# Developing and exploiting optogenetic feedback control in mesoscale neuroscience

**Thesis Proposal**
**Biomedical Engineering PhD Program**
**Georgia Institute of Technology and Emory University**

Kyle Johnsen

10/20/22

As the importance of causal inference becomes increasingly recognized in neuroscience, the need for technology enabling precise manipulation of neural variables becomes apparent. Feedback control is an important class of such manipulations for its ability to increase inference power by reducing response variability. Widely used throughout the engineering disciplines, it has had a significant impact through a variety of techniques (e.g., voltage clamp, dynamic clamp) on cellular neuroscience. However, feedback control has yet to be widely applied at the mesoscale/circuit level despite recent improvements in interfacing technology, such as optogenetics. Challenges to adoption include the complexity of implementing fast closed-loop experiments, the need to adapt the mature methods of control theory to the idiosyncratic constraints of systems neuroscience experiments, and the lack of established technical guidelines for applying feedback control to address complex scientific questions.

In this work I propose to begin to address these challenges in three aims. In Aim 1, I develop a simulation framework for easily prototyping closed-loop optogenetic control (CLOC) experiments *in silico*, thus allowing neuroscientists to test and iterate on experimental designs without the costs of in-vivo experiments or up-front investments in compatible hardware-software systems. In Aim 2, I will translate sophisticated model-based feedback control algorithms to the realistic experimental setting of bidirectional CLOC—the simultaneous use of both excitatory and inhibitory opsins. I will demonstrate some advantages of bidirectional CLOC and how it is not well accommodated by the algorithms previously demonstrated. Finally, in Aim 3, I will explore how recording, stimulation, and control requirements vary in an example application of CLOC—controlling the latent dynamics of simulated neural population activity and assessing their causal relationship with behavior. I will model this population activity with recurrent spiking neural networks trained using state-of-the-art, biologically plausible methods, with differing degrees of brain-like architecture and task complexity. This work will thus provide the systems neuroscience community with a more accessible entry point for CLOC, more powerful algorithms for leveraging bidirectional control, and a point of reference for designing CLOC experiments capable of answering complex scientific questions.

## 0.1 Thesis committee

| | |
|---|---|
| Nabil Imam | Georgia Institute of Technology<br>Computational Science and Engineering |
| Chethan Pandarinath | Emory University & Georgia Institute of Technology<br>Biomedical Engineering |
| **Christopher Rozell (advisor)** | Georgia Institute of Technology<br>Electrical and Computer Engineering |
| Garrett Stanley | Georgia Institute of Technology & Emory University<br>Biomedical Engineering |
| Patricio Vela | Georgia Institute of Technology<br>Electrical and Computer Engineering |

# Table of contents

# Front Matter

# Specific Aims

As the importance of causal inference becomes increasingly recognized in neuroscience, the need for technology enabling precise manipulation of neural variables becomes apparent. Feedback control is an important class of such manipulations for its ability to increase inference power by reducing response variability. Widely used throughout the engineering disciplines, it has had a significant impact through a variety of techniques (e.g., voltage clamp, dynamic clamp) on cellular neuroscience. However, feedback control also has great potential at the mesoscale/systems level, potentially enabling researchers to *unambiguously infer the downstream effects of circuit/population-level neural activity*.

For a number of reasons, though, *this potential has not been widely realized*. The main challenges to wider adoption do not appear to lie with available technology, as the computational power and stimulation/recording requirements of feedback control are met by the ever-improving tools already available to neuroscientists, such as optogenetics and large-scale neural recording. I posit that the main challenges to adoption rather include the **complexity of implementing** fast closed-loop experiments, the need to **adapt the mature methods of control theory** to the idiosyncratic constraints of systems neuroscience experiments, and the **lack of established technical guidelines** for applying feedback control to address complex scientific questions. **The proposed work aims to begin to address these challenges, and thus strengthen the set of causal tools available to probe neural systems.**

## Aim 1: A CLOC experiment simulation testbed

One significant obstacle to closed-loop optogenetic control (CLOC) experiments is the cost of acquiring and configuring compatible hardware-software systems. Moreover, the maintenance of animals or cell cultures inherent in lab experiments can slow the pace of developing novel techniques. In Aim 1, I attempt to address these obstacles by developing a simulation framework for easily prototyping CLOC experiments *in silico*, thus enabling faster, cheaper CLOC experiment design and method development. We demonstrate the software's utility in different virtual experiments and provide it to the public as open-source software with thorough documentation.

## Aim 2: Bidirectional CLOC

Bidirectional CLOC—the simultaneous use of both excitatory and inhibitory opsins—is necessary for precise manipulation of neural systems, especially when maintaining naturalistic activity levels is important. However, the basic control theory methods previously used in conjuction with CLOC do not take actuator constraints into account and are thus inadequate for multi-actuator (i.e., multi-light source) problems. The field of control theory provides elegant, powerful solutions to this class of problems, but applying them requires interdisciplinary expertise. In this aim I will translate more sophisticated model-based feedback control algorithms to the bidirectional CLOC setting and demonstrate the advantages both of bidirectional actuation and these algorithmic improvements.

## Aim 3: Using CLOC to manipulate latent neural dynamics

To our knowledge, CLOC has yet to be applied in answering complex systems neuroscience questions. In this aim, to pave the way for future *in-vivo* experiments that accomplish this, I propose to develop technical and conceptual guidelines as I control the latent dynamics of simulated neural populations. First, I will produce these virtual models by training recurrent spiking neural networks with state-of-the-art, biologically plausible methods—each differing in their degrees of brain-like architecture and training procedure complexity. I will then use the simulation testbed of Aim 1 to explore how recording, stimulation, and control requirements vary with the complexity and size of the system—thus giving researchers some idea of the relative importance of each factor of CLOC. Finally, I will demonstrate the conceptual utility of CLOC by quantitatively assessing the causal relationship between these latent dynamics and "behavior" (model output).

