TECHNISCHE UNIVERSITÄT WIEN

Virtual-Reality System of the Iowa-Gamble-Task for Head-Fixed Mice

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Biomedical Engeneering

eingereicht von

Maximilian Hoheiser BSc.
Matrikelnummer 1229176

ausgeführt am Institut für

Analysis und Scientific Computing
Technische Universität Wien

unter der Anleitung von

Ao.Univ.Prof. Dipl.-Ing. DDDr Frank Rattay

sowie

Univ.-Prof. Mag. Dr. Thomas Klausberger
Dipl.-Ing. Hugo Malagon-Vina, PhD

Wien, 14. Februar, 2021

Kurzfassung

In-Vivo Elektrophysiologie-Techniken im allgemeinen und Multi-Unit-Aufnahmen in Kombination mit high-density Elektroden, an wach und aktiven Nagetieren insbesondere, ermöglichen die Bereitstellung von Daten, um den neuronalen Code zu knacken. In letzter Zeit findet ein Wechsel von frei beweglichen Tieren zu Kopf-Fixierten Virtual-Reality Verhaltensaufgaben vermehrt statt. Dies ist vor allem auf eine bessere Kontrolle vom Verhaltensumfeld, sowie breiterer Verfügbarkeit von Werkzeugen und Techniken für Kopf-Fixierte Aufgaben zurückführbar. Diese Arbeit befasst sich mit der Entwicklung von so einem Virtual-Reality System basierend auf dem Iowa-Gamble-Task, welcher sich durch probabilistisch Ergebnisse für Belohnung und Bestrafung, ohne externe leitende Hinweise zusammensetzt.

Die Arbeit ist in zwei Teile aufgeteilt. Der erste Teil befasst sich mit der Analyse des Virtual-Reality Systems für Kopf-Fixierte Mäuse, basierend auf dem beschriebenen Verhalten, welches aktuell vom Department für Kognitive Neurologie der Medizinischen Universität Wien genutzt wird, sowie der Auswertung von Daten, die mit diesem System erhoben wurden. Der Autor untersucht eine Gruppe von Neuronen, im Präfrontaler Cortex von Mäusen, die einen signifikanten Unterschied in ihrer Feuerrate, abhängig von der Belohnungs-Wahrscheinlichkeit (75%, 25% und 12,5%) des Verhaltens aufweisen.

Der zweite Teil beschäftigt sich mit der Konzeptionierung, Entwicklung und Implementierung eines neuen Virtual-Reality Systems, basierend auf demselben Verhalten. Das neue System kombiniert Open-Source Komponenten, mit einer weiten Verbreitung in der In-Vivo Elektrophysiologie Gemeinschaft, insbesondere der Bpod State-Maschine, und dem OpenEphys Elektrophysiologie Aufnahmesystem. Weiteres wird die Konzeptionierung und Implementierung von maßgeschneiderten Hardware-Komponenten wie eine schallisolierte Box, Schnittstellen zwischen Maus und System sowie Integration des Elektrophysiologie Equipment gezeigt. Abschließend präsentiert der Autor die Anpassung eines standardisierten Frameworks zur Speicherung, Analyse und Handhabung von Daten, die von dem neuen System erzeugt werden, welches auf dem Neuro-Data-Without-Boarders Datenstandard und einer zentralen Datenbank basiert.

Abstract

In-vivo electrophysiology techniques, especially multi-unit-recording of awake and behaving rodents implemented with high-density electrodes, provide the data that forms the backbone for unraveling the neural code. Lately, a shift from historically freely moving behavior tasks to head-constrained virtual-reality tasks is occurring, due to the benefit of higher control over the task environment and wider availability of tools and techniques. This thesis is concerned with the development of such a virtual reality system for a behavior task based on the Iowa-Gambling-Task, comprising changing outcome probabilities of reward and punishment, without external guiding cues, which simulates real-life decision making.

The thesis is divided into two parts. The first part considers both analysis of the virtual-reality system for head-fixed mice, with the above-mentioned behavior task, currently used by the Department of Cognitive Neurobiology at the Medical University Vienna, and analysis of data previously recorded with that particular system. Here the author investigates a group of neurons in the prefrontal cortex of mice, that display a significant difference in their firing rate depending on the reward probability of the gamble-choice (75% 25% and 12.5%).

The second part of the thesis focuses on the design, development, and implementation of a new virtual-reality system for the behavior task, combining open-source components, with a high adoption rate throughout the in-vivo electrophysiology community, such as the Bpod state machine and the OpenEphys electrophysiology acquisition system. In addition, the design and implementation of custom hardware components of soundproof enclosures, interfaces between the mouse and the system, and integrations of electrophysiological recording equipment for the new system are shown. Furthermore, the author proposes the adoption of a standardized framework for storing, analyzing, and managing data generated by the new system, based on the Neuro-Data-Without-Borders data standard and a centralized database.

Acknowledgements

My utmost gratitude belongs to Hugo Malagon-Vina, who not only guided me for the data analysis part but also never was too tired to answer my questions about the task and the setup even at late hours while I was developing the new setup. The discussions with him, more than once helped me to spark new ideas on how to overcome seemingly unending problems. I am especially thankful to him for the enormous work he put in to help and corrections of the written part of the thesis.

I also want to thank Thomas Klausberger, the PI of the lab at the Medical University of Vienna, not only for giving me the chance to work in his lab, but also provide the funds to realize the project. I had the luxury of being able to purchase all the parts and equipment that were needed during the development and for the final setups, without any delay or constraints. This was also made possible by the efforts of Marta Solano and Alexandra Schneider. I am especially thankful for his tireless efforts to keep the lab and the center in operation during the 2020/21 corona situation, allowing me to unhindered continue with my work. It's safe to say that due to his commitment, the only way the pandemic affected me was that my bike commute to and from the lab got more enjoyable since fewer cars were on the street.

In addition, many thanks to Frank Rattay, my advisor at the Technical University of Vienna, who made it possible for me to do my thesis at the Medical University of Vienna. The process of applying and getting the topic approved was one of the quickest and most undemocratic ones I have experienced, emails with questions to him never were unanswered for more than a few hours.

In regards to software development in python, Ben Orli-Natason, José Guzmán, and Alexander Wallerus gave me input and helped me solve particularly nasty problems.

I want to thank my friends Michael Bauer for feedback and discussion during the work and Peter Gerges, who pushed me during the tedious final part of writing the thesis and helped me by motivating me with a bet.

Finally, I want to thank my father Franz Hoheiser-Pförtner for his input on the systematic approach for writing the part about software development, and the countless hours spent on feedback and corrections.

Contents

List of Terms and Acronyms

AMPA alpha-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid.

API Application programming interface.

BCI brain-computer interfaces.

block or probability block, the consecutive trials with the same probability for the large reward at the gamble-side.

BMI brain-machine interfaces.

CSV Comma-separated values.

EEG Electroencephalography.

gamble-side prdefined left or right stimulus screen (opposit of safe-side), which if selected results in a probabalistic large reward.

gamble-task implementation of the Iowa-Gamble-Task in a virtual reality environment for head fixed mice.

Git distributed version control system for tracking changes across files.

GUI Graphical user interface.

Intan electrophysiology acquisition system .

LFP local field potential (corr. local electric potential).

MUA Muli unit activity.

NMDA N-methyl-D-aspartate.

NWB Neurodata Without Borders.

OFC Orbito-frontal cortex.

PCA principal component analysis.

PCB PPrinted circuit board.

PFC Pre-frontal cortex.

Phenosys virtual-reality system .

PSTH Peristimulus time histogram.

safe-side prdefined left or right stimulus screen (opposit of gamble-side), which if selected always results in a small reward.

SMD Surface-mounted device.

TTL a standard signal shifting between low volt (0-0.8V) to high volt (5V) used for simple digital communication.

UI User interface.

Preface

This chapter introduces the reader to the motivation for the thesis and shortly gives an overview of what it is based on. Following the motivation, the problem and the scientific question is outlined. This should prepare the reader for what to expect from the sub-parts. To this aim, the structure of the thesis will be outlined. Since the thesis is tightly interwoven with prior research from the lab and prior art, this chapter concludes, by systematically outlining, what was newly created by the author and what is prior art.

1.1 Motivation

Understanding the higher-level information processing of the brain arguably is one of the biggest challenges yet ahead of humanity. To understand information processing in the brain, it is necessary to understand the neural code, encoding and decoding mechanisms of neurons, signaling models, and network dynamics. A significant aspect of decoding and understanding the neural activity is the ability to measure from multiple individual neurons by recording their action potentials. The method which records action potentials is called electrophysiology. With the newest multi-unit-recording techniques, experimenters today have the ability to simultaneously record from hundreds and up to a few thousands of neurons at a very small timescale. However, to actually decode this data, the necessary step must be to link in-vivo electrophysiology recordings to behavior, with the ultimate goal of reading the agent's thoughts.

With emerging technology and the even closer integration of machines and artificial intelligence, electrophysiology will play a major role in the next developmental step of humanity. The possibility to augment the human cortex especially the prefrontal cortex through a direct connection with modern information processing technologies is for me the next possible evolutionary step for mankind. Though such a connection to external processing power and fast access to information we could unshackle us from the slowness of evolutionary development that it will take to further evolve our internal mental processing capacities.

This thesis builds on the initial work of (Passecker *et al.*, 2019) and a later adaptation of his gamble-task for mice with the initial proof of concept and the first implementation of the

task in a virtual reality setup for mice by Jian Gang and Aron Köszeghy.

1.2 Problem, Question, and Aim

The main problem to address for the virtual-reality gamble-task based on the Phenosys system is twofold. On the one hand, it concerns the training of mice for this particular behavior task and the Phenosys system itself. On the other hand, the encoding, storage, and analysis pipeline of the data generated thereby are not standardized and version controlled.

Behavior data recorded with the Phenosys system and neural data recorded with the in-vivo electrophysiology system have yet to be analyzed and the proper event synchronization between both on a millisecond precision proven. A precise synchronization is a fundamental requirement for analysis. It is suspected, that the synchronization protocol does not fulfill the necessary requirements fully. Furthermore, the already evident issues with the Phenosys setup which is based on closed-source hardware and software are the customizability of the behavioral tasks. Only a proprietary visual scripting language can be used to design behavior paradigms. This software is severely limited. For example, it is not possible to use different distributions for the gamble-side safe-side ratio which dynamically adapt depending on the choices of the animal, only a fixed ratio can be set. In addition, the modification of the task and implementation of new or slightly adapted tasks is very time-consuming and complicated to learn for new scientists. Furthermore, the hardware components compatible with the system are very few and the options are limited. This limits the freedom in designing future behavior tasks with the Phenosys system.

Data generated by the Phenosys system, is based on a proprietary data standard. This standard is not directly compatible with data processing pipelines and analysis packages used by the wider neuroscience and computational neuroscience community. The Phenosys setup and gamble-task paradigm generates in a single session (roughly 30 minutes of recording time) about 1.5 GB of raw data. For the whole experiment cycle, this will easily amount to around 1 TB of data. This data not only has to be preprocessed but stored in such a way that it is later easily retrievable to perform analysis on.

Question 1:

Does there virtual-reality gamble-task implementation based on the Phenosys system displays proper function and are both behavior data and in-vivo electrophysiology data generated by the system correctly synchronized?

Question 2:

Are putative neurons in the electrophysiology data, previously recorded with the Phenosys system from the gamble-task, that have firing rate changes correlated to specific behavior

events, and how is such a potential correlation depending on reward probability, gamble or safe-side and the respective combinations thereof?

Question 3:

What would be the design, development, and implementation of an updated system for the virtual-reality gamble-task for animal training and in-vivo electrophysiology recording, that comprises hardware and software which can be easily configured for modified, integrated with the existing lab-ecosystem and extended for future behavior-tasks?

What data standards and frameworks for encoding, storing, and analyzing data produced with the updated system are available, and how can the most suitable one be implemented?

To answer these questions and overcome the described problems, the aim of the thesis is to first analyze the data from the Phenosys system, based on the behavior task initially developed by Passacker for freely moving rats and implemented by Gang and Köszeghy for head-fixed mice. With insights gained from the Phenosys system, a new experimental setup for in-vivo electrophysiology experiments on awake and behaving mice in a head-fixed virtual reality environment for the gamble-task behavior paradigm shall be designed, developed, and implemented. Open-source components, standardized frameworks for high compatibility with analysis tools, a high level of flexibility and expandability for future behavior tasks should be at the core of the new system. Also, a potential switch to a common, open-source, and unified data standard, with a high adoption throughout the neuroscience community for the new system shall be evaluated.

Adaptability and expandability of the hardware are important, since experiments in neuroscience, especially in the domain of electrophysiology, require significant upfront investments in equipment and expertise both in terms of time and money. A system that can be easily adapted for future experiments and integrated, with the existing ecosystem, will on the one hand bring down costs, and therefore lower the entry barrier, and on the other hand, save precious time until animals can be trained and data can be gathered.

A more standardized approach of how information is encoded, how data is stored, and processing scripts are implemented will potentially lead to better transparency and usability. This will hopefully increase the efficiency for current scientists and make it easier for new scientists and future collaborators to ramp-up their understanding of the past experiments, enabled by easy access to data. A significant benefit of such a standardization will also be that analysis pipelines are much better cross-compatible and don't have to be developed from scratch for each new experiment.

Both innovations for hardware and software are desirable since they lead to a higher scientific output, which is favorable, in particular for the lab and in general to the global neuroscience community to advance our understanding of the brain.

1.3 Structure

The thesis is structured into four parts, first a general introduction about the background: historic developments in neuroscience and of the brain, overview of the current understanding of neuronal signaling and computation of the brain, electrophysiology in general and in-vivo multi-unit recordings in particular, animal behavior, and decision making. The general introduction is followed by a specific introduction to the gamble-task and findings from Passacker, who developed the predecessor for the gamble-task, as well as the task implemented with the Phenosys system by Gang and Köszeghy.

The second part is a detailed description of the methods used in this thesis, followed by the Results. The result part is split into two subparts, the first part focuses on the analysis of neural data previously recorded with the Phenosys system. The second part is about the design, development and implementation of a new hardware and software setup for the gamble-task. Finally, the results are discussed and a conclusion is drawn.

Due to the development process of the new setup and the final product combining and adapting multiple different solutions based on prior art, the author feels that a clear separation for the design, development, and implementation as outline above in the introduction, method, results, and discussion would be hindering to the easy understanding by the reader. Therefore, this separation is somewhat blurred for the second part of the results of the thesis, focusing on the design and implementation of the new system. This part is divided into separate subparts, with each containing introduction to prior art and frameworks, methods of the application thereof, and results in terms of how it is modified and what is newly developed to implement the new system. The discussion is again more clearly separated comprising both first, the data analysis part and second, the new setup.

1.4 Original and Prior Art

To clearly outline, and avoid misunderstandings about false claims, this section gives an overview of what is developed by the author and what is adapted from prior art.

Prior art:

All the datasets that are analyzed are not recorded by the author, but provided by Gang and Malagon. Furthermore, scripts for behavior analysis are based on work by Lagler, and Malagon.

The behavior task for the new system was developed by Gang and Köszeghy and modified by Malagon at the division for Cognitive Neurobiology at the Center for Brain Research of the Medical University Vienna (not published) under the guidance of Klausberger. The concept for the task is based on the initial publication by Burges and the Cortexlab at UCL (*Cortex-Lab/Rigbox* 2021).

The electronic component used for the new setup, comprising the Bpod state machine, rotary encoder module, port breakout module, analog input and output module are designed by Sanworks (*Sanworks* 2021). The firmware for these microcontrollers is also developed by Sanworks. The used Python Application programming interface (API) (PyBpod) for controlling the Bpod parts of the setup is developed by The Scientific Software Platform from the Champalimaud Foundation of the Unknown (*Scientific Software | Champalimaud Foundation* 2021). The hardware for the rig of the setup, especially the design of the mouse-holder is oriented on the rig published by the International Brain Lab (The International Brain Laboratory, Aguillon-Rodriguez, *et al.*, 2020). The head-plates for the system are based on a design from IST Miba Workshop (*IST Austria | Miba Machine Shop* 2021).

The hardware and software for the OpenEphys recording platform is developed by OpenEphys (Siegle *et al.*, 2017). The new synchronization board between OpenEphys and Bpod is based on the 8 channel Analog Output Module from Sanworks (*8 Channel Output Module Bpod* 2021).

The data standard used is developed by Neurodata Without Borders (Teeters *et al.*, 2015). The database implemented with the new system is based on the Alyx Database by the Cortexlab from UCL (The International Brain Laboratory, Bonacchi, *et al.*, 2019).

New and original:

Scripts for analyzing the data are developed by the author. These scripts form the basis for the analysis pipeline implemented with the new system.

The complete Bpod hardware is manufactured by the author based on designs made available by Sanworks. Furthermore, the implementation of the task for the Bpod system based on PyBpod API, and all scripts, therefore are also newly developed during this thesis. This comprises: software to control the Bpod state machine and run the task from the PyBpod Graphical user interface (GUI), software to interact with the rotary encoder module and the stimulus on the screens, software to control the reward system and modifications to the GUI for the implementation of the gamble-task. Software allowing for easy deployment of the new system on any compatible hardware is also created.

The complete design, development and manufacturing of the rig hardware for the new system, both for the training instance and the recording instance is new.

That comprises: the reward system, screens for the stimulus, mouse holder (3D printed parts and CNC machined parts), reward delivery system, and camera setups. For the training instance, the enclosure and soundproof box. For the recording setup, the adapter between stereotaxic frame and head-plate, as well as the soundproof enclosure also is newly designed, manufactured and tested during this thesis. The updated versions of the head-plates are also designed and manufactured by the author. PCs controlling the system are designed and assembled by the author.

A new synchronization hardware board to interface the Bpod with the OpenEphys is developed, manufactured and tested, as well as the software to interface the OpenEphys with the Bpod via this board.

The implementation of the server, both hardware and software, based on Proxmox and the implementation and adoption of the Alyx database to the specific needs of the new setup further is new.

Introduction

This chapter presents a short introduction to the broad field of neuroscience and neuronal signaling processes in general, and to the specific and for the thesis relevant areas, in particular behavioral neuroscience with animal models and in-vivo multi unit electrophysiology. A particular emphasis is placed on gamble-tasks with varying reward probabilities.

In addition, it presents the behavior task which forms the basis for this thesis, how it was adapted to a virtual-reality task and the training and recording system previously for this implementation used by the lab.

2.1 Introduction to historic Development of Neuroscience

> *"Men ought to know that from the brain and from the brain alone arise our pleasures, joys, laughter and jests, as well as our sorrows, pains, griefs and tears."*
>
> Hippocrates

The brain is arguably our most important organ since it acts as a control center for most of our actions and reactions. Its functionality ranges from simple endocrine processes over conscious muscle contractions and movement of body parts to planning of complex behavior to interact with the environment. Building on the tedious work of countless scientists over the past millennia, the understanding of biology and the human body has been mostly integrated in the last century. Yet neuroscience still is a frontier of many unanswered questions.

With significant breakthroughs such as the discovery of neurons by Cajal and Golgi, the imaging of their structure with high-resolution microscopy and the mapping of the mouse brain connectome, research has come a long way. However, there is still a significant portion of the brain which presents an unsolved mystery nowadays.

We are currently able to sequence genes, translate them to sequences of amino acids and observe how they interact and form proteins to build up cells and control cell functions. These functions, especially in the brain, can be measured with imaging techniques and

electrophysiology methods. From the data a common conceptual framework of cellular neuroscience is derived and constantly refined. This advancement represents the bottom-up approach to understanding the brain (Kandel *et al.*, 2013, Chaper 1 and 2).

The top-down approach is represented by the field of behavioral science, cognitive science, and psychology. Long since the ancient Greeks, the understanding of behavior and how it relates to our brain lay in the domain of philosophy. The great philosophers of enlightenment, and especially Descartes in his work *De homine,* focused on the distinction between physical body and mind. In the following decades, Hume and Kant built on that by developing a framework for the structure of the human mind. Despite the fact that both of these efforts contributed greatly to the understanding of the body-mind problem, they were not based on scientific methods. The incorporation of experimental methods at the beginning of the 19th century by Wilhelm Wundt led to the detachment of psychology from philosophy and the accompanying rise of experimental psychology. This development was further linked to the brain by researchers such as Herman von Helmholtz, who used the strict methods of natural sciences thus basing his conclusions on empirical evidence through data gathered from controlled experiments. Arguably, one of the key figures in the top-down approach was Sigmund Freund, who played the foundation stone for the research on cognition.

The rise of behaviorism, and its counter-movements in the 1950s by Chomsky, accompanied by the invention of artificial intelligence as an interdisciplinary field by MacCarthy and Minsky as well as the gaining traction of Shannon's information theory, lead to a blurring of the lines between different fields and their common denominator: the brain. This could be considered the fork of neuroscience and cognitive science. They represent the bottom-up and top-down approaches respectively. (Miller, 2003)

The main task of the 21st century is to integrate, horizontally as well as vertically, the concepts of brain-related sciences to come up with a unified framework of how the mind works - how the functional behavior of cells build networks and circuits and how these translate to behavior and complex mental processes (Rieke, 1999, Chapter 1) (Kandel *et al.*, 2013, Part 1.). In short, we want to be able to understand, model and reconstruct the thinking machine.

2.1.1 Evolution of the Brain

To understand the relation of the later concepts, it is necessary to build up a solid base. The evolution and development of the brain of the modern human is a vital part of this process.

The increase of brain mass, as well as the complexity of its structure, has been a continuous process over millions of years. Driven by changing climatic conditions about 9 million years ago, the environment around our ancestors in the African rain forests begun to change and the jungles gave way to grasslands The changing environment to grasslands introduced the challenge of seasonal differences which forced our ancestors to travel much greater distances in search of food, as well as to rely on fallback food of lesser nutritional quality. Compared to a chimpanzee an australopith usually traveled around twice as far. But is nearly two times more efficient, which leads to no increase in daily energy expenditure. At the end of this

change are the first hunter-gatherer tribes. These two factors, on the one hand, the ability to hunt relatively easy and on the other hand, the need for different food sources lead to a large increase of meat in their diet. The higher nutritional density of meat led to a reduction of the digestive tract. The surplus energy was devoted towards an increase in brain mass. (Lieberman, 2013, Chapter 3)

To emphasize the significant growth of brain size, a chimpanzee's brain is approximately $400cm^3$; the famous fossil Lucy (Australopithecus afarensis) has a skull which suggests a brain of approx $450 - 550cm^3$ and a modern-day human has a brain size of $1400cm^3$

Not only the size changed, the structure of the brain gradually became more complex. The surface area of the cerebral cortex and especially the frontal lobe increased by folding inwards and forming gyri and sulci. The cortex area increased from $0.7m^2$ in a chimpanzee to $2m^2$ in a modern human. This increase is not just an increase in the number of neurons but also in the complexity of circuits. Notably, in the frontal cortex, the cortical changes during the evolution of our ancestors presumably stem from an increase in the number of neural circuits not in the side of the individual circuits. (Hofman, 2014)

2.2 Basic Anatomy of the Brain and Signaling Mechanisms

Most of the computation we will be focusing further in this thesis takes place in the prefrontal cortex and particularly in the medial and lateral prefrontal cortex, this anatomical introduction will mostly focus on this part of the brain.

2.2.1 Basic Anatomy of the Frontal Cortex

The brain consists of the cerebrum, the smaller cerebellum as well as the brain stem connecting each other and via the spinal cord to the peripheral nervous system. Suspended by the cerebrospinal fluid and protected by the three membranes dura, arachnoid, and pia mater, the cerebrum is divided into five different lobes: the frontal lobe, parietal lobe, occipital lobe, temporal lobe, and insula lobe. The division of the brain into these lobes follows mostly a functional separation, with the somatosensory and motor functions mostly located in the parietal lobe; the visual system originating in the occipital lobe; memory and language belonging to the temporal lobe; and emotions originating from the insula lobe (Bear *et al.*, 2015, Chapter 15). Most of the circuits which distinguish us and our brain, from the one of apes, are located in the frontal lobe (Eichert *et al.*, 2020). These circuits, especially in the prefrontal cortex govern higher-order behavior and executive functions such as goal planning for reward maximization as well as forming complex models of the world and projecting the outcome of actions into the future (Smaers *et al.*, 2017). Interestingly, the prefrontal cortex (Pre-frontal cortex (PFC)) in humans is over proportional larger than the overall brain size difference compared to apes (Smaers *et al.*, 2017).

In addition, to the lobe structure, the cerebrum follows a lateral separation due to the central sulcus. Both sides are connected by a thick band of neural fibers through the corpus callosum. There is also a functional division between the left and right hemisphere, with the right side being dedicated to the more creative brain functions such as art music, and imagination where the right side builds up the rational mind for math and logic. (Bear *et al.*, 2015, Chapter 15)

Deeper within the cerebrum lies the limbic system, composed of the thalamus hypothalamus amygdala, and hippocampus. Its main functions are lower-order processing as well as acting as a relay between sensor input higher-order areas in the cortex and output via the basal ganglia and the direct and indirect pathways for controlling motor neurons.

The brainstem is located underneath the cerebrum and connects it to the cerebellum and the spinal cord. It is divided into three subparts: the midbrain, pons, and medulla oblongata. Furthermore, its function comprises regulating the regulation of sleep and the central nervous system. The long-range projecting neurons, which make up the diffuse modulatory system originate in the brainstem.

Overall the brain structure is very diverse on a macro level and specific areas can be attributed to specific functions, but the networks are highly integrated and computational processes depend on a multitude of anatomical areas. Also, on a micro-level, the differences decrease and the basic structure on a cellular level increases in similarity over the different regions, especially in the cerebrum.

2.2.2 Molecular Structures

To understand the big picture of neural computation and decision making in terms of reward maximization, it is necessary to look at the brain on a cellular level.

The main unit of the brain is the neuron, which performs the most basic level of computation. Neurons then are interconnected via complex networks for higher-level computations. Santiago Ramon y Cajal used Nissel and Golgi staining techniques to meticulously created detailed drawings of the brain circuitry and showed that they are made up of neurons connected via axons. Due to his work the "neuron doctrine" became widely adopted. The fundamental statement is, that the computing part of the brain is formed by individual neurons connected via synapses, and signal transduction occurs via action potential propagation.

Most of the synaptic connections are chemical junctions although a few are electrical. A chemical synapse consists of the synaptic-end-terminal belonging to the sending neuron, a synaptic cleft (a physical separation between both cells), and a postsynaptic terminal belonging to the signal receiving neuron. If an action potential is triggered by the soma it will propagate via the opening of ion channels and the resulting depolarization of the axon and reach the synaptic terminal. At the terminal, vesicles will release neurotransmitter which diffuse across the cleft and bind to the postsynaptic receptors, eliciting a response from the receiving neuron.

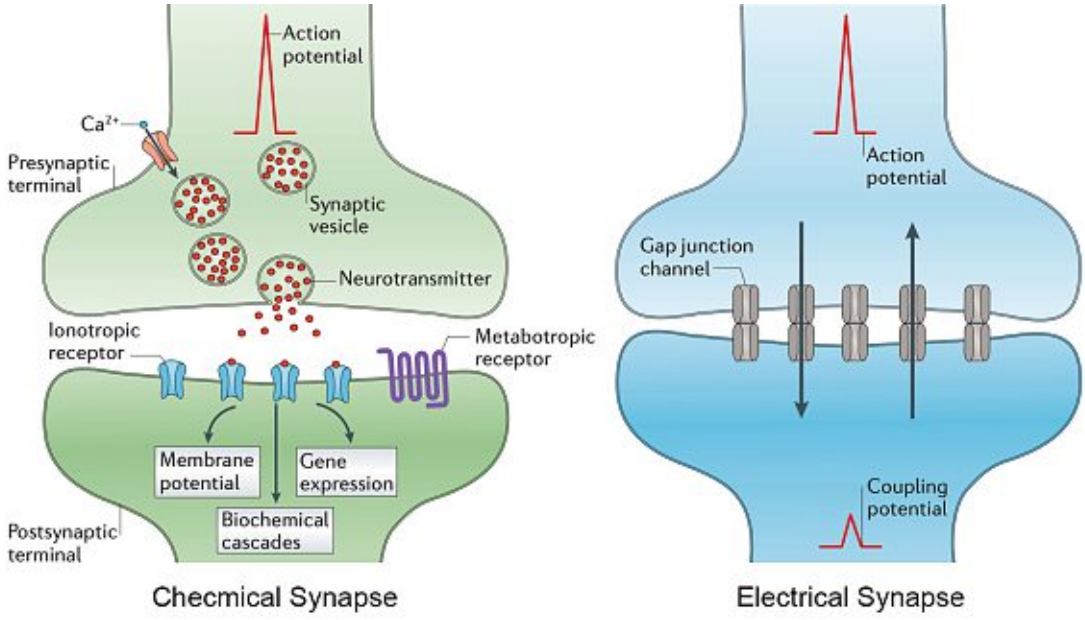


Figure 2.1: Chemical and electrical synapse (adapted from Pereda, 2014, Figure 1)

In a electric synapse both neurons are directly connected via gap junctions. Ions from the action potential of the sending neurons axon will directly diffuse through these junctions to the receiving dendrite.

Each internneuron (not sensory neurons and motor neurons) receives input from other neurons via three different connections: axo-dendritic, axo-somatic connection, and axo-axonic connection. In addition to the location of connection, the type of connection response is distinguished between excitatory and inhibitory synapse. The interplay between excitatory and inhibitory synapses in complex circuits plays a major role in higher-level processing and especially in oscillations and synchronization. (Bear *et al.*, 2015, Chapter 2, Chapter 7)

Each synapse is classified by the neurotransmitter released from its vesicle if an action potential reaches the synaptic end terminal.

The cerebral cortex (greek for bark) is made up of roughly 14-16 billion neurons and is structured in six outer layers (layer 1 being the outermost one) of grey matter formed by the neuronal bodies. These neurons have axonal interconnections to nearby neurons in the same layer and to different layers. Most functional circuits are connections between neurons performing similar roles, due to an increase in communication efficiency they tend to be in close vicinity to each other. (Shipp, 2007)

The Neurons connecting to more distal parts of the cortex and other parts of the brain project their axons inwards through the six layers. These axons form the cortical white matter because they are heavily myelinated to increase the transmission speed and the fat tissue mostly looks white. (Shipp, 2007)

The six layers of the cortex are optically distinguishable due to the different neurons present in the layers.

- **Layer 1:** receives mostly input from regions outside of the cortex and consists of horizontal oriented gabaergic (inhibitory) neurons which play a vital role in synchronizing signal transmission in the cortex. The input axons synapses with the extensions of apical dendrites from pyramidal neurons mostly located in layers 3 and 4.
- **Layer 2:** contains densely packed stellar neurons and smaller pyramidal neurons.
- **Layer 3 and 4:** contain mostly neurons belonging to the above mentioned apical dendrites from layer 1.
- **Layer 5:** from here originate most of the axons that leaf the cortex and are involved in downstream signaling to lower regions such as subcortical structures of the basal ganglia. In the primary motor cortex layer 5 is the location of the cell bodies whose axons form the primary motor neuron traveling all the way through the spinal cord synapsing with the lower motor neurons which activate muscle fibers.
- **Layer 6:** contains smaller spindle pyramidal neurons that form a precise connection with the thalamus.

(Bear *et al.*, 2015, Chapter 7)

As described earlier, the frontal lobe in general and the rostral part, the prefrontal cortex (PFC), in particular, is particularly larger and higher developed in humans compared to apes and other animals. Although the prefrontal cortex seems to be of such high significance for the human brain, it is poorly understood. Based on the overdeveloped difference it is assumed, that the PFC is involved in many of the distinct human characteristics, such as self-awareness, complex planning and problem-solving as well as decision making (Bear *et al.*, 2015, Chapter 24).

Experiments with monkeys imply that the PFC is involved with information storage for short term working memory (Rodriguez and Paule, 2009). Furthermore, experiments such as the Wisconsin card-sorting test (Berg, 1948) suggest that the PFC and working memory are involved in complex problem solving and planning of future behavior. More modern experiments, mostly with rodents and non-human primates, link the prefrontal cortex with decision making. Especially the orbito-frontal cortex (Orbito-frontal cortex (OFC)) seems to be closely involved in decision making and the process of assigning values to different options from which to choose during the decision process (Padoa-Schioppa and Conen, 2017).

2.3 Neuronal Signals

To understand information processing in the brain, it is necessary to take a look at the biophysical mechanisms of neurons, signaling models, and network dynamics.

2.3.1 Action Potential

The basic computing unit of the brain is the neuron, which has the above described biological properties. The neurons communicate with each other via action potentials caused by depolarization of parts of the cell. An incoming action potential caused by depolarisation is referred to as a spike. Although this process can be regarded in most of the cases as binary - no incoming action potential representing 0 and an action potential via a synapse as 1- there is not clear cut, due to complex membrane dynamics and spatial and temporal summation processes (Bear *et al.*, 2015, Chapter 2).

The dynamical properties are controlled by mostly two ion types, potassium ions K+ and sodium ions Na+, and their respective concentration inside and outside of the cell membrane. There are two stages distinguished by their different membrane potential, the resting membrane potential (-65mV) and the depolarised membrane (+20mV, slightly different depending on neuron type). The exact resting membrane potential depends on the equilibrium of all involved ions. It is the steady-state at which the forces of the concentration gradient and the potential gradient on each ion are equal (Rieke, 1999, Chapter 2), (Bear *et al.*, 2015, Chapter 4).

Neurons in the cortex receive typically input from up to 10.000 synapses, which are integrated for downstream signaling. The integration for excitatory input signals, via spatial summation, mostly depends on the location of the synapse. Passive properties of signal transduction in dendrites, follow an exponential decay proportional to the length. This means that the dendrite more distal signals contribute less to the total summation than medial signals (Bear *et al.*, 2015, Chapter 1). In recent years more and more evidence has been found that dendrites also possess active signaling properties and are able to generate dendritic spikes (Manita *et al.*, 2017).

Furthermore, a dendrite or a soma can also receive an inhibitory synaptic input, which leads to a decrease of the membrane potential by opening Cl- ion, a process that is referred to as shunting. Shunting has the biggest influence if the inhibitory axon terminal synapses directly to the soma of the postsynaptic neuron. (Bear *et al.*, 2015, Chapter 4)

If an incoming signal causes a depolarization higher than a certain threshold (a value around -55mV at most neurons), it causes a depolarization of the axonal membrane, and a spike is fired downstream.

The action potential consists of three phases. First, the rising phase where predominantly Na+ channels are opened at the threshold voltage and Na+ ions can enter the cell causing a rapid increase in the membrane potential. At -40 mV potassium channels start to open and K+ ions flow out of the cell slowing the rise of membrane potential until it reaches its maximum at around +30mV, which causes the Na+ channels to be inactivated.

The second phase, repolarisation, follows the peak of the transmembrane voltage and is caused by K+ ions flowing out of the cell via still active K+ channels. K+ channels are inactivated at around -90 mV and therefore the third phase is called hyperpolarization because the potential undershoots the resting membrane potential. The ion pumps slowly restore the steady-state at the resting membrane potential. (Trappenberg, 2010, Chapter 2)

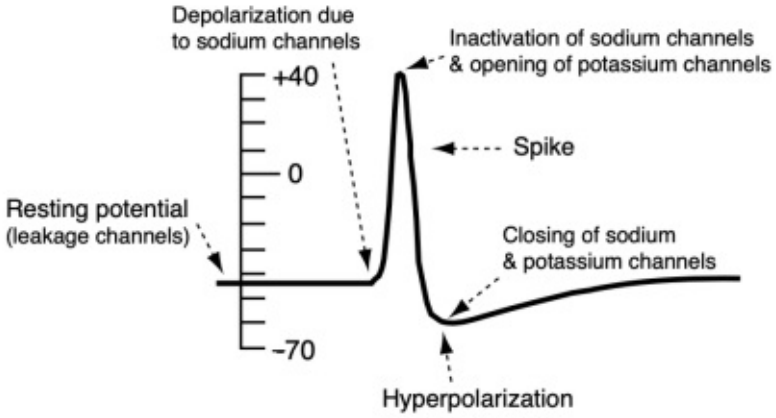


Figure 2.2: Stages of an action potential (Trappenberg, 2010, Page 33)

In the 1950s Hodgkin and Huxley used the voltage clamp method to vary the membrane voltage of the giant axon of a squid, as a step function and measure the injected current, which directly translates, after a short rising phase where the membrane capacitance is loaded, to the leakage current through the ion channels. By selectively blocking sodium and potassium channels, they could map the response curves of sodium and potassium, the two predominant channel types involved in action potentials (Rattay, 1990, Chapter 4).

Neurons transmit information in form of action potentials which are a depolarization of the cell, called spikes. At rest, each neuron is negatively polarized at around -70mV, which is the steady-state between forces on each involved ion of the concentration gradient and electric gradient. If a neuron receives an input that is strong enough to depolarize the cell to -55mV, sodium channels open, and the intracellular voltage quickly rises, an action potential is triggered. After a maximum depolarization of around 40mV Potassium channels are open and a Potassium efflux leads to a repolarisation. After an overshoot to -90mV, an active pump restores the concentration gradient at rest of Potassium and Sodium (Bear *et al.*, 2015, Chapter 4).

To transmit information in the form of spikes between neurons, the signal has to be conducted over a specific distance via the axon and dendrites of each neuron. The depolarization of the membrane is conducted to neighboring areas of an axon which also leads to a depolarization via ion current leading to a propagation wave through the Axon. Due to the inactivation of Sodium and Potassium channels after each spike, which causes a time period, refractory period, where an axon cannot be depolarised, action potentials can not travel back in the direction they came from. This mode is relatively slow with around 10 m/s and depends mostly on the time constant of an axon which is proportional to the membrane capacity, and thus to the diameter of an axon. This means that for faster transmission, the axons must be thicker, which presents a challenge since, especially in the brain, space is limited. (Rattay, 1990, Chapter 6) (Trappenberg, 2010)

To overcome this barrier, and increase the velocity of the transmitted action potential, some neurons are shielded against leakage current across the membrane. To shield against the sequential opening of channels the axon is wrapped in isolating tissue called myelin. To counteract the gradual decrease of the ionic current due to dampening, the myelin sheath has regular openings (Ranvier nodes) where ion channels cause active depolarization and an ion current across this small part of the membrane, refreshing the signal amplitude. The signal velocity of myelinated axons is proportional to the square root of the diameter, which leads to a higher signal transmission for axons with relatively small diameters, consequently enabling the dense packing in the cortex. (Rattay, 1990, Chapter 6)

2.3.2 Synaptic Plasticity

Synaptic plasticity describes the modulation of the strength of the synaptic connections. Depending on the neuronal interactions, the plastic effects can increase or decrease connection strength, referred to as facilitation and depression. The most commonly studied plasticity is divided, depending on the timeframe under which the synaptic changes occur, into short term and long term plasticity.

Short term plasticity involves rapid spike activity-dependent adaptations and ranges between milliseconds to minutes. These types of adaptations play a significant role in sensory modulation and short-lasting memory. Short-term plasticity is modulated by presynaptic calcium concentration. Calcium in turn is believed to directly modulate the readily releasable pool of neurotransmitters, in the presynaptic terminal, by modifying the vesicles (Bear *et al.*, 2015, Chapter 5). Besides the modulation via calcium concentration, most presynaptic terminals can also be modulated via g-protein-coupled membrane receptors, which in term modulate the receptor release behavior. These g-protein coupled receptors can either lead to short term facilitation or depression (Bear *et al.*, 2015, Chapter 6).

Long-term plasticity, on the other hand, ranges from hours to days and is more relevant for learning, in the classical sense, in the form of temporal memory storage (Abraham *et al.*, 2019).

The cellular mechanisms involved in long-term plasticity are mostly based on N-methyl-D-aspartate (NMDA) and alpha-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) receptors. By the consecutively firing of the presynaptic cell in short succession, the AMPA receptors are activated by glutamate neurotransmitter. NMDA receptors are also activated, but they are blocked by magnesium ions. The ions entering through the AMPA receptor increased the positive charge of the postsynaptic neurons, which leads to a de-blocking of NMDA receptor. This leads to an influx of calcium ions into the postsynaptic dendrite. The influx increases the intracellular ca2+ concentration activating a signaling cascade causing increased production and implementation of AMPA receptors in the dendritic membrane. The higher number of AMPA receptors increases the sensibility of the dendrite (Abraham *et al.*, 2019). This phenomenon mostly occurs in dendritic spines. Dendritic spines are small membrane protrusions where axons synapse on dendrites. They are very plastic and change

their shape and volume in down to a few hours. Due to this plasticity, they are believed to facilitate short-term learning by increasing the connection strength between specific existing axons and for new synapses with different axons, thus rewiring the connections. (Segal, 2005)

In a similar manner, long-term depression is caused by lower intracellular calcium levels due to not simultaneous firing of pre and post membrane. The lower concentration leads to activation of protein phosphatase, which causes AMPA receptors to disengage from the cell membrane, thus leading to fewer AMPA receptors and therefore to a lower sensitivity. (Citri and Malenka, 2008) (Bear *et al.*, 2015, Chapter 6)

These phenomena of plasticity are arguable the most important ones of the brain, since they allow our organism to change and adapt to the environment. They play a major role in decision making and integration of sensor inputs from the environment to achieve goal-directed behavior (Passecker *et al.*, 2019).

2.3.3 Signal Encoding

The first rule of neural signal transmission is that it is always noisy. Shannon postulated that the amount of information gained from a source is proportional to the amount of uncertainty, or entropy of the signal. For a reliable information transmission via a noisy channel, a decoding and encoding mechanism is needed. (Kay, 2003). Data obtained from in-vivo recordings provides a sound basis for analyzing these encoding and decoding mechanisms. Most of the measurements discussed in this thesis are obtained by in-vivo electrophysiological recordings.

