Contents

1. Introduction

The tactile sense proved to be beneficial for perception, yet it is an often overlooked modality in the field of robotics (Luo et al, 2017; Wang et al 2020). This oversight in robotic design and functionality is particularly striking given the crucial role of touch in biological organisms (Warren, 2021; Bauer et al. 2018; Polley et al. 2005). In nature, tactile sensing is fundamental for tasks ranging from navigation to object identification, offering a direct and immediate method for interacting with the environment. This is especially true for tactile hair, where only a few implementations in real robots exist (Fox et al. 2012, Salman et al. 2016), yet nearly all mammals have tactile hair, except some apes and humans (Grant et al. 2018).
The effectiveness for navigation can be seen in rats and mice, extensively using their whiskers for spatial awareness and detecting obstacles, showcasing a sophisticated form of tactile sensing that has evolved for survival (Diamond et al., 2008; Harvey et al., 2001; Harris et al., 1999).
Despite the clear advantages observed in nature, the adoption of tactile sensing in robotics for navigation is not as important as other sensors like cameras or LIDAR (Gul et al. 2019). While effective in many scenarios, these traditional sensors can fall short in environments where visibility conditions are poor or (Dai et al. 2022; Yang et al. 2011). The exploration and integration of tactile hair sensors, inspired by the biological use of whiskers in rodents, represent a promising frontier in robotics. This approach not only offers the potential to navigate and understand complex environments where other sensors fail but also opens new paths for robot-environment interaction that are closer to the capabilities found in the natural world (Salman et al. 2016).

The underutilization of tactile sensing in robotics for navigation not only highlights a significant gap in the field but also presents an opportunity to innovate through bioinspired solutions. Drawing inspiration from the sophisticated tactile sensing capabilities observed in rodents, particularly their use of whiskers for spatial awareness and obstacle avoidance, this research seeks to explore and harness the potential of artificial whisker sensors. Such sensors could significantly enhance mobile robotic systems, enabling sophisticated task performance, adaptation to diverse environments, and higher autonomy levels, especially where traditional sensors fail due to limitations in visibility or environmental complexity.

Accordingly, the core research problem this thesis addresses is the development and practical integration of tactile hair sensors into robotic systems, with a focus on a bioinspired

robot mouse model. This model aims to navigate and interact with its environment, effectively using only tactile whiskers as sensory input. It provides a biologically plausible perception while overcoming the challenges present in environments unsuitable for vision or laser-based sensors. By replicating the tactile sensing mechanisms of biological whiskers within a robotic context, this study seeks not only to fill a crucial gap in robotic sensory modalities but also to provide a compelling demonstration of the capabilities and advantages of tactile sensing in real-world robotic applications. This approach underscores the thesis's goal to understand and overcome the technical and conceptual challenges in tactile sensor integration, setting a precedent for future advancements in robotic design and functionality.

The primary purpose of this thesis is the integration of tactile hair sensors into a mouse robot, leveraging the capabilities of artificial whiskers for improved environmental navigation. This study aims:

1. **To design and develop bioinspired whisker sensors**, focusing on replicating the tactile sensing capabilities found in natural rodent whiskers for environmental perception and interaction. They should be cost-effective and space efficient.
2. **To integrate these whisker sensors in a neurorobotic mouse**, to enhance its sensing perception and develop a platform combining bio-inspiration in aspects of anatomy and perception.
3. **To apply Deep Reinforcement Learning for realizing obstacle avoidance behavior**, to enable autonomous navigation of the robot just equipped with sparse whisker information.

These objectives are related to the development of a new concept for robotic sensory systems, aiming to significantly improve the robots' environmental understanding and interaction through the application of bio-inspired tactile sensors and advanced computational techniques.

This thesis explores the integration of bioinspired whisker sensors in neurorobotic platforms, specifically focusing on the neurorobotic mouse, NeRmo, for obstacle avoidance through Deep Reinforcement Learning (DRL). The research delves into the design, development, and application of artificial whisker sensors, drawing inspiration from the tactile sensing mechanisms of rodents. The scope is confined to examining the effectiveness of these sensors in enhancing the robot's navigational capabilities in environments where traditional sensors may be ineffective. Limitations include the environmental complexity the

neurorobotic mouse is tested in, the types of obstacles used for avoidance tasks, and the specific DRL algorithms applied.

The methodology adopted in this thesis is interdisciplinary, combining principles from robotics, sensor technology, and artificial intelligence. Initially, the design and development phase focuses on creating artificial whisker sensors that mimic the tactile sensing capabilities of rodents. These sensors are then integrated into NeRmo, with the setup assessed through mechanical and magnetic simulations to evaluate sensor performance. Subsequently, DRL algorithms are implemented and tuned to process the tactile data from the whisker sensors for obstacle detection and avoidance. The effectiveness of this integrated system is tested in simulated environments and real-world scenarios to validate the proposed approach.

This thesis contributes to the field of neurorobotics by using bioinspired whisker sensors for tactile sensing in bio-inspired robotic applications. It addresses the gap in leveraging tactile data for navigation, particularly in scenarios where visual or other common sensors fall short. The research highlights the potential of combining biological inspiration with advanced computational techniques, such as DRL, to enhance robotic autonomy and adaptability, even with spatially limited tactile sensors. The findings may offer insights into developing more sophisticated robotic systems having biologically plausible perception.

This thesis is structured to guide the reader through the comprehensive study of bioinspired whisker sensors for neurorobotic applications, with a focus on leveraging DRL for obstacle avoidance. The organization is as follows:

Literature Review: Examines the state of the art in tactile sensing technologies in robotics, with an emphasis on the biological inspiration derived from rodents. This includes discussions on the mechanics of tactile sensing, computational models, and the role of DRL in enhancing robotic navigation and interaction.

Methods and Implementation: Details the interdisciplinary methodology employed to develop the whisker sensors and their integration into the neurorobotic rat, NerMo. It covers the design, simulation, and testing processes, as well as the challenges encountered and the solutions adopted.

Experimental Results and Evaluation: Presents the outcomes of the whisker sensor development and testing, both in simulation and real-world scenarios. It analyzes the

performance of the neurorobotic rat in navigating obstacles using DRL, based on the tactile data provided by the whisker sensors.

Discussion: Reflects on the implications of the findings and sets it into context. Additionally it explores its limitations, and proposes further modifications to improve capabilities of the whisker system.

Conclusion: Summarizes the conducted study and gives suggestions for future research directions.

This structure is designed to flow logically from introducing the research topic to exploring its theoretical background, describing the methodological approach, presenting and analyzing the findings, and concluding with the study's broader implications.

2. Literature Review

This chapter presents a comprehensive literature review, focusing on whiskers as tactile sensors, exploring the intersection of biology, mechanical engineering, and computational intelligence. Initially, it offers a glance into biology, where, in the case of rodents, their whiskers serve as a sophisticated tool for exploration and navigation, serving as inspiration for robots. Subsequently, attention shifts to the mechanics underlying these biological sensors and the computational models designed to replicate their functionality. Finally, this subchapter concludes by exploring existing approaches to bio-inspired tactile hair sensors.

The attention then is drawn to the innovative field of Deep Reinforcement Learning, starting with its foundational elements Reinforcement Learning and Artificial Neural Networks. By outlining the principles of traditional Reinforcement Learning, it illustrates how learning from experience is modeled within a technical framework. Given their crucial role in DRL, Artificial Neural Networks are explored, providing the computational architecture necessary for pattern recognition and decision-making processes. The combination of RL's adaptive learning strategies with the ANNs' complex pattern identification capabilities leads to a thorough examination of specific algorithms and implementation techniques to leverage performance.

2.1 Whiskers as Tactile Hairs

Whiskers in this chapter are first investigated in the context of biology to understand their functionality. Then existing approaches are examined that aim on modeling the mechanics of these tactile hair. Furthermore, the existing technical implementations of artificial whiskers are reviewed.

2.1.1 In Biology

Rodents have sensory hairs at multiple locations on their body but the most important ones are located at their snout. These are called sensory hairs, tactile hairs, sinus hairs, whisker or mystacial vibrissae [Schmidt et al 2014]. Since the latter two terms specifically describe the area around the snout, which is most important for navigation, only these terms are referred to in this paper. For sake of simplicity, the terms vibrissae or whiskers are used to refer to mystical vibrissae, as these are the focus of this investigation.

The so-called follicle anchors the whisker hair in the skin and provides it with sensitivity and movability. Each one contains around 200 trigeminal ganglion cells, whose nerve endings convert the energy of mechanical deflection into action potentials (Diamond et al. 2008).
The mystacial vibrissae in rats are aligned as a two dimensional grid at each side of the face (Figure 2.1) and are identical in all rats in this regard. Each whisker can be assigned to a row (A-E) and column (1-9) so there is a consistent description possible across research groups and rats.
Vibrissae signals are further processed in somatosensory cortex's barrels, which mirror the snout's whisker layout, suggesting a direct 'whisker-to-barrel' connection. Neurons in the barrel cortex, influenced most by a single, topographically corresponding whisker, have receptive fields extending to multiple whiskers. This means surrounding whisker excitations modulate each other, which leads to a smooth spatial map and therefore higher accuracy.

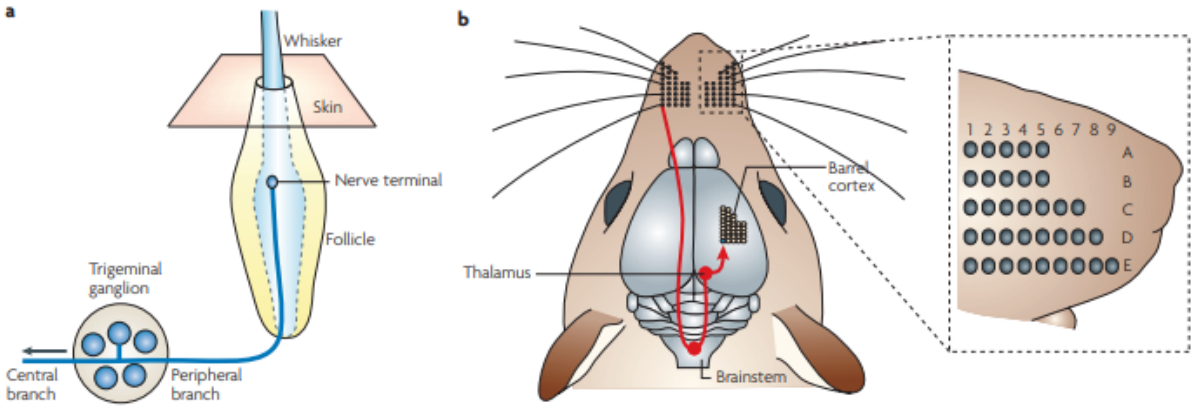

Figure 2.1: Whisker sensory pathway layout: a). Mechanoreceptors in each whisker follicle detect follicle rotation or whisker shaft deflection, providing information on direction, velocity, and duration of movements. These receptors connect to the infra-orbital branch of the trigeminal nerve, leading to the trigeminal ganglion outside the brainstem. The ganglion's central branch extends to the thalamus' trigeminal complex. The vibrissae, arranged in a two-dimensional grid on the snout, consist of rows with 5-9 whiskers each. Neuronal signals from the brainstem cross to the thalamic somatosensory nuclei and then to the primary somatosensory cortex's barrels. Adapted from Diamond et al. 2008.

For active sensing rodents exhibit behavior called whisking. This is a rhythmic movement of the vibrissae, moving them forth and back at an average frequency of about 8 Hz in rats (Adibi 2019). Here, two different patterns can be observed:. The first one is related to exploratory whisking, where there are high amplitude sweeps with a frequency of 1-5 Hz in bouts between 1 and 10 s. Within these bouts, the frequency remains constant, but it can

change between bouts. The other pattern exhibits a narrow-angle high frequency sweep (15 to 25 Hz) for a period of 0.5 and 1 s. This is referred to as foveal whisking because it is similar to the concentrated arrangement of photoreceptors found in the fovea of the retina. The whiskers are moved by two sets of muscles, intrinsic and extrinsic. Intrinsic muscles correspond to individual whiskers and move them forward by pulling the follicle base backwards. Extrinsic muscles are not directly connected to the follicle, they lay in the mystacial pad and move the whole set of vibrissae. The whiskers on both sides can move asymmetrically and asynchronously.

In certain respects, the tactile system bears similarities with the visual system. In humans “saccades”, controlled eye movements, increase efficiency of vision by browsing the environment to extract important information. Rodents exhibit similar behavior with whisking behavior - they sweep their vibrissae and scan the environment, gathering not only information on contact for object localization but have the ability to perceive textures and vibrations. This can be illustrated by human behavior when vision is absent. Humans also use various techniques to feel and explore things with their hands, which is described under the term "haptic exploration." (Lederman and Klatzky, 1987). If one wakes up in the night by a vibrating phone and no light is available there is the need for tactile guidance until the vibration, texture and shape of the object allows the operator to distinguish it as a phone.

Vibrissae are an essential organ for rodents. With the precise control of these tactile hairs they are able to perform various sensory tasks, including wall following (Jenks et al., 2010), estimating distances (Harris et al., 1999) and identifying characteristics like texture (Diamond et al., 2008), as well as shape and size (Harvey et al., 2001).

They are able to localize objects with millimeter precision (Knutsen et al., 2006) and classify them by the combination of shape, texture, roughness, orientation and size (Carvell & Simons, 1990; Grant, Mitchinson, Fox, & Prescott, 2009; Guic-Robles, Valdivieso, & Guajardo, 1989; Krupa, Matell, Brisben, Oliveira, & Nicolelis, 2001; Pammer et al., 2013; Polley, Rickert, & Frostig, 2005).

There are various hypotheses about how texture is perceived by rats, with the slip-stick theory as the most accepted one (Diamond, 2010; Diamond and Arabzadeh, 2013). Here, it is stated that the whisker trajectory is defined by an irregular sticking and slipping of the vibrissa on a textured surface. The discrimination is then achieved by associating these characteristic movements, with an increased rate of slip-stick events signaling for a coarser surface.

