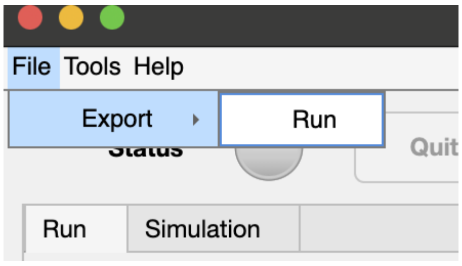


Figure 5.14: Save Run Figures menu option in the File menu.

5.6.3. Save Batch Figures

You can quickly save all Batch plots across all block numbers as separate `.fig` files using the shortcut **Command + S** (on macOS) or **Ctrl + S** (on Windows). Alternatively, you can access this functionality through menu **`File`** → **Export Whole Batch Plots**.

Upon triggering this action, the GUI will prompt you to select a folder. Once a folder is chosen, the following will be generated and saved within a newly created subfolder:

- `.fig` and `.png` files for each block number, providing both interactive and static formats for the plots.

- A movie file illustrating the progression of changes across different blocks.
- A `datasets` folder containing all datasets used to generate the plots.

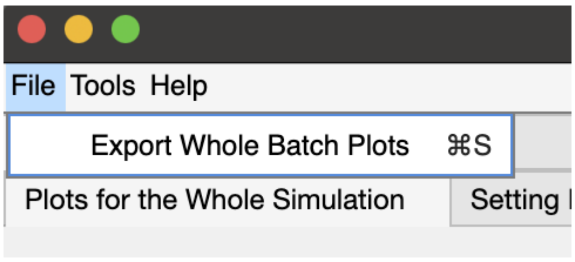


Figure 5.15: Save Figures Menu Bar.

5.6.4. Smoothing

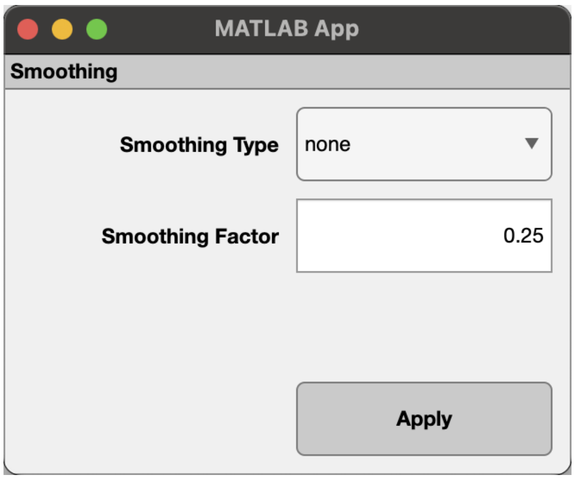
Smoothing can be applied to plots through the menu bar option **`Tools`** → **`Smoothing`**. First, select the data type to smooth (e.g., `Simulation` or `Batch`), which opens a configuration window for specifying smoothing parameters.

Select the desired smoothing method from the dropdown menu. Available options include:

- `movmean` – Moving average over a 2D window. Effective for reducing periodic trends in data.
- `movmedian` – Moving median over a 2D window. Robust against outliers while reducing periodic trends.
- `gaussian` – Gaussian-weighted moving average over a 2D window.
- `lowess` – Locally weighted scatterplot smoothing using linear regression. Computationally intensive but produces smooth results with minimal discontinuities.
- `loess` – Locally weighted scatterplot smoothing using quadratic regression. Slightly more computationally expensive than `lowess`.
- `sgolay` – Savitzky-Golay filter that fits a quadratic polynomial over a 2D window. Particularly effective for rapidly varying data.
- `none` – No smoothing applied. You may select this option to revert to the original data.

Specify the smoothing factor to control the window size. Larger values increase smoothing but may obscure fine details.

Click the `Apply` button to process the selected data type. The plots will automatically update to display the smoothed data.



Smoothing Type: none

Smoothing Factor: 0.25

Figure 5.16: Smoothing Window.

5.6.5. Help

The **`Help`** menu provides access to this documentation on GitHub and contains contact information for technical support.

If GitHub does not display the PDF file directly in your browser, you can download it to your local machine and open it using a PDF viewer application.

Chapter 6

Troubleshooting

6.1. Common Issues

- **Application Crashes:** If the application crashes, try restarting it. If the problem persists, check the console for error messages.
- **Slow Performance:** MATLAB applications can be slow especially when you first opening the GUI. Take your time and be patient for waiting for its initialization. Ensure your system meets the recommended specifications and close unnecessary applications to free up resources.
- **Unexpected Results:** If you encounter unexpected results, double-check your parameter settings and input data.

6.2. Getting Help

If you need further assistance, please refer to the `Help` menu in the application, which provides links to documentation and support resources.

Chapter 7

FAQs

Chapter 8

Tips and Best Practices

8.1. General Tips

- Always save your work frequently to avoid data loss.
- Use descriptive names for your files and variables to make your code easier to understand.
- Comment your code generously to explain complex logic or important decisions.
- Restart the window if you encounter unexpected behavior.

8.2. Best Practices for Simulation

- Start with a small number of trials to test your setup before scaling up.
- Use the built-in visualization tools to monitor your simulations in real-time.
- Document your parameter choices and their justifications for future reference.

Chapter 9

Support and Contact

For support and assistance, please reach out through the following channels:

Wenqing Tu:

Email: alina.tu@qq.com

Chapter 10

Conclusion

In this documentation, we have covered the key features and functionalities of the N-choice model and its GUI implementation. We hope this guide serves as a valuable resource for users and developers alike, facilitating a better understanding of the system and its capabilities.

For further inquiries or support, please refer to the **Support and Contact** Chapter (9) or the **FAQs** Chapter (7).