Decoding mechanisms can be separated into three main methods: rate-based, temporal based, and population-based decoding. The firing rate is one of the methods to observe how neurons encode information. Popular mathematical approaches involve multivariate regression methods (Passecker *et al.*, 2019), dimensionality reduction and space-state methods (Malagon-Vina *et al.*, 2018) (Durstewitz *et al.*, 2010). More and more progress is being made in cracking the neural code by correlating changes in firing rate with behavioral parameters. Information is encoded in the firing rate of a neuron, thus by recording the spikes and analyzing how the firing rate changes over time the information change, encoded by that neuron, can be analyzed.

For example, Adrian and Zotterman could show that the number of spikes in a fixed time window, of muscle spindles, follow the increasing intensity of the stimulus (Adrian and Zotterman, 1926, Chapter 2.1). If external stimuli produce changes in the firing rate of specific neurons, decoding for a particular stimulus could be identified.

Furthermore, by linking the response function of a neuron to a stimulus, it should be possible to decode the stimulus only by looking at the firing rate of the neuron. This relation between stimulus and firing rate is called the tuning curve of a neuron. (Rieke, 1999, Chapter 2)

Although it could be shown that the firing rate significantly correlates with some stimuli, there are several mechanisms that modify the response over time and thus changing the

corresponding firing rate. One of these mechanisms is called adaptation and it leads to a decline in spike rate for continued stimuli. (Rieke, 1999, Chapter 2)

In addition to encoding via the average firing rate, the brain also uses time-dependent response of a neuron to a stimulus. The firing rate changes in accordance with the onset of a stimulus and acts as a detection mechanism. To observe the time-dependent firing rate, a trial is repeated multiple times and the response time to the stimulus is measured. This is usually plotted in a post-stimulus-time-histogram.

Besides the mean and temporal changes in firing rate, another temporal coding mechanism is the individual time between each spike. This temporal coding is gaining more and more relevance as researchers gather insights into the neural code by utilizing larger and larger datasets and are able to precisely manipulate neurons via methods such as optogenetic stimulation (Rieke, 1999, Chapter 1). These features are also the main source for clustering spike data.

Population coding represents yet another important way how information is encoded by the brain. Most of the neural computation is performed by a multitude of single neurons. It is of interest how a neural population identified by the type of neuron, firing behavior or spatial location changes and how these changes can be correlated with behavior or stimuli changes. For certain behaviors, there is a distinct change in which population of neurons responds to which part of the behavior. For repetitive stimuli from such behavior, the shift from one responding population to another directly encodes a stage of the behavior (Passecker *et al.*, 2019) (Malagon-Vina *et al.*, 2018).

The identification of population changes is aggravated by the fact that only a small fraction of the neural population in question actually carries the information. The actual information is represented by a sparse population code (Panzeri *et al.*, 2015). A particular example of neural population dynamic, is the case of topographical column organization, which is strongly present in the visual cortex and the somatosensory and motor cortex. Neural populations are spatially linked with stimuli location. (Trappenberg, 2010, Chapter 7.1)

A significant aspect of understanding the neural code, and thus being able to decode the sensory input from spike trains, is the ability to combine multiple neurons and to decode not only from individual firing rates or temporal signals but from the neural population. With the ultimate goal of reading the agents thought by decoding spike recordings, it still needs to be explored if this is possible by recording from a few distinct areas or if the networks are so interconnected, that the complete picture always depends on populations distributed across the complete brain. (Rieke, 1999, Chapter 1)

2.4 In-vivo Electrophysiology

Decoding and understanding the neural activity necessitates the ability to measure the activity of neurons. This means to record action potentials from single or multiple neurons. The method which records spikes is called electrophysiology. Since this work will exclusively be

working with data derived from in-vivo electrophysiology recordings, the section will only focus on this group of methods.

Electrophysiological methods have been one of the early backbones of neuroscientific research and led to a multitude of breakthroughs.

Beginning with the works of Luigi Galvani and Alessandro Volta investigating muscle twitching by applying electric current, electrophysiology has paved the way for measuring brain signals. Building on the first initial research, others such as Oersted and Helmholz used electrophysiology to measure neural responses of sensor stimuli. (Li, 2016, Chapter 1)

Walter Nernst used measurements of membrane potential and current across the membrane to derive the resting membrane potential. The spread of this discovery, aided by electrophysiological measurements, gradually started to increase with the next significant breakthrough by Hans Berger, who discovered the Electroencephalography (EEG). Soon, thereafter, Alan Hodgkin and Andrew Huxley used precise patch-clamp recordings to minutely map the relationship between transmembrane current and specific neurotransmitter release stages to derive the earlier described Hodgkin and Huxley equation (Office, 2021). Hubel and Wiesel used tungsten electrodes to record in-vivo from the cat's visual cortex, discovering the column organization of V1 (Li, 2016, Chapter 1). O'Keefe discoverd place cells in the Hippocampus of rats, which respond to a particular place in a maze (O'Keefe, 1976), and the Mosers discovered the grid-like organization of cells in the entorhinal cortex (Moser *et al.*, 2008) both via multiunit recordings in rats. This type of mapping has been found in many different areas since.

With the push of emerging multi-unit-recording methods, experimenters today have the ability to simultaneously record from hundreds and up to a few thousands of neurons at a very small timescale. (Covey and Carter, 2015, Chapter Introduction)

The latest breakthroughs are high-density electrodes like Neuropixels(Jun, Steinmetz, *et al.*, 2017) which allow recording from awake animals from several thousand neurons at the same time. The combination of such methods with optogenetic stimulation is now paving the way for large scale circuit mapping (Passecker *et al.*, 2019) (Grosenick *et al.*, 2015) (Grosenick *et al.*, 2015). With the aid of modern machine learning techniques, these massive datasets will lead to a fundamentally better understanding of the neural code.

In vivo electrophysiology also plays a major role in brain-computer interfaces (BCI) and brain-machine interfaces (BMI). Most of the current and early efforts towards BMI (starting in the late 1970s) focused on restoring neuromuscular signals via neuro-protheses. Although most of the early efforts around BMI focused on noninvasive techniques such as EEG, around the turn of the century more and more emphasis was placed on using invasive methods to detect single neuronal action potentials through microelectrodes. These neuroprostheses rely on large-scale brain activity measurements of the motor cortical area and elaborate post-processing to distinguish between different motor control commands. (Luan *et al.*, 2017). Currently, a number of research institutes and private companies are developing different kinds of electrodes allowing for high-density recordings. A major obstacle towards BMI is

the chronic implementation of electrodes without rejection by the brain tissue and active feedback closed-loop systems.

2.4.1 In-vivo Electrophysiology Recording Concepts

Both single unit and multi-unit recordings use the same underlying principle; they measure the local electric potential generated by neurons. The voltage of this electric field is recorded extracellularly in the brain tissue, and corresponds to the ionic changes, due to their movements between the neurons and the cerebral spinal fluid. This can be done with tungsten electrodes, very close to the soma and axon, or with silicon probes recording simultaneously from a proximal group of neurons. The closer the pickup electrode is to a single neuron and the farther away from other neurons, the more of the local field potential (corr. local electric potential) (LFP) is only from the local neuron and not a sum of multiple different neurons.

Each ion flow represents a current, which in turn generates an electric field that propagates through the brain tissue, with a decrease in amplitude relative to the square of the distance to the current source. The electric field does not propagate homogeneously in all directions due to the inhomogeneous capacity of the brain tissue. Cortical tissue also has a different permeability for different frequencies, further increasing the complexity of the LFP spectrum. Action potentials usually create high-frequency signals ($>$100 Hz), but in that range, the permeability of cortical tissue is smallest, thus the field is damped quickest (Li, 2016). Thermal noise, in form of high-frequency fluctuations, further complicates the signal. A typical representation of such a LFP from a single neuron at different locations from the soma is displayed in figure 2.3.

2.4.2 Single-Unit Recordings

Single unit recording methods can be separated into patch-clamp recordings and single wire recordings. The patch-clamp technique uses a very thin glass tube that attaches to a part of a cell wall. The pipet is filled with a conductive solution, from which the current created by the cell can be measured. Measurement is usually performed by a small silver wire inside the tube surrounded by the conducting solution, which is connected via an amplification unit to an analog-to-digital converter and a recording computer. A reference ground is usually placed somewhere on the dura over the cerebellum of the recorded brain. (Covey and Carter, 2015, Chapter 2)

The single-cell juxta cellular recording and labeling technique is in many aspects similar to the patch-clamp technique, but it also allows the experimenter to label the recorded cell for later identification of the neuron, its location, and its projections. The glass pipet is brought to a juxta cellular position and after satisfactory recording, the cell can be labeled using neurobiotin (Pinault, 1996). Depending on the biotin used the brain has to be stabilized after a short period of time. The time has to be long enough for the biotin to be transported to the whole-cell but short enough that it is not yet degraded. After stabilization, the brain

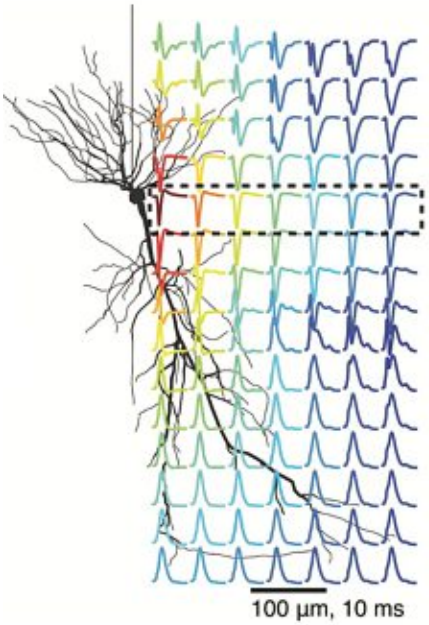


Figure 2.3: Spacial LFP map due to a spike from single neuron source (adapted from Schomburg *et al.*, 2012, Figure 1)

usually is cut with a vibratome and the labeled neuron is visually analyzed under a microscope. This allows for precise identification of the neuron in combination with in-vivo recordings of the neuronal activity (Schomburg *et al.*, 2012), which leads to a better understanding of the electrophysiological properties of the different neuron types (Klausberger and Somogyi, 2008).

The second type of electrode for single unit recordings is a metal wire, usually with a sharpened tungsten core. The wire is placed in a head-stage, that holds it and connects it to an amplifier and a recording unit. The wire is placed very close to the neuron of interest, which is achieved by lowering it with a micro-manipulator through the brain to the region of interest (Schomburg *et al.*, 2012).

2.4.3 Multi-Unit Recordings

If the electrode is not at a distance smaller than a few µm from the neuron of interest, the measured LFP will also have major contributions from neurons in a 140-300µm radius. This allows us to simultaneously record from multiple neurons at the same time (Li, 2016). However, in order to discriminate between all the surrounded-active neurons more than one electrode is necessary. There are two main methods for multi-unit-recordings, tetrodes, and silicon probes. Both use propagation speed to localize individual neurons

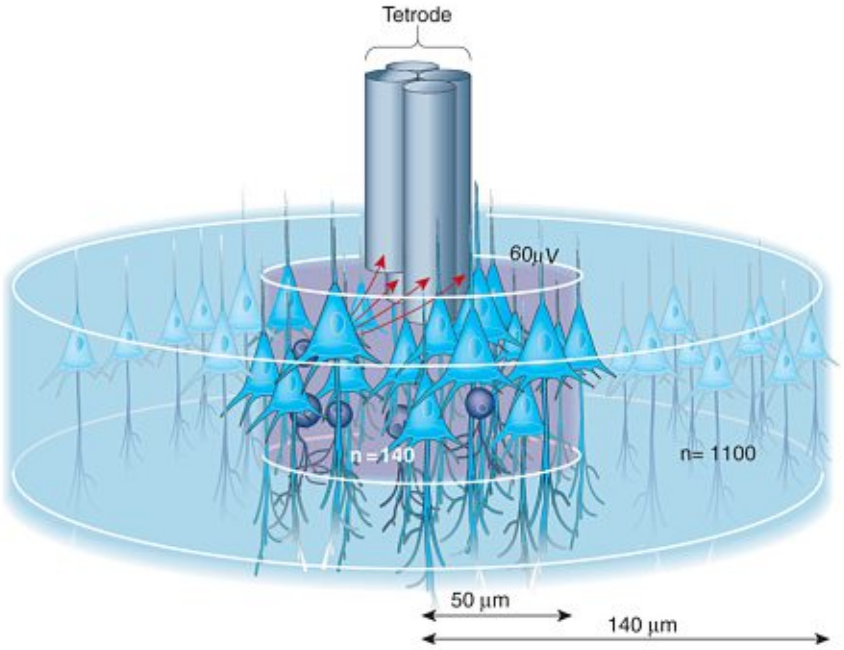


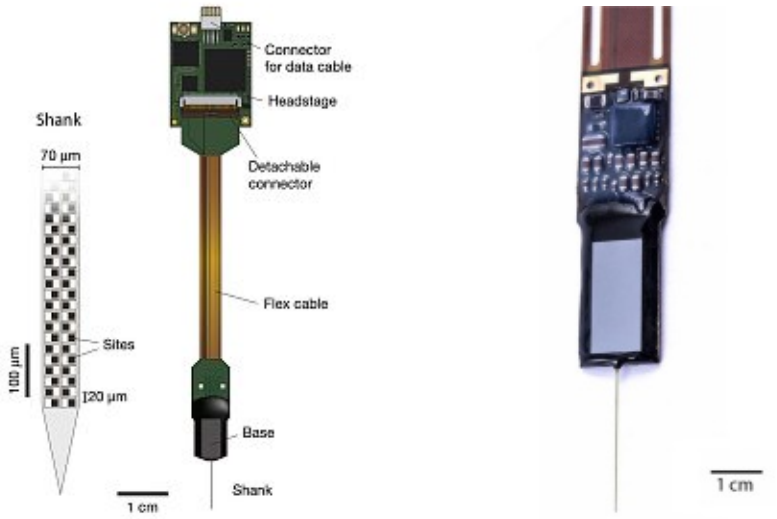
Figure 2.4: Tetrode recording schematic (adapted from Buzsáki, 2004, Figure 1)

Tetrodes use four isolated electrodes bundled together (see figure 2.4). Mounted over the recorded LFP, it is possible to also observe individual spikes, corresponding to different neurons. Due to the different locations in the 3D space around the tetrode, each neuron will have a slightly different distance to each of the 4 electrodes. This will imply that the spike of a single neuron will be observed in each of the electrodes, however, presenting a difference in amplitude. This relative amplitude of each neuron to the 4 electrodes, plus the difference on the spike changes, is used to cluster the spikes of the individual neurons in the area. Due to the only 4 electrodes used, the capability of separating different neurons and tracking their firing over extended periods of time is limited.

To combat the limitations of tetrodes, dense arrays of recording sites are used. These high-density electrodes are produced using similar techniques as in IC waver production. The base of these probes is a silicon CMOS integrated circuit, with an array of low impedance recording sites (see figure 2.5a). The most advanced probes combine around 1000 recording-sites with preprocessing and digitization. (*Home* 2020)

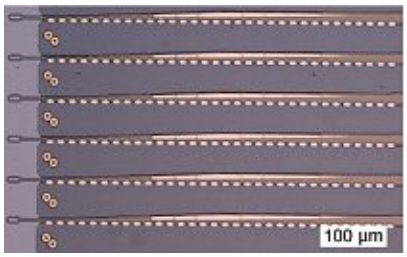
By combining multiple of these probes up to 3000 individual electrodes can be implanted into a rat's cortex(Steinmetz, Zatka-Haas, *et al.*, 2019). A different design to Neuropixels for high-density recordings is based on microwires, with a thickness of down to 6µm and a 20mm length (Musk and Neurolink, 2019). Each flexible wire linked up to 32 individual electrodes together (see figure 2.5c). By combining multiple of these threads it could be shown, that up to 3072 individual electrodes can be implanted into a rat's cortex (Musk and Neurolink, 2019).

To enable real-time processing of the recorded data each individual signal has to be amplified and digitized. By using an array of on-chip analog-to-digital converters capable of processing a large number of input signals, the very faint neuronal signals of <10 µV can be sufficiently amplified and converted. These circuits are placed on a head-stage (see figure 2.5). The amplification and digitization circuits are kept close to the signal recording sites, since the original signal is very weak and has to be amplified and encoded for reliable transmission over longer distances, such as to the acquisition board. This means that, currently, it is possible to measure 3072 different electrode inputs and distinguish 19.300 signals per second on these electrodes which would allow sampling $6 * 10^7$ signals, which in theory could be individual action potentials. This is much reduced by the need for online processing of data and spike detection, which reduces it to around 0,35 Hz per electrode (max . 1075 action potentials) (Musk and Neurolink, 2019) (Steinmetz, Aydin, *et al.*, 2020)

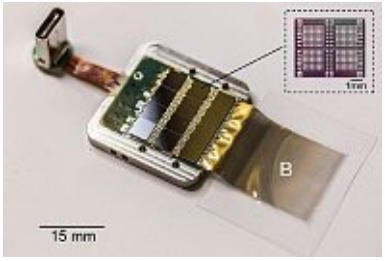


(a) *Schematic of silicon probe (adapted from (Jun, Steinmetz,* et al.*, 2017))*

(b) *Neuropixel Silicon Probe (adapted from* Home *2020)*



(c) *miniature electrode threads (Musk and Neurolink, 2019)*



(d) *on the chip analogy-to-digital amplifier (adapted from Musk and Neurolink, 2019)*

Figure 2.5: Different types of high density recording electrode and head-stage designs

The head stage connects to an acquisition board. It can receive input from multiple head-stages and precisely synchronizes time with incoming signals from each of the channels of the connected head-stages. It also enables the synchronization with additional experimental hardware such as behavior controlling devices to link electrophysiology recordings with behavior. There are multiple commercial and open-source head-stages available on the market, one of the widest used is from the company Intan (*Intan Technologies* 2021)

To post-process the signal, a reference ground is also necessary, which, as with the patch-clamp recording technique, is usually placed over the cerebellum.

Most of the current probes attach via a mechanical connection to a micromanipulator, which is used to lower the probe into the brain.

Due to the much higher density, signals from potential individual neurons will be picked up by more unique sites than on probes with smaller densities. This aids the algorithm in separating the signal, and will enable the detection of more clear separated individual neurons to be recorded from.

2.4.3.1 Spike Sorting Algorithms

With the advancements in recording hardware and the steady increase in the number of recording sites of silicon probes, the necessity for faster-performing spike sorting algorithms arose (Pachitariu *et al.*, 2016). Aided by the tremendous progress in machine learning, most of the current algorithms perform much more reliable and faster spike sorting and clustering of individual neurons out of the recorded signals (Pachitariu *et al.*, 2016).

A reliable, fast, and autonomous way to sort raw recording data is vital for tracking a large population of neurons from ever-growing numbers of single recording sites in silicon probes.

The first step of such spike-sorting-algorithm usually consists of applying a bandpass filter to isolating the spike frequency range (300-600 Hz). To Further remove noise, the average signal of recording sites not directly adjacent to the site in question is deducted from the signal. This is possible due to the high correlation of noise across multiple sites. The averaging algorithms exclude adjacent sites due to the small differences in spike signals for neighboring electrodes. (Jun, Mitelut, *et al.*, 2017)

Different approaches for the next steps exist. Since Kilosort was used for this thesis and is currently one of the fastest and most reliable spike-sorting algorithms the detailed steps of this particular algorithm will be explained in detail.

After denoising the spike waveforms are still overlapping due to the simultaneous firing of multiple neurons at the recording sites and due to multiple recording sites picking up the same spikes. The next step is to determine a spike waveform template for each individual spike source and to compare these to all the other detected spikes in a group of recording sites during a narrow timeframe to detect the same spikes via wave-form matching. To detect the best representation of the waveform, the one with the highest amplitude from all matching

sites is used and transformed via principal component analysis (principal component analysis (PCA)) in a lower representation. (Pachitariu *et al.*, 2016)

The, from the PCA resulting, new base vectors are used to robustly detect spikes originating from the same neuron over a specific time window. It is necessary to account for noise-induced changes and gradual electrode drift, which lead to a change in waveform and would, if not taken into account lead to the not correct classification of the spikes based on the template. (Jun, Mitelut, *et al.*, 2017)

To localize the neuron, the time window has to be at least smaller than the refractory period, but sufficiently large to pick up the time delay at all the sites that record that particular spike. Typically each spike is picked up by 5 to 50 channels, depending also on the topology of the used probe. After matching the waveform over multiple sites, and considering each time delay, the location of the neuron can be estimated. (Jun, Mitelut, *et al.*, 2017) (Pachitariu *et al.*, 2016)

Furthermore, drift is a common problem especially for freely moving animals but is not such a dominant problem in head-fixed animals. To counterbalance a gradual drifting amplitude the algorithm keeps track of the maximum amplitude and its change to neighboring sites for multiple neurons, which would indicate a shift in the shank position relative to the brain tissue. (Jun, Mitelut, *et al.*, 2017)

2.5 Animal Behavior and Executive Functions

To actually decode the neural code the necessary step must be to link in-vivo electrophysiology recordings to behavior.

For questions targeted at the more complex behavior of an organism, it is almost impossible to clearly create a controlled individual stimulus for the neuronal circuit of interest. This implies that a more holistic approach has to be used. Observing the neural activity of a wider array of neurons, while having controlled trials that can be repeated without a significant variance, is the basis of statistical pattern recognition from the neural data. Because organisms interact with their environment on a multitude of levels, while at the same time holding motifs and goals, linking behavior and brain activity is especially complex.

Yet, despite the complexity, many breakthroughs have been accomplished in the last decades and behavior analysis seems to provide a helpful tool for neuroscientists to build models to understand the neurophysical correlation of sensor input, computation, and motor output.

2.5.1 Rational Agents and Goal-Directed Behavior

Borrowing from computer science and particularly from artificial intelligence, each organism can be described as a more or less rational agent, which can sense its environment and act upon it. A rational agent in this regard means an agent who seeks to maximize its reward

function given a performance metric. Based on its internal model of the world and the current world state, it chooses the action with the highest probability of the optimal outcome (Stuart Russel, 2020, Chapter 2.2). Any agent that makes decisions in such a way can be considered rational. Although we sometimes hold conflicting goals and have completely wrong models of the world and seem to act in a way to sabotage our objective, we humans can be considered as the most complex rational agents.

There are several steps and layers to a rational agent. The most basic layer is that the agent as a separate, but embedded entity can sense the world and act on the world. The complexity of the environment and how much of the necessary factors are observable by the agent range from very simple like a linear maze to very complex the real world we live in as the most complex (Stuart Russel, 2020, Chapter 2.1).

To make the optimal decision for the agent, the agent has to have an internal model of the world. In the case of mammals, these are believed to be represented by the circuits and synaptic weights in the neocortex and especially the prefrontal cortex (Padoa-Schioppa and Conen, 2017). This model represents the agent's knowledge of how the world evolves based on a given state, and how each of its actions affects this evolution (schematic view of model-based agent see figure 2.6). The more precise this model, the smaller will be the difference between the expected outcome and the actual outcome. The constant update of the model based on the difference is to learn and evolve and adapt to changing environments.

To update the model the agent must first sense the current world state. Only elements of the world that are involved in potential actions of the agent are of relevance to it. It must also be able to sense the change its actions have on the environment. In a real-world scenario, it is not always possible for an agent to get all the necessary information from the sensor percepts. Such an environment is only partially observable (Stuart Russel, 2020, Chapter 2.3).

Partial observable environments always introduce a level of uncertainty. Large evidence points to bayesian circuits being employed by the brain to infer the posterior probability of a feature not fully observable (Doya, 2011, Chapter 3).

For a rational agent, it is not sufficient to have a correct model, and to be able to observe or infer the world state; a goal driving the actions is necessary. This so-called goal-directed-behavior is the root of all intelligent agents. The goal can be a specific world state or it can be controlled by a utility function, which weighs different combinations of world states (Stuart Russel, 2020).

2.5.2 Decision Making

Decision making is arguably the most influencing factor of goal-directed behavior. For intelligent agents that have to engage in goal-directed behavior to survive and strive, the quality of life often is governed by the decisions they make (Stuart Russel, 2020, Chapter 2.4).

The process of decision making can be divided into three stages:

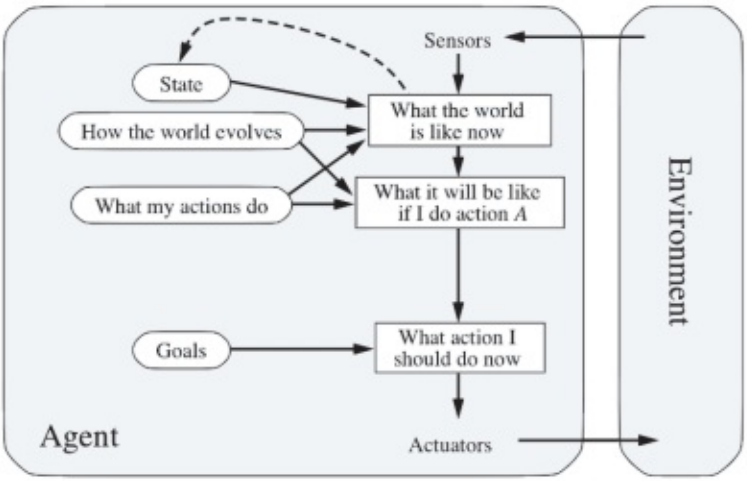


Figure 2.6: Model based / goal based agent diagram (Stuart Russel, 2020, Page 52) arrows: flow of information, solid arrwos: direct influence of next values, dotted arrwo: updat of previous values, round rectangles: background information by the agend of the environment , rectangles: processing steps

The first stage is assigning values to the available options, which the agent will be faced within a complex scenario. As decisions are mostly based on a variety of factors and have to mediate between often conflicting options, a crucial part of the decision process in determining specific values describing the probability of an outcome for a decision. In its simplest way, the values represent a sum of predicted rewards for a specific option (Stuart Russel, 2020, Chapter 2.4). The values assigned to each potential option are derived from external perceptual cues and an internal model learned from past experience. The more complex the organism and the task, the more abstract the model involved. The decision process also necessitates a prediction of how the world will evolve and what effect the chosen action will have on the world. Based on this, the agent can predict the future state of the world and from that derive how satisfiable this potential state will be for him based on its internal or external evaluation criteria. (Stuart Russel, 2020, Chapter 2.4). In real-world environments as well as in standard behavior tasks, the abstract internal models of the world involve a high amount of uncertainty. Bayesian statistical methods provide an approach for dealing with probabilistic outcome distributions and uncertainties. (Passecker *et al.*, 2019)

The second stage is comparing the earlier assigned values, based on the desired outcome and individual preferences. This comparison is based on evaluation criteria, based on a goal or a performance metric. The aim of that process is to choose the option, that will lead to the maximization of the received reward. To further complicate the step, in complex scenarios a reward maximization function can have local maxima, and it might be beneficial for the agent to accept short-time losses for a long-term higher reward (Stuart Russel, 2020, Chapter 2) (Passecker *et al.*, 2019).

While the first and second step focuses on current decisions, the third step adapts the overall

decision model and the ruleset that governs the allocation and comparison of values. By comparing the anticipated outcome of a decision with the actual outcome, the ruleset can be refined. In addition, the agent can also switch between different rules based on changing external conditions. Depending on the situation, different evaluation criteria may apply and only minor perceptual or internal changes can lead to significant changes consequently actions. (Padoa-Schioppa and Conen, 2017).

As outlined above, decisions mostly rely on internal models. Learning such models from experience is to constantly update the internal model based on the deviation between the predicted outcome and the actual outcome of a decision. Due to inherent uncertainty, such feedback is not always available nor accurate (Stuart Russel, 2020, Chapter 2.3) (Passecker *et al.*, 2019).

In the particular case of a neural circuit, encoding these stages, it must fulfill certain criteria to be able to meet the demands faced by decision making in real-world scenarios. It must allow flexible adaptation to the current circumstances but needs certain stability over a wide variety of situations. This dichotomy is necessary for archiving long term success and planning for future outcomes and individual preferences. The observed correlation between neural activity with behavior, form compelling evidence that there actually are such underlying neural representations in the prefrontal cortex (Press, 2020, Chapter 8).

What already has been found is that the prefrontal cortex and especially the medial prefrontal cortex the orbitofrontal cortex play a major role in decision making. The prefrontal cortex is the epicenter for decisions as it orchestrates executive functions differentiating among conflicting decisions and goals. (Padoa-Schioppa and Conen, 2017), (Karlsson *et al.*, 2012), (Passecker *et al.*, 2019)

Still, the underlying network dynamics for rule and strategy switching necessary for adapting to changing external conditions or changes of internal goal and valuation are poorly understood. Nonetheless, it could be shown that the firing rates of individual neurons in the prelimbic and cingulate cortex are correlated with behavior. In particular, firing patterns shift between subclusters of neurons while different rules are learned and applied. (Durstewitz *et al.*, 2010)

To further complicate the approach, real-world decision tasks are not directly guided by external perceptual cues about correct or incorrect decisions. The qualitative level of the outcome of a decision is not binarily distributed but can occupy a wide variety of results (Abbott *et al.*, 2017).

2.5.3 Risk

Stochasticity is inherently present in any natural environment. It ranges from sensor precepts about the state of the environment over the probability of outcome, to feedback and reward. This all leads to a high degree of uncertainty for an agent during the decision process. Risk describes the relation between the probability of an outcome and the magnitude of the anticipated potential reward. A high-risk scenario would comprise a small probability but a

high reward. Interestingly, it is especially prominent the temporal property of the reward. It has been shown that humans, and several other animals, often value decisions that lead to an earlier but smaller reward more favorable than decisions that would result in a later but higher overall reward (Kable and Glimcher, 2007).

Overall, it is safe to say that decision scenarios involving risk are a subset of all decision processes and involve situations, where the agent knows the different probabilities for potential outcomes, in contrast with situations where the probabilities are not known. This separation is based on the notion that, for a risk-taking situation, the agent knowingly engages in risk-based decision making (Burke and Tobler, 2011). For example, it could be shown that acute tryptophan depletion, leading to low availability of serotonin levels in the brain, made monkeys tending towards more risk-seeking behavior compared to normal serotonin levels (Cools *et al.*, 2005). Derived from those findings, it is the argument that serotonin modulates the subjective value of risk (Burke and Tobler, 2011). Also, inhibiting specific neurons, via optogenetic stimulation during unfavorable outcomes of risky decisions, led mice to continuously engage in risky decisions and not adapt their behavior (Passecker *et al.*, 2019).

2.5.4 Modeling Decision Processes

As we have seen, the brain uses an internal representation of the world to model its decision processes. Due to the high complexity of real-world scenarios, a lot of variables are either not directly observable, or only stochastically with a low signal to noise ratio. In addition, each measuring process has inherent noise, which further increases the uncertainty. Finally, the signaling processes in the brain also introduce noise, although they employ noise reduction techniques in the form of decoding and encoding to compensate for the inherent noise of the synaptic transmission. Most of the introduced noise follows a Gaussian distribution due to its thermal origin. Variability in the measurement of the actual real-world parameter, is usually normally distributed due to sampling from the population (Doya, 2011, Chapter 2 and 3).

All these factors lead to a very uncertain representation of the world. In combination with an ever-changing environment, the internal model has to be updated constantly for an accurate prediction of future actions. A very powerful tool to study and model this ever-changing environment is Bayesian inference.

Bayesian methods offer a complete method in modeling these uncertainties. The Bayesian approach provides a framework on how to update a parameter based on prior knowledge and new evidence. Thus, the model of the world should be updated based on new knowledge of how the world works. It provides a way to calculate the respective probabilities of different models, given a measurement. By maximizing for the posterior probability the model that best explains the data can be chosen (Doya, 2011, Chapter 3).

$$Posterior = \frac{(Likelihood \cdot Prior)}{Evidence}$$
$$P(H \mid D) = \frac{P(D \mid H) \cdot P(H)}{P(D)} \tag{2.1}$$

H=Hypothesis, D=Data
Posterior $= P(H \mid D)$*=probability of hypothesis H given data D*
Likelihood $= P(D \mid H)$*=probability of data D given hypothesis H*
Prior $= P(H)$*=probability of hypothesis H*
Evidence $= P(D)$*=probability of measured data D*

The posterior represents the probability of the hypothesis chosen a specific model under the condition of the measured data. The posterior is not directly measurable, but can be derived from the product of the likelihood and the prior normalized by the evidence. The likelihood stands for the prediction of the measured data by the given hypothesis. The evidence describes the probability of the hypothesis, which intern follows a specific distribution derived from the overall model. The Evidence in the denominator is a normative factor to ensure that the sum of probabilities of all hypotheses is not larger than 100%.

After the action has been made and the new environment is observed, the brain is believed to use bayesian mechanisms to again calculate the probability of each of the hypotheses and update the internal model according to deviations between predicted probability and posterior probability (Doya, 2011, Chapter 11).

The Bayesian approach helps in two ways, on the one hand, it enables more accurate interpretation and decoding of the neural data such as spike trains derived from single-neuron activity. On the other hand, it helps to build more accurate models of decision processes and understand how such models could be represented in neural circuits in the brain. Both hopefully lead to better computational frameworks and algorithms, finding new technological applications and bring us closer to the ultimate goal of building an artificial thinking machine seeing eye to eye with the human brain.

2.6 Animal Models in Behavior Experiments

Animal models have been long used for neuroscience experiments and are still the predominant source for recorded data especially for electrophysiology. On the one hand, models of simpler organisms such as Caenorhabditis elegans, with a much less complex nerve system, allow for an easier analysis of the function. These simple models provided experimentalists with a research platform to gradually try to understand the mechanisms of the nervous system and still are widely used for specific experiments. As the understanding and models became more

detailed, and the questions shifted from mere functions to a broader nature, more and more complex animal models are required (Chen, 2020).

Neuroscience research is dominated by a very small number of species used as animal models, with over 50% of all research studies using rodents as animal models. Besides rodents, the fruit fly, zebrafish, roundworm, and non-human primates are the predominant research platforms. This is largely influenced by the readily available tools and techniques for these animals based on significant genetic research, which is simply not available in that quality and quantity for other animal models. (Chen, 2020)

2.6.1 Rodent Models in Behavior Experiments

Rodent (in particular *Mus musculus* and *Rattus norvegicus*) models have played a significant role for neuroscience. Rodents are particularly easy to handle while still being a relative high-level organism.

Rats historically have been the predominant animal for research, but over the last years, a shift towards mice has taken place. This shift is largely driven by the development of genetic knockout mice and other techniques to manipulate the genetic code of experimental animals (Ellenbroek and Youn, 2016). The development of optogenetic manipulation of neurons is one of the tools mostly driving the adaption of mice. There are still some clear advantages of rats over mice. Rats are much easier to handle as they tend to be more comfortable with human contact and surgery tends to be easier due to larger brain size. (Ellenbroek and Youn, 2016)

Rodents are used in several different ways for electrophysiology experiments. Based on the constraint of the animal these can be clustered into four subtypes, anesthetized experiments, awake experiments, and freely moving and awake but head fixed.

Anesthetized rats have been historically used due to the major benefit of providing a very stable recording option. The animal is anesthetized and the skull is rigidly constrained relative to the probes by a fixture. This provides the ability to use micromanipulators to precisely guide the probes and hold them in place over the recording period. For low-level questions, the data measured by in-vivo recordings from anesthetized animals might suffice, but for higher-level functions a fully awake and behaving animal is required.

An awake and moving animal may provide a much better research platform for higher-level behavior since the animals can be trained to perform certain behavior and react to external cues. In addition, the recorded neural activity can be correlated with behavior, a feature certainly lacking while the animal is anesthetized.

The usual approach is to abstract a behavior, related to the question of interest, to a simpler level and then, design a task around this behavior and train the animal to learn this task.

To obtain measurements of the neural activity where background activity can be filtered out and unique activity can be potentially correlated with stages in the task, it is beneficial to

have the animal perform the task many times over while recording. Although this approach holds true for the task of this thesis, there are other behavior tasks, such as anxiety and fear tasks, where a repeated exposure is not desirable.

Experiments performed with freely moving rodents provide a multitude of challenges mostly related to recording stability and variability of the environment. It is much more difficult to obtain stable and precise recordings from freely moving animals compared to head-fixed animals due to relative and absolute movements of probes and animals. A further weakness of freely moving behavior tasks is the inevitable and often unobserved variability between the individual trial runs in a recording session.

2.6.1.1 Head Fixed VR Setups

To further standardize the experimental conditions, a switch from freely moving test subjects in the physical task environments to virtual reality setups is highly beneficial. In such tasks, animals are head fixed for more stable recording and behave in a virtual environment to gain greater control over sensory input and motor output (Schwarz *et al.*, 2010).

Virtual reality setups consist of some sort of visual projection that can be on a computer screen or a monitor projecting on a flat or spherical surface. This type of technique was introduced in the early 2000s into human experimental psychology and quickly was adopted by neuroscience and animal behavior science in particular (Hölscher *et al.*, 2005).

Initial difficulties adapting VR setups for rodents were related to the large visual field used for perceptual exploration of the environment by rodents.(Hölscher *et al.*, 2005). Most of the earlier and current virtual reality systems emphasized to simulate the movement of the rodent. Such systems comprise a treadmill whether in one or two directions. The animal would be restrained in a harness and placed over the treadmill or spherical treadmill so that it can move on it. Via a closed-loop system, the movement direction of the animal would be used to update the present environment on the screen (see figure 2.7). The behavior often involves an active feedback mechanism with a positive reward conditioning, such that a reward can be delivered via a tube placed in close contact with the animal (Thurley and Ayaz, 2017).

These systems can simulate a maze and a behavior task, for example, walking through a corridor and receiving a reward at one end of the corridor.

Still, the range of stimuli is significantly limited compared to real-world tasks, which provides potential limitations when research is aimed to recreate a controlled environment as close as possible to the real world (Thurley and Ayaz, 2017). Nonetheless, VR setups, especially with head-fixed mice, provide a unique possibility for stable recordings, paired with precise behavior experiments. Many of the latest large datasets of multi-channel silicon probe recordings of mice are from such systems.

Although significant advances have been made over the last years, the underlying neural mechanisms governing the decision processes are far from being understood. To unravel the mystery that the brain still presents, It is necessary to integrate technological tools with

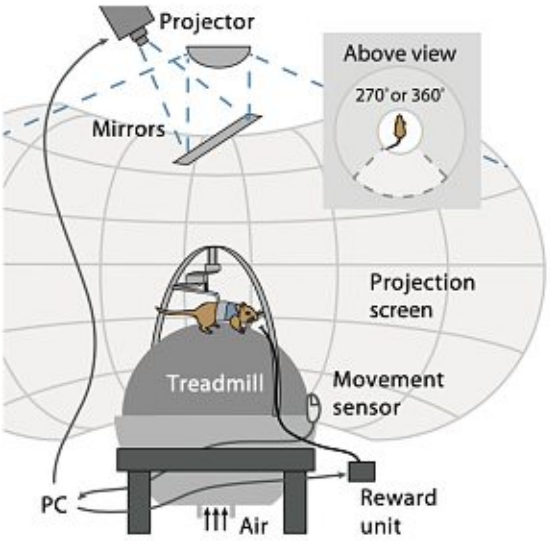


Figure 2.7: Virtual reality system for head-fixed rodent on a spherical treadmill, image inside of sphere created by overhead projector (adapted from Thurley and Ayaz, 2017, Figure 1)

animal models, behavioral tasks, and recording techniques to further probe the mechanisms of decision making.

Nowadays, the challenge for research interested in the underlying neuronal mechanisms of decision making, is to design behavioral tasks that mimic the governing conditions in the real world while being reproducible and focusing on a specific part of the complex process. They also have to allow for recording and manipulation of single neural activity via in-vivo electrophysiology, two-photon microscopy, optogenetics, and other methods. (Abbott *et al.*, 2017)

2.7 Gamble-Task

Gambling tasks with changing reward probability are frequently deployed to investigate flexible decision making and the neural activity, especially in the prefrontal cortex, correlated with such behavior. Gambling tasks provide an experimental framework for decisions guided by inner models of probabilistic distributed outcomes. Gambling, in this aspect, referees decisions without external cue-guidance in an environment with probabilistically changing rewards (Passecker *et al.*, 2019). The combination of perceptual cues with past experience and the necessary adaptation of choice based on changing rewards are necessary for overall reward maximization, and closely represents the natural condition an agent would face in real-world situations. The challenge of an experimental gambling task is to provide probabilistically changing rewards while still keeping the parameters relatively fixed, thus allowing for a standardized behavior training and experiment. (Abbott *et al.*, 2017)

Gambling tasks have a rich history both in psychology and neuroscience. They have been recently adapted for rodents, and especially for in-vivo electrophysiology. Perhaps the most famous and widely used gambling task framework is the Iowa-Gambling-Task, originally introduced by Bechara and colleagues at the University of Iowa in 1994 as a task to "simulates real-life decision-making in the way it factors the uncertainty of premises and outcomes, as well as reward and punishment" (Bechara *et al.*, 1944). The concept of the task was based on the participant having to choose a card from a deck of playing cards. Afterward, they got feedback based on the choice in the form of a big reward (100$), a small reward (50$), or a penalty (-250$). The reward and punishments were assigned to types of cards respectively, with an additional probability for the cards assigned the two rewards to sometimes, instead, hold a penalty. The subject learned this mapping over time, but the assignment can be changed at any time by the experimenter forcing the subject to adapt its internal representation of the model. The combination of uncertainty of outcome and changing rewards represents a very close implementation of the above-described paradigm. The Iowa-Gambling-Task plays an important role in understanding decision-making in humans as well as underlying disorders such as anxiety and addiction (Brevers *et al.*, 2013).

