Trigger latency testing, December 2019

#work/stimulation

Simple audio-visual stimulation

- `python=3.6.x`
- `psychopy=3.3.x`
  - using new `'ptb'` audio backend in 'agressive' mode (3)
  - `winType=pyglet`
  - visual dot stimulus of 4 frames (33.33 ms)
  - sound synthesised as 200 Hz beep with 3.33 ms shorter = 30 ms duration
- parallel port triggering
  - value `1` sent with `callOnFlip` at onset of stimulus (dot)
  - audio targeted to `when=nextFlip` using `ptb`
  - optodiode on `MISC004`, audio recorded on `MISC005`

Stimulation script

audio_syncToFrame_test.py
24 Dec 2019

Analysis script

av_latency_win_lin.py
4 Jan 2020

Data files

```
aud_vis_4x50frames_1screen_linux.fif
aud_vis_4x50frames_1screen_win.fif
```

Recommendations

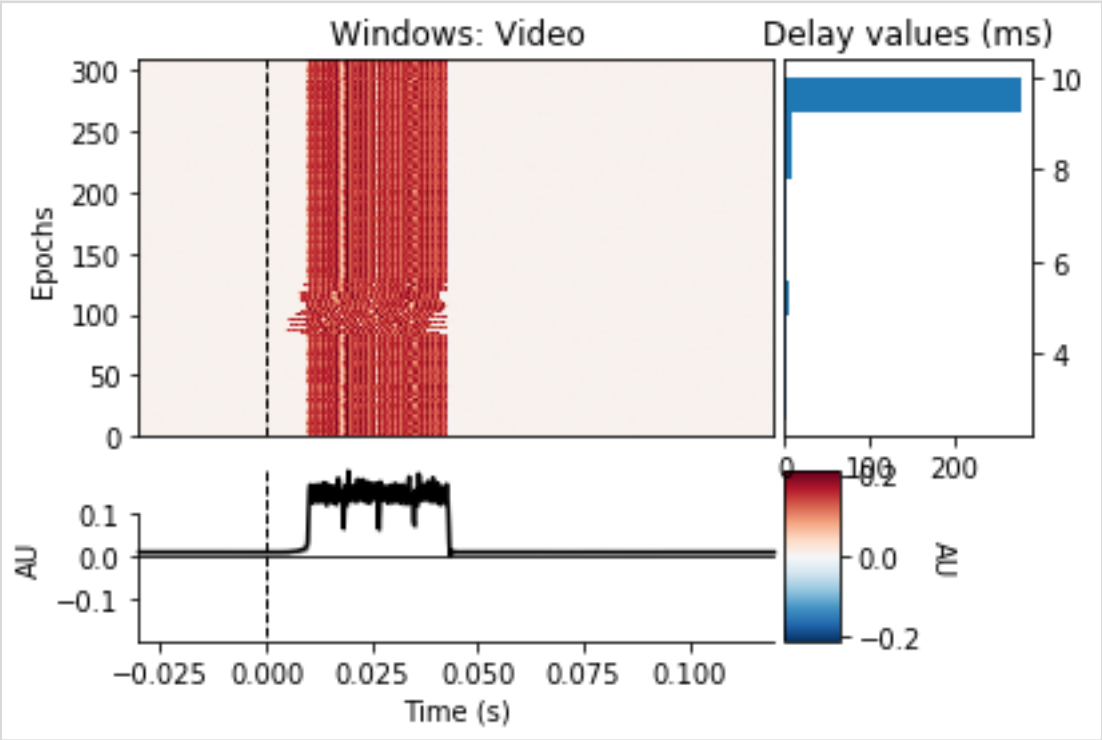
- `ptb` in Linux rocks! Very low-latency, low-jitter stimulation possible
  - useless in Windows (sucky Win10 Creative driver likely culprit)
-

Results: Windows

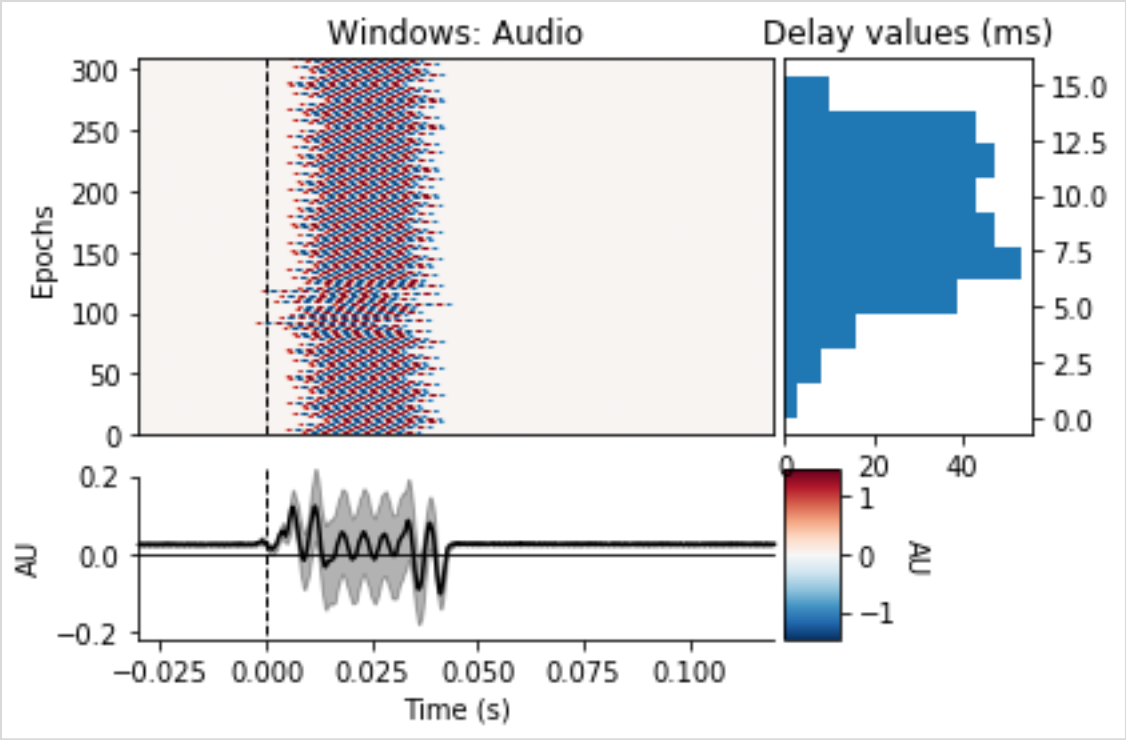
```
Windows: Video
```

```
==============
  Median=9.8 ms [9.4, 10.0] ms [Q10, Q90]
```

Very consistent delay, almost no jitter except for a period around epoch 100 (maybe the OS doing something?).



