

Aalto University
School of Science
Master's Programme in Life Science Technologies

Sathish Kumar Narayanan

Engineering a Multi-Electrode Patch Clamp System:

A novel tool to quantify retinal circuits

Master's Thesis
Espoo, January 29, 2018

Supervisor: Professor Petri Ala-Laurila
Advisor: Dr. Anna Stöckl

Author:	Sathish Kumar Narayanan	
Title:	Engineering a Multi-Electrode Patch Clamp System: A novel tool to quantify retinal circuits	
Date:	January 29, 2018	Pages: vi + 60
Major:	Bioinformatics	Code: SCI3058
Supervisor:	Professor Petri Ala-Laurila	
Advisor:	Dr. Anna Stöckl	
<p>The human brain contains almost 100 billion neurons. They form distinct neural circuits that underlie the computational power of the brain. To understand how these neural networks function, high-detail physiological recordings from multiple identified neurons within a circuit are required. However, the technical possibilities to achieve this have been limited. Simultaneous patch clamp recordings from multiple well-defined neurons at the same time would give an excellent opportunity to obtain a deeper mechanistic understanding of neural circuit function. Thus, the goal of this master's project was to build the first state-of-the-art multi-electrode patch clamp system, along with acquisition and analysis software for retinal studies.</p> <p>This multi-electrode patch clamp system makes it possible for the first time to study at high physiological resolution how identified neurons in the vertebrate retina contribute to processing in small networks. The system is flexible to study other areas of the brain and can be extended to eight electrodes with only a few changes. The custom written software ensures protocol standardization for rig calibration, data acquisition, and analysis. All toolboxes are freely available as open source code, which ensures seamless collaboration between researchers and laboratories.</p>		
Keywords:	patch-clamp, electrophysiology, retina, data acquisition, data analysis, neural circuits, multi-electrode	
Language:	English	

Acknowledgements

I want to express my deepest gratitude to professor Petri Ala-Laurila. Thanks for providing me the great motivation and opportunity to be a part of this interesting project. I would like to thank my advisor Dr. Anna Stöckl for her friendly guidance throughout this project, Sami Minkkinen for assembling the mutli-electrode patch rig, professor Greg Schwartz (Northwestern University, USA) for his thoughtful advice and feedback on software design .

Thanks for the rest of the Twin Labs for the great support. Finally, I would like to thank my mom, dad and my little sister for supporting to be who I am.

Espoo, January 29, 2018

Sathish Kumar Narayanan

Abbreviations and Acronyms

LGN	Lateral Geniculate Nucleus
RGC	Retinal Ganglion Cell
DLL	Dynamic-Link Library
GUI	Graphical User Interface
LED	Light-Emitting Diode
LCD	Liquid-Crystal Display
OLED	Organic Light-Emitting Diode
DLP	Digital Light Processing
DIC	Differential Interference Contrast
IR	Infrared
CCD	Charge-Coupled Device
NDF	Neutral Density Filter
PCI	Peripheral Component Interconnect Express
COM	Communication Port
HDF5	Hierarchical Data Format version 5
API	Application Programming Interface
JSON	JavaScript Object Notation

Contents

Abbreviations and Acronyms	iv
1 Introduction	1
2 Background	3
2.1 Retina	3
2.2 Recording techniques	6
2.2.1 Patch clamp technique	6
2.2.2 Multi-electrode recording techniques	8
2.3 Standard patch clamp setup	9
2.4 Acquisition and Analysis Software	11
2.4.1 Data acquisition system – Symphony	12
2.4.2 Visual stimulation system - Stage-VSS	15
3 Aims	17
4 Electrophysiology Setup	18
5 Software	24
5.1 Data acquisition	24
5.1.1 Architecture	25
5.1.2 Light calibration module	28
5.2 Data analysis	30
5.2.1 Analysis framework	32
5.3 Data Management	35
5.4 Software Maintenance	35
6 Discussion	38
7 Conclusion	41
A Equipments List	49

B	Acquisition Steps	52
C	Photoisomerization calculations	54
D	Light Calibration Data Format	56
E	Example Data Analysis	57
E.1	Analysis project	57
E.2	Acquisition data filter	57
E.3	Analysis execution steps	59

Chapter 1

Introduction

The brain is the most complex biological organ. The human brain contains almost 86 billion neurons, and that of a mouse still totals approximately 70 million neurons [32]. These neurons form distinct circuits that underlie the computational power of the brain and control the behavior of the organisms [11]. Understanding how these neural circuits function is one of the greatest challenges of modern neuroscience [15].

The development of new technologies has been central to the discovery of fundamental principles in neuroscience. A classical example is the use of newly developed histological staining techniques, combined with microscopy, which enabled Ramon y Cajal to lay the foundation for the "neuron doctrine" [30]. This central theorem of modern neuroscience states that the nervous system is composed of discrete individual cells, rather than being a continuous reticular mesh. In a similar way, the development of new technologies has revolutionized the study of the nervous system throughout the last century [1]. These technologies have enabled investigations both at a very high resolution by examining the function of single ion channels using the patch clamp technique [51], or at a low resolution by observing larger brain areas using whole brain imaging techniques such as functional Magnetic Resonance Imaging [22]. Currently, large-scale projects, such as the BRAIN initiative, accelerate the development of tools and infrastructure for mapping individual neurons in identified circuits throughout the entire brain [31].

What is currently missing from the large pool of available neuroscientific techniques is a tool to record the activity of multiple identified neurons in a network at high physiological resolution. Monitoring both the inputs and outputs of identified neurons in networks is crucial for understanding how these networks process information – and how they eventually control complex behavior. While there are many tools to record the activity of multiple neurons in a network [54], only one single technique currently allows

for recording both excitatory and inhibitory inputs of identified neurons, as well as their output signals: the patch clamp technique [51]. However, this technique is currently limited to single cell and paired recordings in retina. Successful efforts have been started in recent years to expand it to four and more neurons in brain slice preparations [34, 69].

This thesis aims at implementing a four-electrode patch clamp setup for recordings in the vertebrate retina. The retina is a part of the brain and it is one of the most successful model tissues for the study of neural circuits, due to its high accessibility, its well-described morphology and the fact that its components can be activated by their natural stimulus: light. Implementing this novel tool for use in the vertebrate retina will greatly enhance our understanding of neural network function.

To achieve this aim, the thesis has two objectives: 1) explaining the design principles and strategy involved in assembling the state art of multi-electrode patch clamp system, 2) to design, develop, and integrate a data acquisition and analysis software for electrophysiological recordings. This four-electrode patch clamp setup is a pioneer of its kind. Detailing the strategy behind the selection of the different hardware components will ease the development of future multi-electrode patch clamp systems. The software developed in this thesis is made freely available and designed to ease sharing experimental approaches and data analysis across laboratories. Together, the detailed description of the setup and the newly developed software are part of the effort to develop new technology to help us understand how the brain works.

Chapter 2

Background

Information processing in neural circuits is best investigated by studying well-accessible model systems [31]. An example of an extensively studied model is the visual system. It is responsible for detecting, processing, and interpreting the visual world. Light from the external world reaches the specialized tissue present at the back of the eye called the retina, where it is processed and then send to the brain via the optic nerve (Figure 2.1). Studying generic neural computations in the retina and correlating them to behavior is a powerful approach to understand how information is processed in neural circuits [20].

2.1 Retina

The retina is part of the central nervous system and is a highly successful model system for neural circuit studies [36] for the following reasons: a) the cellular organization of retina has been extensively characterized [40] and is among the best understood of any sensory system [25], b) retinal circuits can be stimulated with their natural stimulus - light - with tightly controlled spatial and temporal characteristics, c) the outputs of the retina are well characterized and accessible across entire neural populations [4].

Information in the retina travels from the photoreceptors, via bipolar cells to the retinal ganglion cells (RGC), the retina's output neurons. Light from the external world reaches the photoreceptors (Figure 2.2(B)), where the light energy is converted into electrical energy. The photoreceptors use these electrical impulses to transmit the information to 14 different types of bipolar cells [6]. The bipolar cells are then connected to more than 40 different types of RGC [5]. These RGCs act as a parallel image processors and encode feature specific information [25]. Amacrine cells, which connect bipolar cells and RGCs laterally, play a major role in integration of visual

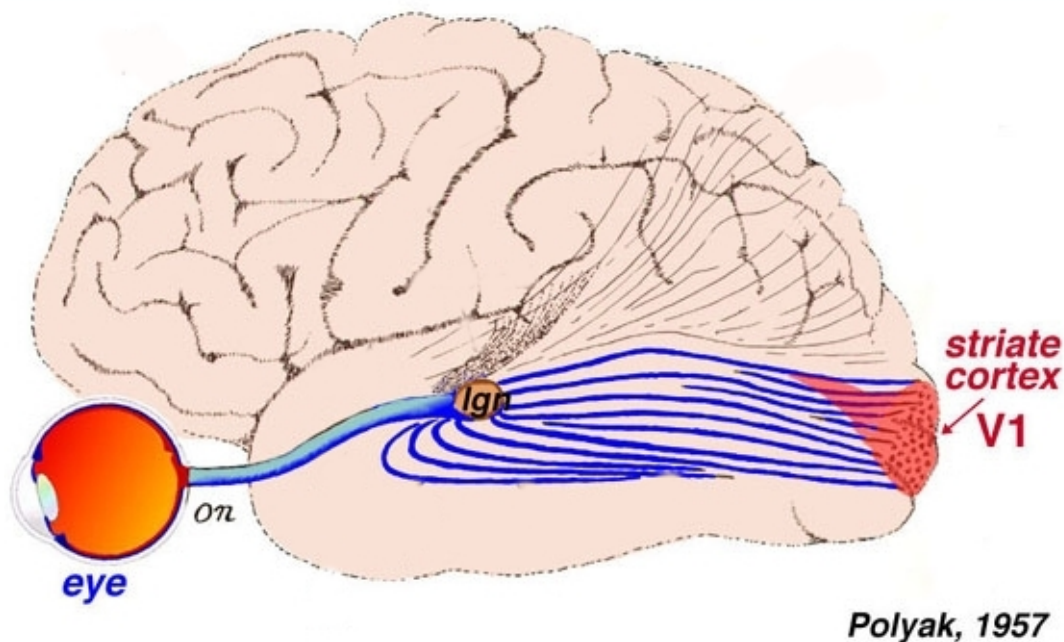


Figure 2.1: Visual information from the eye is relayed to upstream regions of the brain (LGN and primary visual cortex) via the optic nerve (on) modified from [55].

signal as they pass through the retina [7]. To understand how the visual information is processed by these retinal networks, the computations that take place in identified neural populations at each level need to be quantified physiologically.

In the retina and in many brain circuits, populations of neurons, rather than single neurons, encode the entirety of the information [47]. In most of these circuits, the responses of the neurons in the population are correlated, which can improve coding fidelity and reliability. Nonetheless, the response variability, or noise, can also be correlated between neurons, with both beneficial and detrimental effects for information coding [3].

The development of a variety of electrophysiological recording techniques (detailed in 2.2) has enabled the study of synchronous and correlated activity in the retina and the brain. In addition, theoretical and computational models complemented by experimental data, have greatly enhanced the understanding of how signal and noise correlations shape the information carried in a population of spike trains [3]. However, the encoding of naturalistic stimuli is not well explained by these models [66]. The accuracy of the mod-

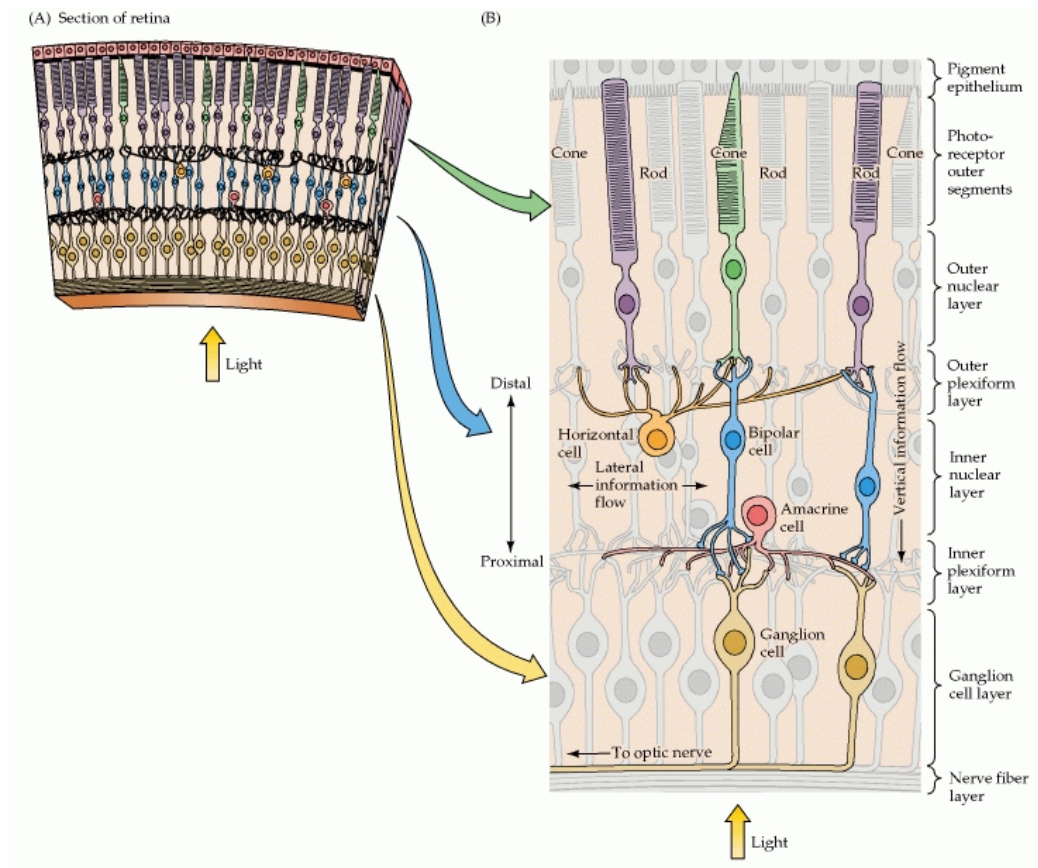


Figure 2.2: The structure of the retina. (A) Section of the retina. (B) The basic circuitry of the retina. Photoreceptors, bipolar cells, and ganglion cells provide the direct information transmission to the brain via the optic nerve. Horizontal cells and amacrine cells mediate lateral interactions in the outer and inner plexiform layers, respectively [52].

els could be greatly improved by experimental data obtained at the level of input currents and output potentials from a small population of identified neurons.

A unique recording technique is required to record input currents from neurons: the patch clamp technique. It is currently only available for pairs of retinal ganglion cells, but not for recordings of larger populations. In the next section, I will give a short overview of the evolution of electrophysiology recording techniques and explain the basic principles of the patch clamp technique.

2.2 Recording techniques

In 1792, Galvani discovered that the function of the nervous system is linked to the electrical activity of neurons and that the ability to measure the electrical activity in neurons is the key to understanding nervous system function. In the 20th century, Cole, Curits, and Hodgkin, Huxley further studied the electrical activity in neurons by inserting mini-electrodes into giant axon of the squid and performed the first direct recordings of resting potentials and action potentials. Cole and George developed a new technique commonly referred as "voltage clamp", that allowed the direct recording of the membrane current at rest and during stimulation. In the late 1970s, Sakman and Neher modified the voltage clamp technique so that it records the ionic currents from a single ion channel which has become known as the patch clamp technique [53, 68].

2.2.1 Patch clamp technique

The patch clamp technique records electrical activity of a small set of ion channels down to a single ion channel of a cell. The technique involves sealing to the cell membrane with the tip of a blunt-ended glass pipette by applying gentle suction to the other end of the pipette to develop a high resistance seal. The tight seal is typically in the gigaohm range and provides the electrical isolation of the membrane patch. The ionic exchanges between the membrane patch and the pipette are then recorded by the chloridated silver electrode inside the pipette, which is connected to an amplifier.

The investigation of the current-voltage relationship of membrane ion channels (Figure 2.3) can be achieved through the voltage clamp configuration of the patch clamp technique. The membrane potential of the cell is compared to the command potential, and the current is injected to compensate the changes in membrane potential in a negative feedback loop. The required compensation current is a direct measure of the currents flowing across the membrane. The patch clamp technique exists in four different configurations depending on the cellular region of interest (Figure 2.4).

