Thalamus: A User Simulation Toolkit for Prototyping Multimodal Sensing Studies

Kayhan Latifzadeh
University of Luxembourg
Luxembourg, Luxembourg
kayhan.latifzadeh@uni.lu

Luis A. Leiva
University of Luxembourg
Luxembourg, Luxembourg
luis.leiva@uni.lu

Abstract

Conducting user studies that involve physiological and behavioral measurements is very time-consuming and expensive, as it not only involves a careful experiment design, device calibration, etc. but also a careful software testing. We propose Thalamus, a software toolkit for collecting and simulating multimodal signals that can help the experimenters to prepare in advance for unexpected situations before reaching out to the actual study participants and even before having to install or purchase a specific device. Among other features, Thalamus allows the experimenter to modify, synchronize, and broadcast physiological signals (as coming from various data streams) from different devices simultaneously and not necessarily located in the same place. Thalamus is cross-platform, cross-device, and simple to use, making it thus a valuable asset for HCI research.

CCS Concepts

• **Human-centered computing → Laboratory experiments**; *Interaction devices*; HCI design and evaluation methods.

Keywords

User Simulation; Behavioral/Physiological Sensing

ACM Reference Format:

Kayhan Latifzadeh and Luis A. Leiva. 2025. Thalamus: A User Simulation Toolkit for Prototyping Multimodal Sensing Studies . In *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization (UMAP Adjunct '25), June 16–19, 2025, New York City, NY, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3708319.3733687

1 Introduction and Related Work

In HCI research and experimentation, gathering multimodal information about the user can be crucial for comprehending the complex and dynamic nature of user interaction [8]. By simultaneously gathering different sensory data, researchers can capture a more complete and nuanced picture of user behavior, and investigate how various modalities interact and influence each other. For instance, monitoring eye movements together with facial expressions and brain activity will provide valuable information about how users interpret and react to visual stimuli, and can reveal patterns and trends that might not be apparent when focusing on just one modality alone [17]. As it enables researchers to comprehend how people use a system [2, 4, 12], simultaneously gathering multimodal information may help design and evaluate novel technologies and interfaces.

On the other hand, collecting multimodal signals is challenging, particularly when recording happens in real-time [19]. It is essential for researchers to ensure that everything works correctly before conducting the main experiment, otherwise they may incur in unforeseen costs derived e.g. from lab setup, participant payment, data post-processing, etc. A beneficial solution to address these issues is simulating the experiment before actually running it. This way, researchers can ensure that the equipment and methods used for data collection are capable of handling the data streams they want to consider, that the data is of good quality, and that the system being tested can handle corner cases such as device disconnections, missing or noisy data being recorded, etc.

Furthermore, data simulation can be used to test and optimize multimodal signal processing methods, evaluate the effectiveness of data analysis, identify potential sources of error, and, in short, save valuable time and resources by identifying and resolving potential problems early on. In a nutshell, simulation is a useful and viable method for research in HCI, as it can be utilized in multiple ways to examine human behavior with technology [10], before actual construction through prototyping [14], offering training in a secure and controlled setting for individuals to utilize new technology.

To the best of our knowledge, there is currently no widely used software that is specifically designed for simulating different data streams in HCI experiments. When it comes to recording multiple signal types simultaneously, many researchers have had to rely on ad-hoc solutions [1, 5, 11, 13, 16, 20–24, 26], which can be time-consuming and difficult to replicate [6]. Moreover, previous work has proposed solutions for collecting multimodal signals, not for simulating them. While ad-hoc solutions can solve a specific problem easily, they are not as generalizable as a dedicated software toolkit, and they may not provide the same level of consistency and repeatability across different experiment setups.

To address this gap, we propose Thalamus, a toolkit for capturing and simulating various data streams, without the need to recruit users at the early stages of design, and even without the need to install or purchase a specific device. Among other features, Thalamus allows the experimenter to modify, synchronize, and broadcast physiological signals (as coming from various data streams) from different devices simultaneously and not necessarily located in the same place. Thalamus is cross-platform, cross-device, and simple to use, making it thus a valuable asset for HCI research.