The Iowa-Gambling-Task was adapted for rodents by (van den Bos *et al.*, 2006) as a multi-arm bandit task. The task is a maze consisting of two choice-boxes with different amounts of reward and different probabilities of obtaining the respective rewards. The difference was between a low quantity high probability reward side (short time reward maximization) and a high quantity low probability side (long term reward maximization) (van den Bos *et al.*, 2006). Since then a lot of different adaptations have been used with rodents.

2.7.1 Passecker Task

Most of this thesis builds on the initial work of (Passecker *et al.*, 2019) and a later adaption of his gambling task for mice in a virtual reality system. To both give the reader the necessary understanding and help to span the arc to the newly developed setup of this thesis, the experimental task and the behavior and neural findings of trained animals will be outlined.

The task is a y-maze adaptation of the Iowa-Gambling-Task (see figure 2.8). There are two terminal arms, one associated with a low quantity but a certain reward (safe arm) and the other with a high quantity but probabilistic reward (gamble arm); there are no punishments. The rewards are pellets (TestDiet) dispensed by an automatic pellet feeder. A session was divided into three blocks, regarding the probability of receiving a reward at the gamble-arm: a block of 12.5% (maximization of reward at saving arm), other of 25% (no clear rational arm preference for maximization), and another of 75% (reward maximization at gamble arm).

Each trial consists of three stages: Run 1, Run 2, and Reward. "Run1 was defined as the episode between the home arm sensor activation and the division point. Run2 was defined as the period between the division point and the reward sensor. The reward episode was defined as the period between the reward sensor activation and the grabbing of the animal." (Passecker *et al.*, 2019) The animal is grabbed by the experimenter to manually place it back at the beginning of the maze (Passecker *et al.*, 2019).

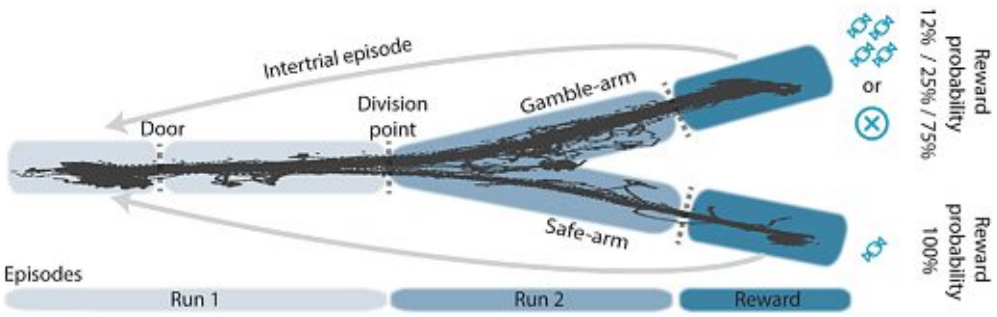


Figure 2.8: Schematic of gamble-task in y-maze with probabilistic rewards, gamble-arm large reward with three different brobabilites 12%, 25% or 75%, Safe-arm always a small rewared, Run 1 to 3: stages of each trial (Passecker *et al.*, 2019, Figure 1A)

The habituation and training of the animal for the task consisted of several stages. Initially, the animal received the same amount at both sides. Once the rats were running towards the end of the maze and were comfortable with the manual replacement of their position to the start of the maze, the probabilities to the gamble-arm were introduced. To help the animal realized the difference between the arms, after each probability changes, the safe arm was blocked for 8 trials. In order to correct for pattern learning of these "forced runs", they were regularly skipped at a later stage. At the next training stage, a door at the beginning of the maze was introduced, forcing the animal to wait for two seconds after each run. The training was completed and the animal was advanced to the recording if a clear preference for the optimal arms in two respective blocks was present and, additionally, the rat had to perform adequately at all three blocks on three consecutive days. The training on average took around three weeks.

The movement of the animal was tracked by LEDs on the head-stage of the rat and a camera placed at a birds-eye view.

In vivo electrophysiology recordings were obtained with 16 electrodes, which were lowered into the prelimbic cortex and kept stable via a head-stage attached to the rat's skull.

To investigate the effect of non-reward trials on the internal model guiding future decisions, optogenetic silencing of the prelimbic cortex at no-reward trials was performed. To enable optogenetic stimulation, channelrhodopsin was expressed by means of virus injection targeting GABAergic neurons.

2.7.1.1 Neural Findings:

A significant correlation between the increase of firing rate and no-reward at the gamble-arm was observed for the majority of recorded neurons in the prelimbic cortex. In addition, firing rate of these neurons correlated to the three different probabilities for receiving high reward. Firing was higher during stages where the choice of the gamble-arm did not yield a higher

overall reward or was not clear (12.5% and 25%) and lower for a statistically clear preference for the gamble-arm. Such a correlation could not be observed for the safe-arm.

The firing rate of the same cells also provided a prediction for strategy changes (switching of arm) for future choices of subsequent trials. Cells again fired with a higher rate if an arm-switch was performed. Thuss it seamed, that the firing rate of no-reward activated neurons is not a clear predictor for trial by trial choices.

The data also showed that the firing rate of the potential choice predicting neurons are independent of the outcome of the previous trial. This suggests that the firing rate during the experience of no-reward is independent of reward-prediction-error. If they would represent the prediction error between the internal model and the feedback, a clear correlation with past experience and thus previous trials would be expected.

The optogenetic silencing of these neurons at the reward phase was used to further confirm the correlation between no-reward trials at the gambling arm and future choice. Optogenetic stimulation was targeted at GABAergic interneurons that released GABA and hyperpolarized postsynaptic neurons, therefore silencing the target area. It could be shown that this optogenetic stimulation to suppress spiking during the reward feedback phase, in a trial where no reward was received at the gamble-arm, led to a tendency of the rats to gamble more. Even during the stages with 12.5% and 25% reward probability a suppression during no-reward led to a significantly higher number of choices to go for the gamble-arm resulting in a decreased overall reward.

Based on the above-described results, the authors interpreted the correlation between firing rate and negative feedback trials as a neural representation of part of the internal model of the animal guiding decision making. They argue that this evidence supports the hypothesis of these neurons guiding decision making especially in situations where the animal is faced with no clear decision scenarios and has to rely on an internal model for choosing the safe option or the risky option.

They also link the firing rate correlated with strategy switching after no-reward gambling trials with the notion of regret. Regret, in this case, is linked to behavior change after reward evaluation linked to a potentially better alternative for a different choice. It can be reasoned, that the animal can infer that it would have received a small but guaranteed reward if it had not chosen the gamble-side, and thus regrets its decision to go for the risky gamble-side, if it did not indeed receive a large reward. The firing rate can be linked to the switch of the internal model to a different decision ruleset.

2.7.1.2 Shortcomings and problems:

Although the data from the electrophysiology recordings of the freely moving rats led to a high number of individual neurons and stable recording during the whole behavior, it was nonetheless accompanied by some major problems and had the potential for optimization.

The first challenge was to fully control the task environment. Although the stimuli that a rat can experience were limited, they still accounted for some variability over each trial. For example, each rat had to be manually picked up and carried to the start of the maze after each trial, which inevitably led to variability. Also, the animal had to run along the maze and thus had options to move differently each time. "This type of motor-related variability can affect neuronal firing in the prefrontal cortex as neurons in that area are also tuned to movements and trajectories" (Malagon-Vina *et al.*, 2018).

Also, related to the design, relying on freely moving animals produces problems in the area of the electrophysiology recording. The movement led to stability issues and needs a stable head-stage. The head-stage and the wires connecting to the acquisition board can lead to a restricted head movement of the rat.

To further standardize the experimental conditions, and overcome these problems, a switch from freely moving to a head-fixed awake and behaving animal task setup was a potential solution. This limits the movement of the subject, reducing the artifacts originating from movement (Schwarz *et al.*, 2010). However, it brought some major challenges.

The biggest challenge was that rats are not suitable for head-fixed awake setups since they are able to break their own neck in a head fixated setup, due to their strong neck muscles. Although some labs have succeeded in sufficiently training rats to also perform head-fixed recordings on them (Schwarz *et al.*, 2010). Thus such a switch would also necessitate a switch to mice as animal models.

On the other hand, the use of mice would also open up a wider range of genetic tools and provide the potential of leveraging the much greater availability of tools and knowledge (Carandini and Churchland, 2013). Furthermore, there is a wide variety of databases available and, in particular, the All Brain Atlas, GENSAT, and the Mouse Brain Architecture Project provide an in-depth understanding of gene expression in the specific regions of the mouse brain and how these regions are interconnected. Transgenic mice breeds are better available, which in combination with tools such as optogenetics or two-photon-microscopy enables a wide range of research to target specific neurons to monitor or manipulate their activity.

An additional huge advantage is the stability of the animal's brain, which enables highly accurate measurement with optical and electrophysiological recording methods. The fixed spatial position allows for the deployment of micromanipulators to hold and move the electrodes into the subject's brain.

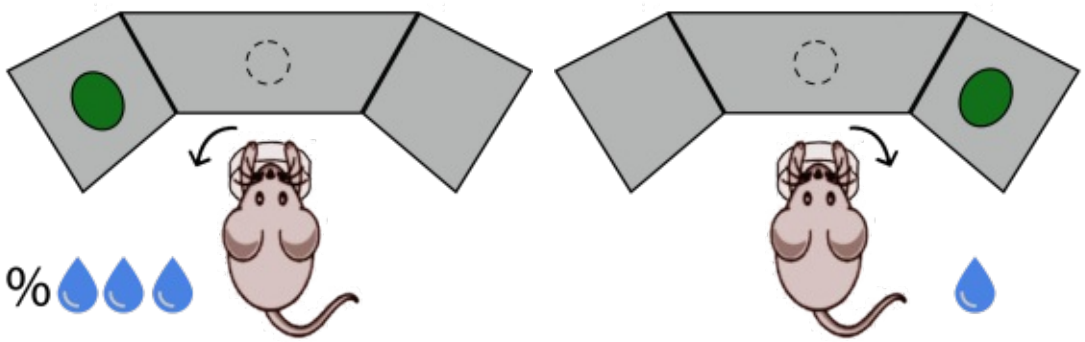
2.8 VR Setup Phenosys

The initial proof of concept and implementation of a compatible task in a virtual reality setup was done by Jian Gang and Aron Kőszeghy in the division of cognitive neurobiology at the Center for Brain Research of the Medical University of Vienna, under the mentorage of Professor Thomas Klausberger. It is based on the "steering-wheel-setup" first introduced by (Burgess *et al.*, 2017) at the Cortexlab from UCL. The basic concept relies on a steering wheel

coupled to a rotary encoder and a stimulus screen developed with the company Phenosys (*PhenoSys JetBall-TFT Virtual Reality System for Rodents* 2021). The mouse can move a visual stimulus on the screen by turning the wheel with its front paws (see figure 2.9) . This allows for different sorts of decision-making task implementations while still allowing for head fixation, since the mouse does not have to move. The reward was administered in liquid form via a tube directly in front of the mouse's mouth so it could lick the dispensed drops. The large and small rewards depending on the gamble-side or safe-side were modulated in the form of the amount of liquid given.

The gamble-task designed by (Passecker *et al.*, 2019) implemented in a y maze was adapted in such a way, that the left or right arm of the maze was mapped to the left or right side of the screen setup.

The mouse can indicate if it chose the safe-side by turning the wheel to the safe-side of the screen e.g. to the left or choose the gamble-side by turning it to the opposite direction e.g. right see figure 2.9



(a) *Gamble-side, probabalistic large reward* **(b)** *Safe-side, always small reward*

Figure 2.9: Concept of decision making in virtual-reality gamble-task, gamble-side probabalistic large reward reward with prbabalitiy blocks 75%, 25% and 12.5%, safe-side 100% small reward

The task consisted of several stages (see figure 2.10). Each trial started with an initial waiting time, where the screen was black. After that time, the stimulus was presented on the screen, but could not be moved by turning the wheel. To check for an engagement of the mouse within each new trial, the wheel had to be static for a preset period of time after the stimulus was presented. If it failed and movement was detected, the trial was aborted and counted as failed. If the wheel did not move for the necessary duration, a tone indicated to the mouse the start of the open-loop period, where it could move the stimulus by rotating the wheel.

If the stimulus on the screen reached a certain position on the left or the right, it was counted as a choice for that side and the reward stage was entered. Depending on the choice and probability, a small reward, a large reward, or no reward was administered, and after an inter-trial time, a new trial was initiated.

The probability of the gamble-side follows the same distribution as the task by Passacker (Passecker *et al.*, 2019) during the high probability block it is 75% likely that the mouse received a large reward, the next lower block is 25%, and the lowest 12.5%. With the safe-arm there was a slight difference in that also a probability was introduced, although it was kept at 90%. This served a similar function as the forced runs from the Passacker task. The concept of the very low likelihood of not receiving a reward at the safe-side, was that if the animal always plays it safe sometimes it is nudged by not receiving a reward to switch from exploitative behavior to explorative and check the gamble-side thus it would more likely discover a change in the gamble-side reward probability.

If the mouse did not move the wheel far enough and the stimulus did not reach a set threshold position, the trial was aborted and a new trial would start.

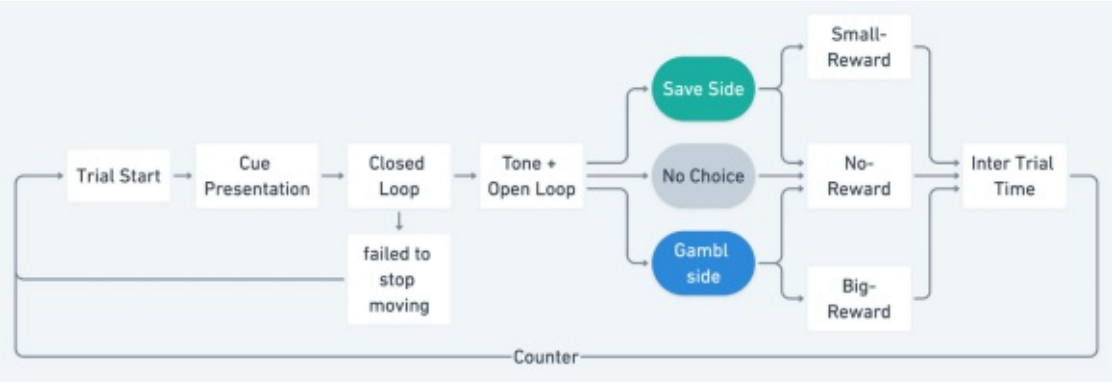


Figure 2.10: Schematic of Phenosys virtual-reality hardware setup

2.8.1 Hardware and Setup

The central system of the setup is the JetBall TFT virtual reality system (see figure 2.11) combined with the Steering Wheel by the company PhenoSys (*PhenoSys JetBall-TFT Virtual Reality System for Rodents* 2021). The system controls 6 screens, but only four were actively used for this setup. The system in its core is a state machine, with predefined state change conditions and actions for each state. It can be programmed via a proprietary visual programming language.

The mice are head-fixed and held in place by screws connecting a surgically implanted head-plate to the cranium and an adapter mounted to a stereotaxic frame by the company Kopf. The wheel was mounted on a plastic adapter plate, on which also the body of the mouse could rest (*PhenoSys Steering Wheel* 2021). The screens are always manually placed in front of the mouse with a distance of approximately 45 cm.

Reward is delivered via a tube placed at the mouth of the animal and the amount is controlled via a gravity-based flow system and a solenoid valve connected to the virtual reality controller.

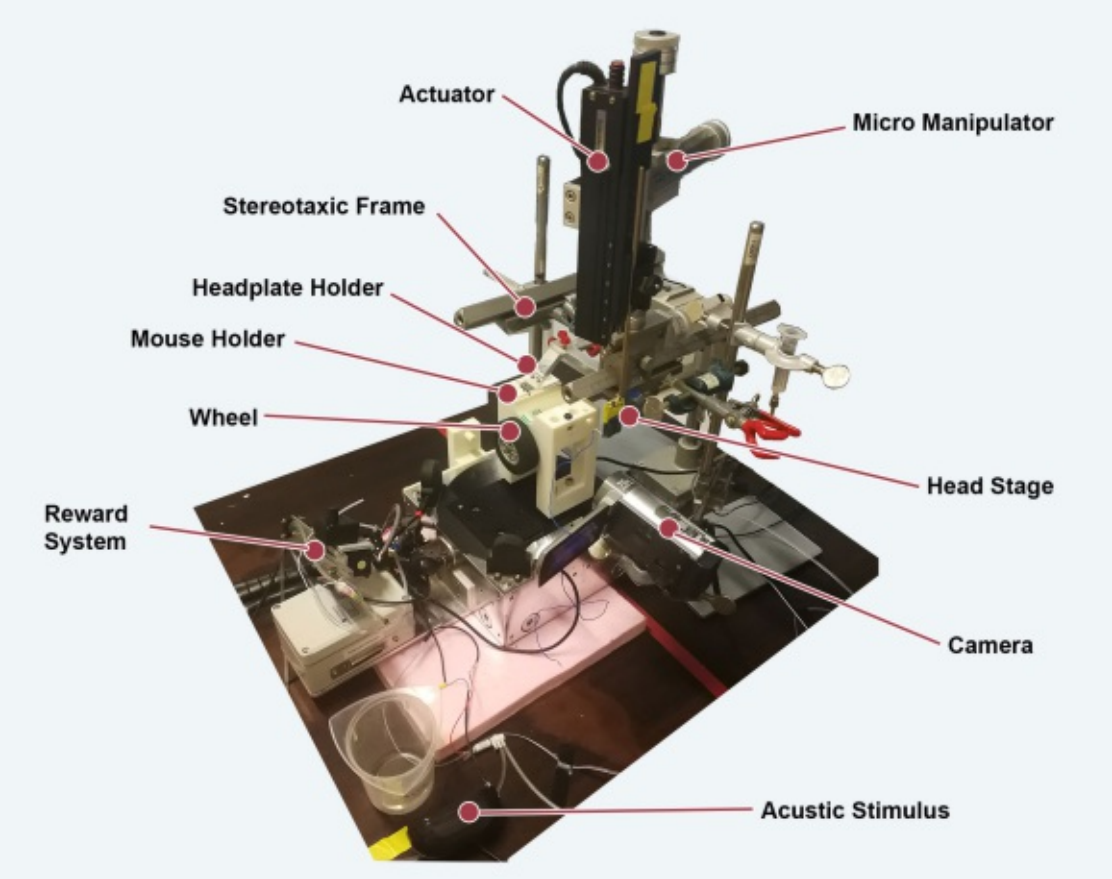


Figure 2.11: Overview of the gamble-task implemented in the first virtual reality setup

2.8.2 Habituation and Training

The animals used for the experiment are mice of the family C57BL/6, a commonly used inbred strain of laboratory mice. Choice of mice was mostly based on the availability at the research institute and proven reliability from similar studies, both from the Klausberger's lab and other labs working with in-vivo electrophysiology in mice.

Mice are bred in-house and kept in a climate-controlled environment in cages. At first, mice are kept together in groups of a few animals, to account for their natural behavior. After starting the experiments by performing a surgery to implant a head-plate, they are placed in separate cages. If not separated, mice will inflict wounds to each other during territorial fighting habits, especially to the tissue around the implanted head-plate. They are kept at a 12h/12h day/night cycle, which is synchronized with the real day/night cycle.

The actual training starts when animals reach the age of 3 months. The first procedure is to perform the head-plate implantation surgery. Then, the animal is given an adequate recovery time of around three to seven days. The actual duration of the recovery period

largely depended on the behavior of the animal during recovery and the weight compared to the initial weight before surgery.

The next stage is the habituation of the animal with the experimenter, the environment and the room where the experiment is conducted. They are brought, in their cages, into the room and held by the experimenter. Based on the confidence of the animal in the room and around the experimenter this phase usually lasts for one to three days, after which water restriction is initiated.

The restriction of water is used to motivate the animal to participate in the behavior and training. This leads to an increase in value of the water-reward during the behavior training for the animal. Water is restricted in such a quantity that the animal weight does not fall below 80-85% of its initial weight after recovery, which approximately resulted in 1-2ml of water given once a day. Food in form of pallets is abundantly made available to the animal during all stages.

Accompanying water restriction, the animal is also habituated to the head restriction on the behavioral apparatus. To familiarize it with the task, water reward is given if the mouse even slightly moves the wheel.

Over the following days, the threshold for necessary movement to receive water, independent of the direction, is gradually increased. Through this procedure, the animal learns to associate reward with the movement of the wheel. This is necessary for the later stages, during which the movement must be deliberate and also covering a wider angle of wheel movement. These requirements are designed to filter out random behavior and guarantee a deliberate continuous choice of one side by the animal.

After successfully mastering the deliberate and continuous movement of the wheel in both directions, the animal is trained to stop wheel movement between consecutive trials. A trial will not initiate if the wheel does not stop for a given number of seconds (between 1 - 3 seconds). After this pause, the trial starts, indicated by the appearance of the stimulus on the screen.

The next stage is arguably the most complex one. The animal learns to distinguish between the two directions in which the wheel is turned and the stimulus is moved, thus learning the difference between the left and the right side, necessary for the final behavior. This is achieved by only rewarding one side with water reward and switching sides during one training session.

For the final training stage not all the correct chosen sides are rewarded 100% of the time, but a probability is introduced.

Finally, the decision to advance a particular animal to the recording stage is made, if the animal displays adaptive behavior, depending on the reward probability and side, and continues probing both sides periodically for around 200 consecutive trials.

2.8.3 Electrophysiology Recordings

After the mice have cleared the final training stage, they are ready for recording. To allow for in-vivo electrophysiology recording a part of the cranium of the animal has to be removed (craniotomy), exposing the dura and the brain for the electrode shank to be lowered into the brain.

Each recording day consisted of one session, but animals are recorded several days in a row. On average animals are recorded 4 to 8 times before being sacrificed by perfusion.

For recordings, electrodes manufactured by NeuroNexus (see figure 2.12a) are used. The silicon-based electrodes have four shanks and a total of 128 to 256 recording sites. The silicon probe is attached to a RHD 128 channel head-stage by Intan Technologies (see figure 2.12b), which is held in place by a micromanipulator from Scientifica. The micromanipulator in term is mounted to a x-y translational arm attached to the same frame that is the adapter for the head-fixation. This way a very stable connection between electrode and head-plate is achieved. An Intan Technologies RHD recording system (see figure 2.12c) is used, which can connect to up to 4 head-stages and address 1024 individual amplifier channels, sampled at 20Khz.

For later analysis of neural signals in comparison with behavior stages, the Phenosys system must be synchronized with the Intan and thus the behavior with the neural recordings. This is implemented in a digital encoded form with TTL pulses from the Phenosys system to a TTL breakout board connected to the Intan recording unit. With a single BNC channels each state and its timing is directly communicated to the Intan, via a encoded message, and with the internal clock by the Intan time-stamped and saved. This guarantees that both the neural recording and the behavior data are time-stamped with the same clock for later alignment.

(a) *NeurNexus probes used with the virtual-reality setup (*SmartProbe *2020)*

(b) *Intan Head-stage (*SmartProbe *2020)*

(c) *Intan acquisition unit (*Intan Technologies *2020) with 8 connected head-stages*

Figure 2.12: Recording syste mcomponents for used together with the Phenosys system

A recording session follows a specific protocol. The animal is brought up from the animal housing room to the lab, to acclimate it. After preparation of the equipment, the animals are fixed to the adapter and the silicon protection (protecting the exposed scalp) is removed from the head-plate to expose the dura. Saline solution is used to cover the exposed tissue. The next step is to lower the connected probe into the brain. Optionally the shanks of the probe could be stained with a die to later trace the path and reconstruct it from the brain slices. For correct alignment relative to the brain, a reference point was set during craniotomy on the bregma. This reference point, together with a brain atlas is used to calculate the entry angle and target depth of the probe. With the use of an optical microscope, the probe is pre-aligned to the bregma.

To account for the friction between brain tissue and probe the lowering speed has to be limited, as to not excessively damage the brain tissue. After the approximate target depth is reached, the speed is further reduced. With the help of the recording software, already displaying recorded LFPs from the electrodes, the final adjustments in-depth to find the optimal position is made. Signals of potential individual neurons are used to guide these last micro-adjustments, with an emphasis to capture as many as possible well-defined spike signals on multiple channels.

After the final position is reached, at least 15 minutes are given to the tissue to resettle and the damage to be processed. Damaged cells during lowering the probe release glutamate to signal microglia, acting as macrophages. These neurotransmitters first had to be metabolized to not affect the recordings.

The behavior part of recording sessions usually lasts around half an hour after which the probe is retracted and the opening of the cranium sealed with silicon.

Depending on the stage the animal is kept for another recording session or it is advanced to perfusion.

2.8.4 Perfusion

To sacrifice the animal and to stabilize the brain tissue, the mice are perfused with a fixative solution based on paraformaldehyde and a buffer. The animal is anesthetized with urethane. The rib cage of the anesthetized mouse is cut open to expose the beating heart. After inserting a cannula, connected to the perfusion pump, via the left ventricle into the aorta, the right atrium is cut and saline solution is passed through the cannula. This first pass of saline solution will remove the blood from the vessels. After some minutes and by controlling the change of color of, mainly, the liver and other internal organs (from a reddish to a white color), fixative is pumped into the animal to fix the tissue.

Between 50 to $100ml$ of fixative solution is pumped into the mouse. The effect of the fixative should be evident by the hardening of the tissue and color change (to yellow) of paws. The brain is piled out of the skull and put into a fixative bath for 24 hours at 4°C. After that the fixative must be buffered. Now the brain can be stored until slicing at 4°C.

Methods

This section will describe the methods used for the analysis of the data from the previous system. As described in the preface, the methods for designing, developing and implementation, as well as testing of the new system, will be integrated in the respective sections describing the new system.

The methods chapter is divided into three parts, first the methods for behavior analysis used both for data from the previous setup and behavior data recorded with the new setup. The following section outlines the complete analysis pipeline for electrophysiology recordings as well as describes in detail the spike sorting approach, the exploratory data analysis steps and the statistical data analysis methods. The third section is dedicated to the methods used for designing and implementing the new system.

3.1 Synchronization of Data

The first step in evaluating the Phenosys system, is to investigate, whether the event time synchronization to the Intan recording system works as designed and is reliable. The synchronization mechanism, as well as the algorithm for alignment, is described in detail in the project work of the author focusing solely on the synchronization and development of a protocol for the new virtual-reality system. The systems communicate via a single TTL chanel with 5V pulses. Each state of the behavior task is encoded with a different length of a TTL pulse, and the start of the event state is synchronized directly by the start of the TTL pulse. The encoding scheme used is outlined in table 3.1: system.

A major problem encountered with this approach was, that sometimes TTL pulses are not recorded by the Intan system, and therefore the synchronization is misaligned. In addition, not all the stages of a trial are encoded and synchronized correctly, since the duration of the synchronizing TTL puls is not exact, leading to a smearing of the encoding to neighboring events. In particular, the TTL pulse encoding for a specific event not always has exactly the required length, therefore can be interpreted as two events. Furthermore, the relative time between both the Phenosys and the Intan experience drift over the duration of a trial of

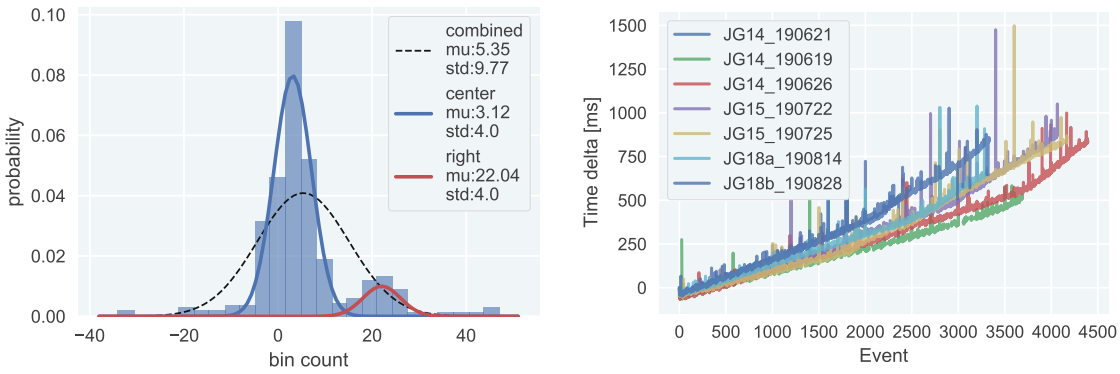
Table 3.1: Scheme for event synchronization via single-channel TTL pulse duration and behavior event

Name	Length $[MS]$	Description
start	1	Trial start time
cue	2	Cue presentation on screen starts
sound	3	Sound indicates start of open-loop
open-loop	4	Open-loop
right-reward	5	
right-noreward	6	
left-reward	7	
left-noreward	8	
no response	9	Mouse did not move wheel far enough
Inter trial	10	Time for reward consummation
end	11	Trial end time

between 10 and 40 milliseconds (example see figure 11a), which also have to be taken into account for alignment tests.

To ensure proper synchronization, which is of utmost importance for the next analysis steps, custom algorithms are used, both in Matlab and in Python. The algorithm loads and extracts the event times and event types both from the binary file from Intan and the Comma-separated values (CSV) file from Phenosys. Since the event and relative event times don't align and not all events are present in both systems, the algorithm searches for the best matching alignment. The algorithm first aligned complete trials by their start and end time as well as trial length. Event times for trial start and stop as well as intermediate events experience a relative drift, with the internal clock of the Intan running faster than the Phenosys. Also, trial lengths do not align for all of the trials and have to be manually shifted to correct for alignment (example see figure 3.1).

Events within the thus aligned trials are adjusted to best fit, with the event type from the Phenosys prioritized over the event type of the Intan, since the Intan experiences inaccurate event transmission due to varying TTL pulse length (example see figure 3.2).



(a) *Trial length difference distribution* **(b)** *Time drift*

Figure 3.1: Synchronization and alignment of event times between Intan and Phenosys

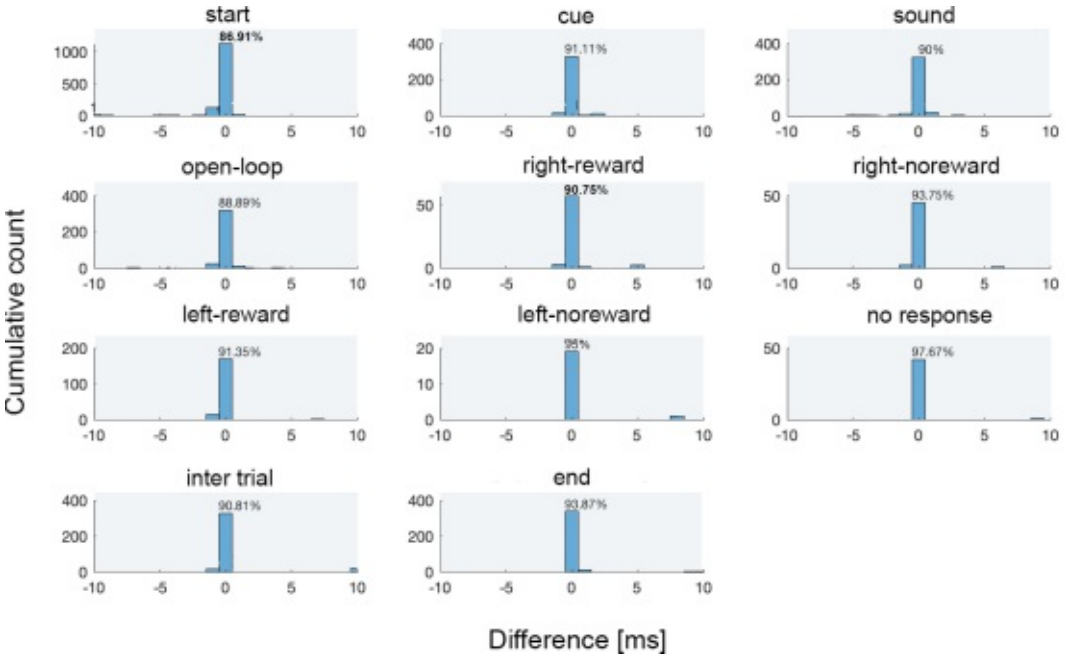
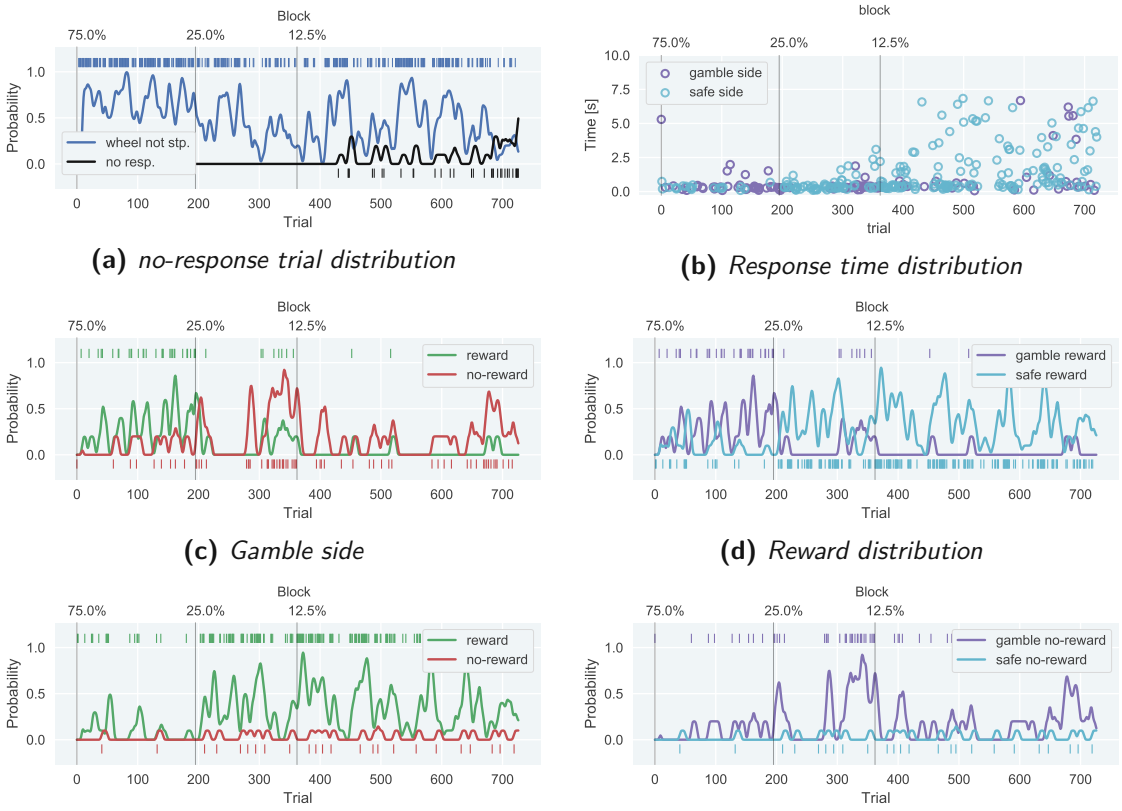


Figure 3.2: Differences in events between Phenosys and Intan, each subplot represents the distribution between correct and incorrect syncronized events in all trials, with 100% representing a perfect synchronized session

3.2 Behavior Analysis

Behavior analysis is performed in two different ways. Analysis of the behavior data from the Phenosys system (previous system) is based on a script (see figure 3.3 for example) developed by Michael Lagler and adapted and modified during this thesis. Behavior data from the new system is analyzed with a different script (see figure 3.4 for example), that was developed

by Hugo Malagon, specifically for the new system. It is oriented on the one of Michael Lagler, but improves on some issues, mostly based on the complexity of use. Behavior data is analyzed before neuronal data will be analyzed. The aim of analyzing behavior, before analyzing neuronal data is to understand how the animal interacted with the task and use this knowledge to select accurate representing sessions for the further analysis pipeline.



(a) *no-response trial distribution*

(b) *Response time distribution*

(c) *Gamble side*

(d) *Reward distribution*

(e) *Safe side*

(f) *No-reward distributionn*

Figure 3.3: Script for analyzing behavior data from the previous system (Phenosys)

Behavior analysis is divided into several parts. The first thing that needs to be explored is the distribution of successfully completed trials; trials where the animal failed to stop the wheel for the required period of time; and uncompleted trials, where the animal failed to respond on time by turning the wheel to the necessary angle.

To account for the undesired behavior of the subjects not indicating a response, such trials should be excluded from further analysis. But for the behavior it is necessary to look at the continuous development of events, thus the period of analysis should consist of consecutive trials. To account for both, trials at the beginning and the end of each session, that significantly exceed the average trial length are removed from further analysis. Focusing on the successive part between these cuts. These trials are in the successive steps of the pipeline referred to primarily as selected trials.

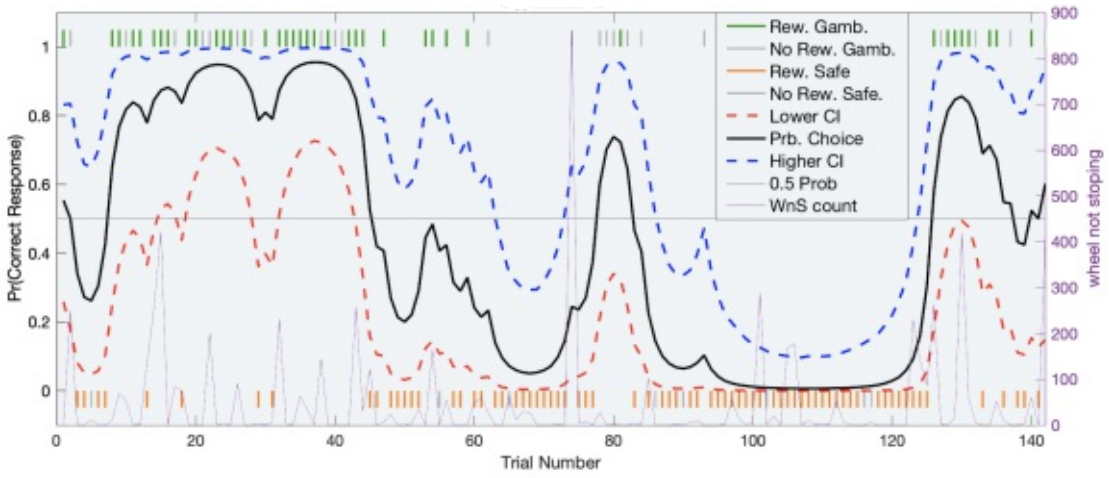


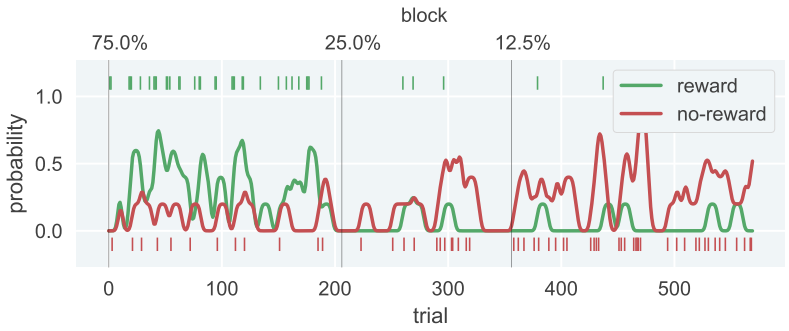
Figure 3.4: Script for behavior analysis from the new system
The title describes the current phase of the animal training. The ticks above and below on the x-axis display the reward and no-reward encountered by the animal both on the gamble-side(above) and the safe-side. The mean choice probability (of going to the gamble arm) with a Markov-Chain using a Monte-Carlo-Method described in (Smith *et al.*, 2004), as well as the 90% confidence interval. The probability of the trials, where the mouse failed to stop the wheel is plotted in magenta on the lower x-axis.

For a detailed investigation of the animal's encountered reward and no-reward both for the safe-side and gamble-side, plots modified from the initial script by Michael Lagler are used. By plotting both rewards and no rewards at both arms it is possible to observe the difference in the respective choice based on the probability block of the gamble-arm. This in, combination with the indicated changes for the probability blocks by the dotted line is used to analyze the expected rationality of the animal's behavior.

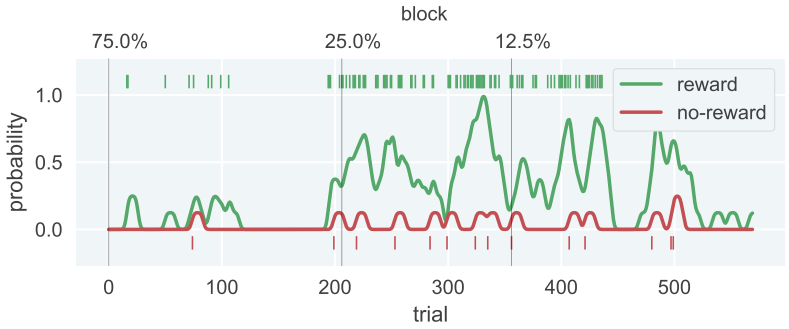
These plots are mostly used for determining whether a session is fit for neural analysis. The main emphasis for this decision lies with two behavior indications. On the one hand, the animal has to display rational behavior, seeking to maximize its reward by choosing most of the time the appropriate side depending on the reward probability of the current block. On the other hand, it still has to display adaptive behavior at changing reward probabilities, therefore it also has to regularly sample the not desired side.

To visualize the distributions of specific subsections of trials for all analyzed sessions, boxplots are used.

Individual sessions are visualized, the mean and variance is displayed by boxplots with mean, lower and upper quartile as wheel as whiskers (delimiting quartile 1-1.5 * Interquartile range and quartile 3 + 1.5 * Interquartile range)



(a) *Gamble side*



(b) *Safe side*

Figure 3.5: Choice behavior for a recording session, dotted lines, indicate a change in reward probability for the gamble-side. The probability is the event count convoluted with a gaussian for a sliding window of 10 events, pluses mark the exact event for each trial.

