LSL hands-on:

What did we learn comparing EEG and Unity data streams?

Tea Time 16.04.2020 - Marc Vidal De Palol

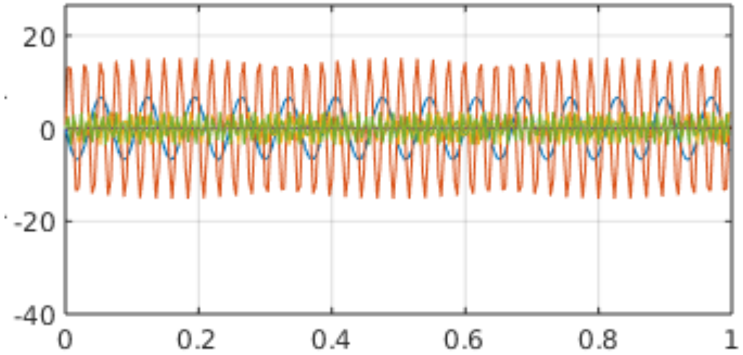
Outline

1. Why aligning data streams is important?
2. Delay, jitter and latency
3. What is LSL (lab streaming layer)?
4. Designing the test
5. How did we analyze the data?
6. Results
7. Conclusions and remarks
8. Future outlook

1. Why aligning data streams is important?

Data streaming devices have different:

- Sampling rates
- CPU clocks



- Plus: they can be connected via network (LAN or WLAN) -> delays

1. Why aligning data streams is important?

Example:

1. Why aligning data streams is important?

Example:

- Westdrive (Unity): 60 FPS

- Toby eye tracker: 90 Hz

- TMSI Rega EEG amplifier: 1024 Hz

2. Delay, latency and jitter

CLIENT
SERVER

Differences:

- Delay: time for some data to move from one endpoint to another
- Latency: one-way delay
- Jitter: delay inconsistency between each packet

Delay contributors:

- Processing: package analysis time
- Queueing: time between being queued and sent
- Transmission: time to push the data into the wire
- Propagation: time influenced by the distance

Source: https://www.callstats.io/blog/2018/03/07/difference-between-jitter-and-latency (https://www.callstats.io/blog/2018/03/07/difference-between-jitter-and-latency)

3. What is LSL (lab streaming layer)?

Two ways of solving the alignment of time series data:

- manually-> more time consuming, but more control
- automatically via soft solution (e.g. LSL) -> less time consuming, less control

Also advantages of LSL:

- open source
- cross platform (Win, Linux, MacOS, Android, iOS)
- multi API language interfaces (C, C++, Python, Java, C#, MATLAB)
- many tools around it
- scientific community support
- XDF stored data

Source: https://labstreaminglayer.readthedocs.io/info/intro.html#what-is-lsl (https://labstreaminglayer.readthedocs.io/info/intro.html#what-is-lsl)

3. What is LSL (lab streaming layer)?

How to use LSL? Requirements:

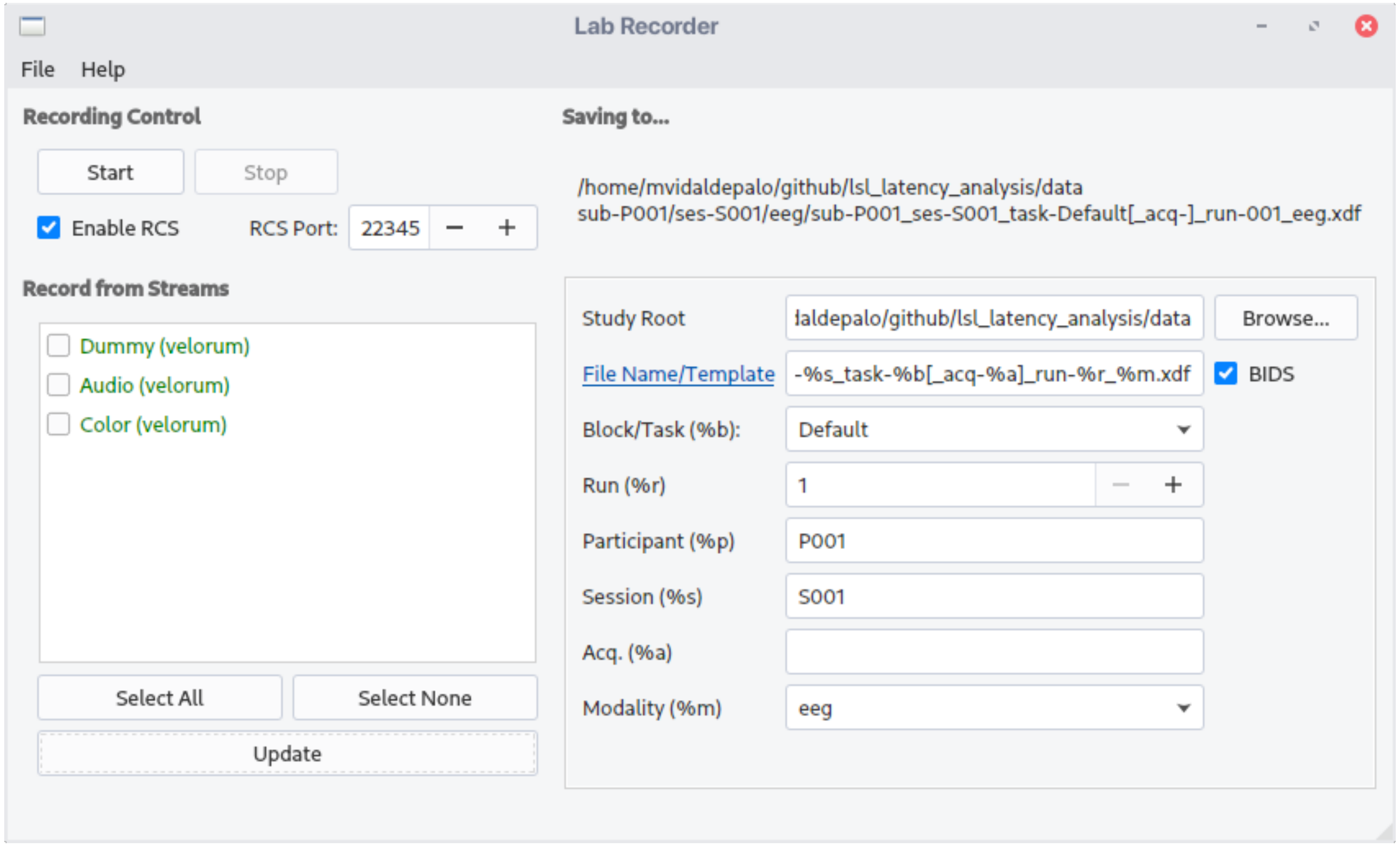
- LabRecorder app
- liblsl library with code defining your data streams

  and/or

- LSL community app

3. What is LSL (lab streaming layer)?

The Lab Recorder (with BIDS support):



☑ Enable RCS RCS Port: 22345

□ Dummy (velorum)
□ Audio (velorum)
□ Color (velorum)

/home/mvidaldepalo/github/lsl_latency_analysis/data
sub-P001/ses-S001/eeg/sub-P001_ses-S001_task-Default[_acq-]_run-001_eeg.xdf

Study Root: aldepalo/github/lsl_latency_analysis/data
File Name/Template: -%s_task-%b[_acq-%a]_run-%r_%m.xdf ☑ BIDS
Block/Task (%b): Default
Run (%r): 1
Participant (%p): P001
Session (%s): S001
Acq. (%a):
Modality (%m): eeg

3. What is LSL (lab streaming layer)?

Code example in C# defining a data stream:

```
using LSL;

public class yourClass
{
    private liblsl.StreamInfo lslStreamInfo;
    private liblsl.StreamOutlet lslOutlet;

    void startingMethod() { // normally Start() in Unity
        lslStreamInfo = new liblsl.StreamInfo(
            sName,
            sType,
            nValues,
            nominalRate,
            LslChannelFormat,
            uuid);
        lslOutlet = new liblsl.StreamOutlet(streamInfo);
    }

    void sendingMethod() { // normally FixedUpdate() in Unity
        var data = new float[size];
        lslOutlet.push_sample(data);
    }
}
```

4. Designing the test

Situation: Every 500ms a beep sound is played and the background color changes for one frame from black to white.