UMAP Adjunct '25, New York City, NY, USA

ACM ISBN 979-8-4007-1399-6/2025/06
https://doi.org/10.1145/3708319.3733687

2 The Thalamus toolkit

Our proposed toolkit, depicted in Figure 1, consists of three key components: a central hub for signal input/output, called *Thalamus Core*, a number of *Recording devices* that can be either simulated or real, and a number of *Clients* that can receive one or more processed or raw sensory signals. Taken together, these components demonstrate how the toolkit can be utilized to simulate and coordinate a variety of data streams, including diverse types of recording devices. Thalamus is written in Python and will be publicly available upon the publication of this paper.

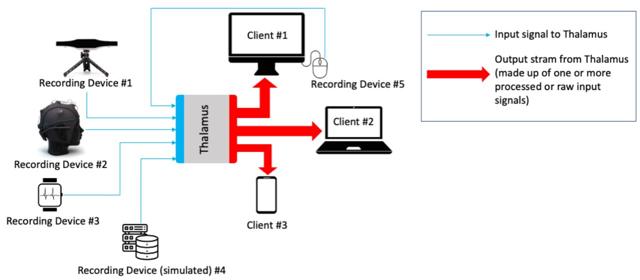

Figure 1: Conceptual diagram of the Thalamus toolkit, illustrating how it can be used to simulate different types of recording devices and synchronize various data streams, including e.g. brain signals and eye movements. Thalamus can receive feedback from any device. For example, here client #1 can also act as a recording device (Recording Device #5), thereby enabling real-time multimodal data collection and analysis.

2.1 Thalamus Core

The central component of our proposed toolkit is named after the Thalamus, a part of the brain that acts as a hub for processing and relaying sensory information to other areas. Conveniently, Thalamus Core acts as the hub for coordinating and synchronizing data streams, both from real and simulated signal recording devices. It is responsible for receiving data streams from various sources, processing and organizing it, and then streaming them to the clients for computation and analysis. This component is implemented using socket programming over the TCP/IP communication protocol, which enables seamless communication and data transfer between the Thalamus, clients, and devices, ensuring efficient and real-time data stream coordination and synchronization.

2.2 Recording devices

In our proposed toolkit, recording devices can be real (i.e. physical devices that capture and record data) or simulated (i.e. virtual devices that mimic the functionality of the physical recording devices). Simulated devices are 'injected' from public datasets through a controller that parses the data and formats it to the expected transport format, based on JSON. We suggest JSON for data communication in Thalamus since it is a popular format that offers key advantages, including its self-describing nature, data types, cross-platform compatibility, and compact size. Any device, whether virtual or physical, is required to supply a coordinated universal time (UTC) timestamp for each recorded sample. This is crucial for coordinating various signals and extracting events. The use of UTC timestamps guarantees that data from various devices is collected and processed in a uniform and standardized manner, making it simple to compare and analyze.

Integrating real-world devices into our proposed software toolkit entails two main challenges. First, some devices may not provide UTC timestamps, which is crucial for extracting events and synchronizing different signals. Second, some devices may not provide easy access to the data stream, either because they are proprietary or they rely on a non-documented exchange format. To solve these challenges, the experimenter can write controllers for each device that can format the data accordingly. For example, many devices write data to a file, and by creating a custom controller, it is possible to access and read the data sample by sample and reformat it. Then, the reformatted data can then be sent to the Thalamus Core component for further processing.

2.3 Clients

Any client that can open a socket connection can communicate with Thalamus, regardless of the operating system or programming languages used. This feature makes it easy for many devices to be considered in an experiment, ranging e.g. from computers, smartphones, and IoT devices. It also allows for greater flexibility and adaptability to the researcher's needs, as they can select the devices that are most convenient for their specific study.

When a device is connected to Thalamus, the Core component will provide the device with a list of available signals that can be recorded and streamed. This allows the device to select and prioritize the signals that are relevant to the specific study. Then the client specifies the type of data they wish to receive and the communication process with the Thalamus is initiated.