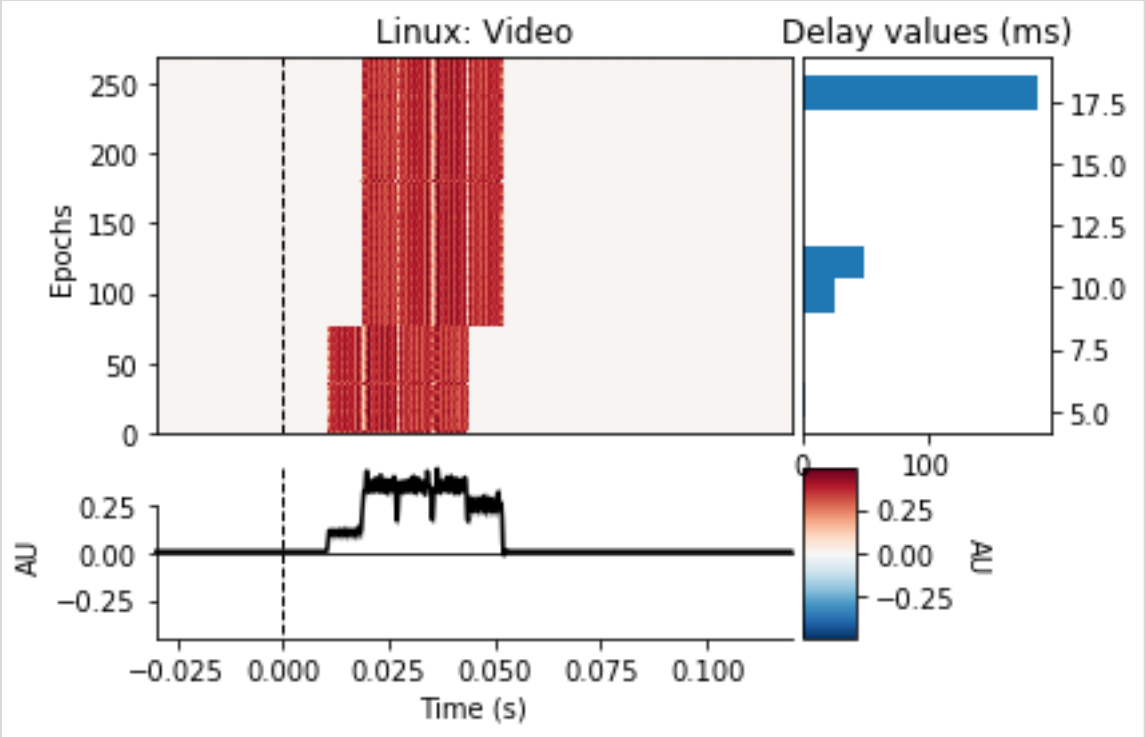

```
Windows: Audio
==============
  Median=9.0 ms [4.8, 13.0] ms [Q10, Q90]
```



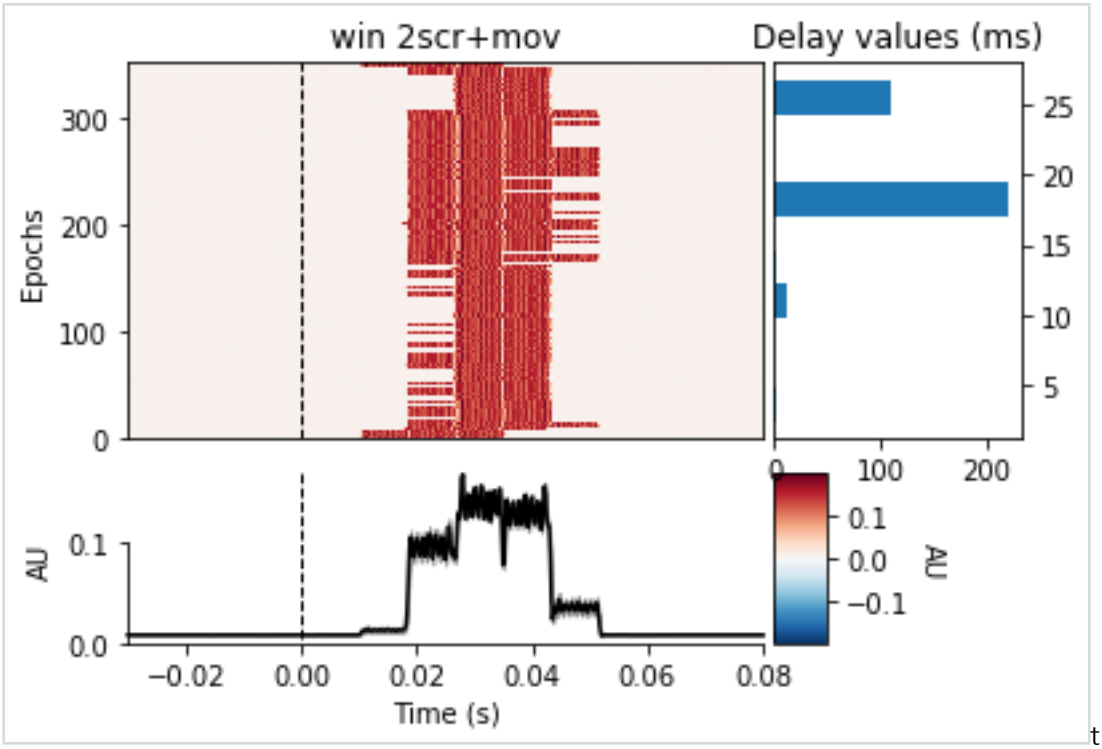
Clearly something badly wrong with the audio driver on Windows! Massive jitter around a long delay—useless.

