Evaluation of Closed-loop Feedback System Delay
A Time-critical Perspective for Neurofeedback Training

Jonatan Tidare, Elaine Åstrand and Martin Ekström
School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden

Keywords: EEG, Closed-loop, LabVIEW, System Latency.

Abstract: Neurofeedback in real-time has proven effective when subjects learn to control a BCI. To facilitate learning, a closed-loop feedback system should provide neurofeedback with maximal accuracy and minimal delay. In this article, we propose a modular system for real-time neurofeedback experiments and evaluate its performance as a function of increased stress level applied to the system. The system shows stable behavior and decent performance when streaming with many EEG channels (36-72) and 500-5000 Hz, which is common in BCI setups. With very low data loads (1 channel, 500-1000 Hz) the performance dropped significantly and the system became highly unpredictable. We show that the system delays did not correlate linearly with the stress-level applied to the system, emphasizing the importance of system delay tests before conducting real-time BCI-experiments.

1 INTRODUCTION

Brain-Computer Interface (BCI) technology using signals from ElectroEncephaloGram (EEG) has emerged in research towards both assistive and rehabilitative technology to alleviate many neurological disorders caused by for example a stroke (Silvoni et al., 2011). After a stroke, the patient may be unable to move certain parts of the body, making physical rehabilitation impossible. As mental rehearsal of physical movements has been reported to enhance physical recovery in these patients, a system that is able to extract Mental Imagery (MI) and project the measure as real-time feedback so that the patient can train MI, holds great promise for promoting cortical reorganization and enhancing rehabilitation after stroke (Ruffino et al., 2017; Ang and Guan, 2017).

A common setup in such research experiments involves the patient sitting in front of a computer screen, on which instructions as to how and when to perform MI is presented visually and/or verbally. Based on recorded signals, feedback can be provided visually, through an assistive robot (e.g. Ang et al., 2015) and/or with Functional Electrical Stimulation (FES; Daly et al., 2009; Mohanty et al., 2017). The patient is instructed to use the feedback in order to upregulate the information-content of interest (e.g. MI) and hence promote mechanisms of neuroplasticity that is thought to lead to enhanced physical rehabilitation (Ang et al., 2010).

To maximize cortical plasticity, it is crucial that the system for acquiring and processing physiological signals and projecting feedback produces maximal accuracy as well as minimal "closed-loop delay" between the time of brain activation and that of the projected visual or tactile feedback (Gomez-Rodriguez et al., 2011). Specifically, such a delay is caused by several components of a system including stimulation software (delays for sending stimulation), online streaming protocols, signal processing and the temporal processing window length. As for the temporal processing window, there is a trade-off between the extraction accuracy, which is often shown to increase with longer temporal windows (Darvishi et al., 2013), and the ability of the patient to control the feedback, as delays longer than 250ms have been shown to significantly reduce the ability of the user to control a robot (Kim et al., 2005). In most BCI studies using oscillatory EEG signals, as the length of this time window is most certainly longer than 250ms, using overlapping temporal windows at a higher frequency might be a useful strategy to decrease the delay. However, this strategy will limit the influence that the most recent data will have on the real-time feedback.

To our knowledge, evaluating the closed-loop system delay caused by the stimulation software and the streaming protocol has not been done in BCI research. Reports may indicate the frequency at which feedback

Tidare, J., Åstrand, E. and Ekström, M.
Evaluation of Closed-loop Feedback System Delay - A Time-critical Perspective for Neurofeedback Training.
DOI: 10.5220/0006598301870193
In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2018) - Volume 1: BIODEVICES*, pages 187-193
ISBN: 978-989-758-277-6

is presented and the hardware or software that was used, but the actual system delay is often not mentioned. All systems are bound to have a delay but minimizing it holds strong promise in increasing the effects of neurofeedback training. Moreover, reporting the system delay will allow for a better interpretation of the training effects with respect to other studies.