In addition to receiving data streams, our toolkit also allows clients to act as recording devices themselves and send their signals back to the Thalamus Core. For example, in an experiment where researchers would like to share the mouse movements of one client with another, the mouse signal can be streamed back to the Thalamus Core and be broadcasted to the other clients.

2.4 Built-in features and functions

Our simulation toolkit offers a variety of built-in functionalities, such as dealing with missing data, applying common filters, synchronizing multiple signals, simulating delays, and adding signal noise. We explain these key features in the following sections.

2.4.1 Missing values: In a real-world setting, missing values are a common issue that can occur during data collection. For example, in eye-tracking recordings, if the participant looks outside the screen, there would be no valid value of eye movement to record. To simulate this issue, Thalamus sends a conventional value such as `"0"` or `"NA"` as an indicator of missing values (see Figure 2). This allows the clients to recognize and properly handle these missing values.

2.4.2 Filters: The ability to apply various kinds of filters to the data streams is another feature of our toolkit. For example, currently

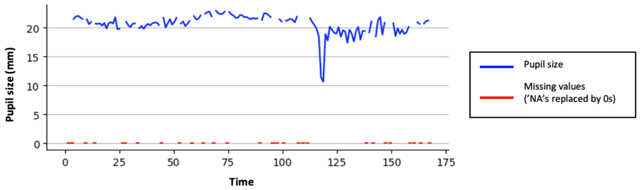

Figure 2: Demonstration of the toolkit's ability to handle missing values. This figure illustrates the pupil size values obtained from a sample of eye-tracking data. Missing values represented by "NA" are replaced with zeros.

Savitzky-Golay [15] (see Figure 3) and Kalman [7] filters can be used. These filters can be used to enhance the quality of the signal by reducing noise, boosting the signal-to-noise ratio. Because of the ability to apply various filters based on the objectives of the study, this feature gives researchers greater flexibility and adaptability.

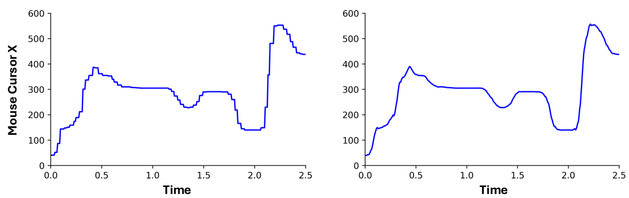

Figure 3: Demonstration of the toolkit's ability to provide common filters. An original signal, in this case the mouse cursor position in the X axis, is show on the left part. The same signal after being filtered with the Savitzky-Golay filter is shown on the right.

2.4.3 Signal synchronization: The ability to synchronize various data streams based on timestamps is another crucial component of our toolkit. This capability is essential for correctly and effectively syncing various data streams and extracting information that is pertinent from the data. It enables researchers to compare and examine data from many sources, as well as to extract events from the data streams. By ensuring that data from many devices is processed in a uniform and standardized manner (using UTC timestamps), it is simple to compare and evaluate data from various devices and during different time periods.

From a technical perspective, all connected or simulated devices should have their system clocks in time and submit their data associated with a Unix timestamp, which is a standardized time reference — the number of (milli)seconds elapsed since January 1, 1970. This way, Thalamus can synchronize all signals and extract specific time periods as needed. For example, in an experiment that is collecting brain signals and pupil size, when presenting a stimulus to the participants, Thalamus can extract this specific time period and synchronize the pupil size signal with the brain signal. Figure 4 illustrates this capability. The temporal precision in our proposed toolkit is in milliseconds, which is convenient for most studies in human-computer interaction.

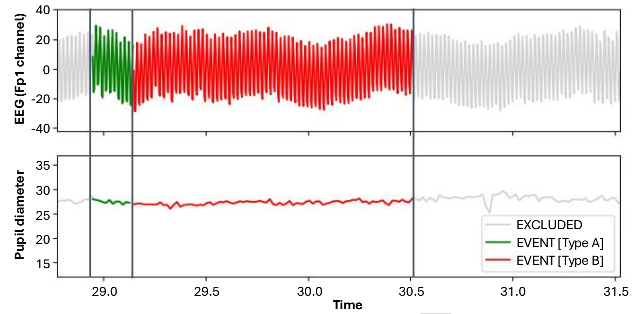

Figure 4: Demonstration of the toolkit's ability to synchronize simultaneous data streams. The upper timeseries represents the brain signal (EEG) whereas the lower timeseries represents the pupil diameter.