Results: Linux

```
Linux: Video
============
  Median=18.4 ms [10.2, 18.4] ms [Q10, Q90]
```

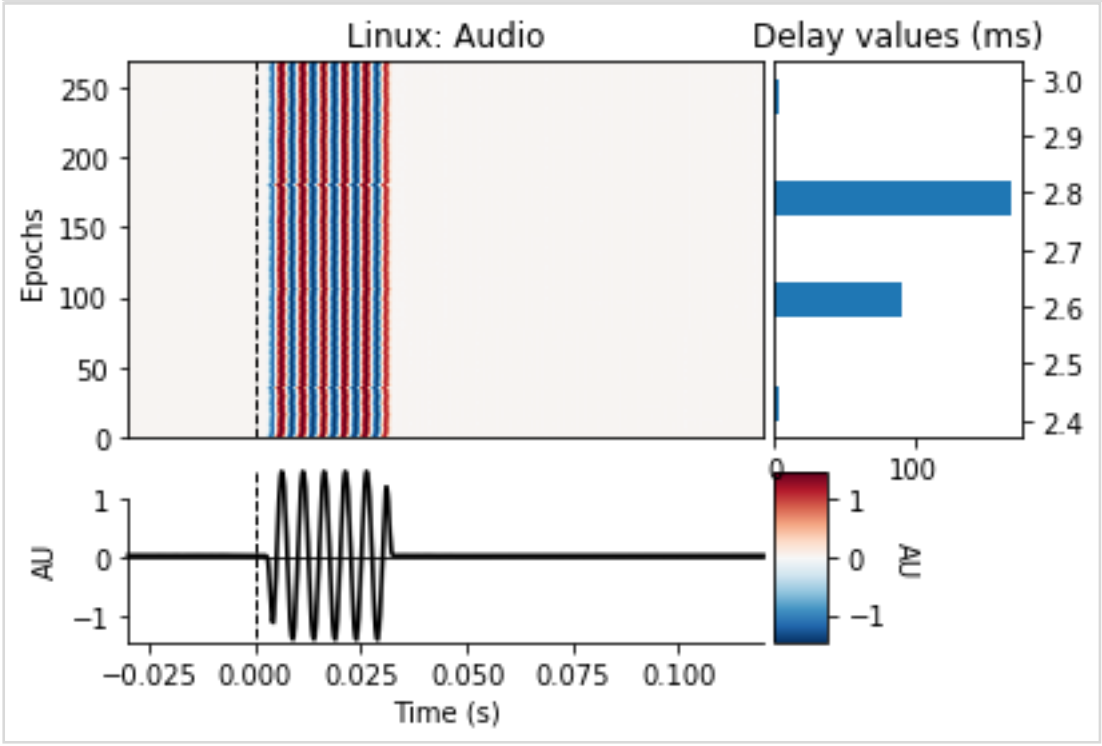


This is a very consistent observation: the delay starts off being stable a



t around 1 frame + few ms. Then it increases by exactly one frame, and stays constant for remainder of script! What's going on?

```
Linux: Audio
============
  Median=2.8 ms [2.6, 2.8] ms [Q10, Q90]
```



Beautiful!

Rapid visual stimulation with multiple screens

- Python and PsychoPy as above, parallel port triggering
- trigger value (2 or 3) indicates length of stimulus dot in frames
  - stimulus alternates with a cycle of 10 frames
- second screen attached to DisplayPort output & fed to matrix switch using a DP ➔ DVI converter
  - designated "secondary" in Nvidia settings
- three repeats for both OS's
  1. 2 screens, with lots of moving and resizing windows on 2nd
  2. 2 screens, with no movement
  3. 1 screen only

Stimulation script

video_rapidrate_test.py
21 Dec 2019

Analysis script

rapid_vis_2scr_win_lin.py
4 Jan 2020

Data files

```
rapid_visual_lin_1screen.fif
rapid_visual_lin_2screens_nomov.fif
rapid_visual_lin_2screens_withmov.fif
rapid_visual_win_1screen.fif
rapid_visual_win_2screens_nomov.fif
rapid_visual_win_2screens_withmov.fif
```

