REAL-TIME EXPERIMENTAL CONTROL WITH GRAPHICAL USER INTERFACE (REC-GUI)

TUTORIAL GUIDE

VERSION 2.0.0 (06/15/2019)

TABLE OF CONTENTS

1   INTRODUCTION

This document describes examples of how to use the "Real-time Experimental Control with Graphical User Interface" (REC-GUI) framework. For additional resources on coding schemes, installation, and modifying the framework to fit your experimental needs, see the *Real-time Experimental Control with Graphical User Interface (REC-GUI) User Manual,* as well as video tutorials found on the REC-GUI website listed below.

> **IMPORTANT:**
>
> *Depending on your experimental needs, additional resources may be required. This document describes example experimental procedures and instructions on how to modify the REC-GUI framework, but you will most likely need to modify and/or create new scripts to apply the REC-GUI framework for your application. The REC-GUI online forum is a valuable resource for customization:*
> *https://recgui2018.wixsite.com/rec-gui/forum*

2   CONTACT

The REC-GUI framework is an open-source project, and we encourage you to contribute to its continued development. To contribute, please contact us via the forum or at any of the following websites or emails.

You can visit us at the following sites:
Lab Website: https://rosenberg.neuro.wisc.edu/
REC-GUI Website: https://recgui2018.wixsite.com/rec-gui
REC-GUI GitHub: https://github.com/rec-gui
REC-GUI YouTube: https://www.youtube.com/channel/UCWqf4wJ5WSkHNhsvKUcmt0A

Please send questions to recgui2018@gmail.com, or visit the FAQ page on the REC-GUI website. The REC-GUI website includes a variety of resources, including a forum to share ideas and solutions, as well as additional video tutorials for running the REC-GUI framework and example codes on your computer.

You can directly reach Byounghoon Kim (bkim10@wisc.edu) or Ari Rosenberg (ari.rosenberg@wisc.edu).

3   CITATION

If you use the REC-GUI framework in your research, please include a citation to:

Byounghoon Kim, Shobha Channabasappa Kenchappa, Adhira Sunkara, Ting-Yu Chang, Lowell Thompson, Raymond Doudlah, and Ari Rosenberg (2019). Real-Time Experimental Control using Network-Based Parallel Processing. eLIFE, 8: e40231.

4   INSTALLATION & SETUP

4.1.   REQUIRED PACKAGES

Main code for the REC-GUI framework:

- Download/clone from GitHub (https://github.com/rec-gui)

Basic packages required for the REC-GUI framework:

- Python2.7: base python interpreter
- Numpy: for calculations using arrays or matrices (https://www.numpy.org)
- Tkinter: for building the GUI (https://tkdocs.com/tutorial/install.html)
  - A programming reference is found on the Tkinter website: https://tkdocs.com/tutorial/firstexample.html
- MATLAB (2017a or greater is recommended)
  - MATLAB's Computer Vision System Toolbox
  - MATLAB's Data Acquisition Toolbox (if using USB-1680G or I/O)
  - Psychtoolbox 3 for visual stimulus generation (https://psychtoolbox.org)

IMPORTANT:

The REC-GUI framework is written in Python and MATLAB. To implement the framework, additional software packages run by Python and MATLAB are required. Since these packages have different developers, we recommend checking for updates and usage. Skipping required packages can cause serious system errors.

4.2.   FILES INCLUDED WITH THE REC-GUI FRAMEWORK

The REC-GUI framework uses a variety of files and file types. The GUI uses a combination of configuration files and Python codes to communicate with MATLAB. The files are contained in the GitHub distribution and include both required and example files that may be run for specific experiments. Please refer to the REC-GUI Technical User Manual for descriptions on each file.

5   SETTING UP THE NETWORK SWITCH

After downloading the required packages and files described in the preceding sections, a network switch must be set up. A network switch is an additional piece of hardware that connects two computers (CPUs) directly together and allows data to be passed between the CPUs in real-time. Setting up the network switch is a critical step in running the REC-GUI framework. Without a properly functioning network switch, the MATLAB and GUI CPUs will not be able to communicate with one another. Please follow each of the listed steps below with extreme care.