Unity (90 FPS):

- color change (black or white background)
- 100 ms beep sound (audio playing or not)

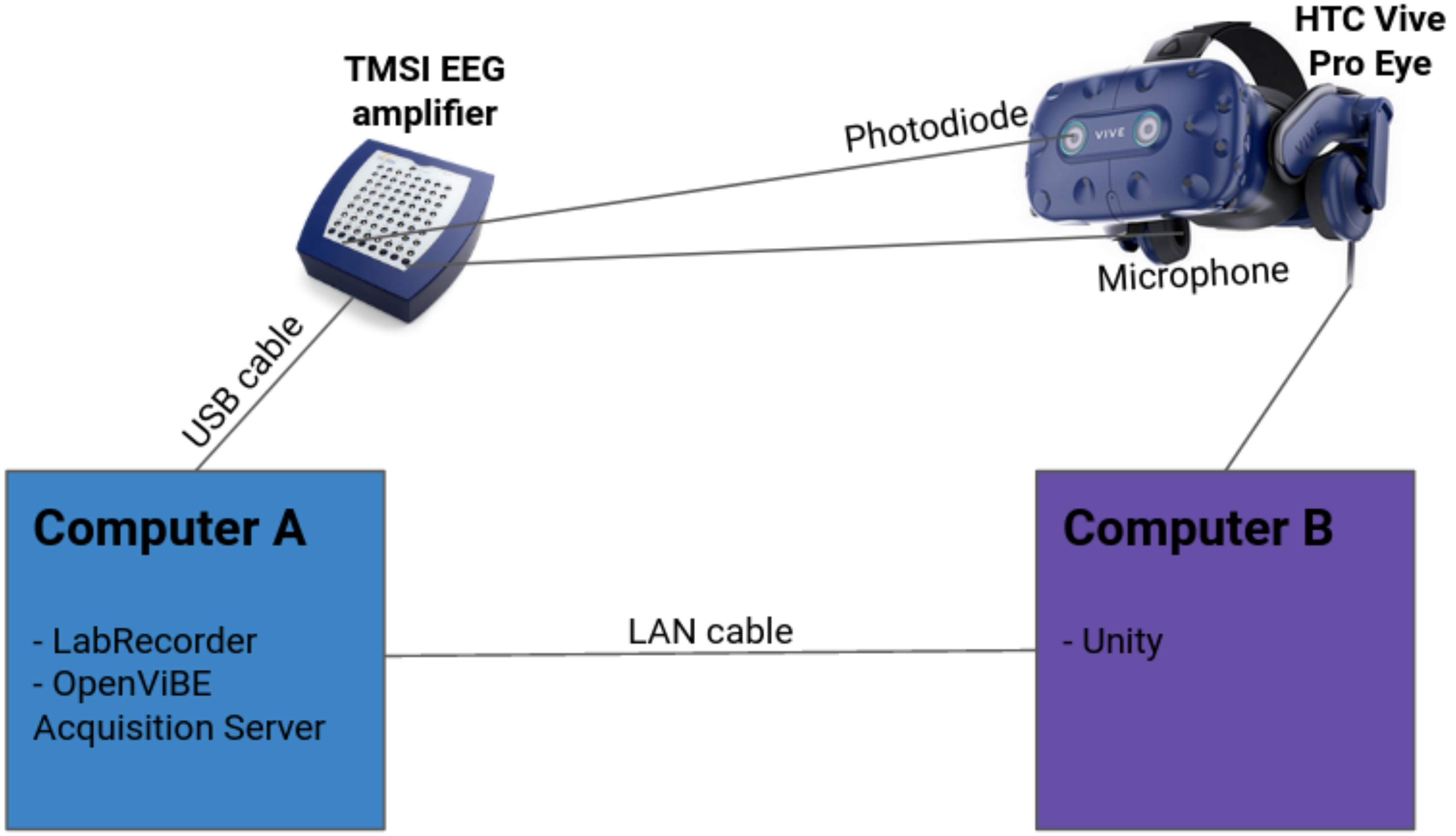
EEG (1024 Hz):

- photodiode (light sensor)
- microphone (audio sensor)

Recording setups with different:

- HMDs (Head-mounted display)
- computers (single and two LAN-connected)
- Unity builds
- long and short recordings

4. Designing the test



TMSI EEG amplifier

HTC Vive Pro Eye

Photodiode

Microphone

USB cable

Computer A

- LabRecorder
- OpenViBE Acquisition Server

LAN cable

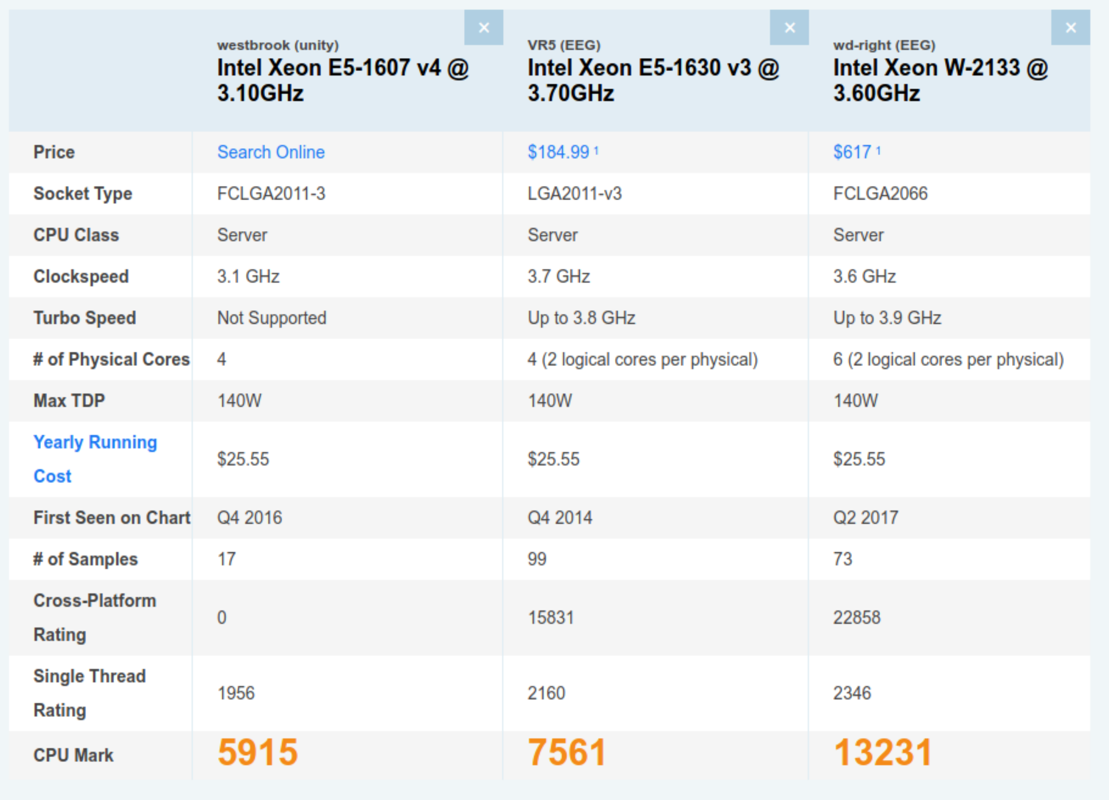
Computer B

- Unity

4. Designing the test

Computers specs:

	CPU	GPU	RAM
westbrook	Intel Xeon E5-1607 v4 (4) @ 3.10GHz	NVIDIA GeForce GTX 1070	32GB
VR5	Intel Xeon E5-1630 v3 (8) @ 3.80GHz	NVIDIA GeForce GTX 1080 Ti	32GB
wd-right	Intel Xeon W-2133 (12) @ 3.60GHz	NVIDIA GeForce GTX 1080 Ti	16GB

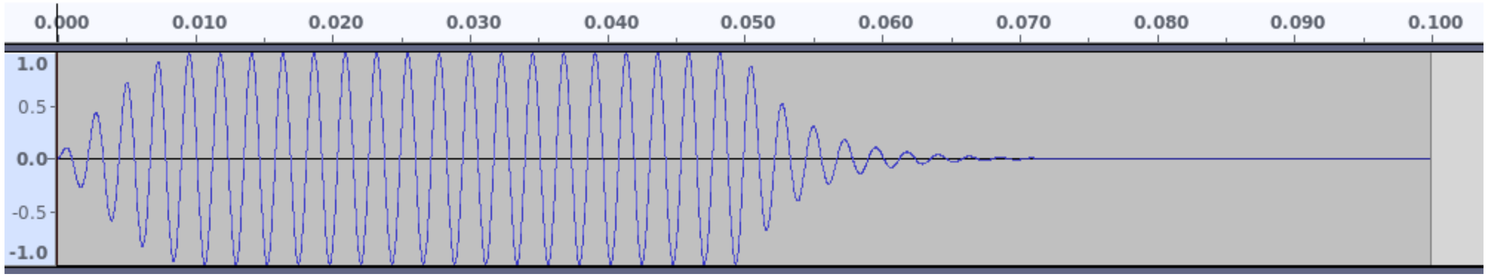


	westbrook (unity) **Intel Xeon E5-1607 v4 @ 3.10GHz**	VR5 (EEG) **Intel Xeon E5-1630 v3 @ 3.70GHz**	wd-right (EEG) **Intel Xeon W-2133 @ 3.60GHz**
Price	Search Online	$184.99 [1]	$617 [1]
Socket Type	FCLGA2011-3	LGA2011-v3	FCLGA2066
CPU Class	Server	Server	Server
Clockspeed	3.1 GHz	3.7 GHz	3.6 GHz
Turbo Speed	Not Supported	Up to 3.8 GHz	Up to 3.9 GHz
# of Physical Cores	4	4 (2 logical cores per physical)	6 (2 logical cores per physical)
Max TDP	140W	140W	140W
Yearly Running Cost	$25.55	$25.55	$25.55
First Seen on Chart	Q4 2016	Q4 2014	Q2 2017
# of Samples	17	99	73
Cross-Platform Rating	0	15831	22858
Single Thread Rating	1956	2160	2346
CPU Mark	**5915**	**7561**	**13231**

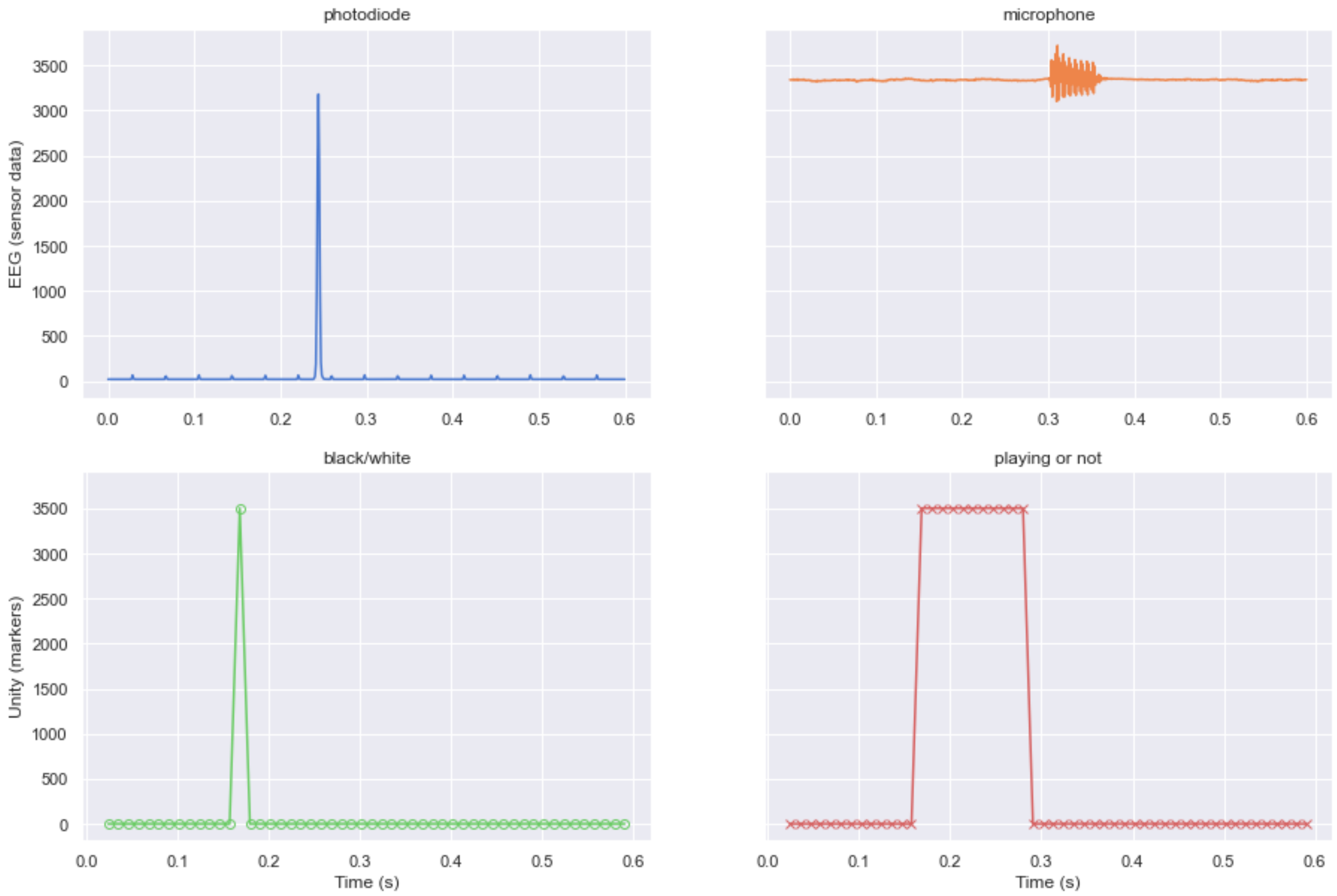
4. Designing the test

Background:

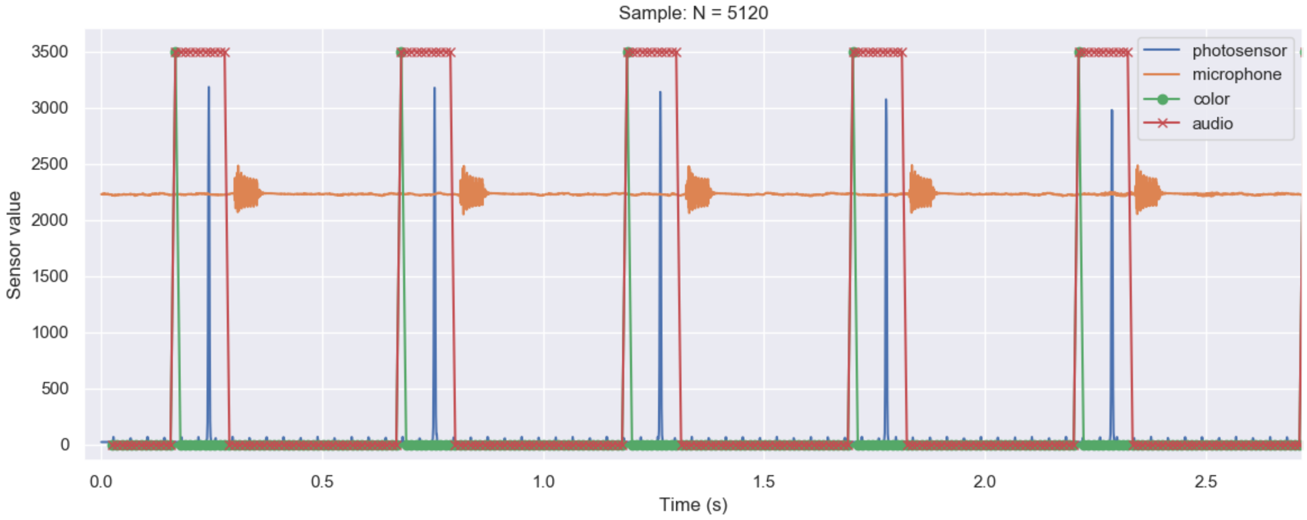
Audio:



5. How did we analyze the data?



5. How did we analyze the data?



1. Read the XDF files and select the right data
2. Recalculate the timestamps from 0
3. Visualize the data
4. Timestamps comparison (length, duration, sample count...): file info vs original vs calculated
5. Descriptive statistics of timestamps distributions
6. Peak detections and latency calculations

6. Results

How the recordings look like?

	eeg_starts	length_unity_audio	length_unity_color	lsl_aligned	two_computers	computers	duration (')
ftest1	✔	✔	✔	✔	✗	westbrook	2.03
ftest_build1	✔	✔	✔	✔	✗	westbrook	2.08
ftest_lsl12	✔	✔	✔	✗	✗	westbrook	2.21
ftest_build2	✔	✔	✔	✔	✗	westbrook	2.23
ftest3	✔	✔	✔	✔	✗	westbrook	2.28
ftest2	✔	✔	✔	✔	✗	westbrook	2.31
ftest_build3	✔	✔	✔	✔	✗	westbrook	2.43
long2	✔	✔	✔	✔	✔	westbrook & VR5	30.05
long3	✗	✔	✔	✔	✔	westbrook & VR5	30.81
long4	✔	✔	✗	✔	✔	westbrook & VR5	35.68
final_test	✔	✔	✔	✔	✔	westbrook & wd-right	65.81

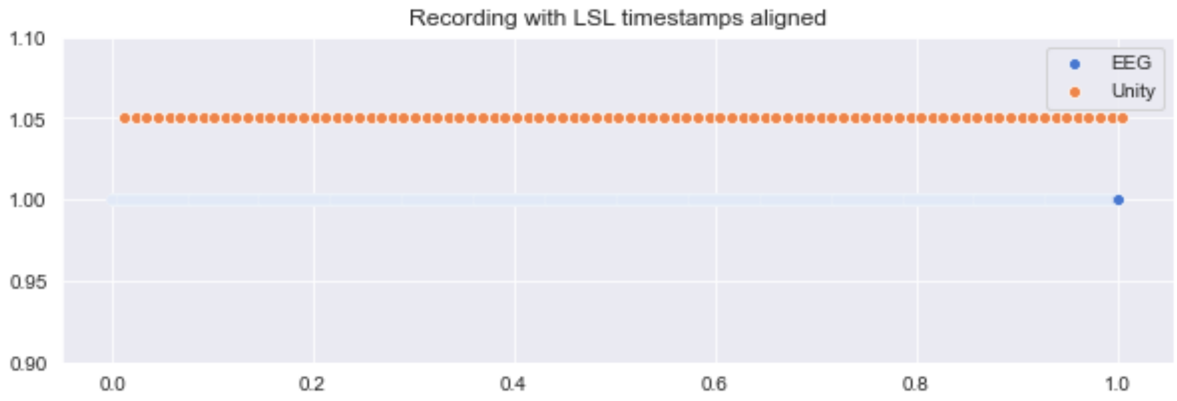
6. Results

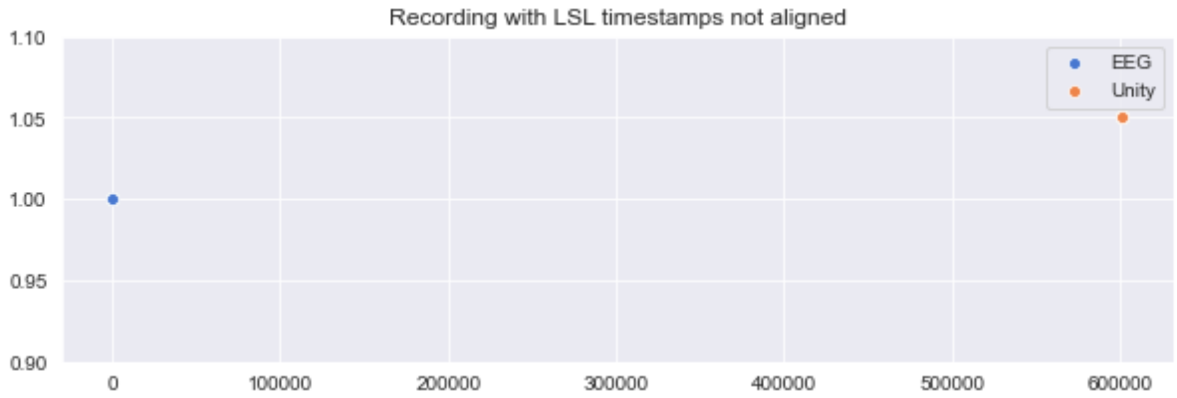
How the recordings look like?