One crucial aspect of the whisker system is its role in motion planning, particularly in the absence of visual cues. Researchers placed rodents on a treadmill to observe movements of

the limbs under varying sensory conditions. This experiment was particularly insightful as it manipulated the available sensory information by alternating between periods of complete darkness and enlightened obstacles, and by trimming the animals' whiskers. The study revealed that rodents rely heavily on their whiskers for detecting obstacles and adjusting their movements accordingly, even at high speeds. When whiskers were trimmed, the rodents' ability to adjust their paw height to match the height of the obstacles was significantly impaired. This resulted in a deterioration of the accuracy of their paw landing positions, underscoring the importance of whiskers in precise spatial navigation and obstacle avoidance. Furthermore, the experiment demonstrated that while rodents have access to visual information, their reliance on whisker input was crucial for anticipatory adjustments such as reducing running speed upon encountering an obstacle. This indicates that whiskers provide high-fidelity information crucial for real-time decision-making in dynamic environments (Warren et al 2021).

2.2.2 Mechanics and Models

Due to the pure mechanical nature of signals obtained in the whisker and then transduced to the base, there have been several studies about the mechanical properties of the vibrissae and how they interact with collisions.

The follicle, where the mechanical signal is translated into neuron excitation by mechanoreceptors, has a complex shape with varying structures within it. (Rice et al., 1993). This shape seems to understand more than one mechanical variable of the whisker (Mitchinson et al., 2004).

The shape has an important influence on the mechanical properties, hence a deep examination is crucial to understand the dynamics of the vibrissae system. The whisker is basically a linear truncated cone with tapered trip and a circular cross section with a hollow medulla at the base (Quist et al. 2011). Fully grown animals have whiskers between 10 and 50 mm long (Haidarliu and Ahissar, 2001), depending on location on the mystical pad. Shortest are at the front near the mouth. Diameters vary between 20 and 50 μm at the base and decrease to about 5 μm at the tip, as depicted in Figure 2.2 (Pearson et al., 2007). The vibrissae have intrinsic curvature fitting an quadratic function with increasing curvature towards the tip. The vibrissa has three different layers: The internal one is called medulla, which is mostly hollow, followed by the middle one, the cortex, and the external layer named cuticle. The medulla narrows progressively from base to tip and fades in the distal half.

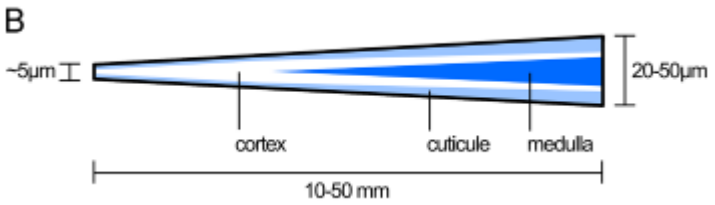

Figure 2.2: Scheme of internal whisker structure with geometrical dimensions. Consists out of three layers, medula (inner), cortex (middle) and cuticle (outer) with a length of 10-50 mm and a varying diameter of 20-50 µm at the base to 5 µm at the tip. Adopted from Pearson et al. 2007.

The structure of the whisker is therefore inhomogenous and has different elastic properties. Along its length, the average modulus is about 4 GPa at the base and decreases to about 3 GPa (Quist et al., 2011), while also the different layers radially exhibit different Moduli (Adineh et al. 2015).

When modeling the whisker, a distinction can be made between two different approaches: The first one are rigid body models (RBM), in which the components of the bodies are not deformable, but the whole mechanical system consists of different parts connected at nodes with spring-damper properties. Here the deformation can be measured by looking at the relative angles of the different bodies. The other approach is continuum modeling (CM), in which every infinitesimal part of the continuum is evaluated.

One representative example of a RBM was introduced by Quist et al. in 2014 and can be seen in Figure 2.3 They developed a vibrissa model designed to simulate the dynamic behavior of vibrissae. The model included important dynamics like inertia, damping and collision. The vibrissae were represented as 13 rigid 2D links with a linear Young's modulus and uniform density. Discrete Lagrange mechanics modeled their behavior, including torsional springs and dampers at link joints, with parameters derived from D1 vibrissae data obtained by Hartmann et al., 2003. Their model was able to quantify the time-dependent forces and moments at the vibrissa follicle during both non-contact whisking and whisker-object collision.

Another more sophisticated model by Huet et al 2015 increased the dimensionality into 3D space. In a 2D model not all information may be included, therefore a 3D model could provide better insights in transmission of mechanical signals during whisking behavior. Their quasi-static model studied the relationship between whisker bending and the forces and moments at the base, but with the compromise of not incorporating dynamic effects like vibration that follow whisker collisions. However, a large component of whisking can be described with quasi-static models. They also include the intrinsic curvature of vibrissae in their model, which also differs from the former assumed ideal shape. By quantifying the

physical signals transmitted to the mechanoreceptors in the follicle during active whisking behavior, this study provided a foundation for correlating 3D forces and moments at the whisker base with neural signals.

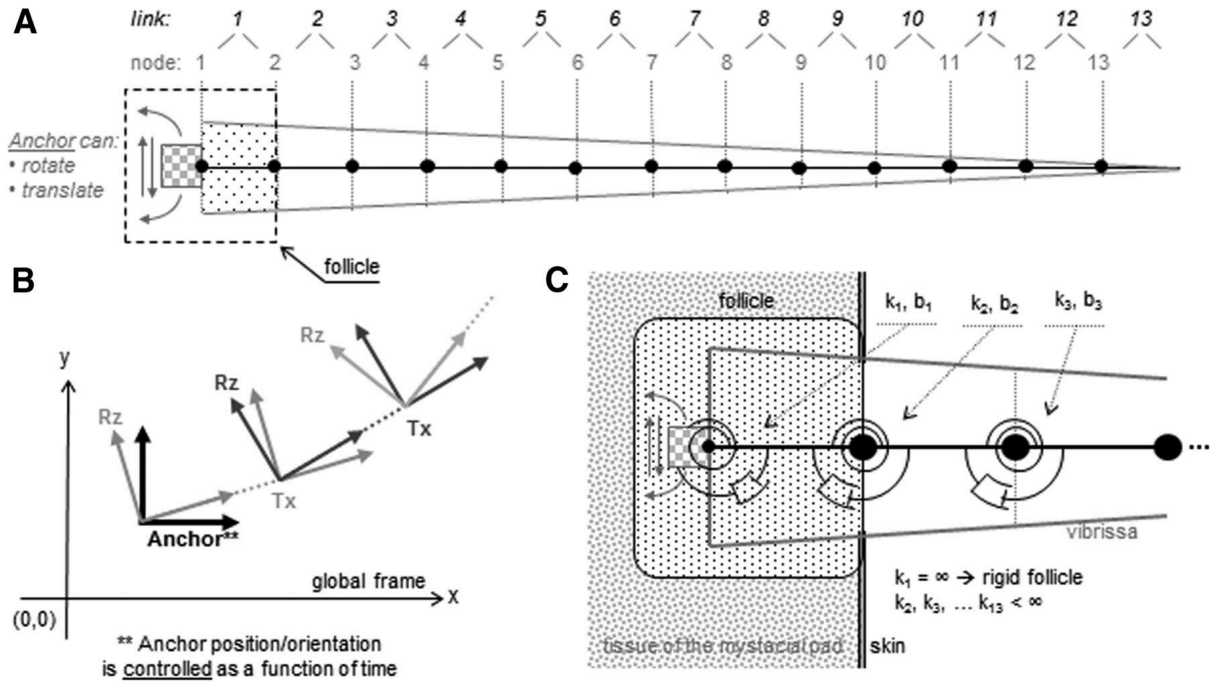

Figure 2.3: 2D model of the whisker. (A) Discretized whisker model with 13 links, connected by nodes and anchored at the base. Each link is represented by a truncated cone with a point mass in the center (B) The global position of each node n is dependent on the position and orientation of former node n-1. The global coordinates can be calculated by using the variable rotation as well as the constant translation along the local x-axis (C) The mechanical properties are modeled by rotational springs with torsional spring with stiffness k and torsional damper with damping b. Adapted from Quist et al. 2014.

As already mentioned, the whisker hair does not contain mechanoreceptors, it only transfers mechanical signals to the follicle. Only there at the vibrissa base the touch receptors can translate the mechanical signals into neural signals. This means the rat has to solve a so-called inverse problem to map the mechanical signals at the follicle into the 3D contact point location (Huet et al. 2022).

How does the rodent determine where exactly in space along its vibrissa the collision occurred? It must be possible to calculate the position of the contact point from the resulting forces and moments in the follicle. To simplify this, one can do this independently of the rats facial orientation and the whisker cycle by looking at it in a whisker based coordinate system. Here, the contact point localization can be represented by the radial distance r, azimuthal angle 0 and polar angle phi.

According to Bernoulli-Euler beam theory, the whisker can be mechanically modeled as a truncated conical beam with a free tip and a fixed boundary condition at the base (Hires et al. 2013; Yan et al. 2013; Huet et al. 2022). In this configuration the information of the mechanical signals F_x, F_y, F_z, M_x, M_y, and M_z at the base are always sufficient to calculate the contact point location. (Tsujimura et al. 1989; Clements et al. 2006).

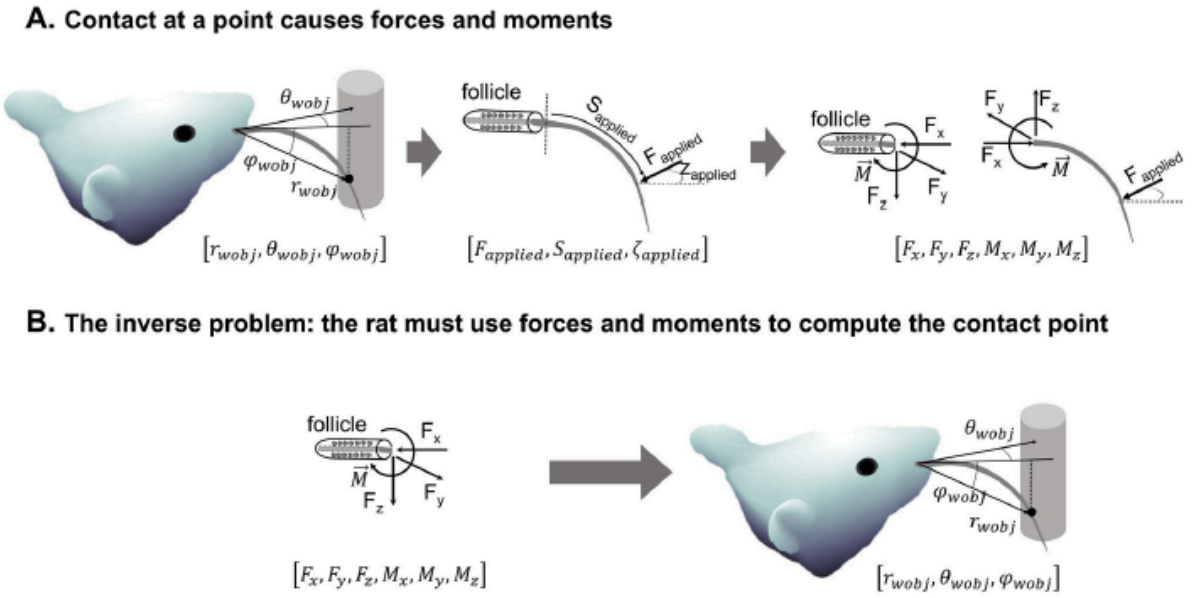

Figure 2.4: Force and Moment Generation During Rat Whisker Contact with an Object for 3D Localization. (A) Labeling 3D contact point between the whisker and the object as r_{wobj}, θ_{wobj}, and φ_{wobj}. This point applies force (Fapplied) to the whisker, generating reaction forces and moments at its base. (B) Inverse problem of the rat: it has to map from mechanical signals to contact location. Adapted from Huet et al. 2015.

The mechanical signals can be converted regarding magnitude and direction of transverse forces and bending moments using the following equations:

$$\textit{Magnitude of the transverse force}: \; F_T = \sqrt{F_y^2 + F_z^2} \tag{1}$$

$$\textit{Direction of the transverse force}: \; F_D = atan\left(\frac{F_z}{F_y}\right) \tag{2}$$

$$\textit{Magnitude of the bending moment}: \; M_B = \sqrt{M_y^2 + M_z^2} \tag{3}$$

$$\textit{Direction of the bending moment}: \; M_D = atan\left(\frac{M_z}{M_y}\right) \tag{4}$$

These signals are more meaningful in relation to the collision location and are helpful when extracting spatial information from forces and moments, though they are not independent from each other, for instance F_D and M_D are about 90° offset from each other, due to the reaction moment occurring mainly by the reaction force. This is worth mentioning as there are approaches where researchers aim to reduce the dimensionality of the mechanical characterization. One of the most recent ones compresses the six variables into just three components that are enough to approximate the mapping to the contact location using a so-called triplet like F_x, M_B, M_D (Huet 2022).

2.1.3 Artificial Whisker

Even though tactile perception has not received as much attention in robotics as visual sensors, it has been studied for decades. A more specialized topic is artificial whiskers, and there have been many approaches to developing artificial vibrissae to pick up spatial information. Making use of this kind of modality can be helpful when visual sensors are not affected by poor conditions such as fog or complete darkness. Even though the overall principle of these artificial whiskers is the same, the solutions for this problem vary in many different aspects, especially regarding transduction methods and application.

The principle of transduction of physical mechanical signals and converting it into electronic information offers a wide range of techniques. Generally speaking, very common here are piezoresistive, capacitive and piezoelectric, but also magnetic and optic effects exist in tactile sensors.

Piezoresistive

Piezoresistive whiskers function based on the piezoresistive effect, which involves a change in the electrical resistance of a material due to applied mechanical stress (Kottapali et al. 2015; Wei et al. 2019). These whiskers are designed to emulate tactile sensing found in animal whiskers - such as those of mammals - and are commonly used in robotics and sensory applications to detect touch, vibration, or airflow.

The material and structure of piezoresistive whiskers are typically made from substances exhibiting piezoresistive properties, like certain polymers or composites. They are shaped to resemble animal whiskers, being flexible and elongated. When these whiskers encounter external forces such as pressure, touch, or air movement, they bend or vibrate, leading to a change in the material's electrical resistance. This variation in resistance is measured and converted into an electrical signal, which is processed to interpret the nature of the external force. These whiskers find applications in various fields, including as tactile sensors in robotics, flow sensing in fluid dynamics and as environmental sensors for detecting air or water currents.