3.3 Neural Data Analysis

The neural data analysis is based on three stages: first spike sorting, second exploratory data analysis and third statistical data analysis (see figure 3.6).

3.3.1 Spike Sorting

To analyze the spikes of single neuron recordings, the raw data has to be post-processed. As described in chapter multi-unit recording, waveform matching has to be applied to sort the LFP signals according to individual neurons.

Several methods are currently available with KlustaKwik (Rossant *et al.*, 2016), Kilosort (Pachitariu *et al.*, 2016) and Spike2 (*CED Spike2: Spike Sorting* 2021) amongst the most widely used software packages.

For this thesis, Kilosort is chosen to perform waveform matching and automatic sorting, due to its high processing speed of large raw LFP recordings from a high number of electrodes.

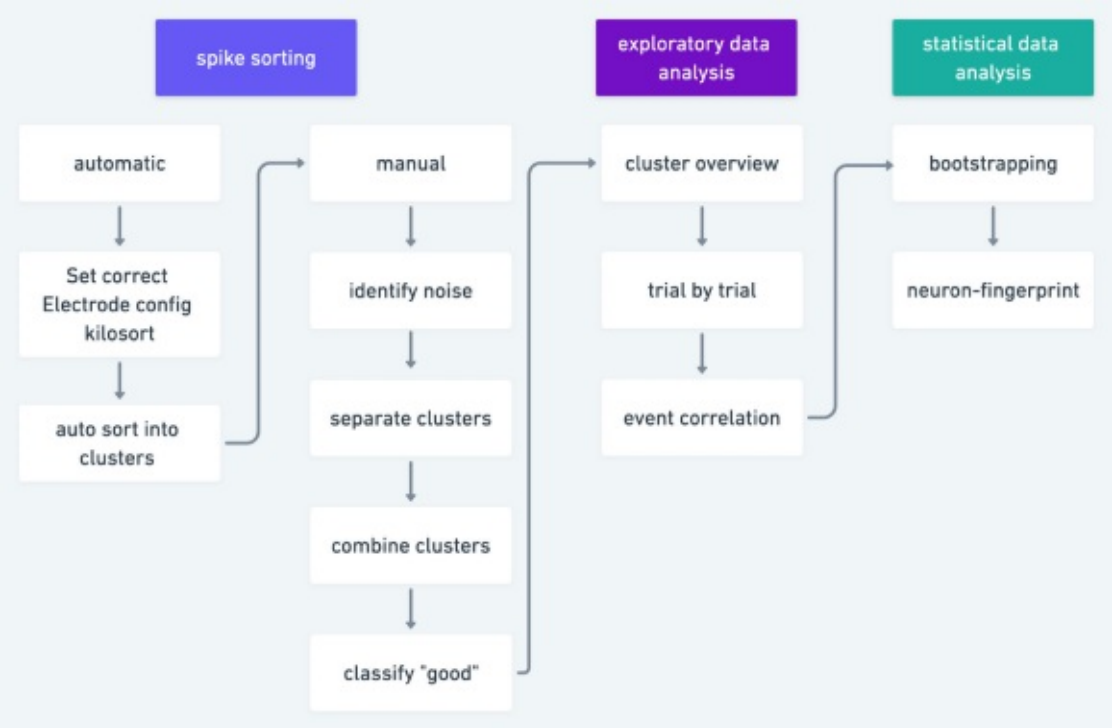


Figure 3.6: Data analysis pipeline overview

The software makes use of low-dimensional approximations performed on a GPU using Nvidia Cuda and thus is able of a high level of parallel processing.

The used version of Kilosort is run on MATLAB version 2019a and uses the Parallel Computing Toolbox, Signal Processing Toolbox, and Statistics and Machine Learning Toolbox. The deployed hardware was an Intel i7 six-core 3.6Ghz processor and a Nvidia GTX1080 GPU.

The matching electrode layout configuration and a sampling rates defined, and the high-pass filter threshold is set to 300 Hz. The algorithm takes less than 3 minutes to sort raw data from a file of around 2Gb.

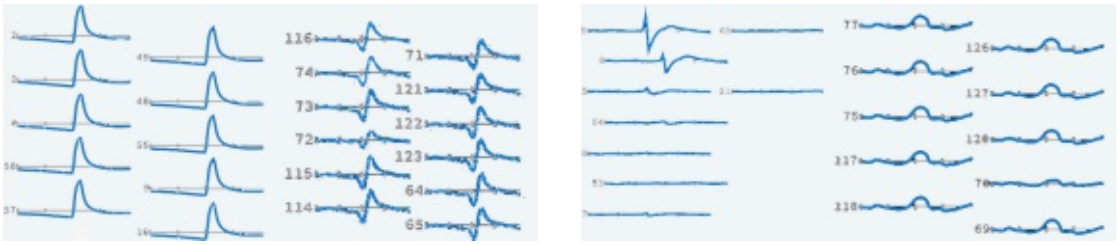
Although the level of correct matched waveforms by Kilosort, and thus identified putative individual neurons and respective spikes is very high, manual post-processing and clustering still have to be performed for an adequate level of accuracy for later analysis.

Kilosort not only matches spikes to putative neurons but also classifies these neurons (also called clusters) into four categories: good, noise, Muli unit activity (MUA) and unclassified. Clusters, classified as noise mostly contain waveforms not resembling a spike waveform of physical neurons. MUA stands for multi-unit-activity and contains potentially the spikes of not a single neuron but multiple neurons, which could not be completely separated.

The task of manually sorting the clusters was performed with the python software package Phy (*Introduction - Phy* 2021).

Manual sorting is done in four stages: first removing noisy clusters, second separating MUA, third combining clusters, and finally marking good clusters

Noisy clusters are identified by spike waveform (some examples of good and noise waveforms see figure 3.7).



(a) *Good spike waveform* **(b)** *Examples of waveform classified as noise*

Figure 3.7: Waveform examples for manual spikesorting

The second manual post-processing step is to separate MUA into individual neurons where possible. This is necessary, because MUA clusters cannot be used for further analysis, since a potential correlation between neuronal activity and behavior activity from MUA clusters is not necessarily due to causal correlation. This is due to the problem that a single neuron, distributed across multiple MUA clusters, can lead to a much higher number of falsely correlated clusters, although only the individual neuron would display such a correlation. Basically, only the autocorrelation of the neuron, divided across multiple MUA clusters is detected.

If a MUA cluster is split, only a single new cluster resulting from the MUA is kept. This step is made to account for the above-described phenomenon of autocorrelation correlation. Separation was mostly based on observable sub-clusters in the feature view part of the Phy GUI.

The third step, the opposite of the second, is to combine clusters that potentially describe the same physical neuron. A high emphasis is placed on the electrode view. Combination is only performed for clusters belonging to the same shank, since the same neuron would excite only recording sites in the near vicinity.

Also due to the refractory period, a single neuron can only fire at the same time, with a short silence after the spike, thus the combined cross correlogram of two clusters containing spikes of the same neuron still has to have a low autocorrelation.

The fourth and final step is to, again, go through all clusters and finally decide on good and MUA cluster categorization. Also, the number of total spikes for clusters to be classified as "good", has to be over the threshold of around 100-120 spikes for the complete session (for comparison vary active neurons have more than 50k spikes for the same session).

Although Kilosort2 and the manual sorting steps result in a high accuracy the clusters are putative neurons.

3.3.2 Exploratory Data Analysis

Next, the data, resulting from the previous spike sorting step is explored, using mostly spike trains - event plots, Peristimulus time histogram (PSTH) plots of firing frequenz in combination with spike trains to inspect the individual neurons resulting from clustering. Only neurons previously group as good clusters are analyzed.

As a further quality check, isi distributions over all neurons indicated as good clusters are plotted.

Event plots are derived by selecting all spike times for a specific neuron, that fall into each trial, aligning the times per trial to the start of the trial and stacking the spike trains (time of spikes) for each trial vertically. Trial length is plotted in seconds.

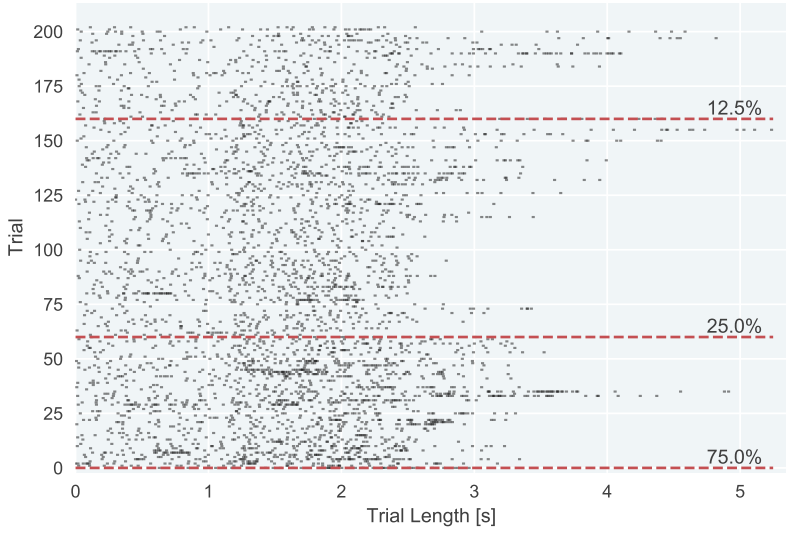


Figure 3.8: Spike train plots for selected trials stacked, the red dashed lines marke changes in the reward probability for the gamble side, with the following probability marked

The reward probability for the gamble-side for each block and the trial, where the block changes is marked with the red dotted line

To visually inspect neurons for a potential correlation between an event in the trial and spike rate, PSTH plots are used (see figure 3.8). Spike times for each trial are derived the same way, but not the complete trials are displayed, but only spike falling into a window around a specific event in each trial, spike times relative to the event time in each trial, +- the window are selected and aligned for each trial relative to the respective event, by subtracting the event time. These aligned spike times are stacked per trial horizontally and frequency is derived by binning and counting over all trials.

The behavior is set up in such a way, that every stage until the end of the open-loop, has a fixed length. These events are fixed relative to the start of each trial. The alignment of the reward and the events preceding the reward onset, is not possible over all trials. This is

per design, since the animal can determine the open-loop duration, if it moves the stimulus faster or slower to the threshold position, and triggers the next stage. The time between the trial start and the reward is also not the same for all trials, since the animal is given more time to consume the large reward.

Due to this possible trial-by-trial difference in length of the event windows, the alignment of the events, with the above-descibed approach will result for different events, in different distributions (see figure 3.9).

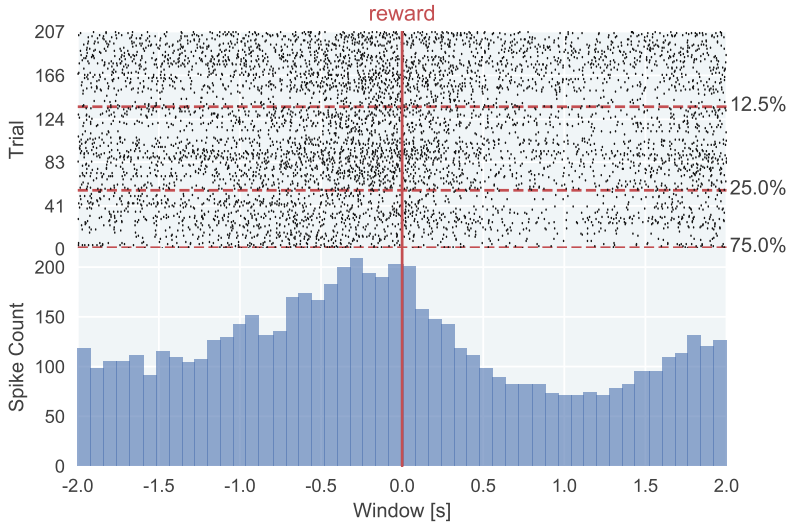


Figure 3.9: Spike train and PSTH plots for all trials aligned on reward event, the red vertical line marks the reward or no-reward event, the horizontal lines mark the change in probability blocks with the following probability.

Both plots show the spike train for each trial for the window around the respective event in each trial in the upper part, and the histogram for the distribution of the spike times in the lower window.

3.3.3 Statistical Data Analysis

To analyze potential visually identified correlations between spike rate and a specific event in all trials or in a subselection of trials, several statistical methods are used.

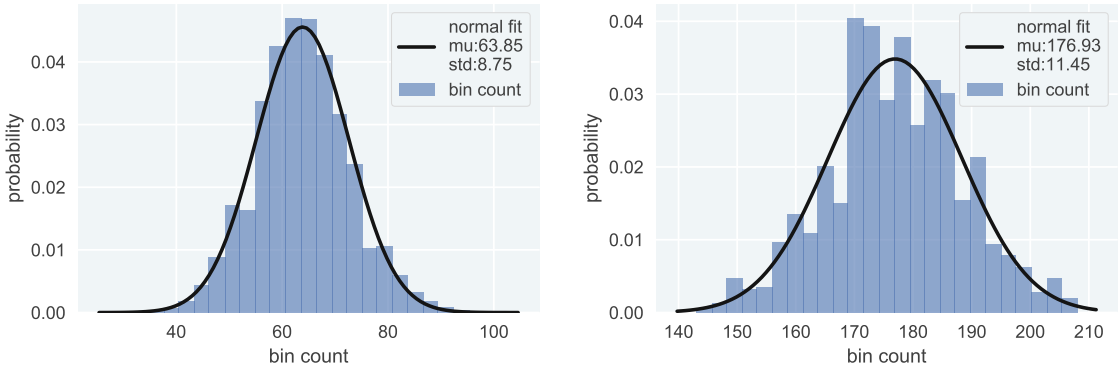
3.3.3.1 Bootstrapping Aproach

The bootstrapping approach is used, to distribution difference, between the mean firing rate for reward aligned spike trains, compared to randomly sampled spike trains. The bootstrapping approach is a technique of random sampling with replacement to compute distribution metrics, such as variance, mean and confidence intervals by random sampling from observed data.

Since the distribution of the measured spike rate is not known, bootstrapping allows to determine this distribution.

The first step in determining the overall distribution of the spike rate for each neuron over the selected trials is to randomly sample from that data. Normal distributed random time sampled from each trial are drawn and the spikes in a window of plus-minus 2 seconds around these random points are sup-selected. This is done for 1000 random samples per trial for each neuron. The spike times are binned into 50 discrete intervals for each trial.

Both the binned spike count over all the random windows from one trial and overall trials should follow a normal distribution. To check for normality, the Shapiro Wilch test is performed on randomly selected samples (example of distribution see figure 3.10).



(a) *JG14_190621 all random windows all trials* **(b)** *JG18b_190722 random windows single trial*

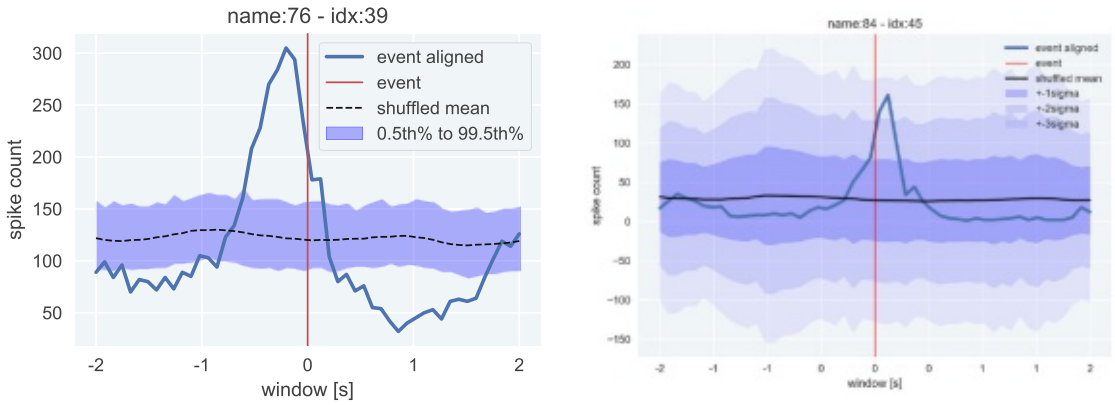
Figure 3.10: Example distribution for randomly select sample trial and scaled binned spike count for all 1000 random windows for the selected trial

Although the spike rate from the randomly sampled windows across one neuron, as well as for a single trial follows a normal distribution, the spike data is not normally distributed across all recorded neurons. This is expected, since different neurons display different spike rates.

The next step is to use the distribution metrics from the previous bootstrapping approach, to identify whether the spike rate of putative reward coding neurons are significantly linked to the reward event. In the first general approach, no subselection of trials is performed, but all trials are used both for bootstrapping and for the distribution of the reward aligned spike rate. Two different levels for the confidence interval are chosen, the lower level is for 90% of the data and stretches from the 5th to the 95th percentile and the higher level is for 95% of the data and stretches from one sigma around the mean.

3.3.3.2 Fingerprint of Neurons

To further break down each neuron to a lower-dimensional feature space, a "fingerprint" for each neuron is calculated. The fingerprint is based on the binned spike rate aligned with the reward event for each trial compared with the 99.5th percentile and the 0.5th percentile. The



(a) *90% confidence interval* **(b)** *95% confidence interval*

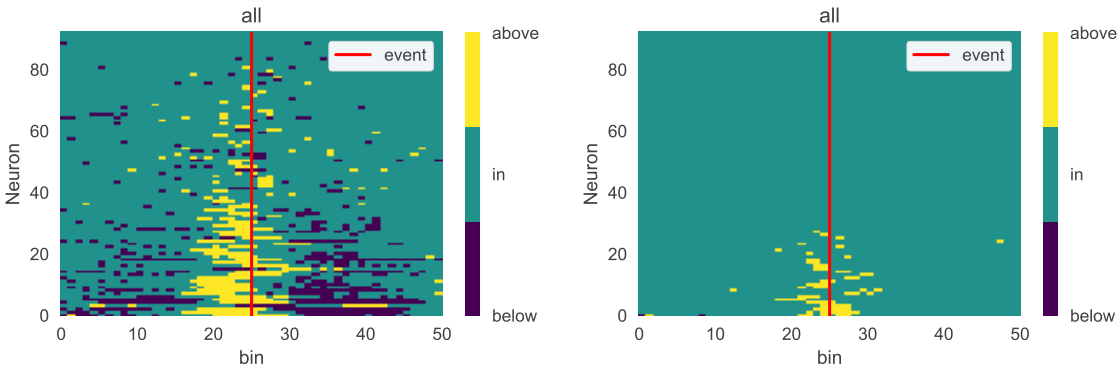
Figure 3.11: Distribution of random samples compared to distribution of reward aligned mean spike rate for a window of +-2seconds binned for 50 bins of 80ms duration. For the higher confidence level, also 2 and 3 sigma confidence levels are plotted. The headline indicates the reward event, to which each trial is aligned.

binned spike rate is considered significant, if it is outside of 90% percent of the randomly sampled data. If it lies above the confidence interval, the fingerprint is +1, if it lies within it is 0 and if it lies below it is -1.

To visualize the fingerprint for all neurons in a session, a 2d color map is plotted for all neurons, that have at least one bin above the 90% and 97,2% (2 sigma) confidence interval respectively

A 2D accumulated scatterplot for the sum of bins that are above or below the confidence interval displays the neurons that have been linked with a significant correlation of spike rate is used to further visualize the difference between sessions.

There are multiple different subselections that are of interest, rewarded trials vs not rewarded and safe-side vs gamble-side and each respective combination thereof. In addition, neurons can be discriminate depending weather their spike rate significantly increases before the reward event, across or after it. For a neuron to be counted as significant, at least two blocks in a 5 block window before, across or after the reward event have to fall into a range higher than the 90th percentile. To visualize the number of neurons for each of the above mentioned discriminating factors and to see the distribution thereof, a bar chart diagram is used.



(a) *90% confidence interval* **(b)** *97.2% (2 sigma) confidence interval*

Figure 3.12: 2D color map visualizing bins above, or below of the respective confidence interval obtained with the bootstrapping random sampling approach. Blue marks bins, that are below the interval and yellow bins, that fall above, the red line denote the reward event.

To test for significant difference between two different subselections whether for probability blocks, occurrence of bins, or reward-no-reward and gamble-safe of the distribution, a chi-2 test is used.

Results: Data Analysis

This section presents the first part of the main body of work of the thesis. It is the first part of the two results chapters. This part focuses on the analysis of the Phenosys virtual-reality system system, comprising synchronization between Phenosys and Intan electrophysiology acquisition system, as well as the data analysis of training sessions recorded by Gang with the Phenosys system. The data analysis part is divided in behavior analysis and neural data analysis. Sessions follow a strict naming convention, the first two letters are the initials of the experimenter followed by the number identifying the animal and if both hemispheres were recorded a letter, the 6 digit number is the date of the recording.

4.1 Behavior Analysis

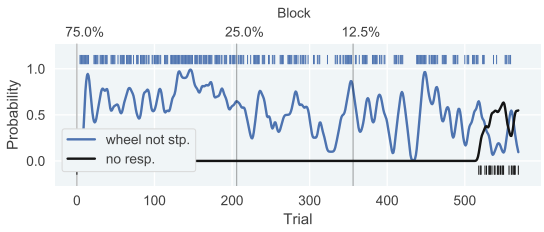
Table 4.1: Overview of all behavior analyzed sessions, [1]:blocks stand for the consequitive trials were the reward porbability for the gamble-side follows eather 75%, 25% or 12.5%, [2]: wheel not-stipping trials

id	blocks[1]	tot. trials	wheel ns[2] trials	no resp trials	selected trials	reward	no-reward	gamble	safe	gamble re-warded	safe re-ward	gamble no-reward	safe no-reward
JG14_190621	75, 25, 12.5	569	303	26	208	151	57	82	126	37	114	45	12
JG14_190619	12.5, 25, 75	1143	865	9	257	203	54	88	167	53	150	35	17
JG14_190626	12.5, 25, 75	1297	939	43	274	211	63	73	197	34	177	39	20
JG15_190722	12.5, 25, 75	936	498	58	271	207	64	44	197	15	177	29	20
JG15_190725	75, 25, 12.5	899	427	76	184	113	71	128	56	63	50	65	6
JG18a_190814	12.5, 25, 75	649	250	29	307	234	73	134	56	78	50	56	6
JG18b_190828	75, 25, 12.5	726	352	33	203	150	53	79	124	39	111	40	13

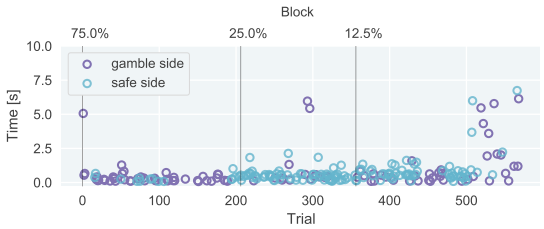
The major part of the behavior analysis is focused on the optimality of choice by the animal and the adaptability of decision of the animal based on the changing reward probability. With the notion of a rational agent, the mouse is expected to use both reward and no-reward as inputs guiding decisions. Based on the knowledge acquired during training and the familiarity of the animal with the task, an initial phase of explorative behavior, sampling both sides to detect the initial probability block of the gamble-side is expected. The analysis steps outlined in methods is used.

The trials for all of the analyzed sessions display a similar distribution across, trial length, ratio of total trials to trials, where the animal did not stop the wheel, and where the animal failed to respond during the open-loop. The uncompleted trials, in regard to wheel-not-stopping, are more or less evenly distributed along with the complete session. One such example of the distribution of such failures can be seen in **??**.

Furthermore, the most noticeable behavior concerning no-response trials, found in nearly all animals, is that no-response trials seem to be clustered near the end of the session starting usually after 200 trials. In addition, the response time of the animal also varies on a trial by trial basis. Interestingly, the response time and no-response trials seem to be correlated. A clear pattern towards the end in some sessions is observable (see figure 4.1 and figure 4.2). The no-response time, inherently are the longest response-times, since the system will wait the maximum amount of time for a potential response, but all the other response times are shorter than the no-response time.

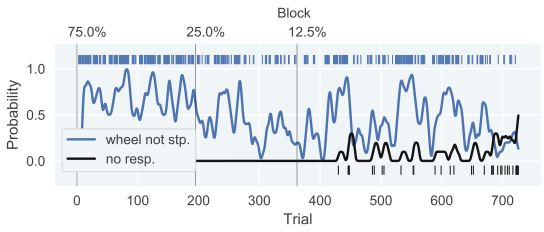


(a) *No-response distribution*

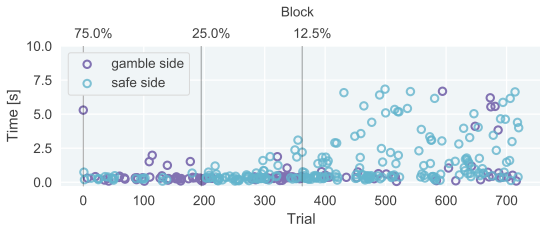


(b) *Response time distribution*

Figure 4.1: Session JG14_190621



(a) *No-response distribution*

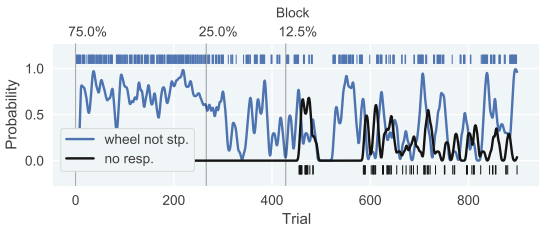


(b) *Response time distribution*

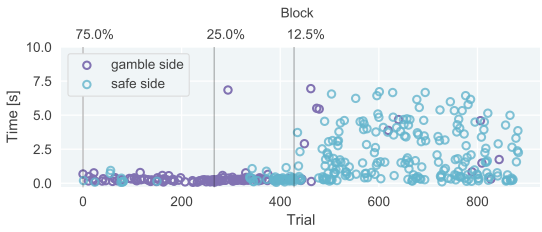
Figure 4.2: Session JG18b_190828

Although a majority of sessions showed the clear separation of no-response and response trials with no-response occurring mostly at the end, the lack of engagement is not limited to the end of the session for all recorded sessions. During some sessions, there is a high amount of no-response trials during the whole session which is accompanied by a significant increase in response time (see **??** and **??**).

As can be observed in figure 4.4, the count for trials, where the animal fails to stop the wheel makes up a large proportion of the total trials. This is mostly caused by the fact that animals more or less ended up stopping the wheel after around 300 trials for all trials. Trials where the animal failed to give a clear indication of the side chosen, are much less compared to total trials. The range of the count of trials selected for further analysis is small.



(a) *No-response distribution*



(b) *Response time distribution*

Figure 4.3: Session JG15_190725

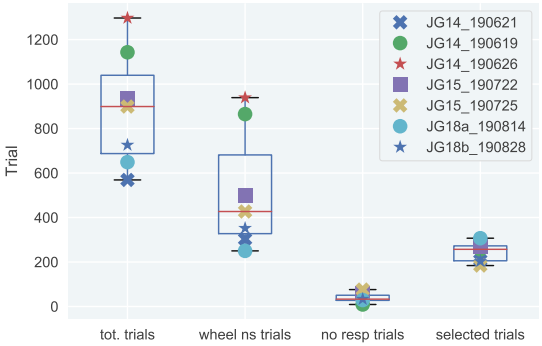


Figure 4.4: Distribution of trials for all recorded sessions, focusing on wheel-not-stopping trials (wheel ns) and other trials

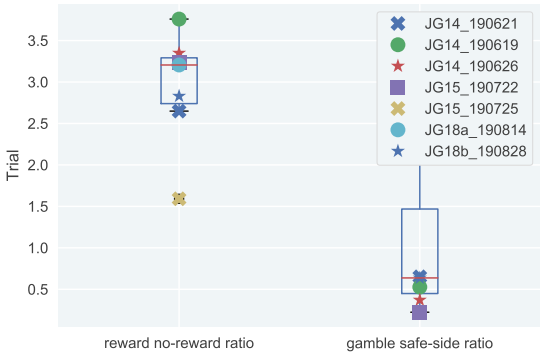


Figure 4.5: Distribution of reward and no-reward trials for selected trials

After filtering out the trials, where the mouse failed to stop the wheel, and selecting the response is distributed between three blocks: gamble-side, safe-side, and no-response (see figure 4.6). No-response results from the animal failing to turn the wheel a certain degree and move the stimulus over a certain position on the screen to indicate a clear choice.

To analyze the behavior in regard to reward vs no-reward as well as gamble-side vs safe-side the absolute ratio of both cases are of interest. The ratio of rewarded trials to not-rewarded is heavily in favor of the rewarded for all recorded sessions (see figure 4.5). The ratio of rewarded trials to not-rewarded is heavily in favor of the rewarded for all recorded sessions (see figure 4.5).

Both for the final training milestone and for later correlating neuronal activity with behavior data, an adaption of the choices based on a probability change of the large reward administered at the gamble-side is important. A rational agent will choose the safe-side for a reward probability at the gable-side of 12.5% and 25% but nearly exclusively stick to the gamble-side for the 75% reward probability section. Due to the behavior design, the animal additionally has to regularly check the not so desirable side to pick up cues for a change in reward probability of the gamble-side. Such behavior is, for example, clearly observable in figure 4.7 , where the

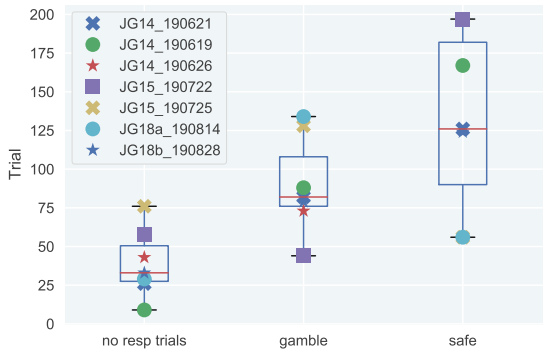
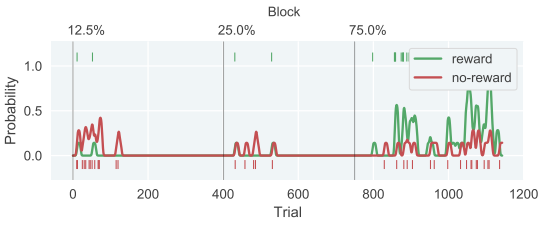
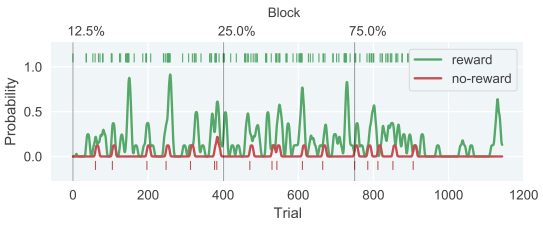


Figure 4.6: Distribution of trials with wheel-not-stopping trials filtered for all recorded sessions In regard to the distribution between no-response and gamble- vs safe-side chosen (successful)

animal chooses the safe-side for the 12.5 and 25.% block and the gamble-side for the 75% block. From the no-rewarded trials it is observable, that the animal keeps probing regularly for different sides, and as soon as it starts to receive frequent rewards at the gamble-side, changes its behavior.

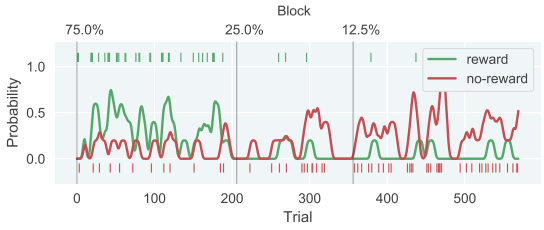


(a) *Gamble side*

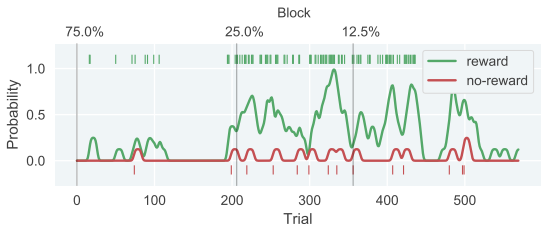


(b) *Gamble side*

Figure 4.7: Choice behavior for recording session JG14_190619



(a) *Gamble side*



(b) *Gamble side*

Figure 4.8: Choice behavior for recording session JG14_190621

(a) *Gamble side* **(b)** *Gamble side*

Figure 4.9: Choice behavior for recording session JG15_190722

A clear separation between the high probability reward block and the low reward blocks and the preferred choice for the safe-side during the low reward blocks is observable. This expected behavior is not limited to the increasing reward probability but also observable for a decreasing reward probability for the high reward gamble-side, compare figure 4.7 and figure 4.8.

Both figure 4.7 and figure 4.8 are examples of very rapid adaptation of the behavior to the change in reward probability at the gamble-side, but not all animals and sessions display such behavior. The shift can also occur much slower and the perceived reward much lower than the overall maximum possible reward like for example in session JG15_190722 (see figure 4.8).

From the total of 7 analyzed sessions, two are selected for further analysis. This decision is mostly based on the adaption of the animal (or lack thereof) to changing blocks.

A chi square test comparing the chosen side, depending on the probability block, shows that for all analyzed sessions, the chosen side is depending on the probability block, (see table 4.2).

Table 4.2: Results of Chi square test for independency

block	dep	stat	dof
all	dependent	353.802895	4
75	independent	2.788187	1
25	dependent	280.640091	1
12.5	dependent	296.427729	1

All individual sessions also were the chosen side depending on the trial, except for Block 75% and session JG14_190619 and JG14_190619, for which the side choice was independent. The gamble-side and safe-side choices for both sessions are displayed in table 4.3.

Table 4.3: Not dependent sessions for 75% probability lock

session	block	dep	stat	dof	critical
JG14_190619	75	independent	2.333333	1	3.841459
JG18b_190828	75	independent	2.000000	1	3.841459

4.2 Neural Data Analysis

The sessions previously selected are spike sorted, based on the in methods described procedure. As stated, the aim of the experiment is to find potential correlations between individual neuronal activity and behavior paradigms. Since the spike is the basic level of information transmission of the neuron such an approach will likely first focus on spikes and spike times of individual neurons. Event synchronization and alignment for analyzed sessions between electrophysiology data recorded with the Intan and behavior data recorded with Phenosys is checked following the procedure outlined in methods.

4.2.1 Exploratory Data Analysis

The number of good, MUA, and noise clusters depending on the recording session are listed in table 2. From the data, it is obvious that even after manual curation there are a significant number of MUA clusters, yet still a high number of good clusters. Since there are still a high number of good clusters, relatively strict requirements for good clusters are chosen.

Hier and thereafter the clusters are seen as putative neurons are referred to simply as neurons. Usually, a good rule of thumb is to expect a similar number of individual neurons as the number of individual electrode recording sites, since the probes used to record the raw LFP data have 128 channels the number of neurons extracted from the raw data is satisfactory (see table 4.4).

Table 4.4: Overview of sorted clusters for recording sessions

tot. clusters	nr. good	nr. mua	nr. noise
167	92	61	14
385	187	174	24

Moreover, the distribution of spikes for each neuron and thus the spike rate per neuron, is of interest. Since a majority of the information and computation in the neural networks is related to spike rate a wide range of firing rate is expected from real neurons, thus such a wide range in the data further speaks for the correlation of processed clusters and the putative neurons they represent.

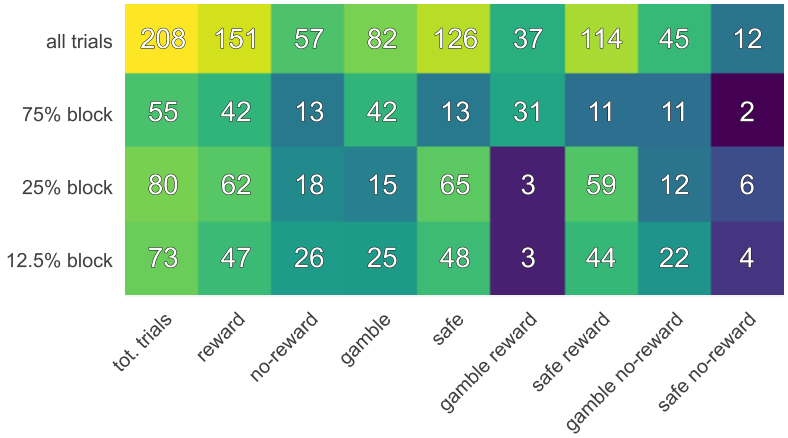


Figure 4.10: Colormap of side and reward distribution , depending on chosen side and reward, for all trials and respective reward blocks for analyzed session JG14_190621

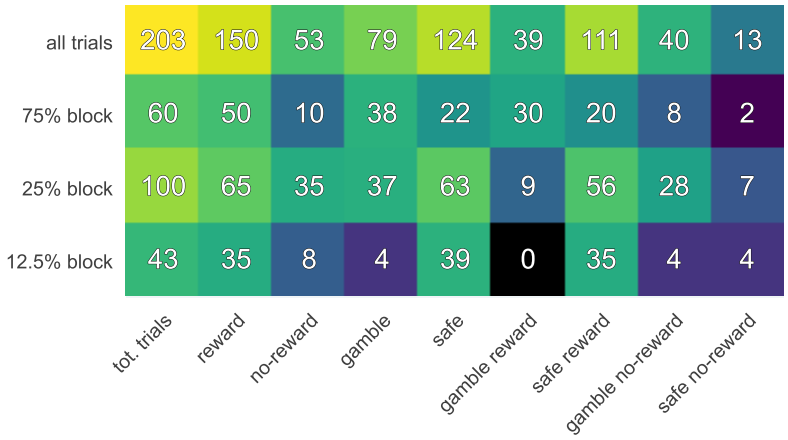
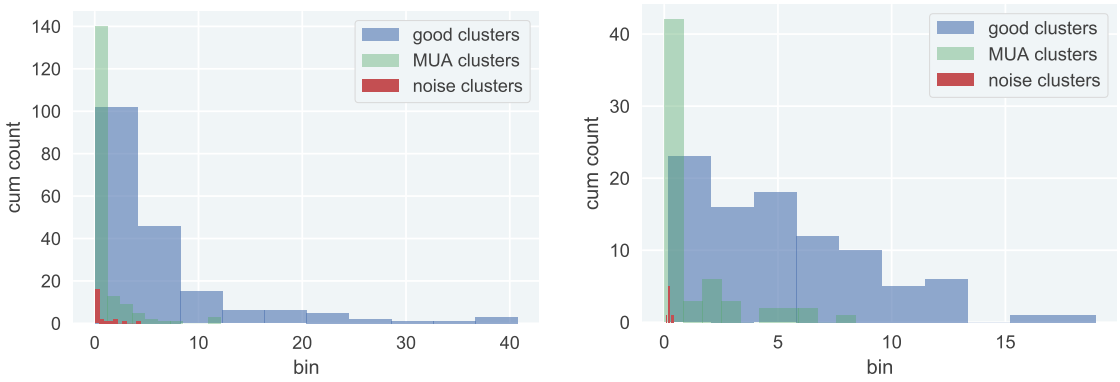


Figure 4.11: Colormap of side and reward distribution , depending on chosen side and reward, for all trials and respective reward blocks for analyzed session JG18b_190828

A heavy-tail distribution of activity across neurons is likely to be expected with many neurons contributing on average and few neurons contributing a significant amount of spikes, thus having a very high activity. Due to the high diversity of neuronal types in the PFC, a recorded spectrum will additionally be broadened.

As we can see from figure 4.12 these expected heavy-tail distribution is met and the automatic and manually clustered data likely represents actual physical neurons. Interestingly is the large difference in maximum firing rate between both of the sessions, although both are similar in length (JG18b_190828: 26 minutes and JG14_190621: 18 minutes)

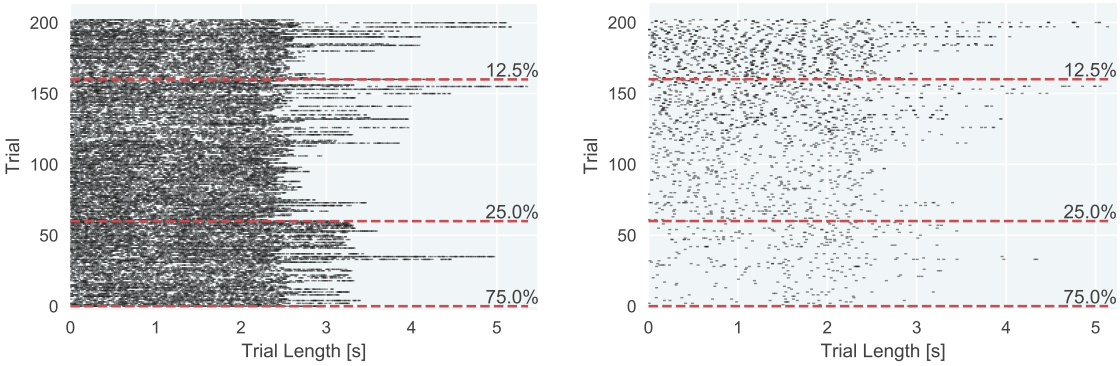


(a) *Session JG18b_190828* **(b)** *Session JG14_190621*

Figure 4.12: Distribution of total spike numbers across good clusters representing putative neurons

4.2.1.1 Clusters per Trial Analysis

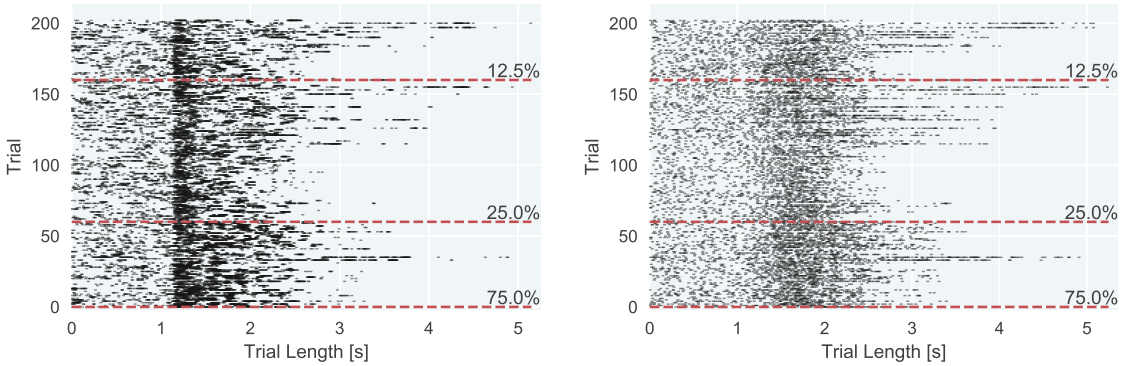
Although heavily relying on manual pattern recognition, plots of the spike train for all trials, provide a sound starting point. This step yields some first interesting insights into how individual neurons differ and how the firing of some neurons is related to stages in each trial.



