If the headfixed mouse is available, run the session with the mouse on the wheel and the camera adjusted as well as can be. Run the commands to issue sound stimuli and laser shutter commands. Make sure that the events are posting in the Events window during sound stimuli and laser shutter commands. Rotate the platform through the full 360 degrees slowly and then do fast rotations as in a real session.

**AHV**

[AHV\_tsd] = Get\_AHV([]); plot(AHV\_tsd.tvec, AHV\_tsd.data); % make sure AHV looks normal

[STtstart, STtend, tlistX] = findSteadyPeriodsManual() % plots AHV for isolating stationary periods

**LASER**

[start\_time, ~, laser\_on, laser\_off, arraysize, ~] = SortBrainstemEventLabels3;

d = diff(laser\_on)

% laser events in a sequence should each be 4 seconds apart

**SOUND**

events\_ts = LoadEvents([]); % Make sure the event labels look correct

wrapper = @(events\_ts) strcmp(events\_ts, 'ShutterSound On');

A = cellfun(wrapper, events\_ts.label);

sound\_label = find(A); % index which label corresponds to 'ShutterSound On'

sound\_times = events\_ts.t{sound\_label};

sound\_times = sound\_times - start\_time;

e = diff(sound\_times)

% sound events in a sequence will each be 4.2 seconds apart. The 0.2 seconds is processing time from matlab.

**EYETRACKING**

Run facemap. Run detectSaccadesManualCheck6([]); See how the saccades look.

WHEEL

updownTSD = getQEupdown([]);

state\_tsd = ConvertQEUpDownToState(updownTSD);

[~, wheel\_tsd] = ConvertQEStatesToAngle([], state\_tsd);

[d, speed, cfg\_w] = ConvertWheeltoSpeed([], wheel\_tsd);

speed.tvec = speed.tvec - sd.starttime;

speed.data = -speed.data; % we want forward motion to be displayed as a positive velocity

OR

get\_wheel\_speed.m

plot(speed.tvec, speed.data)

OR

sd = LoadSessionData([]); plot(sd.speed.tvec, sd.speed.data)

**NEURONS**

If neurons were recorded, cut a cell in MClust 3.5

plot\_AHVtuningMegaplot10(iCell) should bring up the subplots for that cell and include everything