- a. **Cell-attached recording:** The cell membrane is not tightly sealed to the patch pipette and the membrane voltage is recorded. The advantage of cell-attached recordings is that they do not affect the intracellular contents. However, they cannot describe the sub-threshold activity of the neurons, only their action potentials.
- b. **Whole-cell recording:** The patched membrane is disrupted by briefly applying a strong pulse of suction. As a result, the interior of the cell

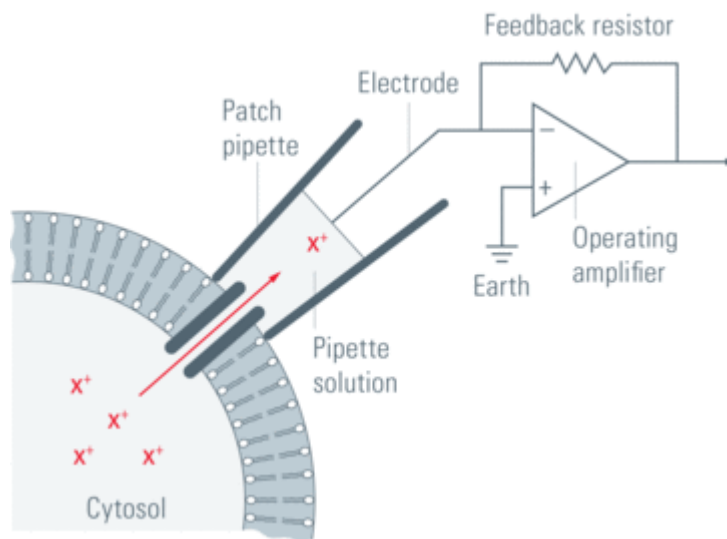


Figure 2.3: Patch-clamp recording setup with amplifier and feedback configuration [67].

body (cytoplasm) is accessible from the patch pipette. This method can be operated in voltage clamp mode by measuring the current from the whole cell under constant voltage, or the current clamp mode by measuring the voltage from the whole cell under constant current. In the whole cell configuration, the input currents the neuron receives from other cells can be measured, even when they are below the action potential threshold.

- c. **Inside-out recording:** The membrane sealed to the pipette is retracted, which breaks off the patch and provides access to the inner cytoplasmic membrane. It allows measuring single channel activity of the cell by exposing the cytosolic surface.
- d. **Outside-out recording:** This is an extension of the whole-cell recording in which the pipette attached to the cell membrane is retracted such that it causes rupture of the membrane followed by annealing. In this configuration, the channel activity can be studied by influencing the extracellular medium.

In addition to the different recording configurations, the voltage clamp technique provides a unique opportunity to isolate excitatory and inhibitory inputs to a neuron.

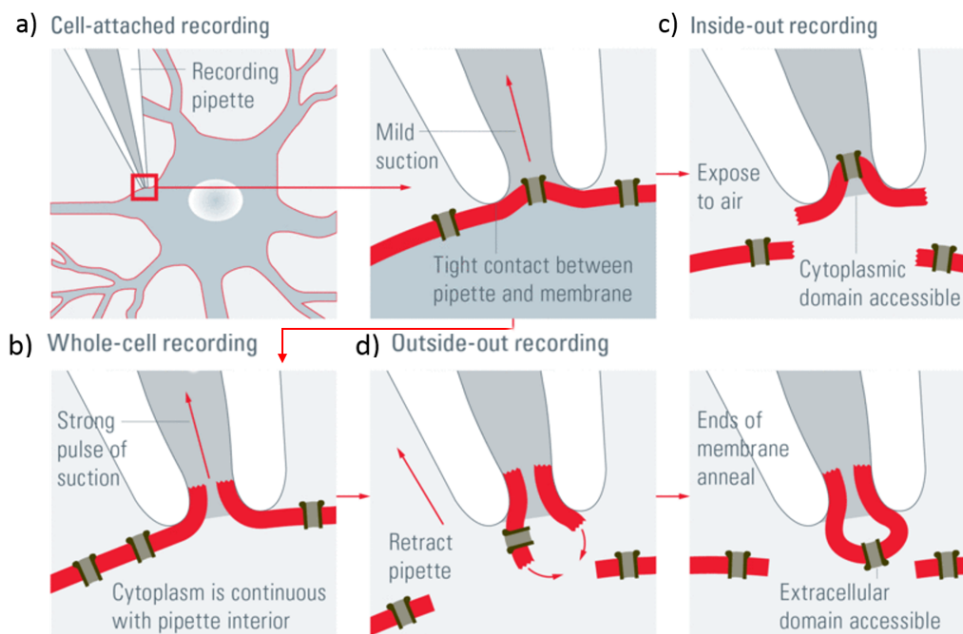


Figure 2.4: The four recording configurations of the patch clamp technique a) Cell-attached b) Whole-cell c) Inside-out d) Outside-out [67].

2.2.2 Multi-electrode recording techniques

Simultaneous physiological recordings from multiple neurons requires techniques such as multielectrode arrays (MEA) or multi-electrode patch clamp. The MEA is a recording technique, which enables extracellular recording from dozens to hundreds of neurons [37]. Although it has good spatial and temporal resolution, it cannot perform intracellular recording and lacks support for cell identification via neuron morphology. An example of a dense MEA placed over a patch of salamander retina is shown in Figure 2.5.

Multi-electrode patch clamp recordings are the only technique that provides access to input currents and output potentials of multiple cells at the same time (or multiple locations in a single cell), while also allowing identification of cell types by their characteristic morphology. However, multi-electrode patch clamp recordings of more than two neurons currently only exist for brain slice preparations [34, 69], but not for the retina. Thus, in this prime model for the study of neural circuit functions currently no technique is available to measure the input currents and output potentials of morphologically identified neurons in more than two cells. To overcome this gap, this master's thesis project developed a four-electrode retinal patch clamp setup. In the next subsection, the components required for a standard patch

clamp setup – which forms the basis for the setup assembled in this thesis – are introduced.

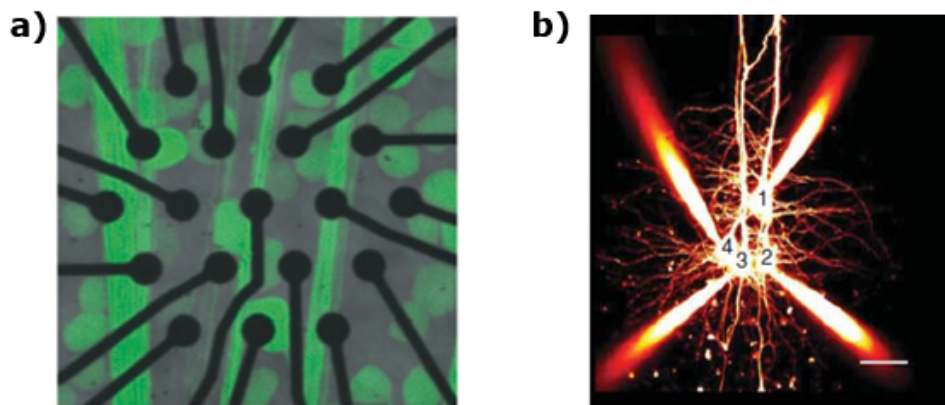


Figure 2.5: a) A hexagonal multi-electrode array placed over a piece of salamander retina (scale bar: 30 μm) [58] b) A 4 electrode whole-cell patch clamp recording performed in acute neocortical slices (scale bar: 50 μm) [34].

2.3 Standard patch clamp setup

A typical patch clamp recording generally includes the following steps: imaging the region of interest and selecting the neurons to record from, performing the electrophysiological recordings while stimulating the neurons. The signals are then amplified and digitized with the desired sampling rate using a data acquisition system. The following section briefly describes the components required for patch clamp recordings (Figure 2.6).

1. **Faraday cage and anti-vibration table:** the Faraday cage isolates the electrophysiological recording from electrical noise. One of the potential sources of noise is capacitive coupling which is generated by alternating electrical signals through stray capacitance. A grounded Faraday cage reduces such capacitive coupling between the noise sources and the recording electrodes. In addition, mechanical vibrations in the room or building are prevented from affecting the recordings by the anti-vibration table.

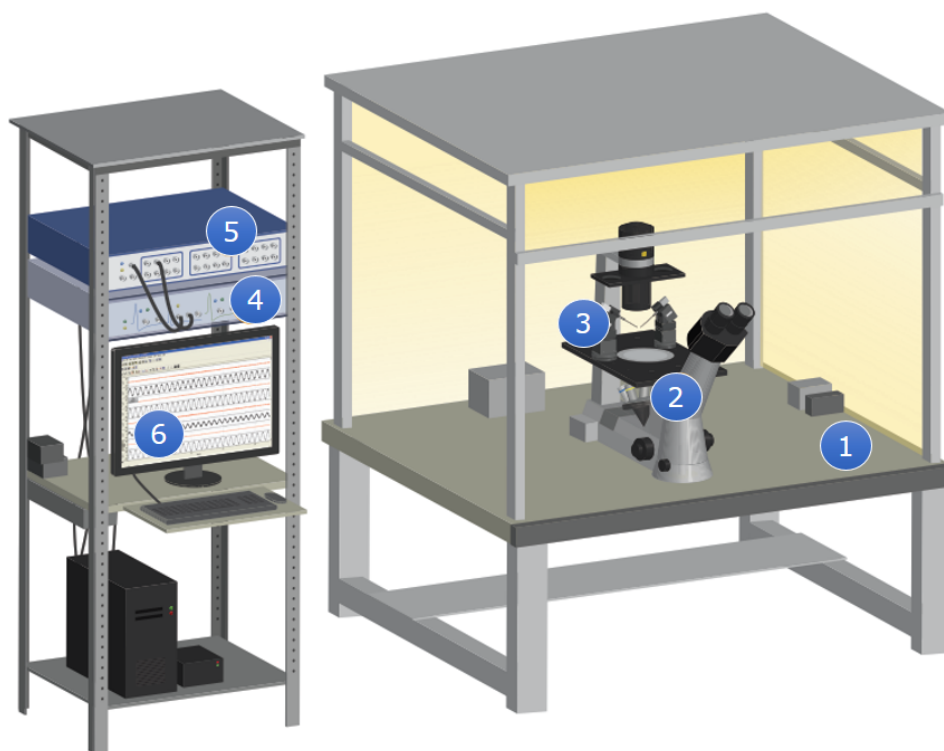


Figure 2.6: Standard patch clamp setup. Modified from [29].

2. **Microscope:** The microscope provides an optical magnification of the tissue under study and provides visual guidance while moving the patch pipettes to the target neuron.
3. **Micromanipulators:** Micromanipulators control 3-dimensional movement of the electrodes with micrometer precision. They play a key role in enabling stable electrophysiological recordings, as the patch pipette has to be placed with high precision on the cell membrane.
4. **Amplifier:** The observed ionic currents measured during the patch clamp recordings range in the order of pico (or) nano ampere, while the membrane voltages are in the order of millivolts. Additionally, the experimenter can stimulate the cells by injecting current. In order to input and record such small signals, a high fidelity amplifier is required. The ideal amplifier should have variable gain control and a high signal to noise ratio.

5. **Digitizer:** The output of the amplifier is analog. To display and analyze the acquired data with a personal computer, it has to be digitized. A digitizer carries out the analog to digital conversion by sampling the analog signal at a fixed frequency (the sampling frequency) and representing the time varying signal as binary values. The time and amplitude resolution of the signals depends on the sampling frequency. If the analog signal contains noise in the range of the sampling frequency or at higher frequencies, then the sampling can generate artifactual signals resulting from aliasing. In order to avoid such an aliasing error, the signal is generally low-pass filtered before digitization with a Nyquist frequency [35].
6. **Software:** It integrates the amplifier, digitizer, and other electrophysiology hardware, generally with a straightforward user interface. It facilitates the acquisition, analysis and storage of the acquired data.

The acquisition software is the key component that interacts with the electronic components of the patch clamp setup and defines the recording data format. A brief overview of the available open source acquisition software is given in the next section.

2.4 Acquisition and Analysis Software

In any electrophysiological experiment, efficient software is required to interface the devices of the setup and provide the experimenter with full control and optimal support. Although most of the devices in a patch clamp recording setup come with a proprietary software, using commercial programs has severe limitations. As the source codes of these software are closed, it is difficult to integrate them with other devices and software programs. Moreover, the range of experiments one can perform with the available closed set of features is limited. Hence, custom-developed, open source, device agnostic software programs are an important feature of modern electrophysiology setups, especially of those that push the boundaries of existing techniques, such as the multi-electrode patch clamp setup developed here.

Examples of such open source, custom-written data acquisition programs are AC4Q [13], a Python based framework; and OpenEphys [60], a C++ based data acquisition system. However, AC4Q is limited by the ability to annotate and analyze the experimental data in real time, whereas OpenEphys have a steep learning curve with the programming language used to develop the software.

The ideal acquisition software should have at least the following functionalities [24],

- to record the electrophysiological data together with the stimulation information.
- to capture the metadata of the device that was used to record the data
- to annotate the experiment with the information required for subsequent neuroscientific analysis

In addition, the acquisition software interface should be easy to understand, to use and to adjust by the experimenter, while abstracting the hardware specific source code. The next section introduces "Symphony" - a data acquisition tool that fulfills the above criteria.

2.4.1 Data acquisition system – Symphony

Symphony [12] is an open source Matlab based data acquisition system, which is used for electrophysiological recordings. It has a device agnostic programming interface and provides a built-in Graphical User Interface (GUI) to browse through the experimental data and to annotate the data during experiments.

Architecture: The Symphony architecture is split into two main components: Symphony core and the Symphony Matlab GUI. The Symphony core provides the necessary hardware abstraction and the Matlab GUI provides a well-designed and customizable user interface.

The core data unit of the Symphony software is called an "epoch". One epoch is a unit of time corresponding to a logical trial in the experiment (for example one repetition of a flash of light). The properties associated with each epoch can be changed in an experimental trial. For example, the time before the stimulation is defined as `preTime`, and the time during stimulation is called the `stim time` etc. These parameters are collectively called the `Stimulus Protocol` parameters. Symphony generally supports such `Stimulus Protocol` based data acquisition (Figure 2.7).

Symphony core: The core of the data acquisition software is developed in the C#.NET framework. It contains support for digitizers such as ITC-18 ; ITC-16 ; ITC-1600 (*InstruTECH, New York, USA*); NI-DAQmx (*National Instruments, "Austin, TX", USA*); and amplifiers such as MultiClamp 700A

Amplifier ; MultiClamp 700B Amplifier ; Axopatch 200A Amplifier ; Axopatch 200B Amplifier (*Molecular Devices, "Sunnyvale, California", USA*). These devices are configured mostly using dynamic linking libraries (DLL) wrapped in C# objects and used in the Matlab environment. The digitizer streams are controlled by the ITCMM DLL library, whereas the MultiClamp 700B (amplifier) data streams are sent as telegraphic messages which are trapped and intercepted using win32 interoperability libraries [45].

Symphony Matlab GUI: The Graphical User Interface (GUI) allows the user to select the configuration of the patch clamp setup (rig configuration), to choose the `Stimulus Protocol` to start the acquisition, and to save the acquired data. It is built on the Symphony core framework.

Before the start of an experiment, the user configures the mapping of external devices such as the amplifiers to the channels of the digitizer. The configuration is loaded from the GUI and saved as a system preference. During the experiment, Symphony core maintains a queue of epochs that are to be presented to the neurons. The user interface controls the start, end or pauses of the queue at any time. If the user pauses the queue in the middle of an epoch, the acquisition stops and the incomplete epochs are discarded. To add epochs to the queue, the user select the `Stimulus Protocol` from the interface. Its settings are displayed to the user in an editable mode. On click of `play/record` from the user interface, the Symphony core requests the next block of epochs from the `Stimulus Protocol` and appends those epochs to the queue of epochs to be presented, and for which data is acquired. The type of protocol and its setting can be saved as `Protocols Preset`. This allows the user to recall the `Stimulus Protocol` conveniently. The saved acquisition data can be visualized using the Data Manager user interface. The Data Manager (Figure 2.8) a) allows the user to annotate the recorded data with notes and keywords. It also provides the possibility to screen the previously recorded epochs and the associated stimulation parameters. Symphony persists the recorded streams, `Stimulus Protocol` parameters, and device-specific information in the hierarchical description file format - HDF5 [27]. The advantage of using HDF5 is that it is readable by many client libraries such as Java, Python, Matlab, and Julia. More detailed information about the data format is given in the Symphony documentation [12].