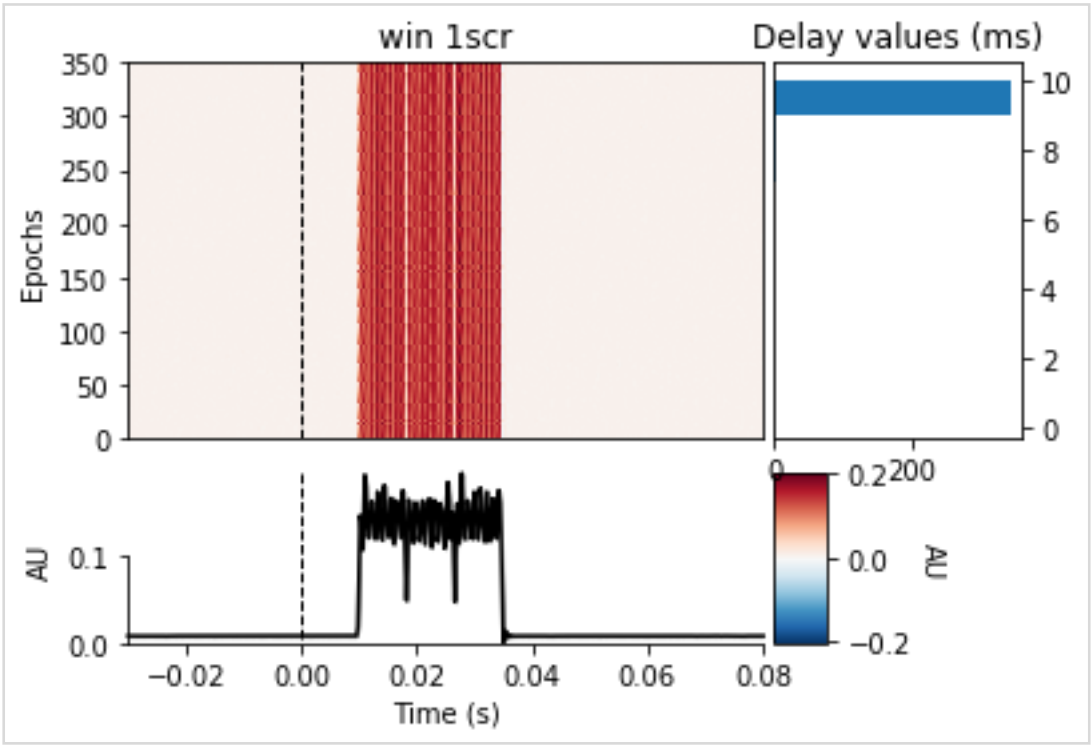
Recommendations

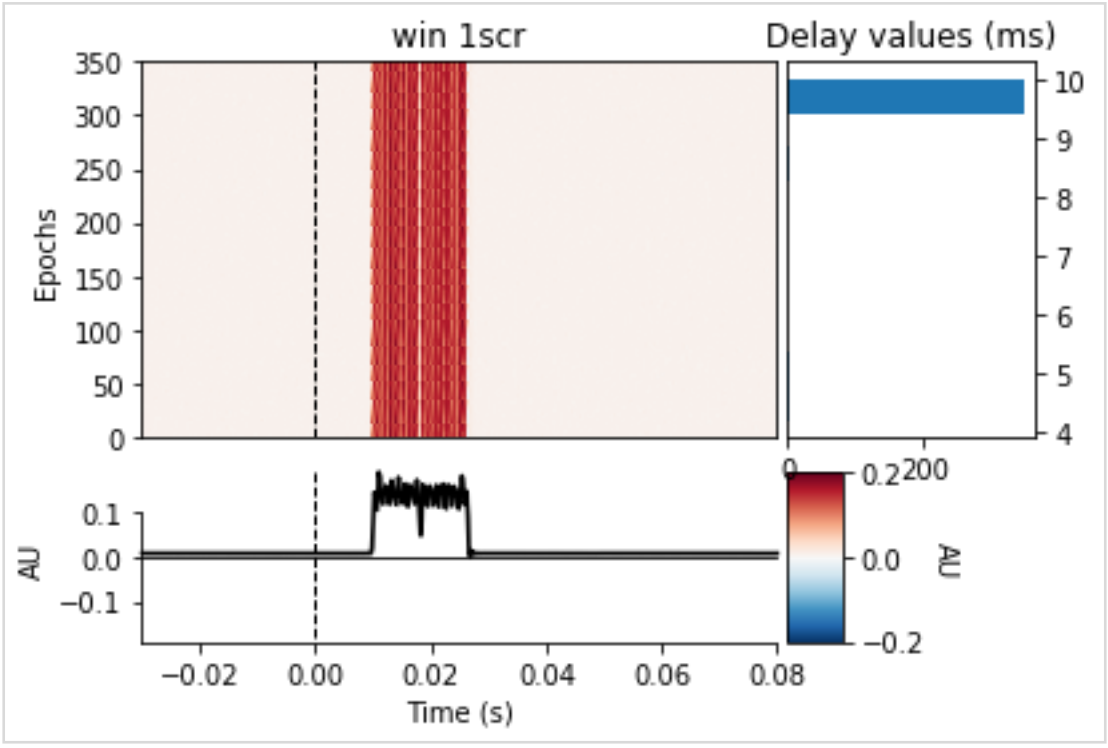
- in single screen mode, Windows seems to be stable at ~10 ms delay
- Linux has a weird shift (by 1 frame) during the course of a script
  - has to be solved before Linux can be used for rapid visual stimulation
- addition of 2nd screen seems OK for Windows, and even stabilises Linux!
- lots of GUI activity on 2nd screen destroys Windows timing, but Linux might still be OK?

Results: single screen

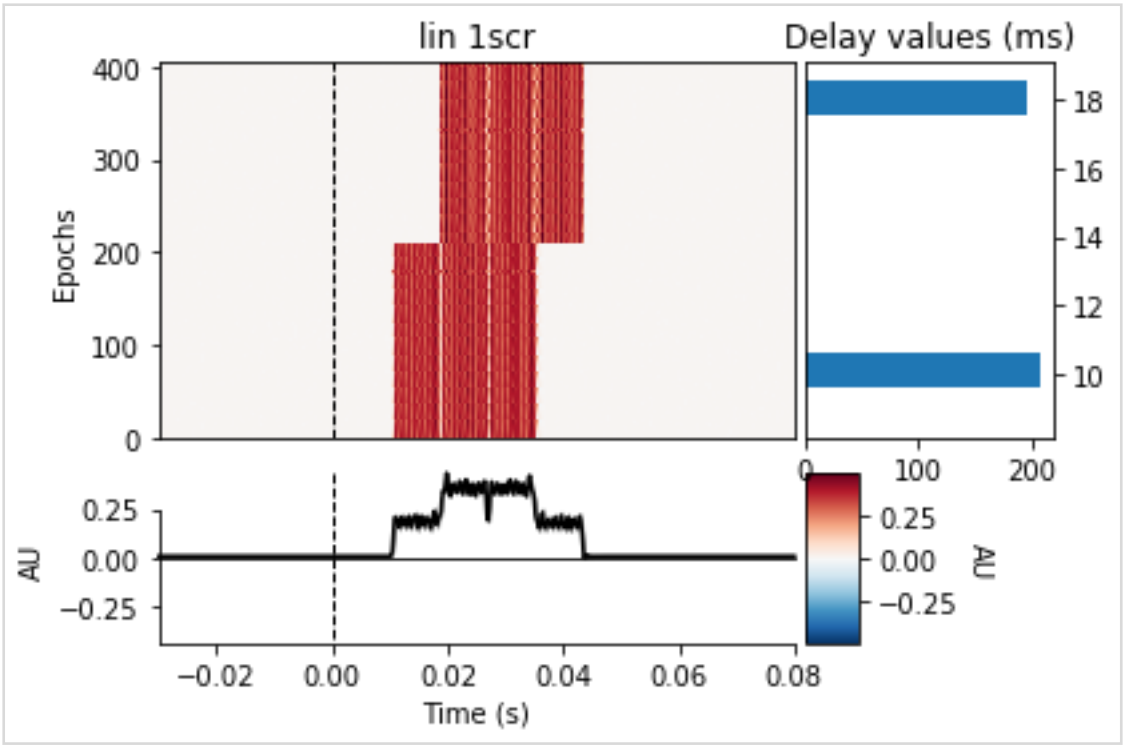
Windows and Linux initially have same absolute delay (~10 ms), but after 200 epochs, Linux delay jumps to ~18 ms?!