- 4 long recordings (3x 30 minutes + 1x 1 hour)
- 6 short recordings (6x 2 minutes, 2 different unity builds)
- ftest_lsl12 removed from the analysis because timestamps were not aligned

6. Results

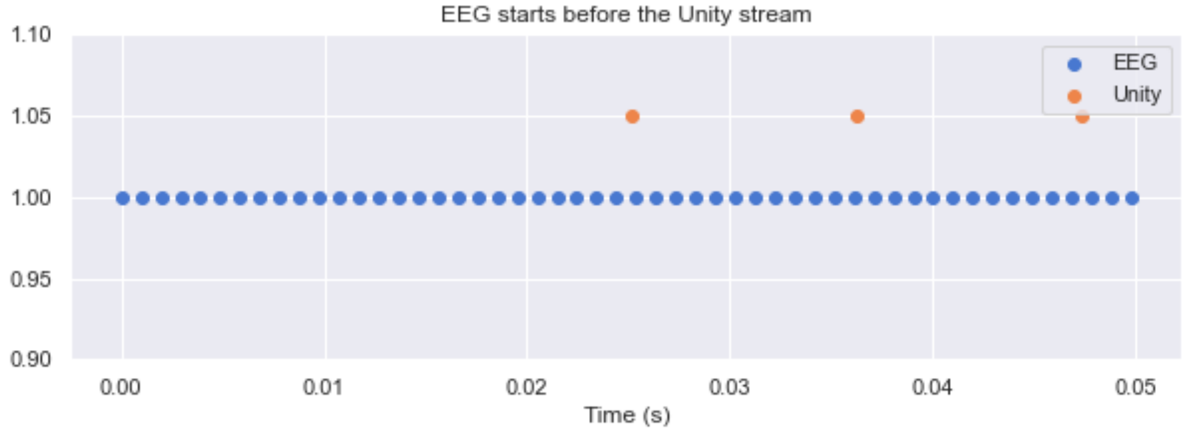
LSL timestamps: aligned vs not aligned

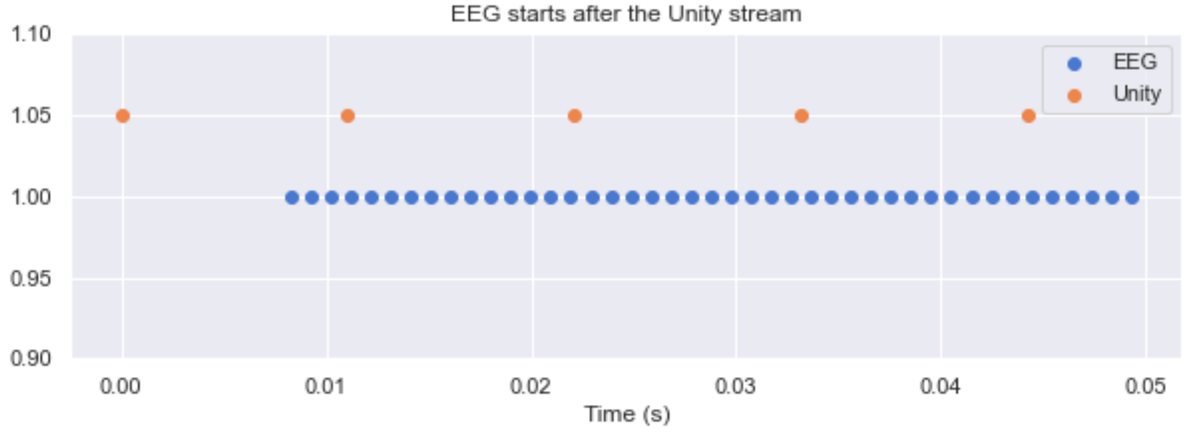




6. Results

How does the start of `long2` vs `long3` look like?





6. Results

How constant are the framerates?

EEG (1024 Hz or sample each ~ 0.98 ms)

	ftest1	ftest_build1	ftest_build2	ftest3	ftest2	ftest_build3	long2	long3
count	124959.000000	128031.000000	137247.000000	139807.000000	141855.000000	148511.000000	1846271.000000	1892831
mean	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977
std	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
min	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977
25%	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977
50%	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977
75%	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977
max	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977	0.000977

6. Results

How constant are the framerates?

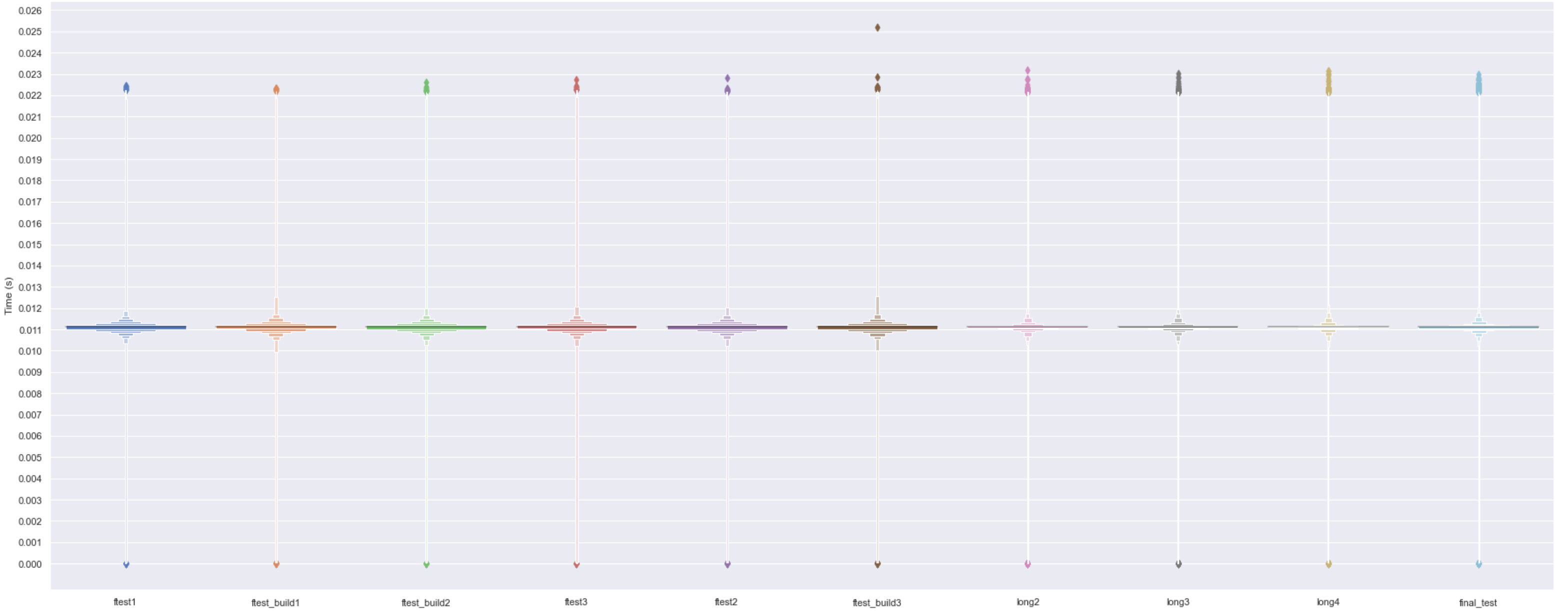
Unity (90 FPS or sample ~ 11.11 ms)

	ftest1	ftest_build1	ftest_build2	ftest3	ftest2	ftest_build3	long2	long3
count	10981.000000	11206.000000	12061.000000	12285.000000	12465.000000	13096.000000	162286.000000	166381.000000
mean	0.011110	0.011110	0.011110	0.011111	0.011111	0.011110	0.011110	0.011110
std	0.001141	0.001266	0.001182	0.001209	0.001135	0.001251	0.000844	0.000840
min	0.000023	0.000023	0.000023	0.000023	0.000023	0.000024	0.000022	0.000022
25%	0.011022	0.011026	0.011020	0.011023	0.011022	0.011019	0.011065	0.011065
50%	0.011109	0.011110	0.011111	0.011110	0.011108	0.011111	0.011106	0.011106
75%	0.011195	0.011194	0.011198	0.011194	0.011196	0.011202	0.011148	0.011148
max	0.022469	0.022343	0.022632	0.022717	0.022827	0.025182	0.023186	0.023030

6. Results

How constant are the framerates?