5.1.   HARDWARE

To begin setting up the network switch, the hardware between the two CPUs must be properly set up. Using the REC-GUI framework requires at least two CPUs: one running the GUI in Linux and another running MATLAB. The MATLAB CPU can run with Windows, Linux, or MacOS. In this tutorial guide, the MATLAB machine is run on Windows, though we will be providing you some basics for using MacOS with the REC-GUI.

We recommend using a TRENDnet 5-port Gigabit GREENnet Switch; however, you may use any Gigabit network switch that you desire.

1. Connect one ethernet cable from the network switch to the Linux machine
2. Connect a second ethernet cable from the network switch to the MATLAB machine

---

IMPORTANT:

Modern laptops are built without an ethernet port. In the event that your computer does not have an ethernet port, a USB-to-ethernet adapter must be used to connect the two CPUs with the network switch. If you are using a Mac, you will likely need to use the Apple ethernet adapter.

5.2. SETTING IP ADDRESSES

Altering each CPU's IP address is required so that the CPUs may communicate via the network switch. Finding the IP address of both the Linux and MATLAB machines is described below. We recommend using the IP addresses listed below; however, the IP addresses can be modified to whatever you desire. Note, though, that the IP addresses will need to be changed in the files *default.conf* and *start_coding_closedloop.m*

IMPORTANT:

In this Tutorial Guide, we are running Linux (Ubuntu 16.04 LTS), Windows 10, and MacOS (Mojave 10.14.4). For newer software versions, the steps for finding the IP addresses may be slightly different, but the general process will hold true.

5.2.1. LINUX MACHINE

The Linux machine is the computer that will be running the GUI and any Python codes. To find the IP address of this CPU follow these steps:

1. Find the common network icon (e.g., the "WiFi" icon).
2. Select "Edit Connections"
3. Select "Wired connection 1" under "Ethernet Connection" and click "Edit"
4. Go to the IPv4 Settings
   a. Change "Method" to "Manual"
   b. Select "Add"
   c. Change the IP address to the following settings
      i. IP Address: 100.1.1.3
      ii. Net Mask: 255.255.255.0
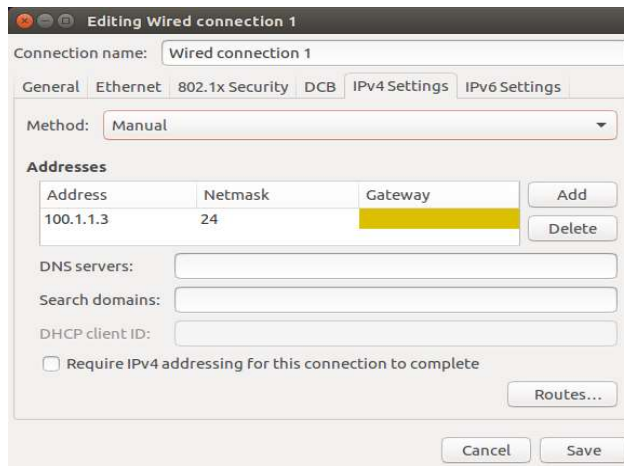   iii. Gateway: Empty
   d. Click "Save"

Figure 5.2.1 Setting the IP Address on the Linux Machine

5.2.2. MATLAB MACHINE

The MATLAB machine is the computer that will be running MATLAB and its codes. To find the IP address of this CPU follow these steps:

5.2.2.1. FOR WINDOWS

1. Search for and open "Command Prompt"
2. Type "ping" and the IP address to ping (e.g., the Linux machine IP address)
   a. Using our IP address, type "ping 100.1.1.3"
   b. "Reply" comments will appear to show that the Windows machine connected to the Linux machine (see **Figure 5.2.2**).
3. Search "Ethernet" on the CPU and select "Change Ethernet Settings"
4. Under "Related Settings," select "Change Adapter Options"
   a. Right-click the Ethernet that is connected
      i. Note that multiple ethernets may appear (e.g., if you have a VPN installed). You must select the ethernet that is connected.
   b. Select "Properties" from the bottom of the drop-down menu
   c. Select "Internet Protocol Version 4 (TCP/IPv4) (see **Figure 5.2.3**)
   d. Select "Properties" at the lower right corner
      i. Select "Use the following IP address"
      ii. Change the IP address to the following settings
         1. IP Address: 100.1.1.2
         2. Subnet Mask: 255.255.255.0
      iii. Gateway: Empty
      iv. Select "OK"