(a) *Neuron 63* **(b)** *Neuron 31*

Figure 4.13: PSTH plots for selected trials stacked, with gamble-side reward probability blocks marked for recording session JG14_190621 and neuron 63 and 31 (changes in probability block are marked by the red dotted line, with the following probability indicated)

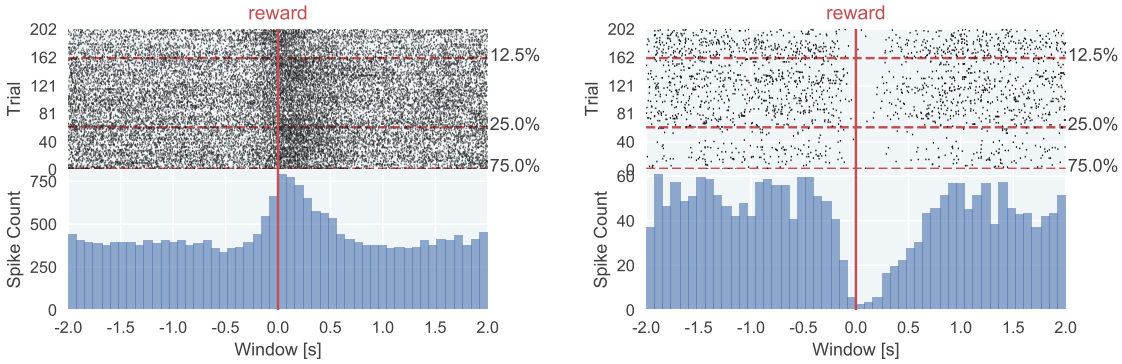
As expected from the total spike count per cluster, the event plots of individual neurons differ (some of the more contrasting different neurons can be observed in figure 4.13). Several distinct patterns that are similar for multiple neurons, can be observed. The most interesting pattern, that can be observed in the PSTH plots, is that multiple neurons display a change in firing rate that seems to be correlated to some specific event in each trial (see figure 4.14).



(a) *Neuron 106* **(b)** *Neuron 117*

Figure 4.14: Fringe rate and trial event correlation in PSTH plots for selected trials stacked, with gamble-side reward probability blocks marked for recording session JG18b_190828 and neuron 106 and 117

The observed increase in firing rate of some neurons clearly is timed with the reward onset, an example of such a neuron is shown in figure 4.15a



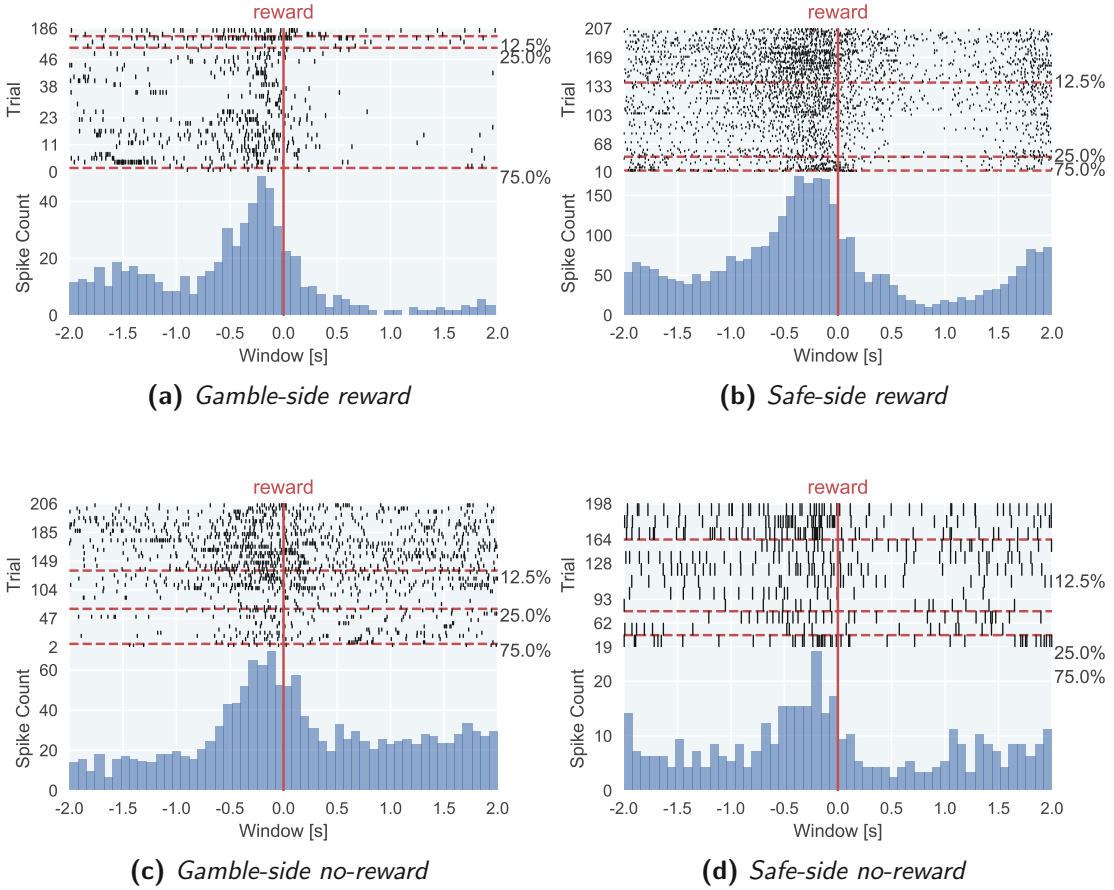
(a) *Positive correlation for euron 104* **(b)** *Negativ correlation for neuron 2*

Figure 4.15: Positive and negative firing rate and trial event correlation in PSTH plots and histogram for selected trials stacked for gamble-side and reward combination, vertical dotted red lines marking the transitions for reward probability blocks, with the following probability marked for the recording session JG18b_190828

But there is also a group of neurons displaying the opposite behavior and decreasing the firing rate around the reward (figure 4.15b). From both examples, which can be found in all recorded and analyzed behavior sessions with a multiple neurons, it is clear that these neurons somehow correlate to the reward. Nonetheless, most of the trials do not fall in one of the two above categories, but display no significant change in firing rate across the time window.

As described earlier, Passecker *et al.*, 2019 found a correlation between the firing rate of specific neurons and the strategy chosen by the animal. Although the change in these

neurons is developing sequentially over multiple trials, neurons might also be exclusively linked to different instances of reward, independent of the trial history in the new behavior task. Furthermore, focusing on single trials for correlation patterns can nonetheless lead to insights into neural encoding. To investigate if such neurons can also be found in mice with the translated task into the virtual reality system, plots of the above-identified groups of neurons, that have a high correlation of change in firing rate to the onset of the reward event are further analyzed. Each trial is split, regarding the reward event, into several subcategories. Reward and no-reward trials, as well as gamble-side and safe-side trials are separated (see figure 4.16). Also, the respective combination of both as of reward+ gamble-side vs reward+ safe-side and no-reward+ gamble-side and no-reward+safe side is inspected.



(a) *Gamble-side reward*

(b) *Safe-side reward*

(c) *Gamble-side no-reward*

(d) *Safe-side no-reward*

Figure 4.16: Reward and no-reward as well as safe- and gamble-side firing rate and reward-event correlation in PSTH plots and histogram for selected trials stacked, with gamble-side reward probability blocks marked for the recording session JG14_190621 neuron 76

From manual visual inspection of the spike trains a much stronger pattern of correlation to the reward event, compared to the other events of each trial, is observable. That is inline,

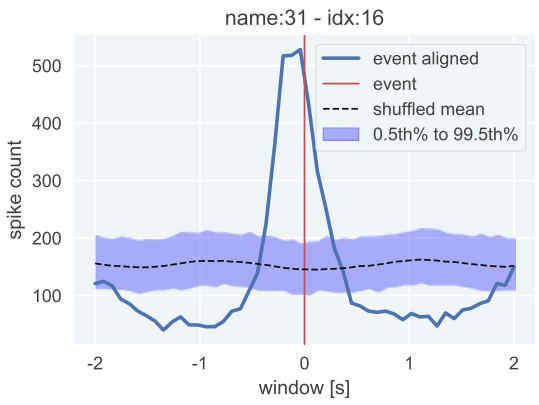
with what was the aim of the design of the behavior and is favorable in answering the initial question. Although particular attention has to be placed in avoiding confirmation biases introduced by the scientific question.

4.2.2 Statistical Data Analysis

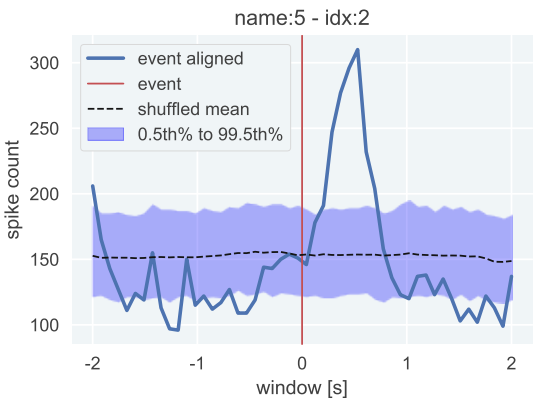
The significance of the visually identified correlation between the reward event and how this correlation depends on specific subsections of trials is further analyzed, using the bootstrapping approach described in the methods sections. A window size of $+-2$seconds is chosen. For both sessions, the bootstrapping approach described in section 3.3.3.1 is used to determine the distribution of the spike rate in the chosen window. The resulting distribution is compared with the firing rate for trials aligned to the reward / no-reward event, for all trials and respective subselections based on the chosen side and reward or no-reward and in addition the probability blocks.

This comparison yielded a number of neurons that are significantly above the confidence interval (see figure 4.17) As expected, these neurons are the same as identified in the previous exploratory data analysis step. Similarly, in both sessions, neurons that increased the firing rate, before the event, and after the event were found.

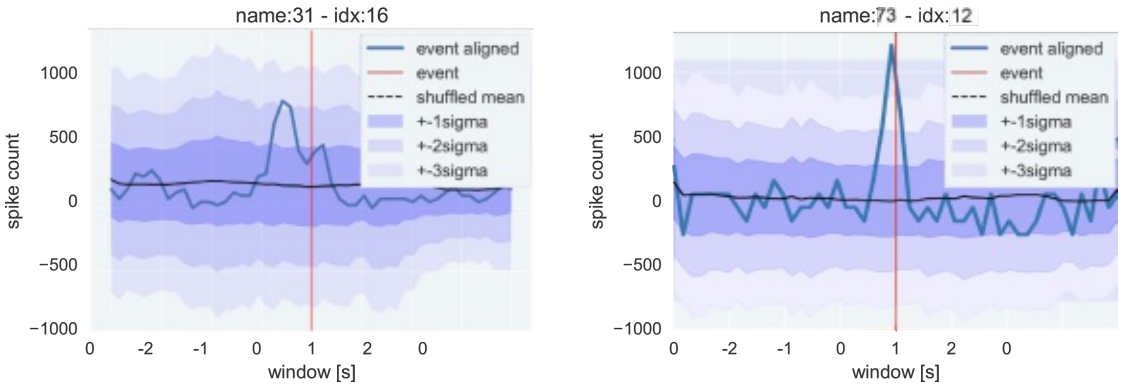
Determining the significance of neurons via a comparison between the distribution of randomly sampled windows via the bootstrapping approach and firing rate from reward-aligned trials is defined by the two different ways described in the methods chapter. With the 2 sigma significance definition, fewer neurons are determined to be significant, but this metric is less stable in term of small sample size for specific subselections such as gamble and reward for the 75% probability block, therefore the 5th to 95th percentile definition is used to determine the significance of a neuron. The results for both sessions are plotted in figure 4.18 and figure 4.19 respectively.



(a) *JG14_190621 90% convidence*

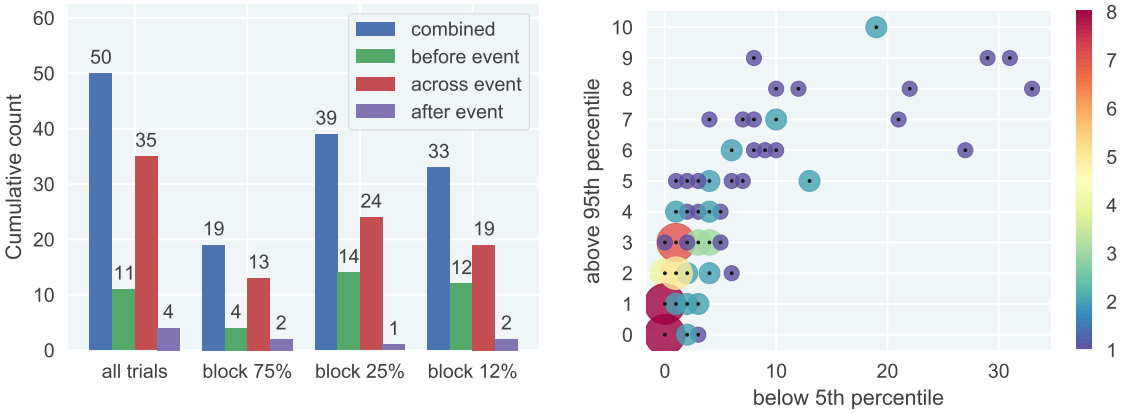


(b) *JG18b_190828 90% convidence*

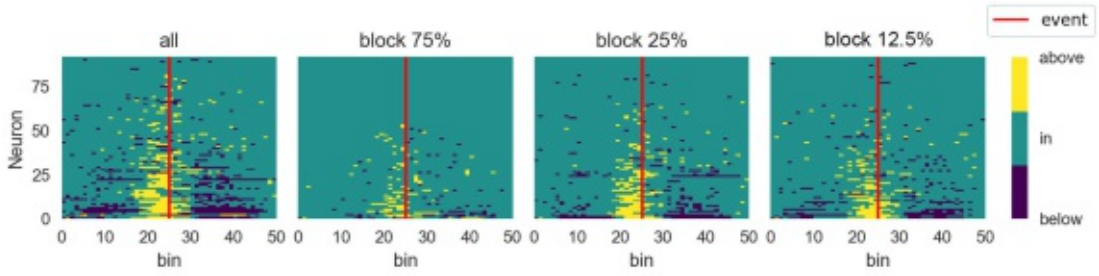


(c) *JG14_190621 2 sigma* **(d)** *JG18b_190828 2 sigma*

Figure 4.17: Comparison of mean firing rate from bootstrapping approach and reward aligned spike times for neurons responding before the reward event and after, both compared to a confidence interval between the 5th and 95th percentile as well as 2sigma.

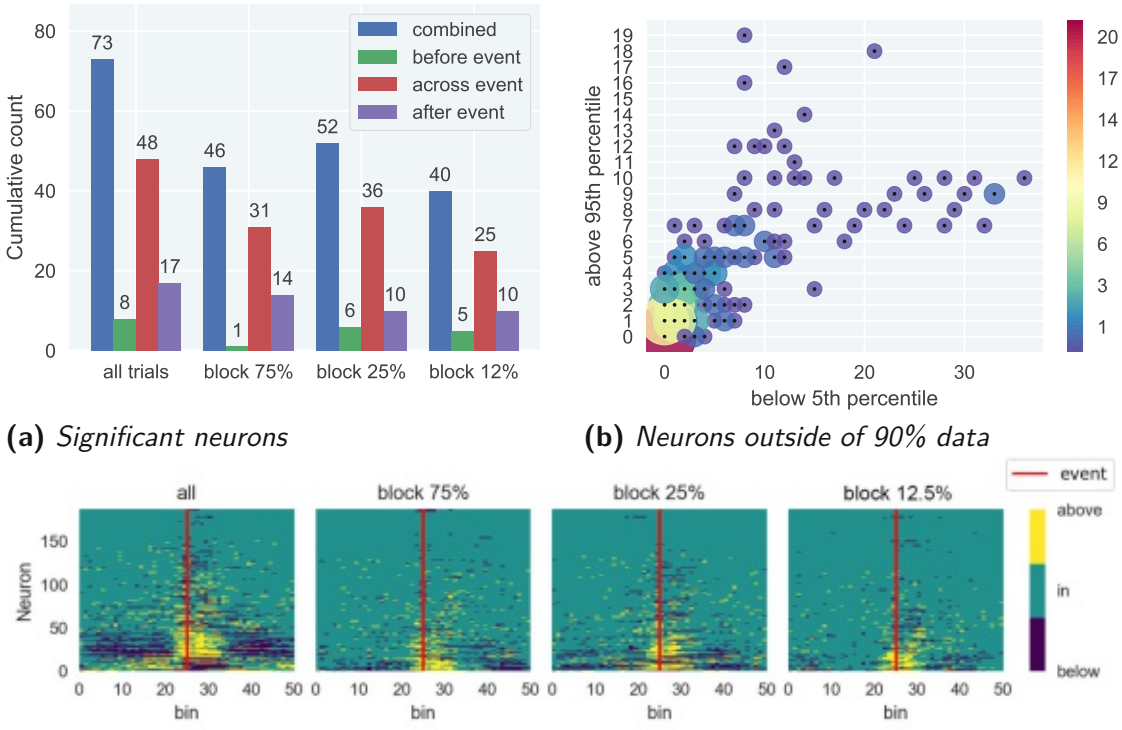


(a) *Significant neurons* **(b)** *Neurons outside of 90% data*



(c) *Neuron-fingerprint for respective blocks*

Figure 4.18: Summary of the neuron fingerprint for session JG14_190621 all, bar chart displaying the distribution of neurons that have at least two bins outside of the 90% confidence interval, color map displaying the unique fingerprint of each neuron for the session



(a) *Significant neurons*

(b) *Neurons outside of 90% data*

(c) *Neuron-fingerprint for respective blocks*

Figure 4.19: Summary of the neuron fingerprint for session JG18b_190828 all, bar chart displaying the distribution of neurons that have at least two bins outside of the 90% confidence interval, color map displaying the unique fingerprint of each neuron for the session

For both analyzed sessions, quite a large number of neurons display a change in firing rate, that falls above 90% of the data obtained via the bootstrapping method from random samples.

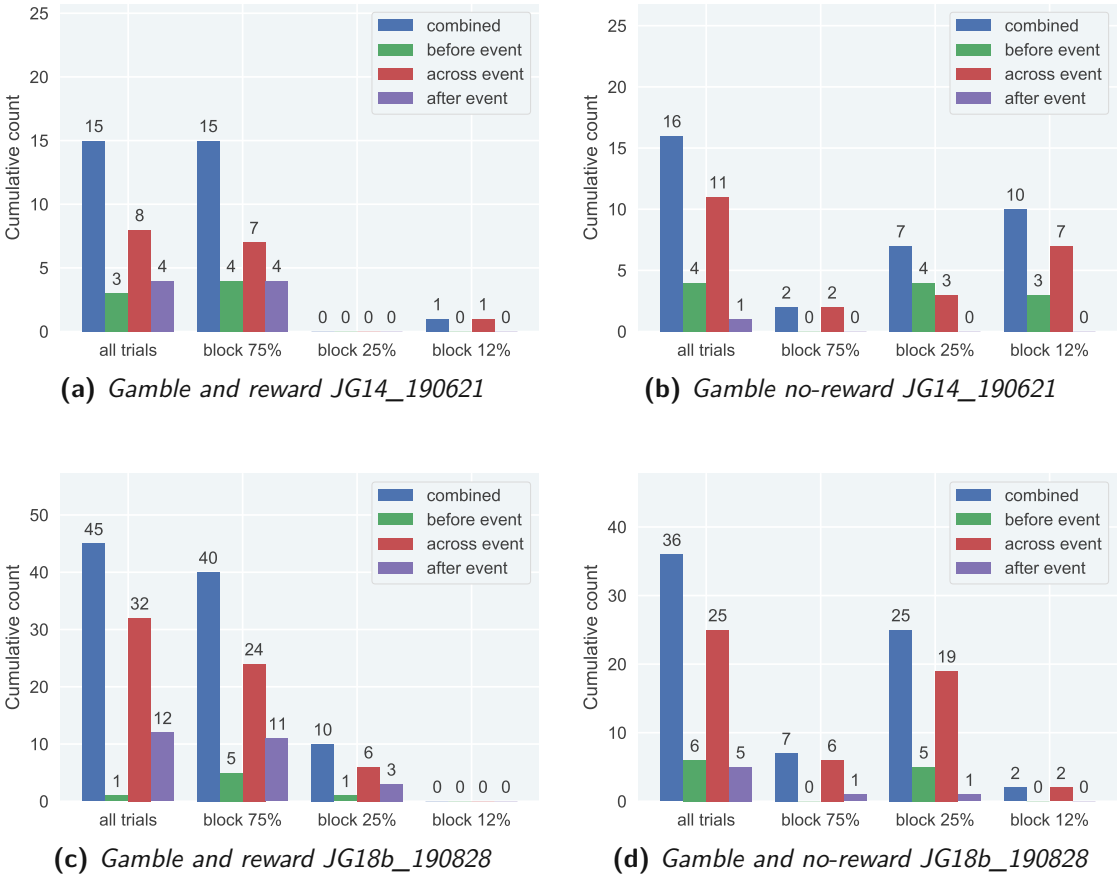
Interestingly, the two recording sessions display a different property of the significant neurons, in session JG14_190621 neurons mostly have a significant increase before the reward event, while in session JG18b_190828 this increase mostly occurs after the reward event. This difference holds true for all blocks. To determine the significance of this behavior, a chi-square independence test of both sessions for the number of neurons occurring before and after for all trials, as well as the 75% probability block and the 12.5% probability block is conducted. It reveals, that the H0 of independence of the number of significant neurons in terms of before or after has to be rejected, thus they are not independent. This suggests that the two sessions are significantly different in terms of how neurons respond to the reward event.

The second question, which is based on the main question for data analysis of this thesis, is whether the number of neurons responding to gamble-side and reward or no-reward are dependent on the probability block for the gamble-side. The basic hypothesis underlying this question is the findings from Passecker *et al.*, 2019, that the firing rate of specific neurons,

encodes the choice of the animal and responds differently to reward or no-reward at the gamble-side.

The absolute count of significant neurons, for the gamble-side and rewarded choices compared to not regarded choices for both analyzed sessions, is shown in figure 4.20.

figure 4.20.



(a) *Gamble and reward JG14_190621*

(b) *Gamble no-reward JG14_190621*

(c) *Gamble and reward JG18b_190828*

(d) *Gamble and no-reward JG18b_190828*

Figure 4.20: Number of neurons responding to the respective reward or no-reward event at trials where the animal choose the gamble-side for all trials, and trials falling in the three probability blocks, for session JG14_190621 and JG18b_190828

A difference between the number of significant neurons that respond in the 75% probability block and the two lower probability blocks is observable for both sessions. The difference is, that for the 75% probability block, more neurons respond for gamble-side rewarded trials, as for the 25% and 12.5% probability blocks, more neurons respond for gamble-side not-rewarded trials. To put his into perspective whether it is also a similar difference for other combinations of specific outcomes such as safe-side reward/no-reward or both sides, colormaps of both sessions are plotted (see figure 4.21 and figure 4.22 respectively).

Although the number of neurons seems to be strongly influenced by the number of trials in the sessions, falling in the respective combination, not all combinations follow this paradigm. Also here the strong difference between gamble-side reward and not-rewarded trials for the different probability blocks is observable. For example, although 11 trials fall into the combination of block 75%, gamble-side, and no-reward only two neurons seem to be significantly responding for these trials.

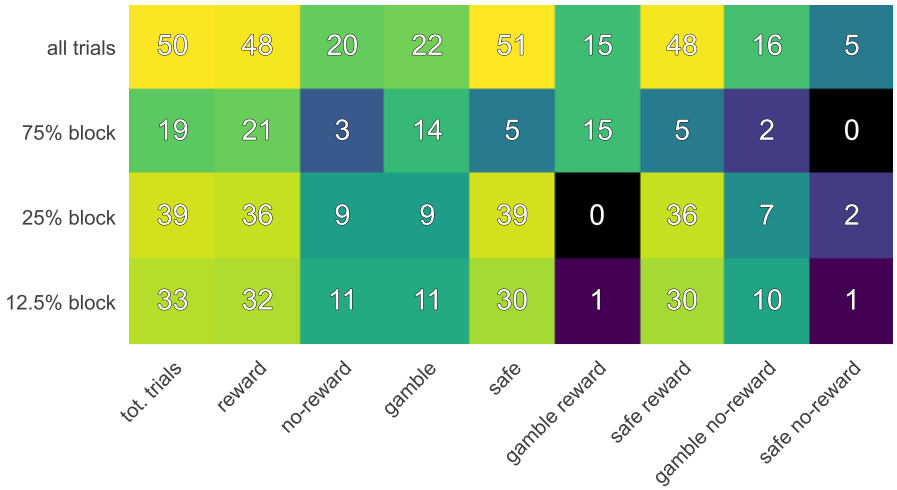


Figure 4.21: Distribution of significant neurons for session JG14_190621, gamble-side reward and gamble-side no-reward contingencies for respective combinations, independency test combination marked by the x

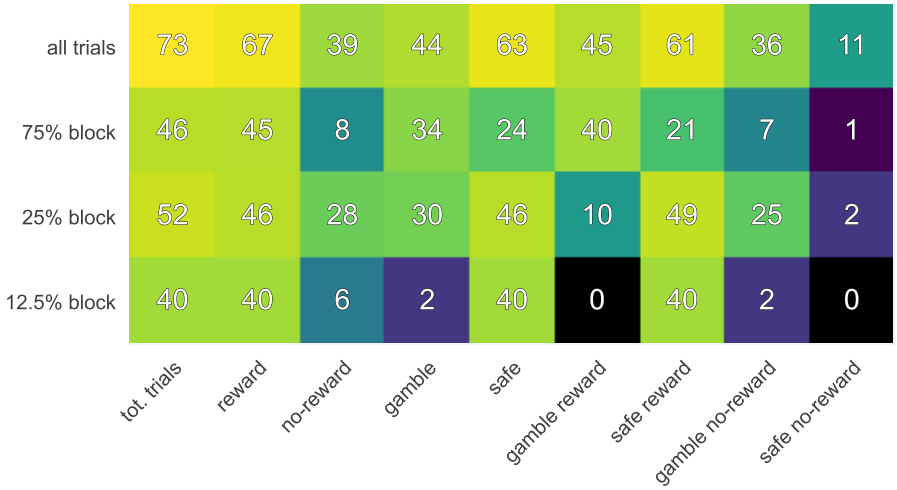


Figure 4.22: Distribution of significant neurons for session JG18b_190828, gamble-side reward and gamble-side no-reward contingencies for respective combinations, independency test combination marked by the x

To investigate, whether these differences are significant, again a chi square independence test is performed. The results of the chi-square independence test for gamble-side choices in terms of reward vs no-reward for the respective session are displayed in table 4.5 and table 4.6 for both sessions.

Table 4.5: Results chi-square independency test session JG14_190621 of gamble-side reward and gamble-side no-reward contingencies for respective combinations, independency test combination marked by the x

All trials	75% Block	25% Block	12.5% Block	H0 independent	Teststatistic	Dof
x				indep	0.758	1
	x			not-indep	9.941	1
		x		not-dep	8	1
			x	not-indep	6.4	1
	x	x		not-indep	14.962	1
	x		x	not-indep	12.887	1
	x	x	x	not-indep	24.262	2

Table 4.6: Results Chi-suare independency test session JG18b_190828 of gamble-side reward and gamble-side no-reward contingencies for respective combinations, independency test combination marked by the x

All trials	75% Block	25% Block	12.5% Block	H0 independent	Teststatistic	Dof
x				indep	0.842	1
	x			not-indep	16.2	1
		x		not-dep	4	1
			x	indep	2	1
	x	x		not-indep	16.16	1
	x		x	indep	3.102	1
	x	x	x	not-indep	20.67	2

Interesting is, that although for both sessions, comparing the high 75% probability block and the low 12.5% probability block the resulting reward and no-reward outcome are not independent of the probability for session JG14_190621, this is not the case for session JG18b_190828.

An interpretation of the results by the author is presented in section 6.2.

Results: New System

The following section will describe the design and implementation of the new standardized behavior setup. This comprises software and hardware to train animals, record single-unit activity using in-vivo electrophysiology, and process the generated data. As described in the preface, due to readability it will compromise on separation in introduction, methods, and results, but combine the parts for each subchapter. For an outline of what is prior art and what was newly developed, the section 1.4 can be consulted.

Based on the theoretical advantages of a virtual reality system over classical physical behavior tasks outlined in Chapter section 2.6.1.1 and the experience from the Phenosys system outlined in Chapter section 2.8, the basic concept of the new system will be based on the same concept.

5.1 Main Problem and considerations

The main problem to address in regard to the existing setup is twofold. On the one hand, it concerns the training and recording setup, and on the other hand the data management.

The issue with the setup and thus the Phenosys system (hardware and software) is the customizability of the behavior tasks, the adaptability of the system, and the extensibility with different hardware. The Phenosys system is closed source and only a proprietary visual scripting language can be used to design behavior paradigms. This software is severely limited. For example, it is not possible to use different distributions for the gamble-side safe-side ratio which dynamically adapt depending on the choices of the animal. Only a fixed ratio can be set. In addition, the modification of the task and implementation of new or slightly adapted tasks, is very time-consuming and tedious.

The hardware components compatible with Phenosys are very few and the options are limited. This limits the freedom in designing behavior tasks. Adaptability and expandability of the hardware are important, since experiments in neuroscience, especially in the domain of electrophysiology, require significant upfront investments in equipment and expertise both in terms of time and money. A system that can be easily adapted and integrated, with the

existing ecosystem, will on the one hand bring down costs, and therefore lower the entry barrier, and on the other hand, save precious time until animals can be trained and data can be gathered. Both are desirable since they lead to a higher scientific output, which is favorable if the global community wants to advance their understanding of the brain.

Data generated with and by the Phenosys and Intan recording setup used in the previous setup, rely on proprietary data standards. These standards are not directly compatible with data processing pipelines and analysis packages used by the wider neuroscience and computational neuroscience community. Especially the deployed data standards are of growing interest since the advances in electrode design are leading to more and more data being gathered per experiment. This data needs to be stored and analyzed, consequently requiring more and more standardized frameworks and pipelines to cope with the fast amount of data. To put this into perspective for the Phenosys setup and the gamble-task paradigm, described earlier, a single session (roughly 30 minutes of recording time) generates about 10 GB of raw data (for 128 channels sampled at 20KHz).. For the whole experiment cycle, this will easily amount to around 1 TB of data. This data not only has to be preprocessed but stored in such a way that it is later easily retrievable to perform analysis on. The specific shortcoming of the currently deployed system with data is that behavior data saved using the CSV file format with an encoding from Microsoft. The encoding used by Phenosys to store events and event-times of a behavior session is not easily adaptable which again severely limits the degree of possible automatization down the line.

To overcome these problems, a switch to a common, open-source, and unified data standard, with a high adoption throughout the neuroscience community is desirable. Several initiatives are tackling the problem of defining data standards for neuroscientific research. These are oriented on and inspired by other large-scale scientific data-gathering endeavors, such as in nuclear physics or in molecular-biology, to foster collaboration and sharing of datasets, methods, tools and to enable comparison and speed up the overall progress of neuroscience research. Neurodata without borders, International Neuroscience Laboratory, and OpenNeuro are among the leading ones. Using such a standard, will allow for easier collaboration with other labs.

Potential Solutions

Current most labs build their custom solutions based on microcontrollers or buy commercially available setups. Both of these approaches have their own advantages and disadvantages, which are summarized in table 5.1

	What is available		What is needed
	Commercial solutions	bespoke solutions	
Implementation	- plug and play for basic functions - forced to use company hardware - customisation as a service very limited	- complex & long implementation - no standardisation	- easy to implement - works with existing hardware and measurement devices - quick extension with new measurement methods - integrated with standardised data storage format - integrate into existing data analysis pipelines & software - integrate with existing hardware ecosystem
Flexibility	- closed source & proprietary algorithms - minimal adaptability - only work with company accessories - customisation expensive and very time consuming	- maximum flexibility but very time consuming	- standardised and open source system - easy modifiable and expandable (software and hardware) - easy adaptable to new behaviors - sharing and adaptation from collaborators platforms and new methods
Cost	- very expansive (300k€ and up for single setup)		- affordable also for smaller labs in 2nd world countries

Table 5.1: Summary of available solutions and desired solutions for rodent behavior experiment setups

Besides the above-mentioned widely used bespoke solutions based on microcontrollers and commercial limited solutions, there are several more standardized open-source frameworks under development and used by the wider neuroscience community.

- **Rigbox (Bhagat *et al.*, 2019)**

  Rigbox is a "Matlab toolbox for managing behavioral neuroscience experiments" (*Cortex-Lab/Rigbox* 2021). The Matlab toolbox was developed by Christopher Burgess at the CortexLab at UCL. It uses a modular approach with one central controller that can manage multiple slaves. Each slave drives a behavior setup. For interaction with sensor hardware, a National Instruments data acquisition card (NI DAQ USB 6211) is used. Visual stimuli are controlled by the Matlab toolbox Psychtoolbox (*Psychtoolbox-3 - Overview* 2021), a widely used package for behavior experiments both in social science and behavioral neuroscience. Results and measurements of each slave are synchronized with the master and stored in a central database.

- **Bpod (*Sanworks* 2021)**

  Bpod is a state machine based on an Arduino compatible microcontroller (Teensy 3.4 and 3.6). It was developed by Josh Sanders at the Kepecs Lab at Cold Spring Harbor Laboratories. It is based on an earlier concept of B-Controle (*Bcontrol* 2021) developed at Brody Lab at Princeton University. The Bpod has a high adaptation

rate for real-time neuroscience experiments. The microcontroller acts as state machine and executes behavioral tasks in real-time. This allows for high accuracy of state time synchronization since the latency is very low. Both the hardware and software are open source. The Bpod state machine can be expanded with several different peripheral devices but it is not designed to handle visual stimuli.

The state machine can be controlled via an API, which speeds up behavior design and adaptation since a high-level language instead of the Arduino (C/C++) based language can be used. Their Bpod API library is currently available for Matlab and Python (PyBpod). While the original Matlab library was developed by Sanders and still is maintained by him, the python implementation was developed by The Scientific Software Platform (SWP) from the Champalimaud Foundation in Lisbon (*Scientific Software | Champalimaud Foundation* 2021) during their ongoing cooperation with major research labs under name of International Brain Laboratory (*International Brain Laboratory* 2021), also including the Cortexlab (Rigbox) and Kepecslab (Bpod). The Matlab based solution has been around for longer than the python on and has a much larger user base.

Due to its close resemblance with the behavior task, the steering wheel virtual reality task (The International Brain Laboratory, Bonacchi, *et al.*, 2019), developed during the International Brain Lab initiative, is especially interesting.

- **Autopilot (Saunders and Wehr, 2019)**

  Autopilot, developed by Jonny Saunders and Michael Wehr at the University of Oregon, is a complete system based on a Python framework that uses distributed Raspberry Pi 4 nodes as a controller for a multitude of different behavior tasks. Tasks can be developed via a non-code User interface (UI). Autopilot is one of the latest and, when it will be fully developed and matured, most complete frameworks. It not only integrates with other tools like OpenEphys or Deeplabcut and will provide data already in the Neurodata Without Borders file format but it also is built as a truly modulatory remote manageable system. This approach allows for easy deployment and management of a large cluster of autopilot controlled tasks. The use of a Raspberry Pi as the core computing unit presents a powerful platform for more complex applications. This is especially important since visual computing will probably become of major significance in the future for behavior tracking.

- **PyControle (*pyControl* 2021)**

  PyControle is a system of both hardware and software for controlling behavior experiments. It is based on Micropython and compatible microcontroller. It is being developed by the OpenEphys foundation. Compatible peripheral devices include nose-poke, audio boards, LEDs, rotary encoder, and stepper motors.

All of the above-mentioned solutions have their advantages and disadvantages. With the constraints of the gamble-task as a behavior paradigm and its necessity for a visual stimulus, the pyControle is no option. Since an emphasis is placed on open source and the preexisting

experience of the author with python, Matlab based solutions such as the Rigbox and Bpod with Matlab are also removed. Thus leaving Autopilot or PyBpod from the already existing solutions.

The main question that needed to be answered is whether a complete custom solution, not based on existing frameworks should be developed or one of the two frameworks shall be adapted to the gambling-task. Since the CortexLab and the International Brain Lab are still actively developing the PyBpod and the core concept of the gamble-task is very closely modeled to the task deployed by these labs, the PyBpod will already be well integrated out of the box. One significant downside is the lack of integration for visual stimuli and the used Bonsai solution is very complex to adapt and not well suited for that task. Also, the documentation is substandard for PyBpod. The lab is also using an earlier version of the Bpod with Matlab for another behavior task, which arguably will help in resolving inevitable issues while implementing the newer python controlled version.

Autopilot in contrast, has a more complete concept and has a better adaptability for the future. The only major downside of Autopilot is that the development of the system still is in a very early stage and many of the major milestones are still to be reached.

A complete custom solution might be the quickest option to implement the basic gamble-task, but it will require much more work to make it adaptable. Also, cross-compatibility will not be available, if it is not developed. Since the available time and manpower, such a custom approach will very likely not lead to the desired setup in time. For a comparison of all potential options see figure 5.1.

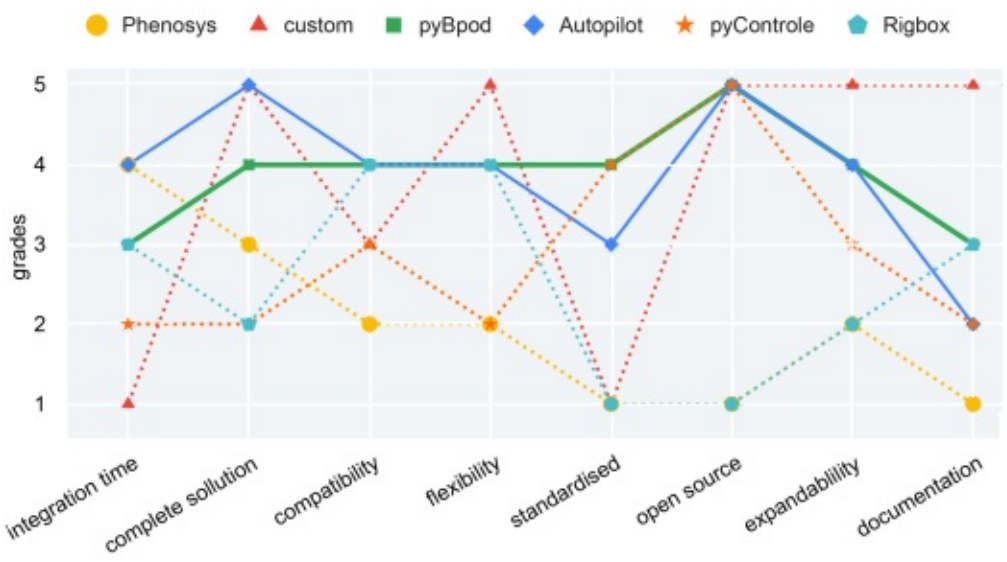


Figure 5.1: Comparison of different possible solutions fo the new VR-system, grading with 5: best and 1: worst, systems in closer considerations are highlighted with solid lines (PyBpod, and Autopilot are highlighted)

Based on the above-described pros and cons, the decision was made to implement the PyBpod solution, which was mostly based on close resemblance and in-house experience with an older setup.

5.2 System Overview

The new system is based on a three-tier concept (see figure 5.2 for high-level overview): the experimental platform, which comprises the hardware and software to train and record animals, the data format and storage framework, and the processing and analysis pipeline.

Open-source components, a standardized frameworks for integration, and high level of flexibility and expandability for future behavior tasks are at the core of the new system.