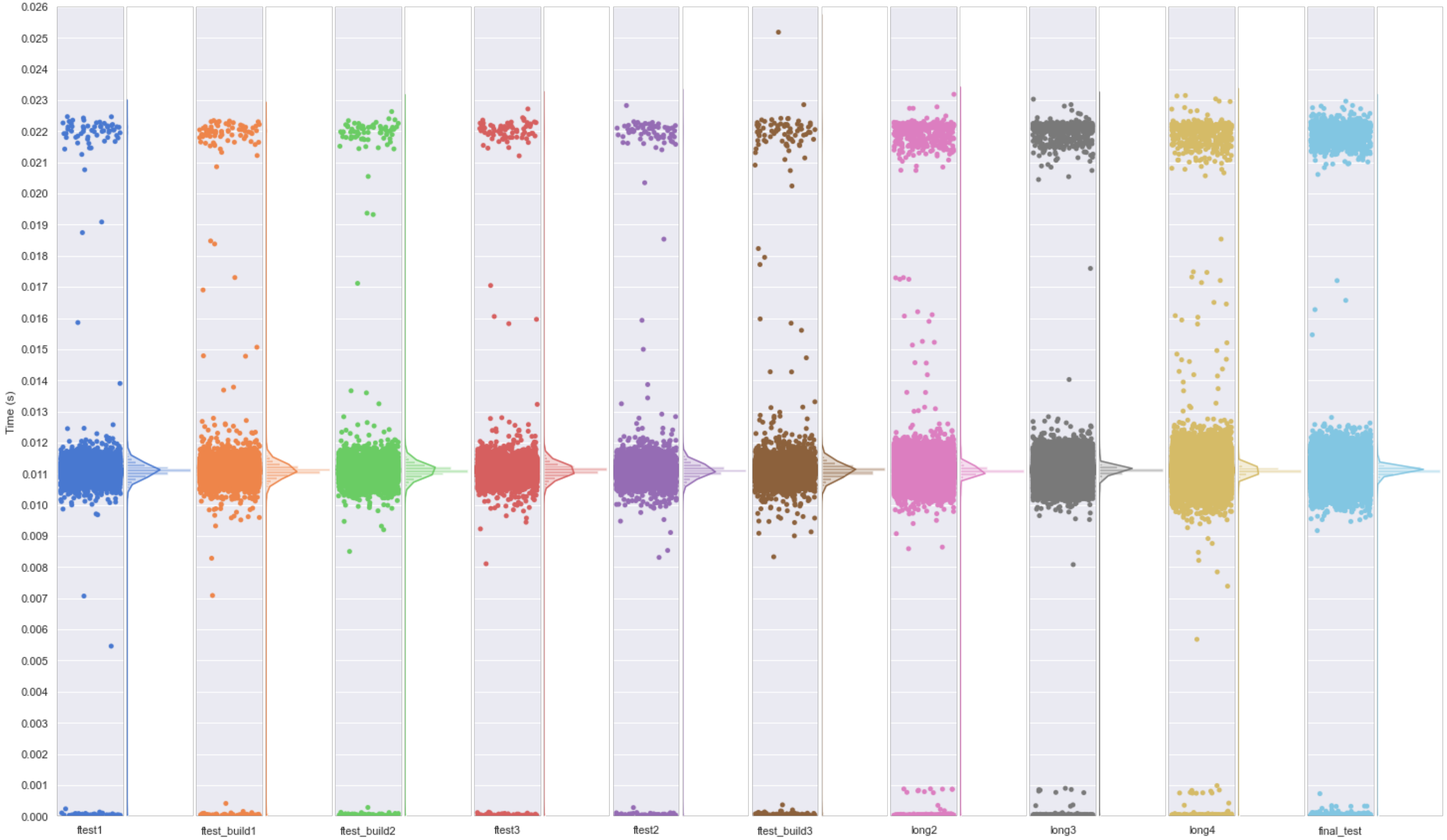
Unity (90 FPS or sample each 11.11 ms)



6. Results

How constant are the framerates?

Unity (90 FPS or sample ~ 11.11 ms)



6. Results

How constant are the framerates?

Unity (90 FPS or sample ~ 11.11 ms)

	N	N 0	% 0	N Middle	% Middle	N 2x	% 2x	N 2x -> 0	% 2x -> 0	N 0 -> 2x	% 0 -> 2x
ftest1	10981	62	0.56461	10843	98.74328	76	0.69210	38	0.34605	25	0.22767
ftest_build1	11206	89	0.79422	11007	98.22417	110	0.98162	53	0.47296	41	0.36588
ftest_build2	12061	78	0.64671	11890	98.58221	93	0.77108	53	0.43943	38	0.31507
ftest3	12285	84	0.68376	12099	98.48596	102	0.83028	54	0.43956	36	0.29304
ftest2	12465	76	0.60971	12286	98.56398	103	0.82631	55	0.44124	33	0.26474
ftest_build3	13096	103	0.78650	12859	98.19029	134	1.02321	76	0.58033	62	0.47343
long2	162286	468	0.28838	161206	99.33451	612	0.37711	295	0.18178	152	0.09366
long3	166381	464	0.27888	165306	99.35389	611	0.36723	312	0.18752	168	0.10097
long4	192663	631	0.32751	191124	99.20120	908	0.47129	533	0.27665	253	0.13132
final_test	355351	1191	0.33516	352669	99.24525	1491	0.41959	673	0.18939	420	0.11819