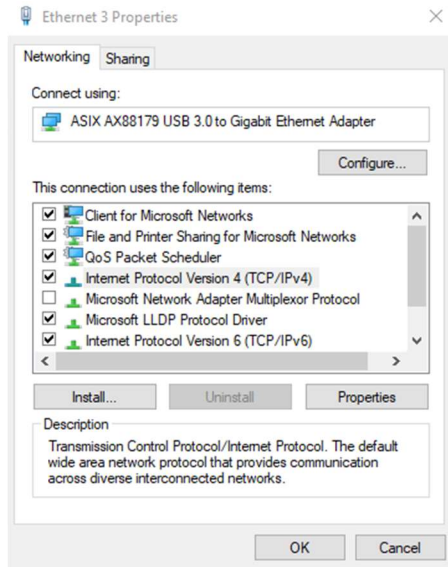
Figure 5.2.2 Ping Test on the Windows Machine

Figure 5.2.3 Accessing Ethernet Properties on the Windows Machine
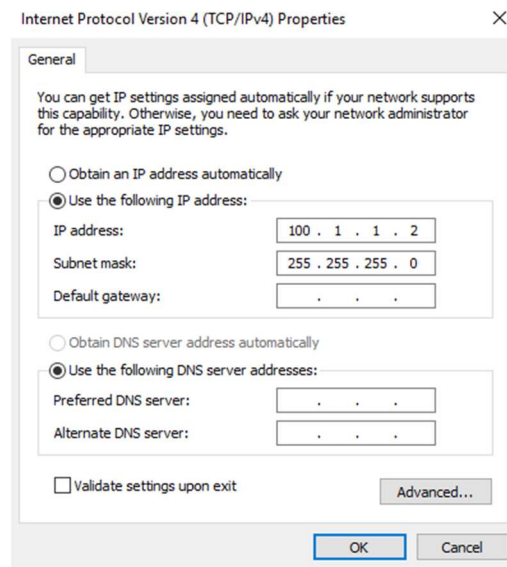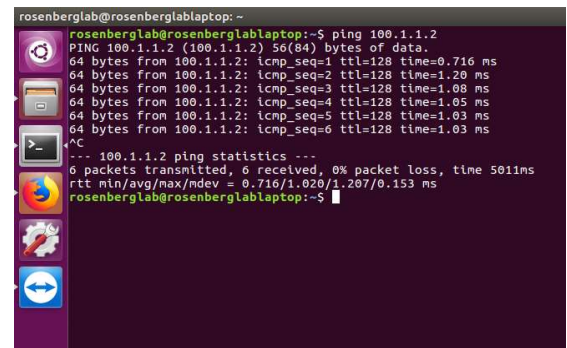
**Figure 5.2.4 Setting the IP Address on the Windows Machine**

5. Close the ethernet settings
6. Return to the Linux machine
   a. Type "ping" and the IP address to ping (e.g., the Windows machine IP address)
      i. Using our IP address, type "ping 100.1.1.2" (see **Figure 5.2.5**)
      ii. "Reply" comments will appear to show the connection
      iii. Type "ctrl+c" to stop pinging
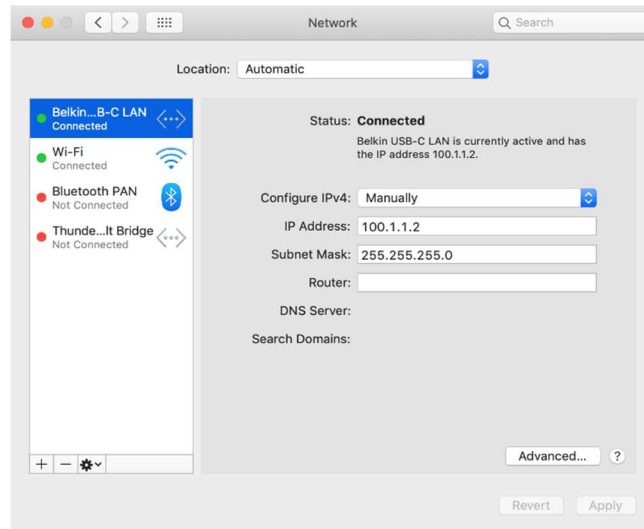