# 1 Background

## 1.1 Closed-loop control in neuroscience

Mesoscale neuroscience is currently undergoing a revolution fueled by advances in neural manipulation (*1–8*) and measurement (*9–16*) technologies as well as data analysis methods (*17–22*). These have yielded unprecedented datasets (*23*, *24*) and insights into network activity and plasticity (*25–29*), as well as novel experimental paradigms such as direct closed-loop control of neural activity (*30–40*).

An exciting emerging possibility is closed-loop control of neural activity (*30*, **other-reviews?**), enabling intervention in processes that are too fast or unpredictable to control manually or with pre-defined stimulation, such as sensory information processing, motor planning, and oscillatory activity. Unlike other forms of closed-loop control altering the environment [cite examples, mouse knee rotation, visual stimuli to achieve target response,] or using neurofeedback training (*8*) to achieve a neural or behavioral target, the direct control of neural activity itself can unambiguously reveal the downstream effects of that activity.

### 1.1.1 Types of closed-loop control

Closed-loop control of neural activity can be implemented in an event-triggered sense [cite a bunch of examples, inhibiting seizures, altering power, SWR disruption, ]—enabling the experimenter to respond to discrete events of interest, such as the arrival of a traveling wave [cite Reynolds] or sharp wave ripple [cite some review paper]—or in a feedback sense [cite 2 Bolus papers, all-optical, any others], driving the system towards a target or along a trajectory. The latter has multiple advantages over open-loop control (delivery of a pre-defined stimulus): by rejecting exogenous inputs, noise, and disturbances, it reduces variability across time and across trials, allowing for finer-scale inference. Additionally, it can compensate for model mismatch, allowing it to succeed where open-loop control based on imperfect models is bound to miss the mark. Moreover, whereas traditional perturbation methods include lesioning (*41*), unnatural silencing, or extreme stimulation, feedback control poses a more naturalistic alternative, increasing generalizability.

## 1.2 Various scales and tools for closed-loop control

Closed-loop control of neural activity can be performed at multiple scales and with different sets of tools. At the smallest, sub-neuron scale, dynamic clamping has yielded decades of fruitful research (**hodgkin53?**, **cite?** some-review-paper) in the forms of tools such as the dynamic clamp and voltage clamp, controlling electrical properties of small patches of membrane. The frontiers of this small-scale neuroscience often involve scaling up to multiple neurons and scaling down to subcellular structures such as dendrites (**cite?**), but multi-electrode intracellular recording face limitations (*42–44*, **cite?** more, not just patch clamping). Optical tools—e.g., optogenetics and fluorescence microscopy—can circumvent the difficulties of working with electrodes at such small scales, but an optical approach is not yet feasible for this purpose. The obstacles lie mainly in recording technology: the kinetics of both voltage indicators (*16*) and intracellular calcium (**cite?**) are too slow to capture phenomena faster than a typical action potential.

By contrast, the current state of technology is ripe for innovating closed-loop control methods at larger scales of neural activity, from single neurons to populations and circuits. Several promising combinations of recording and stimulation modalities are possible and still relatively novel: electrode recording with optogenetic stimulation (*45*, our papers), fluorescence microscopy with electrical stimulation (**cite?**), fluorescence microscopy with photostimulation (all-optical control) (*36*, *37*, *46*, *47*, **flytzanis14?**), and fMRI with optionally transcranial (*48*, *49*) photostimulation. Each of these tool combinations has pros and cons in terms of spatial and temporal resolution, crosstalk (*50*), and degrees of freedom.

A natural starting point for many neuroscientists is the first of these tool sets—electrode recording combined with optogenetics—since the two methods are so widely used, interfere little with each other (as long as metal electrodes are not directly illuminated (*45*, *50*)), and allow for genetically targeted stimulation. I will henceforth refer to this combination as CLOC, following the convention established by previous works (*34*, *35*).

## 1.3 Previous work

The proposed work builds on the work my lab and collaborators have done previously in implementing CLOC feedback control. Newman et al. (*33*) demonstrated bidirectional CLOC for fixed firing and slowly varying rate targets and using a model-free proportional-integral (PI) control scheme *in silico* (Figure **??**), as well as unidirectional integral control in the anesthetized rat. Bolus et al. (*34*) used PI control again, but developed a more principled approach to set estimation and control parameters and tracked dynamic firing rate trajectories down to a ~100-ms timescale (Figure **??**). Bolus et al. later employed more sophisticated and scalable optimal feedback control methods which are more robust to disturbances—important especially in awake animal experiments, where dynamic brain state changes contribute to high per-trial variability (Figure **??**).