This study focuses on the closed-loop system delay caused by both the stimulation software and the streaming protocol as compared to only the streaming protocol. These system delays are evaluated under different stress-levels in order to characterize their performance. An increasingly popular game-engine was chosen as the stimulation software to test its usability in BCI research. EEG data was streamed in real time to an external software for aligning and signal processing.

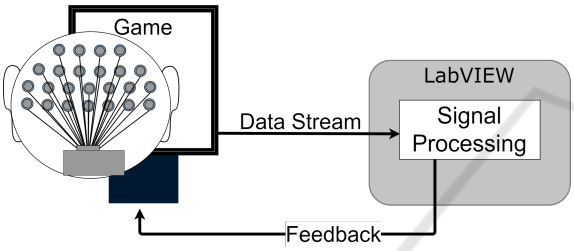

Figure 1: Overview of the future experimental setup. A subject is placed in front of a computer screen. EEG data is recorded simultaneously as the subject is performing a computer task. EEG data is continuously streamed in real time to LabVIEW for signal processing. Feedback from the recorded EEG data (visual and/or tactile) is then provided back to the subject.

2 METHOD

The ”Real-time Neurofeedback Brain-computer interface” (RNB) is composed of the following elements: a computer screen, a photodiode that is connected to the screen, Unity game engine (version 4.6), EEG acquisition device, and a LabVIEW client. The idea for future experiments is that EEG signals will be recorded from test-subjects while they are engaged in a computer task (run by Unity). Data will be streamed continuously to LabVIEW for alignment, signal processing and analysis in order to determine and send feedback to be visualized on the computer screen (i.e. closed-loop system). A general overview of the future experimental setup can be visualized in Figure 1.

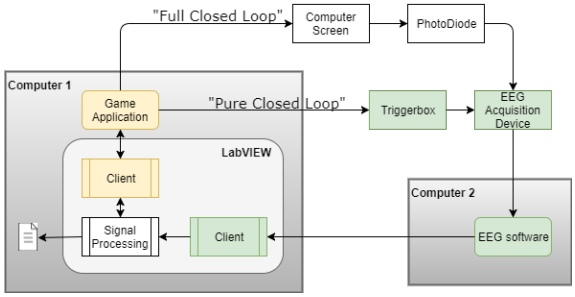

Figure 2: System overview. The RNB comprises 2 computers: computer 1 on which the game engine (yellow) and LabVIEW (light grey area) are running, and computer 2 on which the EEG acquisition software (green) is running. The game engine communicates with the EEG device through the triggerbox and with LabVIEW through a specific LabVIEW client. The EEG device communicates with the computer screen through a photodiode and with LabVIEW through a specific LabVIEW client. LabVIEW contains threads for logging, data signal processing and inter-software communication (clients)).

2.1 Hardware

Computer 1 has Windows 7, 64-bit, service pack 1 with an Intel Core i7-6800K 3.40 GHz and 32 GB RAM. As this is the most powerful computer, we had the Unity game engine and LabVIEW running on it (Figure 2). Computer 2 has Windows 7, 64-bit, service pack 1 with and Intel(R) Xeon(R) E5620 2.40 GHz. Only the EEG software was running on computer 2 (Figure 2). Both computers were running on Windows 7 with a computer screen refresh rate at 60 Hz.

2.2 Unity Computer Task

For simplicity in measuring the closed-loop system delay, a grey square was used in the computer task as a start- and stop-event of the closed loop. The task started by the presentation of a grey square in the upper right corner of the screen. As soon as the Signal Processing thread in LabVIEW (Figure 2) detected the event, a command was sent to Unity in order to replace the grey square with a black square (simulated feedback). No signal processing was performed in order to measure the ”pure” delay of the closed-loop system.