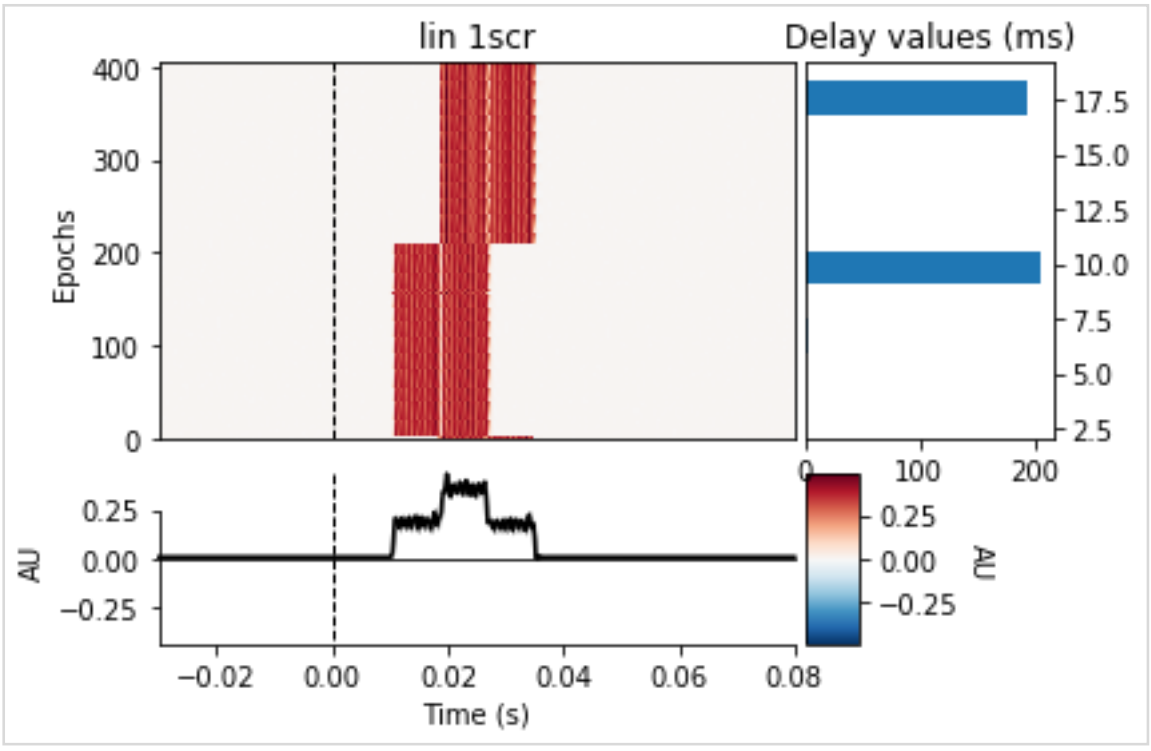
Windows





Linux



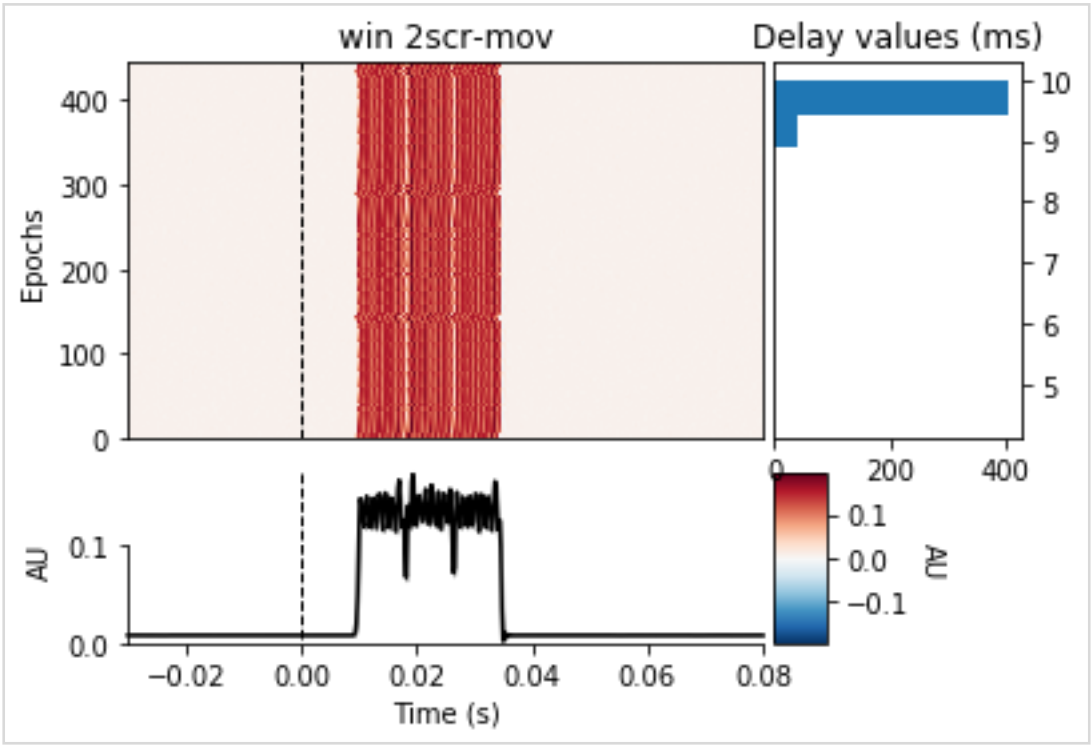


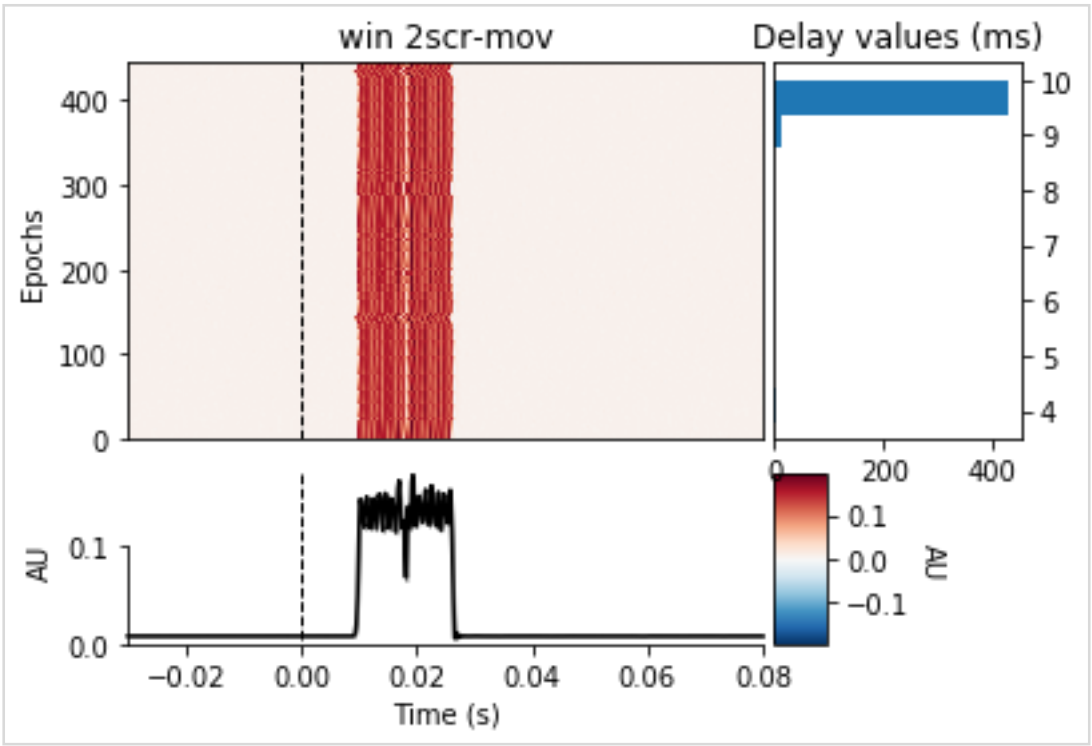
Results: dual screen, no movement on second