Symphony optionally supports the online analysis of recorded epochs and presents the results at the end of each epoch. However, Symphony itself only provides the epoch description, while the actual analysis has to be developed and attached to each `Stimulus Protocols` by the user. The analysis results are not saved during the process.

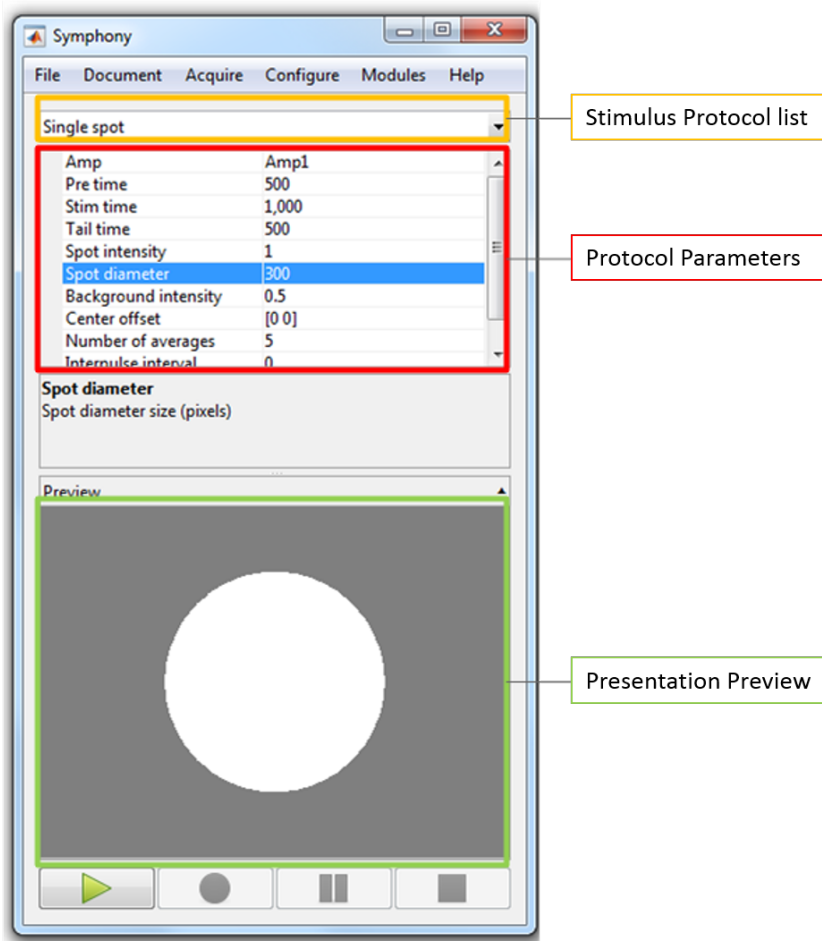


Figure 2.7: Symphony Matlab GUI. Shown is the acquisition user interface highlighting the `Stimulus Protocol` parameters window for a simple flash of light stimulus. The highlighted yellow region lists the available Stimulus Protocol, the red region shows the protocol parameters, and the green region shows the stimulus preview.

The `Stimulus Protocols` can be configured for various input devices, such as a Light Emitting Diode (LED), an external display such as Liquid Crystal Display (LCD), Organic Light Emitting Diode (OLED) microdisplay, or a Digital Light Projector (DLP). For external displays, stimuli with fine spatial and temporal characteristics can be created using the Stage-Visual Stimulus System (Stage- VSS), which is explained in next section.

2.4.2 Visual stimulation system - Stage-VSS

The Stage-VSS can be used to stimulate the retina with visual patterns with controlled spatial and temporal characteristics. It uses the OpenGL [72] standard for primitive graphic libraries to render patterns composed of different shapes and videos. The stage-VSS controls the graphics buffer and provides full access to the renderer via a simple Matlab object. The rate at which the actual pattern is displayed is limited by the underlying hardware.

Stage-VSS is designed as a client-server architecture [63]. Based on the choice of hardware, such as the graphics card and the RAM, the client server can be installed on a single computer or on two different computers. The stage server app supports remote stimulus presentation. The stage client connects to the server and the client sends information about the visual stimuli to the server. The server renders the stimuli to the graphics buffer. The visual stimuli can be composed of movies, images, or arbitrary shapes. The collection of visual stimuli to be presented during an epoch is called a `Presentation`. A detailed tutorial for creating a visual `Presentation` using Stage-VSS is presented in the Stage-VSS documentation [38].

Symphony combined with Stage-VSS provides an ideal acquisition system which can be seamlessly integrated with most common electrophysiology devices such as amplifiers, digitizers and external displays.

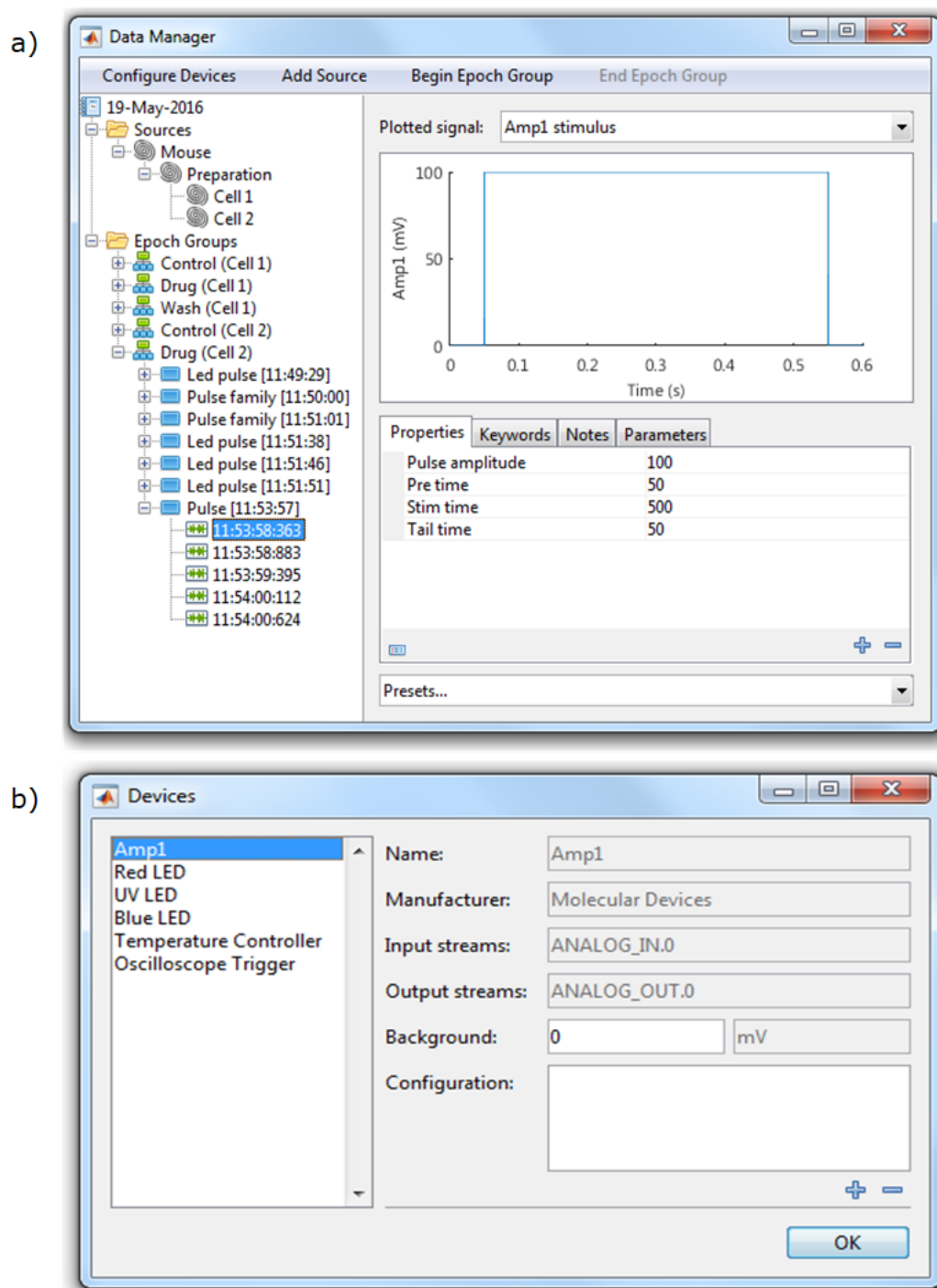


Figure 2.8: Symphony Matlab GUI a) The Data Manager facilitates data annotation and gives a simple preview of previously recorded epochs b) The Devices panel contains a list of configured devices and its properties.

Chapter 3

Aims

The aim of this master's thesis project was to extend the technical possibilities for studying neural processing in the retina substantially, by building a novel recording setup with integrated software programs:

- On the hardware side, a highly-automated multi-electrode patch clamp setup was to be assembled, to allow recordings of input currents and output potentials from four well-defined retinal neurons.
- On the software side, data acquisition and analysis systems were to be engineered, which support the above electrophysiological recordings.

Chapter 4

Electrophysiology Setup

The components of the multi-electrode patch clamp system, which were assembled for this master's thesis project, closely follow the building blocks of a generic patch clamp setup described in Section 2.3. The current section explains the main components of the multi-electrode rig in detail (Figure 4.1), and what makes them ideally suited to support effective patching of multiple cells at the same time in the retina.

Faraday cage & anti-vibration table

A custom built Faraday cage was mounted on the SmartTable OTS-UT2 (*Newport, "Irvine, California", USA*) anti-vibration table (Figure 4.1 e).

Microscope & Imaging

The SliceScope (*Scientifica, East Sussex, UK*) microscope is used to image the retina, via the following objectives: Olympus PLN10X/0.25 (*Olympus, Tokyo, Japan*) for a wider overview and Olympus LMPLFLN 60XW (*Olympus, Tokyo, Japan*) for the high magnification required for targeting the neurons. Swapping between the objectives is enabled by the SliceScope Motorized objective Changer (MOC) (*Scientifica, East Sussex, UK*). The MOC was calibrated for para focality using LinLab2 [57], a software developed by Scientifica to control their motorized microscope and manipulator devices. The Slicescopes Patch Pad controls the vertical movement of the microscope and the change between the objectives. Slicescopes is assembled with a motorized stage - the Motorized Movable Top Plate (MMTP) (*Scientifica, East Sussex, UK*)- to control the position of the retinal sample. The horizontal movement of MMTP is controlled by SliceScope patch pad (Figure4.1 c, d).

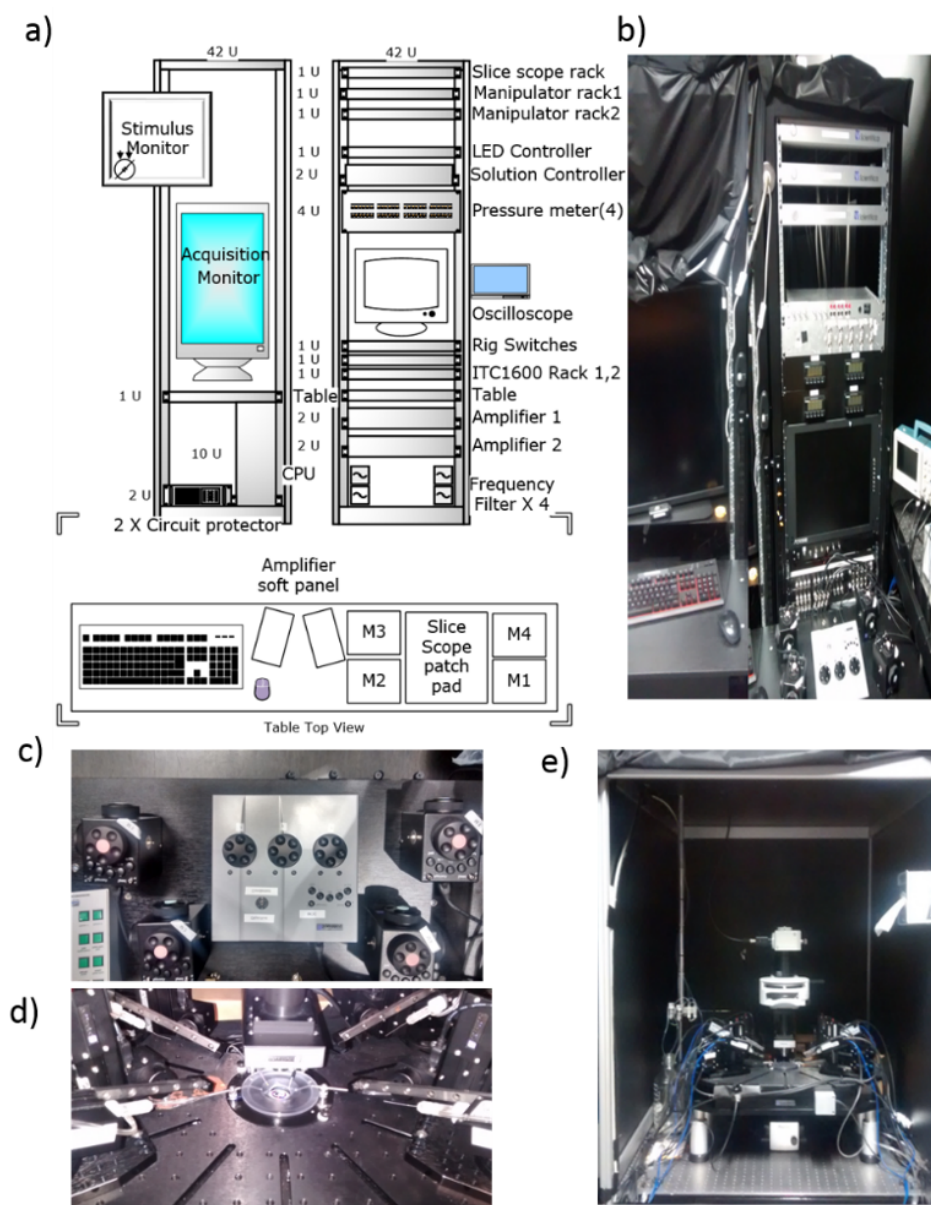


Figure 4.1: Components of the multi-electrode patch clamp setup. a) Layout of the multi-electrode patch clamp setup b) Actual rig components stacked on a rack, c) Top view of the micromanipulator control cubes, microscope and stage control panel, d) Microstar manipulators (M1, M2, M3 and M4) attached to the Scientifica Motorized Movable Top Plate, e) Four- electrode patch clamp setup inside the Faraday cage, assembled on top of the anti-vibration table.

The Slicescope is equipped with Differential Interference Contrast (DIC) microscopy. DIC microscopy ensures that less light is scattered and allows for deeper imaging into the tissue [18]. The retinal preparation is illuminated with an Infrared (IR) LED, with a peak wavelength of 940 nm. This ensure that imaging can be performed without light adaptation or bleaching of the retina, as the IR illumination is not detected by the photoreceptors. The IR illumination is controlled by the Lambda TLED (*Sutter Instrument, "Novato, California", USA*) controller and the DIC image is captured using a charge-coupled device (CCD) camera - WAT-910HX (*Watec France, Maine et Loire, France*).

Stimulation system

Once the patch clamp recording is established, the retinal neurons are stimulated with visual cues using the DLP LightCrafter 4500 (*Texas Instruments, "Dallas, Texas", USA*). LightCrafter 4500 provides a software interface to Symphony & Stage-VSS (section 2.4.1, 2.4.2) for controlling the color and pattern rate of the displayed stimulus.

In a patch clamp setup, the stimulus can either be projected through the condenser of the microscope, or through its objective. In our setup we chose the light path through the condenser, as it allows for functional imaging and visual stimulation to be performed simultaneously [71]. The size of the projected image and the light intensity is controlled using the optical components shown in Figure (4.2 b). The lenses in the optical path focus an image on the sample plane at micrometer resolution. In order to control the light intensity, light from the projector is attenuated using neutral density filters (NDF) in the FW102C - Six-Position Motorized Filter Wheel (*Thorlabs Inc, "Newton, New Jersey", USA*). The motorized filter wheel is controlled by the data acquisition software. As a result, the stimulus from the projector illuminates a defined region of the retina with controlled spatial and temporal characteristics and at a controlled light intensity.