Piezoelectric

Piezoelectric whiskers function by converting mechanical strain into electrical signals (Tiwana et al. 2016). This is achieved through the piezoelectric effect, where certain materials generate an electric charge in response to applied mechanical stress. These whiskers are designed to mimic the sensory functions of animal whiskers, particularly those found in marine mammals like seals (Beem 2016; Jiang et al. 2021).

When a piezoelectric whisker is subjected to mechanical forces, such as bending or vibration, the piezoelectric material within the whisker generates a voltage. This voltage is proportional to the degree of deformation the whisker experiences. The generated electrical signals can then be measured and analyzed to infer information about the external environment, such as fluid flow, object proximity, and texture (Jiang et al. 2021; Tiwana et al. 2016).

Particularly interesting research was developed by Jiang and colleagues in 2021. This study developed an electronic whisker sensor using a PVDF ring with symmetrical electrodes set at the root of a fiber beam. The sensor could sense frequencies and amplitudes of displacements, the height and width of platforms, or the heights of other irregular protrusions. It was also capable of detecting airflow magnitude or direction.

The E-whisker's design and fabrication imitated the structure and function of animal whiskers, using a flexible optical fiber and a PVDF ring around its root to sense bending

deformation.

Capacitive

Capacitive sensors for whisker applications typically consist of two conductive plates (parallel plate capacitors) separated by a dielectric material. When they are subjected by external forces through whisker collision, their position changes slightly, which leads to a change in capacitance between the plates (Delemare et al. 2016). These small alterations can be measured with high precision and then be converted into electrical signals that can be analyzed to determine the direction, amplitude, speed or other characteristics of the impact (Stocking et al. 2010).

The sensitivity is quite high, so that even the smallest changes in the position of the whiskers can be detected. An interesting approach is a sensor developed using 3D printing to create a whisker structure and its base, by Delemare and colleagues in 2016. They created a co-planar capacitive sensing mechanism, where a dielectric material alters the capacitance when moved into the electric field of the capacitor. This setup is achieved by mounting the whisker on a patterned PCB that utilizes the fringing fields of co-planar capacitances to measure the displacement of dielectric parts at the whisker base. The whisker sensor was able to measure both rotations and force at the base, functioning similarly to a torsional spring like that found at the base of a rat whisker for determining contact force and location along the whisker.

Optical

Another innovative method for measuring mechanical strain is the use of vision based sensors. Unlike when a camera is used to explore the environment, here it is mounted inside the robot and looks at a membrane to which the whisker is attached (Lepora et al. 2018).

When the whisker interacts with an object or experiences a force, it bends or vibrates, causing a change in the position or orientation of markers inside the membrane. This movement is captured and analyzed by the optical system to determine the nature of the contact, such as the location, magnitude, or direction of force. Optical whiskers are used in robotics for tactile sensing and environmental interaction, offering detailed feedback about contact surfaces.

In the WhiskSight system, stiff carbon fiber rod whiskers are magnetically connected to a flexible elastomer skin (Kent et al. 2020). This setup allows for easy adjustment and reconfiguration of the whiskers. Below this skin, a camera captures the entire array of whiskers. The movement or rotation of these whiskers, caused by touch or airflow, is recorded through the camera, tracking the position changes of special markers located on the elastomer and the whiskers. This system is particularly adept at distinguishing between

direct touch, where the whisker contacts an object, and indirect stimuli like airflow, by analyzing the captured movement patterns.

The TacWhisker arrays, evolving from the TacTip sensor concept, come in two versions: static and dynamic (Lepora et al. 2018). The static array simulates non-moving whiskers similar to rodent microvibrissae, while the dynamic array features motor-driven whiskers that mimic the active whisking behavior seen in animals like rodents. In the dynamic TacWhisker, whiskers are attached to a tendon system that rhythmically moves them back and forth. This motion is key to the system's ability to sense the environment effectively, allowing the whiskers to make repeated and controlled contacts with objects. An internal camera captures these movements, and the data is processed to determine the location and nature of the contacts made. The dynamic TacWhisker shows particularly effective sensing capabilities, with its whisking motion leading to precise and reliable object location perception.

Magnetic

Magnetic whisker sensors make use of the Hall effect. They are low cost and ready for immediate use. The basic principle behind this is to install a permanent magnet at the base of the vibrissa with a Hall sensor proximal to this sensor (Kim et al. 2020; Zhao et al 2022). If the whisker is now moved, the position of the magnet changes and the sensor measures changes in the magnetic field. Ordinary Hall sensors can be used for this application, but 3D magnetic sensors are often used because of the higher information content.

In one insightful study by Kim and colleagues in 2020, the sensor consists of a fin-like structure suspended by a micro-fabricated spring system with a permanent magnet attached beneath the spring. The motion of the fin, influenced by airflow, causes the magnet to move, which is then detected by a low-cost 3D magnetic sensor located below the magnet. This sensor is designed to capture multi-directional airflow with the fin designed to have high air drag and an optimized spring stiffness for improved performance.

Another study presents a novel whisker sensor using a spring structure to transfer deformation generated by the whisker tip to a permanent magnet at the base, changing its position (Zhao et al. 2022). This movement is detected by the MLX90393 Hall sensor, providing precise object positioning. The sensor's detection range can be expanded by selecting different whisker materials and magnet sizes. The MLX90393, combined with an Arduino UNO development board, reads the flux density signals to display detection results in real time. Calibration experiments were conducted to establish the relationship between tip position changes and magnetic induction intensity data from the MLX90393. Positioning tests verified the sensor's accuracy in determining positions in the detection plane.

2.2 Deep Reinforcement Learning

Within the field of Artificial Intelligence, Deep Reinforcement Learning represents a quite powerful approach that enables agents to make decisions in environments of high complexity and uncertainty. Interestingly enough, DRLcombines two approaches that both were inspired by biology in the beginning: Reinforcement learning, which was inspired by biological learning rules formulated by Bush and Mosteller and later by Rescorla and Wagner, describe how animals learn from rewards and punishments in their environment. Artificial Neural Networks mimic the way the brain works on an abstract level by connecting many nodes, also called neurons, together and allowing them to learn and process data by changing the strength of their connection.
The combination of traditional Reinforcement Learning with neural nets leads to artificial systems that are capable of superhuman performance (Schulman et al. 2017; Berner et al. 2019). This can be seen when playing Atari video games or even competing with professional gamers in highly dynamic multi-agent environments such as Dota 2.

2.2.1 Reinforcement Learning

Reinforcement Learning (here referred to as RL) in general describes the method by which an agent learns optimal behaviors through trial-and-error interactions with an environment. This learning paradigm is distinct from supervised or unsupervised learning as it involves active decision-making by the agent based on the outcomes of its actions. The RL process, illustrated in Figure 2.5, can be compared to many learning processes in the real world, such as driving a car: The novice driver (agent) learns strategies for steering a vehicle (policies) by receiving direct feedback from the instructor (interacting with the environment) and then iterating this learning cycle with adjusting behavior.

Agent: In RL, an agent is the decision-maker, analogous to the novice driver in the vehicle scenario. It can be a program, robot, or any system capable of making decisions.
Environment: The environment encompasses everything the agent interacts with. In the driver analogy, it includes the car, the road and all components of traffic, each governed by specific rules.

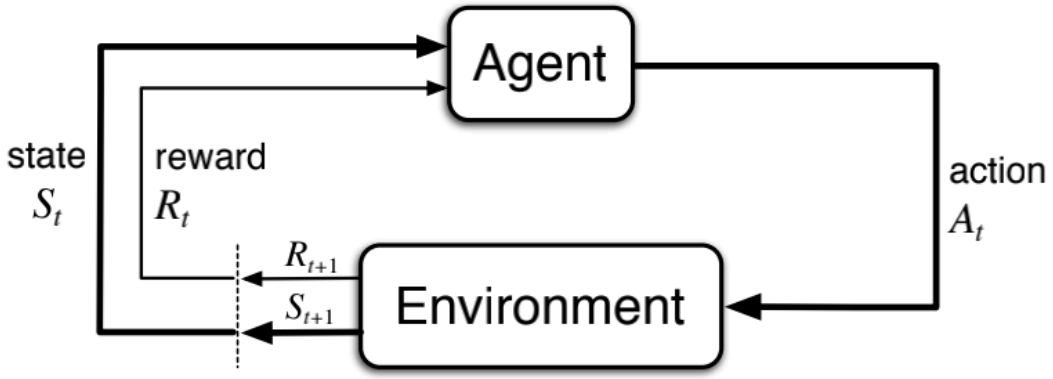

Figure 2.5: Reinforcement Learning Loop. Adopted from Wang et al. 2021.

Markov Decision Processes: The Core of RL

So called Markov Decision Processes (MDPs) offer a formal framework for sequential modeling of the decision-making process in RL. An MDP is defined by states, actions, transitions, and rewards:

States (S): The set of all possible situations or configurations in the environment.
Actions (A): Choices or moves the agent can make.
Transitions: The outcome of taking an action in a state, leading to the next state. In driving, pressing the gas pedal and steering the wheel changes the car's position.
Rewards (R): After each action, the agent receives feedback. In driving, traveling a distance safely on the road might be seen as a positive reward, while losing getting off the road or having an accident could be negative.

Before going deeper into RL, it is important to mention that a fundamental aspect of MDPs is the Markov Property. This characteristic asserts that the future state depends only on the current state and action, not on the history of states and actions. This property simplifies the complexity of decision-making and forms the basis for many RL algorithms. So if the agent has all information about the current state that is required for a decision, this is referred to as a fully observable Markov decision process. On the other hand, there is also the partially observable Markov decision process (POMDP). Here in every instance, the agent gathers only observations influenced by the current state. However, the agent's decision-making is based solely on the history of past observations and actions, as it lacks the capability to directly perceive the present state.

The latter is worth mentioning because in the real world it is often not possible to fully describe the current state. Depending on the complexity of the system, there are factors in a state that are not observable but can be neglected based on other observations, e.g. the

speed of the car not only depends on the gas pedal but also all the processes in the motor, friction between the wheels and the ground, slope and so on. Nevertheless, the whole problem can be simplified by just linking the pedal with the speed and adjusting the pedal if one approaches a slope and wants to keep the same velocity as before.

So an RL agent engages with its environment over a series of time steps. At each time step, denoted as t, the agent is presented with a state st from the state space S and chooses an action at from the action space A, guided by a policy $\pi(a_t|s_t)$. This policy represents the agent's behavior, essentially a function mapping states st to actions at. Upon taking an action, the agent receives a scalar reward r_t and moves to the next state s_{t+1}, as determined by the environment's dynamics, encapsulated in the reward function R(s,a) and the state transition probability $P(s_{t+1}|s_t,a_t)$. In episodic scenarios, this sequence continues until reaching a terminal state, after which the process restarts.
The agent's objective is to maximize the expected cumulative reward over time, known as the return. This return is discounted by a factor γ within the range (0,1] and is mathematically represented as

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}.$$

The aim is to optimize this long-term return for each state within discrete or continuous state and action spaces.

So the different states can be measured regarding how good they are with respect to the expected, accumulative, discounted future reward. Mathematically speaking, the state value can be evaluated by the value function:

$$v_\pi(s) = \mathbb{E}[R_t|s_t = s], \text{ where, } R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k},$$

The same is true for the action value that can be assigned to each state-action pair following the policy π,

$$q_\pi(s,a) = \mathbb{E}[R_t|s_t = s, a_t = a],$$

The value function $v_\pi(s)$ is articulated through the Bellman equation, breaking down the function's components into a recursive relationship with s' as the previous state.

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')].$$

Since a value can be assigned to each state, there is also a maximum that represents the optimum state value v_*. It can be achieved by any policy for a given state s by taking the maximum of the action value function $q_*(s, a)$ over all possible actions a.

$$v_*(s) = \max_\pi v_\pi(s) = \max_a q_{\pi^*}(s, a),$$

This optimal state value $v_*(s)$ can also be formulated in a recursive manner, illustrating that the optimal value of a state s is the highest expected return for taking an action a, also considering the immediate reward r and the discounted value of the next state s'.

$$v_*(s) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_*(s')].$$

The action value function $q_\pi(s,a)$ predicts the expected return from taking an action a in state s, then following a policy π, combining the expected immediate reward and the discounted future rewards.

$$q_\pi(s, a) = \sum_{s',r} p(s', r|s, a)[r + \gamma \sum_{a'} \pi(a'|s')q_\pi(s', a')].$$

The optimal action value function $q_*(s,a)$ is the highest value that can be obtained by any policy for a specific state-action pair.

$$q_*(s, a) = \max_\pi q_\pi(s, a),$$

So q_* decomposed into the Bellman equation, q_* represents the expected return for taking action a in state s, followed by choosing future actions that maximize the action value function in subsequent states.

$$q_*(s, a) = \sum_{s',r} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')].$$

The optimal policy is described by π_*.

When addressing the challenge of optimizing the agent's decision making, there exist different approaches. They are briefly explained in the following, laying out all corresponding equations would go beyond the scope of this thesis.

Dynamic Programming (DP): Dynamic Programming is a strategy used when the environment's complete model is known. It involves decomposing a complex problem into simpler subproblems, solving each subproblem only once, and storing their solutions - often

referred to as the principle of optimality. DP methods, such as Policy Iteration and Value Iteration, are powerful in structured problems where decisions are sequenced in stages and the effect of actions are deterministically known. Despite their theoretical robustness, DP methods can be limited by computational constraints, especially in environments with large state spaces, as they require calculations across all states.

Monte Carlo (MC) Methods: Monte Carlo methods offer a model-free approach, learning directly from episodes of experience without a priori knowledge of the environment's dynamics. These methods estimate value functions and policies based on averaging the returns (total accumulated rewards) observed in complete episodes. The key advantage of MC methods is their ability to learn from the actual experience, making them applicable to a wide range of problems, especially where obtaining a full environmental model is challenging or impractical. However, one limitation is their reliance on episodes being completed, which can be a constraint in continuing environments or tasks with long or infinite horizons.