Windows seems to be more-or-less unchanged by addition of second screen, when nothing particular is going on on the second.

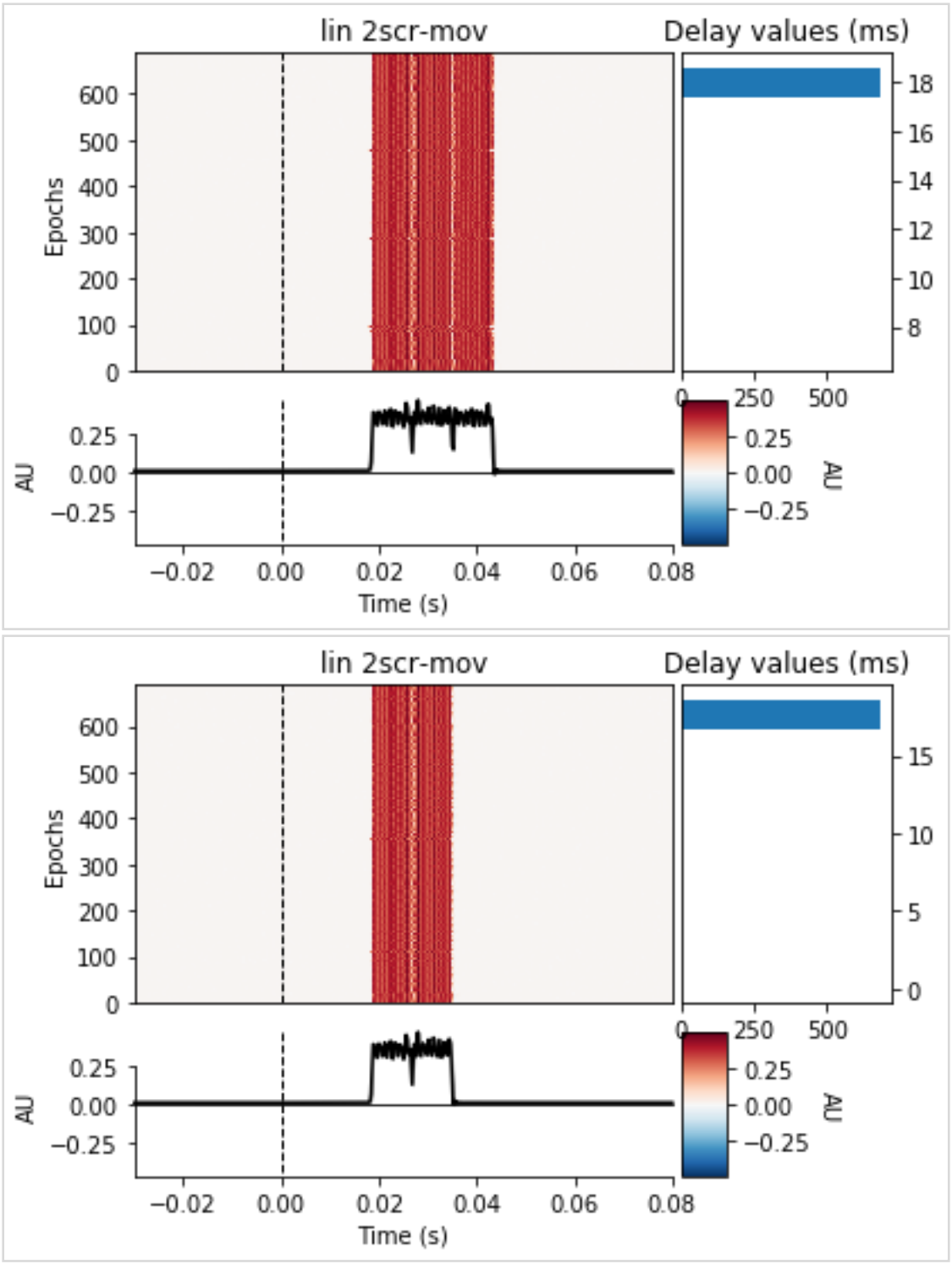
On Linux, the addition stabilises the delay to around 18 ms!?