Micromanipulators

Patching four cells in the retina simultaneously requires precise and smooth movements of the patch pipettes. Such micromanipulations are enabled by motorized manipulators with a freely moving three-dimensional axis and a fixed operating range. However, assembling four such micromanipulators is constrained by the space available on the stage. After thorough evaluation of

the commercially available solutions, we chose the manipulators - Microstar with control cubes (*Scientifca, East Sussex, UK*), mounted on the Motorized Movable Top Plate (MMTP) (*Scientifca, East Sussex, UK*). The advantage of using the MMTP and relatively small Microstar manipulators is that there remains enough space to accommodate four additional micromanipulators, and thus make it possible to extend the setup to 8 electrodes with very few changes.

Furthermore, the LinLab2 software [57] enables the synchronized movement of the MMTP and the micromanipulator with a specialized operating mode called "follow control".

Amplifiers

During data acquisition, the signals from the patched neurons are amplified by the pre-amplifier MultiClamp 700B Headstage CV-7B (*Molecular Devices, "Sunnyvale, California", USA*), followed by a further amplification via MultiClamp 700B Amplifier (*Molecular Devices, "Sunnyvale, California", USA*). MultiClamp 700B amplifiers can control the signal from two electrode channels. To enable recordings from all 4-electrode channel, two units of MultiClamp 700B were purchased and controlled by the MultiClamp 700B commander software [19]. As a next step, the amplified analog signals are low pass filtered using Bessel filter- 900CT (*Frequency Devices, "Ottawa, IL", USA*) to avoid noise-aliasing (Background section 5) and then send to the digitizer.

Digitizer

The analog to digital conversion of multiple channels requires synchronized processing of inputs and outputs. Hence, ITC-1600 (*InstruTECH, New York, USA*) was chosen as a digitizer, as it provides synchronization of up to 8 digital to analog channels and 16 analog to digital channels through the Peripheral Component Interconnect Express (PCI) (Figure 4.3). The digitized signals are then pulled by the acquisition software and presented to the electrophysiologist via the Symphony GUI.

In the next section, the detailed customization of the acquisition and analysis software is presented.

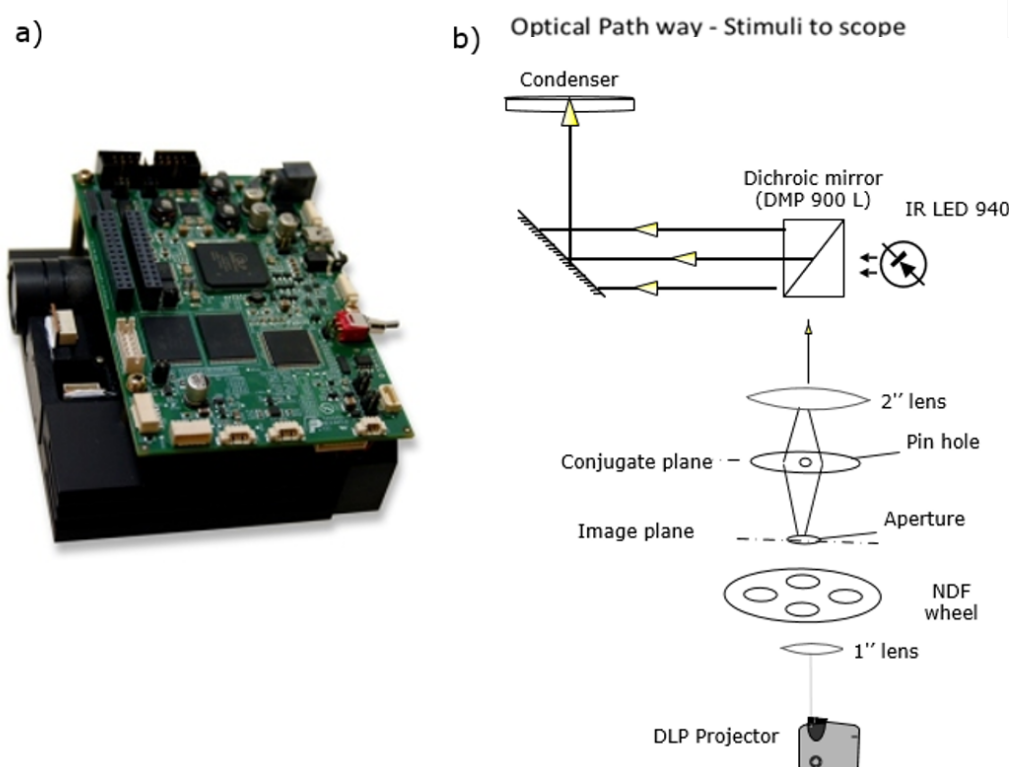


Figure 4.2: Visual stimulation in the patch clamp setup. a) Texas Instruments DLP Lightcrafter 4500 is used to project the visual stimuli. b) Optical path from the projector to the microscope's condenser. The neutral density filter (NDF) wheel provides adjustable light attenuation. The aperture, pin-hole and lenses in the optical path focus the image on a sample plane at micrometer resolution. The dichroic mirror enables visual stimulation at the same time as IR imaging.

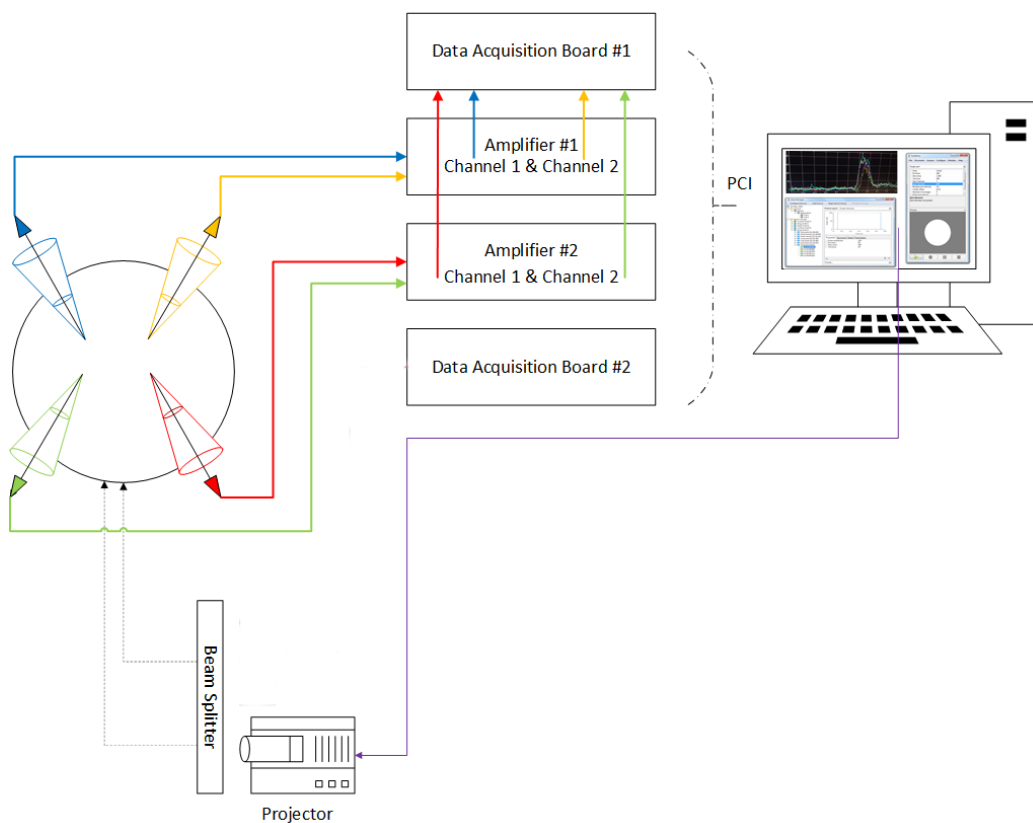


Figure 4.3: Wiring diagram of the amplifiers, digitizer, and stimulus device in the four-electrode patch clamp setup. The head stages are connected at the back of the amplifiers to the respective channels (1 and 2). The analog input from the amplifiers is then converted to Digital streams using a digitizer, and then sent to acquisition computer via the PCI interface.

Chapter 5

Software

At the software level, I developed Matlab toolboxes for 1) light source calibration, 2) online analysis and 3) offline analysis. The custom written software is integrated with the open source data acquisition system Symphony [12]. The newly developed software toolboxes ensure protocol standardization for rig calibration, data acquisition, and analysis across laboratories, and thus enable seamless collaboration between researchers. To align our source code with the systems that other leading laboratories in our research field use, we collaborated with the group of Prof. Greg Schwartz (Northwestern, USA). I directly coordinated with Sam Cooler, a PhD student in the Schwartz Lab, to assure that the newly developed code is compatible with existing standards. I primarily designed and developed the data analysis framework, inspired by an analysis tool previously developed by the Schwartz Lab [56]. All the source code is openly available at <https://github.com/Schwartz-AlaLaurila-Labs/> under the MIT license [42]. This section includes my contribution to the data acquisition and data analysis software.

5.1 Data acquisition

The acquisition software is the key component that interacts with all the electronic components for data acquisition and defines the structure of the recorded data. As explained in the background (Section 2.4.1), the Symphony data acquisition software provides a customizable interface by abstracting the acquisition hardware with simple Matlab code. Symphony already had built-in support for our acquisition hardware, the ITC-1600 digitizer and the Multi Clamp 700B amplifier. However, the stimulus projection device (in our case the DLP LightCrafter 4500) and a motorized neutral density filter wheel still required integration with Symphony. The primary requirement when

integrating these devices was to ensure that the source code could be used across different research labs. Therefore, the software was designed according to object-oriented principles, as this promotes re-usability [23].

5.1.1 Architecture

The data acquisition source code is organized in three different categories, as illustrated in Figure 5.1

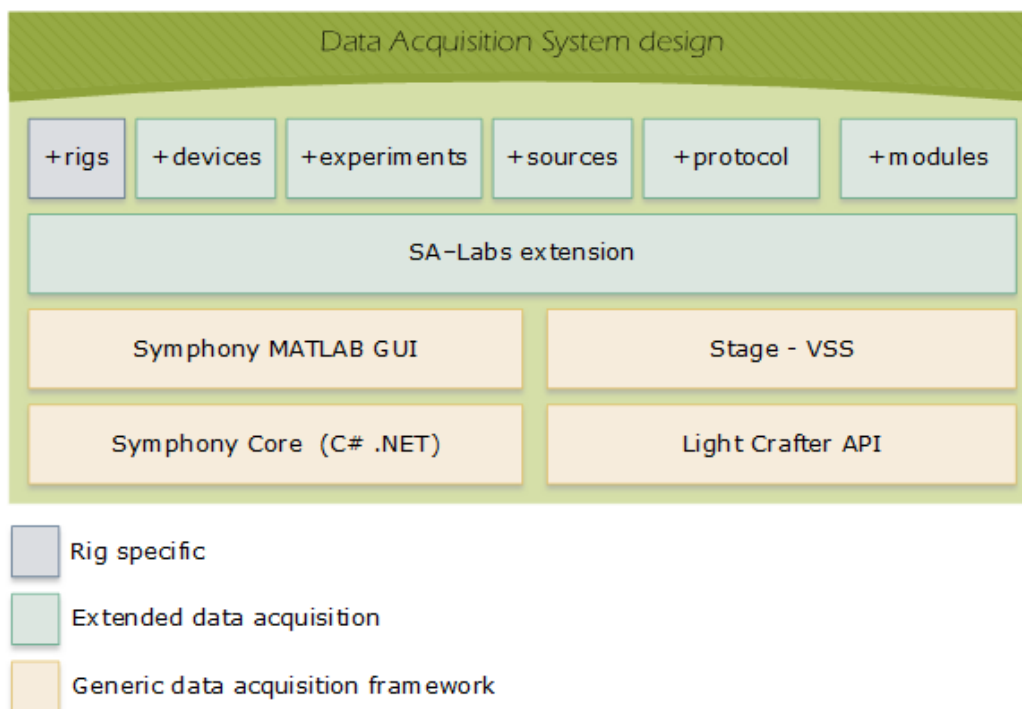


Figure 5.1: Data acquisition architecture: It features the three different levels that make up its building blocks. While the `+rigs` block is specific to every individual setup, the extended and generic data acquisition frameworks can be shared across laboratories.

- a. **Rig specific:** this building block contains the source code that is generic to each electrophysiology setup (rig) and cannot be shared between laboratories. The device specific information such as the hardware configuration, calibration parameters are available in the `Rig Description`. The `+rigs` directory bundles the available `Rig Description` and other setup specific code.

- b. **Generic data acquisition framework:** these blocks are based on the Symphony software [12] for data acquisition and the Stage-VSS software [38] for visual stimulation. Details about these toolboxes are given in the background section 2.4.1, 2.4.2.
- c. **Extended data acquisition:** it contains the source code to support external devices and the user controls for device configuration, the functionality for annotation during the experiment, the stimulus parameters and the stimulus device calibration.

The components of the extended data acquisition represent the first layer of the code architecture, together with the rig specific block (Figure 5.1). Each block `+devices`, `+experiments`, `+sources`, `+protocol`, `+modules` is organized as a Matlab package [41]. They are explained in detail in the following paragraph.

The source code to interface devices that are not built-in to the Symphony software is added to the `+devices` Matlab package. This includes Matlab software to control the DLP LightCrafter 4500 projector and, the motorized NDF wheel via the communication port (COM).

Annotation of an experiment is achieved by simply extending the `Experiment Description` class in the Symphony software. The class is present in the Matlab `+experiments` package and takes information about the purpose of the experiment, the name of the experimenter, the laboratory and the experiment date.

The information about the type of experimental animal and tissue is provided by extending the `Source Description` class in the Symphony software. The class is placed in the Matlab `+sources` package and takes information about species, genotype, the tissue condition, the type of the cell (or) cell cluster, and the recording location (Figure 5.2).

The visual input presented to the recorded neurons is defined by extending the `Stimulus Protocol` class from the Symphony software. The visual stimulus is essential for physiological recordings in the retina, as it triggers the retinal neurons to respond - generally in stimulus specific ways. The `Stimulus Protocol` properties should ideally be shared between laboratories, as this makes it possible to investigate questions with comparable experimental conditions in different setups and on different animal species. The `Stimulus Protocol` is programmed for display device like DLPs or

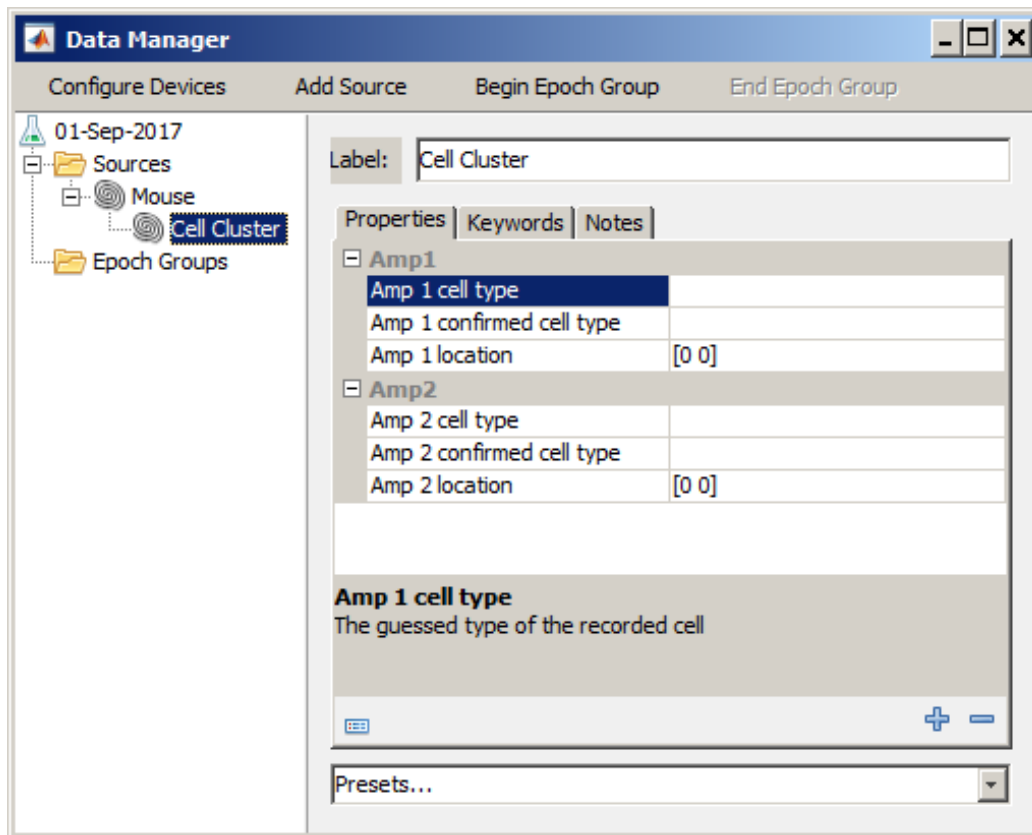


Figure 5.2: `Source Description` : It provides detailed information about the experimental animal and tissue (here labeled "Mouse") and about the recorded "Cell cluster".

