From: wolf zinke [mailto:wolf.zinke@vanderbilt.edu]

Sent: Tuesday, August 18, 2015 2:11 PM

To: Heitz, Richard P

Subject: Re: old plexon format again

Hi,

I thought that I had some progress with the file conversion but I always get stuck on details. Here are a couple of things where i would be grateful for some clarifications.

I guess more questions might follow.

wolf

##

1) For example, there is a block for the saccade location in your script:

```
 %Actual Saccade Location
for pos = 0:7
    if ~isempty(find(TrialStrobeValues(1:relevant) == pos + 7000,1))
        SaccDir_(newindex,1) = pos;
    end
end
```

I do not find codes between 7000-7007. Is this code block outdated?
I added that later on, so it might be missing for some datasets, but only very old ones. I don't know… actually I thought I used 8000 number ranges. Can't recall

##

2) Similar, there is another block that identifies the presented stimuli:

```
 %Actual Stimuli Presented
for pos = 0:7
    if ~isempty(find(TrialStrobeValues(1:relevant) == pos + 6000,1))
        Stimuli_(newindex,pos + 1) =
TrialStrobeValues(find(TrialStrobeValues(1:relevant) == pos + 6000,1) + 1);
    end
end
```

Here, I do not understand the coding scheme. It seems that the values following the codes 6000-6007 are 0 except for one. I assume 0 means distractor (or does it mean that no item was at this location), but the number not 0 is not just 1. What is the meaning of these numbers?

I think this codes the actual stimulus identity in each of the locations. They should be something like 11-14 and 21-24. The 10's are "T" and the 20's are "L" and the 1's place codes the rotation.

##

3) One code refers to 'Hold Time' (3021) which is explained as: '*Hold Time (use to determine search or memory guided)*'. What does this actual mean?

this is probably the ms value for how long the fixation point remained solid. If 0, it was a regular search trial. If > 0, was a memory guided.

##

4) Also, there is 'Trial Hold Time' (3027, '*taking into account jitter - value actually used*') - what does this mean? Does this indicate kind of the onset of the fixation spot prior to the search array onset?

The Tempo parameter for hold time was a constant (e.g., 750 ms). However, in practice it was jittered by adding some offset drawn from a uniform or exponential distribution. Originally only the former was written. I later added the exact hold time + jitter

##

5) What is the difference between the two error codes 2621 (FixError_) and 2623 (HoldError)?

mmm, probably 2621 is a broken fixation during a search trial, while hold error is a broken fixation during memory guided?

##

6) I have in some correct trials with more than one code for reward, both occur before the EoT code but do not have the same timing across trials. Is this something intended?

not sure, but perhaps Tempo wrote the reward code even for manually delivered reward. That usually happened early in the session. You can either check the tempo code to see if those codes got written or just see if those rogue codes tend to be in the beginning of a session.

##

7) In the EventTranslator_longBase code there is some workaround to get the Fix Acquire Time. Is this time referring to the onset of the fixation spot or does it refer to the time when the animal starts fixation? It seems to utilize additional photo diode strobes. However, I have a hard time to understand what is going on in the snippet. It seems to work only for D and E where codes were used for the fixation spot, and this was accompanied by a photo diode event. However, instead looking for the first photo diode flash after the fix onset event code something different is done in the code below. I have no clue what was the reasoning behind this code. What is going on there?

For this you have to realize that determining the exact time monkey fixates is a really big problem. He might dart inside and out of the fixation box. He might break fixation, then come back and start a trial. There are so many possibilities. In Tempo, I exploited a variable that codes (in real-time) whether or not the eye was in a window. Each time the eye entered the window, I wrote a code. However, since the monkey is free to do as he pleases, we really only want to keep the ***last*** time that occurred, because he would have had to enter, then stay, for a specified duration before the stimulus turned on.