2.4.4 Noise: The presence of noise in the data streams is another pervasive issue that arises often in real-world circumstances. To address this, our toolkit has the capacity to mimic several kinds of noise, including fixed (constant), random, and Gaussian noise (see Figure 5). With the help of this feature, researchers may test and challenge their experimental setup in a controlled setting while accounting for any noise that may emerge during a genuine experiment using actual equipment.

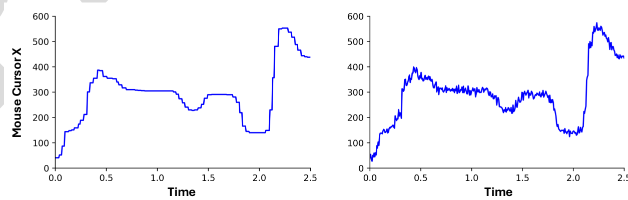

Figure 5: Demonstration of the toolkit's ability to simulate noise. An original timeseries, in this case the mouse cursor position in the X axis, is shown on the left part. The same signal is incorporated Gaussian noise, as shown on the right part.

2.4.5 Delay simulation: Delays in receiving data streams may be problematic in real-world situations. Our toolkit has the capacity to simulate delays, based on the timestamps or a buffer window (see Figure 6). By simulating various delay scenarios and variations, this capability allows researchers to find and refine strategies for synchronizing and processing data streams, which can help to increase the system's robustness by anticipating possible unstable connections.

3 Use case examples

In the following we illustrate a few use cases that allow for a comprehensive evaluation of Thalamus in practical applications.

3.1 Real-time applications

A common scenario while streaming data in real-time application is that delays may happen due to network issues or any other reason.

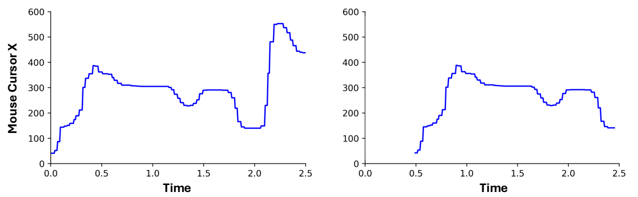

Figure 6: Demonstration of the toolkit's ability to simulate delayed signals. An original timeseries, in this example the mouse cursor position in the X axis, is shown on the left part. The same signal is incorporated a delay, as shown on the right part.

Researchers can simulate these delays using the built-in features of Thalamus to troubleshoot such suspected situations. Also, common filtering and preprocessing procedures can be applied. This step is conducted before streaming the data to clients, resulting in a significant reduction in computational costs. By implementing these procedures once and transmitting the processed data to all clients, Thalamus enhances efficiency and optimizes resource utilization for researchers.

3.2 Try before you buy

A research aims to conduct a study using Brain-Computer Interfaces (BCIs) to measure the emotional responses of people while watching video clips. Since there is a variety of BCIs in the market to collect EEG signals, each with a different number of channels, the researcher plans to utilize publicly available datasets such as DREAMER [3], MAHNOB [18], and SEED [25], which are specifically focused on emotional responses to videos using EEG signals, for the simulation. The researcher uses Thalamus as a cost-effective test method to decide if they should purchase a 62-channel EEG device (like in SEED) or if a 14-channel device (like in DREAMER) would suffice.

3.3 Device stress-testing

A group of researchers are planning to conduct an experiment to measure cognitive load, utilizing various modalities such as eye-tracking, ECG, mouse movements, and skin temperature. They have all the necessary recording devices but they do not know their operational ranges, so they use Thalamus to test a few recordings with each device. This approach also allows them to optimize their data processing methods, ensure their pipeline works as expected, and select the most suitable device for their experiment, all while saving time and resources.