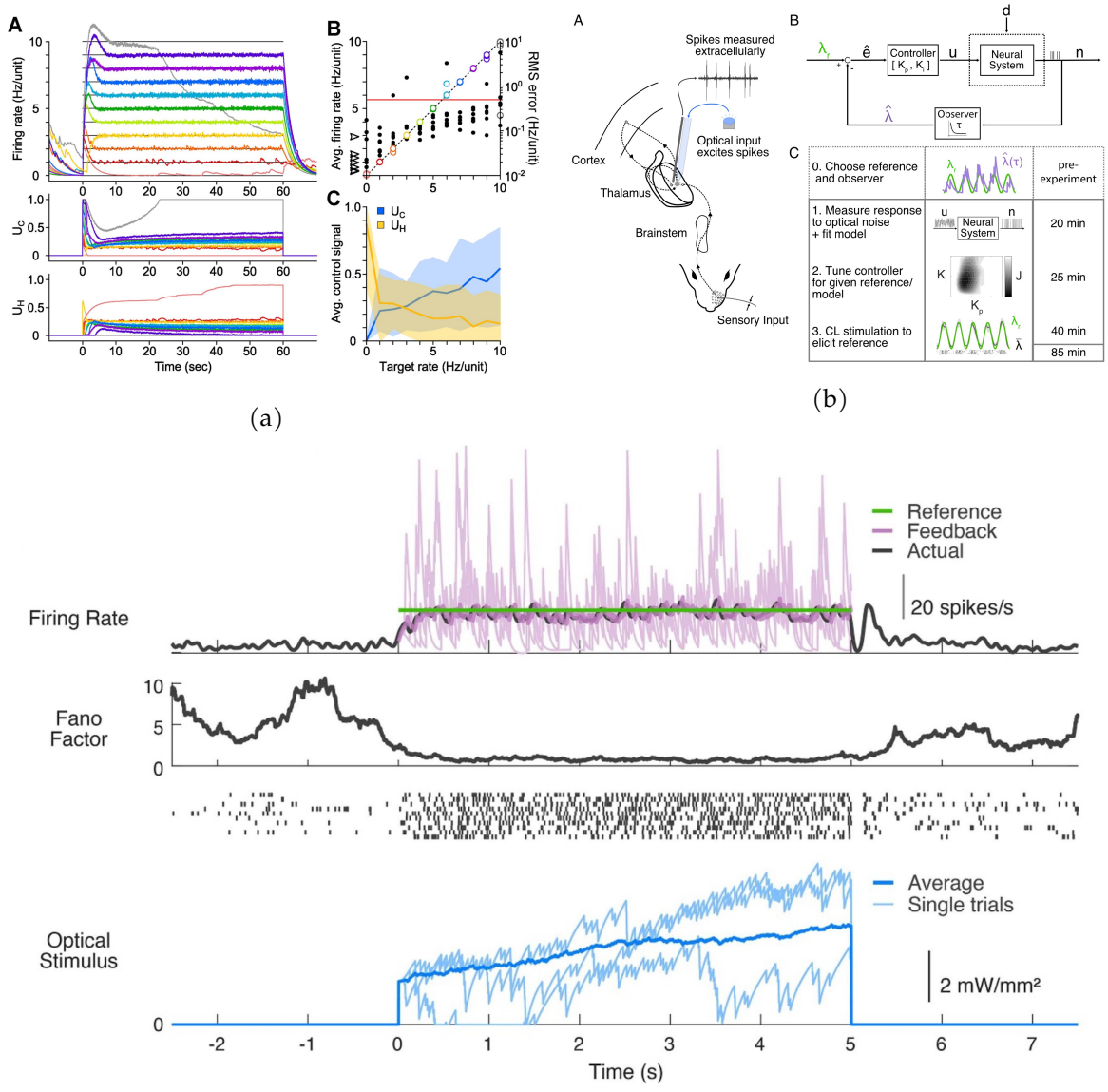


Figure 1.1: Previous CLOC experiments. (A) Figure 2 from (*33*), demonstrating control clamping a cultured neuron to different firing rates. $U_C$ refers to the control signal for channelrhodopsin2(H134R) ($ChR2_R$), parametrizing 470-nm light delivery. $U_H$ likewise parametrizes 590-nm light delivery to activate enhanced halorhodopsin-3.0 (eNpHR3.0). (B) Figure 2 from (*34*), outlining an *in-vivo* experiment setup, a control diagram showing how optical input is determined from the firing rate estimated in real time, and a strategy for tuning the controller. (C) Figure 4b from (*35*), showing how CLOC clamps the firing rate of a single thalamic neuron in the awake mouse over multiple trials, reducing response variability (as measured by the Fano factor).

## 1.4 Potential applications in mesoscale neuroscience

Seeing that a suitable technological foundation for feedback control of neural activity has already been laid with CLOC, I turn now to a discussion of several exciting areas where CLOC could further causal hypothesis testing. These deal with concrete questions of scientific interest, as opposed to the conceptual/technical advantages previously explained (Section **??**). Neuroscientists often identify specific variables or phenomena to assess their role in a larger neural system, in search of interpretable components of brain activity. A natural application of CLOC is to control these variables and phenomena of interest directly to enable stronger inference of their relationship to downstream variables. Examples of these potential targets for control include the activity of different cell types (*51*, **more?**); the type (*52–54*), frequency (*55*), amplitude (*55*), spike coherence (*56*, *57*) and relationship (*58*, *59*) of different oscillatory patterns (*60*); discrete phenomena such as bursts (**cite?**), sharp wave ripples (*61*), oscillatory bursts (*8*, *62*, *63*), traveling waves (*64*, *65*), or sleep spindles (**cite?**); and latent states describing neural dynamics (*66*, **cite?** review, Shenoy lab causal test, Hantman), including those most relevant to behavior (*20*, *67*, *68*).

While some of these targets lend themselves easily to CLOC, others require continued innovation in interfacing technology. While recent advances in recording technology allow us to infer neural state with unprecedented precision (*9*, *10*, *15*, *16*, **calcium-imaging?**, **GEVI?**), available actuation technologies are much more limited in their degrees of freedom and thus unlikely to sufficiently control what is observed. For this reason, the development of micro-LED arrays (**cite?**), multi-channel optrodes (*8*), and holographic optogenetic stimulation (*37*, **cite?**) are of particular interest. Moreover, rigorous investigation of the importance of recording and stimulation capabilities relative to each other would be helpful in guiding technological development and experimental design.

In addition to controlling variables of interest, CLOC can serve a paradigm of decoupling variables. This could be in the context of a circuit, where clamping the activity of a given node decouples its activity from all inputs except for the controller. This functionally severs links in the circuit, aiding in circuit identification and in testing the function of different nodes and connections (**willats-clinc?**). Moreover, CLOC could be used in the more general sense of controlling for confounding variables. For example, one might want to manipulate the synchrony of a population without changing the mean firing rate, or vice-versa. Whereas the conventional open-loop stimulation approach might accomplish this through tedious titration of stimulation parameters (*69*), the feedback control approach could simultaneously manipulate both variables as desired, requiring only a passable model of the system.