Temporal Difference (TD) Learning: TD learning combines ideas from both DP and MC methods. Like MC, TD methods can learn directly from raw, incomplete experiences, and like DP, they update estimates based on other learned estimates (also called bootstrapping). A significant advantage of TD learning is its ability to learn online from every step of experience without waiting for the episode's conclusion, making it more versatile and generally faster than MC methods. TD methods, such as Q-learning, are particularly effective in situations where episodes are long or indefinite and in environments where immediate feedback is available for actions. So in the context of DRL this method is the most common one, allowing for a broad range of problems with varying complexity. Within this method, there are also various algorithms for solving the optimization problem, which are discussed later in the section on DRL algorithm.

Function approximation

Until now, every method belongs to tabular RL, which means one needs a table where each state-action value has to be visited many times to get good results. If a state was never visited, no policy can be derived, still the initial value will be encountered. Therefore we need the ability to somehow generalize the policy between states. This can be done because very similar states have a high probability to have the same optimal action.

Tabular RL just allows a small discrete action space, discretizing continuous action spaces or a large action space in general leads to a so-called curse of dimensionality, as an increased number of features increases the problem exponentially.

Therefore, the use of function approximation is an important tool to boost the capability of

RL. There are many different methods that can be employed to approximate the value functions; the easiest one would be to project the state information into a feature space for a parameterized representation and apply a linear algorithm to estimate the value function. In respect of more complex solutions that are more suited for real world problems it seems like neural networks are the most powerful way to do that.

2.2.2 Artificial Neural Networks

Neural networks got their name as they are inspired by how the brain works, which connects a huge amount of neurons with different strengths to work as a non-linear parameterized function estimator. In an Artificial Neural Net this is done on an abstract level, having a relatively small amount of neurons that are typically arranged in different layers. It is possible to utilize them for both regression tasks and classification, which allows for continuous and discrete outputs. All stated information is based on the chapter “Artificial Neural Networks” in “Fundamentals of Neural Networks” by Yin (2022).

Artificial Neurons

The most basic cell of Artificial neural nets is the Artificial neuron (also called Perceptrons, Units or Nodes). It can be depicted in two units and is illustrated in Figure 2.6. First there is the aggregator unit. Here first all inputs are multiplied with the corresponding weight and then sums everything up with the bias. Afterwards, the aggregated sum is inserted into the non-linear activation function that introduces a non-linearity to the whole system before resulting in the output. The non-linearity implementation is a very important factor for applying the learning algorithm.

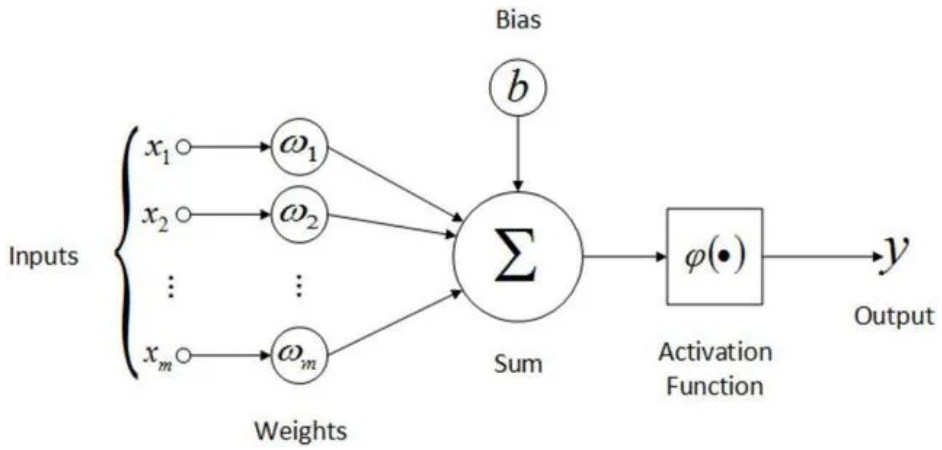

Figure 2.6: Basic Artificial Neuron. The inputs x are multiplied with the weights w_i, then summed up with b and inserted in the activation function to get the output y. Adapted from De Oliveira, 2017.

Stochastic Gradient Descent

SGD is employed to optimize the parameters of deep neural networks, which are instrumental in approximating the policy or value functions.

The essence of SGD lies in its iterative approach to minimize a loss function L, which, in the context of DRL, often involves terms like the temporal difference error. This is distinct from supervised learning, where the loss function typically measures the discrepancy between predicted outputs and actual targets. In DRL, the loss function is designed to refine the policy or value estimates, guiding the agent towards optimal behavior.

At the heart of SGD's application in neural networks is the backpropagation algorithm. Backpropagation computes the gradient of the loss function with respect to the network's weights. These gradients indicate the direction in which the weights should be adjusted to reduce the loss. The network's weights w_i and biases b_i are then updated in the direction of the negative gradient, as illustrated in Figure 2.7.

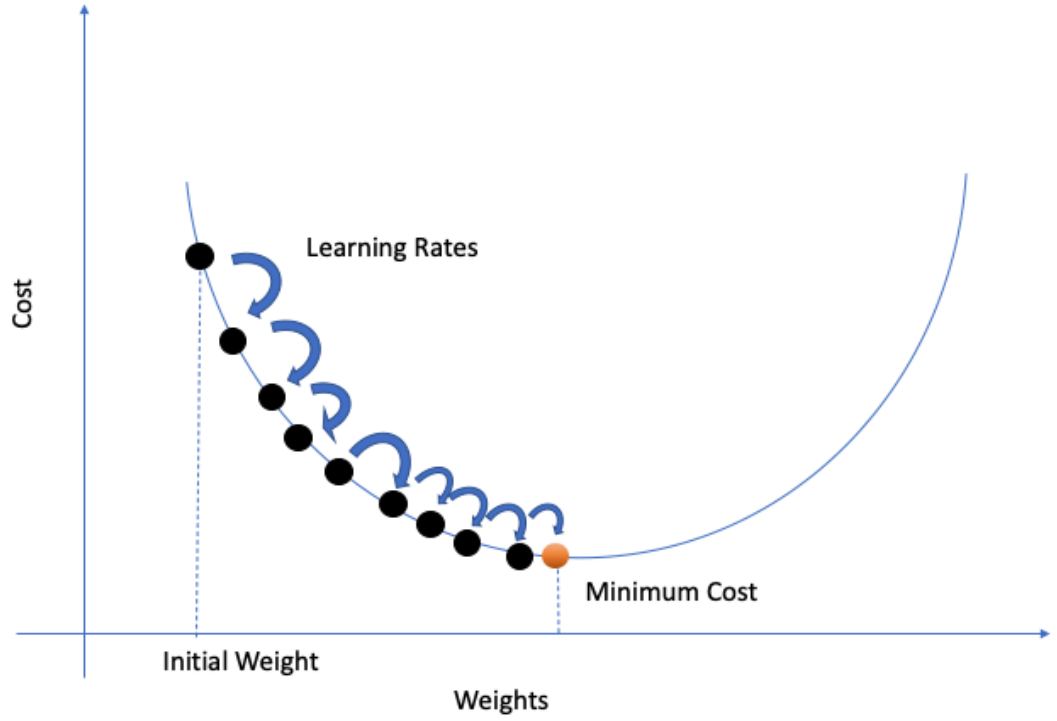

Figure 2.7: Stochastic Gradient Descent. Adopted by Gosh et. al 2020.

In practical DRL applications, additional considerations include choosing an appropriate batch size and implementing techniques like gradient clipping to mitigate the issue of exploding gradients. The use of experience replay in DRL also affects gradient computation and application, as it allows for more stable and efficient learning from past experiences.

Multilayer Perceptron

By connecting these neurons sequentially in different layers a neural network is formed, a system that can be optimized to learn the non-linear relationship between input and output which is depicted in Figure 2.8. This can be done by calculating the predicted output as described before and comparing it to the actual value. The loss function computes a loss that is then fed into the optimizer. There through stochastic gradient descent a weight update is performed to change the behavior of the neural net for a slightly better prediction regarding the seen values. This process is repeated for many steps until it seems like the minimal value of the loss function is reached.

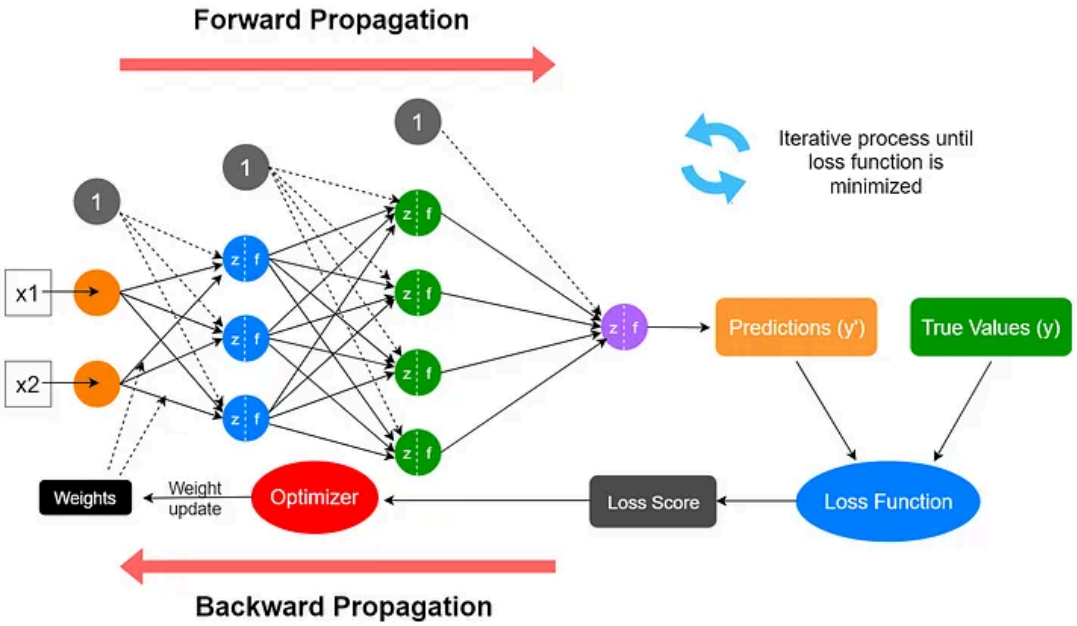

Figure 2.8: Neural Network and its optimization process. After computing the output y', it is compared to the true values to generate the loss. This value is then put into the optimizer to modify the weights of the neural net.

2.2.3 Proximal Policy Optimization

The proximal policy optimization (PPO) algorithm developed by Schulman and colleagues in 2017 at OpenAI is an extended version of the Trust Region Policy Optimization. It has proved to be quite powerful but still isn't too sensitive to hyperparameters that even without much fine tuning good results can be achieved. It is known for its stability and robustness compared to other policy gradient methods.

The core concept of PPO is in balancing exploration and exploitation while ensuring stable and consistent policy updates. In simpler terms, it seeks to improve the policy without making drastic changes that could lead to poor performance or instability.

One of its novel features is the clipped surrogate objective. There also exists another approach where a penalty on KL divergence is used but this is just mentioned for the sake of completeness because it performs worse than the clipped one.

The clipped objective function is proposed by them in this equation:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

This function is defined as the expectation of the minimum between two terms: the product of the probability ratio r_t (θ) and the advantage estimate A_t, and the clipped version of this product. The probability ratio $r_t(\theta)$ is calculated by the ratio of the probability of an action under the current policy to its probability under the previous policy. The clipping mechanism, governed by a hyperparameter ϵ, restricts this ratio to a specified range, typically between $1-\epsilon$ and $1+\epsilon$. This clipping ensures that the policy updates do not deviate too drastically, safeguarding the learning process against harmful large updates.

The advantage function A_t in PPO is often computed using Generalized Advantage Estimation (GAE). GAE combines the idea of TD learning with a bias-variance tradeoff, controlled by the parameter λ. The advantage estimator provides a measure of how much better an action is compared to the average action at a given state, guiding the policy towards more rewarding actions.

$$\hat{A}_t = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T)$$

In addition to the clipped objective, the total loss function in PPO includes a value function loss and an entropy bonus. The value function loss penalizes the deviation of the estimated state values from the values predicted by the current value function. The entropy bonus, $S[\pi\theta](s_t)$, encourages exploration by rewarding more diverse action selections. The coefficients c_1 and c_2 balance the contributions of the value function loss and the entropy bonus, respectively.

The iterative process of PPO involves using stochastic gradient ascent to maximize the total loss function $L_t(\theta)$. This process refines the policy by integrating the insights from the clipped objective, the value function, and the entropy term, leading to a balanced and effective learning trajectory.

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\big[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\big],$$

3 Methods and Implementation

This chapter presents research methods and implementations employed to integrate artificial tactile whiskers into a neurorobotic mouse, termed NeRmo. This implementation is crucial for the implementation of tactile sensing in the mouse robot, requiring a fusion of mechanical engineering, artificial intelligence, and computational simulation.

Initially, the details of the NeRmo robot are laid out, showcasing its major role in this research. A comprehensive understanding of NeRmo design, operational mechanics, and pre-existing features is essential. This foundation is critical for the successful integration of tactile whiskers, which promises to significantly enhance the robot's sensory and navigational abilities.

The focus then transitions to precisely describing the development process of the tactile whiskers for the mouse robot. Beginning with the testing and prototyping phase, the general concept was chosen and checked for suitability for this study. Then the hardware design is examined, laying out the constructions and modifications done by using Computer-Aided Design (CAD).
Subsequently, the setup of a Finite Element Analysis (FEA) is described which is used for validation of models functionality. Additionally, this analysis is used to examine misalignments between sensors in the idealized scenario and those in the real-world robot, thereby clarifying impact on their behavior.
All CAD and FEA tasks described in this research were conducted using Autodesk Fusion 360 and ANSYS, respectively.

Further, the chapter explores the deployment of these tactile sensors within the context of Obstacle Avoidance. This exploration is twofold: initially focusing on the virtual implementation within a physics simulator to refine and validate their functionality, followed by an in-depth exploration on the conceptual framework for application of Deep Reinforcement Learning. This segment highlights how a neural net is trained for robot's decision-making and navigational strategies based on tactile feedback. Concluding with the robotic implementation, the trained neural net is utilized for making real-time decisions of the physical NeRmo robot. This final step is crucial, validating the practical use case of the simulated work, thus successfully demonstrating the implementation of tactile sensing in a bio-inspired robotic mouse.

