**NOTES on software and on running subjects for Steady-Triplet stimuli.**

**Last updated: Oct 2011**

**My script/instructions (for pre-session, June 2011)**

You will hear sequences of A\_B\_A repeated for 4 mins. Depending on Freq A and Freq B you may hear the repeated pattern:

ABA ABA … galloping … play a 20-40 sec sample w/ DF=1. “One stream”

or

two separate steams, simultaneously, if DF is large:

A A A A A and B B B B B …., the second at ½ the presentation rate. “two streams”

Play 20-30 sec example. w/o using keypad.

Play 20-30 sec examples each…. First w/o using the keypad. With DF=1 and DF=10

For some cases, intermediate DF, you may hear the pattern change….alternate between galloping and 2 streams…. Play a 40 sec example, DF=5. Play again and use keypad, leftarrow for 1 stream and rightarrow for 2 streams. Do it again if necessary JR CHECK the monitor to see if correct.

JR could show them a visual analog, like the vase/faces or a plaid.

If non-GUI-driven presentation: each trial will run through some conditions (ncon) with 20 sec pause before each condition, then 3 sec warning beep, then 2 sec and the condition’s train begins. There will be one-min pause btw trials.

+++++++++++++++++++++++++++++++++++++++

**Volume**:

For headset, JR Logitech, use vol setting #7 (Control panel, Sounds & Audio Devices… not the icon tray sound device). JR must check this volume… Ask Weiwei.

+++++++++++++++++++++++++++

**NOTES on sound production and presentation.**

**GUI-driven.**

This code

steady\_triplet\_Ntrials.m is in SABBATICAL 2011\Ascend Descend Matlab\steady triplet Ntrials

it prob evolved from

steady\_triplet\_Khold.m is in SABBATICAL 2011\Ascend Descend Matlab\Bernhard E

Early results, prior to June 2011, were obtained with GUI-driven sound/presentation codes.

The parameter set was specified for each run with the GUI and then results stored in a structure as indicated just below. The results for multiple cond’ns could be stored in a given matrix, but still only 1 trial per each cond’n.

**Not GUI – multiple trials and multiple conditions.**

The code is in:

SABBATICAL 2011\Ascend Descend Matlab\steady triplet Ntrials

In June 2011 (while in China), developed

StTripNtrials.m that was not GUI-based.

This code requires the user to define the ‘conditions’ internal to the code.

The triplet seq’ce (time series) for a cond’n is pre-generated by the code SteadyTripSeq.m

i.e., function [steadyseq]=SteadyTripSeq(tones,indL,indH,gap,nreps).

These seq’ces are stored as \*.mat files and loaded by StTripNtrials.m

The code SteadyTripSeq.m depends on Tones(Tdur).m which defines the unitary tones to form triplets.

+++++++++++++++++++++++

Here is how the params for running StTripNtrials are defined in the code:

txt\_subjdate='10.23.06.2011\_3DF\_2trl';

ncon=3;numtrial=2;

for k=1:ncon;result(k).names=txt\_subjdate;end;

Tdur=120;gap=120;repeats=500;

%result(condn).params=[Tdur repeats indL indH gap condn# or DF];

result(1).params=[Tdur repeats 6 2 gap 4];

result(2).params=[Tdur repeats 9 4 gap 5];

result(3).params=[Tdur repeats 12 6 gap 6];

%

% randomize cond ordering (and across subjects) from one trial to the next

%

% set up the trials (each row of indcond is a trial) and load pre-generated sound sequences

%

indcond=[4 5 6; 5 6 4; 6 4 5;4 6 5]; % this example allows for 4 trials.

steadyseq4=load('HLHdf4seq.mat')

steadyseq5=load('HLHdf5seq.mat')

steadyseq6=load('HLHdf6seq.mat')

++++++++++++++++++++++++

NOTE: on Oct 30, 2011 JR realizes that although these runs were carried out properly and randomly ordered wrt DF they were not stored in the proper component of ‘result’… ie not all put in the correct result(condn).data. So on Nov 1, 2011 JR reordered the data using the code:

result\_reorder.m in the folder: SABBATICAL 2011\Sussman\_AECOM\steady\_triplet\_data.

ON DEC 1, 2011 JR realizes that there is an error in the reorder code for the reassignment. the second and third rows of kcvals should be interchanged … see comments in the code.

I did this for subjects: 04, 08, 09, 10… the only ones that had randomized cond’ns btw trials in the time period June 2011 to Oct 30, 2011.