Unity communicates with the EEG acquisition device by two paths (Figure 2): 1) by sending a message from the parallel port of computer 1 to the triggerbox, which is an extension of the trigger input port of the EEG acquisition device, or 2) by changing the luminosity of a set of pixels in the upper right corner of the computer screen, on which a photodiode is con-

nected that is fully integrated with the EEG acquisition device (no issue of time synchronization). The first communication path was used to send a general identity of the visual stimulus that was presented on the computer screen (i.e. task event) to be integrated in the EEG-data. The digital port of the EEG acquisition device requires trigger pulses according to TTL specifications. This is a standard interface in EEG research for trigger co-registration. The second communication path provided an accurate timestamp of when the task event occurred. Unity also communicates directly with LabVIEW by sending a specific identity of the task event.

2.3 EEG Acquisition

The EEG equipment consists of 64 Ag/AgCl active electrodes (ActiCHamp, Brain Products), a photodiode (Brain Products) connected to the auxiliary (AUX) port of the amplifier, an additional of 7 AUX ports for recording EMG, and a triggerbox that enables input markers from different sources to the EEG with high temporal precision ($<$ 1 ms, (Triggerbox, Brain Products)). The open source software PyCorder, was used to acquire the EEG signals and the PyCorder RDA-client streamed data and markers to LabVIEW. Different channel configurations and sampling rates are available in the PyCorder software and we used 1, 36 or 72 active channels each with different sampling rates: 500 Hz, 1 kHz, 5 kHz, 10 kHz, 25 kHz, 50 kHz and 100 kHz. However, due to technical limitations in the EEG acquisition device 72 channels was used with only up to 50kHz sampling rate.

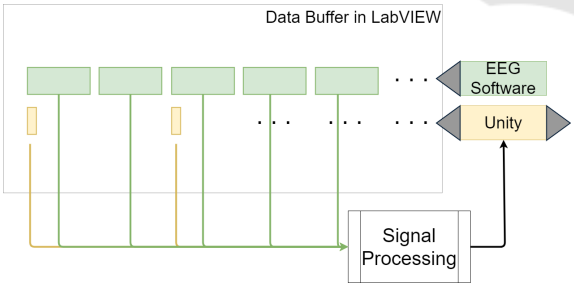

Figure 3: Streaming protocols. EEG data is streamed from the EEG software to LabVIEW in buffer times of 20 or 50 ms. The game engine streams events directly to LabVIEW. Data is immediately sent to the signal processing thread in LabVIEW and feedback is sent back to the game engine.

2.4 Data Streaming Protocol

The RDA-client integrated in PyCorder uses TCP/IP as a communication protocol to stream EEG-data and event markers from the PyCorder to an external software (i.e. LabVIEW client, figure 2). The PyCorder program was set to buffer data during either 20ms or 50 ms before sending it to the RDA-client, leading to an additional varying stream-latency of up to 20 or 50 ms (here referred to ”EEG packet wait time”, figure 3). Unity uses JSON as a communication protocol to send task event data to the LabVIEW client (Figure 3).

2.5 LabVIEW

The signal-processing program is built in LabVIEW and has 3 threads: 1 thread for data logging and real time signal processing of the data, and 2 threads (named ”clients”, Figure 2) for inter-software communication with the Unity game engine and the EEG software. The Signal Processing thread performs no signal processing in these tests, except for detecting the grey square in the photosensor data (AUX). The client between the Signal Processing thread and the EEG software only forwards stream data to the Signal Processing thread. The Client between the Signal Processing thread and Unity forwards the two communication paths (parallel port or visual stimulus) to Unity and can send event identities back to the Signal Processing thread from Unity.

2.6 Closed-loop System Delay

The closed-loop system delay including only the streaming protocol will be referred to as the ”pure closed-loop system delay” (Figure 2). This system delay was measured without taking EEG packet wait time into consideration (Figure 4) therefore reflecting mainly system delays caused by data streaming. The pure closed-loop system delay was calculated by measuring the time between two consecutive task events (start and stop, see section on Unity computer task) as detected from EEG-data trigger markers in LabVIEW and subtracting EEG data packet wait time.