3.4 Remote data collection

A researcher wants to develop an application that sends an alarm when high cognitive load is detected during online lectures, where attendees are wearing physiological sensors. In this scenario, content is streamed to all attendees through the Internet, who may be located in different places. Before creating the application, the researcher uses Thalamus to simulate multiple instances of different sensors using public datasets, as demonstrated in Section 3.2, and troubleshoot challenges like external noise, that could potentially affect the accuracy of some of the signals.

3.5 Broadcasting signals

A medical researcher is planning to conduct an experiment on telemedicine, in which a group of cardiologists will provide their opinions on the status of a patient. They should all receive the exact same ECG stream on their devices (e.g., computers, tablets, etc.) in real-time. The researcher uses Thalamus to stream a batch of pre-recorded, partitioned, and status-labeled ECG data to multiple devices, allowing the researcher to assess the efficiency of the designed experiment.

4 Discussion, Limitations, and Future Work

We have introduced Thalamus, a toolkit for simulating, capturing, and manipulating different data streams. Thalamus allows researchers to conduct a dry-run of their experiments, to test their equipment and setup before running the actual experiment with real users, which allows them to save valuable resources and expenses. Our toolkit includes features such as synchronizing multimodal signals, noise injection, delay simulations, and more. In sum, Thalamus provides researchers with a practical and streamlined method for simulating different data modalities.

One limitation of Thalamus is that it may not be able to simulate all types of data streams or devices one may think of "out of the box". It currently supports device simulation from structured files such as CSV or JSON. Future work will also consider relational databases such as MySQL or Postgres. Another limitation worth of mentioning is that programming skills may be needed to write custom controllers for some devices, for example those that do not provide direct access to the captured sensor data.

We also plan for future work to add more built-in functions to the Thalamus Core, to alleviate the need to write them if they are commonly used by researchers. Since our software is open source, we welcome contributions from the community. We also plan to develop a graphical user interface program that will allow the experimenter to configure our toolkit in a visual environment, similar to the popular Max/MSP software [9]. Thalamus is publicly available at https://github.com/kayhan-latifzadeh/Thalamus.

Acknowledgments

This work is supported by the Horizon 2020 FET program of the European Union through the ERA-NET Cofund funding (BANANA, grant CHIST-ERA-20-BCI-001) and Horizon Europe's European Innovation Council through the Pathfinder program (SYMBIOTIK, grant 101071147).

References

[1] P. Bækgaard, M. K. Petersen, and J. E. Larsen. 2014. In the twinkling of an eye: Synchronization of EEG and eye tracking based on blink signatures. In *Proc. Intl. Workshop on Cognitive Information Processing (CIP)*.
[2] N. Dillen, M. Ilievski, E. Law, L. E. Nacke, K. Czarnecki, and O. Schneider. 2020. Keep calm and ride along: Passenger comfort and anxiety as physiological responses to autonomous driving styles. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.
[3] S. Katsigiannis and N. Ramzan. 2017. DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. *IEEE journal of biomedical and health informatics* 22, 1 (2017), 98–107.

[4] S. Kim, K. A. E. Patra, A. Kim, K.-P. Lee, A. Segev, and U. Lee. 2017. Sensors know which photos are memorable. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2706–2713.

[5] E. Y. Kimchi, B. F. Coughlin, B. E. Shanahan, G. Piantoni, J. Pezaris, and S. S. Cash. 2020. OpBox: Open source tools for simultaneous EEG and EMG acquisition from multiple subjects. *eNeuro* 7, 5 (2020).

[6] K. Latifzadeh and L. A. Leiva. 2022. Gustav: Cross-device Cross-computer Synchronization of Sensory Signals. In *The Adjunct Publication of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–3.

[7] Q. Li, R. Li, K. Ji, and W. Dai. 2015. Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. IEEE, 74–77.

[8] P. Lopes, L. L. Chuang, and P. Maes. 2021. Physiological I/O. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–4.