## 1.5 Innovation

Despite CLOC's great promise to be applied in these areas, it has not yet been widely applied in mesoscale neuroscience. As outlined in **?@sec-aims**, I identify three main reasons for this,

which I will begin to address. First, CLOC experiments are difficult and costly. I propose lowering the barrier to entry and the cost of experiment design and method development for CLOC experiments by creating a simulation framework, since **existing mesoscale neural network simulators do not contain the necessary ingredients for CLOC simulation**. Second, the algorithms previously used for CLOC are not well suited for actuation via multiple simultaneous light sources. I propose adapting more **powerful control theory methods** to enable bidirectional CLOC, which to our knowledge **has not been done previously**. Third, technical and conceptual guidelines for the effective application of CLOC do not exist because **CLOC has not yet been applied to answer a complex scientific question**. I propose to model how this can be done by controlling latent neural dynamics *in silico*, exploring how technical requirements scale with model and experiment parameters and inferring a causal relationship between latent variables and model behavior.

# 2 Aim 1: A CLOC simulation testbed

## 2.1 Rationale

CLOC experiments are difficult and costly. This can be a barrier to entry for neuroscientists that find CLOC's advantages attractive. Their lab might lack the funds to invest in necessary hardware or the time to invest in adding high-performance signal processing loop to their experimental workflow. Or, they may possess the resources but do not want to spend them without some assurance that their proposed experiment would be fruitful. Finally, when the proposed experiment requires signal processing/control method development, iterating on designs *in-vivo* may be cumbersome, given the additional cost of animal care and training.

Taken together, these factors not only make CLOC experiments a significant investment for a researcher, but one laden with risk. Unknown properties of the system under study can make it hard to predict whether a proposed experiment (e.g., whether a given population of neurons can be controlled in a given way) is likely to succeed, and even more so when working with innovative methods with little precedent in the literature.

However, these costs and risks can be mitigated through in silico prototyping. Given a reliable model of the system of interest, one can simulate a proposed experiment, assessing the effectiveness of a given setup. Alternate models can be tested to assess robustness of the given method to unknown model properties, or a single method can be validated on a variety of models to determine its general applicability. This strategy not only allows for a researcher to evaluate and optimize methods before committing significant resources to them, but also accelerates the development cycle.

For these reasons, I have developed Cleo: Closed Loop, Electrophysiology, and Optogenetics experiment simulation testbed. Unlike existing software, Cleo simultaneously provides a high-level interface to fast and flexible neural network simulations; easy, model-independent injection of electrode recording and optogenetic perturbations; and a real-time, closed-loop processing module capable of modeling communication and processing delays inherent in real experiments. I thus provide a "free trial" to researchers who are unsure if CLOC will serve their research agenda, as well as a low-cost environment to design experiments and develop methods, for those who are already committed to the technqiue.

## 2.2 Guiding principles

Two factors drove our choice of recording and stimulation models to integrate into Cleo. First, because a main purpose of Cleo is to enable prototyping of experiments, we focused on models at the level of the parameters an experimenter is able to alter. Because parameters such as electrode location, channel count and density, and optic fiber depth and size are all defined naturally in spatially extended models, Cleo's electrode and optogenetics modules require a "spatial" network model where relevant neurons must be assigned coordinates in space.

Second, we assumed that highly realistic models of individual neurons are not needed to capture most mesoscale-level phenomena. Accordingly, Cleo was developed with the intention of using point neurons models, rather than multi-compartment neuron models with realistic morphologies. Simulating simpler neuron models offers advantages of speed and intuitiveness, which are important in the context of prototyping an experiment. This decision had consequences in our software (Section **??**) and LFP modeling (Section **??**) decisions.

In addition to our modeling priorities, the goals of usability, flexibility, and extensibility guided our choices in software dependencies and infrastructure. Ease of use is important to make Cleo as accessible as possible, especially to researchers with primarily experimental backgrounds who may not have extensive experience with computational modeling. This usability goal also motivated Cleo's modular design, which allows users to add different recording or stimulation devices with little or no modification to the underlying network model, easing the burden of testing a variety of experimental configurations. Flexibility in the underlying simulator, in addition to enabling compatibility with a wide variety of models, is necessary for arbitrarily interacting with the simulation in a closed-loop fashion. Finally, extensibility is important for the testbed to remain relevant under changing needs in the future, allowing for new functionality to be easily added in a "plug-in," modular architecture.

### 2.2.1 Limitations and workarounds

Because we prioritize point neuron models, Cleo does not currently provide tools for recording realistic extracellular potentials. This would preclude realistically simulating such methods as spike sorting. Currently, the sorted spikes signal Cleo can record assumes perfect sorting but could be made more realistic by adding sorting noise.

## 2.3 Closed-loop simulation architecture

We chose Brian 2 (*70*) as the spiking neural network simulator to build Cleo around. Brian is a relatively new spiking neural network simulator written in Python with multiple advantages. It flexibly allows (and even requires) the user to define models mathematically rather than

selecting from a pre-defined library of cell types and features, while maintaining the ease of a high-level interface. This keeps model and experiment details together and enables us to define arbitrary recording and stimulation models that easily interface with the simulation. Moreover, the Python programming language has the advantages of being open-source, intuitive to learn (*71*), and widely used in computational neuroscience (*72*, *73*). Users do not need to use any other languages to use Brian. Brian is also relatively fast (especially since it is developed primarily for point neuron simulations), as shown in benchmarks (*70*).

Cleo provides three modules—recording, stimulation, and closed-loop processing—for integration with an existing Brian model (see Figure **??**). Cleo's functionality centers around a `CLSimulator` object that orchestrates the interactions between these three modules and the Brian simulation by injecting devices, running the simulation, and communicating with an `IOProcessor` object at each time step. The `IOProcessor` receives recorder measurements according to a specified sampling rate and returns any updates to stimulator devices.