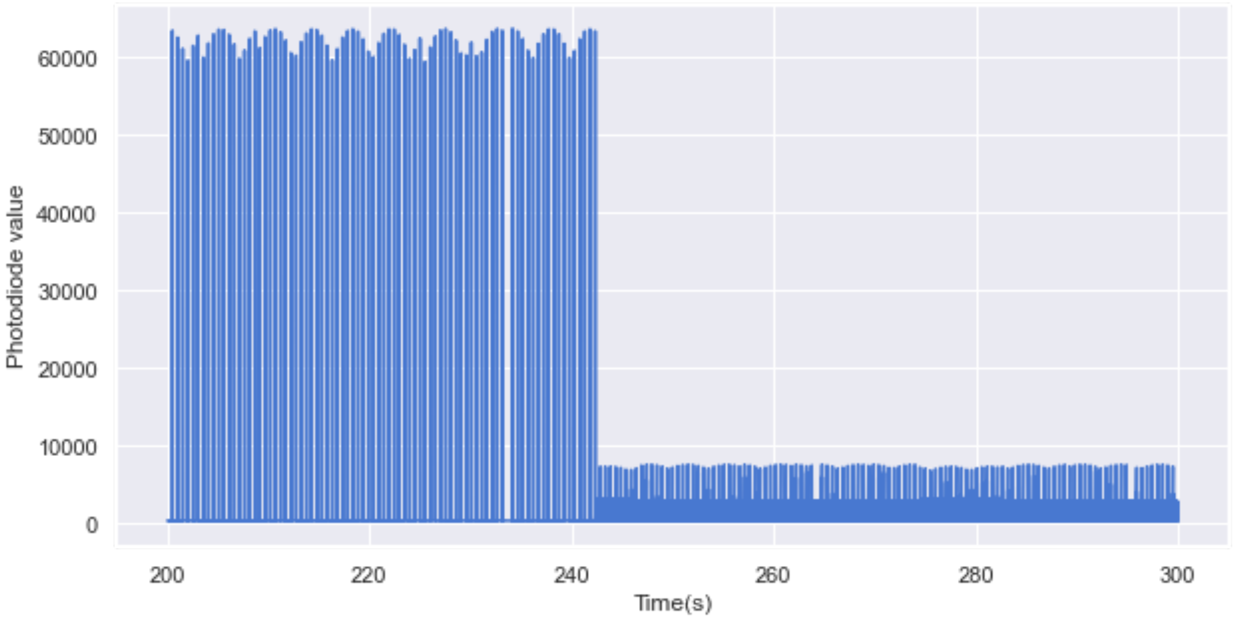
6. Results

Peak detections

	maximum	u_color	% u_color	e_color	% e_color	u_audio	% u_audio	e_audio	% e_audio
ftest1	244	239	97.95082	237	99.16318	239	97.95082	239	100.00000
ftest_build1	250	244	97.60000	242	99.18033	244	97.60000	244	100.00000
ftest_build2	268	262	97.76119	258	98.47328	262	97.76119	262	100.00000
ftest3	273	268	98.16850	264	98.50746	268	98.16850	267	99.62687
ftest2	277	271	97.83394	268	98.89299	271	97.83394	271	100.00000
ftest_build3	290	286	98.62069	282	98.60140	286	98.62069	284	99.30070
long2	3606	3532	97.94786	3520	99.66025	3532	97.94786	3532	100.00000
long3	3696	3621	97.97078	3604	99.53052	3621	97.97078	3621	100.00000
long4	4280	4195	98.01402	4172	99.45173	4195	98.01402	4195	100.00000
final_test	7895	7748	98.13806	7706	99.45792	7747	98.12540	7744	99.96128

6. Results

Peak detections



6. Results

Latencies

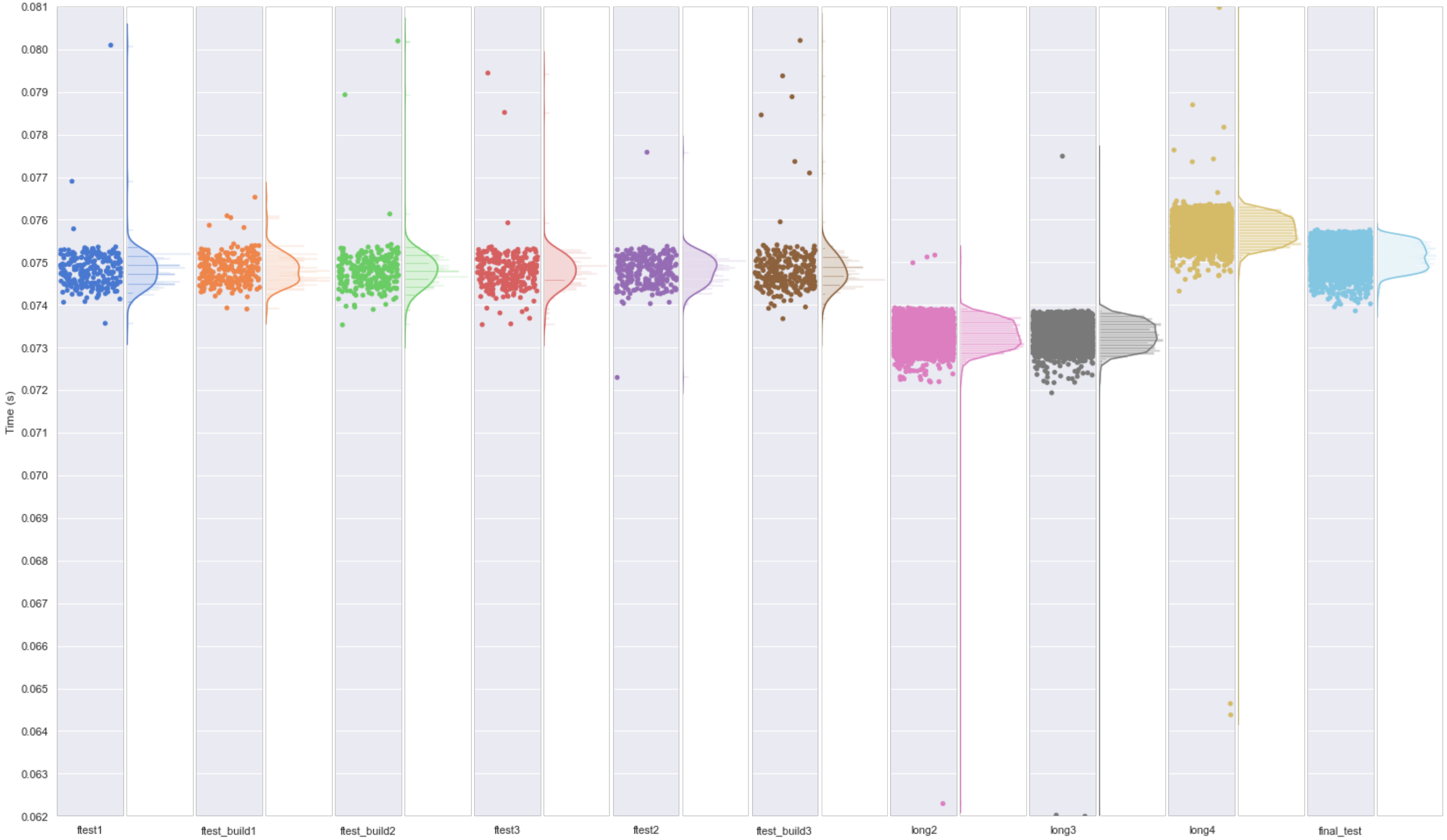
Time distance between unity color markers and the diode peaks

	ftest1	ftest_build1	ftest_build2	ftest3	ftest2	ftest_build3	long2	long3	long4	final_
count	237.000000	242.000000	258.000000	264.000000	268.000000	282.000000	3520.000000	3604.000000	4172.000000	7706
mean	0.075043	0.075053	0.075050	0.075026	0.075027	0.075070	0.073522	0.073504	0.075977	0.075
std	0.000495	0.000348	0.000543	0.000507	0.000383	0.000660	0.000365	0.000406	0.000406	0.000
min	0.073792	0.074127	0.073756	0.073762	0.072521	0.073900	0.062520	0.062223	0.064602	0.074
25%	0.074742	0.074797	0.074765	0.074752	0.074783	0.074765	0.073278	0.073266	0.075732	0.075
50%	0.075020	0.075041	0.075039	0.075005	0.075050	0.074964	0.073521	0.073511	0.075978	0.075
75%	0.075293	0.075276	0.075288	0.075263	0.075274	0.075275	0.073777	0.073758	0.076228	0.075
max	0.080321	0.076752	0.080420	0.079667	0.077808	0.080432	0.075394	0.077715	0.081206	0.075

