Instructions for Demo of Pavlovian Control of Walking

Written by Ashley Dalrymple, PhD

Last updated: Sept. 2, 2019

Purpose of Program:

- Controller for intraspinal microstimulation to restore walking in a feline hemiplegia model
- Used to compare traditional rule-based (reactive) control and Pavlovian (predictive) control
  - Rule-based control uses thresholds on sensor data recorded during walking trial
    - The thresholds need to be manually tune for each new subject
  - Pavlovian control uses reinforcement learning to learn predictions of sensors recorded during walking trial, and thresholds on the prediction signals produce walking
    - The thresholds do not need to be tuned at all for each new subject

Note: it is useful to review the manuscript "Personalized Control of Over-ground Walking using a Spinal Cord Implant" by Dalrymple et al. before using this program. This manuscript is currently under review with Nature Machine Intelligence.

This demo focuses on the Pavlovian controller, as this is the novel contribution of this work. It demonstrates the learning of predictions for sensor data collected during a walking trial, as well as how those predictions initiated transitions between the phases of the walking cycle.

Hardware and Software Requirements:

- This controller has run on multiple versions of Matlab including 2017B – 2018B inclusive on a Windows operating system (Windows 7 or 10 64-bit). Older versions of Matlab may not work as we make use of a newer function (bsxfun).
- Sensors used to record walking-relevant information include custom force plates (ground reaction force, sum of forces, moving average of force, unloading) and gyroscopes (angular velocity) for each hind-limb.
- Data acquisition was done using a Grapevine Neural Interface Processor (Ripple, Salt Lake City, UT, USA) and associated software, which digitized the data at 1 kHz after custom-made hardware filtering using a Butterworth filter (f_C = 3 Hz, 2^{nd} order). The mex file xippmex is for communicating with the DAQ.
- Stimulation to the spinal cord for intraspinal microstimulation was delivered using a customized current-controlled stimulator (Sigenics Inc., Chicago, IL, USA)

Note: the demo version of this program does not require the sensors, DAQ, or stimulator to run. It does require Matlab.

Installation instructions:

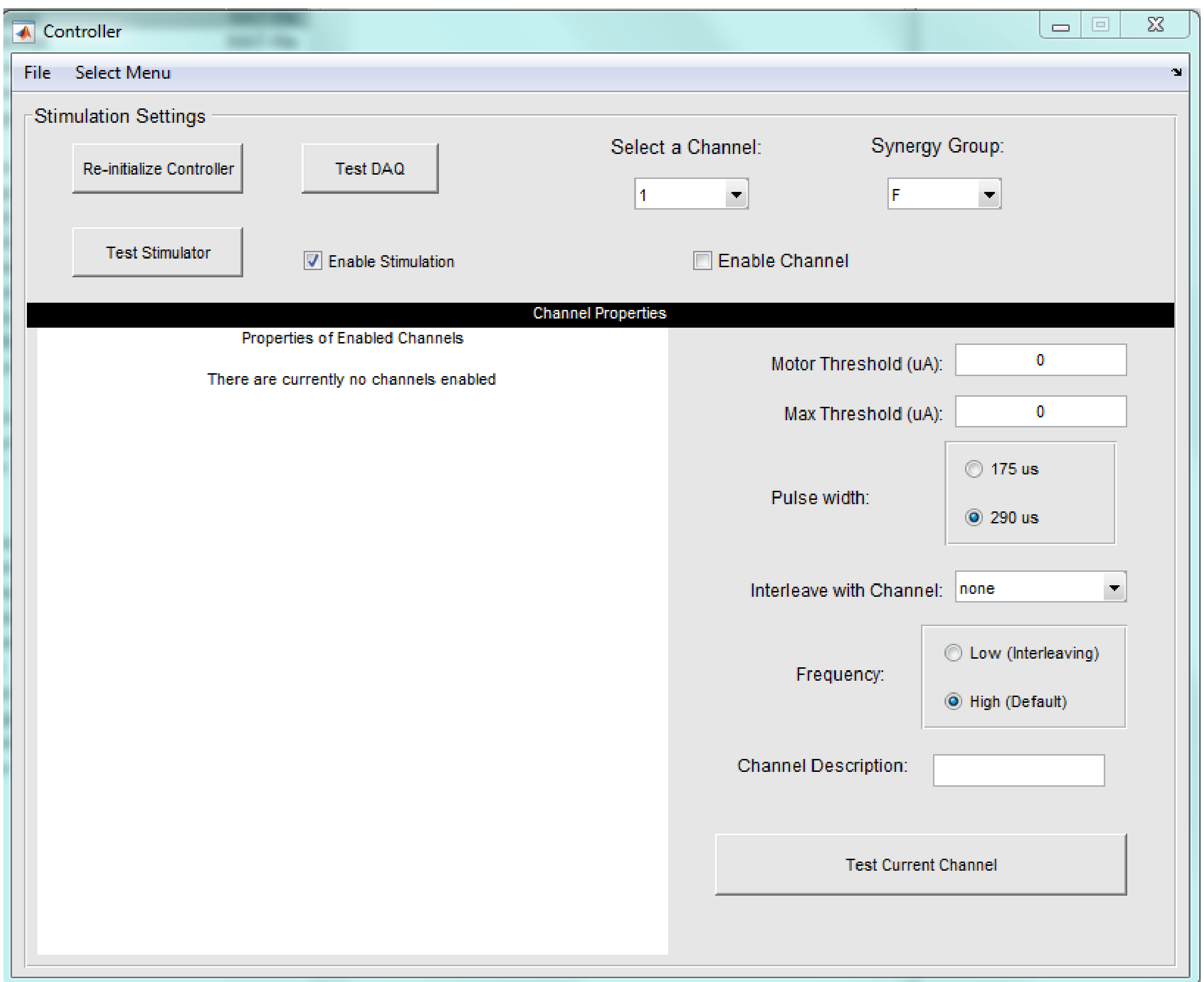
Download the zipped files "Code" and "Data files". Put the contents in their own folder together. They contain all the files and code needed to run the demo. Installation time should only be a few seconds.