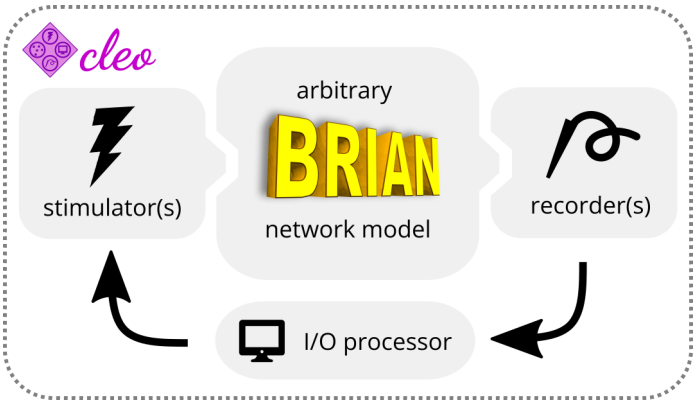


Figure 2.1: A conceptual diagram of Cleo's functionality. Cleo wraps a Brian network model, injects stimulation and recording devices, and interfaces with the network in real time through a simulated "I/O processor".

In order to simulate the effects of latency in closed-loop control, Cleo provides a `LatencyIOProcessor` class capable of delivering control signals after some delay. It does this by storing the outputs calculated for every sample in a buffer along with the time they can be delivered to the network. For example, if a sample is taken at 20 ms and the user wishes to simulate a processing and communication latency of 3 ms, the control signal, along with the output time of 23 ms is stored in a buffer which the simulation checks at every timestep. As soon as the simulation clock reaches 23 ms, the control signal is taken from the buffer and applied to the stimulator devices. Because the user has arbitrary control over this latency, they can easily simulate probabilistic delays if, for example, they wish to assess the effect of the experimental platform stalling occasionally.

By default, `LatencyIOProcessor` samples on a fixed schedule and simulates processing samples in parallel—that is, the computation time for one sample does not affect that of others.

Some alternatives are available, as illustrated in Figure **??**.

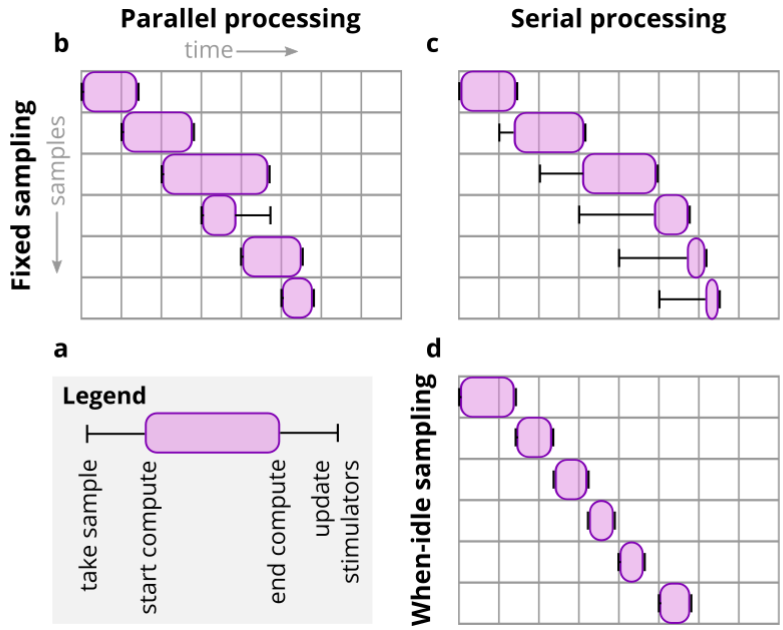


Figure 2.2: Latency emulation strategy and available configurations. (A) Cleo registers the time a sample is taken from the recording devices, the times the computation starts and ends, applying the user-specified delay, and updates stimulation devices when finished. (B) The default, parallel processing/fixed sampling mode. (C) The serial processing/fixed sampling case reflects when computations are not performed in parallel, but sampling continues on a fixed schedule. (D) The final processing mode avoids buffer overflow by sampling only once the computation for the previous step has terminated.

### 2.3.1 Limitations

One downside of using Brian is that it does not have the same first-class support for multi-compartment neurons as the popular NEURON (*74*) simulator. Using point neurons precludes advanced, morphology-dependent features of neural dynamics as well as recording and stimulation. However, if this feature is needed, Cleo could be developed further to integrate with Brian's morphological neuron features.

Both a strength and a limitation of Cleo's design is that it works with whatever model the user provides. This avoids the pitfall of offering stock models that claim to represent a researcher's system of interest, leaving it up to them to identify and/or develop a model that adequately describe the phenomena being studied. If a sufficiently realistic model for the studied system

does not exist, for example, developing one may be prohibitively costly, becoming a computational project in its own right as opposed to simply a stepping-stone towards an experiment. In these cases we suggest that a workaround could be to test a variety of potential models to identify which experimental configurations would be robust to unknown properties of the system. Indeed, the desired experiment in this case could be one that best adjudicates between these hypotheses (**willats-clinc?**).

## 2.4 Electrode recording

### 2.4.1 Spiking

Because we have prioritized point neuron simulations, the electrode functionality currently implemented in Cleo does not rely on raw extracellular potentials, which can only be computed from multi-compartment neurons (*75*, *76*). This biophysical forward modeling approach has been taken in other software (*77–80*).

To approximate spike recording without filtering and thresholding of extracellular potentials, Cleo simply takes ground-truth spikes and stochastically determines which to report using a detection probability function. This function is parametrized by a perfect detection radius, within which all spikes are reported, a half detection radius, at which distance there is a 50% chance a spike will be detected, and a cutoff probability, below which all neurons are ignored. The detection probability function is interpolated between the parametrized points with a $1/r$ function (*81*, *82*), where $r$ is the distance between the neuron and the electrode (see Figure **??**).

Cleo provides spike recording functionality in two forms: multi-unit and sorted. Multi-unit spiking reports every spike detected by every channel, without regard for the origin of the spike. Thus, each channel can report spikes from multiple neurons and a single spike can be reported on multiple channels. Sorted spiking reports all spikes detected on at least one channel, where each neuron is identified by a unique index. While real-time spike sorting is currently not feasible in practice for large numbers of contacts, this sorted spiking option could be used to emulate a common workflow of isolating one or a few neurons to report spikes from in real time.