[9] A. Momeni and D. Wessel. 2003. Characterizing and Controlling Musical Material Intuitively with Graphical Models. In *Proceedings of the New Interfaces for Musical Expression Conference*.

[10] R. Murray-Smith, A. Oulasvirta, A. Howes, J. Müller, A. Ikkala, M. Bachinski, A. Fleig, F. Fischer, and M. Klar. 2022. What simulation can do for HCI research. *Interactions* 29, 6 (2022), 48–53.

[11] G. M. Notaro and S. G. Diamond. 2018. Simultaneous EEG, eye-tracking, behavioral, and screen-capture data during online German language learning. *Data Brief* 21 (2018).

[12] X. Peng, C. Meng, X. Xie, J. Huang, H. Chen, and H. Wang. 2022. Detecting challenge from physiological signals: A primary study with a typical game scenario. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–7.

[13] M. Ragot, N. Martin, S. Em, N. Pallamin, and J.-M. Diverrez. 2017. Emotion Recognition Using Physiological Signals: Laboratory vs. Wearable Sensors. In *Proc. Intl. Conf. Applied Human Factors and Ergonomics (AHFE)*.

[14] A. Riegler, A. Riener, and C. Holzmann. 2019. AutoWSD: Virtual reality automated driving simulator for rapid HCI prototyping. In *Proceedings of Mensch und Computer 2019*. 853–857.

[15] A. Savitzky and M. J. Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639.

[16] E. J. Shah, J. Y. Chow, and M. J. Lee. 2020. Anxiety on Quiet Eye and Performance of Youth Pistol Shooters. *J. Sport Exerc. Psychol.* 42, 4 (2020).

[17] N. J. Shoumy, L.-M. Ang, K. P. Seng, D. M. Rahaman, and T. Zia. 2020. Multimodal big data affective analytics: A comprehensive survey using text, audio, visual and physiological signals. *Journal of Network and Computer Applications* 149 (2020), 102447.

[18] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic. 2011. A multimodal database for affect recognition and implicit tagging. *IEEE transactions on affective computing* 3, 1 (2011), 42–55.

[19] Y. Sun, F. Guo, F. Kaffashi, F. J. Jacono, M. DeGeorgia, and K. A. Loparo. 2020. INSMA: An integrated system for multimodal data acquisition and analysis in the intensive care unit. *Journal of biomedical informatics* 106 (2020), 103434.

[20] D. Szajerman, P. Napieralski, and J.-P. Lecointe. 2018. Joint analysis of simultaneous EEG and eye tracking data for video images. In *Proc. IEEE ISEF*.

[21] R. Taib, B. Itzstein, and K. Yu. 2014. Synchronising Physiological and Behavioural Sensors in a Driving Simulator. In *Proc. Intl. Conf. Multimodal Interaction (ICMI)*.

[22] F. Wolling, C. D. Huynh, and K. Van Laerhoven. 2021. IBSync: Intra-body synchronization of wearable devices using artificial ECG landmarks. In *Proc. Intl. Symposium on Wearable Computers (ISWC)*.

[23] R. Xiao, C. Ding, and X. Hu. 2022. Time Synchronization of Multimodal Physiological Signals through Alignment of Common Signal Types and Its Technical Considerations in Digital Health. *J. Imaging* 8, 5 (2022).

[24] J. Xue, C. Quan, C. Li, J. Yue, and C. Zhang. 2017. A crucial temporal accuracy test of combining EEG and Tobii eye tracker. *Medicine* 96, 13 (2017).

[25] W.-L. Zheng and B.-L. Lu. 2015. Investigating Critical Frequency Bands and Channels for EEG-based Emotion Recognition with Deep Neural Networks. *IEEE Transactions on Autonomous Mental Development* 7, 3 (2015). https://doi.org/10.1109/TAMD.2015.2431497

[26] P. H. Zimmerman, J. E. Bolhuis, A. Willemsen, E. S. Meyer, and L. P. Noldus. 2009. The Observer XT: A tool for the integration and synchronization of multimodal signals. *Behav. Res. Methods* 41, 3 (2009).