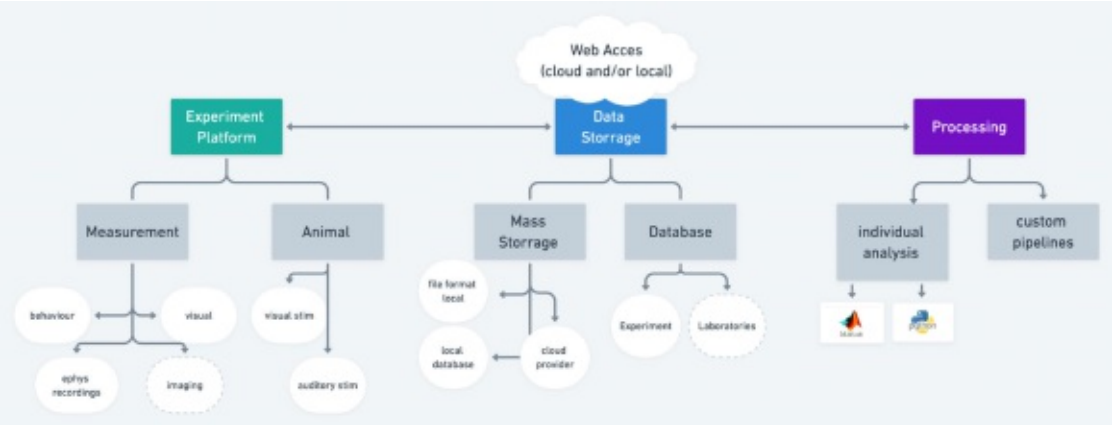


Figure 5.2: Overview of the new standardized open-source VR system, dotted line representing future implementations not yet fully ready

Although the core framework of the PyBpod and the task implemented by the International Brain Lab is already well suited for the gamble-task and forms the basis for the new concept, the overall implementation still has major differences.

There are two different versions of the hardware rig: One for training the animal and one for electrophysiology recordings. The rig comprises the screens for the stimulus, the rotary encoder and wheel, the mouse adapter, and the head-fixture. The difference is mostly due to the required high stability for the recording setup to minimize relative movement between the probe and the animal's brain. Both rigs are enclosed in a soundproof housing, to allow simultaneous training and recording from multiple animals in the same room without acoustical interference.

In addition to the new recording and training setups, the electrophysiology recording system is updated with a new acquisition board. The closed source Intan unit is replaced with the open-source OpenEphys board. This allows for much tighter integration and thus automatization with the new system. A new time and event synchronization system between the Bpod state

machine and the OpenEphys acquisition board, which is also backward compatible with the Intan is implemented. The whole system can be expanded in the future with additional hardware such as acoustic stimuli or can directly control lasers for optogenetic experiments.

5.3 Bpod Hardware

The central brain of each individual system is a Bpod state machine from Sanworks, the latest available hardware iteration 2.3 is used. Since the hardware and software are open-source under the GNU general public license (version 3), replication, modification, and distribution are freely allowed. Sanworks offers an assembly service to buy finished state machines. Nevertheless PPrinted circuit board (PCB) design files are available, and each Bpod board can be self-assembled, which is significantly cheaper compared to the ready-assembled boards. For this setup, PCBs were printed and Surface-mounted device (SMD) and non-SMD parts soldered manually.

All circuit diagrams, Arduino firmware, and libraries are made available by Sanworks:

`https://github.com/sanworks/Bpod-CAD`
`https://github.com/sanworks/Bpod_StateMachine_Firmware`

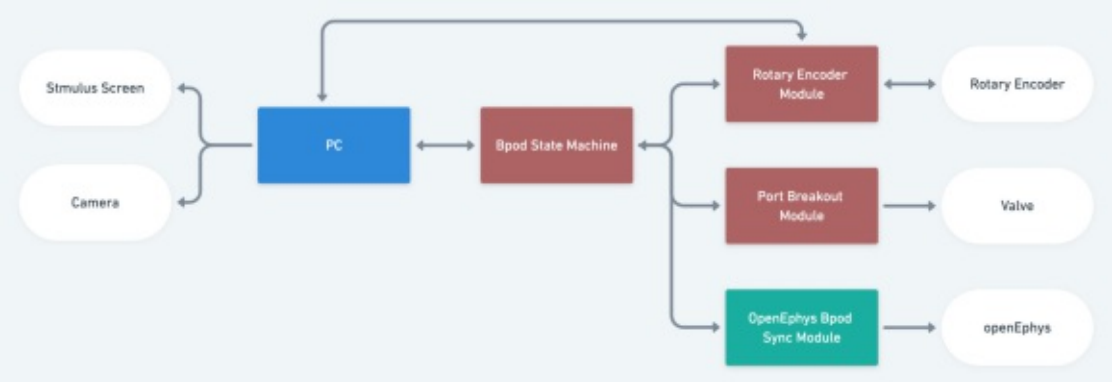


Figure 5.3: Schematic overview of the different communication channels between the Bpod state machine and connected hardware boards

5.3.1 Bpod State Machine

The Bpod state machine connects to sensors and modules via JR-45 plugs (see figure 5.4). There are two types of modules that can be connected to it, behavior modules and serial modules. Behavior modules are for nose poke detection and valve drivers. Serial module ports come in different shapes and comprise different sensors or actuators. Besides the two RJ-45 connector arrays (see figure 5.4) there are also 4 BNC connectors (two for output and two

for input) for digital communication via 5V TTL pulses. These can be used to communicate with components such as an electrophysiology acquisition board.

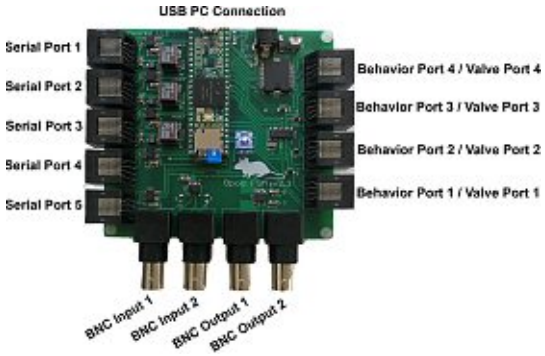


Figure 5.4: Overview of the ports of the Bpod state machine version 2.3

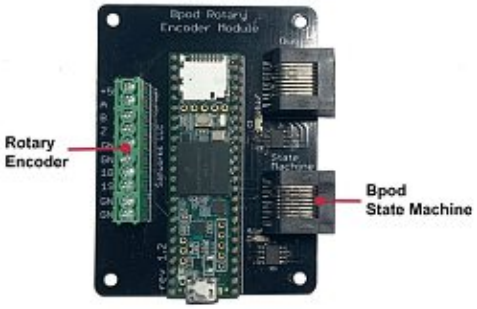


Figure 5.5: Bpod rotary encoder module

For the current gamble-task setup, the port-breakout-module (behavior module), the rotary-encoder-module, the analog-input- and analog-output-module (serial modules) are used . The Bpod state machine directly interacts with the rotary encoder via the rotary-encoder-module, controls reward valves via the port-breakout-module and sends and receives TTL signals via the input and output-modules. The stimulus and the camera are not directly controlled by the Bpod state machine, but indirectly controlled via the connected PC (schematic overview figure 5.3).

5.3.2 Rotary Encoder Module

The rotary encoder module (see figure 5.5) uses a Teensy 3.5 Arduino compatible microcontroller to interface with a Yumo E6B2-CWZ3E rotary encoder. The rotary encoder is connected to a LEGO wheel and is used by the mouse to move the stimulus on the screen. Furthermore, wheel positions can be saved on a sd card.

The Yumo E6B2-CWZ3E has a resolution of 0.35 degrees (1024 steps per 360°) and can be used for continuous rotation measurements.

5.3.3 Port Breakout Module

The reward, in form of water or other liquids with a high viscosity, is delivered via a gravity-based system and controlled with a solenoid valve connected to the port-breakout-module. The port-breakout-module is not connected to a serial port like all the other modules but has to be connected to one of the behavior ports.

The used valve is from the medical supplier company Lee-Company (*Ported* 2021, Valve Nr: LHDA1233115H). It has one input flow port and two output ports and can hold up to 30 psi

of differential pressure between these two ports. For the setup, only one output port is used, the other is plugged.

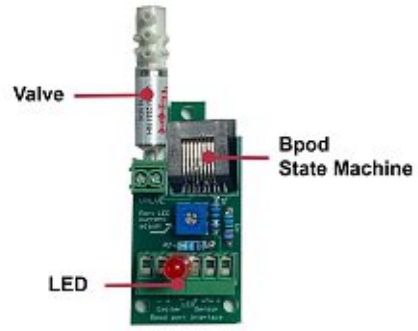


Figure 5.6: Port breakout board with valve

5.3.4 Analog Input and Output Module

Since the two input and two output BNC ports on the state machine are not enough for synchronizing with peripheral devices, such as lasers or TTL controlled cameras, additional TTL channels are provided by input and output modules (see figure 5.7 and figure 5.8). Multiple of these modules can be simultaneously connected to the Bpod state machine.

Each input module can record from 8 input channels with a range from -10V to +10V and a 12 bit per voltage precision. The maximum sampling rate is 20kHz, if all 8 channels are active, and 50kHz, if only two or fewer channels are active. Each output module has 8 output channels capable of a range of -12V to +12V, a 16bit voltage precision. The sampling rate is 100kHz, with two active channels, a reduced 50kHz, with four active channels, and 25kHz, with all 8 channels active.

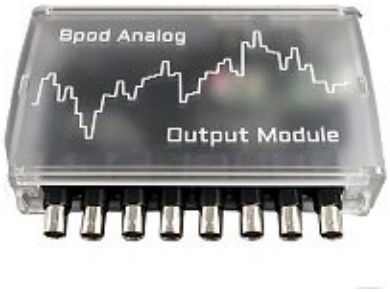


Figure 5.7: Bpod analog output module image from Sanworks (*8 Channel Output Module Bpod* 2021)

Figure 5.8: Bpod analog input module image from Sanworks (*Sanworks* 2021)

5.4 Graphical User Interface for gamble-task

The Bpod can be controlled via a QT5 based graphical user interface, with a PyQt and PyBpod as backend. This user interface (see figure 5.9) can be used to configure the state machine, load setups defining behavior protocols, load and save subject data and run experiments.

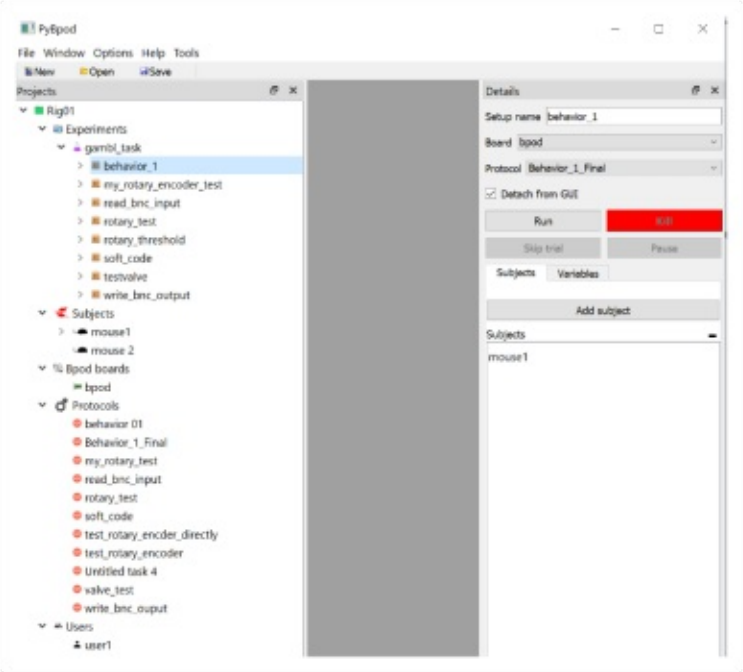


Figure 5.9: Main PyBpod graphical user interface window

It uses a hierarchical structure for necessary fields and configuration settings. The first level is a setup or rig. Each rig consists of experiments, subjects, boards, protocols, and users.

Subjects are used to keep track of animals and to map the animals in the data structure to the behavioral files. Animals and users can be synchronized to and from a centralized database. A state machine is connected as a board to the GUI. Protocols are the configuration files for each behavior and are based on a python script, all the states are encoded there. Experiments, finally, combine, boards with protocols. Experiments are executed by the active user with an attached subject to run a session.

For each rig, multiple experiments can be set, with different protocols, but on each board, only one experiment at a time can be executed. This user interface is not designed for visual stimuli, which need dedicated screens. With the gamble-task each user interface is assigned to a single board and rig, thus limiting the centralization of user control somewhat with the already implemented GUI

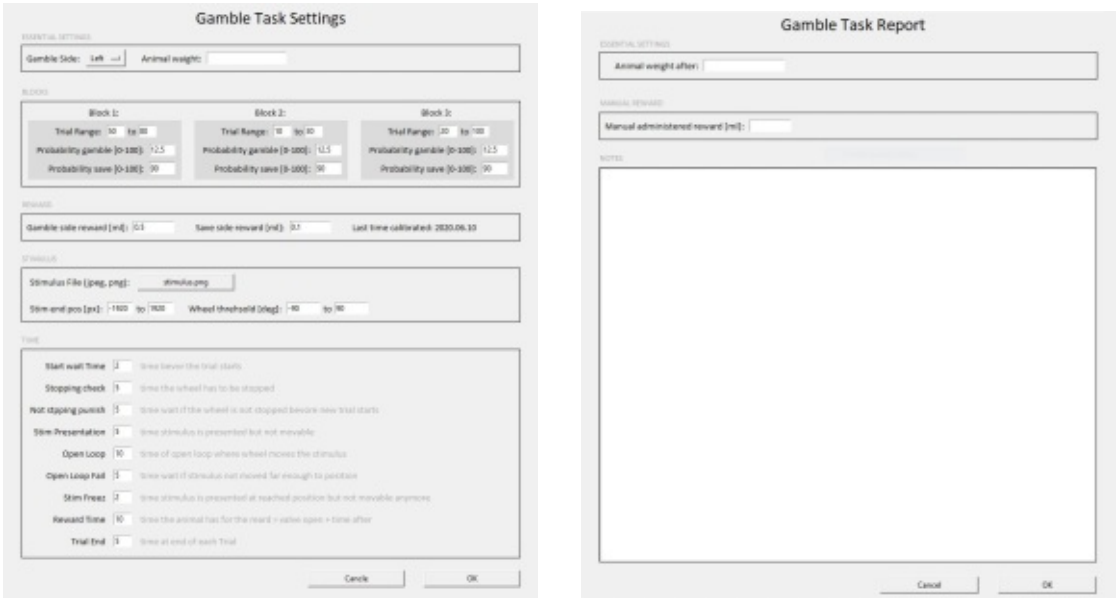
5.4.1 GUI Plugin

The user interface can be expanded with plugins. Plugins can be used to adapt the capability of the UI and to customize it according to individual needs. A generic framework for GUI plugins called `PyformsGenericEditor` is included with the PyBpod library. It can be used to extend and overwrite core functionalities of the Bpod board. Furthermore, does it serve as the graphical output for plotting data from the experiment and trials. A few plugins written by the community are currently available such as a water calibration plugin or a trial timeline plugin.

Plugins are available here:

`https://pybpod.readthedocs.io/en/latest/getting-started/plugins.html`

To allow the experimenter to set and change behavior-specific variables, a custom GUI plugin specifically developed for the gamble-task is used. This plugin was partly based on the PyformsGenericEditor and partly on Tkinter. Currently, there are two different windows implemented, one at the beginning of each session and one at the end (see figure 5.10).



(a) *pre-session input window* **(b)** *post-session input window*

Figure 5.10: Gamble-task specific GUI elements

This expanded user interface allows the experimenter to control the parameters for the probability and reward amounts of each block and set the gamble-side. The length of the different stages of each trial and the stimulus can also be set. In addition, they serve as a reminder for the experimenter to weigh the animal before and after each training session and gather these measurements.

5.4.2 Django Web UI

To address the shortcomings of the PyBpod QT5 GUI, a web-based alternative based on the Django framework is being developed for the gamble-task. This will help to make the control of each system independent of a screen, so all the connected screens can be used strictly for stimulus display to the animal. By switching to a web-UI multiple rigs can be remotely controlled allowing for centralized management, independent of the location. This will help to scale this experiment setup to a much larger number of training- and recording-setups.

5.5 Bpod Software, API and Gamble-Task Implementation

As explained in the initial introduction of the Bpod system, the python-based PyBpod API is used for configuring the state machine. The PyBpod python module is also open-source and licensed under the MIT license without restrictions.

The module is available via the python package index and on GitHub:

`https://github.com/pybpod/pybpod-api`

Experiments are controlled python scripts using the PyBpod API. The main configuration of the state machine is handled by the `pybpodapi.bpod.Bpod` module (see code snippet 5.1). A Bpod object can be created by passing in the COM of the connected hardware device. The COM port can also be read from the GUI via the connected board settings. The main part of the script is to create a set of states which define the behavior for each trial in a loop and flash it to the state machine. As soon as all the states are flashed to the state machine, the Arduino executes the routine. Specific parameters can be changed from trial to trial during the creation of these states.

All the states of one session are created from the module `pybpodapi.state_machine.StateMachine` Bpod object. Each state is defined by: an unambiguous name, a time it will be active (in seconds), a state change condition (from a specific API dictionary), and an output action a tuple (also from an API dictionary) (for example see code snippet 5.1).

```python
bpod = Bpod("COM12") # create the bpod object

sma = StateMachine(bpod) # construct the object state

sma.add_state(
        state_name="signal_bnc1",
        state_timer=5,
        # time state is in seconds, if set to 0 -> change
        # if change condition is met
        state_change_conditions={"Tup": "exit"},
        # conditions under which change to next state
        # => dictionary with keys in ovicial Input Events
        output_actions=[("BNC1", 1)],
        # active which this state envoces must be trouble with ovicial
        )

bpod.send_state_machine(sma) # setd the state machine object to the bpod
```

Code Snippet 5.1: Example of a state, controlling the Bpod state machine via the PyBpod API

If state time is set to 0, the state will only change if the state change condition is met. Each state change condition is a tuple of the change condition and the state to change to if the condition is met. Multiple state change conditions can be set. The state change condition can be the state timer or any other condition. The last state always has to be named "exit".

The output action is the Bpod action, which is active while the state is active. For example, to activate the BNC channel 1 from the Bpod hardware for 5 seconds, a state with that output action is created, the state change condition is set to "Tup" (time up), and the output action defining the BNC1 port as target and activating it with the parameter 1 (0 for deactivating it) (see code snippet 5.1).

A set of all available output actions can be found in the PyBpod documentation :

`https://pybpod.readthedocs.io/projects/pybpod-api/en/v1.8.1/getting_started/output_action_codes.html`

For the gamble-task, changes in the probability of the gamble-side reward will need the same state change condition, e.g. the mouse moving the wheel to a certain position, leads to a different reward state. Also, the probabilistic reward nature at the gamble-side needs a dynamic change in the output action of the reward state. If the probability distribution defines the current trial as a potential reward gamble-trial, the output action of the reward state, of the current trial, will be to open the valve. If it is not a gamble-reward trial, the same reward state will lead to a no opening of the valve. These dynamic state assignments are realized before execution while creating the states for each trial (see code snippet 5.2)

```
for trial in range(settings_obj.trial_num):
probability_dict = settings_obj.probability_list[trial]
sma = StateMachine(bpod)
if probability_dict["gambl_left"]:  #check for gmble side

    if probability_dict["gambl_reward"]: #check for probability of reard
        # big rewaerd
        sma.add_state(
            state_name="reward_left",
            state_timer=settings_obj.time_dict["open_time_big_reward"],
            state_change_conditions={"Tup": "reward_left_w"},
            output_actions=[("SoftCode",
                             settings_obj.SC_END_PRESENT_STIM),
                             ("Valve1", 255)
                             ]
                )
        sma.add_state(
                state_name="reward_left_waiting",
                state_timer=settings_obj.time_dict["time_big_reward_w"],
                state_change_conditions={"Tup": "inter_trial"},
                output_actions=[]
                )
    else:
        # no reward
        sma.add_state(
                state_name="reward_left",
                state_timer=0,
                state_change_conditions={"Tup": "reward_left_w"},
                output_actions=[("SoftCode",
                                 settings_obj.SC_END_PRESENT_STIM)
                                 ],
                )
        sma.add_state(
                state_name="reward_left_waiting",
                state_timer=settings_obj.time_dict["time_reward"],
                state_change_conditions={"Tup": "inter_trial"},
                output_actions=[]
                )
```

Code Snippet 5.2: Dynamic state assignment for each trial depending on probabilities for gamble-reward and gamble-side

The complete gamble-task, with all states and transitions, is visualized with the state diagram in figure 5.11. It is based on the Phenosys task and created by Hugo Malagon.

https://whimsical.com/pybpod-gambling-task-vr-mice-1-block-SbEWn6ZrsCBgPW5Fv1YrxV

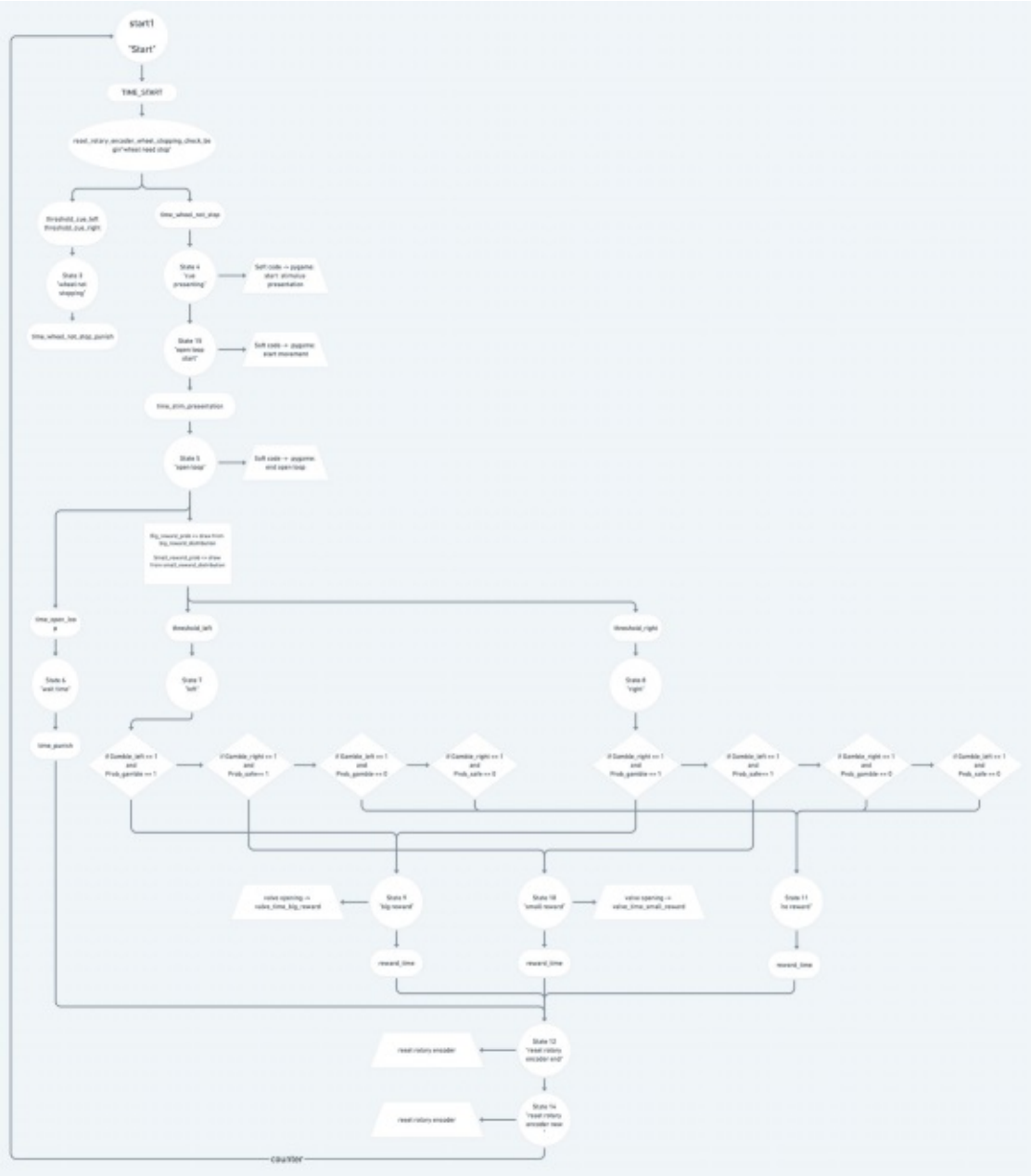


Figure 5.11: Complete state and transition diagram for gamble-task (for better readability open the link) For the cases where the reward probability is not 100%, the outcome is drawn from a binomial distribution with the specific probability set by the current block.

5.5.1 Rotary Encoder Module

The rotary encoder serves two main functions: on the one hand, it triggers a state change if a specific threshold is reached by turning the wheel; on the other hand, it streams the position of the wheel to the connected PC for updating the stimulus on the screen.

The rotary encoder interfaces via the rotary encoder module with the Bpod and the API integration is provided via `pybpod_rotaryencoder_module.module_api.RotaryEncoderModule`. It is adapted via a custom class written specifically for the gamble-task. The rotary encoder is also programmed by creating an object from the custom class by passing in the COM of the rotary encoder module.

There are two ways to interact with the rotary encoder module. One is direct via the python API from the host PC. The other one from the Bpod, which can communicate directly via serial messages with the module. These two communication channels are serving different purposes, some functions overlap, but not all are present in both. The PC channel is used to set specific settings, which stay the same for all trials of one session, while the Bpod channel is used to set trial by trial-specific settings.

The direct communication channel is used to define thresholds for the rotary encoder. If the wheel rotation crosses these thresholds, a signal is sent to the state machine, and the current state is changed. This method allows for a very low latency between rotation and state changes. To properly program the thresholds two steps have to be executed, defining the values and enabling them. The number of possible thresholds is fixed with 8, but not all 8 have to be used.

The values for each threshold are set in relative degrees to the zero position with the `set_thresholds` method (see code snippet 5.3). To allow for continuous position reading of the wheel movement the wrap point is set to 0, which disabled it. The wrap point can be used to account for relative half and full rotations. If set to a value (in increments) the upstream position from the rotary encoder module will be set to 0 if the position crosses this value, marking a full rotation. For the gamble-task implementation, the relative position change is handled by the stimulus program, thus an unmodified position stream of the rotary encoder is preferred.

The thresholds are enabled by passing a boolean array with exactly 8 values, true for the enabled positions and false for the disabled ones to the rotary encoder (see code snippet 5.3). Finally, to use these set thresholds as state change conditions by the state machine, the transmission to the state machine is enabled with the `enable_evt_transmission` method.

```
rc=RotaryEncoderModule('COM4')
rc.set_wrap_point(0) # disables wrap point

rc.set_zero_position() # Not necessarily needed

rc.set_thresholds([-2,2]) # max 8 integers
rc.enable_thresholds([True,True,False,False,False,False,False,False])

rc.enable_evt_transmission()
```

Code Snippet 5.3: Example of threshold implementation for the rotary encoder module

After these steps, the thresholds can be used as state change conditions for constructing states. If the threshold is reached, a serial message is sent to the state machine, which is read, and the state is changed. To encode the specific serial message, the API uses an automatically created dictionary (see code snippet 5.4). Each threshold has a specific name that can be passed into the state as state change condition. The first state of the first rotary encoder is named `RotaryEncoder1_1` etc.

```
state_1="RotaryEncoder1_1"
state_2="RotaryEncoder1_2"

# open loop detection
sma.add_state(
            state_name="open_loop",
            state_timer=settings_obj.time_dict["time_open_loop"],
            state_change_conditions={
                            "Tup": "stop_open_loop_fail",
                            state_1: "stop_open_loop_reward_left",
                            state_2: "stop_open_loop_reward_right",
                            },
            output_actions=[],
            )
```

Code Snippet 5.4: Example of using rotary encoder threshold as state-change-condition

A caveat, which caused a lot of issues for the author is that a threshold, once crossed, is automatically disabled and will not lead to an upstream message, if crossed again. To re-enable each threshold, the rotary encoder has to be reset. Also, the current position of the rotary encoder has to be set as the new zero position, at the beginning of each trial (to detect wheel-not-stopping) and at the beginning of each open-loop, to stream correct position changes and to detect threshold crossing. Both can be done from the PC or from the state machine via specific serial bits message. The serial messages necessary for proper communication between state machine and rotary encoder have to be defined and loaded to the rotary encoder (see code snippet 5.5).

```
re_reset = 1 # can be defined with any number between 1 and 255
bpod = Bpod("COM12") # create the bpod object

bpod.load_serial_message(
    (bpod.modules["RotaryEncoder1"])[0],
    re_reset,
            [ord('Z'), #'Z'(ASCII 90): Set current encoder position to 0
             ord('E'), #'E'(ASCII 69): Enable all position thresholds.
            ],
    )
```

Code Snippet 5.5: Enabling serial message for resetting the rotary encoder

5.5.2 Valve Driver Module & Reward Calibration

Interaction with the valve via the port-breakout-module can be simply done via output-actions during a state. To open the valve the serial message has to be set to 255, (see code snippet 5.6).

```
sma.add_state(
    state_name="reward_left",
    state_timer=settings_obj.time_dict["open_time_big_reward"],
    state_change_conditions={"Tup": "reward_left_waiting"},
    output_actions=[
                    ("SoftCode", settings_obj.SC_END_PRESENT_STIM),
                    ("Valve1", 255)
                    ]
    )
```

Code Snippet 5.6: Example of opening the valve connected the lowest serial port.

To calibrate the deposited reward a simple calibration protocol is implemented.

5.5.3 Visual Stimulus

For the transition from Phenosys setup to the PyBpod setup, the decision was made to not use the Bonsai visual computing program for visual stimuli. Although Bonsai enjoys an increasing adoption rate in the neuroscience community, it is not necessarily the best approach for combining the Bpod hardware, PyBpod framework, and visual stimuli. The lack of available documentation and the overall complexity of the software, are the two main points, why it was not used. One of the main initial tasks, for the new setup, was easy adaptability and both the above-mentioned aspects speak against Bonsai fulfilling this task.

The Visual stimulus for the gamble-task is implemented based on the PyGames framework. The decision to use PyGames was mostly based on the simplicity of the framework. The

system will likely be used for multiple years and the easier it is to implement different behavior paradigms, the smaller the upfront investment will be for new scientists to use the system or to switch to the system. Although in hindsight there are several shortcomings to choosing PyGames over other Python game development frameworks or graphical drawing libraries the main argument of simplicity still holds true.

In the future, the stimulus will probably be switched to the Pyglet framework, which incorporates a lot of the functionality and simplicity of PyGames paired with much more up to date libraries allowing for functions such as true multi-monitor support and support of GIFs. Since it uses OpenGL for drawing, it is also much more responsive and less resource-intensive than PyGames.

For the gamble-task, the Stimulus position is controlled with the position stream from the rotary encoder. Visual stimuli can be displayed spanning multiple displays and up to 60fps guarantee a smooth movement. To interact with the PyGame instance from the state machine during live trials, to start displaying the stimulus precisely when the state changes, the soft-code functionality of the Bpod API is used (see figure 5.12 for overview of the visual stimulus integration).

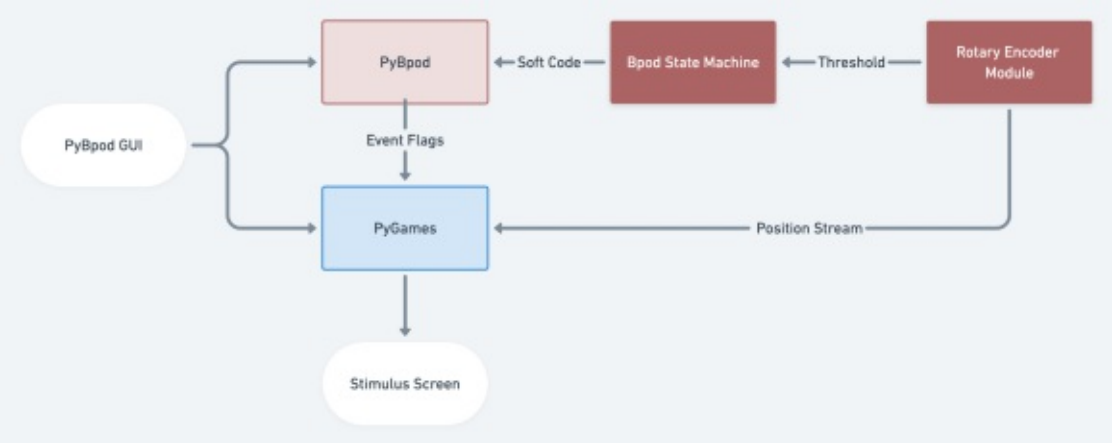


Figure 5.12: Diagram of communication channels for stimulus integration in the gamble-task setup

Soft-code is a way to communicate from the state machine with the connected computer in the form of output-actions. It uses a simple UDP protocol via the USB connection.

Each soft-code is a byte message. To enable communication, soft-codes just have to be used as output actions (see code snippet 5.7)

```
sma.add_state(
    state_name="start1",
    state_timer=1,
    state_change_conditions={"Tup": "start2"},
    output_actions=[("SoftCode", 1)],
    )
```

Code Snippet 5.7: Example of soft-code used as an output action

To use the soft-code sent to the computer by the state machine, a handler function is defined, with custom conditions and actions triggered if the specific soft-code byte is received (see code snippet 5.8).

```
bpod = Bpod("COM12") # create the bpod object

# softcode handler
def softcode_handler(data):
    if data == 0:
        print("soft code o")
    elif data == 1:
        do x

bpod.softcode_handler_function = softcode_handler
```

Code Snippet 5.8: Example of soft-code handler function

This soft-code functionality is used for the gamble-task to start the stimulus presentation, start the open loop, end the open-loop and end the stimulus presentation after the open-loop.

The stimulus can be any PNG or JPEG image, for the current design, it is a green circle. The window consists of a black background, which can span over multiple screens and the projected stimulus image. The initial position is set to be centered in the window both in x and y position, but can be easily modified. Via the specifically programmed gamble-task GUI plugin for the PyBpod QT GUI, the end position on the screen both left and right for the set threshold of the rotary encoder can be set. Furthermore, the relative gain, between the stimulus movement and the rotation of the rotary encoder to increase or decrease the relative speed of the stimulus to the wheel movement is also modifiable via the GUI.

To deploy the Python script controlling the PyGame the multithreading module is used. During the initial development, the Python multiprocessing module was used to run the PyGames loop on one core and the PyBpod loop on another core. Communication between both was implemented using simple multiprocessing event flags, but a persistent bug in the event flag based communication to the PyGame instance could not be resolved. This bug was not encountered with the multithreading library. The computing load of both threads is still low enough to not encounter any noticeable delays for multithreading, but depending on the complexity of the stimulus true parallel processing might be favorable in the future.

5.6 Training Setup

Each setup consists of the rig, the controller PC, soundproof box, and additionally for the recording setup, the electrophysiology hardware and air table. There are two different types of rig designs: the training rig and the recording rig. The recording-rig uses, in contrast to the training rig, a modified mouse holder with a stereotaxic frame, providing the necessary rigidity and attachments for the electrophysiology hardware. The training rig is much more cost-efficient and has a smaller footprint. It is only used for training animals since it lacks the frame and adapters to perform in-vivo electrophysiology recordings. All systems are sufficiently isolated from the environment to be able to run independent experiments beside each other, without any influence among them. This allows for easy scalability of the system. The training rig is also designed to be modular, to allow for stacking of multiple systems both in x and y direction, to maximize space utilization. For example, 9 training-setups (3x3) can be placed on a 2m x 0.6m x 2m rack easily fitting in a small laboratory room. A lab deployment would ideally comprise at least one recording setup and multiple training setups.

The rig is influenced by the previous Phenosys setup and by the setup of the International Brain Lab (The International Brain Laboratory, Aguillon-Rodriguez, *et al.*, 2020). 3D printed materials are used where constraints allow for it and otherwise, CNC machined metal parts are deployed. A high emphasis is placed on making the whole setup as easy as possible to assemble and to replicate while keeping overall costs low.

Each training setup comprises a mouse holder, reward system, stimulus system, soundproof box, controller pc and bpod setup (see figure 5.13).

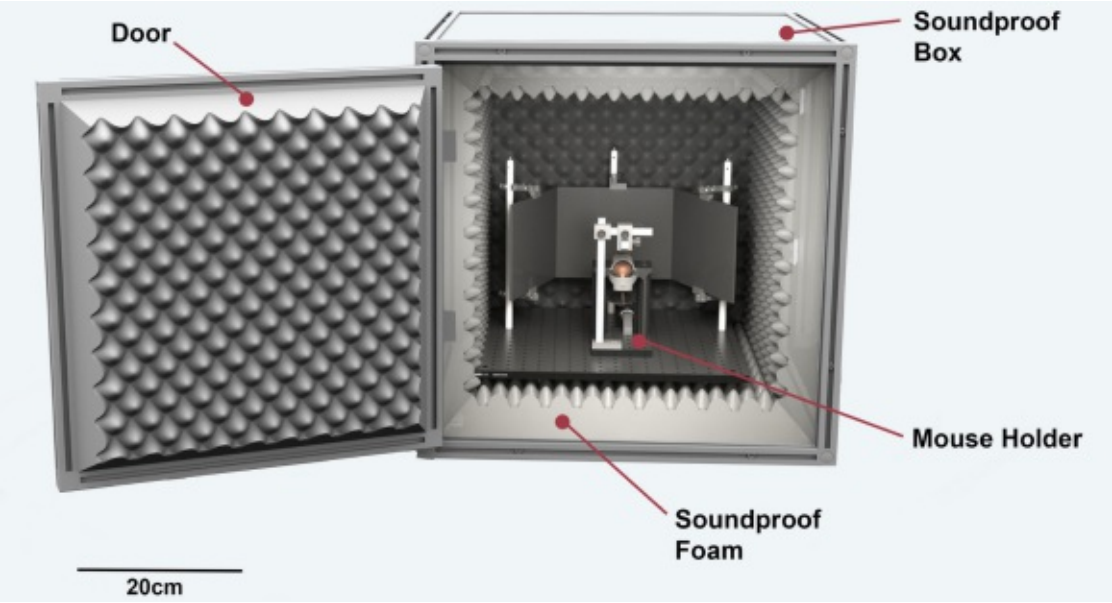


Figure 5.13: Overview of the training setup

5.6.0.1 Mouse Holder & Reward System

The reward system and mouse holder are integrated. The internal part of the reward system, the mouse holder, and the head plate holder are all attached to the same aluminum plate (see figure 5.14), thus allowing for easy handling. The whole assembly can be moved out of the soundproof box by the experimenter during mouse handling to make it much easier to attach the mouse to the head plate holder and to place the reward correctly.

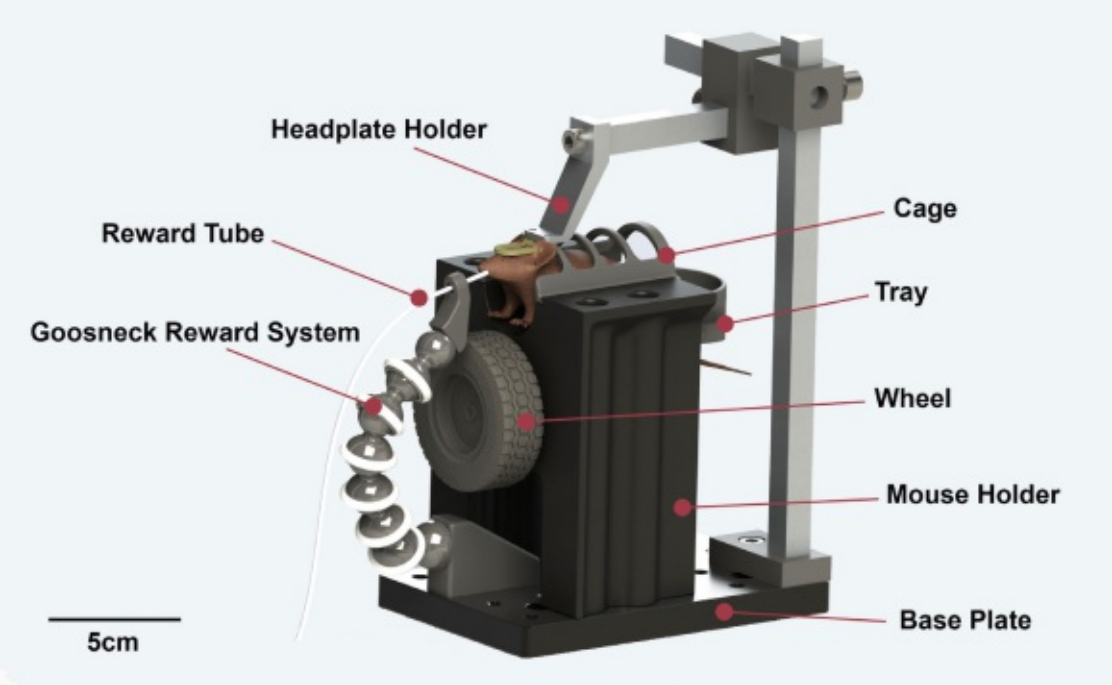


Figure 5.14: Reward system and mouse holder for training rig

The mouse holder serves as a rest for the mouse while containing excessive movement via a restrictive cage. It is based on a modified design from the Scientific Hardware Platform from the Champalimaud Centre for the Unknown (*A Standardized and Reproducible Method to Measure Decision-Making in Mice* 2020). To reduce contamination and necessary cleaning, potential excrements are collected by a removable tray, which can be easily detached and cleaned after each session. The second purpose of the mouse holder is to hold the rotary encoder with a LEGO wheel attached to it. The wheel serves as a steering wheel for the mouse. The rim of the LEGO wheel is replaced by a custom 3D printed rim, that can attach to the shaft of the rotary encoder.

The reward, controlled via the above-described valve-driver and the solenoid valve, is delivered from a reservoir mounted outside on the soundproof box and connected via a medical-grade silicone tubing to the valve. The valve is also placed outside of the box since the opening produces an audible clicking sound. The outflow port of the valve is connected via the tubing

to a bent metal pipe, that is held in place by a gooseneck, such that the tip is placed on the month of the head constraint animal inside the setup. The gooseneck allows for easy placement of the tube by the operator. This is of interest, since depending on the mouse and the position of the head-plate holder, the final position will vary slightly. The attachment of the gooseneck to the breadboard and the connector to the reward pipe are both 3D printed.