5.2.5 Ping Test on the Linux Machine

5.2.2.2. FOR MAC OS

1. Search for and open "Terminal"
2. Type "ping" and the IP address to ping (e.g., the Linux machine IP address)
   a. Using our IP address, type "ping 100.1.1.3"
   b. "Reply" comments will appear to show that the Mac machine connected to the Linux machine (see **Figure 5.2.6**).
   c. Type "ctrl+c" to stop pinging
1. Find the common network icon (e.g., the "WiFi" icon).
2. Select "Open Network Preferences"
3. Select the connected ethernet that says "Self-Assigned IP" underneath
   a. Select "Manually" from the dropdown menu next to "Configure IPv4"
      i. Change the IP Address to the following settings
         1. IP Address: 100.1.1.2
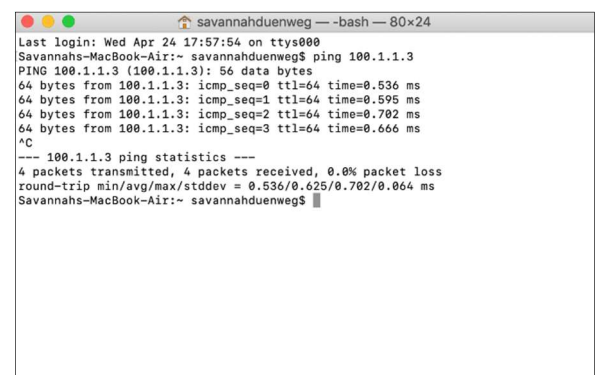
5.2.6 Ping Test on the Mac Machine

7

2. Subnet Mask: 255.255.255.0
ii. Router: Empty
iii. Select "Apply"

5.2.7 Setting the IP Address on the Mac Machine

4. Return to the Linux machine
    d. Type "ping" and the IP address to ping
       (e.g., the MATLAB machine IP address)
        i. Using our IP address, type
           "ping 100.1.1.2"
           (see **Figure 5.2.5**)
        ii. "Reply" comments will appear to show the connection
        iii. Type "ctrl+c" to stop pinging

5.3. FINDING LOCAL/SERVER PORTS

Local and server ports are defined in the files *default.py* and *start_coding_closedloop.m*. In these files, they are set as "5001" and "5002" by default. You may need to change them for your CPUs if those ports are in use. To find available local and server ports, see the following steps. For more information regarding local and server ports, see the REC-GUI User Manual.

1. On the Linux machine, type in "netstat -ano"
    a. A comprehensive list of ports will appear under the heading "I-Node"
2. On the MATLAB (Windows) machine, open the Command Prompt
    a. Type in "netstat -ano"
    b. A comprehensive list of ports will appear under the heading "PID"
3. On the MATLAB (MacOS) machine, search "Activity Monitor"
    a. A comprehensive list of ports in use will appear under the heading "PID"

> **IMPORTANT:**
>
> If the port 5000 is unused, you may use the default parameters in the files *default.py* and *start_coding_closedloop.m*. If these ports are in use, you will need to change the values to something unused (e.g., 4000). These values must be opposites in the Linux and Windows machines (e.g., if the local port is 5001 and the server port is 5002 in *start_coding_closedloop.m*, the local port must be 5002 and the server port must be 5001 in *default.py*).

6   LEARNING THE REC-GUI BASICS

To get familiar with the REC-GUI framework, use the file *start_coding_closedloop.m*. This is a basic MATLAB script that will allow you to track cursor movements in the Monitoring Panel, but more importantly, learn the basics of the GUI.