3.1 NeRmo Robot Foundations

Through a collaboration between the departments of zoology and computer science at the Technical University of Munich the NeRmo robot was developed as part of the Human Brain Project (Bing et al. 2023). The primary goal was to provide insights into the control mechanisms needed to create motions and actions similar to those of the biological counterparts. The design of the neurorobotic mouse emulates physiological and functional properties of the real animal by incorporating biomechanical features like the use of artificial tendons. This allows its application for detailed studying of locomotion and navigation in rodents. Even though the original NeRmo included much more functionalities and sensors, here the simpler version NeRmo Lite is described, which was used to decrease complexity of the system before implementing a sophisticated tactile system. In the further text always the latter one is meant without mentioning explicitly the Lite version.

3.1.1 Key Design Features

Regarding the functional parts of NeRmo, particularly the spine and the limb mechanics differ from traditional four-legged robots. In figure 8 is an exploded view of the NeRmo robot.

Spine: Central to NeRmo innovative design is its backbone, which represents a significant departure from the rigid structures commonly found in traditional robotics. Unlike conventional designs, NeRmo features a flexible spine comprising several vertebrae, each designed with holes on both sides for guidance of artificial tendongs. These vertebrae are connected by a narrower structure, specifically engineered to allow for lateral and sagittal flexion. This enables the spine to bend towards the left and right as well as up and down, closely mimicking the flexible spine movement seen in real rodents.
The movement of the spine is controlled by a servo mechanism. Attached to a servo in the back, the artificial tendons are routed through the vertebrae on both sides and fixed in the hip. This arrangement not only facilitates the precise control of spinal movements but also effectively replicates the biomechanics of a rodent's spine.

Limbs: The design of NeRmo's limbs incorporates a three-bar linkage mechanism that simulates the movement of a rodent's legs. Each limb is controlled by two servos which manage the movement of the first two bars in parallel, with the last bar acting as the point of contact with the ground. This configuration allows for a natural sequence of movements seen in rodent locomotion, enabling the limbs to perform a complete cycle of motion that includes lifting backwards and stepping forwards.

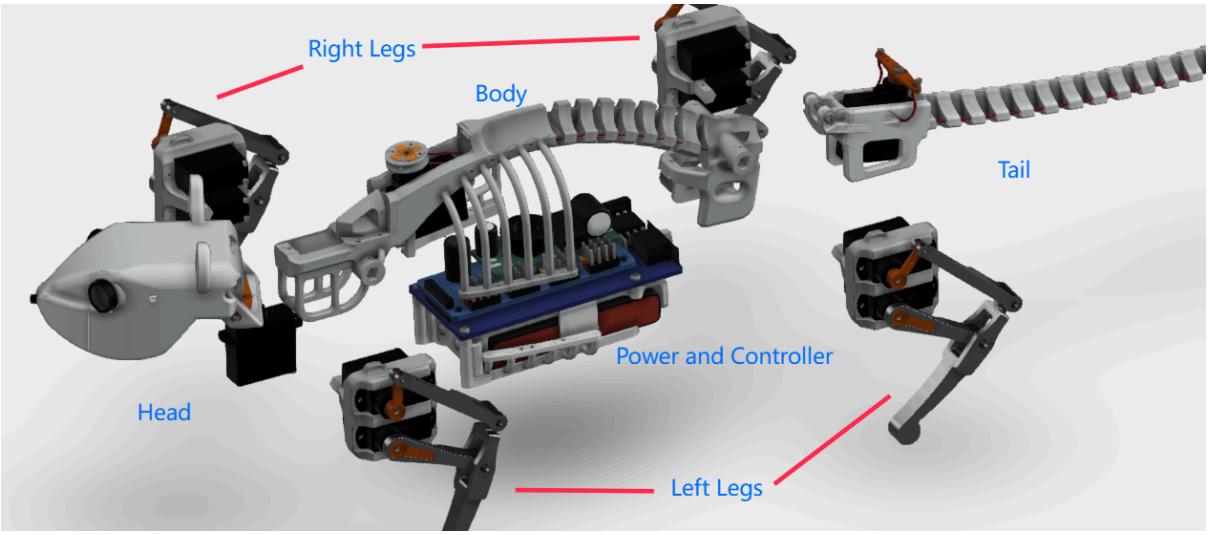

Figure 3.1: Exploded view of the robot. Provided by TU Munich.

3.1.2 Construction and Communication

The core structure of the NeRmo robot is its 'skeleton', which is designed to be fabricated by a 3D printing process. It provides the mechanical framework and houses the electronics.
The selection of the electronic components for the mouse robot, including the 12 servo motors, the 7.4V 2s LiPo battery, the ESP01 Wi-Fi module, the voltage regulator, the PWM motor board, and the assembly materials such as the 0.6mm fishing line and screws, were conducted by colleagues at TU Munich. This bill of materials ensured a balance between functionality and accessibility of the components, allowing for a cost-effective assembly, with the total expense amounting to approximately 400 €, of which the servo motors constituted the major expense at 240 €.

Lacking onboard hardware for complex computation, the design leverages the ESP01 module to facilitate TCP/IP communication with a computer providing a hotspot the mouse robot can connect to.
This setup allows for the remote control of the robot's servos, providing a user-friendly interface for real-time interaction while still allowing for relatively large computations that can be useful in many scenarios like utilization of neural nets.

3.1.3 Controller

To control the locomotion of the robot, a basic controller as well as a gait model from the TU Munich is employed, designed specifically for biomimetic locomotion of this robotic mouse.

The controller's functionality begins with the establishment of fundamental movement parameters. This includes specifying the frequency cycle, which dictates the pace at which the robot executes its gait, along with the phase difference between the individual legs to ensure coordinated movement. Additionally, the amplitude for the spine's movement is determined, defining the range of its bending and flexing motions. This configuration specifies not only the nature of the robot's gait but also offers the flexibility to modify these parameters as needed to achieve different locomotion styles.
By internally keeping track of the current phase of the gait cycle and integrating this information to the leg model, the subsequent motor commands are calculated when requested. In essence, each leg operates in a cyclic motion, phased differently from the others, allowing coordinated movement for locomotion. When walking straight forward, the spine oscillates as well in the same frequency as the legs to increase the speed of the robot.

Furthermore, the controller was modified so the robot was able to execute turns by adjusting the spine's amplitude and fixing it in place. Steering is achieved by assigning a positive value for left turns or a negative value for right turns to the spine's maximum amplitude. This adjustment allows for controlling the robot's walking direction, significantly improving its maneuverability and capacity to navigate complex environments.

3.2 Whisker Sensor Development for Mouse Robot

Before developing a sophisticated whisker sensor for the robotic mouse, a suitable concept had to be selected and tested using a prototype. After proven successful, adjustments for the more specific implementation into the robot had to be made.

It's imperative to note that the documentation of this development process, while presented in a sequential manner, was conducted in an iterative manner. Each iteration served as a platform for refinement - ranging from enhancing the isotropy of the components to simplifying the insertion of modules into the robot's head. This iterative approach not only ensured the functional efficacy of the design but also its optimization for performance and ease of assembly.

3.2.1 Proof of Concept and Prototyping

To develop an advanced whisker sensor for the NeRmo robot, it was essential to first identify specific requirements necessary for a meaningful implementation.
Subsequently, a concept of the reviewed literature matching these requirements was identified. A testing setup was developed to check if the system behaves as expected and is suitable for biometric tactile perception in the neurorobotic mouse.

Requirements

The design of the tactile whisker sensor for the NeRmo robot prioritizes ease of manufacture to ensure that the overall costs of the system remain manageable, without necessitating expensive machinery or materials. This aspect is crucial for enabling an experimental setup conducive to iterative enhancements, where adjustments can be made efficiently and cost-effectively. Moreover, the straightforward and accessible design facilitates replication and adaptation by other researchers, promoting wider application and further innovation in the field.
Furthermore, the sensor system needs to be compact enough to accommodate a minimum of two whiskers on each side of the NeRmo robot's head, given the constraints of limited available space. This design approach draws inspiration from the research conducted by Fox and colleagues in 2012.

The sensor should provide meaningful data, enabling extraction of both impact direction and amplitude as the biological counterpart does (Huet et al. 2015). Implicit in this requirement is the necessity for replicability, ensuring consistent sensor readings from similar whisker deflection. Furthermore, processing raw data to determine the direction and amplitude of impacts should be straightforward, facilitating real-time analysis.

Concept selection

After evaluating various principles of existing artificial whiskers, the magnetic whisker concept stands out as the most suitable option. It effectively balances the requirements for specific materials and manufacturing machinery, ensuring cost-effectiveness while maintaining production efficiency. Additionally, the resulting sensor design can be crafted with space efficiency in mind without compromising on accuracy. Therefore, the decision is made to adopt the magnetic whisker concept, utilizing a 3D Hall sensor, as the foundational approach for the NeRmo robot's tactile whisker sensor.

Hall sensors are inherently compact, which allows them to be integrated into tight spaces without compromising functionality. They are usually cost-effective and are available with 3D measurement of the magnetic field. This enables the sensor to determine the exact direction of the magnet's movement, which corresponds to the bending of the whisker.
Moreover, the sensor's ability to gauge the magnetic field's intensity translates into an accurate measure of the amplitude of whisker deflection. As the attached magnet moves in response to contact, the Hall sensor captures this movement in real-time, providing immediate feedback on the strength of the interaction.
The simplicity of the system's design is noteworthy. The components needed for the magnetic sensing are not only widely accessible but also can be easily interfaced with prevalent microcontrollers, making the system user-friendly and easily replicable.

Finally, the real-time data processing capability of the magnetic setup is vital for applications requiring immediate feedback. The quick response of the Hall sensor to changes in the magnetic field allows for a dynamic analysis of tactile data.

Testing

To evaluate the practicality of a whisker system incorporating a magnet and a 3D Hall sensor, a specialized testing setup was developed, as illustrated in Figure 3.2. This setup included the acquisition of an Infineon TLE493D-P2B6 MS2GO development kit, complemented by a whisker prototype having a compact neodymium magnet, measuring 1 mm in both diameter and height, glued to a 3 cm brush hair. Additionally, two custom components, extensively discussed in the forthcoming text, were constructed and 3D printed using a Fused Deposition Modeling (FDM) printer with a standard 0.4 mm nozzle.

The centerpiece of the experimental setup was the design of a flexible bearing, achieved through the fabrication of a planar spiral structure made from Thermoplastic Polyurethane (TPU). In the center an open pipe with an inner diameter of 1 mm was integrated into the design to house the magnet. The selection of a spiral geometry was based on its ability to elongate the path length from the point of impact to the magnet. This design principle was crucial because a longer path length results in a more pronounced deformation and, consequently, a larger rotation of the magnet for any given force. Such a feature significantly enhances the sensitivity of the system to tactile stimuli.
Moreover, the spiral design ensures an isotropic response to deformations. Regardless of the direction of the applied force, the spiral structure deforms in a manner that consistently translates mechanical energy into rotational motion of the magnet. This isotropy is vital for a

sensor system designed to mimic the omnidirectional sensitivity of biological whiskers, ensuring uniform sensitivity across all directions of force application.

The external end of the spiral was designed to connect to a box-like positioning component, featuring an open side for easy insertion of the sensor board. This component made from rigid Polylactic Acid (PLA) played an important role for the assembly's functionality in two significant ways. Firstly, it ensured the magnet was securely aligned directly above the sensor, preventing any movement. This stationary positioning, in combination with the close proximity of the magnet to the sensor, was crucial for maintaining a consistently strong signal. Secondly, it allowed for the sensor board's straightforward lateral insertion. This design feature was critical for maintaining repeatability of measurements: even after the board was detached from the box, its design ensured a precise fit upon reinsertion, preserving the exact positioning and orientation relative to the magnet and sensor. This careful arrangement ensured reliable and consistent data collection across various testing periods.

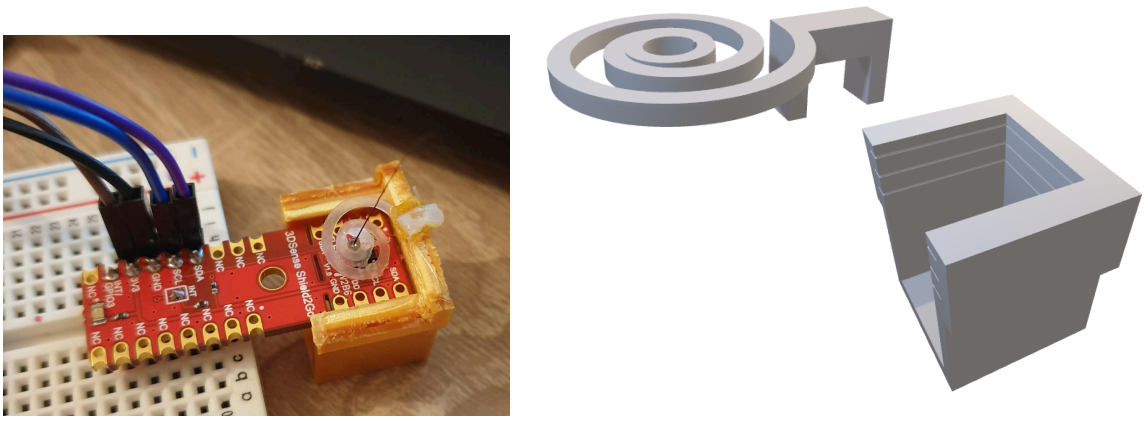

Figure 3.2: Testing Setup for Proof of Concept. Left: Real Setup. Right: flexible mounting spiral and in-place holding box.

To validate the system's sensitivity and repeatability, a series of tests were conducted to assess whether a consistent deflection of the whisker, applied in a specific direction, would yield repeatable results. Each test included manually displacing the whisker tip by a measure of about 1 cm, executed three times consecutively in both positive and negative orientations along the X and Y axes. These axes lie parallel to the plane of the spiral structure, integral to the sensor's design. Through a visual comparison of the sensor signals obtained from these methodically controlled displacements, the objective was to confirm the system's capacity for generating reliable readings irrespective of the direction of impact. This approach not only tested the system's repeatability and sensitivity but also allowed for an assessment of its

isotropic response capabilities, evaluating whether the sensor's performance remained consistent across all tested directions.