LEDs, and for electrical stimulation using the amplifier input channels. It is organized in the Matlab `+protocols` package. The execution steps are described in an activity diagram [8] in the Appendix B.

To manage the configuration of the interfaced devices or to implement GUI controls that simplify performing an experiment, the `Module` class is used. `Module` can be customized to control the functions of Symphony and its extensions. The following section explains the light calibration module that manages the DLP projector and its LED, as well as calibration of the neutral density filters(NDF). All the `Module` classes are located in the Matlab `+modules` package. The guidelines for creating the `Source Description`, `Stimulus Protocol`, `Module` packages are described in the Symphony documentation [12].

5.1.2 Light calibration module

The light calibration module provides a user interface to quantify the properties of the visual stimulation. This is crucial to ensure that all experiments are performed with a defined light intensity, measured in photoisomerizations which occur in the photoreceptors when they receive light. How the photoisomerizations are calculated is described in detail in the Appendix C.

The light calibration module provides a semi-automated tool to acquire and structure the calibration measurements. It stores the calibration measurements from both spatially uniform and from structured stimulus sources such as LEDs or DLP projectors. The measurements include the stimulus intensity, light source nonlinearity, the spectrum of the light source and the attenuation factor of the neutral density filters. In addition, it compares the new calibration values with the previous history and verifies the consistency of calibration measurements. The calibration data file format ensures that the electrophysiology setup is calibrated at regular intervals (see Appendix, D).

Intensity calibration: Due to age-related changes in the light source, the absolute intensity it provides can fluctuate. The fluctuation is quantified by measuring the power output of the light source associated with a standard input to the light source (voltage or LED current) at regular intervals. If the difference to the previous measurement is not marginal (more than 3%), then the visual stimulation component of the setup is examined to ensure it still functions correctly before starting a new experiment. The newly developed Symphony `Module` (Figure 5.3) automates the intensity calibration process. It integrates the power output of a photometer (in our setup:UDT S470 flexOptometer (*Gamma Scientific, "San Diego, CA", USA*)) and stores the information along with the calibration date and additional user information.

Neutral density filter calibration: To study the retina at a low light levels requires the attenuation of the incoming light from the light source. Neutral Density Filters (NDF) are designed to attenuate light by a specific factor, which is generally measured as optical density (OD). The OD value given by the manufacturer often contains error margins that are larger than what can be tolerated for the experiments planned with this setup. NDFs therefore require calibration measurements to calculate the actual OD value of the filters, when installed in the setup.

The user interface (Figure 5.4) provides the information required for calibrating the NDFs. It can be used for both manual and motorized filter wheels.

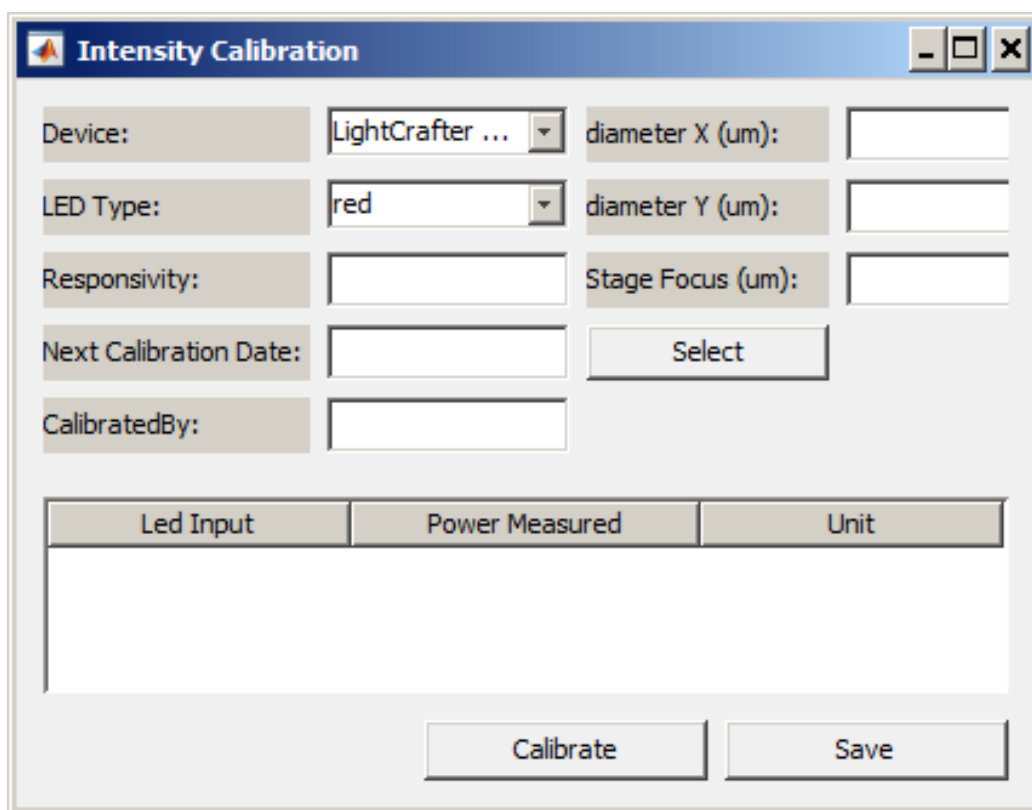


Figure 5.3: `Module` : Intensity Calibration GUI. It prompts the user for calibration specific information and measures the power output from a photometer for a given input to the light source (LED).

Nonlinearity calibration: The response of many light sources to its control input often does not follow a linear relation, especially at low driving intensities. Hence, it is important to measure and store the input/output relationship of the light source, in our case an LED. To this aim, the photometer measures the power output of the light source for a range of inputs to the light source.

The user interface for calibration is customized from the Symphony (Figure 5.5). During calibration, the photometer’s analog channel is connected to an analog input stream of the digitizer. The LED current is set before every epoch and the power measured from the photometer is stored.

Spectrum Calibration: The spectrum of the light source is another crucial component to quantify the amount of light presented to the tissue. It is measured using a spectrometer. We use the UDT S470 flexOptometer

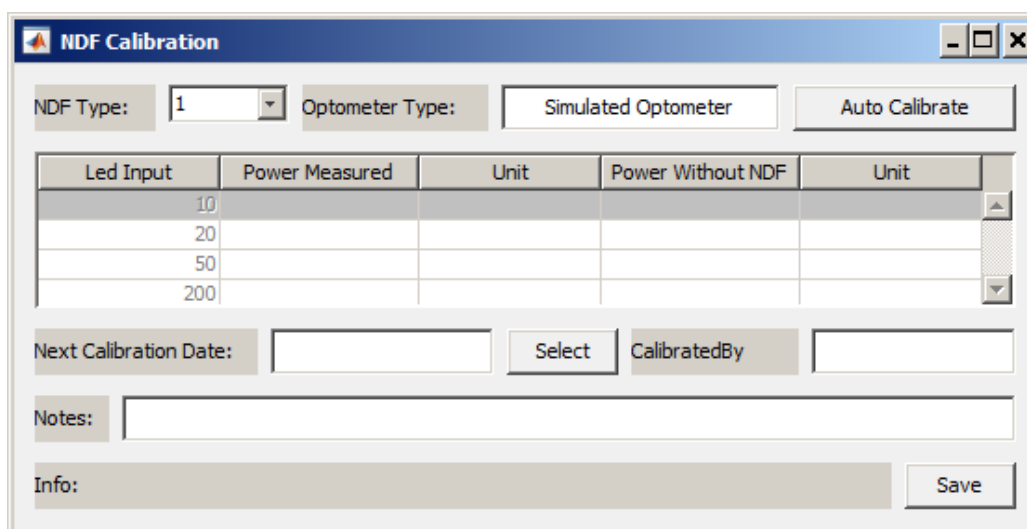


Figure 5.4: `Module` : NDF Calibration GUI. It integrates the output from a photometer that measures the power of the light source in two different configuration: one with a NDF in the optical path and one without it. It is stored for further light calibration calculations.

(*Gamma Scientific, "San Diego, CA", USA*), with its accompanying software Spectrasuite [46]. The result of the measurement is stored as a part of the calibration information. An example of a noise corrected power spectrum is shown in the Appendix C.

Once the rig is configured for the digitizer, amplifier, stimulation device-sand protocols, and when it has been calibrated; it is ready for the start of the experiment. A brief description of the steps involved in the experimental recordings using Stage-VSS and Symphony is given in the Appendix B. Using Symphony, acquired data can be analysed during the experiment, and visualized in near real time. More about the analysis is explained in the next section.

5.2 Data analysis

The data analysis can be separated into an offline and an online component, based on the time when it is performed. The offline analysis is performed after the experiment, while online analysis is performed during the experiment.

The Symphony software stores the raw data in a hierarchical description file (HDF5) format [27]. In order to extract the raw data, including stimulus

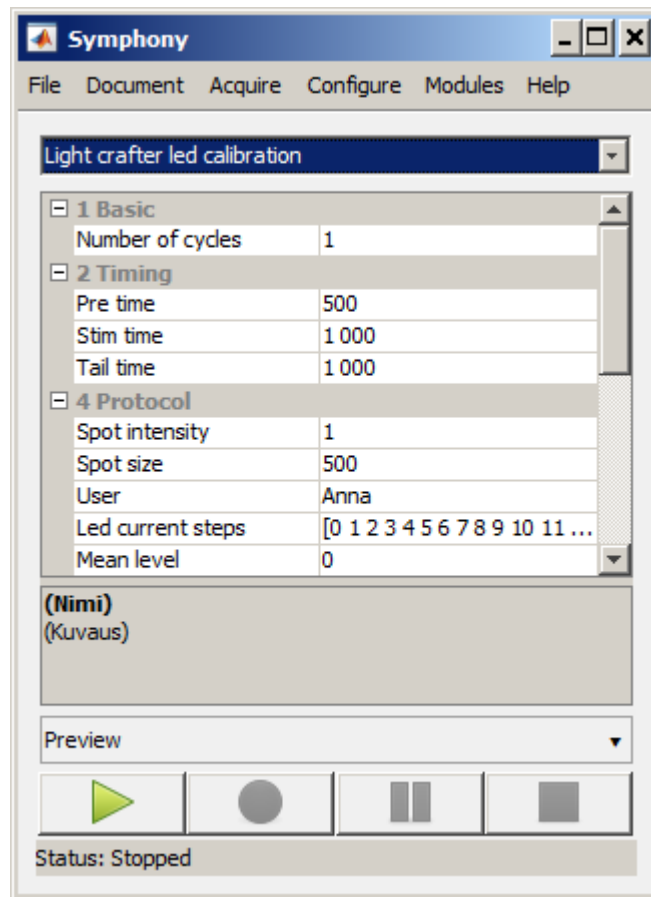


Figure 5.5: `Module`: Nonlinearity calibration GUI. A photometer measures the power output of the light source (LED) for a range of input values and quantifies the non-linear relationship between the LED input and output.

information and other metadata, the HDF5 file needs to be parsed. From this parsed information, data can be analyzed and visualized. During the whole process, the analysis code has access to the entire experimental data. As a result, offline analysis enables data processing without great time or space constraints to the analysis computer.

Online analysis, on the other hand, analyses the data streams at the end of each epoch during the experiment. Therefore, the complete dataset is not known in advance, which constrains data curation and extracting new features. In addition, since online analysis needs to occur in near real-time, to not delay the experiment, it has higher demands on computing resources on the acquisition computer. Therefore, online analysis is generally not as comprehensive as analysis offline. It is rather used as a tool to give a brief insight

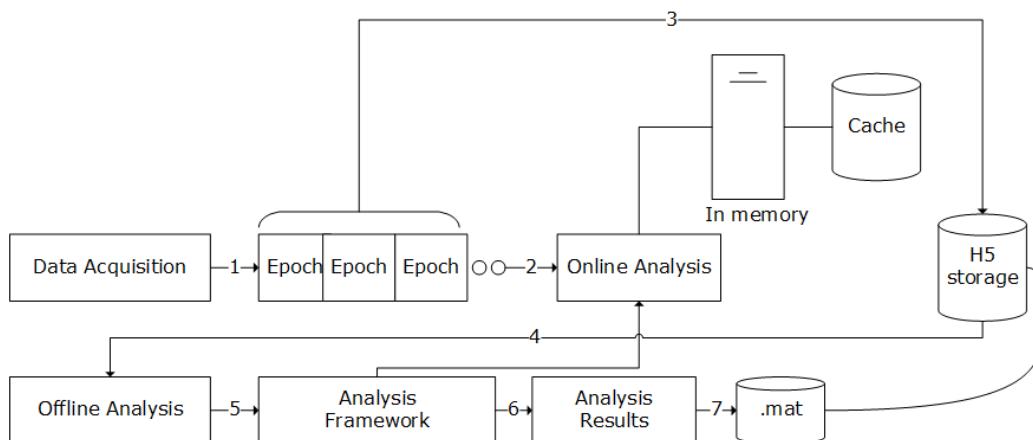


Figure 5.6: Flow diagram for data from the acquisition system to the analysis framework. As the epochs are saved as an HDF5 file by the data acquisition system, they are simultaneously processed through online analysis. During offline analysis, the HDF5 (.H5 file) is parsed through the analysis framework and the result is saved as .mat file.

into the experimental results, which improves experimental productivity and strategy.

An overview of workflow between online and offline analysis is shown in Figure 5.6. Details about the analysis framework are given in the next section.

5.2.1 Analysis framework

The analysis framework toolbox abstracts the data structure and the storage hierarchy of the analysis results and provides simplified Matlab functions. The framework captures the necessary information about the parameters required for analysis; about the source code to build the analysis and about the raw data. As a result, it is easy to share and replicate the analysis with minimal effort across laboratories. In addition, it can seamlessly integrate with Symphony and can therefore be used to perform online analysis with the same logical framework.

The analysis design focuses on three main principles [39],

- **Single Responsibility Principle:** The source code is written as a Matlab class or function featuring specific logical objects. For example, `CellData` is the Matlab class which manages cell-specific information and contains all epochs recorded from a particular cell or a cell cluster. Each epoch is managed in the class `EpochData`, which contains

information about the `Stimulus Protocol` , as well as a link to the raw data.

- **Open Closed Principle:** A new behavior can be added by creating a new class or function rather than modifying the old code.
- **Dependency Inversion Principle:** High-level components such as the data structure or storage are not dependent on the low-level functions which are used to perform the analysis. For example, the analysis results are grouped hierarchically in the form of a tree data structure [14]. This tree structure is used to organize and store the analysis results. It is not dependent on the functions that are used to analyze the raw data.

The following paragraphs briefly describe the steps involved in performing an analysis,

Data parser & curation: Symphony stores the raw data in the highly compressed HDF5 format. It contains the `Stimulus Protocol` properties, the experimental annotation, and the recording raw data from the amplifier channels. However, searching and adding tags to the compressed data requires more time than doing so in a less compressed format. To speed up the search, a secondary index to the HDF5 file is created using the data parser. It extracts the `Stimulus Protocol` properties, the experimental tags and creates pointers to the amplifier channel data. The result is saved as a `CellData` ".mat" file.

Each individual epoch that is stored in the `CellData` file has to be examined for recording quality. Unwanted epochs are excluded from further analysis (to this aim, I have previously developed a Data Curator with a GUI [44]). The analysis framework handles these steps along with data preprocessing. Preprocessing includes features such as spike detection in cell-attached recordings, automated tagging of `EpochData` and deriving new parameters such as photoisomerization values from the `Stimulus Protocol`

Creating an analysis project: Data organization is a key step in every research project [70]. It is highly dependent on the system that was used to generate and analyze the data. The analysis framework provides this data organization of the parsed `CellData` and the intermediate analyses results. Each analysis is stored as a text file in the Java Script Object Notation

(JSON) format [9]. JSON is an open standard format consisting of property-value pairs. It is a simple and highly flexible way to organize the data analysis so that it is shareable between research group members and even across laboratories. An example JSON project file is shown in Appendix E.1.