### 2.4.2 LFP

In order to approximate cortical LFP without recurring to morphological neurons and biophysical forward modeling, we implemented the kernel LFP approximation from (*83*), which we term TKLFP (Teleńczuk kernel LFP). This method approximates the per-spike contribution to LFP (termed uLFP: unitary LFP) with a delayed Gaussian function, where amplitude and delay depend on the position of the neuron relative to the electrode. While the original

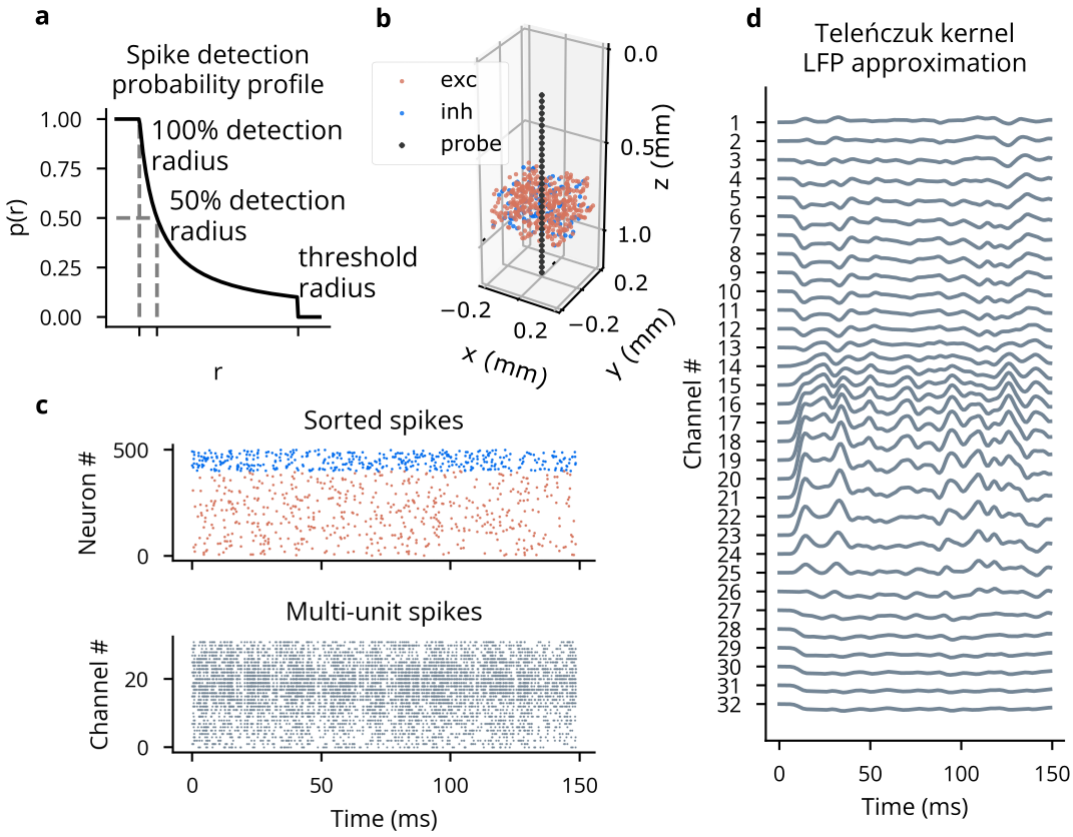


Figure 2.3: Illustration of LFP and spiking from Cleo's electrophysiology module. (A) The probabilistic spike detection model. All spikes within the 100% detection radius, 50% of spikes at the 50% detection radius, and none of those outside the threshold radius are recorded. The detection probability decays with 1/r. (B) An example plot generated by Cleo showing the positions of neurons and electrode contacts. (C) Randomly generated spikes for the neurons shown in B. Top: the sorted spike signal, which gives the ground truth source neuron for every spike as a perfect proxy for spike sorting. Bottom: multi-unit spikes, where spikes are reported on every channel they are detected on, regardless of the source neuron. (D) The TKLFP signal generated from the spikes in C for each channel. Y-axis units are not shown.

study included reference peak amplitude $A_0$ values at just four cortical depths, we inferred these values for arbitrary depths by performing cubic interpolation on data from their figure 5 and assumed that this profile dropped to zero at 600 μm below the soma and 1000 μm above. This implementation is available as a standalone Python package on PyPI (*84*). Accuracy of this implementation is verified in automated test suites in both TKLFP and Cleo packages.

### 2.4.3 Limitations

Using point neurons requires approximation of extraceullar potentials, which do not perfectly match ground-truth signals. The TKLFP approximation Cleo uses, for example, underrepresents high-frequency signals. Fortunately, there is an alternate, potentially more accurate, LFP approximation for point neurons that can be added to Cleo in the future if needed (*85*).

## 2.5 Optogenetic stimulation

### 2.5.1 Approach

#### 2.5.1.1 Light model

Cleo simulates optogenetic stimulation by combining a model of light propagation with an opsin model relating light to current. The light model is taken from (*86*) and uses Kubelka-Munk light propagation. This is a simplified model operating on the assumption that the medium is optically homogeneous, which, while not true, was shown by Foutz et al. to be a suitable approximation. Cleo includes absorbance, scattering, and refraction parameters for blue, 473-nm light as given in (*86*).

#### 2.5.1.2 Opsin models

Independent of the light propagation model, Cleo provides two different opsin models. One is a four-state Markov model as presented in (*87*), which captures rise, peak, plateau, and fall dynamics as the opsin is activated and deactivated through a Markov process. Additionally, by defining conductance rather than current directly, the model is able to reproduce the photocurrent's dependence on the membrane potential (see Figure **??**). While the four-state model fits experimental data fairly well, the code is structured so that the three- or six-state models in (*87*) could also be easily implemented.