For calibration, the reward tube, still attached to the gooseneck, can be placed above a container on a scale with a resolution of 0.01g. The scale used is the Ohaus SPX222 Scout Portable Balance, with a USB interface.

To monitor the animal during training a camera connected to the controller PC is placed close to the animal holder. A 8 MP camera with an automatic IR cut filter and IR LED lighting is used for sufficient image quality to monitor the animal in the relatively dark active training environment.

5.6.0.2 Stimulus System

The stimulus is displayed on three screens to fill most of the animal's field of view and to provide an immersive experience (see figure 5.15).

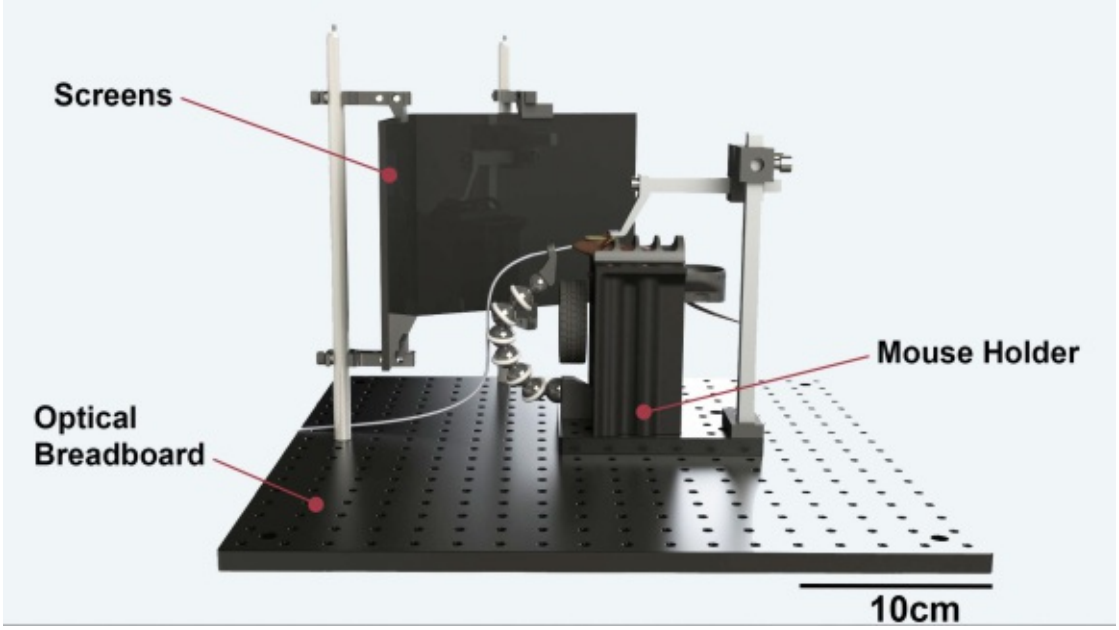


Figure 5.15: Mouseholder and stimulus screens (3rd screen removed for better visibility of the rendering)

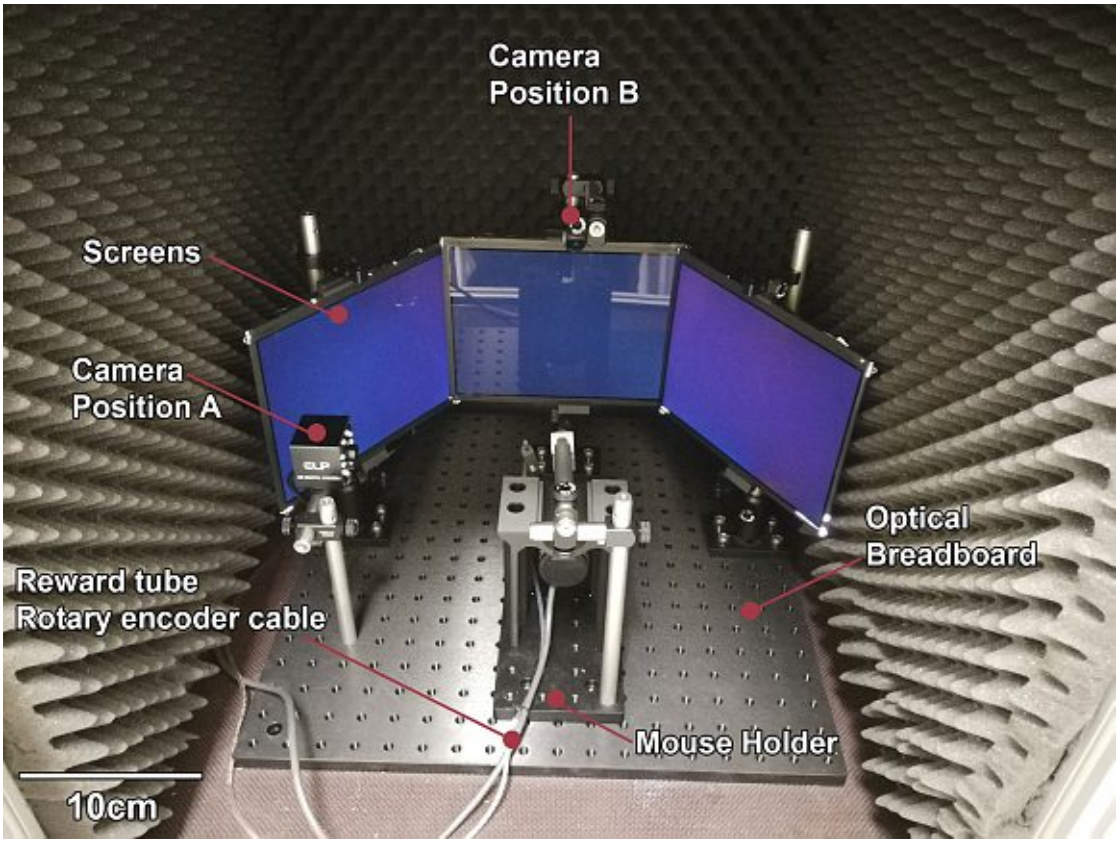


Figure 5.16: Overview of the stimulus system with three screens

The screens are based on the 9,7" iPad LCD panel with a resolution of 2048 x 1536 pixel, both the panel and the necessary driver board are available from Adafruit and are comparatively inexpensive. The advantage of using panels instead of complete computer monitors is the form factor. The panels are mounted on an aluminum rod each with two 3D printed adapters.

The rods are attached to a large 250cmx250cm aluminum optical breadboard, which also provides a plug-interface with the mouse holder to guarantee proper alignment while allowing for easy removal.

5.6.0.3 Soundproof Box

Each box has the external dimensions of 60x60x60cm and is made of Nut6 30x30 Bosch profiles aluminum extrusion, and polyurethane coated 8mm baltic birch plywood (screen printing plate). For sound insulation, a sandwich composite of high-density composite foam, bitumen foil, and studio-grade acoustic convoluted foam is used. The high-density composite foam combined with the bitumen foil guarantee, due to high mass, high acoustic isolation against the environment. In addition, the acoustic convoluted foam absorbed mid and

high-frequency spectrum to dampen the reflected sounds inside the chamber. The foam is glued to the inside walls except on the bottom, where it is loosely placed allowing for easy removal to clean particles from the animals.

The large breadboard (attaching screens and mouse holder) is mounted on cylindrical spacers slightly above the bottom foam. Doors can be closed with a lockable hinge and are isolated against sound leakage with regular window sealing tape. To the panel of the front door a whiteboard foil is attached, to allow for annotating and quick marking of information on each box.

The soundproof box does not only increase the independence of the system by isolating it against the surrounding and enable multiple experiments to be run in the same room, but it also allows the systems to be stacked. Multiple boxes can be stacked on top of each other in a rack, thus increasing the usage of space and enabling a higher throughput of animals. An example of such a application is shown in figure figure 5.17.

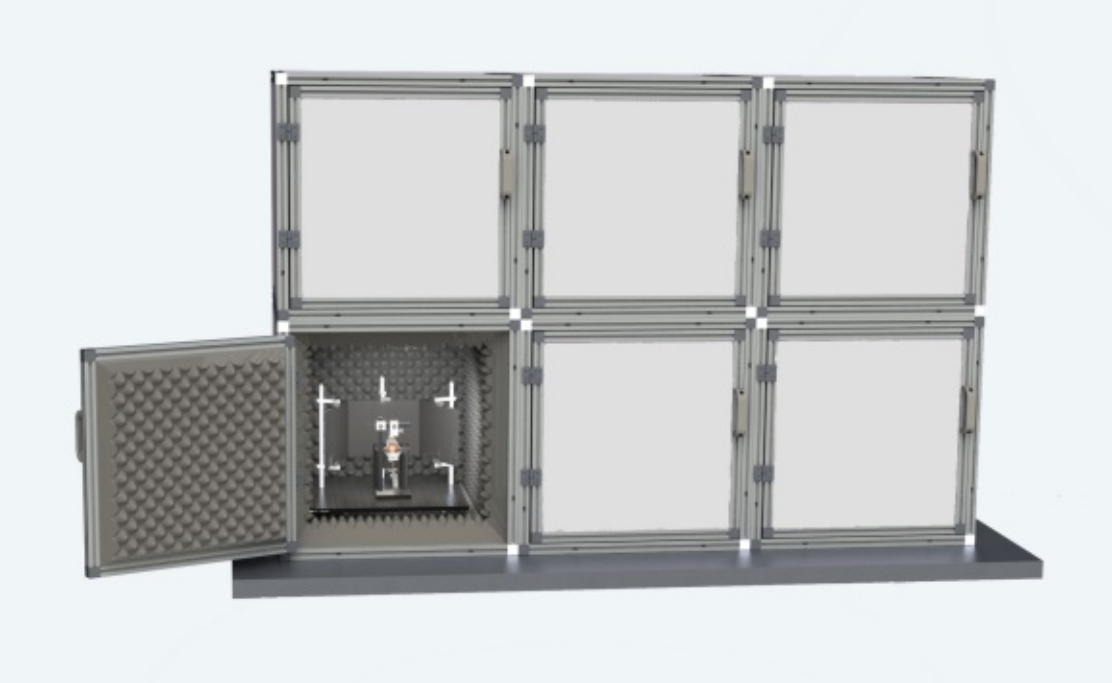


Figure 5.17: Multiple training boxes stacked for efficient usae of lab space

5.7 Recording Setup

The recording setup has different requirements compared to the training setup. It is not designed for minimal spatial requirements and stackability, but to give the experimenter the necessary space and access to the animal. It must also fulfill the much higher stability requirements demanded by the electrophysiology recording system. But the reward system, and stimulus system are the same as for the training-setup. In contrast, to fulfill the higher constraint a stereotaxic frame from Kopf serves as the central attachment point. The recording setup also comprises the complete electrophysiology system with the OpenEphys acquisition board and an Olympus SZ61-TR stereo microscope, mounted to a Leica M320 microscope floor stand. To dampen low and mid-level vibrations between the system and the building it is placed on an air suspension table with a breadboard top by Supertech Instruments (see figure 5.18).

Picture of complete Setup + microscope + box usw

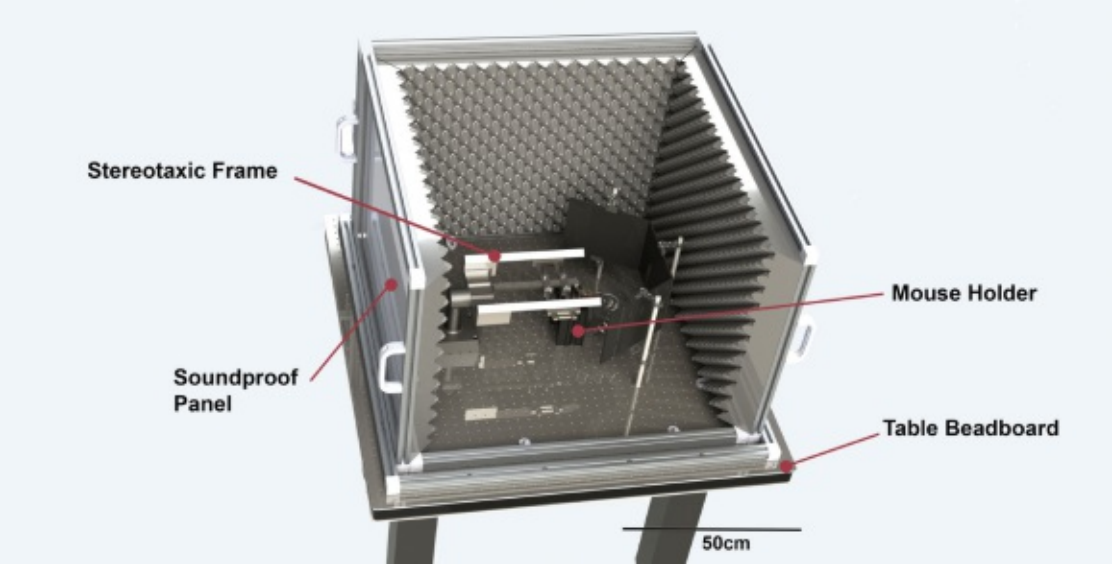


Figure 5.18: Overview of all the components of the recording setup

5.7.0.1 Stereotaxic Frame

Based on the experience with the Phenosys setup, the same stereotaxic frame DKI 1430 from Kopf (*Model 1430 Stereotaxic Frame | Kopf Instruments* 2021) is used. It forms the stable base to which the animal is head-fixed. The actuator holding the head-stage and electrode shanks is also attached to the frame via the 4 axis micromanipulator 1761 from Kopf (*Micro Manipulators Models 1760, 1760-61 | Kopf Instruments* 2021). The micromanipulator is necessary for pre-positioning the electrode holder relative to the entry point and to set the entry angle determined by the final position in the brain to be recorded from.

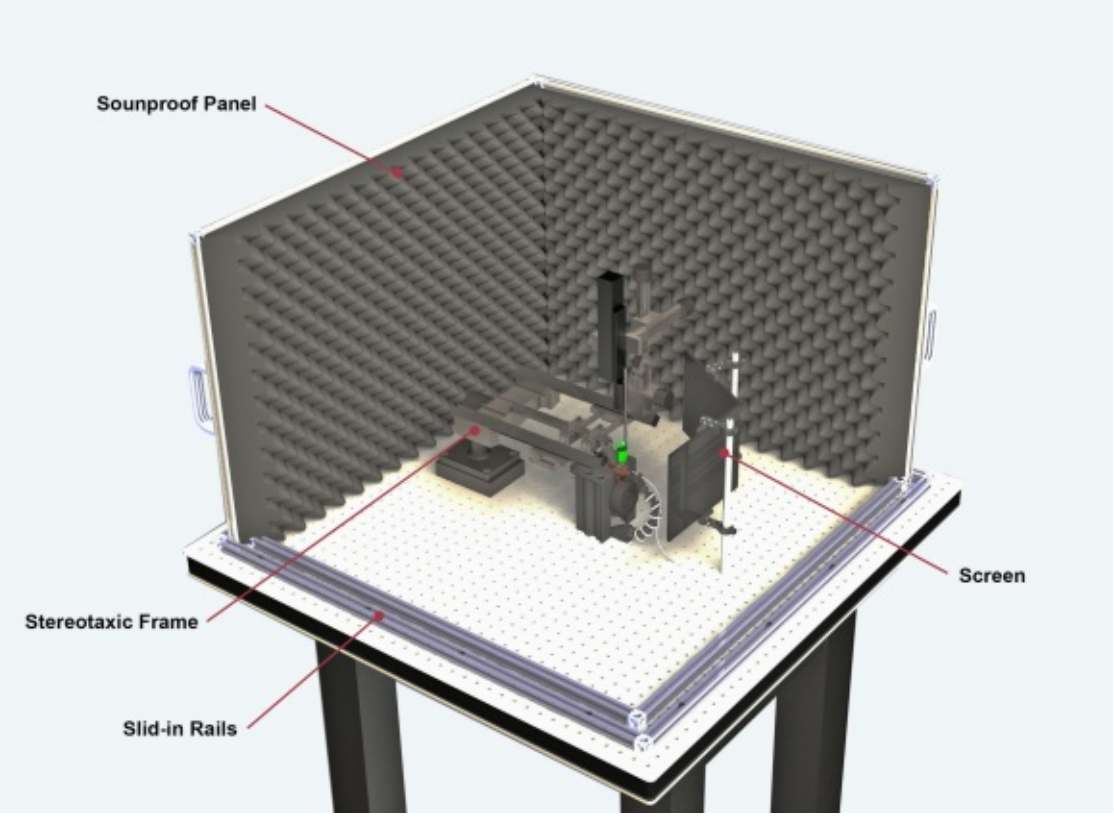


Figure 5.19: Stereotaxic frame and micromanipulator in the recording setup

It is possible to also attach additional micromanipulators to the frame since it has two rails. This would allow for an even higher number of recording sites by inserting multiple electrode probes into different locations at the same time.

5.7.0.2 Head Plate Adapter and Holder

To actually head restrain the animal the skull of the mouse has to be rigidly connected with the stereotaxic frame. This is done via an adapter plate (see figure 5.20) milled from fiber-class composite material surgically attached to the skull of the mouse via dental cement. To allow for the insertion of recording electrodes the adapter has a cutout in the center (see figure 5.20). The adapter plate is based on a design from the IST Miba Workshop (*IST Austria | Miba Machine Shop* 2021). Several different versions were designed, manufactured, and tested.

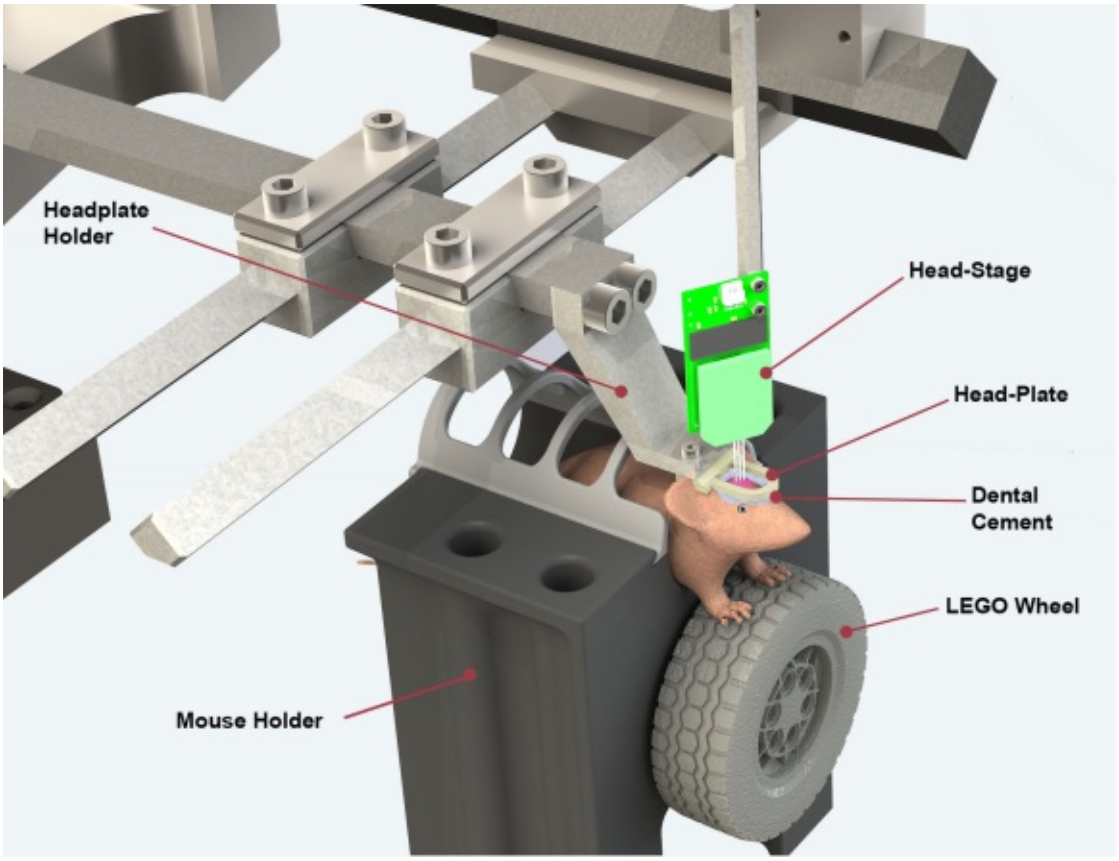


Figure 5.20: Detailed rendering of the recording system, the Mouse Holder and the new Head Plate attached via the new Head Plate Holder to the Stereotaxic Frame, the reward system is removed for better visibility

To interface the adapter-plate with the Kopf frame, an adapter is designed and manufactured from high strength tool steel (see figure 5.21). The adapter allows for free adjustment of the position in the x-y plane, allowing for perfect alignment of the animal. If different viewing angles are required in the future the front part of the adapter can easily be switched with a new one, engineered to accommodate the new angle.

Rudimentary finite element simulations for deflection under load from the animal's head and mid-frequency vibration coupling through the adapter were performed using Altair HyperWorks. Both were neglectable and thus the design was realized.

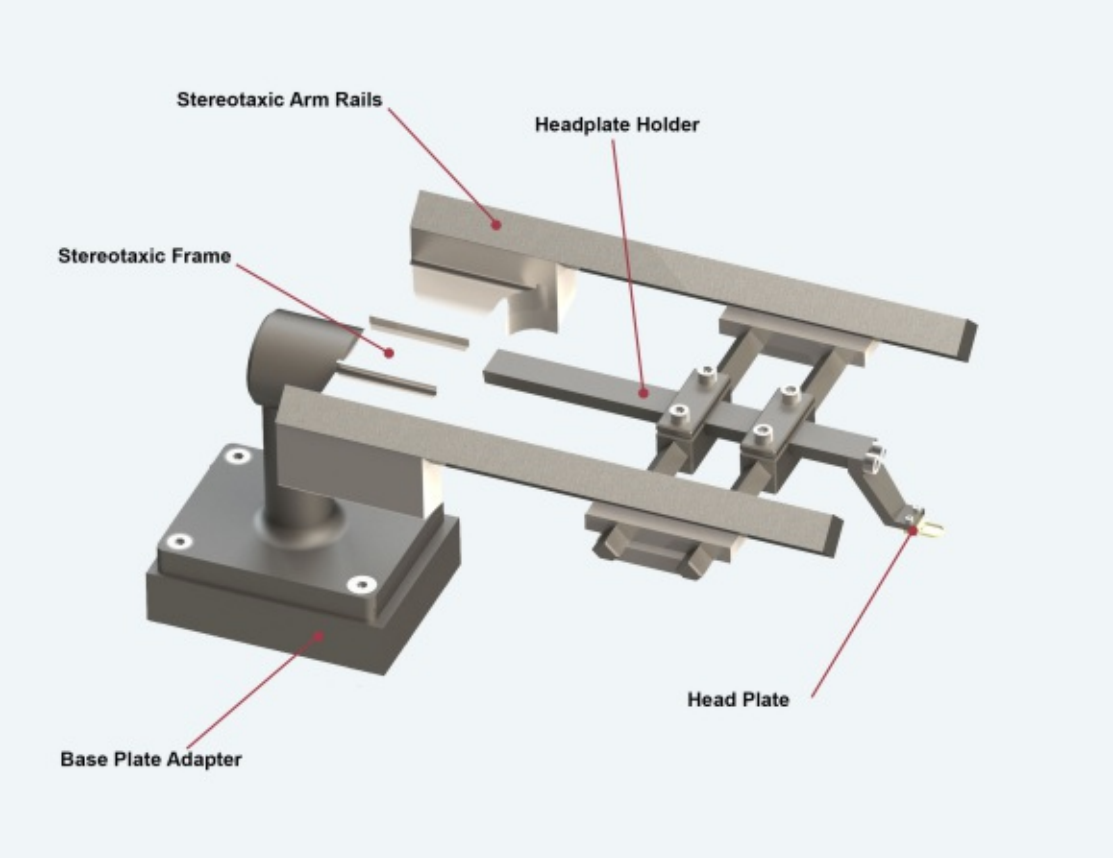


Figure 5.21: Stereotaxic Frame and new Headplate Holder, and new Head Plate

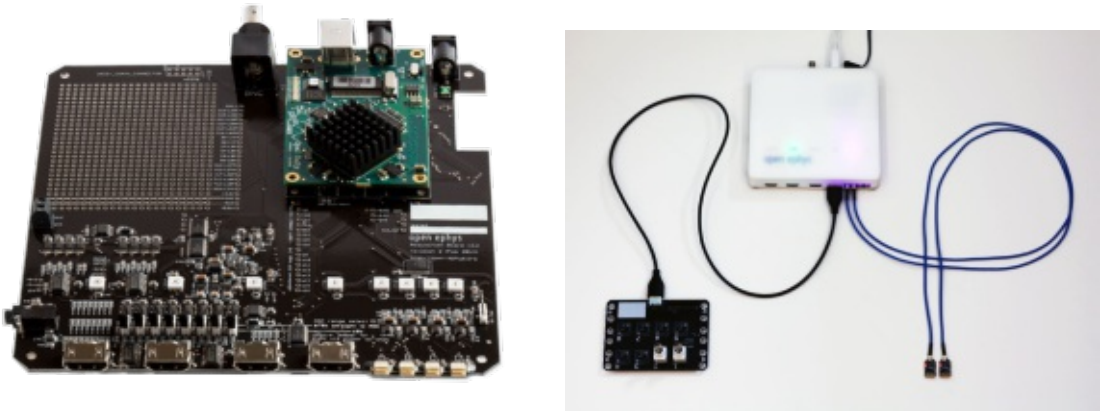
5.7.0.3 Soundproof Box

The enclosure for the recording setup is based on a modular system. It allows for the necessary access from different angles to the animal while still allowing for quick assembly, to provide complete acoustic isolation during recording sessions. Each side panel and top panel can be independently removed. This is especially necessary to allow for sufficient access with the microscope to positioning the electrode shanks. To attach the sides, they snap into a grove on the tabletop and can be connected on the top together with a quick-release system.

The panels are also constructed from Boschprofil aluminum extrusion, but to keep the individual panels as light as possible and allow for easy handling, smaller Nut6 20x20 and 6mm plywood is used. The soundproof foam sandwich construction is the same as with the training boxes.

5.8 Electrophysiology Recording System

The main purpose of the recording-setup is to perform in-vivo electrophysiology recordings to measure single neuron activity. Although the same head-stage by Intan and probes from NeuroNexus as well as the linear micromanipulator from Scientifica (*Scientifica IVM Single Motorised Micromanipulator* 2021) are used, the acquisition board is switched to OpenEphys (Siegle *et al.*, 2017) (see figure 5.22). This decision was based on a significant cost advantage and also better integration and future expandability of the OpenEphys acquisition board, compared to the Intan. The OpenEphys is nearly 5 times cheaper, if bought assembled, and it is also open-source hardware and can be self-assembled to future reduce costs. By switching to the new acquisition board, Neuropixels (Steinmetz, Aydin, *et al.*, 2020) ultra-high-density probes from Imec with over 1000 individual recording sites can be used. The OpenEphys is widely used with Neuropixels and is especially suited for such high-density probes.



(a) *OpenEphys acquisition board (*OpenEphys Acquisition Board *2021)*

(b) *OpenEphys breakout board (*Enginursday: Open Ephys - News - SparkFun Electronics *2021)*

Figure 5.22: OpenEphys hardware

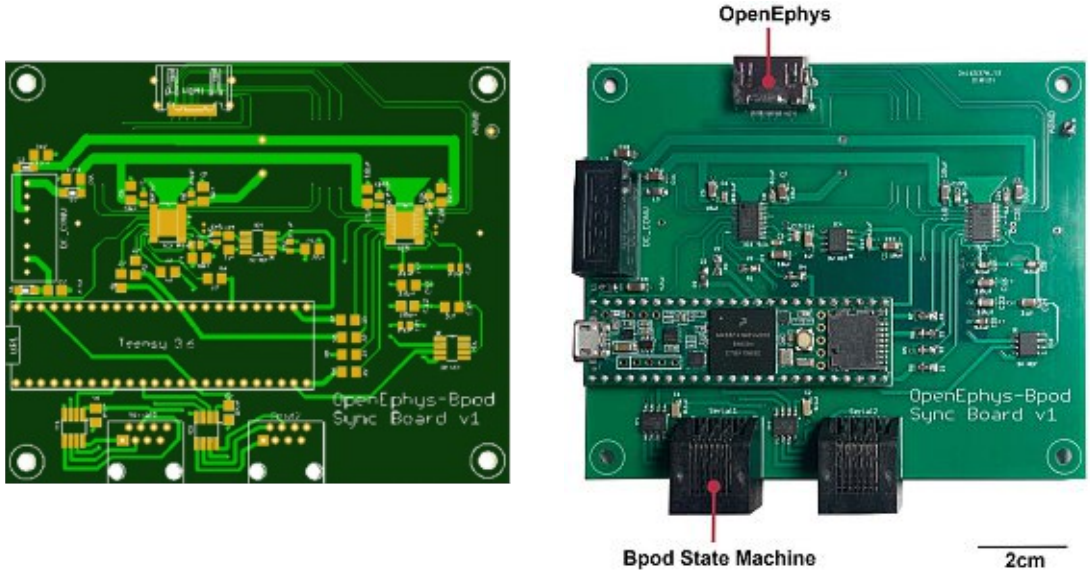
It uses the same industry-standard Intan chipset as the Intan acquisition unit and can stream up to 512 channels via USB3 to the driver software. It is possible to connect up to four 128-channel head-stages to a single OpenEphys acquisition board (see figure 5.22).

The acquisition board is controlled by the experimenter from the same PC that controlled the PyBpod GUI, via the OpenEphys GUI.

5.8.1 Bpod OpenEphys Synchronization Board

To enable event synchronization and connection between the Bpod state machine and the OpenEphys acquisition board a custom interface board was designed and manufactured as

well as a subroutine implemented in the PyBpod behavior script. The circuit diagram of the new interface board is based on the Bpod 8 channel digital output module from Sanworks, but modified to directly interface with the HDMI connectors provided by the OpenEphys (see figure 5.23). The PCB was ordered from the overseas manufacturer JLCPCB and components were hand soldered to the board. For more details request the project-work of the author .



(a) *Design of the new synchronization board PCB*

(b) *Synchronization Board with components*

Figure 5.23: Custom OpenEphys-Bpod sync board for percise event synchronization

For further integration regular 8 channel BNC breakout boards can be connected to the OpenEphys board and synchronize events via TTL signals. These boards are ready assembled and implemented with the behavior task.

5.9 Data Standards

In addition to the Bpod state machine and the software controlling it, standardizing and integration the generated data and processing pipeline was a major goal of the new setup. A problem with the data part of the Phenosys setup and the surrounding lab ecosystem is the heavy customization of each and every experiment. On the one hand, this gives every scientist the creative freedom to realize his ideas and implement her approaches but the major downside is the lack of horizontal compatibility. Everybody basically writes his own scripts and they are barely compatible with each other. This leads to a lot of redundancy and overhead since multiple people have to develop the same things. Furthermore the brain-drain and lost information and insights into an experiment if a scientist leaves the lab is inevitable. A more standardized approach for where information is saved, how data is stored, and processing scripts are developed will lead to better transparency and make it easier for new scientists and lab members to ramp-up their understanding of the past experiments. A significant benefit of such a standardization will also be that analysis pipelines are much better cross-compatible and don't have to be developed from scratch for each new experiment. The initial question was what kind of data framework the new setup should be based on. It has to be standardized to allow for straightforward application of data analysis tools but flexible enough to preserve the freedom to realize all the individual needs each experimenter will have.

Similar to the different available options for the behavior part of the gamble-task, there are multiple options already available and used by the neuroscience community. Neurodata Without Borders (NWB) (Ruebel *et al.*, 2019; Teeters *et al.*, 2015), International Brain Laboratory (*International Brain Laboratory* 2021) and OpenNeuro (*A Free and Open Platform for Sharing MRI, MEG, EEG, iEEG, and ECoG Data - OpenNeuro* 2021) are the most promising projects. All try to define frameworks oriented on other large scale data-gathering endeavors such as in nuclear physics or in molecular-biology. The aim is to foster collaboration of data and methods across tools, to enable comparison across labs and to speed up the overall progress of cognitive neuroscience.

The Neurodat without borders data format is by far the most widely used framework in the neuroscience community. It was started in 2004 with the goal of creating a unified data format for cellular-based neurophysiology data across different laboratories. The data format defines not only a wide variety of data standards for raw and meta-data, but also analysis methods (see figure 5.24). There are currently a broad range of software tools compatible with the data standard and more and more are adding out-of-the box compatibility.

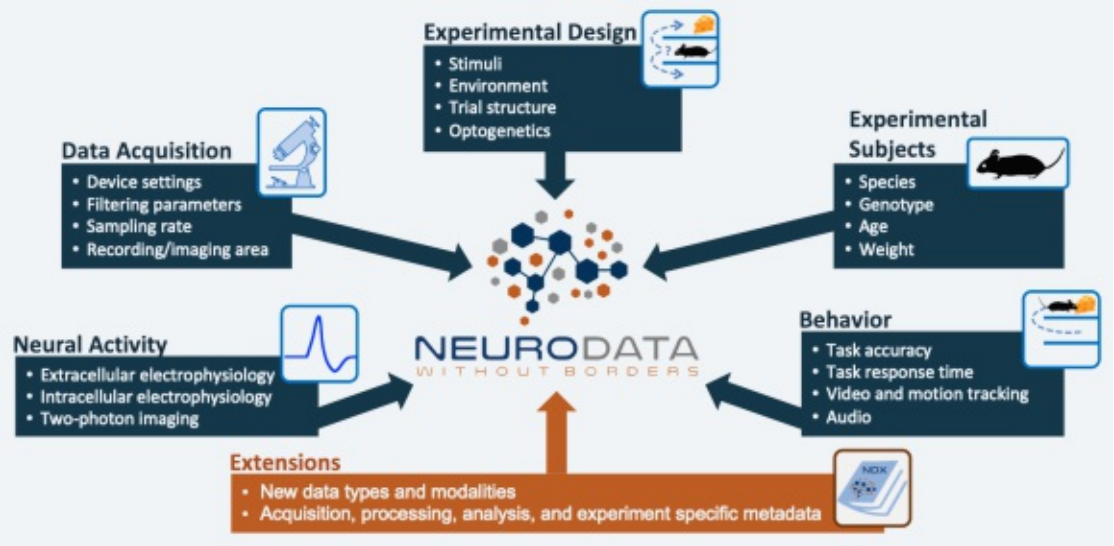


Figure 5.24: Neurodata Without Borders integration schematic adapted from (*The NWB* 2021)

Since the standard is a file format relying only on it, the data storage would be file-based, similar to the Phenosys setups. Although such an approach is relatively easy to implement and requires little to no pre-training and knowledge of the user it has some shortcomings. The biggest problem of file-based data storage is accessibility and version control. Since multiple scientists are going to collaborate on the production and analyze the data, a system that inherently synchronizes versions across all users and tracks changes is vital. An option with the file-based NWB approach would be to use a secondary version control and synchronization tool such as Git, but this requires a high level of discipline. Not everybody will have the same level of data discipline and everyone has a slightly different approach, thus such a system might lead to loss of integrity of the data.

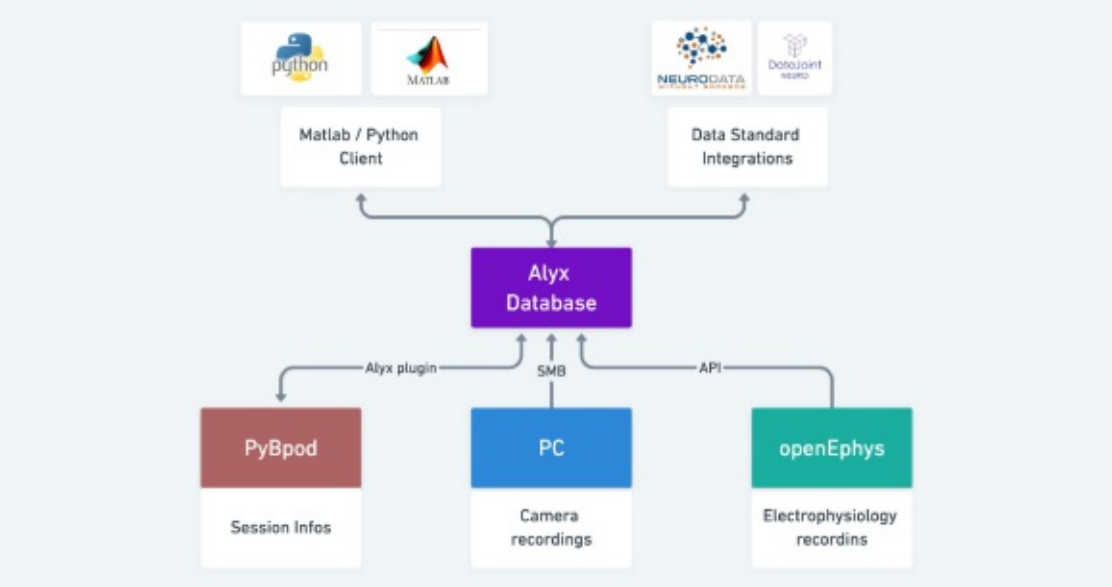


Figure 5.25: Schematic of data flow in the new gable-task setup

5.9.1 Alyx Database

A centralized database, from which data can be queried by each user and to which all setups and users synchronize their data is the ideal approach. Interestingly the International Brain Lab is actively developing exactly such a solution, the Alyx database (*Cortex-Lab/Alyx* 2021).

It is a Postgres database with a Django web frontend for storing and managing data for neuroscience labs. The database stores data closely resembling the structure of NWB, and data can also be exported to the NWB file format. To interact with it from the user side there is a Python and Matlab client available that can retrieve data via the REST API. Also a user plugin for PyBpod to sync Bpod data to the Alyx database is available.

Although it is far from complete the decision was made to not develop a custom solution integrating the NWB data standard with a central database, but to adapt the already existing Alyx database to fit the specific needs of the new gamble-setup. The most relevant decision factor was the integrated REST API, and the user clients that allows for quick integration of data analysis apps and scripts for custom analysis pipelines.

The Alyx database adapted for the gamble-task acts as the central hub to which data is fed and from which data is retrieved for analysis (see figure 5.25).This database stores behavior data from all training and recording sessions and raw data from electrophysiology recordings. In addition to the already available data, the new system also handles general animal tracking, such as weight, surgical procedures, training stages and much more. It will replace the traditionally used lab-books and allow for a higher automatization of data analysis.

5.9.1.1 Implementation of the Database

For scalability, and to allow for easy management, the database is deployed in a container environment. There was the option between multiple Docker containers managed by Docker Compose deployed on a Linux derivate running on a bare metal server or to use LXC containers running on the Proxmox hypervisor. Since Proxmox provides an easy to operate web UI, which will be beneficial for future administration, Proxmox was chosen as the main hypervisor. The additional advantage of using Proxmox is that it can be used with GPU passthrough for Cuda accelerated spike sorting with Kilosort, more on that in the Processing Pipeline chapter. The Proxmox is deployed on an Intel i7-8700 six core 4.6 GHz CPU, with 32 GB of non-ECC RAM. Two Raid-Z arrays are used for storage, one Z5 array with 5 3TB HDDs and one Z1 array with two 1TB SSDs.

The Alyx database, deployed on a LXC container, requires a python distribution as well as a PostgreSQL database and an Apache webserver. Installer files and instructions are available on the GitHub page (*Cortex-Lab/Alyx* 2021) of the project. A Letsencrypt instance is also deployed to generate signed SSL certificates to encrypt the Alyx DB webpage.

After installing and configuring the web server, the database has to be customized. It will provide out of the box, a set of standard data fields such as users, labs, experiments, but all the other parameters have to be individually set. The current setup uses a minimum of necessary data fields, but during the first phase of active usage, more and more customization will be performed. The database can be managed and all tables accessed via the web UI.

To synchronize both the training-setup and recording-setup with the database, the PyBpod Alyx Module (*Pybpod/Pybpod-Gui-Plugin-Alyx* 2019) is used. It implements a native method to communicate via the PyBpod GUI with the REST API of an Alyx database instance. After logging in with a valid user via the PyBpod GUI, animals assigned to the user are automatically synchronized with the setup in the PyBpod GUI. To push data from PyBpod and OpenEphys to the database, a custom python script maps specific data to data fields in the database. This script is used to push the newly generated data to the database after each training and recording session. As of the current status, the script is in active development and still needs some refinement.

A possible option is to also upload previously generated data to the database, to allow for the same easy access to historic data.

5.10 Processing Pipeline

To make use of the new framework for the gamble-task a large amount of the postprocessing can be automatized and integrated. Analysis scripts can be also included in this pipeline. This not only speeds up the time from recording until the first insight into the data is available, but also allow for

Recorded data, that is pushed to the Alyx database, can be automatically sorted by Kilosort. If new data is uploaded, a script can be triggered that automatically loads the necessary configurations, depending on the head-stage used, to a Kilosort instance and starts the spike sorting process. The output of Kilosort is synchronized back with the Alyx database, and the data is ready for manual sorting with Phy2. The Kilosort instance runs on a headless Matlab installation, on a virtual machine on Proxmox on the same server that hosts the Alyx database. The necessary GPU acceleration by Nvidia Cuda is still possible, since Proxmox allows for GPU passthrough to virtual machines.

Behavior analysis and visualizations are made available even faster, since they don't depend on the spike sorting step. It is crucial for the experimenter to track the progress of his animals during the training. Custom plots, like the ones discussed in chapter: Results: Data Analysis, help by visualizing the behavior of the currently trained animal. These plots can be automatically generated, by python scripts, running in a LXC container on the server, and stored in custom fields in the database.