Building the analysis: Data analysis commonly involves filtering the data for a particular condition, and applying a mathematical function to extract the relevant features. The filtering conditions depends on the particular dataset. In the case of Symphony data, it depends on the `Stimulus Protocol` properties. However, `Stimulus Protocol` properties depend on the type of experiment. Therefore, each user of the analysis framework defines their own filters and attaches their analysis functions to them. An example analysis filter definition is presented in Appendix E.2

The filter definition is often hierarchical in structure (as multiple levels of parameters can be nested, such as selecting a particular `Stimulus Protocol` : `Light Step` , with a particular light intensity: `Intensity` and stimulus length: `Stim Time`). Therefore, the filter results are stored as a N-array tree [17]. The N-array tree is a data structure to represent hierarchical groups [61]. The hierarchical group includes the list of epochs that match the filter criteria, as well as the analysis results which were extracted using a specific computational function.

The analysis results are described as `Features` in the analysis framework. `Features` are results of a mathematical transformation or a statistical summary of a group of epochs. The Matlab function that is used to derive the features is called a `Feature Extractor` . Appendix E.3 shows an example of how to build the analysis.

Visualizing the analysis result: Data analysis results are best assessed using a visualization. Thus, the results from the analysis should be easily plottable in Matlab. Querying the relevant analysis results from the N-array tree requires tree traversal methods [14] and the MatlabQuery toolbox [10]. MatlabQuery eliminates the need for looping through the Matlab arrays or cell objects and performs simple data selection or transformation. The data obtained with the query processor can then be visualized using the Matlab plotting functionality.

The next section describes the management system for raw data and analysis results.

5.3 Data Management

A database system is required to search, validate and replicate the analysis results. Such database systems are constantly developing based on the advent of modern algorithms and storage architecture [16]. As a result, a standard interface is required to adapt to the changing data management system. In addition, any change in the data management system should not affect the analysis steps explained in the previous section.

The analysis framework provides a simple and easy-to-use data management interface. Currently, a directory-based data management system is implemented. It saves the Matlab objects as ".mat" files in the specified directory. Depending on the scale of the data, this directory-based data management system can be migrated to relational database systems (RDBMS) such as Data Joint [74] without affecting the existing analysis workflow.

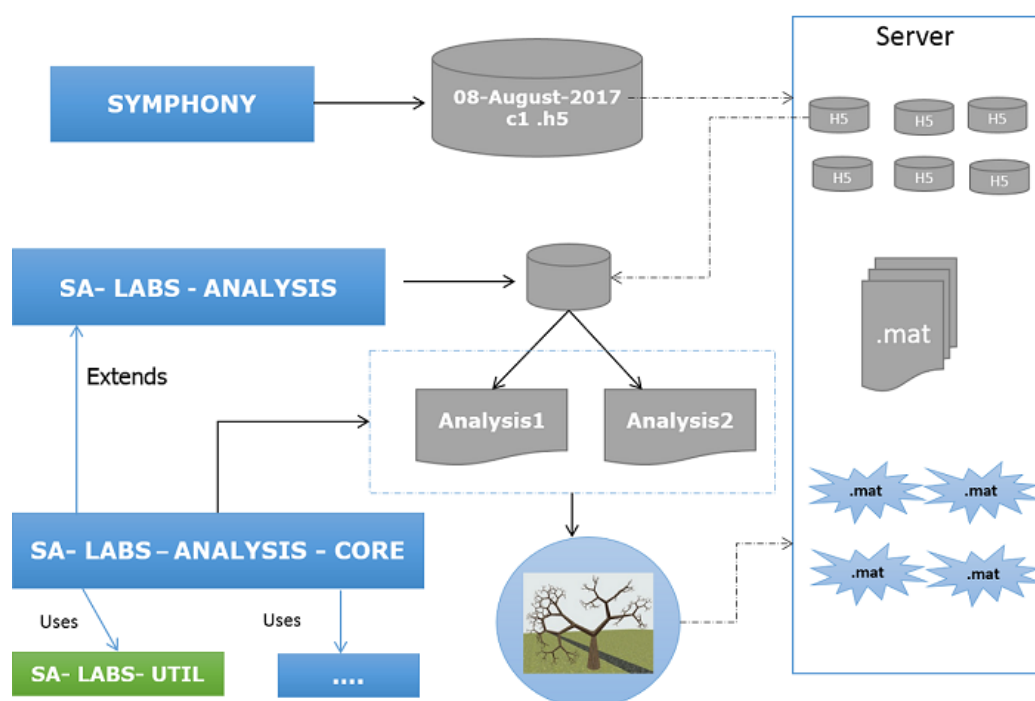
Data synchronization

To ease the data sharing among members of the research group, as well as between collaborating laboratories, data synchronization is required. To achieve this, the raw data acquired by Symphony is manually copied to a file server and organized according to experiment date. During analysis, the users copy the raw data to their local system directory and work on it. After completion, the analysis results directory is copied back to the server for further data sharing (Figure 5.7).

5.4 Software Maintenance

A professional software build cycle [21] is implemented to ensure that the existing acquisition and analysis software functions without flaws. The entire source code for acquisition and analysis is maintained in a GitHub Organization - Schwartz-AlaLaurila-Labs (<https://github.com/Schwartz-AlaLaurila-Labs/>). In addition, the external library for managing the data acquisition and data analysis framework is managed by ToolboxToolbox [28] - a Matlab based declarative dependency management tool.

When a developer synchronizes a change in the source code to the GitHub server, Aalto University's continuous integration server - Jenkins (<https://build.nbe.aalto.fi/>) is notified. Jenkins [62] then builds the source code, executes test cases, and notifies the lab users about changes in the code. Once the code changes are reviewed and passed the quality checks, they are synchronized to the Test Setup. Regression testing is performed in the Test



copy the raw data to their local system directory and work on it. After

Figure 5.7: Data Flow Diagram: The raw data from Symphony is synchronized to the server. The analysis framework (Sa-Labs-Analysis) parses the Symphony data and creates a secondary index ".mat" `CellData` file. This secondary index is then used to build the analysis; to store the results as a Matlab tree structure [14] and to synchronize the analysis results with the server.

Setup. After the validation of the changes, the source code is synchronized to the experimental setup computer. The whole process is illustrated in Figure 5.8.

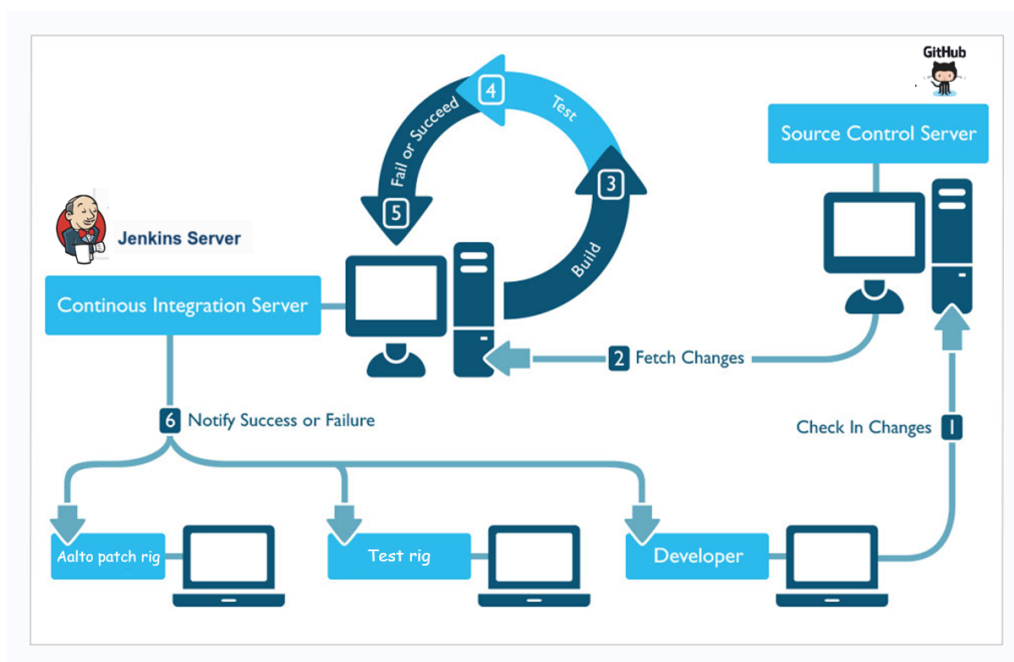


Figure 5.8: Source code integration between the developer, the Test Rig, and the Experimental Rig environment using GitHub and Jenkins.

Chapter 6

Discussion

In this section, we compare our electrophysiology setup and custom-written software with other state-of-the-art multi-electrode patch clamp setups, as well as open source acquisition softwares, and evaluate the advantages and improvements.

Electrophysiology setup

Our setup provides a novel technique to investigate information processing in the neural circuits of the retina. There are a number of other efforts to establish multi-electrode patch clamp setups in the world, but they all work on brain slice preparations [34, 69]. While cell types can also be targeted and interactions between neurons can be investigated, brain slice preparations lack one crucial aspect that the retina provides: the activation of neurons with their natural stimulus. Light stimulation in the retina can activate particular subsets of retinal ganglion cells in a naturalistic, and stimulus specific manner [4]. This allows to study particular retinal circuits, and their functional role in visual processing, which is not only important to better understand the visual sense, but also to understand how neural circuits encode information that is relevant for the natural behavior of the organisms.

There are some recent developments in the automation of patch clamping that could further improve our setup and make experiments more efficient. A computer assisted multi-electrode patch clamp setup [48] performs the majority of the logistical tasks of moving and placing the up to 12 electrodes automatically. While our system is using the Scientifica “follow control”, which automatically moves the electrodes in synchrony with the position of the stage, all other electrode maneuvers have to be conducted manually. Therefore, expanding our setup with a more automatic control of the elec-

trode position might increase experimental productivity even further.

In recent years, not only the electrode positioning, but the entire patch clamp procedure, has been automated [33], recently even including visual targeting of cells [2, 64]. However, this automation has only been tested in brain slide preparations, and the applicability to the retina is not established yet. Patching cells in the retina comes with additional challenges, such as removing the inner limiting membrane and cleaning the tissue before a cell can be patched. In addition, the entire experimental procedure happens in a complete dark room and makes the automated patching more difficult. Therefore, developing automated tools that aid the experimentalist, rather than completely take over control of the procedures, might be more favorable for patch clamp recordings in the retina. Examples for such semi-automated tools are a 3D-visualizing system for maneuvering the electrodes, as well as machine learning classifier for real-time cell type identification based on morphology and physiological responses.

One other area where our multi-electrode patch clamp setup could be upgraded is imaging. With the advent of retinal transcriptomics, genetic tools are available to label specific cell types in the retina with fluorescent markers [59]. This provides a unique opportunity to record from a group of genetically labeled neurons. The great advantage of this technique would be that cell types would not need to be manually tested, as is the case now, but could be selected by their fluorescent markers. This would make experiments much more efficient.

Software

As the scale of the acquired electrophysiology data grows, memory intensive operations can be delegated to different computer nodes in parallel. The analysis framework can be extended to handle distributed data processing for large datasets. It will then be possible to execute the analysis and synchronize the results in a distributed computing environment with access to the data and to Matlab.

While working in a manner that is straightforward and easy to control, visualizing the analysis results using the existing Matlab GUI is rather time intensive, especially for online analysis. As the size of the datasets, or the complexity of the online analysis increase, faster solutions for data visualization might be required. Browser-based analytics dashboards such as Grafana and Plotly [50] are considerably faster at plotting comparable data [49, 73]. Integrating the analysis framework with these platforms could speed up data visualization considerably in the future.

Any future extensions and improvements of the acquisition and analysis frameworks are helped by the fact that they are freely available under open source licenses. The acquisition software Symphony already has attracted a large active visual neuroscience community that extends and improves it, compared to other open source electrophysiology options [13, 60, 65]. In addition, this community is recruited primarily from retina research labs. Recruiting this community to the software solutions presented in this thesis, and remaining engaged in future open source developments, provides an ideal environment to improve the current tools for retina research, and developing new solutions in the future.

Chapter 7

Conclusion

We built a state-of-the-art multi-electrode patch clamp system in the Alalaurila laboratory. This pioneering technology allows to mechanistically dissect neural processing in small neural networks. It is one of only a few setups of its kind world wide, and the first one that is designed to study the vertebrate retina.

In addition, I newly developed software that allows to perform and reproduce comparable experiments across different electrophysiology setups in the same lab and in collaborating research labs. The analysis software ensures that data analysis results are comparable and repeatable across laboratories. The entire system remains flexible to be extended and updated with additional components and software tools.

Together, the newly developed setup and software tools enable mechanistic studies of neural processing in small retinal networks. For example, the role of correlated activity can be studied for the first time at the level of input currents in complete functional populations of retinal ganglion cells. Moreover, the four electrodes make it possible to study the flow of information through the retina across all functional layers.

Bibliography

- [1] AKAY, M., SAJDA, P., MICERA, S., AND CARMENA, J. M. Advanced technologies for brain research [scanning the issue]. *Proceedings of the IEEE* 105, 1 (Jan 2017), 8–10.
- [2] ANNECCHINO, L. A., MORRIS, A. R., COPELAND, C. S., AGABI, O. E., CHADDERTON, P., AND SCHULTZ, S. R. Robotic automation of in vivo two-photon targeted whole-cell patch-clamp electrophysiology. *Neuron* 95, 5 (2017), 1048–1055.
- [3] AVERBECK, B. B., LATHAM, P. E., AND POUGET, A. Neural correlations, population coding and computation. *Nature Reviews. Neuroscience* 7, 5 (May 2006), 358–366.
- [4] BADEN, T., BERENS, P., FRANKE, K., ROSÓN, M. R., BETHGE, M., AND EULER, T. The functional diversity of retinal ganglion cells in the mouse. *Nature* 529, 7586 (Jan 21, 2016), 345–350.
- [5] BAE, J. A., MU, S., KIM, J. S., TURNER, N. L., TARTAVULL, I., KEMNITZ, N., JORDAN, C. S., NORTON, A. D., SILVERSMITH, W. M., PRENTKI, R., ET AL. Digital museum of retinal ganglion cells with dense anatomy and physiology. *bioRxiv* (2018), 182758.
- [6] BEHRENS, C., SCHUBERT, T., HAVERKAMP, S., EULER, T., AND BERENS, P. Connectivity map of bipolar cells and photoreceptors in the mouse retina. *Elife* 5 (2016).
- [7] BLOOMFIELD, S. Retinal amacrine cells. In *Encyclopedia of Neuroscience*, L. R. Squire, Ed. Academic Press, Oxford, 2009, pp. 171 – 179.
- [8] BOOCH, G., RUMBAUGH, J., AND JACOBSON, I. The unified modeling language user guide. The Addison-Wesley object technology series. Pearson Education, 1999, ch. 8 Activity View, pp. 95–99.

- [9] BRAY, T. The javascript object notation (json) data interchange format. <https://tools.ietf.org/html/rfc8259>, 2017.
- [10] BRIAN, L. Linq -like queries in matlab. <https://github.com/brian-lau/MatlabQuery>, 2015.
- [11] BULLOCK, T. H. *Neural Circuits and Behavior*. Humana Press, Totowa, NJ, 2005, pp. 121–127.
- [12] CAFARO, M., MIDGLEY, F., NEWARD, T., AND WARK, B. Symphony data acquisition system (<http://symphony-das.github.io/>), 2017.
- [13] CAMPAGNOLA, L., KRATZ, M. B., AND MANIS, P. B. Acq4: an open-source software platform for data acquisition and analysis in neurophysiology research. *Frontiers in Neuroinformatics* 8 (2014).
- [14] CAMPBELL ROB, T. J.-Y. A matlab class to represent the tree data structure. <https://github.com/raacampbell/matlab-tree>, 2017.
- [15] CARANDINI, M. From circuits to behavior: a bridge too far? *Nature neuroscience* 15, 4 (2012), 507–509.
- [16] CHEN, Y., YI WANG, Z., YUAN, G., AND HUANG, L. An overview of online based platforms for sharing and analyzing electrophysiology data from big data perspective. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 4 (July 1, 2017), n/a.
- [17] CORMEN, T., LEISERSON, C., RIVEST, R., AND STEIN, C. *Introduction To Algorithms*. MIT electrical engineering and computer science series. MIT Press, 2001.
- [18] DAVIE, J. T., KOLE, M. H. P., LETZKUS, J. J., RANCZ, E. A., SPRUSTON, N., STUART, G. J., AND HÄUSSER, M. Dendritic patch-clamp recording. *Nature Protocols* 1, 3 (September 2006), 1235–1247.
- [19] DEVICES, M. Axon multiclamp 700b commander. http://mdc.custhelp.com/app/answers/detail/a_id/20059/related/1, Aug 2016. v 2.2.0.
- [20] DOWLING, J. E. *The retina: an approachable part of the brain*. Belknap press of Harvard University press Cambridge, MA, 2012.
- [21] DUVALL, P., MATYAS, S. M., AND GLOVER, A. *Continuous Integration: Improving Software Quality and Reducing Risk (The Addison-Wesley Signature Series)*. Addison-Wesley Professional, 2007.