Because the Markov model depends on somewhat realistic membrane potential and resistance values, however, it is not well suited for many pre-existing models that do not. For example, many commonly used leaky integrate-and-fire (LIF) neurons define the membrane potential as ranging from 0 to 1, rather than -70 mV to -50 mV, rendering both the units and values

(relative to the opsin's reversal potential) incompatible. One solution would be for users to adapt their neuron models for compatibility with this Markov opsin model, but since this would be burdensome, we developed an alternative model that simply delivers photocurrent proportional to the light intensity at each neuron. Thus, users can retain their original model with no need to add units or modify parameters.

In addition to options for opsin modeling, Cleo allows the user to specify both the probability that cells of a target population express an opsin and the per-cell expression level. This allows for the study of the impact of heterogeneous expression on the outcome of an experiment.

### 2.5.2 Results

The light model from (*86*) was successfully replicated, and the light intensity-firing rate relationship was qualitatively similar to that originally reported, though differing in some respects. This can be attributed to the use of point neurons rather than morphological neurons. See Figure **??** for details. Additionally, preliminary experiments show that the simplified, proportional current opsin model is able to produce a firing response qualitatively similar to that of the Markov model.

### 2.5.3 Limitations

Currently only channelrhodopsin-2 model parameters are included—comparing the effectiveness of different opsins will require first obtaining parameters. This is important since new opsins have been engineered with improved characteristics (*3*, *46–48*, *88–90*), as well as chloride pumps (*91–93*), channels (*94*, *95*), and other innovations (*4*, *93*)]. Thankfully, parameters for many opsins are available in published literature (*96–100*).

Another limitation is support for multiple simultaneous opsins or light sources. At present, the user could manually include separate current terms for each opsin in neuron model equations or approximate spectral overlap by adding a fraction of a light source's intensity to that of another, but this potentially slow workflow is antithetical to Cleo's goal of requiring minimum work to test different scenarios. We plan on adding this functionality shortly, since it will be important for Aims 2 and 3.

## 2.6 Usability & accessibility

### 2.6.1 Open-source code and documentation

Cleo is open-source and can be installed from the Python Package Index under the name `cleosim`. The code can be found on GitHub. Documentation, including an overview, tutorials,

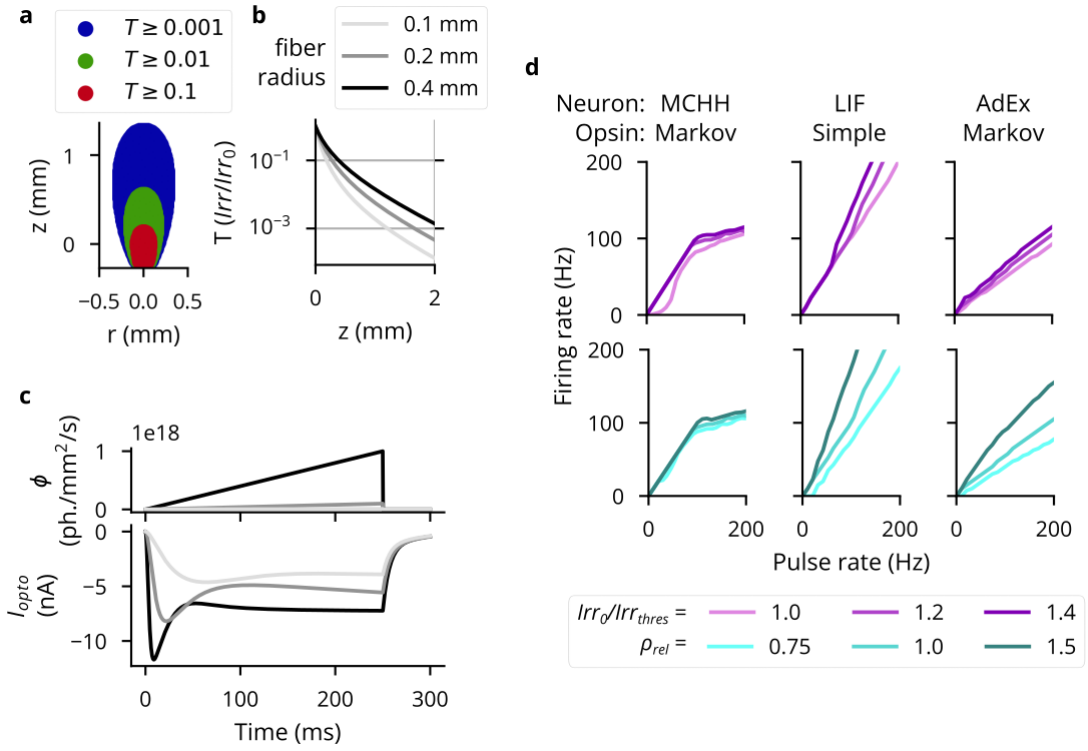


Figure 2.4: Validation of the optogenetics module. (A) Light transmittance T as a function of radius and axial distance from the optic fiber tip. Transmittance refers to the irradiance Irr as a proportion of the irradiance at the fiber tip Irr0 . After Figure 2a from (*86*) (B) Light transmittance T as a function of axial radius z for different optic fiber sizes. After Figure 2b from (*86*) (C) Photocurrent Iopto for ramping light of different intensities. After the figure produced by the "ramp" protocol from the default PyRhO simulator (*87*) (D) Neuron firing rates in response to optical stimulation with 5-ms pulse frequencies ranging from 1 to 200Hz. The left column re-plots data from (*86*). The middle column shows results for an LIF neuron with a simple opsin, and the right column for a tonic AdEx neuron with a Markov opsin model. The top row shows results for different light intensities: 100%, 120%, and 140% of the threshold for producing a single spike with a 5-ms pulse. The bottom row shows results for different expression levels relative to the default, $\rho_{\text{rel}}$. The irradiance used for these simulations was 120% of the single-spike threshold.

and API reference, can be found at .