3.2.2 Hardware Design

A critical aspect was the implementation of a magnetic whisker sensor within the robot's head. This procedure required several stages, including the design of a whisker system for tactile perception and modification of the head for mounting them along with the associated sensor's printed circuit board (PCB). All CAD modeling work, including the development of the whisker system and modifications to the robot's head, was conducted using Autodesk Fusion 360.

The design and sourcing of the PCB was done by a colleague, after discussing the requirements. The result was a compact PCB that accommodates two 3D Hall sensors, mounted one centimeter apart from each other. This decision was guided by the intent to equip each side of the NeRmo robot with two whiskers.

Whisker Module

The whisker module is central to the tactile hair system, designed to translate impact forces received by the whisker into deflections of the magnetic field detectable by Hall Sensors fixed in the mouse robot head. Achieving this requires the deformation of a flexible structure that not only connects the magnet to the head mounting but also maintains high sensitivity and nearly isotropic response characteristics to accurately interpret signals from any direction of impact.

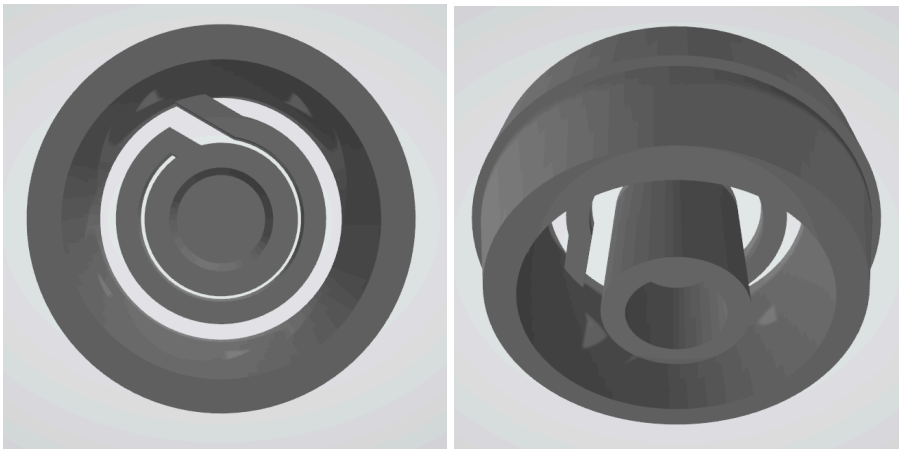

Figure 3.3: CAD model of whisker module that holds magnet and nylon string.

The initial phase involved creating a design that addresses the core requirements: compactness for dense packing within the system, and nearly isotropy for consistent signal interpretation regardless of impact direction. This design was conceptualized with a focus on integrating a flexible structure to house the magnet and connect it to the head mounting, ensuring that even minimal forces from whisker deflections are effectively translated into magnetic movements. The resulting whisker model is illustrated in Figure 3.3.

Given the need for flexibility and sensitivity, the module was fabricated using TPU, the same material as the spiral in the previously mentioned testing chapter. Again, this was chosen for its superior properties in capturing small deflections without compromising the module's integrity. This choice contrasts with more rigid materials, like PLA, which could not provide the same level of sensitivity required for the system's functionality.

Addressing the challenges of miniaturizing and fabricating the whisker module required innovative solutions to overcome the inherent limitations of FDM 3D printing. Key among these was the issue posed by the standard 0.4 mm nozzle, which restricts the minimum size of printed features in xy-plane, crucial for achieving the module's compact design. By employing a specialized 0.2 mm nozzle, it became possible to produce finer structures necessary for the module's detailed design.
However, this adaptation introduced a significant challenge with TPU 'stringing.' This phenomenon, where elastic and sticky TPU material extrudes unintended filaments between print sections due to its viscous nature. This threatened to compromise the module's structural integrity and sensitivity, having small strings between the inner and outer cylinder. To mitigate this, design adjustments were essential. Optimizing the spacing between the module's inner and outer walls was a critical step in minimizing strings connecting these structures, as they can only travel small distances.
Furthermore, the module's outer wall featured a variable thickness to facilitate easier installation into the head model, ensuring consistent insertion depth across all modules. This consistency is vital for the system's uniform sensitivity, underscoring the complexity of designing and fabricating a miniaturized yet highly sensitive and reliable whisker module.

Head Modifications
The design of the head had to be modified for both securely attaching the whisker modules and ensuring the Hall Sensor on the PCB remained precisely positioned without shifting.

For the installation of the whisker modules, in the CAD model pairs of holes were precisely created on each side of the head, with their centers spaced one centimeter apart. These

holes were strategically aligned on a horizontal plane. One hole was placed towards the front, allowing the mouse to detect objects in its direct path, while the other was positioned on the side to facilitate wall detection and lateral sensing. Consequently, when the mouse faces forward with the head straight, the whiskers, once inserted, align in a straight line parallel to the ground.

To accommodate the PCB within the head, modifications were necessary to integrate its shape into the design. To ensure a proper fit against the inner surface of the head's outer wall so the Hall-Sensor is as close to the magnet in the whisker module as possible. Given the use of a flexible PCB with compact design, a narrow groove was developed to include the PCB, strategically positioned to be accessible from the open underside of the head. To secure the PCB against displacement due to gravity, a pair of subtle overhangs were introduced on the inner aspect of the outer wall. Additionally, a support column was implemented to prevent any inward movement of the sensor board, serving as an effective retention mechanism for the pre-bent PCB. To simplify the insertion process, this column was designed with a chamfer, allowing the circuit board to be easily slid into place with a simple push, ensuring it remained securely at the predetermined location.

3.3 Optimization, Validation and Insights with Finite Element Analysis

In the task to enhance the tactile sensory capabilities of the NeRmo robot, the development of an optimized whisker module stands as a key role. This chapter delves into the application of Finite Element Analysis (FEA), a crucial computational tool utilized to examine and refine the design of the whisker module. This approach involved two different simulations - mechanical and magnetic - each serving as focal points, addressing distinct yet complementary objectives crucial to the module’s functionality..

The mechanical aspect of FEA was used to validate the structural integrity and isotropic responsiveness of the whisker module under various mechanical deflections. This analysis ensures that the module can accurately translate these mechanical interactions into consistent rotations and amplitudes of the embedded magnet, regardless of the direction of force application.

In addition to the mechanical evaluation, the magnetic FEA was conducted to examine the response of the system to mechanical deformation and to align the theoretical magnetic responses with actual measurements obtained from the module. This alignment is important for fine-tuning the sensor's position relative to the magnet, ensuring the module's design not only meets theoretical expectations but also works in real-world operational contexts. By precisely approximating the position of real sensors in relation to the magnet, the magnetic FEA facilitates the accurate conversion of mechanical movements into detectable magnetic field changes, thereby enabling the sensor system to capture meaningful data regarding external stimuli.

3.3.1 Mechanical Simulation for Optimization and Validation

The first step in the mechanical FEA of the whisker module involved importing the assembled CAD model, which included the whisker, magnet, and whisker module components.

The assignment of mechanical properties to the existing geometry, the default values of the materials in the library provided by ANSYS were used. The whisker got the material nylon as it is in the real hardware. Accordingly, TPU was used for the whisker module. The magnet was designated as steel in the analysis since the specific material, Neodymium-iron-boron, was not available in the library. However, as these metals are significantly harder than polymers, this substitution will not affect the results. Theoretically, the magnet can be considered non-deformable.

Before transforming the continuous geometry to the generated mesh with discrete nodes, the whisker module, as the point of interest, was selected to be the area of a mesh refinement. This was necessary as the accuracy, represented by fineness of the mesh, at this component is important, opposed to the magnet or whisker.

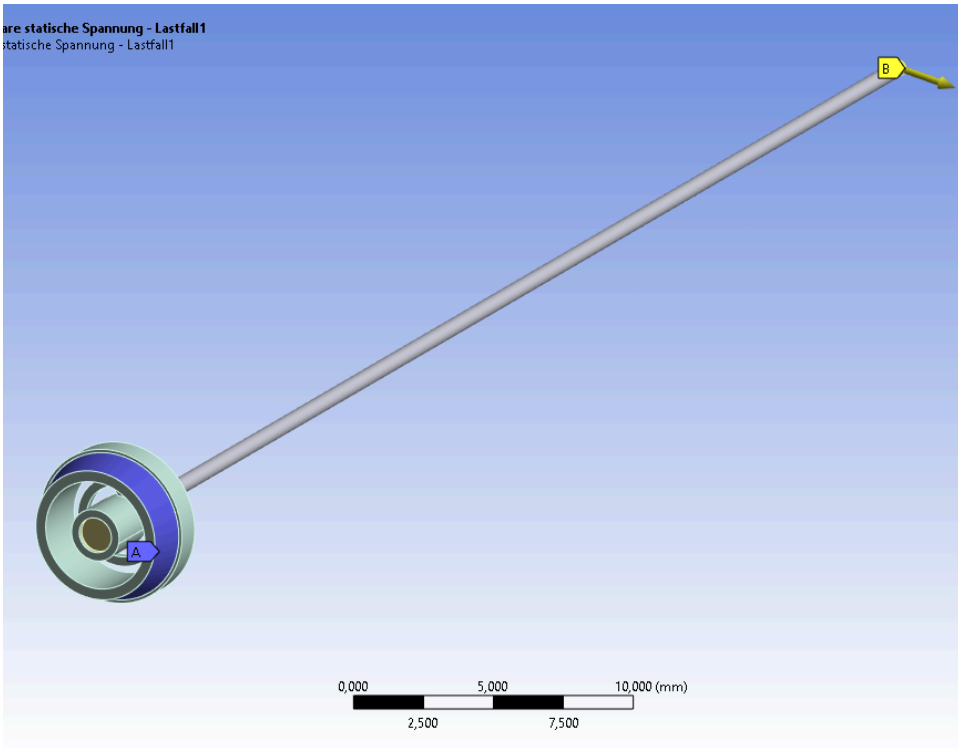

Figure 3.4: Boundary conditions and geometry of mechanical simulation. The yellow arrow labeled “B” represents a displacement at the whisker tip, while the blue area labeled “A” illustrates a fixed support at the head mounting.

The explicit boundary conditions applied to the resulting mesh were defined by one fixation and one displacement, which are depicted along with the geometry in Figure 3.4. A fixed support was applied to the module's sloping surface that interfaces with the head, mimicking its mounted state. To accurately simulate operational conditions, displacement of 10 mm was applied at the whisker's tip in four distinct load steps. These steps - displacing the tip in the positive x, positive y, negative x, and negative y directions - were designed to assess the module's isotropic behavior. By evaluating the module's response to these varied directional forces, this aimed to ensure that the sensor system would exhibit nearly isotropic behavior, crucial for accurately detecting and interpreting external stimuli.
The simulation automatically incorporated bounded contact between the touching areas at the geometries. So fixing the whisker as well as the magnet to the whisker module, this ensured a realistic interaction within the simulated environment.

For evaluation, the solution types von-Mises Stress, elastic strain and total deformation are added. Another critical factor affecting the magnetic field at the sensor's position is the rotational angle of the magnet. To assess this, the flat surface of the magnet pointing inside the head was defined as an external point to assign a variable to it. Using a ANSYS

Parametric Design Language script, the angles were measured between the original position and the deformed state, and then extracted to the graphic user interface.

3.3.2 Magnetic Analyzation

For the magnetic field analysis within the artificial whisker system, a magnetostatic analysis was performed. Two separate simulations were prepared fo comparing the magnetic flux values at the sensor's sensitivity position in two distinct states:

Original Geometry: The original CAD models were imported without any changes, including whisker module, magnet and Hall-Sensor Sensitivity Area.
Deformed State: The altered geometry, resulting from displacement in the negative y direction of the mechanical FEA, was also imported.

The simulation environment emulated the physical setup, with air characterized by an "Isotropic Relative Permeability" of 1 to simulate a non-magnetic influence. The sensor area was modeled with the magnetic properties of steel, given an "Isotropic Relative Permeability" of 1500, to accurately reflect its sensitivity to magnetic fields.
For the magnet, material properties were directly taken from its datasheet, with a Remanence (Br) of 1.45 T and a Coercive field strength (HcB) of 780 kA/m. These values were inputted into ANSYS as "Residual Induction" and "Coercive Force," respectively. The magnetic field of each simulation was carefully assigned to match the actual magnets orientation, ensuring the magnetic field's direction was correctly represented in both cases.

The analysis specifically focused on the total magnetic flux along with its individual components along the x, y, and z axes. This enabled a detailed comparison between the magnetic flux values of the system in its original position and the deformed state, as induced by the mechanical simulation.

A special point of interest here is the sensor's sensitivity area. It is important to note the strong assumption made regarding the precise alignment of the sensor's sensitivity area with the magnet when comparing simulated and real values. While the assumption of perfect alignment facilitates a clearer simulation environment, slight displacements or angular deviations in practice could lead to significant changes in the magnetic flux values. However, the fundamental mechanics modeled in the simulation remain valid, and the findings provide valuable insights into the system's magnetic behavior. The closer the alignment in the actual system, the more realistic the simulation results will be.

When comparing the measured magnetic flux components with all the simulated values of all nodes in the FE model, the node with being the closest to the real values can be used as an approximation for the relative position of the actual sensor's position.
Subsequently, that estimated position can be checked in the deformed states magnetic FEA and identified how the real sensor values should behave in this case.
This allows for an analysis of the sensor's position influence on the measured values.

3.4 Obstacle Avoidance

To employ Deep Reinforcement Learning for obstacle avoidance with the modified NeRmo robot utilizing whiskers, several steps were required. Initially, the physics simulator needed setup, integrating the whiskers into the virtual robot's design. Subsequently, a simulation environment interface was designed for DRL, encompassing the definition of agent dynamics, such as interaction mechanisms and reward systems. Ultimately, the integration of the trained neural network into the physical robot was conducted.