- [22] FOX, M. D., AND RAICHLER, M. E. Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nature Reviews Neuroscience* 8, 9 (2007), 700–711.
- [23] GAMMA, E., HELM, R., JOHNSON, R., AND VLISSIDES, J. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [24] GIBSON, F., OVERTON, P. G., SMULDERS, T. V., SCHULTZ, S. R., EGLER, S. J., INGRAM, C. D., PANZERI, S., BREAR, P., WHITTINGTON, M., SERNAGOR, E., CUNNINGHAM, M., ADAMS, C., ECHTERMEYER, C., SIMONOTTO, J., KAISER, M., SWAN, D. C., FLETCHER, M., AND LORD, P. Minimum information about a neuroscience investigation (mini): Electrophysiology. Tech. rep., Nature Precedings, Apr 9, 2009.
- [25] GOLLISCH, T., AND MEISTER, M. Eye smarter than scientists believed: Neural computations in circuits of the retina. *Neuron* 65, 2 (-01-28 2010), 150–164.
- [26] GOVARDOVSKII, V. I., FYHRQUIST, N., REUTER, T., KUZMIN, D. G., AND DONNER, K. In search of the visual pigment template. *Visual neuroscience* 17, 4 (2000), 509–528.
- [27] GROUP, T. H. Hierarchical data format version 5. <http://www.hdfgroup.org/HDF5>, 2000- 2017.
- [28] HEASLY, B. Toolboxtoolbox. <https://github.com/ToolboxHub/ToolboxToolbox>, 2017.
- [29] INSTRUMENTS, A. Axon conventional patch clamp, -03-20T15:04:45-07:00 2014.
- [30] JONES, E. G. Colgi, cajal and the neuron doctrine. *Journal of the History of the Neurosciences* 8, 2 (Aug 1999), 170–178.
- [31] JORGENSON, L. A., NEWSOME, W. T., ANDERSON, D. J., BARGMANN, C. I., BROWN, E. N., DEISSEROTH, K., DONOGHUE, J. P., HUDSON, K. L., LING, G. S. F., MACLEISH, P. R., MARDER, E., NORMANN, R. A., SANES, J. R., SCHNITZER, M. J., SEJNOWSKI, T. J., TANK, D. W., TSIEN, R. Y., UGURBIL, K., AND WINGFIELD, J. C. The brain initiative: developing technology to catalyse neuroscience discovery. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 370, 1668 (May 19, 2015).

- [32] KOCH, C., AND REID, R. C. Neuroscience: Observatories of the mind, -03-21 2012.
- [33] KODANDARAMAIAH, S. B., FRANZESI, G. T., CHOW, B. Y., BOYDEN, E. S., AND FOREST, C. R. Automated whole-cell patch-clamp electrophysiology of neurons in vivo. *Nature methods* 9, 6 (2012), 585–587.
- [34] LALANNE, T., ABRAHAMSSON, T., AND SJÖSTRÖM, P. J. Using multiple whole-cell recordings to study spike-timing-dependent plasticity in acute neocortical slices. *Cold Spring Harbor Protocols* 2016, 6 (Jun 01, 2016), pdb.prot091306.
- [35] LANDAU, H. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE* 55, 10 (1967), 1701–1706.
- [36] LONDON, A., BENHAR, I., AND SCHWARTZ, M. The retina as a window to the brain—from eye research to cns disorders. *Nature Reviews. Neurology* 9, 1 (Jan 2013), 44–53.
- [37] LUO, L. *Principles of Neurobiology*. Garland Science, 2015.
- [38] MARK, C. Stage visual stimulus system (<http://stage-vss.github.io/>), 2017.
- [39] MARTIN, R. C. Agile software development: principles, patterns, and practices. Prentice Hall, 2002, ch. 8,9,11.
- [40] MASLAND, R. The neuronal organization of the retina. *Neuron* 76, 2 (October 18, 2012), 266–280.
- [41] MATLAB. R2017a https://in.mathworks.com/help/matlab/matlab_oop/scoping-classes-with-packages.html, 2017.
- [42] MIT. The mit license. <https://opensource.org/licenses/MIT>.
- [43] MURPHY, G. J., AND RIEKE, F. Electrical synaptic input to ganglion cells underlies differences in the output and absolute sensitivity of parallel retinal circuits. *Journal of Neuroscience* 31, 34 (2011), 12218–12228.
- [44] NARAYANAN, S. sa-labs-analysis. <https://github.com/Schwartz-AlaLaurila-Labs/sa-labs-analysis>, 2017.
- [45] NATHAN, A. *.NET and COM: The Complete Interoperability Guide*. Sams White Book. Pearson Education, 2002.

- [46] OPTICS, O. Spectrasuite software. <https://oceanoptics.com/support/technical-documents/spectrasuite/>, 02 2010.
- [47] PANZERI, S., MACKE, J. H., GROSS, J., AND KAYSER, C. Neural population coding: combining insights from microscopic and mass signals. *Trends in cognitive sciences* 19, 3 (Mar 2015), 162–172.
- [48] PERIN, R., AND MARKRAM, H. A computer-assisted multi-electrode patch-clamp system. *Journal of Visualized Experiments: JoVE*, 80 (Oct 18, 2013), e50630.
- [49] PLOTLY. Plotly — benchmarks. <https://plot.ly/benchmarks/>, 2017.
- [50] PLOTLY. Plotly dash. <https://github.com/plotly/dash>, 2017.
- [51] PURVES, D. Neuroscience. Sinauer Associates Incorporated, 2004, ch. 4, p. 70.
- [52] PURVES, D., AUGUSTINE, G. J., FITZPATRICK, D., KATZ, L. C., LAMANTIA, A.-S., MCNAMARA, J. O., AND WILLIAMS, S. M. *The Retina*, neuroscience, 2nd edition. ed. Sinauer Associates, 2001.
- [53] SAKMANN, B., AND NEHER, E. Patch clamp techniques for studying ionic channels in excitable membranes. *Annual review of physiology* 46, 1 (1984), 455–472.
- [54] SCANZIANI, M., AND HAUSSER, M. Electrophysiology in the age of light. *Nature* 461, 7266 (October 15 2009), 930–939. LR: 20170220; GR: Howard Hughes Medical Institute/United States; GR: Wellcome Trust/United Kingdom; JID: 0410462; SY7Q814VUP (Calcium); RF: 100; ppublish.
- [55] SCHMOLESKY, M. Webvision: The organization of the retina and visual system [internet] figure 8, [visual input to the brain]. <https://www.ncbi.nlm.nih.gov/books/NBK11524/figure/ch31visualcortex.F8/>, /06/14 2007.
- [56] SCHWARTZLAB, N. U. Symphony analysis. <https://github.com/SchwartzNU/SymphonyAnalysis>, 2017.
- [57] SCIENTIFICA. Linlab2. <http://www.scientifica.uk.com/products/scientifica-linlab-2>, 07 2016. v1.6.78.

- [58] SEGEV, R., GOODHOUSE, J., PUCHALLA, J., AND BERRY, M. J. Recording spikes from a large fraction of the ganglion cells in a retinal patch. *Nature Neuroscience; New York* 7, 10 (Oct 2004), 1154–61.
- [59] SHEKHAR, K., LAPAN, S. W., WHITNEY, I. E., TRAN, N. M., MACOSKO, E. Z., KOWALCZYK, M., ADICONIS, X., LEVIN, J. Z., NEMESH, J., GOLDMAN, M., MCCARROLL, S. A., CEPKO, C. L., REGEV, A., AND SANES, J. R. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell* 166, 5 (2016), 1308 – 1323.e30.
- [60] SIEGLE, J. H., LÓPEZ, A. C., PATEL, Y., ABRAMOV, K., OHAYON, S., AND VOIGTS, J. Open ephys: An open-source, plugin-based platform for multichannel electrophysiology. *Journal of Neural Engineering* (Feb 07, 2017).
- [61] SKLYAROV, V., AND SKLIAROVA, I. High-performance data processing over n-ary trees. In *High-Performance Computing Using FPGAs*. Springer, 2013, pp. 245–277.
- [62] SMART, J. *Jenkins: The Definitive Guide: Continuous Integration for the Masses*. O’Reilly Media, 2011.
- [63] SMITH, P. *Client server computing*. SAMS, Carmel, Ind, 1992.
- [64] SUK, H.-J., VAN WELIE, I., KODANDARAMAIAH, S. B., ALLEN, B., FOREST, C. R., AND BOYDEN, E. S. Closed-loop real-time imaging enables fully automated cell-targeted patch-clamp neural recording in vivo. *Neuron* 95, 5 (2017), 1037–1047.
- [65] SUTER, B. A., O’CONNOR, T., IYER, V., PETREANU, L., HOOKS, B. M., KIRITANI, T., SVOBODA, K., AND SHEPHERD, G. M. G. Ephys: Multipurpose data acquisition software for neuroscience experiments. *Frontiers in Neural Circuits* 4 (2010).
- [66] TURNER, M., AND RIEKE, F. Synaptic rectification controls nonlinear spatial integration of natural visual inputs. *Neuron* 90, 6 (June 15, 2016), 1257–1271.
- [67] VEITINGER, S. The patch-clamp technique. <http://www.leica-microsystems.com/science-lab/the-patch-clamp-technique/>, /11/09 2011.

- [68] VERKHRATSKY, A., AND PARPURA, V. History of electrophysiology and the patch clamp. *Methods in Molecular Biology (Clifton, N.J.) 1183* (2014), 1–19.
- [69] WANG, G., WYSKIEL, D. R., YANG, W., WANG, Y., MILBERN, L. C., LALANNE, T., JIANG, X., SHEN, Y., SUN, Q.-Q., AND ZHU, J. J. An optogenetics- and imaging-assisted simultaneous multiple patch-clamp recording system for decoding complex neural circuits. *Nature Protocols 10*, 3 (Mar 2015), 397–412.
- [70] WILKINSON, M. D., DUMONTIER, M., AALBERSBERG, I. J., APPLETON, G., AXTON, M., BAAK, A., BLOMBERG, N., BOITEN, J.-W., DA SILVA SANTOS, L. B., BOURNE, P. E., ET AL. The fair guiding principles for scientific data management and stewardship. *Scientific data 3* (2016), 160018.
- [71] WILMS, C., JACOBY, J., AND SCHWARTZ, G. Scimethods: Visual stimulation of retinal explants on a standard multiphoton microscope, 2017.
- [72] WOO, M., NEIDER, J., DAVIS, T., AND SHREINER, D. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*, 3rd ed. Addison-Wesley Longman Publishing Co., Inc.
- [73] YAIR, A. Plot performance — undocumented matlab. <https://undocumentedmatlab.com/blog/plot-performance>.
- [74] YATSENKO, D., REIMER, J., ECKER, A. S., WALKER, E. Y., SINZ, F., BERENS, P., HOENSELAAR, A., COTTON, R. J., SIAPAS, A. S., AND TOLIAS, A. S. Datajoint: managing big scientific data using matlab or python. *bioRxiv* (2015).

Appendix A

Equipments List

- 1 Lambda TLED (*Sutter Instrument, "Novato, California", USA*)
- 2 WAT-910HX (*Watec France, Maine et Loire, France*)
- 3 Olympus PLN10X/0.25 (*Olympus, Tokyo, Japan*)
- 4 Olympus LMPLFLN 60XW (*Olympus, Tokyo, Japan*)
- 5 SliceScope Motorized objective Changer (MOC) (*Scientifca, East Sussex, UK*)
- 6 Motorized Movable Top Plate (MMTP) (*Scientifca, East Sussex, UK*)
- 7 SliceScope (*Scientifca, East Sussex, UK*)
- 8 Microstar with control cubes (*Scientifca, East Sussex, UK*)
- 9 SmartTable OTS-UT2 (*Newport, "Irvine, California", USA*)
- 10 DLP LightCrafter 4500 (*Texas Instruments, "Dallas, Texas", USA*)
- 11 FW102C - Six-Position Motorized Filter Wheel (*Thorlabs Inc, "Newton, New Jersey", USA*)
- 12 MultiClamp 700B Amplifier (*Molecular Devices, "Sunnyvale, California", USA*)
- 13 MultiClamp 700B Headstage CV-7B (*Molecular Devices, "Sunnyvale, California", USA*)
- 14 900CT (*Frequency Devices, "Ottawa, IL", USA*)
- 15 ITC-1600 (*InstruTECH, New York, USA*)
- 16 NI-DAQmx (*National Instruments, "Austin, TX", USA*)

- 17 UDT S470 flexOptometer (*Gamma Scientific, "San Diego, CA", USA*)
- 18 JAZ Spectrometer (*Ocean Optics, "Dunedin, Florida", USA*)

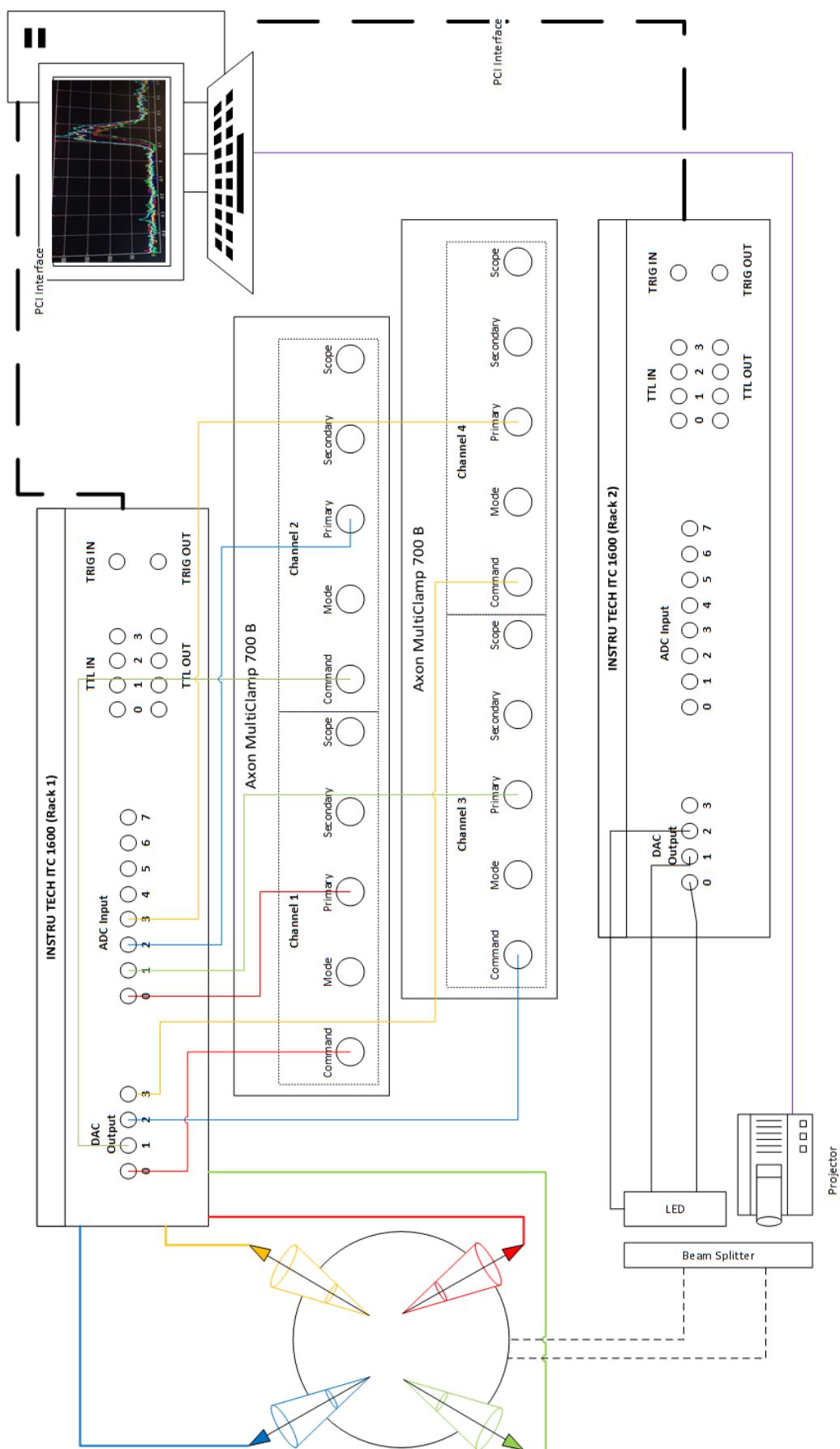


Figure A.1: Wiring diagram of the ITC 1600 digitizer and the Multiclamp 700 B amplifier channels.