You mentioned an additional photo diode event – perhaps you are referring to a strobe for the moment the fix spot turned on. It used to be that just the Tempo-coded value was written, but those are terrible, so eventually I went in and added a new photo diode event. I did that after starting with D & E, which is why it only works for them.

```
          %============================
        %============================
        % Find Fixation Acquire Time
        %FOR FIXATION ACQUIRE TIME, NEED TO HAVE THE PREVIOUS TRIAL AS
        %WELL. ALSO MAKE SURE THIS DOES NOT FAIL ON THE FIRST TRIAL
        if i == 1;
            TrialEvents_AcquireTime = find(Strobe_Times <
TrialStart_TimeStamps(i+1));
        end

        if i > 1 && i < maxTrialNo*multifilenameNo+residual;
            TrialEvents_AcquireTime = find(Strobe_Times >= TrialStart_TimeStamps(i-
1) & Strobe_Times < TrialStart_TimeStamps(i+1));
        end

        %if last trial
        if i == maxTrialNo*multifilenameNo+residual;
            TrialEvents_AcquireTime = find(Strobe_Times >= TrialStart_TimeStamps(i-
1));
        end

        TrialStrobeTimes_FixAcq  = Strobe_Times(TrialEvents_AcquireTime);
        TrialStrobeValues_FixAcq = Strobe_Values(TrialEvents_AcquireTime);

        %Find all 'fixation acquired' strobes.  There may be several
        AcquireTimes = TrialStrobeTimes_FixAcq(find(TrialStrobeValues_FixAcq ==
31))*1000 - (TrialStart_(newindex,1) + BASELINE);

        %The fixation acquired strobe we want to keep in the latest one
        %that occurs before target onset. We already corrected them, so it
        %should be the max negative number

        %These are to be interpreted as ACTUAL times.  I.e., if FixAcqTime == -800,
then fixation was acquired
        %800 ms before the array appeared.
        if ~isempty(AcquireTimes)
            FixAcqTime_(newindex,1) = floor(max(AcquireTimes(find(AcquireTimes <
0)))); %using floor because Tempo is going to be late in sending code, not early.
        end
        %============================
        %============================
```

On 08/13/2015 12:00 PM, Heitz, Richard P wrote:

> Hi Wolf. Strange you emailed me; I was actually going to email you the other day to let you know that you were RIGHT about R. It is the best, and I'll never use Matlab again unless I absolutely have to.
>
> You are correct, the first target_on_Evt2 time *after* 1666 is the real stimulus onset time. The reason there are multiple events is simply due to the fact that the photo diode fires each time the screen refreshes. If you look at the diff(), it should correspond pretty closely to the refresh rate of the monitor, though for some reason I don't think it continued to fire for the entire 80 or 100 ms the stimulus was visible...don't quite remember. Those codes are received completely differently from all other event codes -- the event codes come in as character strings, while the EVT2 is a BNC port waiting for, basically, a TTL.
>
> Also, regarding the event codes, I seem to recall that most of the important information is written *after* the trial (i.e., after 1667), so be careful not to exclude them. You also have to be careful because some of those post-trial event codes will repeat if say, you have a completed trial followed by some broken fixations/aborts. In my translator code, you will see a portion that limits the event code search to (1:relevant), where relevant was some index I could be certain still belonged to the previous completed trial.
>
> Oh also, in practice I found that a 500 ms baseline wasn't long enough. I later increased it quite a bit to 2000 or something like that. You might want to keep that in mind to avoid any problems later.
>
> -----Original Message-----
>
> From: wolf zinke [mailto:wolf.zinke@vanderbilt.edu]
>
> Sent: Thursday, August 13, 2015 11:46 AM
>
> To: Heitz, Richard Philip
>
> Cc: Heitz, Richard P
>
> Subject: old plexon format again
>
> Hi Rich,
>
> Sorry to come back to this one. I am back on my attempt to convert the

old plexon format into a more standardized matlab format I am using now for recent data.

As I understand there is no clear definition for the event codes to what trial they belong, so I right now constrain the event codes to the time window defiined by the TrialStart_ code (1666) and the first subsequent Eot_ code (1667) ignoring for now that there are additional codes outside this windows that might belong to this trial.

I try now to identify the time for the target onset.

If I understand it correctly the target onset is defined in the Event_Translator.m file as:

```
        %Target align time
        Target_(newindex,1) = ceil(target_on_Evt2
(i)*1000)-TrialStart_(newindex);
```

Trial start was defined a couple of line above this as:

```
        TrialStart_(newindex,1) = ceil(target_on_Evt2(i)*1000) - 500;
```

Basically this means that Target_(:,1) = TrialStart_ + 500;

target_on_evt2 is defined in the file instance above (e.g. Translate_SAT.m or macTranslate_SAT.m) as

```
        target_on_Evt2(ii_t) = Strobe_Times_2(min(find(Strobe_Times_2 >
TrialStart_TimeStamps(ii_t))));
```

Here, TrialStart_TimeStamps refers to the event code 1666, and it seems that the first photo diode event is used as target onset. However, when I check for photo diode occurrences within my trial time window I

> get up to four photo diode strobes (see below an example, i play right now with). It does not seem that there is a tempo event code for stimulus onset that I could use to clearly identify the photo diode event. Is it correct that the first code is the stimulus onset, or does it refer to fixation onset? What is the meaning of these four photo diode flashes?
>
> thanks for the help,
> wolf

No, it IS the trial start, at least from the perspective of TEMPO. if the diode fires too, you are confident that it is real.