The closed-loop system delay including both the stimulation software (i.e. Unity game engine) and the streaming protocol will be referred to as the ”full closed-loop system delay” (Figure 2). This system delay is measured as the time between two consecutive task events as detected from the photodiode data in LabVIEW (Figure 2). In this measurement, the EEG data packet wait time was included in order to assess the full closed-loop system delay similar to a real-time experiment. A total of 50 repetitions were made for each closed-loop system delay measurement.

3 RESULTS

In order to characterize the use of Unity as a visual stimulation software in BCI research, the delay between the time of the task event trigger marker in the EEG-data and the actual time of presentation on the screen was measured. We observed a median delay across repetitions (n=100) of 36ms with an associated median absolute deviation of 4.8ms (Figure 4).

To further investigate the RNB without taking the visual stimulation software into consideration, the pure closed-loop system delay was measured (Figure 5A, 6A) using a PyCorder buffer size of 50ms. Several different channel and sampling rate configurations was used in order to create different degree of stress to the RNB. The lowest system delay was found when streaming with either low sampling rates (500 Hz, 1kHz, 5kHz) using high number of channels (36 or 72) or high sampling rate (10kHz, 25kHz, 50kHz, 100kHz) using only one channel (Figure 5A). In these cases, the median pure closed-loop system delay was below 25 ms and the maximum pure closed-loop system delay was below 30 ms (Figure 5A, 6A, 36 and 72 channels, 500Hz, 1kHz, and 5kHz: median 20.0 ms, 1 channel, 10kHz, 25kHz, 50kHz, and 100kHz: median 21.7 ms). Using 36 or 72 channels, the median system delay increases significantly when the sampling rate increases from 25kHz to 100 kHz for 36 channels and 10kHz to 50kHz for 72 channels (Figure 5A, 36 channels: 30.6 ms, $p<0.001$; 72 channels: 26.9 ms, $p<0.001$, Wilcoxon test).

The system delay is 51.3 ms longer at low sampling rates when streaming with only one channel (1kHz: 71.5 ms, 10kHz: 20.2 ms; $p<0.001$, Wilcoxon test) and the maximum system delay is drastically increased (Figure 6A, 1kHz: 220.0156ms, 10kHz: 22.4019ms). By increasing the sampling rate for one channel to 10 kHz or 25 kHz the median delay decreases to below 25 ms with a slight increase for 25kHz sampling rate.

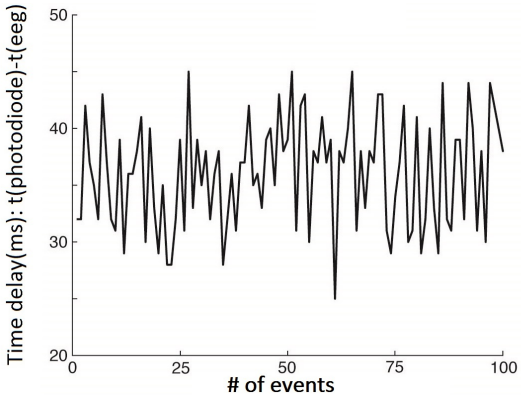

Figure 4: Visual presentation delay. The delay between the timestamp of the EEG input trigger (the game engine sends a trigger to the EEG via a parallel port) and the presentation of a visual stimulus on the screen, as detected by the photodiode, is shown for all measurements (n=100).

The full closed-loop system delay was evaluated with identical channels, sampling rate, and buffer size configurations as for the pure closed-loop system delay in order to enable a fair comparison when also taking the stimulation software, the time for presenting a visual task event on the computer screen, and EEG packet wait time into account (Figure 5B, 6B). Streaming with 36 and 72 channels produces similar system delays for each sampling rate (Figure 5A, 36 channels: median = 101.5 ms, 72 channels: median = 102.0 ms. $p>0.05$, Wilcoxon test). A minimum system delay of 90ms is measured at best (36 channels and 5 kHz sampling rate) and the maximum system delay reaches 131ms (36 channels and 100 kHz sampling rate).