**Playing the sound trains/recording responses :**

Using earbuds: just plug into the jack. Set the volume to moderate.

Using headphones:

* Plug into USB port
* Go to Start; go to Control Panel; go to Sound & Audio Device; click “Audio” tab;
* Set the ‘Sound Playback’ device, e.g. Logitech USB Headset
* Set the volume to ‘XXX’; to intermediate level, I seem to recall.

Using the external keypad: plug into USB. Make sure that “numlock” is off, not lit.

**NOTES on steady-triplet data sets**

25may2011

The naming convention for the files:

XX.dd.mm.yyyy.ZZ

XX is ID # of subject

dd.mm.yyyy is the date

ZZ is session #1 or #2:

Each session was one run with each cond’n.

In RunSheet A:

#1 is button clicks

#2 is button presses (BPs) or keypresses/keyholds (KPs)

In RunSheet B:

#1 is Keyholds … 5 DFs

#2 is Keyholds… another repeat of 5 DFs

In RunSheet C:

#1 is Keyholds …. 3 DFs … repeat of 3 DFs

#2 is Keyholds …. 3 DFs… repeat of 3 DFs

**KeyPresses data storage**

These data from KeyRecorder are stored in files like:

04.13.10.2011.2.KPs.mat

If load such a file into Workspace it shows up as KeyPresses, a cell array: cell(2,1)

With structures for Up and Down in two cells for leftarrow key and rightarrow key, respectively:

KeyPresses{1} = struct('Down',[],'Up',[]);

KeyPresses{2} = struct('Down',[],'Up',[]);

To see the times in leftarrow down:

KeyPresses{1}.Down

To see the time of first leftarrow down:

KeyPresses{1}.Down(1,:)

To see the min & sec of first leftarrow down:

KeyPresses{1}.Down(1,5:6)

In multi-trial runs (as in Staircase) I may concatenate KeyPresses:

result(kc).rawdata=[result(kc).rawdata;KeyPresses];

Now when I load ‘result’, I get:

>> result.rawdata

ans =

[1x1 struct]

[1x1 struct]

[1x1 struct]

[1x1 struct]

I do:

result(iTrial).rawdata{iKeypress}

where iTrial indexes trials and iKeypress indexes Keypresses.

Alternatively, to get at the data, I can work backwards:

>> KeyPresses=result.rawdata

KeyPresses =

[1x1 struct]

[1x1 struct]

[1x1 struct]

[1x1 struct]

>> KeyPresses{1}

ans =

Down: []

Up: []

Soooo, KeyPresses on leftarrow is empty… there were no presses/releases

But on rightarrow, there in one down/up.

>> KeyPresses{2}.Up

ans =

1.0e+003 \*

2.0110 0.0100 0.0260 0.0110 0.0180 0.0318

>> KeyPresses{2}.Down

ans =

1.0e+003 \*

2.0110 0.0100 0.0260 0.0110 0.0180 0.0315

**Analysis of data: (June 2011)**

The data are stored in a Structure… maybe 3 or 4 components

result.names

result.params

result.data

result. rawdata this contains the times of keypresses/releases.

…and there could be several runs (one for each cond’n) in a single Structure… eg, with DF=4,5, 6 say result(condnum).names, etc,

The component: .data has two colns: (durn, percept) where percept =1 or 2 for one stream or 2 steams. The first line in each trial is (0,0) … these should not be counted in # of events or in data analyses. (NOTE: in analyses before July 2011, JR mistakenly included them in computing means of durns for “2”… MOST of XXrnet data sets were redone in Sept 2011 to correct this mistake.)

**Histograms of durations are plotted** using

durshist\_onetrial\_result (resultAA, condnum) June 2011

To implement this analysis

1. Put “result” from XX.dd.mm.yyyy.ZZ into the workspace by clicking on it the current directory.
2. Then in cmd window:

resultAA=result

1. Check the \*.names and \*.params, esp the latter to check the trial of interest
2. In cmd window type:

durshist\_onetrial\_result (resultAA, condnum)

This gives histograms of durns for 1 stream and 2 streams

Could also use:

function durshist\_Ntrials\_result(result,condnum,maxT,numbins)

after JR corrected the code to not count the ‘zero’ rows btw trials.