If he gives a correct response, the sequence will be:

1666
diode
..stuff
1667
…lots of stuff

1666…

Does that make sense? I know this is totally confusing...

Richard P. Heitz, PhD
Center for Integrative and Cognitive Neuroscience
Vanderbilt University
Office: 527 Wilson Hall
Web: http://www.heitz.org

> On Apr 8, 2015, at 5:39 PM, Zinke, Wolf <wolf.zinke@vanderbilt.edu> wrote:

but 1666 is not the trial start, correct? How would I find out when codes for the next trial starts if he gave a correct response?

On 04/08/2015 05:30 PM, Heitz, Richard Philip wrote:

> another point, if you didn't already know. what can happen often is:

1666…. then monkey screws around, so 1667, abort, codes. codes. codes…..then 1667, abort…codes..codes…codes…then 1667, abort…codes..codes..codes…THEN 1666..trial is real.

so when you find 1666 -> 1666, the relevant event codes are just up until the first abort. that's what that (1:relevant) line is doing in my EventTranslator code

> ---

Richard P. Heitz, PhD
Center for Integrative and Cognitive Neuroscience
Vanderbilt University
Office: 527 Wilson Hall
Web: http://www.heitz.org

> On Apr 8, 2015, at 4:13 PM, Zinke, Wolf <wolf.zinke@vanderbilt.edu> wrote:

Hi,

Did you have a chance to remember the logic of the codes? Most relevant question for me really is how I could identify the complete chunk of codes that belong to a trial. If there is a reliable way to identify such a block I likely will be able to figure out most of the other things by myself. But so far I was not able to identify anything that might be a trial marker that comprises all codes.

cheers,
wolf

On 03/31/2015 07:40 PM, Heitz, Richard Philip wrote:

Yikes that's a lot. I'll get you answers but give me a few days!

Sent from my iPhone

> On Mar 31, 2015, at 2:31 PM, Zinke, Wolf <wolf.zinke@vanderbilt.edu> wrote:
>
> Hi Rich,
>
> I work on the fun part now to convert various plexon files into a common format that contains all relevant information in an easily accessible way. I started with data created by Josh, and I think he used a tempo code configuration that was based on files from Dave Godlove. This worked out pretty well, and I want to adapt the code now to cope with the T/L search data and SAT data..
>
> I realized that the way of encoding information is very different in these data files. I reverse engineered the tempo configuration files you gave me, as well as various translation matlab files. This gave me a pretty good idea what to do, but I am stuck in details where it might be very simple for you to quickly provide me the relevant information when me digging through thousands lines of tempo code.
>
> Thirst puzzling thing I noted was that far more end of trial codes (1667) are found than trial start codes (1666). I ignore this for now and loop in a stupid way over all trial start codes to find the first trial end code that occurs afterwards in order to define events corresponding to one trial.
>
> But anyway, why are there far more EoT codes than trial start codes? This does not make any sense to me.
>
> The trial start seems to be used in a slightly different way , it seems to encode the time when the animal starts fixation (codes 30 and 31 for FixOn and Fixate precede 1666). Do you have codes that clearly identify the code block that belongs to a single trial, i.e.

marks the start and end of the codes?

Am I reading it correct that codes for events and codes containing information about the trial are intermixed? Dave/Josh did use markers to encode the start and end of an information block, that follows the event time codes of the trial itself, this si not the case for the previous plexon data, right? How do you identify the information that belongs to the current trial, if it could preced the trial, could occur during a trial, or afterwards?

Am I right that codes like 8000+lTgtP (target position), 8100 and 8300 are only used to indicate saccade information for online use with NeuroExplorer but are irrelevant for the data translation (this code does not seem to be utilized in any of the matlab translation files)?

There are also codes with 9000 plus something preceding 1666. This encodes the target position for the current condition for use with NeuroExplorer, correct? Therefore, it should be consistent with the position indicated by the code following a TargetPos code (3013) and is redundant information, right?

There might more question come up, but right now the part I am stuck ist the assignement of codes to a clear trial organization.

Best,
wolf