6.1.   RUNNING THE GUI

The GUI is run on the Linux machine, whereas stimulus for the GUI is run on MATLAB. To open the GUI, you must access the directory that *main.py* is housed in. When working in Linux, the following commands can be used:

- cd: change directory
- cwd: current working directory
- ls: lists contents of current working directory
- pwd: prints working directory to the terminal

An example of how to open the GUI is shown below. It is not necessary to list the contents of the directory after each time a new directory is opened; however, it may be helpful to see the contents while you are first getting used to opening the GUI. The required lines to open the GUI are highlighted below.

```
rosenberglab@rosenberglablaptop:~$ cd rec-gui-master/
rosenberglab@rosenberglablaptop:~/rec-gui-master$ ls
GUI_1.1  LICENSE  MATLAB  README.md  REC-GUI User Manual Version 1.01.pdf
rosenberglab@rosenberglablaptop:~/rec-gui-master$ cd GUI_1.1/
rosenberglab@rosenberglablaptop:~/rec-gui-master/GUI_1.1$ ls
analog.py               eye_interpret.py        Receptive_Field_Mapping.conf
arbitrator_server.py    eye_interpret.pyc       select_eye_recording.py
arbitrator_server.pyc   file.py                 select_eye_recording.pyc
button_funs.py          file.pyc                staircase.py
button_funs.pyc         Fixation.conf           start_coding_closedloop.conf
Calibration.conf        global_parameters.py    subject_configuration
calibration.py          global_parameters.pyc   subject_default.conf
calibration.pyc         gui.py                  test.py
constants.py            gui.pyc                 utility.py
constants.pyc           logger.py               utility.pyc
data_collection.py      logger.pyc              vergence_version.py
data_collection.pyc     logs                    vergence_version.pyc
default.conf            main.py                 widget_api.py
digitalIO.py            read_task_parmeters.py  widget_api.pyc
rosenberglab@rosenberglablaptop:~/rec-gui-master/GUI_1.1$ python main.py
```

To set up the GUI for use with *start_coding_closedloop.m,* use the following parameters:

- After the GUI opens, you will be prompted to choose an "Eye Method Selection." Select "None"
- In the Task Control Panel, type "start_coding_closedloop.conf"

6.1.1 Load *start_coding_closedloop.conf* in the Task Control Panel

- In the Subject Name Panel, type "test"

6.1.2 Load the Subject Configuration file *test*

- Under the Monitoring Panel, select "Dragging"

For more information regarding the Task Control Panel, creating Subject Configuration files, and the Monitoring Panel, see the REC-GUI Technical User Manual.

6.2. RUNNING *START_CODING_CLOSEDLOOP.M*

The MATLAB script *start_coding_closeloop.m* is included in the files on the REC-GUI GitHub page.

- Open *start_coding_closeloop.m* in MATLAB
- Use the following line in the command window of MATLAB

```
start_coding_closedloop(1)
```

- A pop-up in the GUI will appear stating "Third Party Server Connection Established" (see **Figure 6.2.1**)
- Click "OK"
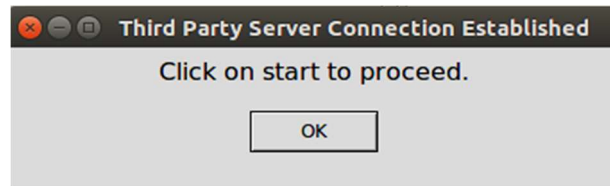- Select "Start" in the Task Control Panel