Included files:

- All .m files and other files for running the controller and all associated functions
- All .m files and data files for Selective Kanerva coding. The prototypes in the state space were randomized once and held constant for all experiments (SKC6sensors.mat)
- PrevLearnSett.mat: learning settings from the trial previous to the demo. Used to demonstrate learning that continued from one trial to the next.
- DemoSettings.mat: controller settings used for the demo trial
- DemoTrial.mat: contains a reduced amount of data collected from the walking trial in the demo. It is used to provide sensor signals that were recorded since the hardware is not present for the demo. The file was reduced due to size restrictions upon uploading.
- Original_Trial_Outcome.mat: contains the same information as DemoTrial.mat but is used for plotting the predictions and comparing them to the output from the demo.

Using the Demo Program:

- controller.m is the master file for running the software. It contains a GUI and all relevant initialization steps for communicating with the hardware.
- Run controller.m by opening Matlab and typing controller into the command window, then hit enter. The following GUI appears:

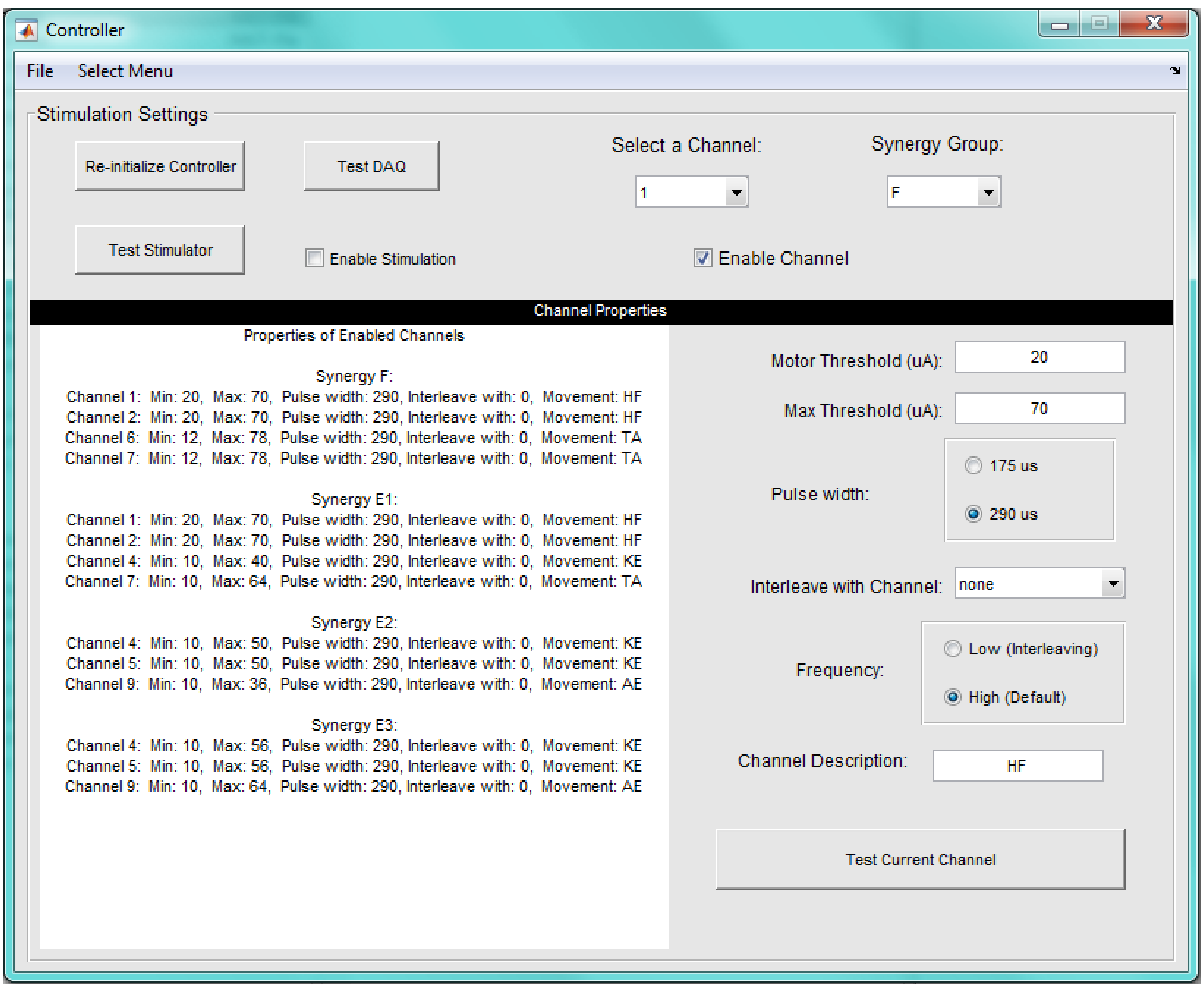


This Channel Settings Menu contains the following functions:

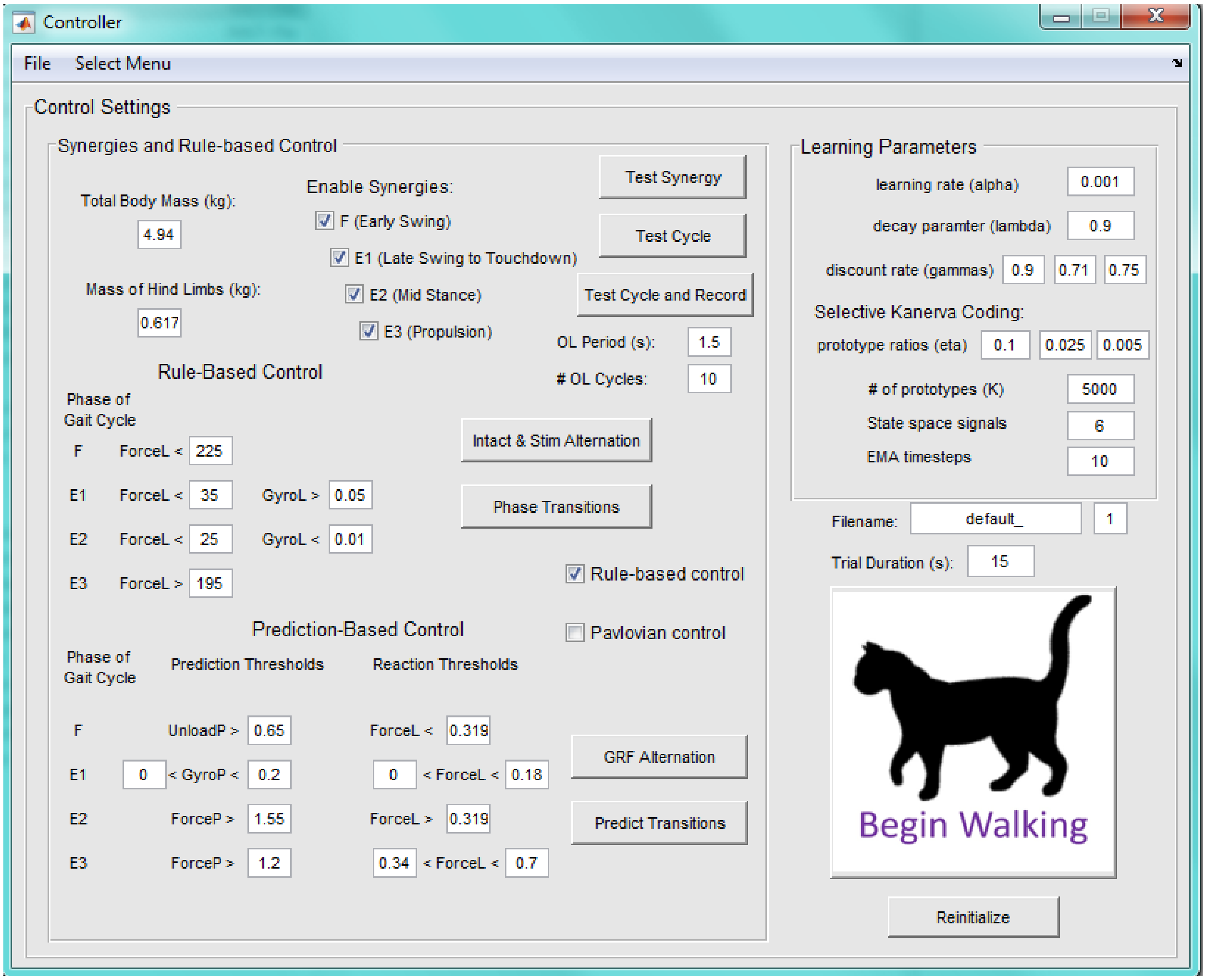
- Initialization of the controller/stimulator

- Testing stimulator with a 1s pulse through channel 1
- Testing of the DAQ
- Enable stimulation
- Selecting channel for stimulation settings
- Selecting the walking synergy to allocate the stimulation channel
- Enable the channel for stimulation
- Channel properties for customizing the stimulation settings for each channel within a synergy
  - Motor threshold (for ramping between this lower limit to upper limit)
  - Max threshold (max amplitude for that channel)
  - Pulse width
  - Options for interleaving channels
  - Stimulation frequency (relevant if interleaving)
  - Channel description (describes muscle output produced by single channel)
  - Test current channel with settings
  - Display of all channels and synergies

For demo purposes, load the stimulation settings from the demo file by selecting “File” from the menu at the to, and selecting “Load Controller Settings”. Select the .mat file “DemoSettings”. This will load the stimulation settings from an experiment:



More options can by found by selecting “Select Menu” following by “Hemi Settings”. This menu contains additional testing abilities and options for the different control strategies. Since the previous settings were already loaded, the menu will look like this:



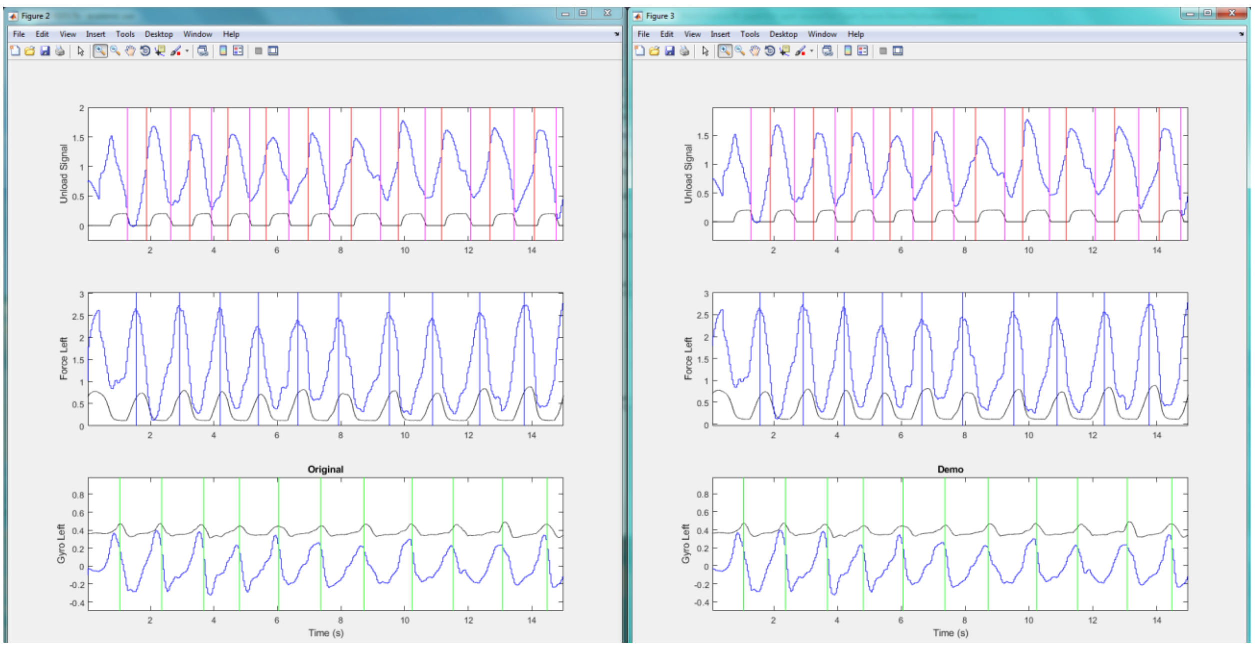
This menu allows users to:

- Enter the mass of the animal; the goal mass of the hind-limb is automatically calculated
- Enable synergies
    - Enable individually for testing and adjusting in previous menu
- Test cycle: open loop cycle for testing synergies together in a full walking cycle
    - Can adjust the walking period, number of cycles, and can record sensor data if desired
- Select walking control system
- Change settings for walking control systems
- Run walking trials with different control systems

To run the demonstration of Pavlovian control with reinforcement learning:

- Select the “Pavlovian control” checkbox
- If desired, change the filename to what you like. For the demo, we have entered “Test”. The file number automatically increments after a trial.

- Click the large button that says “Begin Walking”. There will be a prompt to get ready; click “OK”.
- The program will automatically load previously learned variables for the eligibility trace (e), weight vector (w), and the previous prediction of the sensor signals (Vold) from the file “PrevLearnSett”.
- It will also automatically load the Demo trial for acquiring the sensor data, since there is no hardware to collect it live as in an experiment.
- The program will run for the duration of the trial time (15 seconds) and will ding when done.
- The prediction output can be found by clicking “Predict Transitions” on the GUI. A dialogue will open asking for the file that you wish to plot. Select “Test_1.mat” (if you also named your trial “Test”). To compare with the original output, repeat by again clicking “Predict Transitions” and selecting “Original_Trial_Outcome.mat”. Here is what is expected when the two files are plotted side-by-side. These examples have been zoomed in for easier viewing.



The black traces are the normalized sensor data loaded from “DemoTrial”. The blue traces are the predictions learned. In the original, these predictions were formed online during walking. In the demo, these predictions were learned in the exact same way but with loaded data instead of streamed data. The vertical lines indicate the phases of the walking cycle that were/would be triggered using the predicted data to transition the stimulation-controlled limb to the next phase of the walking cycle. Solid lines indicate a prediction-initiated transition (as opposed to a back-up reaction). Red = early swing (F); Green = late swing to paw touch (E1); Pink = mid-stance (E2); Blue = push-off (E3).