Appendix B

Acquisition Steps

As a first step, start the stimulus software Stage-VSS by selecting the desired display configuration, including the type of monitor and its resolution. Once the Stage-VSS is setup and ready, start the Symphony.

As a next step, Symphony prompts for selecting a rig configuration, which is packaged in `+rigs`. On clicking `initialize`, Symphony connects with the configured hardware and opens the acquisition window, which lists all the available `Stimulus Protocols` (these are packaged in `+protocols`). Before starting a recording, a location for saving the recorded data needs to be provided using the `Data Manager` window. The user can then change the `Stimulus Protocol` parameters and either press `play` (without recording) or `record`. The `play/record` cycle of the `Stimulus Protocol` will continue as long as the experiment continues. The data recorded from each stimulus repetition is called an `Epoch`. Figure B.1 describes the activity diagram of the acquisition controller.

As a final step, the saved data will be moved to the data management server for further analysis and sharing.

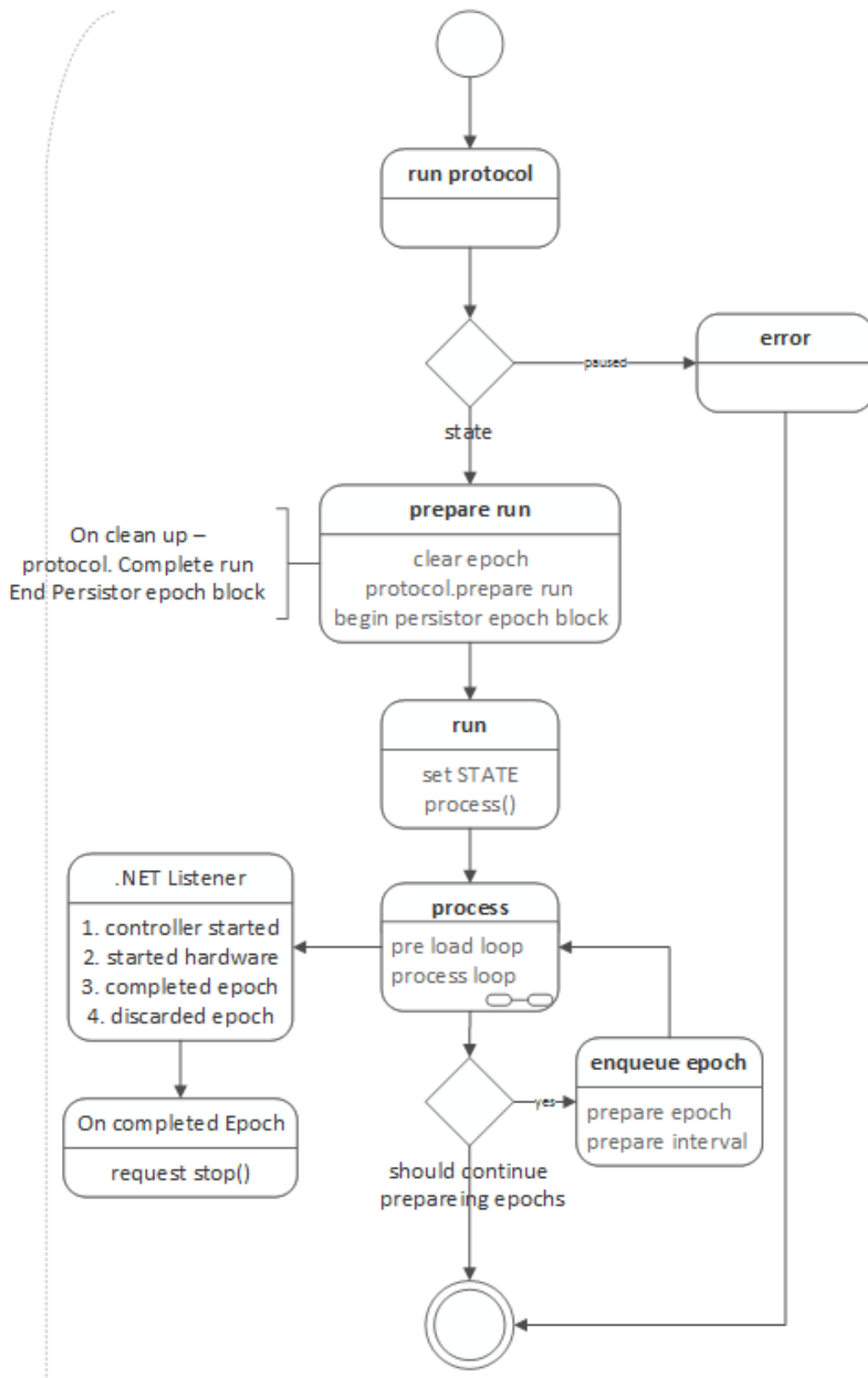


Figure B.1: Activity diagram of the Acquisition Controller along with the Stimulus Protocol execution steps.

Appendix C

Photoisomerization calculations

The stimulation of the retina is performed using a light source such as an LED (or) a DLP projector. The light from the external device enters the retina and activates the photoreceptors. The photoreceptor activation is measured as photoisomerization per rod per second $R^*/\text{rod}/\text{sec}$. Photoisomerization is calculated based on three different parameters: a) the effective photon flux of the stimulus device at the location of the retina b) the match between the photoreceptor absorbance spectrum and the spectrum of the light source and c) the photoreceptor collecting area for light.

Effective Photon flux (ϕ) of the stimulus device: The effective photon flux ϕ ($\text{photons} * \text{sec}^{-1} * \text{m}^{-2}$) of the stimulus device is the effective number of photons per second per unit area. It can be calculated using irradiance of the light source (Figure C.1) measured with the JAZ Spectrometer (*Ocean Optics, "Dunedin, Florida", USA*). The photon flux (ϕ) is calculated as follows,

The measured irradiance of the light source is, $P(\lambda)$ ($\text{Joules} * \text{sec}^{-1} * \text{m}^{-2}$) and Energy per photon is E ($\text{Joules} * \text{photon}^{-1}$)

$$E = h * c / \lambda$$

$$\phi = P(\lambda) / E$$

where h = Planck's constant ($6.63 * 10^{-34} \text{J} * \text{sec}$), c = speed of light ($3 * 10^8 \text{m}/\text{sec}$) and λ = Wavelength of light (nm).

Photoreceptor absorbance spectrum: The photoreceptor absorbance spectrum gives the probability that a photoreceptor absorbs a photon of a

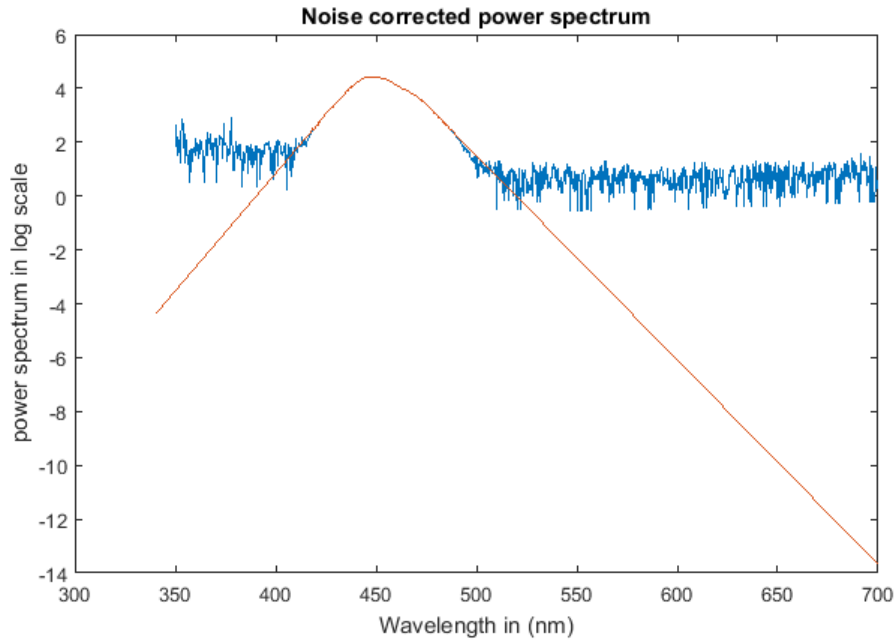


Figure C.1: Noise corrected noise corrected interpolated power spectrum for the blue LED of the DLP projector in the multi-electrode patch clamp setup.

given wavelength. It is estimated from the Govardovskii Nomogram [26], using the peak of the absorption spectrum of the photoreceptor.

The collecting area of photoreceptor: The effective light-capture area of a photoreceptor is comprised of its light funneling characteristics, quantum efficiency, the length and diameter of the outer segment and the density of the rhodopsin molecules. The collection area used in our calibration module for rods is $0.5 \mu\text{m}^2$ [43].

Combining all components, the photoisomerizations produced by light stimulation in the setup are calculated as:

$$\text{Photoisomerization} = \phi \bullet \text{Absorbance Spectrum} * \text{Photoreceptor Area} \\ (R^*/\text{rod}/\text{second})$$

Appendix D

Light Calibration Data Format

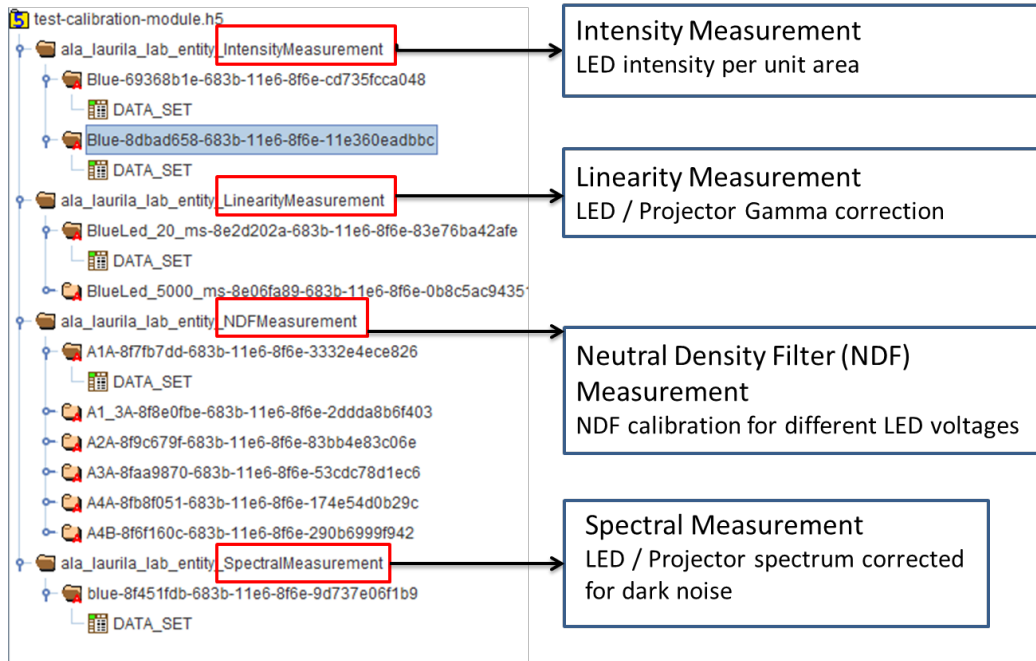


Figure D.1: Stimulus device calibration HDF5 file. It stores the history of the calibration measurements, which include intensity, linearity, NDFs, and spectral measurements.

Appendix E

Example Data Analysis

E.1 Analysis project

Using the date of the experiment, the analysis project is created as follows,

```
[project, offlineAnalysisManager] = createAnalysisProject(...  
'Example-Analysis_01',... % Name of the project 'experiments',  
{'101217Dc*Amp2'},... % Experiment date with amplifier channel  
'override', true); % Would you like to override the project
```

The result of the analysis project is described in Figure E.1.

E.2 Acquisition data filter

Consider a `Stimulus Protocol` : `LightStep` . It is a simple flash to stimulate the circuits in the retina. Its main stimulus parameters are `displayName` , `intensity` and `stimTime` . `displayName` defines the name of the `Stimulus Protocol` ; `intensity` describes the input to the light source; `stimTime` defines the duration for which the stimulus is displayed. To filter the epochs with these properties requires the following filter definition,

```
% Matlab code  
%  
% ' displayName ' First level grouping of filter.  
% | It matches all the epoch which has parameter name  
% | "displayName" and groups them according  
% | to its value.  
% |
```

```
% ' intensity      ' Second level grouping of filter.
%   |              It similarly matches the intensity parameter
%   |              and groups according to its value
%   |
% '  stimTime      ' So on ...
%

analysisFilter = struct();
analysisFilter.type = 'LightStepAnalysis';
analysisFilter.buildTreeBy = {'displayName', 'intensity', 'stimTime'};
analysisFilter.displayName.splitValue = {'Light Step'};
```

```
{
  "identifier": "Example-Analysis_01",
  "description": "Hope it will be defined later !",
  "experimentList": [
    "101217Dc*Amp2"
  ],
  "cellDataIdList": [
    "101217Dcc2_Amp2"
  ],
  "analysisDate": "20171215",
  "analysisResultIdList": [],
  "performedBy": "narayas2",
  "file": "C:\\Users\\narayas2\\data\\analysis\\Projects\\
}
```

Figure E.1: Example project definition: each project has a project identifier, a brief description of the project, a list of `CellData`, the analysis date, the analysis results and the name of the person who performed the analysis.

E.3 Analysis execution steps

The function `buildAnalysis` generates the analysis tree as defined by the epoch filter, and updates the project file with the analysis results.

```
buildAnalysis(...
'Example-Analysis',... % Name of the analysis project
analysisFilter)      % Type of analysis filter(s)

% This results in below analysis tree
%
%      ,          project==Example-Analysis_01 (1)          ,
%      ,
%      ,          |          ,
%      ,  analysis==LightStepAnalysis-101217Dcc2_Amp2 (2)  ,
%      ,
%      ,          |          ,
%      ,          displayName==Light Step (3)              ,
%      ,
%      ,          |          ,
%      ,          intensity==1 (4)                          ,
%      ,          +-----+-----+                        ,
%      ,          |          |                              ,
%      ,  stimTime==20 (5)          stimTime==500 (6)      ,
```

Further data transformation is performed by attaching a Matlab function to the filter definition. This Matlab function is described as a `FeatureExtractor`. To perform the feature extraction, the Matlab function handle is assigned to the desired level of the analysis tree and the analysis is rebuilt. The example below assigns `psthExtractor` to the lowest node (`stimTime`) in the example analysis tree built for the `LightStepAnalysis`. The `psthExtractor` generates a peri-stimulus time histogram (PSTH) of the neuron's responses in the selected epoch group.

```
analysisFilter.stimTime.featureExtractor = {...
@(analysis, epochGroup, analysisParameter)...
    sa_labs.analysis.common.extractors.psthExtractor(...
```

```
analysis,...  
epochGroup,...  
analysisParameter)...  
};
```

```
buildAnalysis('Example-Analysis',... % Name of the analysis project  
             analysisFilter)         % Type of analysis filter(s)
```

The details of the example functions are documented in <https://ala-laurila-lab.gitbooks.io/sa-labs-analysis-docs/content/building-analysis-pipeline.html>