Using 36 channels, the median system delay increases significantly when the sampling rate increases from 25kHz to 100kHz (Figure 5B, increase: median 39.0 ms, $p<0.001$, Wilcoxon test). As for the pure closed-loop system delay (Figure 5A), streaming with 1 channel leads to lower full closed-loop system delays when the sampling rate increases, specifically 61.8 ms of difference from using 500Hz to 10kHz sampling rate (Figure 5B). A worst-case system delay is measured at 294.0 ms (Figure 6B, 1 channel and 500Hz sampling rate).

Using 36 channels and a sampling rate of 1kHz (a common setup in BCI research), the system delay increases by an average of 74.0 ms when the visual stimulation software, the time for presenting a visual task event on the computer screen, and the EEG packet wait time is taken into consideration (full closed-loop system delay).

As an additional comparative evaluation, the PyCorder buffer size was decreased to 20ms and the full closed-loop system delay using identical channel and sampling rate configurations was measured (Figure 7). The full closed-loop system delay with 50 ms buffer time was similar that of using 20 ms buffer time when using 1 channel with 500Hz or 1kHz sampling rate (Figure 7, $p>0.05$ for both sampling rates, Wilcoxon test). The median difference becomes negative when sampling rate increases (i.e. system delay with 20 ms buffer time is shorter than that of 50 ms buffer time).

Using 36 and 72 channels, the system delay is on average 18.0 ms shorter for sampling rates up to 50kHz when the buffer size is set to 20ms (Figure 7, $p<0.1$, Wilcoxon test). A dramatic decrease of 258.0 ms (424.0 ms maximum delay) in the system delay

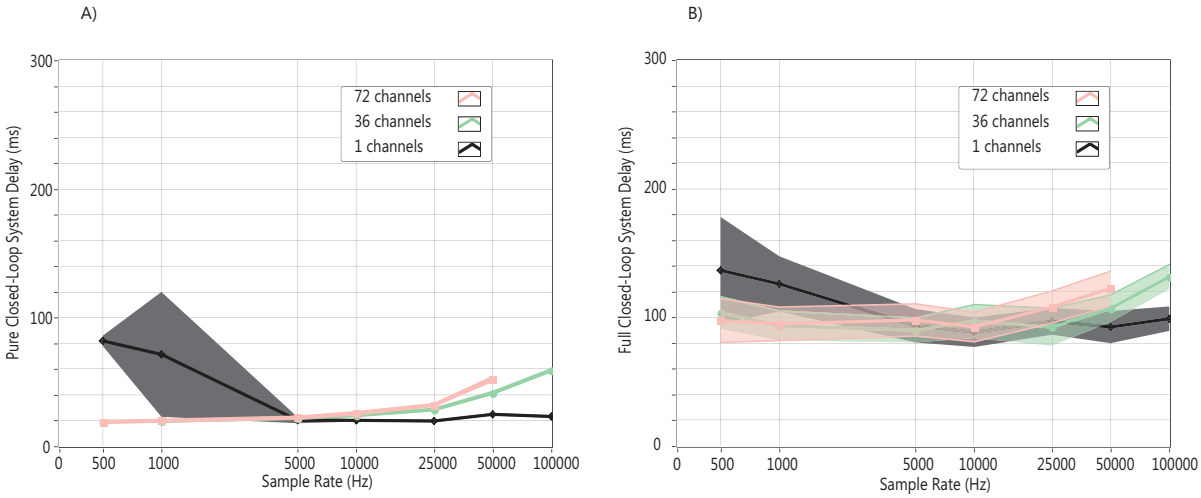

Figure 5: (A) Pure closed-loop system delay. Median system delay with associated median absolute deviations across measurements (n=50) are shown as a function of sampling rate (Hz) for streaming with 1 channel (black), 36 channels (green), and 72 channels (pink). (B) Full closed-loop system delay. Same as in (A).