Windows





Linux

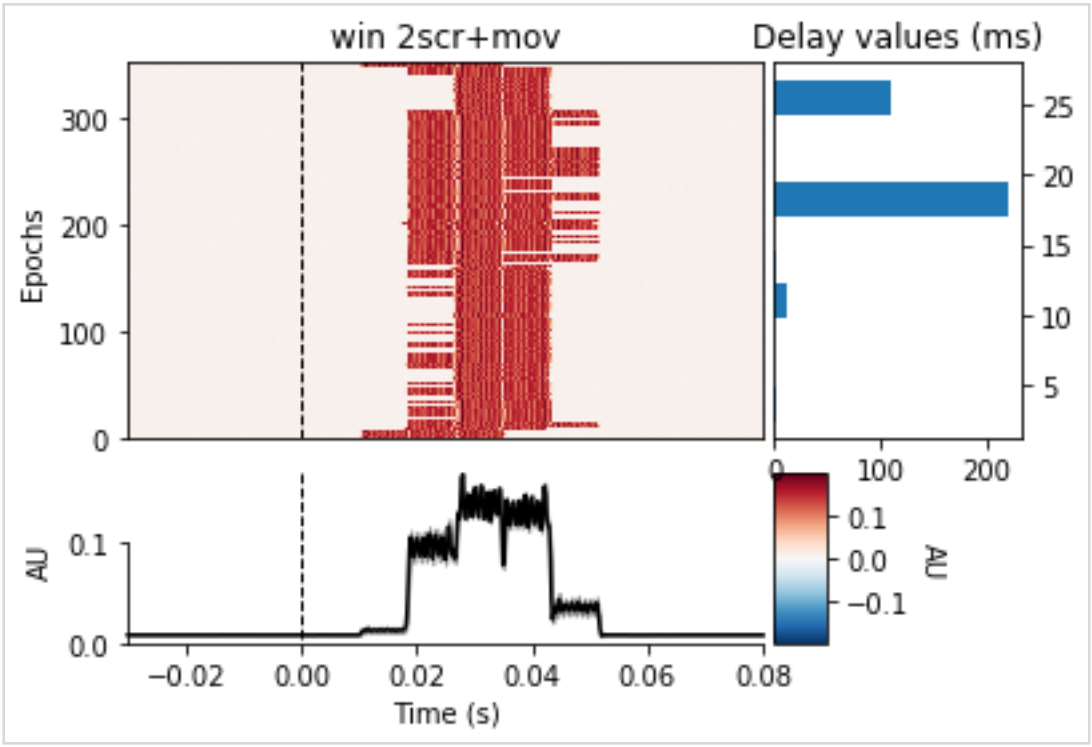


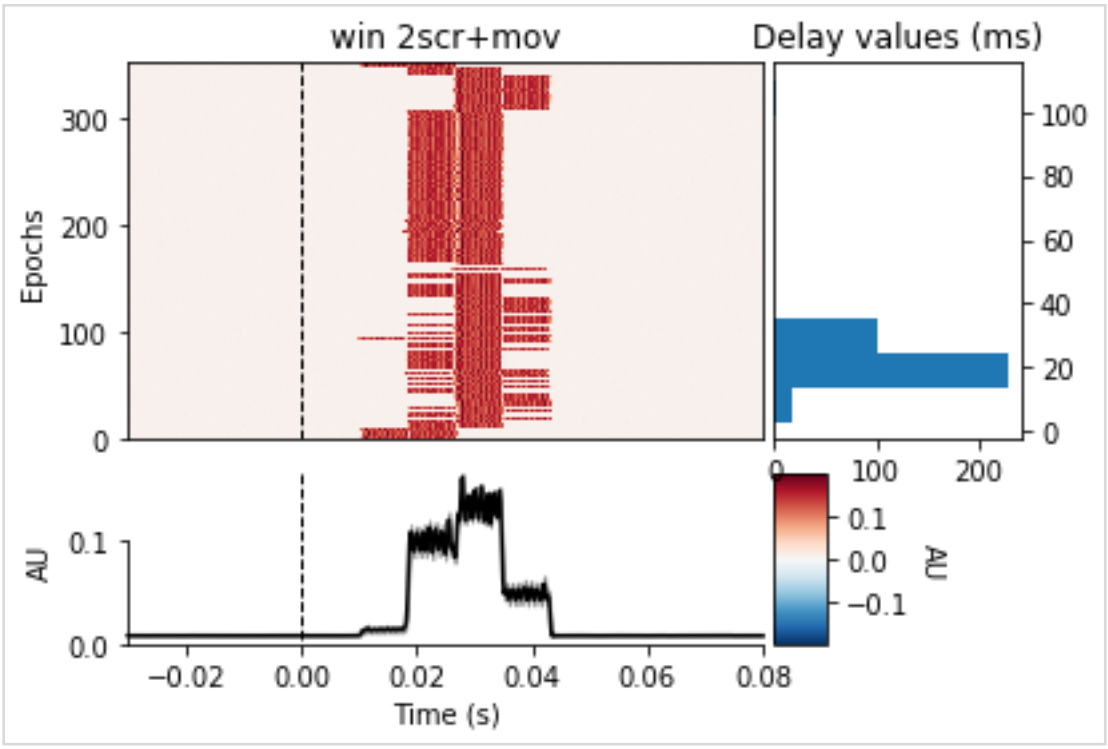
Results: dual screen, lots of action on second

Windows is clearly very unstable when the second screen is heavily used (moving and resizing windows). The bulk delay shifts to around 18 ms, but the jitter is unacceptable.