This is for a single subject with Ntrials and it can read data from files like:

04.12.10.2011\_3DF\_2trl.mat

This data set can also be analyzed by

function mstat=durshist\_group\_rnet(result,condnum,maxT,numbins)

(see below)

**Analysis of Accumulated Runs, individ subj (XX): June 2011 & Sept 2011**

The data are stored in a Structure…

**XXrnet**

Or, trivially, used for a single user data set like:

04.12.10.2011\_3DF\_2trl.mat

The data are stored in a Structure… maybe ZZ components:

result.names

result.params

result.data

result. rawdata

Here, the organization differs from that for a single session. Here, the Structure is organized around result.data… a systematic ordering of the data from different conditions and different runs.

Our first pass of accumulating data ( June 18, 2011) leads to a structure with 3

components, corresponding to the 3 conditions (ZZ=3): DF=4, 5, 6 – respectively, (obtained by vertical concatenating the result.data from different runs for a given DF). For example, the data for DF=YY might be in run 3 from session r011 and in run 2 from session r012 and in run 1 from session r013:

rnet01(1).data=vertcat(r011(3).data,r012(2).data,r013(1).data)

Here, result.names is identical for each component. It gives subj ID, the list of data sets from which runs are taken, reporting condition – say KHs, KeyHolds.

For example:

rnet01(1).names 🡪

'sub=01' [1x58 char] 'KHs'

rnet01(1).names(2) 🡪

ans =

'data sets: 01.25.05.2011.2&01.08.06.2011.1&01.08.06.2011.2'

Here, result.params gives the conditions. For this example we see the ST #’s yield the DFs of 4,5,6 :

rnet01.params 🡪

ans =

120 500 6 2 120 0

ans =

120 500 9 4 120 0

ans =

120 500 12 6 120 0

The ‘rawdata’ is meaningless in this configuration… go back to the original data sets.

Analysis: just use

durshist\_onetrial\_result

Eventually, I could automate the assembly of XXrnet by just giving data set names and have histogram be part of this code automating code.

**NOTES from Sept 2011, Analysis of ‘grouped data’**

These codes are in the folder:

SABBATICAL 2011\Ascend Descend Matlab

This program can be used for an individual XXrnet or for grouped data say as in

‘groupXX.mat’.

First, ‘import’ the data into workspace.

Then, BY HAND, go into this .mfile

**function mstat=durshist\_group\_rnet(result,condnum,maxT,numbins)**

and insert some info (highlighted below) about this data set: a name for the data file and the conditions under analysis. The program ‘output’ will be (Matlab \*.fig) histograms for 1, 2 streams and a matrix of stats: ‘mstat’. The input ‘result’ has the format of a structure… The m-file reads the ‘result.params’ and ‘result.data’ for each cond’n.

the argument: condum is to identify which ‘condition’ is being analyzed. Eg condnum=1 means DF=4.

the argument: maxT is for range of durn’s that will be plotted in the histogram (although, all durns are used in computing the stats.)

the argument: numbins is for number of bins over durn range: (0,maxT)

mstat has 2 rows for the 2 percepts and 4 cols. Rows 1:2 for percept number 1or 2. Col 1: percept number, col 2: mean durn, col 3: median durn; col 4: mode.

(JR find out how to produce nice finished plots and/or data for plotting w/ Illustrator or smth.)

For example:

Calling the m-file

s10=durshist\_group\_rnet(rnet10,3,20,20)

Grabbing means for sub10 and putting into a matrix of means for a set of subj’s [first form a vector ‘meansXX’ for this subj 10 as horiz concat’n of means for 2 percepts: elements 1,2 are for condnum=1 and elements 3,4 are for condnum=2 and elements 5,6 are for condnum=3].

means10(5:6)=s10(1:2,2)

now put these into matrix ‘means’ that includes means for several several subj’s with col 1 indicating subj number.

means(6,:)=[10,means10]

save('meansAE3C3','means') …saves the matrix into \*.mat format

to save it into ascii, eg:

save('meansAE3C3.asc','means','-ascii')

+++++++++++++++++++++++++++

function mstat=durshist\_group\_rnet(result,condnum,maxT,numbins)

%Sept 2011

% determine and plot the histograms of grouped data: durations

% for 1 and 2 streams in the structure data set 'groupXX.mat' and

% for condition=condnum , eg, DF=4, 5, 6 as condnum=1,2,3

%

% JR come into this code and provide subjnames and datatype by hand

subjnames='GroupAE3C3: 01 03 06 08 09 10';

%subjnames='01net';

%subjnames='subj10: 10';

datatype='KHs, HLH, 4 min trials';

DF=4;

dataparms='tone&gap 120, reps 500, DF=4, high 6, low 2';