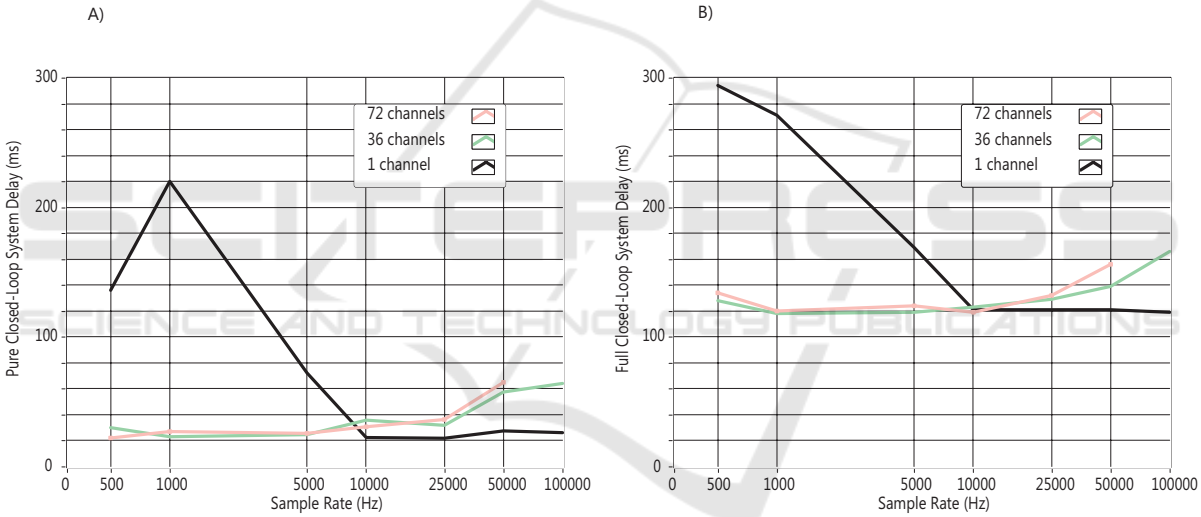

Figure 6: (A) Maximum pure closed-loop system delays across measurements (n=50) are shown as a function of sampling rate (Hz) for streaming with 1 channel (black), 36 channels (green), and 72 channels (pink). (B) Maximum full closed-loop system delay. Same as in (A).

when the buffer size it set to 20 ms can be observed for 36 channels using a sampling rate of 100kHz (Figure 7).

4 DISCUSSION

The RNB measures EEG data which is streamed in real time to an external software for further processing. The use of an external software facilitates the fusion of other physiological measurements. The RNB includes a game engine in order to enable more dynamic and real-world tasks and feedback.

The system delays and the implications of uncertain and large system delays in BCI research are rarely reported and discussed. We show that the system delays, both pure- and full closed-loop system delay, did not correlate in a linear fashion with the applied stress-level. Specifically, for both system delays, streaming data from only one channel with low sampling rates below 5kHz, causes a substantial temporal lag and uncertainty in the system delay (Figure 5A, 5B). The poor performance of the RNB at this configuration is surprising. By investigating the behavior of the RNB more closely, system delays were observed in discrete levels which was caused by the PyCorder software that appears to buffer up to 4 data packets

before sending them to LabVIEW. This unexpected behavior is highly undesired when presenting feedback in real time. Also in order to minimize system load due to the streaming and processing of a large amount of data, it is common to reduce the number of EEG channels to only the ones that carry relevant information. As this study shows, this procedure can cause devastating consequences for the overall system delay.

Streaming a moderate amount of EEG data, with 36 or 72 channels using a sampling rate of below 100 kHz, led to reduced system delays for the pure- and full closed-loop with approximately 20 ms and 100 ms, respectively. The variability was also at its lowest using these configurations. These results are promising because most BCI studies would probably be included in this configuration. It is however important to carry out system performance tests before a BCI experiment in order to evaluate the system delay.