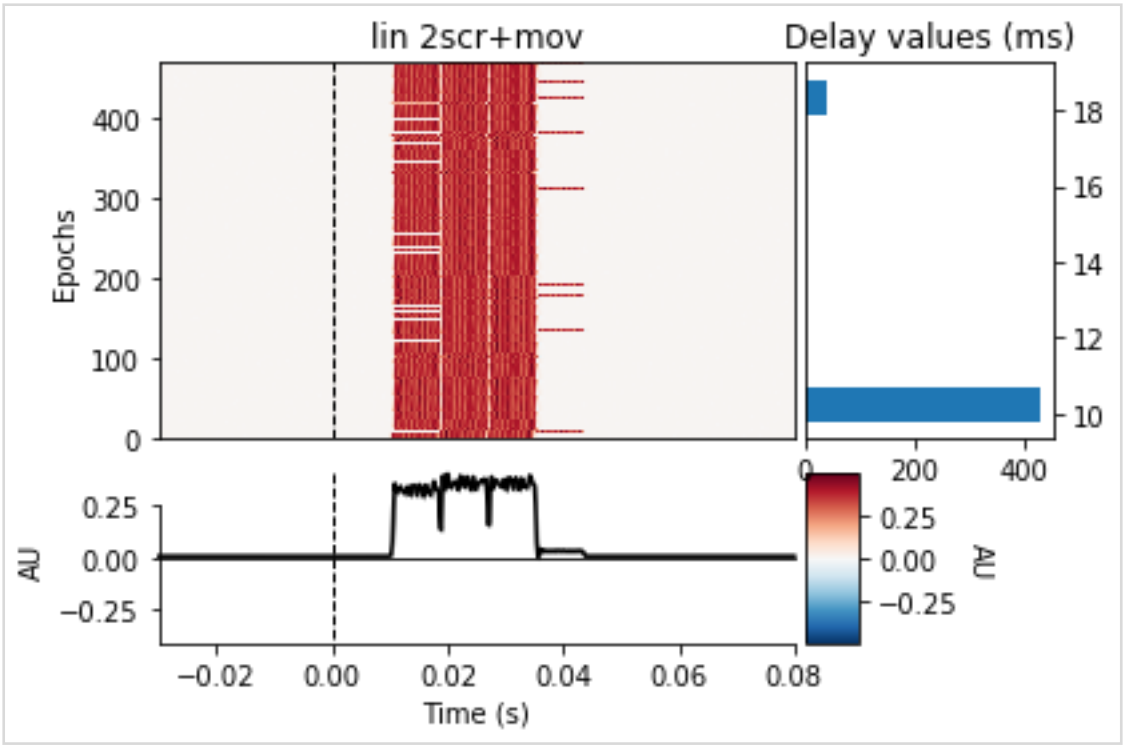
On Linux, the bulk delay reduces to ~10 ms!? However, the occasional one-frame additional delay is only borderline-acceptable.

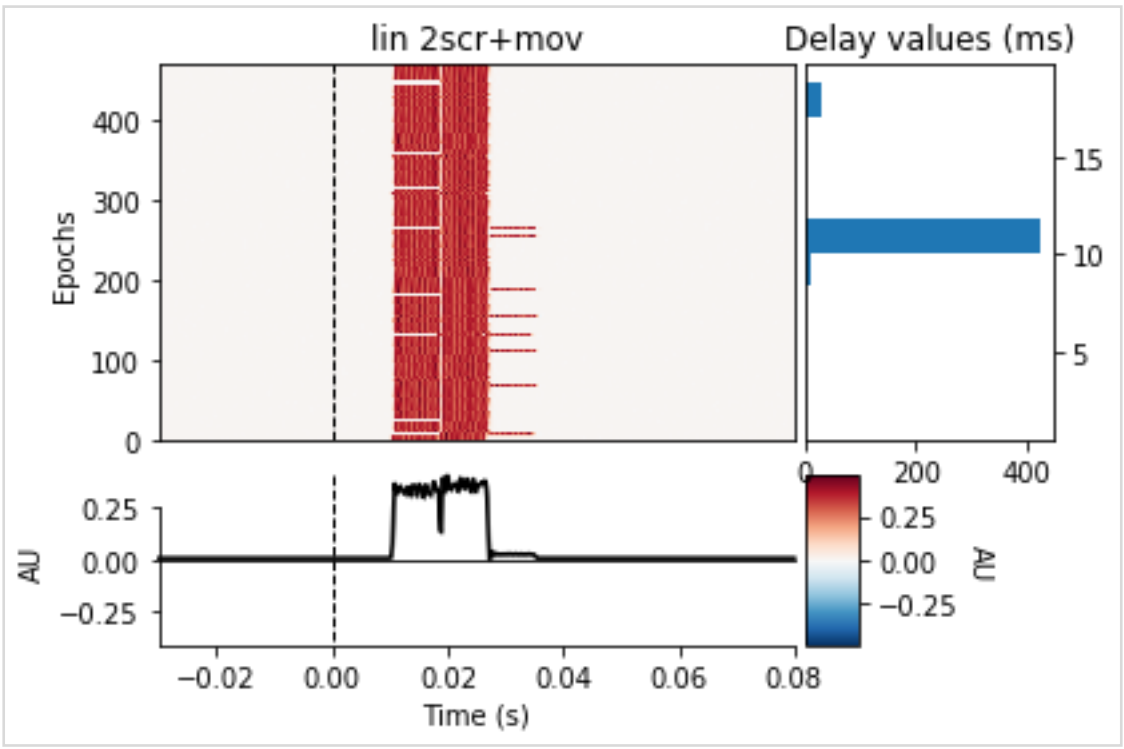
Windows


win 2scr+mov
Delay values (ms)
Epochs
AU
Time (s)
AU


win 2scr+mov
Delay values (ms)
Epochs
AU
Time (s)
AU

Linux


lin 2scr+mov
Delay values (ms)
Epochs
AU
Time (s)


lin 2scr+mov
Delay values (ms)
Epochs
AU
Time (s)