3.4.1 Physics Simulator

After having the final hardware, the new design of the mouse robot had to be implemented into a simulation to virtually conduct DRL with it. For this several physics simulators exist, but the researchers from TU Munich already implemented the original NeRmo model in mujoco, and thankfully provided it, so this one the physics simulator of choice.

Mujoco (short for Multi Joint Contact) is an open source physics engine that represents a good choice for researchers of robotics and other fields due to its fast and accurate simulation (Todorov et al. 2012). Especially in respect to DRL it is a very popular choice, having also some pre-built models like a humanoid or a quadrupedal robot “Ant”. The widely used Gymnasium (Towers et al. 2023) environment also contains prebuilt environments based on mujoco.
For building a model, one can create new or modify existing .xml files that contain attributes in the specific MJCF modeling language.

whisker modeling
The existing model already contained the mechanical components for walking but lacked the whiskers so they had to be added.

During this study, since MuJoCo did not support soft materials, the bending properties of the whiskers were simulated using a sequence of rigid elements, which corresponds to the RBM approach (Huet et al. 2015, Quist et al. 2014). A suitable method for this was to use a python file to modify the existing model to add the four whiskers. Ten thin cylinders in series that are connected with hinge joints having 2 rotational Degrees of Freedom, the resulting mujoco model of this can be seen in Figure 3.5. These connections were assigned with spring damper properties to mimic a deformable structure. Each joint was assigned with a stiffness and a damping parameter to ensure that, after deflection, the whisker returns to its initial position via the modeled spring mechanism. The inclusion of damping characteristics serves to dissipate energy, thereby preventing the oscillations that would occur with a spring alone, without a damper.

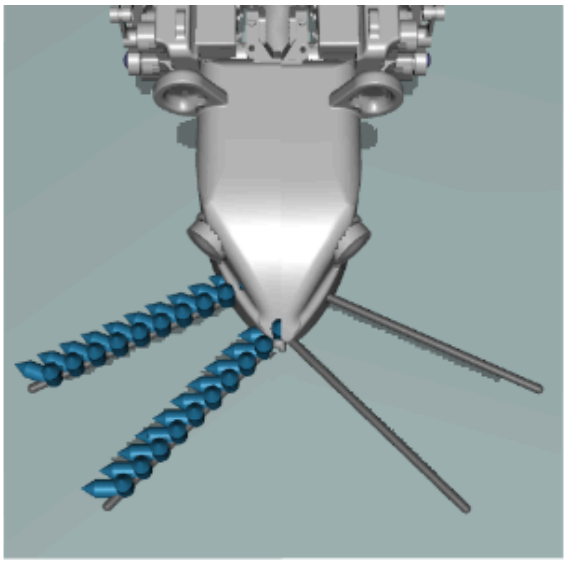

Figure 3.5: modelling of the whiskers in mujoco by sequentially connecting cylinders through joints with stiffness and damping properties.

To determine optimal values for the stiffness and damping parameters, a quick grid search was conducted. The goal was for the robot's whiskers to quickly return to their original position after deflection. Although literature review revealed models for the mechanical properties of whiskers (Huet et al. 2015, Quist et al. 2014), including their implementation approach in simulations, none were compatible with Mujoco, the physics engine used for simulation in this study. Adapting these parameters to Mujoco proved complex due to differing computation techniques.

While incorporating these models could have been beneficial for NeRmo, it was complicated by the need to convert parameters to Mujoco. Moreover, even if implemented, these models

would not fully capture the mechanics, as NeRmo's hardware differs from real mice, utilizing a constant 0.6mm nylon thread instead of a linear truncated cone.

Consequently, experimentation through trial and error became necessary. A Python script was employed to vary damping and stiffness parameters, with results analyzed visually to gauge effectiveness.

Selection of Simulation Time Step

In configuring the Mujoco simulation for our study, a critical decision involved determining the appropriate simulation time step. The selection of time step size is a balancing act between achieving high resolution (and thus accuracy) in the simulation and managing computational resource demands.

A smaller time step size enhances the simulation's resolution, contributing to greater accuracy in the representation of physical interactions and dynamics. However, this increased resolution comes at the cost of higher computational resource requirements. Specifically, simulating the same amount of virtual time with smaller time steps necessitates a greater number of calculations, thereby extending the real time required for computation. This aspect was a significant consideration, given the limitations of our available computational resources.

Conversely, increasing the time step size to reduce computational load presented its own set of challenges. Larger time steps can adversely affect the stability of the simulation. One notable issue is the delayed detection of collisions. For instance, if a collision with a virtual wall is detected too late, the reaction force exerted on the object (in our case, the robotic mouse) is significantly higher. This delayed response often resulted in the mouse being abruptly catapulted in the opposite direction, an outcome that was detrimental to the learning objectives of the simulation.

After a thorough process of trial and error, it was determined that a time step of 0.005 seconds represented the most effective compromise. This time step size has an optimal balance between maintaining the speed of the simulation and ensuring its stability, particularly in the context of collision detection and response.

3.4.2 Deep Reinforcement Learning

In this research, a DRL framework for a sophisticated multi-worker system was developed, with the aim to generate obstacle avoidance with a robotic mouse agent.

stable baselines3

Stable-Baselines3 is a Python library developed on the PyTorch framework. It is recognized in the field for its dependable and user-friendly implementations of various RL algorithms. The library's primary objective is to offer a more robust and efficient framework, which is pivotal for both research and practical applications in RL. A distinguishing feature of Stable-Baselines3 is its extensive benchmarking against other RL codebases, highlighting its reliability and effectiveness (Raffin et al., 2021).

Stable-Baselines3 provides comprehensive scripts for training and evaluating agents, along with tools for hyperparameter tuning. Its open-source nature, coupled with its user-friendly and easily extendable design, makes it a preferred choice within the scientific community for RL applications.

Gym-like custom environment

The use of stable-baselines3 requires the use of a Gym-like methodology in the environment, following the standards set by OpenAI's Gym framework. This ensures compatibility, making the environment interface with the software library in a standardized manner. In the implementation the following aspects are crucial for this.

The environment was designed, taking into account the high temporal resolution of the physics engine used. Given this high resolution, conventional equal-length time steps were unsuitable for an effective learning process. To address this, we adopted a novel approach where one agent step corresponded to 10 steps of the physics engine. This equivalence was calculated to be 0.05 seconds.

The custom environment inherits from gym.Env, a core component of the Gym library. This inheritance brings with it a structured framework requiring the implementation of key methods such as step and reset.

At initialization a specific action space and observation space was defined as described in the previous chapters, but specifically using the “spaces” module. The action space, represented by gym.spaces.Discrete(num_actions), where nu_actions specifies the number of discrete actions available to the agent. In contrast, the observation space is established

using gym.spaces.Box(low=-np.inf, high=np.inf, shape=(num_states*num_frames,)), where 'low' and 'high' define the obligatory lower and upper bounds of the observation space, but also allowing infinite for unbound values. 'shape' determines the dimensionality of the observations made in one frame multiplied by the number of frames observed.

By aligning the custom environment with the Gym-like structure, it facilitated a consistent and standardized approach to reinforcement learning, allowing for more straightforward integration of DRL and potentially enhancing the reproducibility of the experiments of this study.

State Space

The state space design was a fundamental element of this DRL strategy. It was crucial to ensure that the state mirrored the parameters available to the actual robot. This alignment was important to ensure the transferability of our simulation findings to real-world applications. The state for each frame included four continuous whisker angles in horizontal plane, reflecting the tactile feedback mechanism of the robot.

Additionally, a discrete value representing the last action taken by the robot was included, which leads to five observations for each frame. The inclusion of the last action variable was pivotal in stabilizing the robot's movements. Without it, we observed a tendency for the robot to exhibit erratic behavior, often leading to it flipping over – an undesirable state in real-world applications.

This configuration created a partially observable state, posing unique challenges in terms of training and algorithmic efficiency.

The total state space resulted in merging observations of different frames. The nuances of addressing these challenges are further elaborated in the “Training Enhancement” section, particularly under “Frame Stacking and Frame Skipping”.

Action Space

To simplify the robot's movement mechanics an intermediate control layer was developed by translating continuous motor command values into discrete walking actions. This translation was not arbitrary but was instead based on a predetermined gait pattern. This gait pattern was a result of the researchers at TU Munich and was provided in a leg model that inputted the phase of the gait cycle and returned the next motor commands as explained previously. The model allowed for the setting the frequency of leg movements for adjusting speed. For example, reversing the direction of leg movements enabled the robot to walk backwards. The spine's movements were also intricately programmed into these patterns. When moving

straight, the spine oscillated at predetermined angles, whereas for turning maneuvers, it would bend to a fixed position in the directional change.

To facilitate understanding of the model's actions, continuous motor command values were translated into seven discrete walking actions based on a predetermined gait pattern. Below is the mapping of these actions:

1. Forward Straight
2. Forward Left
3. Forward Right
4. Backward Straight
5. Backward Left
6. Backward Right
7. Stand Still

This translation ensures that the robotic mouse can navigate effectively, by using discrete actions. This range of movements was essential to ensure that the robotic mouse could navigate effectively in all necessary directions.

Reward Function

The development of an appropriate reward function is a crucial aspect of the DRL approach. This function's primary objective was to guide the agent towards behaviors that avoided obstacles using its whiskers, without resulting in collisions. To achieve this, the reward function emphasized forward motion, particularly in scenarios devoid of obstacles, as this was a key indicator of successful navigation. The function calculated rewards based on the horizontal velocity of the robot, reinforcing the importance of swift, unimpeded movement.

A nuanced penalty system was also integrated into the reward function. Any collision that occurred, excluding those involving whisker contacts, resulted in a negative score of -0.01. This penalty was designed to discourage behaviors leading to non-whisker collisions. Moreover, scenarios where the robot turned upside down - a significantly undesirable state - were heavily penalized with a score of -1. This heavy penalty was strategically placed to minimize the occurrence of such states, ensuring that the robot remained upright and functional for the majority of its operational time.
The traveled distance per time step was maximally 0.002.

Event	Reward	Done
Collision	-0.01	False
Upside down	-1	True
Moving	traveled distance	False

Table 3: Reward Function for Obstacle Avoidance.

Training Enhancements

Early Termination

One of the innovative strategies implemented to enhance the efficiency of our DRL process was the concept of early termination. This approach was designed to address situations where the robot reached an irrecoverable state, such as being flipped upside down. By monitoring the angle between the horizontal plane and the mouse body, we were able to establish a threshold beyond which continued operation was deemed non-productive. Specifically, if this angle exceeded 45 degrees, we programmed the simulation to terminate the current episode by setting the 'done' variable to True. This early termination served two crucial purposes: firstly, it conserved computational resources by preventing the system from expending energy on scenarios where the robot was incapable of meaningful interaction with its environment. Secondly, it refined the learning process by preventing the agent from accumulating experiences from actions that had no impact on its state, thus maintaining the focus on relevant and impactful learning experiences.

Frame Stacking and Frame Skipping

Given the partially observable nature of our simulation environment, it was essential to provide the agent with some form of temporal context. To achieve this, frame stacking was selected as it is relatively simple compared to more complex solutions like Long Short Term Memory networks. Frame stacking involved consolidating the state information from multiple consecutive frames into a single, extended state. This approach allowed the agent to perceive a sequence of events over time, thus gaining a rudimentary form of temporal awareness. It was empirically determined that stacking ten frames struck provided a sufficient time horizon in this study, which differs from the original implementation from OpenAI in Space Invaders (Schulman et al., 2017).

Additionally, a technique was employed known as frame skipping to further enhance the temporal dimension of our state representation. This technique involved skipping a predetermined number of frames, in this work we used 9, before incorporating a new frame into the stack. By doing so, this extended the temporal window that each stacked state

represented to 5 seconds. Advancing directly to the tenth frame multiplies the time scale of one environment step of 0.05 by ten. With each observation consisting of ten frames, this approach ultimately spans a 5-second period per observed state, enhancing the ability to analyze extended sequences of events within a condensed computational framework. This longer window provided the agent with a more comprehensive view of the temporal dynamics within the environment, crucial for making informed navigation decisions based on past and present states.

Curriculum Learning

This approach also incorporated the concept of curriculum learning, a technique shown to be beneficial in complex learning scenarios (Narvecar et al. 2020).

Initially, the agent was introduced to a simple, unrestricted environment. This initial phase was critical for allowing the agent to learn the fundamentals of movement without the added complexity of navigating obstacles. After the agent visually seemed to exhibit effective locomotion in this simplified environment, more complexity was added to the learning task. The enhancement involved constraining the environment with walls, thus necessitating the use of whisker sensors for obstacle detection and avoidance. The learning environment was further complicated by introducing randomly placed boxes of varying sizes within the confined space. These boxes served as unpredictable obstacles, providing a more realistic and challenging navigation scenario. This box placing was dynamically altered every 10,000 time steps, ensuring that the agent was continuously exposed to new and varied learning experiences. This progressive increase in task complexity was crucial in developing a robust and adaptable learning algorithm capable of handling real-world navigation scenarios.

Selection of PPO algorithm

In the exploration of reinforcement learning algorithms, PPO was selected due to its notable advantages that align with the objectives of this study. This section elaborates on the rationale behind choosing PPO, highlighting its flexibility, user-friendliness, and accessibility, followed by a discussion on the specific hyperparameters adopted for the experiments.

Flexibility in Action Space: PPO's capability to handle both continuous and discrete actions positions it as a versatile choice for varied application scenarios. Initially, actions were discretized to expedite the learning process by simplifying the action space. This decision was based on the premise that reduced complexity leads to faster learning. However, for future experiments, it may prove insightful to bypass this discretization, thereby directly mapping actions to motor commands. Such an approach could unveil the impacts of a continuous action space on learning efficiency and outcomes.

User-Friendliness: PPO is distinguished by its ease of use, attributed to its generally robust default behaviors and a minimal set of hyperparameters. This simplicity facilitates experimentation, reducing the complexity inherent in tuning and testing, thereby making PPO an attractive option for both novice and experienced researchers in the field.

Ready-Made Implementations: The availability of PPO implementations in libraries such as Stable Baselines 3 mitigates the need for development from scratch. This accessibility not only saves significant time but also minimizes the risk of errors that might arise from custom implementation efforts.