When a high stress-level was applied to the RNB, (36 channels with samplingrate >25 kHz or 72 channels with samplingrate > 10 kHz), a moderate increase in the system delay could be observed for both the pure- and the full closed-loop system delay. As the increase in system delays occurs at a higher samplingrate using 36 channels as compared to streaming 72 channels, it seems like the system delay is influenced by the total amount of data in a threshold-like manner. These results combined show the general trend of the system delays depending on the stress-level using the RNB. Although the results presented in this study

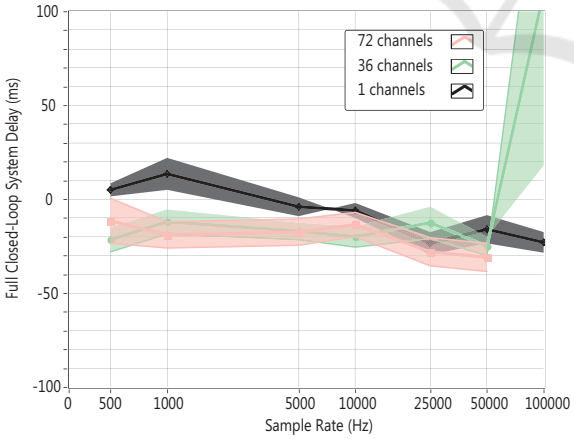

Figure 7: Effects on the full closed-loop system delay by using a PyCorder buffer time of 20 ms. The full closed-loop system delay using a buffer time of 50ms is subtracted from the full closed-loop system delay using a buffer time of 20ms. The median difference is shown with associated median absolute deviations across measurements (n=50) as a function of different sampling rates (Hz) for streaming with 1 channel (black), 36 channels (green), and 72 channels (pink).

are specific to the RNB system used here, it emphasized the vulnerability of a system that depends on the amount of data that is streamed. It calls for immediate attention concerning the time delay in a system and encourages researchers to evaluate their system delay. For future research, it will be important to investigate the impact of delayed feedback on learning.

When comparing the pure and full closed-loop system delays, similar delays can be observed for each system delay relative its different configurations (Figure 5A, 5B). However, an offset of approximately 80ms distinguishes the two system delays. This offset is partly due to a delay between Unity and visual screen presentation (Figure 4) which depends on several factors including Unity, OS-scheduling and screen resolution. However, the offset is also due to the RDA-client buffer time in the PyCorder software which was initially set to its default value of 50 ms. Using this default PyCorder buffer time configuration has resulted in a variable streaming delay of 50- 100 ms through the EEG signal pipeline (hardware filters - a/d converters - USB communication - windows OS - RDA server - Ethernet Connection (TCP/IP) - RDA Client) (BrainProducts, 2015). Reducing the buffer time to 20 ms decreased the full closed-loop system delay with approximately 10-20 ms. Theoretically, it is possible to further reduce the buffer time but practically, as this requires a very powerful computer with reduced cycles of scheduling, it was not a viable option for our system. Replacing the RDA client with a Software Development Kit (SDK) to directly access the ActiCHamp hardware, the system delay can be significantly reduced (BrainProducts, 2015).

An important issue in BCI research is how the temporal delay of neurofeedback influences the ability of subjects to understand and extract relevant information from the feedback. In a typical neurofeedback experiment, subjects receive discrete or continuous feedback based on recorded brain signals (Figure 1). One behaviorist theory in reinforcement learning states that feedback must be given immediately in order to reinforce correct behavior (e.g. Skinner, 1954). In line with this theory, previous human studies have demonstrated a clear behavioral benefit for having immediate feedback as compared to temporally delayed feedback (Lieberman et al., 2008; Opitz et al., 2011). The behavioral gain is particularly important when there is no prior knowledge of correct behavior (Lieberman et al., 2008) which is typically the case in neurofeedback training. Specifically, Lieberman and colleagues show that in a motor learning task during which participants had minimal information on which movement was correct, immediate feedback was required for the participants to learn (Lieberman et al.,

2008). In addition, they observed that the effect of the temporal delay of feedback depended on the amount of information that had to be held in working memory (Lieberman et al., 2008). Despite these observations pointing towards the importance of minimizing the temporal delay between brain-activation (and recording) and projection of feedback, few studies using neurofeedback address this issue. The present study demonstrates a highly variable closed-loop system delay depending on the system configuration. These results show the importance of measuring and reporting the system delay in order to correctly interpret the behavioral effects of neurofeedback training.