Figure 6.2.1 GUI Pop-Up Once MATLAB has Started

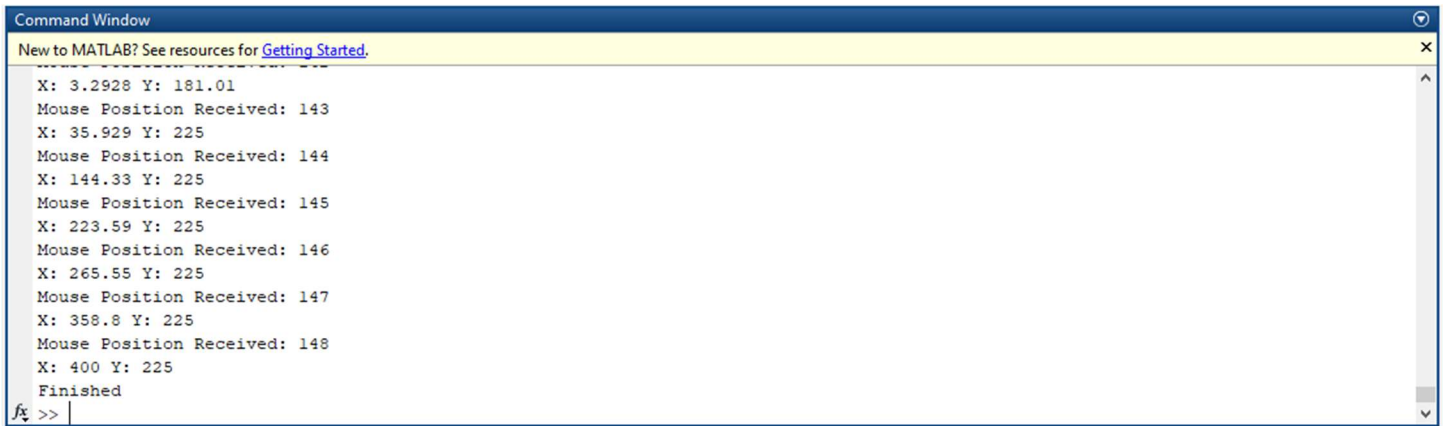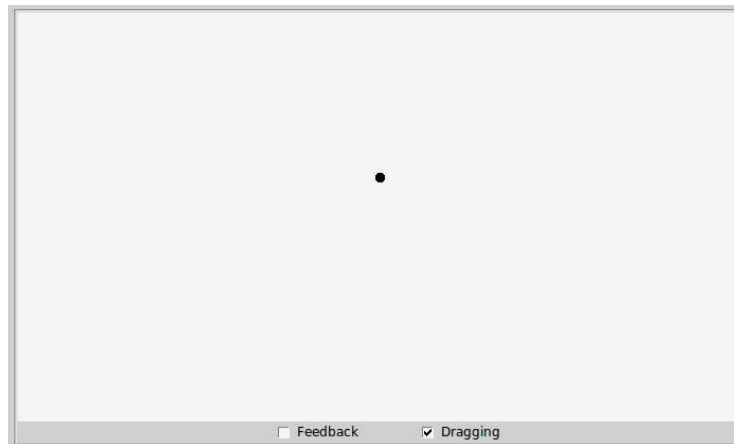- Click on the Monitoring Panel and move the mouse around

Figure 6.2.2 The Monitoring Panel

- In the Command Window of MATLAB, Mouse Positions will appear ((see **Figure 6.2.3**)


```
Command Window
New to MATLAB? See resources for Getting Started.
    X: 3.2928 Y: 181.01
    Mouse Position Received: 143
    X: 35.929 Y: 225
    Mouse Position Received: 144
    X: 144.33 Y: 225
    Mouse Position Received: 145
    X: 223.59 Y: 225
    Mouse Position Received: 146
    X: 265.55 Y: 225
    Mouse Position Received: 147
    X: 358.8 Y: 225
    Mouse Position Received: 148
    X: 400 Y: 225
    Finished
fx >> |
```

Figure 6.2.3 Mouse Positions in the MATLAB Command Window

- To temporarily stop the GUI, select "Pause"
- To end the GUI, select "Stop." MATLAB will respond "Finished" in the Command Window.

IMPORTANT:

If an error appears in the Command Window of MATLAB stating, "fatal error," "asynchronous process," or any other error message, the local and/or server ports may have been set up incorrectly.

6.3. SUCCESSFUL SET UP

Congratulations! You have officially set up the REC-GUI framework. From here, you may follow our tutorials for running the other example codes we have provided on the REC-GUI Github, or you may begin creating your own projects. If you create your own experiment, remember to follow the same basic coding structure that we have used in our example codes.