After manual clustering, the experimenter can use the Python or Matlab client to query the new session and perform manual steps of exploratory data analysis. To aid in this process a custom Python library based on the analysis performed by the author on already recorded session (see chapter: Methods) can be used for a high degree of automatization, with minimal manual steps.

Statistical analysis scripts can also be integrated with the pipeline to automatically process data, once the necessary manual preprocessing is finished. In the future modeling of behavior using classical models and machine learning approaches can also be integrated with the pipeline.

Discussion

This chapter will present the authors interpretation of the results. First a conclusion will look at the initial questions and to what answers the results led. Second, a detailed discussion, following again the the same two part approach from previous chapters will present the interpretation. First it focuses on the discussion of the data analysis part, and second on the results from the new virtual reality setup.

6.1 Conclusion

Question 1:

Does the virtual-reality gamble-task implementation based on the Phenosys system displays proper function and are both behavior data and in-vivo electrophysiology data generated by the system correctly synchronized?

Answer:
The data analysis of previously recorded sessions could show, that the Phenosys system and Intan recording system both performed as expected and reliable behavior data and neural data is generated. However, the synchronization is not 100% reliable and leads to some inconclusive outcomes. Mostly the significant difference between increase of firing rate of reward / no-reward responding neurons before and after the occurrence of the event casts doubt on the correct synchronization of events. Further investigation is necessary to determin if data from the Phenosys system can be used, but it is very likely that under certain constraints the data does not have to be disregarded, but instead can lead to conclusive insights.

Question 2:

Are putative neurons in the electrophysiology data, previously recorded with the Phenosys system from the gamble-task, that have firing rate changes correlated to specific behavior

events, and how is such a potential correlation depending on reward probability, gamble or safe-side and the respective combinations thereof?

Answer:
A clear correlation for a subgroup of neurons between firing rate change and reward / no-reward event could be shown. The results from the statistical data analysis could proof, that these neurons respond significantly differently towards rewarded and not-rewarded gamble-side trials. Whether this correlation is due to common causation still needs to be answered by more complex analysis procedures.

Question 3:

What would be the design, development, and implementation of an updated system for the virtual-reality gamble-task for animal training and in-vivo electrophysiology recording, that comprises hardware and software which can be easily configured for modified, integrated with the existing lab-ecosystem and extended for future behavior-tasks?

What data standards and frameworks for encoding, storing, and analyzing data produced with the updated system are available, and how can the most suitable one be implemented?

Answer:
Both tasks of designing and implementing a new behavior system and a standardized data framework could successfully be proven. The behavior system already shows very promising results, and fulfills all the set requirements towards usability, adaptability, and integration. Reliability over long term has yet to be proven, but since most components already have been tested and proven reliable by a wide range of tasks by the community, the author is confident that despite inevitable minor bugs the system will perform over the long run.

6.2 Interpretation Data Analysis

6.2.1 Behavior Analysis

Nearly all analyzed sessions showed a significant increase in no-response trials towards the end of each session. This bulk of no-response trials at the end, could be interpreted as disengagement of the animal from the task and might be linked to several reasons. It can be simply due to satiety by the already received reward and thus a steadily decreased desire to participate in the task. It might also be linked to exhaustion and or frustration. The increased response time can be seen as an indicator for frustration and exhaustion manifesting in not clear choices by the animal compared to previous trials, given the same external cues as indicators for decision making.

The reason for the trial-by-trial occurrence of such a disengagement is likely the same as for the session where such an occurrence is clustered towards the end. It could be related to a combination of exhaustion and disinterest by the animal which leads to frustration by changing reward probabilities and conflicting external indicators with the internal decision guidelines. This conflict, between feedback and the internal model, leads to an increased response or no clear response at all. By failing to clearly choose a side and responding to the trial the animal will not get feedback and thus further decreasing the amount of information available from the choice environment to correct the internal model, which inevitably leads to a higher level of ambiguity.

The very frequent occurrence of trials, where the mouse failed to stop the wheel during the necessary period, in the beginning, is related to the level of training of the animal. The amount of these trials slightly decreases over the duration of training an animal in the final form of the behavior-task. It will probably not be possible to train a mouse to such a level, that significantly fewer of these trials occur over the duration of a session. Interestingly is the fact, that contradictory to the expected outcome, these failed trials are more or less evenly distributed across a complete session. The author would have expected the occurrence to be similarly distributed like the no-response in time trials, and accumulating towards the end of a session. Since this is not the case, the underlying mechanisms leading to the wheel not stopping trials and no response in time trials likely is different. A general lack of focus towards the behavior task by the animal argued above as one possible reason for the no response in time trials would also suggest an aggregation of the wheel not stopping trials towards the end since the focus required in general is the same. The different distributions suggest, that an increase in lack of focus is not the main contributing factor.

Furthermore is the behavior of the animals, in regard to the decision ratio of gamble-side compared to safe-side significantly depending on the probability in the block, with an expected preference for the safe-side during 25% and 12.5% blocks. Interestingly for two sessions, JG14_190619 and JG14_190619, there was no significant difference for the 75% probability block between the safe-side and gamble-side. This describes an expected behavior from a rational agent, since the reward maximization results inside choice depending on the probability block.

The two sessions, where the chosen side for the high 75% probability block was not depending on the chosen side, might be explainable by extensive probing behavior by the animal. As mentioned, there are no external cues guiding the decision of the animal, thus the above indicated, regular probing is vital for overall reward maximization. The animal has to accept a short term loss to gain a long term maximum. The animal also has to weigh rewarded gamble-arm trials with not rewarded gamble-arm trials and track this ratio to guide its decision. This probing behavior can be observed at the recording session JG14_190619, where, at the beginning, the animal clearly samples both sides multiple times despite the not rewarded trials.

The complexity for the animal is further increased by the fact that it never sees the true distribution, but only a sample from the ground truth. If the sample is too small it can heavily distort the distribution, but if the animal takes a larger sample size it risks decreasing the overall reward.

6.2.2 9.1.2. Neural Findings

The results from spike sorting, and the resulting numbers of good clusters is, what would be expected from the used number of recording sites. The observed distribution of firing rate across all neurons is expected to be heavily skewed to the lower end of the spectrum (Doya, 2011). Since it is also more energy-efficient for a neuron to saturate lower frequencies first. The observed distribution for the inter-spike-interval, further strengthens the notion, that the spike sorted clusters actually represent putative physical neurons.

All indicators about the clusters point towards them actually representing real individual neurons in the PFC and therefore building a sound source for correlating their activity with the behavior of the animal during the experiment to better understand how the neural computation mechanisms work.

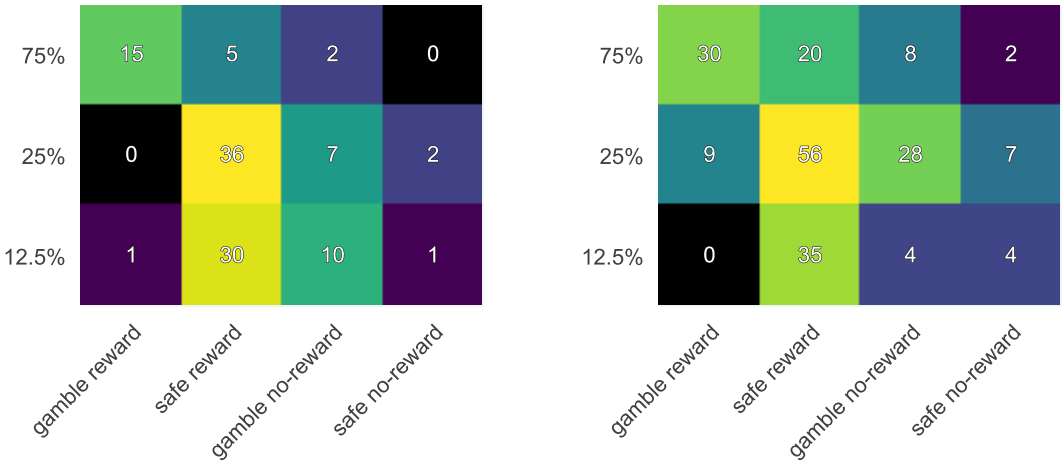
What could be observed with other sessions, during spike sorting, was that a small number of electrode recording channels had very strong noise, which resulted in a low accuracy in spike sorting and a significant increase in MUA clusters. A potential problem with these recordings, could be, that these electrode recording channels had hardware issues, and would have to be filtered out for spike sorting. Although this was not encountered with the two sorted sessions, the recording hardware was the same as used with the sessions that encountered the channel defects. Potential defect channels also present in the sorted sessions, can lead to a significant decrease in signal to noise ratio, which would obscure a lot of the potential observed correlations. Further analysis has to be conducted to exclude this problem with the recording hardware.

The combination of exploratory data analysis and comparison of reward aligned spike rate with the distribution obtained from random samples by the bootstrapping approach lead to a sound identification of significant neurons. Neurons identified visually from the PSTH plots are also identified by the algorithm and vice versa. The only problem present with both

approaches is that for some subselections of trials e.g. gamble-side block 12.5% and reward only very few or no trials fall into that category, thus it is not possible to observe the behavior of neurons ins such a condition.

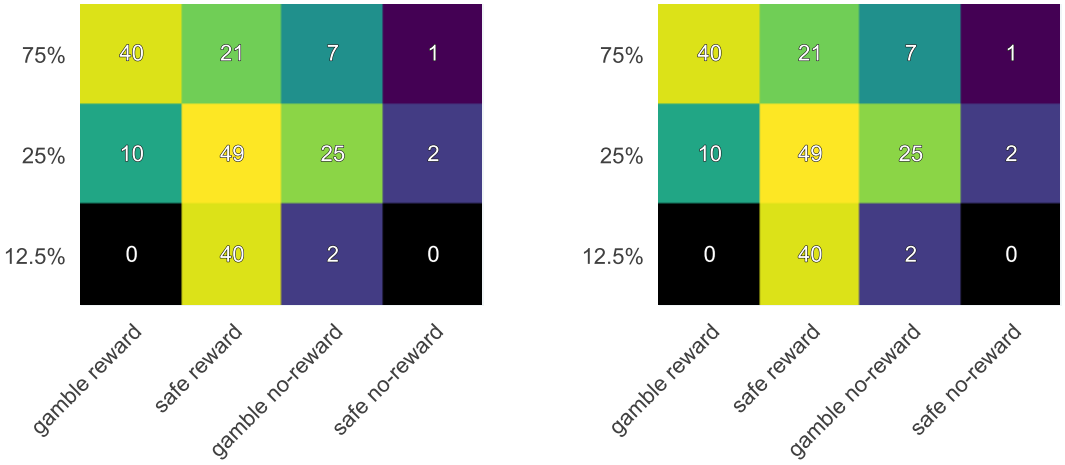
The results from the statistical data analysis for the number of responding neurons during gamble-side choices comparing reward with no-reward outcomes based on the neural fingerprints comparing the 75% high probability block and the 12.5% low probability block show a difference, for both sessions. The number of significant neurons, both responding before, across, and after the event time gradually decreases for reward outcome from block 75% to 25% and 12.5%. In contrast, the number of significant neurons observed during no-reward trials increases for session JG14_190621, they do not continuously increase for session JG18b_190828.

This behavior seems to be directly related to the number of occurrences of the specific type of trial since for session JG14_190621 the number of trials for gamble-side reward decrease from block 75% to 25% and 12.5% and increase for no-reward trials. For session JG18b_190828, where not a clear increase of the number of significantly responding neurons for no-reward gamble-side for the 25% block to the 12.5% probability block is observed, the number of trials matching this condition is also not increasing but decreasing between block 25% and 12.5%. Compared to the session JG14_190621, both the number of significant neurons responding to gamble-side no-reward and trials matching this condition increases between block 25% and 12.5%, see figure 6.1 and figure 6.2. This potential influence will need further investigation and statistical analysis in the future. Such an influence will likely also have to be regarded with future design iterations of the behavior task.



(a) *Distribution of significant neurons* **(b)** *Distribution of trials*

Figure 6.1: Comparison of distribution of significant neurons for specific combinations of gamble-side reward and no-reward to distribution of occurrence of trials for this subselections for session JG14_190621



(a) *Distribution of significant neurons* **(b)** *Distribution of trials*

Figure 6.2: Comparison of distribution of significant neurons for specific combinations of gamble-side reward and no-reward to distribution of occurrence of trials for this subselections for session JG18b_190828

A significant difference in the amount of responding neurons, depending on the probability block would be expected, if the observed neurons are coding for reward and not reward. Although a direct comparison of 75% and 12.5% leads to a not conclusive result, a comparison including all trials, results in both sessions being not-independent. But a comparison between all three blocks results in significantly dependent numbers of responding neurons to the side and block. This would be expected if the neurons are responding differently to reward and no-reward occurrence.

A potential problem in regard to both 75% block and 12.5% block is that total trials where both conditions are met can be very low, thus the sample size very small, and the impact of the randomly chosen sample much higher. The distribution from the small sample size does not necessarily represent the actual distribution. This probably leads to a distortion of the statistical test. The analysis pipeline has to be further expanded to take this into account and correct for it.

6.2.3 Problems of the Phenosys System

The two sessions, significantly different for neurons, had a significant correlation with the reward no-reward event in whether the neurons increased theri spike rate before the event or after. Most of the responding neurons in session JG14_190621 responded before the event, suggesting a reward predicting capability, as with session JG18b_190828 a significantly larger amount of neurons increased their firing rate after or at the event, which would not be how reward predicting neurons behave. As it is very unlikely that the observed neurons in the two sessions, for the same behavior task, respond with such a significantly different behavior, a

potential issue could stem from the synchronization of behavior events between the Phenosys system and the Intan electrophysiology acquisition board. This is of grave importance, if this issue can not be resolved or otherwise proven to be not the cause, all recorded data with the Phenosys system, and more than two years of work could potentially not be usable.

For the new setup, special focus has to be placed on absolute correct synchronization. Further test protocols have to be developed, in addition to the initial testing performed.

6.3 Interpretation New VR Setup

The new system could already be tested over the duration of three weeks by training a fresh mouse.

6.3.1 Performance

Different training stages (the same as used for the Phenosys system section 2.8.2) in terms of reward probabilities for the gamble-side have been implemented by manually adapting the respective probabilities and number of blocks from the pre-session window figure 5.10a . The duration of training stages for an animal completely trained on the new system are similar to the old Phenosys system. The performance of animals in the last stage also is very desirable and closely resembles the performance expected from animals trained on the Phenosys system. Examples of a training sessions in the orientation stage are shown in figure 6.3.

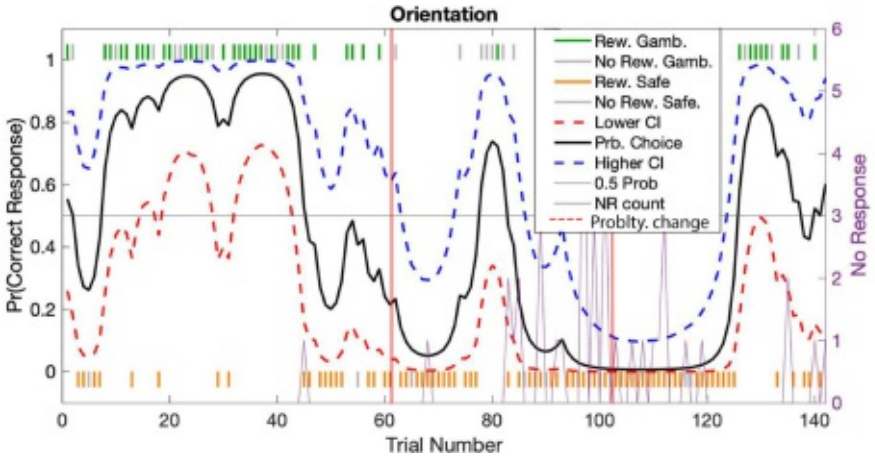


Figure 6.3: Behavior of examplary training session with the new system, phase: orientation, reward probabilites: block 1 - 75% (trial 0-60), block 2 - 12.5% (trial 61)-103), block 3 - 75% (trial 104-end)

The animal clearly shows adaption to changing reward probability for the gamble-side, while still regularly probing to detect continuous changes. Such behavior is desired and essential for the gamble-task, to observe it also with the new system, counts towards the reliability and function of the new system. Also the no-response trials are as expected with the orientation phase.

During the initial usage, the reward system had to be slightly adapted by Hugo Malagon. The needle used between the end of the reward tube and the animals mount was replaced with a rounded syringe tip. Overall the system performed as expected.

6.3.2 Data Standards and Integration

The customization of the database has yet to fully take shape. The first period of use, will likely result in a lot of changes and adaptations of the custom data fields. Also, only a very rudimentary data analysis pipeline (the one used for the first part of this thesis) is integrated, which will be likely to grow over time.

Also a sound backup procedure for all the data both on the respective computers for training and recording system, but mostly for the database and data storage used by the database has to be implemented.

6.3.3 Future Expandability and Adaptations

The modular approach of the hardware and software for the new gamble-task setup allows for a wide range of future expansions. Integrating a TTL controlled laser for optogenetic stimulation is definitely one of the most interesting options. This can easily be accomplished via additional output-modules connected to the Bpod state machine. Via additional states or output-actions, the laser can be precisely switched on and off, timed with behavior task states. For a more complex control of the laser via arbitrary pulses the PulsPal also developed by Sanworks can be used. The advantage of using the PulsPal is that it integrates out of the box with the Bpod and the PyBpod API, thus allowing for quick and easy deployment.

Integration of visual computing also provides a huge opportunity to expand the capabilities of the gamble-task setup. At the core of machine learning-based visual computing for behavior setups are camera images. To synchronize the frame acquisition of these camera TTL pulses can be used. Again integrating and controlling such cameras is relatively straightforward via state output actions with the new setup.

A future optimization, currently in development for the gamble-task, is to change the gravity and valve-based system with a precision micro pump. The used gravity-based system always has the issue of changing the deposited amount, since the pressure of the water column decreases during a trial since the water column decreases. This can be mitigated by using a relatively large vessel with a large diameter causing a neglectable decline in the height of the water column over the period of a trial, but still, the exact same height has to be filled in every time, thus introducing inevitable systematic errors. A system based on a pump, which arguably is more complex, has several advantages that increase consistency and accuracy of the delivered reward.

A potential addition to the existing database, would be to import previously recorded data with the Phenosys system to the Alyx database. This would be possible with a short python script, and could result in much higher accessibility of these datasets.

Independent of what challenges the next scientific questions might bring and what hardware and software tools will need to be integrated the gamble-task platform, based on the Bpod framework, will provide a sound basis for years to come.

Bibliography

8

1. *8 Channel Output Module Bpod*, Sanworks (2021-02-02; https://sanworks.io/shop/viewproduct?productID=1018) (cit. on pp. 5, 83).
2. *A Free and Open Platform for Sharing MRI, MEG, EEG, iEEG, and ECoG Data - OpenNeuro* (2021-01-23; https://openneuro.org/) (cit. on p. 106).
3. *A Standardized and Reproducible Method to Measure Decision-Making in Mice: CAD Drawings for Items Linked in Appendices.Zip*, figshare (2021-01-23; /articles/online_resource/A_standardized_and_reproducible_method_to_measure_decision-making_in_mice_CAD_Drawings_for_items_linked_in_appendices_zip/11638362/2) (cit. on p. 96).
4. L. F. Abbott *et al.*, *Neuron* **96**, 1213–1218, ISSN: 08966273, (2020-04-10; https://linkinghub.elsevier.com/retrieve/pii/S0896627317311364) (2017-12) (cit. on pp. 27, 32).
5. W. C. Abraham, O. D. Jones, D. L. Glanzman, *npj Science of Learning* **4**, 1–10, ISSN: 2056-7936, (2021-01-25; https://www.nature.com/articles/s41539-019-0048-y) (1 2019-07-02) (cit. on p. 15).
6. E. D. Adrian, Y. Zotterman, *The Journal of Physiology* **61**, 465–483, ISSN: 0022-3751, pmid: 16993807, (2021-01-21; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1514868/) (1926-08-06) (cit. on p. 16).
7. *Bcontrol* (2021-01-12; https://brodylabwiki.princeton.edu/bcontrol/index.php/Main_Page) (cit. on p. 77).
8. M. F. Bear, B. W. Connors, M. A. Paradiso, *Neuroscience: Exploring the Brain Fourth, North Americ Edition by Bear PhD, Mark F., Connors PhD, Barry W., Paradiso PhD, Mich (2015) Hardcover* (Jones & Bartlett Learning, Philadelphia, 4th Edition, 2015-02-17), 975 pp., ISBN: 978-0-7817-7817-6 (cit. on pp. 9–16).
9. A. Bechara, A. R. Damasio, H. Damasio, S. W. Anderson, *Cognition* **50**, 7–15, ISSN: 0010-0277, pmid: 8039375 (1944-04) (cit. on p. 33).
10. E. A. Berg, *The Journal of General Psychology* **39**, 15–22, ISSN: 0022-1309, pmid: 18889466, (2021-01-26; https://doi.org/10.1080/00221309.1948.9918159) (1948-07-01) (cit. on p. 12).
11. J. Bhagat *et al.*, "Rigbox: An Open-Source Toolbox for Probing Neurons and Behavior", preprint (Neuroscience, 2019-06-15), (2020-04-10; http://biorxiv.org/lookup/doi/10.1101/672204) (cit. on p. 77).
12. D. Brevers, A. Bechara, A. Cleeremans, X. Noel, *Frontiers in Psychology* **4**, ISSN: 1664-1078, (2021-01-26; https://www.frontiersin.org/articles/10.3389/fpsyg.2013.00665/full) (2013) (cit. on p. 33).

13. C. P. Burgess *et al.*, *Cell Reports* **20**, 2513–2524, ISSN: 22111247, (2020-04-10; https://linkinghub.elsevier.com/retrieve/pii/S2211124717311725) (2017-09) (cit. on p. 36).
14. C. J. Burke, P. N. Tobler, *Frontiers in Neuroscience* **5**, ISSN: 1662-453X, (2020-12-05; https://www.frontiersin.org/articles/10.3389/fnins.2011.00121/full) (2011) (cit. on p. 28).
15. G. Buzsáki, *Nature Neuroscience* **7**, 446–451, ISSN: 1097-6256, 1546-1726, (2020-12-02; http://www.nature.com/articles/nn1233) (2004-05) (cit. on p. 21).
16. M. Carandini, A. K. Churchland, *Nature Neuroscience* **16**, 824–831, ISSN: 1546-1726, (2020-04-10; https://www.nature.com/articles/nn.3410) (7 2013-07) (cit. on p. 36).
17. *CED Spike2: Spike Sorting* (2021-01-05; http://ced.co.uk/products/spkssss) (cit. on p. 48).
18. A. Chen, *Animal Model Diversity in Neuroscience*, Harvard Neuro Blog (2020-12-08; https://www.harvardneuroblog.com/blog/2019/3/12/animal-model-diversity-in-neuroscience) (cit. on p. 30).
19. A. Citri, R. C. Malenka, *Neuropsychopharmacology* **33**, 18–41, ISSN: 1740-634X, (2020-11-23; https://www.nature.com/articles/1301559) (1 2008-01) (cit. on p. 16).
20. R. Cools *et al.*, *Neuropsychopharmacology* **30**, 1362–1373, ISSN: 1740-634X, (2021-01-21; https://www.nature.com/articles/1300704) (7 2005-07) (cit. on p. 28).
21. *Cortex-Lab/Alyx*, The Cortical Processing Laboratory at UCL, 2021-01-14, (2021-01-24; https://github.com/cortex-lab/alyx) (cit. on pp. 108, 109).
22. *Cortex-Lab/Rigbox*, The Cortical Processing Laboratory at UCL, 2021-01-05, (2021-01-11; https://github.com/cortex-lab/Rigbox) (cit. on pp. 4, 77).
23. E. Covey, M. Carter, Eds., *Basic Electrophysiological Methods* (Oxford University Press, Oxford, New York, 2015-04-16), 240 pp., ISBN: 978-0-19-993980-0 (cit. on pp. 18, 19).
24. K. Doya, Ed., *Bayesian Brain: Probabilistic Approaches to Neural Coding* (MIT Press, Cambridge, Mass., 1. MIT Press paperback ed, 2011), 326 pp., ISBN: 978-0-262-51601-3 (cit. on pp. 25, 28, 29, 114).
25. D. Durstewitz, N. M. Vittoz, S. B. Floresco, J. K. Seamans, *Neuron* **66**, 438–448, ISSN: 08966273, (2020-04-10; https://linkinghub.elsevier.com/retrieve/pii/S0896627310002321) (2010-05) (cit. on pp. 16, 27).
26. N. Eichert *et al.*, *eLife* **9**, ed. by T. Verstynen, J. I. Gold, T. Verstynen, K. Heuer, e53232, ISSN: 2050-084X, (2021-01-25; https://doi.org/10.7554/eLife.53232) (2020-03-23) (cit. on p. 9).
27. B. Ellenbroek, J. Youn, *Disease Models & Mechanisms* **9**, 1079–1087, ISSN: 1754-8403, 1754-8411, pmid: 27736744, (2020-12-08; https://dmm.biologists.org/content/9/10/1079) (2016-10-01) (cit. on p. 30).
28. *Enginursday: Open Ephys - News - SparkFun Electronics* (2021-01-28; https://www.sparkfun.com/news/1541) (cit. on p. 104).
29. L. Grosenick, J. H. Marshel, K. Deisseroth, *Neuron* **86**, 106, pmid: 25856490, (2021-01-25; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4775736/) (2015-04-08) (cit. on p. 18).
30. Hippocrates, *On the Sacred Disease* (Library of Alexandria, 2007), 29 pp., ISBN: 978-1-4655-2804-9 (cit. on p. 7).
31. M. A. Hofman, *Frontiers in Neuroanatomy* **8**, ISSN: 1662-5129, (2020-11-15; https://www.frontiersin.org/articles/10.3389/fnana.2014.00015/full) (2014) (cit. on p. 9).
32. C. Hölscher, A. Schnee, H. Dahmen, L. Setia, H. A. Mallot, *Journal of Experimental Biology* **208**, 561–569, ISSN: 0022-0949, 1477-9145, pmid: 15671344, (2020-12-08; https://jeb.biologists.org/content/208/3/561) (2005-02-01) (cit. on p. 31).

33. *Home*, Neuropixels (2020-12-04; https://www.neuropixels.org/) (cit. on pp. 21, 22).
34. *Intan Technologies*, Intan Technologies (2021-01-25; http://www.intantech.com) (cit. on p. 23).
35. *Intan Technologies*, Intan Technologies (2020-12-16; http://www.intantech.com) (cit. on p. 41).
36. *International Brain Laboratory* (2021-01-11; https://www.internationalbrainlab.com/) (cit. on pp. 78, 106).
37. *Introduction - Phy* (2021-01-05; https://phy.readthedocs.io/en/latest/) (cit. on p. 49).
38. *IST Austria | Miba Machine Shop* (2021-01-23; https://ist.ac.at/en/research/scientific-service-units/machine-shop/) (cit. on pp. 5, 101).
39. J. J. Jun, C. Mitelut, *et al.*, "Real-Time Spike Sorting Platform for High-Density Extracellular Probes with Ground-Truth Validation and Drift Correction", preprint (Neuroscience, 2017-01-19), (2020-04-10; http://biorxiv.org/lookup/doi/10.1101/101030) (cit. on pp. 23, 24).
40. J. J. Jun, N. A. Steinmetz, *et al.*, *Nature* **551**, 232–236, ISSN: 1476-4687, (2020-12-04; https://www.nature.com/articles/nature24636) (7679 2017-11) (cit. on pp. 18, 22).
41. J. W. Kable, P. W. Glimcher, *Nature neuroscience* **10**, 1625–1633, ISSN: 1097-6256, pmid: 17982449, (2021-02-08; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2845395/) (2007-12) (cit. on p. 28).
42. E. R. Kandel *et al.*, *Principles of Neural Science, Fifth Edition* (McGraw Hill Professional, 2013), 1761 pp., ISBN: 978-0-07-139011-8 (cit. on p. 8).
43. M. P. Karlsson, D. G. R. Tervo, A. Y. Karpova, *Science* **338**, 135–139, ISSN: 0036-8075, 1095-9203, pmid: 23042898, (2020-04-10; https://science.sciencemag.org/content/338/6103/135) (2012-10-05) (cit. on p. 27).
44. M. Kay, *Information Theory, Inference and Learning Algorithms - David J. C. MacKay, David J. C. Mac Kay - Google Books* (2021-01-21; https://books.google.at/books/about/Information_Theory_Inference_and_Learnin.html?id=AKuMj4PN_EMC&redir_esc=y) (cit. on p. 16).
45. T. Klausberger, P. Somogyi, *Science* **321**, 53–57, ISSN: 0036-8075, 1095-9203, pmid: 18599766, (2021-01-25; https://science.sciencemag.org/content/321/5885/53) (2008-07-04) (cit. on p. 20).
46. X. Li, Ed., *Signal Processing in Neuroscience* (Springer Singapore, Singapore, 2016), ISBN: 978-981-10-1821-3 978-981-10-1822-0, (2021-02-11; http://link.springer.com/10.1007/978-981-10-1822-0) (cit. on pp. 18–20).
47. D. Lieberman, *The Story of the Human Body: Evolution, Health and Disease* (Penguin UK, 2013-10-03), 507 pp., ISBN: 978-1-84614-393-9, Google Books: 7bdqAAAAQBAJ (cit. on p. 9).
48. L. Luan *et al.*, *Science Advances* **3**, e1601966, ISSN: 2375-2548, (2020-12-01; https://advances.sciencemag.org/content/3/2/e1601966) (2017-02-01) (cit. on p. 18).
49. H. Malagon-Vina, S. Ciocchi, J. Passecker, G. Dorffner, T. Klausberger, *Nature Communications* **9**, 1–13, ISSN: 2041-1723, (2020-04-10; https://www.nature.com/articles/s41467-017-02764-x) (1 2018-01-22) (cit. on pp. 16, 17, 36).
50. S. Manita, H. Miyakawa, K. Kitamura, M. Murayama, *Frontiers in Cellular Neuroscience* **11**, ISSN: 1662-5102, (2020-11-19; https://www.frontiersin.org/articles/10.3389/fncel.2017.00029/full) (2017) (cit. on p. 13).
51. *Micro Manipulators Models 1760, 1760-61 | Kopf Instruments* (2021-01-23; http://kopfinstruments.com/product/micro-manipulators-models-1760-1760-61/) (cit. on p. 100).
52. G. A. Miller, *Trends in Cognitive Sciences* **7**, 141–144, ISSN: 13646613, (2020-11-13; https://linkinghub.elsevier.com/retrieve/pii/S1364661303000299) (2003-03) (cit. on p. 8).
53. *Model 1430 Stereotaxic Frame | Kopf Instruments* (2021-01-23; http://kopfinstruments.com/product/model-1430-stereotaxic-frame/) (cit. on p. 100).

54. E. I. Moser, E. Kropff, M.-B. Moser, *Annual Review of Neuroscience* **31**, 69–89, ISSN: 0147-006X, pmid: 18284371 (2008) (cit. on p. 18).

55. E. Musk, Neurolink, *bioRxiv*, 703801, (2019-12-18; https://www.biorxiv.org/content/10.1101/703801v1) (2019-07-17) (cit. on pp. 21, 22).

56. J. O'Keefe, *Experimental Neurology* **51**, 78–109, ISSN: 0014-4886, (2021-03-11; https://www.sciencedirect.com/science/article/pii/0014488676900558) (1976-01-01) (cit. on p. 18).

57. F. Office, *A Short History of European Neuroscience - from the Late 18th to the Mid 20th Century*, FENS.org (2021-02-08; https://www.fens.org/News-Activities/News/2014/021/A-Short-History-of-European-Neuroscience---from-the-late-18th-to-the-mid-20th-century/) (cit. on p. 18).

58. *OpenEphys Acquisition Board*, Open Ephys (2021-01-28; https://open-ephys.org/acquisition-system/eux9baf6a5s8tid06hk1mw5aafjdz1) (cit. on p. 104).

59. M. Pachitariu, N. Steinmetz, S. Kadir, M. Carandini, H. Kenneth D., "Kilosort: Realtime Spike-Sorting for Extracellular Electrophysiology with Hundreds of Channels", preprint (Neuroscience, 2016-06-30), (2020-04-10; http://biorxiv.org/lookup/doi/10.1101/061481) (cit. on pp. 23, 24, 48).

60. C. Padoa-Schioppa, K. E. Conen, *Neuron* **96**, 736–754, ISSN: 08966273, (2020-04-10; https://linkinghub.elsevier.com/retrieve/pii/S0896627317309005) (2017-11) (cit. on pp. 12, 25, 27).

61. S. Panzeri, J. H. Macke, J. Gross, C. Kayser, *Trends in Cognitive Sciences* **19**, 162–172, ISSN: 1364-6613, pmid: 25670005, (2020-11-24; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4379382/) (2015-03) (cit. on p. 17).

62. J. Passecker *et al.*, *Neuron* **101**, 152–164.e7, ISSN: 08966273, (2020-04-10; https://linkinghub.elsevier.com/retrieve/pii/S0896627318309589) (2019-01) (cit. on pp. 1, 16–18, 26–28, 32–34, 37, 38, 66, 70).

63. A. E. Pereda, *Nature Reviews Neuroscience* **15**, 250–263, ISSN: 1471-0048, (2021-01-25; https://www.nature.com/articles/nrn3708) (4 2014-04) (cit. on p. 11).

64. *PhenoSys JetBall-TFT Virtual Reality System for Rodents*, PhenoSys (2021-01-22; https://www.phenosys.com/products/virtual-reality/jetball-tft/) (cit. on pp. 37, 38).

65. *PhenoSys Steering Wheel*, PhenoSys (2021-01-22; https://www.phenosys.com/products/virtual-reality/steering-wheel/) (cit. on p. 38).

66. D. Pinault, *Journal of Neuroscience Methods* **65**, 113–136, ISSN: 0165-0270, pmid: 8740589 (1996-04) (cit. on p. 19).

67. *Ported*, The Lee Company (2021-01-27; https://www.theleeco.com/products/electro-fluidic-systems/solenoid-valves/control-valves/lhd-series/3-port/ported/) (cit. on p. 82).

68. T. M. Press, *Cognitive Search | The MIT Press* (2020-04-27; https://mitpress.mit.edu/books/cognitive-search) (cit. on p. 27).

69. *Psychtoolbox-3 - Overview* (2021-01-11; http://psychtoolbox.org/) (cit. on p. 77).

70. *Pybpod/Pybpod-Gui-Plugin-Alyx*, PyBpod, 2019-11-07, (2021-01-24; https://github.com/pybpod/pybpod-gui-plugin-alyx) (cit. on p. 109).

71. *pyControl* (2021-01-11; https://pycontrol.readthedocs.io/en/latest/) (cit. on p. 78).

72. F. Rattay, *Electrical Nerve Stimulation: Theory, Experiments and Applications* (Springer-Verlag, Wien, 1990), ISBN: 978-3-211-82247-0, (2021-02-21; https://www.springer.com/gp/book/9783211822470) (cit. on pp. 14, 15).

73. F. Rieke, Ed., *Spikes: Exploring the Neural Code* (MIT Press, Cambridge, Mass., 1. paperback ed, 1999), 395 pp., ISBN: 978-0-262-68108-7 978-0-262-18174-7 (cit. on pp. 8, 13, 16, 17).

74. J. S. Rodriguez, M. G. Paule, in *Methods of Behavior Analysis in Neuroscience*, ed. by J. J. Buccafusco (CRC Press/Taylor & Francis, Boca Raton (FL), ed. 2, 2009), ISBN: 978-1-4200-5234-3, pmid: 21204334, (2021-01-26; http://www.ncbi.nlm.nih.gov/books/NBK5227/) (cit. on p. 12).

75. C. Rossant *et al.*, *Nature neuroscience* **19**, 634–641, ISSN: 1097-6256, pmid: 26974951, (2021-01-05; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4817237/) (2016-04) (cit. on p. 48).

76. O. Ruebel *et al.*, "NWB:N 2.0: An Accessible Data Standard for Neurophysiology", preprint (Neuroscience, 2019-01-17), (2020-04-15; http://biorxiv.org/lookup/doi/10.1101/523035) (cit. on p. 106).

77. *Sanworks* (2021-01-11; https://sanworks.io/about/about.php) (cit. on pp. 5, 77).

78. *Sanworks* (2021-02-07; https://sanworks.io/shop/viewproduct?productID=1021) (cit. on p. 83).

79. J. L. Saunders, M. Wehr, *bioRxiv*, 807693, (2021-01-11; https://www.biorxiv.org/content/10.1101/807693v1) (2019-10-17) (cit. on p. 78).

80. E. W. Schomburg, C. A. Anastassiou, G. Buzsáki, C. Koch, *Journal of Neuroscience* **32**, 11798–11811, ISSN: 0270-6474, 1529-2401, pmid: 22915121, (2020-12-02; https://www.jneurosci.org/content/32/34/11798) (2012-08-22) (cit. on p. 20).

81. C. Schwarz *et al.*, *Somatosensory & Motor Research* **27**, 131–148, ISSN: 0899-0220, 1369-1651, (2020-04-30; http://www.tandfonline.com/doi/full/10.3109/08990220.2010.513111) (2010-12) (cit. on pp. 31, 36).

82. *Scientific Software | Champalimaud Foundation* (2021-02-10; https://fchampalimaud.org/platforms/scientific-software) (cit. on pp. 5, 78).

83. *Scientifica IVM Single Motorised Micromanipulator*, Scientifica (2021-01-24; https://www.scientifica.uk.com/products/scientifica-ivm-single) (cit. on p. 104).

84. M. Segal, *Nature Reviews Neuroscience* **6**, 277–284, ISSN: 1471-0048, (2021-02-22; https://www.nature.com/articles/nrn1649) (4 2005-04) (cit. on p. 16).

85. S. Shipp, *Current Biology* **17**, R443–R449, ISSN: 09609822, (2020-11-18; https://linkinghub.elsevier.com/retrieve/pii/S0960982207011487) (2007-06) (cit. on p. 11).

86. J. H. Siegle *et al.*, *Journal of Neural Engineering* **14**, 045003, ISSN: 1741-2552, pmid: 28169219 (2017-08) (cit. on pp. 5, 104).

87. J. B. Smaers, A. Gómez-Robles, A. N. Parks, C. C. Sherwood, *Current biology: CB* **27**, 714–720, ISSN: 1879-0445, pmid: 28162899 (2017-03-06) (cit. on p. 9).

88. *SmartProbe* (2020-12-16; https://neuronexus.com/products/neural-probes/smartprobe/) (cit. on p. 41).

89. A. C. Smith *et al.*, *Journal of Neuroscience* **24**, 447–461, ISSN: 0270-6474, 1529-2401, pmid: 14724243, (2021-02-04; https://www.jneurosci.org/content/24/2/447) (2004-01-14) (cit. on p. 47).

90. N. A. Steinmetz, C. Aydin, *et al.*, *bioRxiv*, 2020.10.27.358291, (2021-01-21; https://www.biorxiv.org/content/10.1101/2020.10.27.358291v1) (2020-10-28) (cit. on pp. 22, 104).

91. N. A. Steinmetz, P. Zatka-Haas, M. Carandini, K. D. Harris, *Nature* **576**, 266–273, ISSN: 1476-4687, (2021-01-21; https://www.nature.com/articles/s41586-019-1787-x) (7786 2019-12) (cit. on p. 21).

92. P. N. Stuart Russel, *Artificial Intelligence: A Modern Approach, 4th Edition* (ed. 3, 2020), (2021-01-21; /content/one-dot-com/one-dot-com/us/en/higher-education/program.html) (cit. on pp. 25–27).

93. J. L. Teeters *et al.*, *Neuron* **88**, 629–634, ISSN: 0896-6273, pmid: 26590340, (2020-04-15; https://www.cell.com/neuron/abstract/S0896-6273(15)00919-8) (2015-11-18) (cit. on pp. 5, 106).

94. The International Brain Laboratory, V. Aguillon-Rodriguez, *et al.*, "A Standardized and Reproducible Method to Measure Decision-Making in Mice", preprint (Neuroscience, 2020-01-17), (2020-04-10; http://biorxiv.org/lookup/doi/10.1101/2020.01.17.909838) (cit. on pp. 5, 95).

95. The International Brain Laboratory, N. Bonacchi, *et al.*, "Data Architecture for a Large-Scale Neuroscience Collaboration", preprint (Neuroscience, 2019-11-02), (2020-04-10; http://biorxiv.org/lookup/doi/10.1101/827873) (cit. on pp. 5, 78).

96. *The NWB:N Data Standard – Neurodata Without Borders* (2021-01-24; https://www.nwb.org/nwb-neurophysiology/) (cit. on p. 107).

97. K. Thurley, A. Ayaz, *Current Zoology* **63**, 109–119, ISSN: 1674-5507, 2396-9814, (2020-04-10; https://academic.oup.com/cz/article-lookup/doi/10.1093/cz/zow070) (2017-02) (cit. on pp. 31, 32).

98. T. Trappenberg, *Fundamentals of Computational Neuroscience* (Oxford University Press, Oxford ; New York, 2. Edition, 2010-01-18), 416 pp., ISBN: 978-0-19-956841-3 (cit. on pp. 13, 14, 17).

99. R. van den Bos, W. Lasthuis, E. den Heijer, J. van der Harst, B. Spruijt, *Behavior Research Methods* **38**, 470–478, ISSN: 1554-3528, (2020-12-09; https://doi.org/10.3758/BF03192801) (2006-08-01) (cit. on p. 33).