6. Results

Latencies

Time distance between unity color markers and the diode peaks



6. Results

Latencies

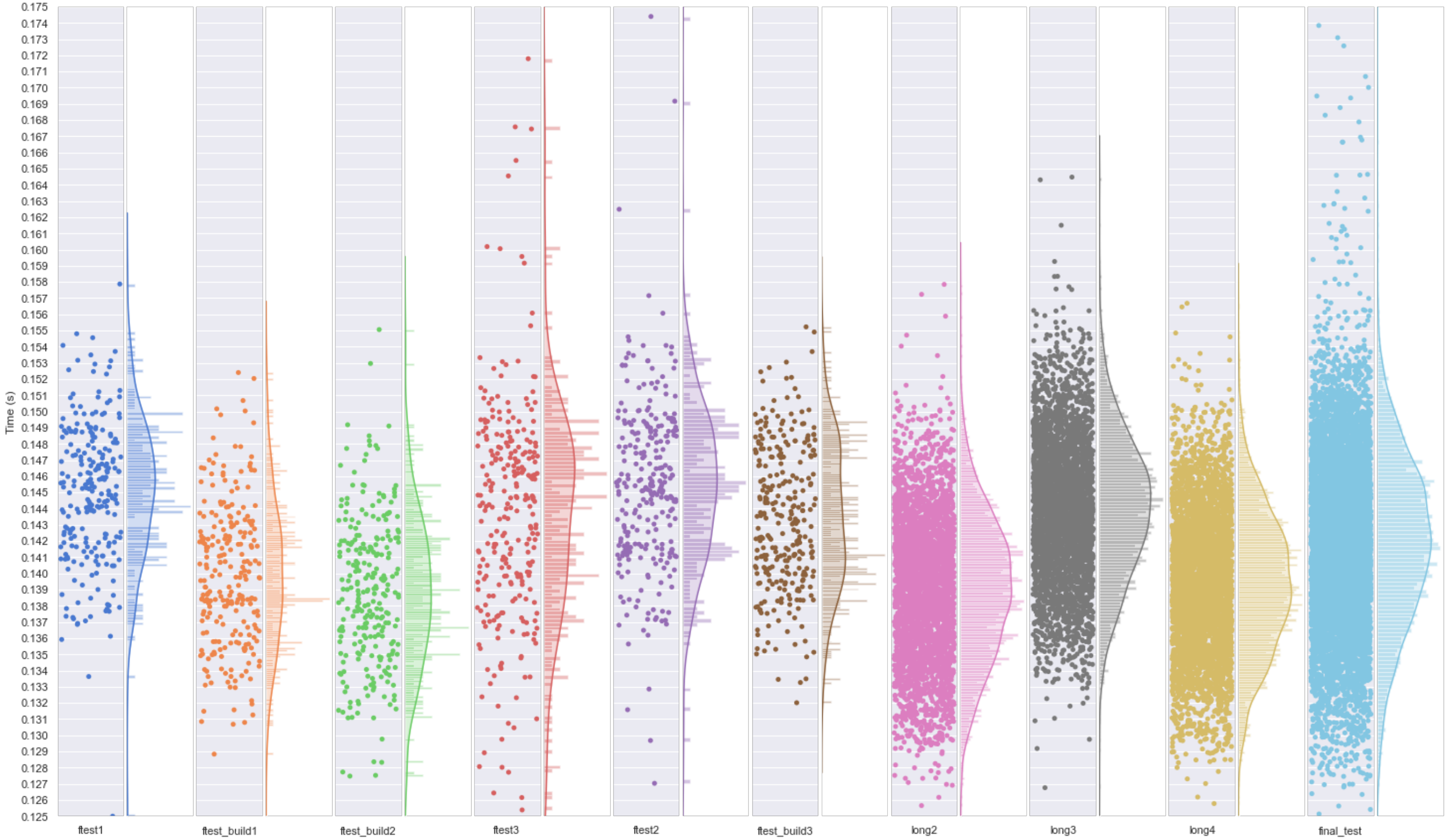
Time distance between unity audio playing markers and microphone peaks

	ftest1	ftest_build1	ftest_build2	ftest3	ftest2	ftest_build3	long2	long3	long4	final_
count	239.000000	244.000000	262.000000	267.000000	271.000000	284.000000	3532.000000	3621.000000	4195.000000	7744
mean	0.144945	0.139820	0.138772	0.143700	0.145448	0.143370	0.139008	0.143943	0.139353	0.141
std	0.004377	0.004409	0.004587	0.006956	0.005367	0.004463	0.004401	0.004385	0.004341	0.005
min	0.124942	0.128778	0.127420	0.125329	0.126975	0.131950	0.125599	0.126703	0.125724	0.125
25%	0.141982	0.136593	0.135626	0.139428	0.141689	0.140020	0.135901	0.140924	0.136362	0.138
50%	0.145270	0.139689	0.138773	0.144235	0.145351	0.143312	0.139070	0.144048	0.139267	0.141
75%	0.148013	0.142652	0.141839	0.147746	0.148775	0.147163	0.142038	0.146843	0.142362	0.145
max	0.157813	0.152345	0.154999	0.171731	0.174338	0.155167	0.157790	0.164417	0.156623	0.173

6. Results

Latencies

Time distance between unity audio playing markers and microphone peaks



7. Conclusions and remarks

- "Senseless" time differences between starts and endings of the data streams are not wrong recording indicators
- Dejittering is not taken into account on streams time information. Not a bug.
- EEG framerate is extremely constant
- Unity framerate is (even in the simplest Unity 2D project possible) not really constant (1% ~0ms or ~2xµms)
- Unity skips 1-2% of the triggers
- A better CPU on the recorder computer seems to contribute on a more precise aligning of streams
- No difference between: short/long recordings, different Unity builds, one/two computers
- The position of the diode on the HMDs affects the latency measurements (±1-2ms)
- Latency for video ~75ms (std of 0.4ms)
- Latency for audio ~132ms (std of 4ms)
- **Using LSL for recording Unity and EEG data looks very reliable**

8. Future outlook

- New recordings and data analysis using a "heavy" Unity project
- Try very short click sounds for a better estimation of audio latency

Resources

- Liblsl (https://github.com/sccn/liblsl/releases)
- LabRecorder (https://github.com/labstreaminglayer/App-LabRecorder/releases)
- LSL4Unity (https://github.com/labstreaminglayer/LSL4Unity)
- LSL Apps (https://github.com/sccn/labstreaminglayer/tree/master/Apps)
- LSL documentation reference (https://labstreaminglayer.readthedocs.io/)
- LSL latency analysis (https://github.com/mvidaldp/lsl_latency_analysis)
- Unity latency project (https://github.com/mvidaldp/black_n_white)
- Audio tones generator (https://github.com/mvidaldp/pytonegen)