### 2.6.2 Example experiments

In order to demonstrate Cleo's utility to the public, we implemented three example experiments to feature in the upcoming publication:

1. Closed-loop inhibition of a traveling wave in a rodent somatosensory cortex model (*101*).
2. Feedback control of layer 2/3 interneurons, disrupting plasticity in a model of primary visual cortex (*102*).
3. Optogenetic evocation of sharp wave-ripples in an anatomically detailed model of hippocampus (*103*, *104*). See Figure **??** for details, and note that feedback control, not looking ahead, so to speak, fails to evoke the reference signal at the desired time. The strategy I propose in Chapter **??** should remedy this.

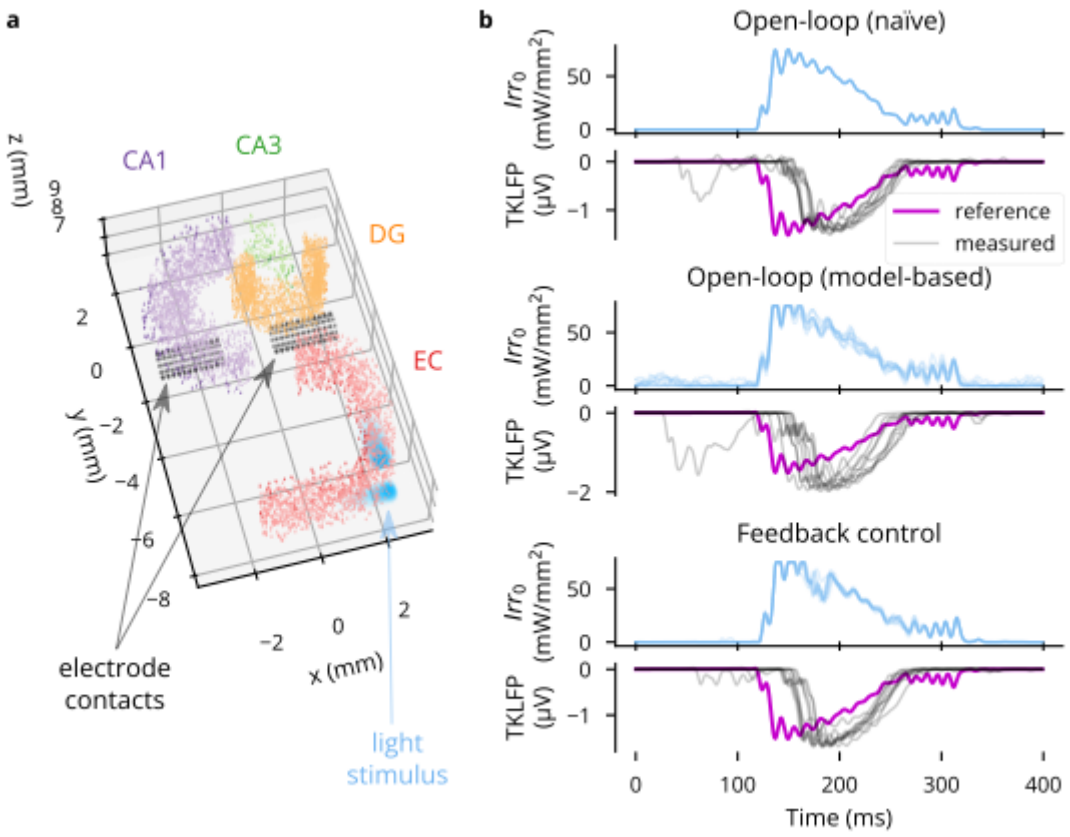


Figure 2.5: An example application of Cleo to the anatomical hippocampus model by Aussel et al.(*103*). (A) A 2.5 mm-thick slice of the 15 mm-tall model is shown using Cleo's built-in experiment plotting utilities. The model consists of four regions, entorhinal cortex (EC), dentate gyrus (DG), CA3, and CA1. Electrode contacts are represented as black dots and are in the same location as in the original model. Two light sources are shown in EC. Nine other such pairs (for a total of 20 light sources) not pictured here were spaced regularly parallel to the z axis. (B) Results are shown for ten trials each of three stimulation paradigms: naïve open-loop, where the input is simply a mirrored, rectified version of the reference signal; model-based open-loop, where a controller is run offline on a simulated model; and feedback control, where a controller is run using online measurements. Input Irr0 is the light intensity at the tip of each optic fiber. The system output TKLFP refers to the Teleńczuk kernel LFP approximation.

# 3 Aim 2: Multi-input CLOC

## 3.1 Rationale

### 3.1.1 Advantages of bidirectional control

The power of closed-loop optogenetic control (CLOC, henceforth referring specifically to feedback control; see Section **??**) is limited by the degrees of freedom provided by the optogenetic actuation scheme. One important distinction in possible actuation schemes is between "unidirectional"—control by a single opsin type—and "bidirectional"—where both excitatory and inhibitory opsins are used. Unidirectional control has obvious shortcomings: for example, an excitatory opsin alone can only raise the firing rate of target neurons, not lower it or even clamp to a baseline level. This setup would also be unable to *lower* the firing rate quickly, in the case of a dynamic reference trajectory.

### 3.1.2 Advantages of model-based, optimal control

While a previous study (*33*) has already laid the foundation for bidirectional CLOC, it does not feature the generalizability and scalability of the model-based, optimal control algorithms introduced by later work (*35*) (see Section **??**) for unidirectional actuation. This adaptive linear-quadratic regulator (LQR) approach is more robust to disturbances and can scale to multi-input multi-output (MIMO) systems. Moreover, its behavior can be easily configured by setting penalties on state error, the control signal, and even the derivative of the control signal to encourage smooth actuation.

### 3.1.3 Challenges of combining

Thus, a natural goal for furthering CLOC is to combine the advantages of bidirectional actuation and model-based optimal control—however, this poses additional challenges and opportunities. Unfortunately, the adaptive LQR method previously developed is unsuitable for bidirectional actuation because it does not model the constraint that the input (light intensity)