5 CONCLUSION

This study shows the importance of testing the system delay with the final experimental setup before conducting a real-time BCI experiment. We specifically observe the lowest system delays when streaming a moderate amount of data through the RNB. A small amount of data may cause substantially larger system delays due to inbuilt data aggregation of the software.

REFERENCES

Ang, K. K., Chua, K. S. G., Phua, K. S., Wang, C., Chin, Z. Y., Kuah, C. W. K., Low, W., and Guan, C. (2015). A randomized controlled trial of eeg-based motor imagery brain-computer interface robotic rehabilitation for stroke. *Clinical EEG and neuroscience*, 46(4):310–320.

Ang, K. K. and Guan, C. (2017). Eeg-based strategies to detect motor imagery for control and rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(4):392–401.

Ang, K. K., Guan, C., Chua, K. S. G., Ang, B. T., Kuah, C., Wang, C., Phua, K. S., Chin, Z. Y., and Zhang, H. (2010). Clinical study of neurorehabilitation in stroke using eeg-based motor imagery brain-computer interface with robotic feedback. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 5549–5552. IEEE.

BrainProducts (2015). Getting eeg data in real-time for bci, neurofeedback and more. *[online] Available at: http://pressrelease.brainproducts.com/real-time-eeg/ [Accessed 4 sep. 2017]*.

Daly, J. J., Cheng, R., Rogers, J., Litinas, K., Hrovat, K., and Dohring, M. (2009). Feasibility of a new application of noninvasive brain computer interface (bci): a case study of training for recovery of volitional motor control after stroke. *Journal of Neurologic Physical Therapy*, 33(4):203–211.

Darvishi, S., Ridding, M. C., Abbott, D., and Baumert, M. (2013). Investigation of the trade-off between time window length, classifier update rate and classification accuracy for restorative brain-computer interfaces. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 1567–1570. IEEE.

Gomez-Rodriguez, M., Grosse-Wentrup, M., Hill, J., Gharabaghi, A., Schölkopf, B., and Peters, J. (2011). Towards brain-robot interfaces in stroke rehabilitation. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–6. IEEE.

Kim, T., Zimmerman, P., Wade, M., and Weiss, C. (2005). The effect of delayed visual feedback on telerobotic surgery. *Surgical Endoscopy and Other Interventional Techniques*, 19(5):683–686.

Lieberman, D. A., Vogel, A. C. M., and Nisbet, J. (2008). Why do the effects of delaying reinforcement in animals and delaying feedback in humans differ? a working-memory analysis. *The Quarterly Journal of Experimental Psychology*, 61(2):194–202.

Mohanty, R., Sinha, A., Remsik, A., Allen, J., Nair, V., Caldera, K., Sattin, J., Edwards, D., Williams, J. C., and Prabhakaran, V. (2017). Machine learning-based prediction of changes in behavioral outcomes using functional connectivity and clinical measures in brain-computer interface stroke rehabilitation. In *International Conference on Augmented Cognition*, pages 543–557. Springer.

Opitz, B., Ferdinand, N. K., and Mecklinger, A. (2011). Timing matters: the impact of immediate and delayed feedback on artificial language learning. *Frontiers in human neuroscience*, 5.

Ruffino, C., Papaxanthis, C., and Lebon, F. (2017). Neural plasticity during motor learning with motor imagery practice: Review and perspectives. *Neuroscience*, 341:61–78.

Silvoni, S., Ramos-Murguialday, A., Cavinato, M., Volpato, C., Cisotto, G., Turolla, A., Piccione, F., and Birbaumer, N. (2011). Brain-computer interface in stroke: a review of progress. *Clinical EEG and Neuroscience*, 42(4):245–252.

Skinner, B. F. (1954). The science of learning and the art of teaching. *Cambridge, Mass, USA*, pages 99–113.