For the purposes of this study, the PPO algorithm was employed with minimal deviations from its default implementation in stablebaselines3. Notable adjustments included the enlargement of the hidden network size to 256x256 and the specification of the n_step update rate to 128. These modifications were anticipated to optimize the learning process without substantially departing from the proven efficacy of PPO's standard configuration.

Robustness Testing

The resilience and adaptability of the neurorobotic mouse, henceforth referred to as NeRmo, were rigorously evaluated through a series of robustness tests. These tests were meticulously designed to assess NeRmo’s performance under a variety of operational conditions that diverged from those encountered during its training phase. The essence of these evaluations was to simulate real-world inconsistencies, such as variations in walking speed and sensor accuracy, to gauge the neural network's capacity for generalization in dynamic environments. A comprehensive set of 100 episodes was conducted to ensure the reliability and statistical significance of the performance data collected. Additionally, the length of each episode was carefully monitored as a measure to identify instances of early termination, providing an insightful metric into the system's robustness under varying conditions.

The evaluation encompassed several distinct scenarios, each aimed at challenging NeRmo in unique ways.

Initially, NeRmo was tested under baseline conditions, identical to those used during its training, to establish a control scenario for subsequent comparisons.

Variations in walking speed were explored by reducing the speed frequency by 20%, from a nominal setting of 1 to 0.8. This was followed by an adaptation of the frame skipping

technique to align the temporal frequency of inputs with those of the training conditions, aiming to mitigate the impact of reduced movement speed on performance.
The introduction of Gaussian probability noise to the joint sensors simulated sensor inaccuracies at two intensity levels, labeled as small and high noise conditions. This aspect of the testing aimed to uncover the neural network's robustness to variations in sensor input quality.
A scenario combining both reduced speed and the presence of small noise was also evaluated to ascertain the cumulative effect of multiple deviations from the training environment on NeRmo's operational efficacy.
The simulation environment and analytical tools utilized for these evaluations were chosen for their proven accuracy and reliability in replicating the operational dynamics of neurorobotic systems. Performance metrics, including the mean returns and standard deviations across the 100 episodes, were computed for each testing scenario, providing a quantitative foundation for assessing NeRmo’s robustness.

The recorded episode lengths, alongside the calculated performance metrics, offered a granular view into how each tested scenario impacted NerMo's ability to navigate and complete tasks effectively. These measurements were crucial for identifying conditions under which the robot experienced early terminations, shedding light on its adaptability limits and guiding future enhancements to its design and operational algorithms.

3.4.3 Robot Implementation

Figure 3.6 illustrates the communication and control flow between the robot and the laptop, responsible for the operational synchronization of the neurorobotic system. The robot, equipped with a Wi-Fi module, forms the core of the hardware interface, receiving motor commands to actuate the servos and concurrently transmitting sensor data back to the laptop.

Upon reception, the sensor data undergoes preprocessing by the laptop to format it appropriately for input into the neural network. The neural network, through its trained model, interprets this data and decides on the subsequent action. This action is then passed to the controller, which updates its configuration if necessary and calculates the current step phase for the leg model. The leg model translates the step phase into precise control signals that the robot's servos can execute to match the desired position.

As the control signals are dispatched back to the robot, completing the cycle, the system is primed to receive updated sensor information, enabling a continuous and dynamic feedback loop.

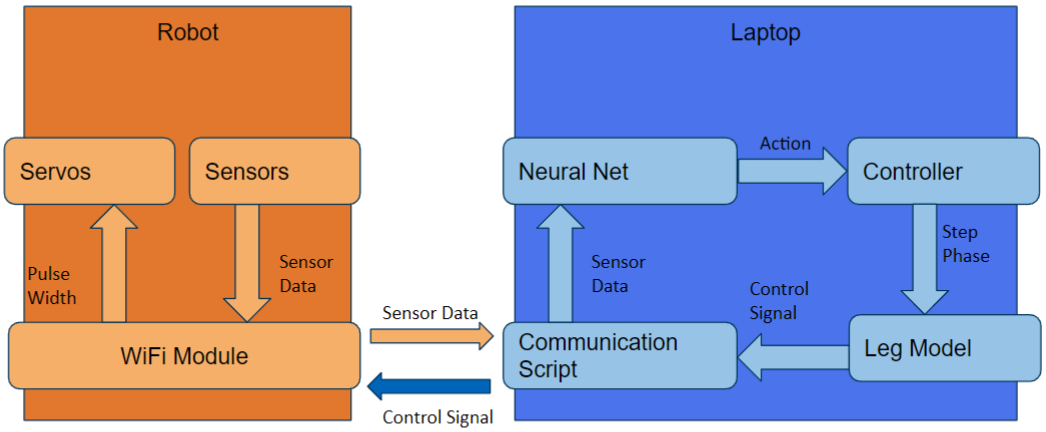

Figure 3.6: communication and control flow between the robot and the laptop.

Implementing the neural network model into the robot necessitated some modifications for a successful application. On one hand, the laptop-to-robot communication needs to happen in brief time increments to facilitate smooth and continuous motor operations, avoiding any perceptible jerky motions. On the other hand, this concerns aligning the framework with the training parameters specified in the simulation for achieving the desired behavior.
The requirement of a high communication rate between NeRmo and laptop was simple to implement. As the wifi communication provides fast data exchange, a time step of 0.025 seconds seemed to be reasonable and proved to lead to smooth movements.
As the environment steps with the defined frame skipping are 20 times larger, the communication steps are just executed five times more often.
Adjusting the raw magnetic flux data to be compatible with the neural network's input format involved a process of baseline correction and data simplification.

Baseline Correction: Upon powering up the robot, initial sensor readings were taken to establish a baseline for the undisturbed magnetic flux state. Subsequent sensor readings were then adjusted by subtracting this baseline value, effectively removing the offset and highlighting the deviations caused by whisker deflections.
Simplification and Clipping: The adjusted flux readings from the x, y, and z components were further processed by converting them to absolute values and summing them to produce a singular metric per sensor. This approach, while not directly translating to whisker angles, was considered to be sufficient for the context of static obstacle avoidance, where the magnitude of change in magnetic flux serves as an proportional indicator for whisker

displacement. To ensure compatibility with the simulation's parameters and prevent the values from exploding, these summed values were clipped at a maximum threshold of 1 radian, which would correspond to about 57 degrees.

4 Experimental Results and Evaluation

This chapter will present the outcomes of the sensor development and illustrate its applicability for obstacle avoidance of a mouse robot in both simulated and real environments. Initially, the chapter examines and illustrates the findings from experiments on the method of tactile hair perception employing a magnetic system, offering a proof of concept. Subsequently, the results of the FE simulation are showcased, providing deeper insights into the mechanical dynamics of the whisker deflection and the consequential magnetic variations at the sensor's location upon deflection. This segment also includes a comparison between the anticipated values at the optimal sensor position and the actual sensor readings. Following this, the chapter delves into the exploration of the DRL results for utilizing the whisker for obstacle avoidance in a simulation. This examination encompasses an analysis of the training process, the overall action selection distribution, and a qualitative review of a specific trial. Finally, the chapter details the real robot's behavior when using the neural network for obstacle avoidance, offering a comprehensive overview of its operational efficacy.

4.1 Comprehensive Analysis of the Artificial Whisker System

In this section, the focus shifts to a detailed examination of the artificial whisker system's performance and its integration into the mouse robot. First, the initial testing results are explored, where the fundamental concept of tactile perception through a magnetic system is assessed. This evaluation provides the groundwork for understanding the system's capabilities and limitations.

The narrative then progresses to a thorough mechanical simulation of the whisker module. This simulation offers valuable insights into the mechanical behaviors and responses of the whisker system under various conditions, allowing for a deeper understanding of its operational dynamics and the factors influencing its performance.

Subsequently, the magnetic impact at the sensor's position as a result of mechanical deformation of the whisker system is examined. This analysis explores the complex magnetic responses and changes resulting from the whisker's deflection, providing a detailed account of how these interactions are harnessed for tactile sensing. The section concludes with an examination of the magnetic values observed when the system is implemented in a robotic context. This final part synthesizes the theoretical and empirical

findings, showcasing the practical implications of the magnetic interactions and their significance in enhancing the robot's sensory capabilities.

Throughout this exploration, the emphasis is placed on the systematic investigation and empirical validation of the artificial whisker system, aiming to offer a comprehensive understanding of its design, functionality, and application in robotic systems.

4.1.1 Testing Results of Concept Evaluation

The series of tests designed to evaluate the system's sensitivity and repeatability involved manually deflecting the whisker in specific orientations along the X and Y axes, as outlined in the Methods section.

Before examining the effects of deflection, it's essential to establish the baseline sensor values in the whisker's undeflected state. In this state, the Z value is the highest, followed by the Y, with the X value being the lowest, establishing a clear reference for assessing the impact of directional deflections.

These deflection tests aimed to validate the system's capability to consistently produce comparable sensor responses under controlled deflections. As illustrated in Figure 4.1, the results confirmed the system’s high repeatability. Specifically, sensor data indicated predictable patterns of response:

- Deflection in the positive X direction increased the X value, with corresponding decreases in Y and Z values.
- Deflection in the negative X direction decreased the X value, with increases in Y and Z values.
- Positive Y direction movement increased the Y value, while X and Z values decreased.
- Negative Y direction movement decreased the Y value, with increases in X and Z values.

Given the broad nature of these validation tests, a comprehensive quantitative analysis of the values is beyond the scope of this initial assessment. The primary aim at this stage is to verify the system's consistent performance across different directions, a fundamental requirement for a sophisticated whisker system. Detailed quantitative investigations are done in further sections, using the actual whisker module.

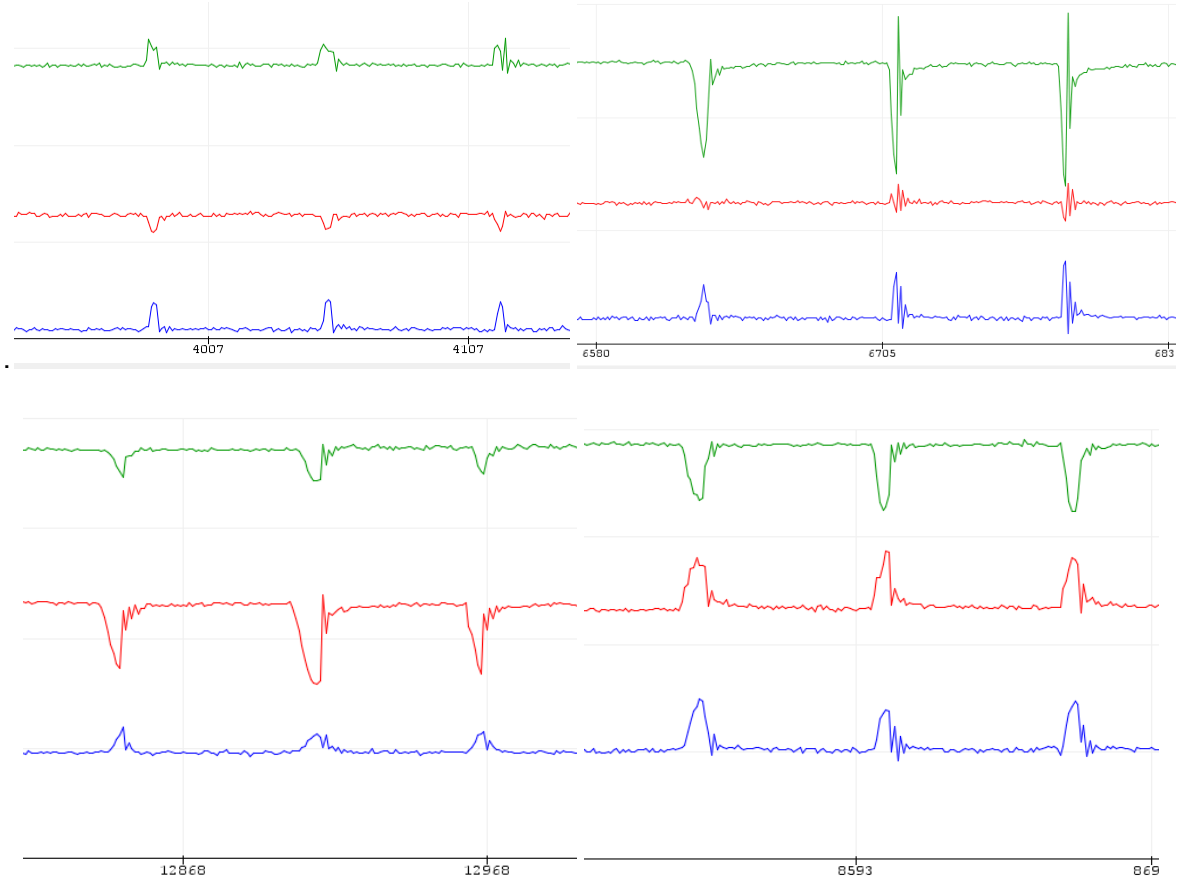

Figure 4.1: Results of the repeatability test, illustrating whisker movements in different directions with respect to the x and y axes. The first row depicts deflections along the x-axis, while the second row focuses on the y-axis. Columns are organized by directionality, with the first column representing positive displacement and the second showcasing negative displacement. The graph employs color coding to denote magnetic flux components: blue for the x component, red for the y component, and green for the z component.

4.1.2 Mechanical Simulation of the Whisker Module

For examining the mechanical response of the whisker system to deflection of the tactile hair, a mechanical FEA is conducted.

In this context, the whisker module, magnet, and whisker hair were modeled with bounded contact to simulate fixed connections between the components. A fixed support was applied to the module's sloping surface, representing its mounted state within the head, as visualized in the “Methods and Implementation” chapter under the section “Optimization with mechanical Finite Element Analysis” in Figure 3.4 where this surface is highlighted in blue (labeled 'A'). The displacement applied at the tip of the whisker, indicated by a yellow arrow

(labeled 'B'), facilitated the examination of the module's response to impacts on the whisker from various directions.

Through the implementation of four distinct load steps, displacements of 10mm were applied to the whisker tip in positive x, positive y, negative x, and negative y directions.

For reference, Figure 4.2 illustrates the orientation of the whisker module within the coordinate system, viewed from the sensor side.

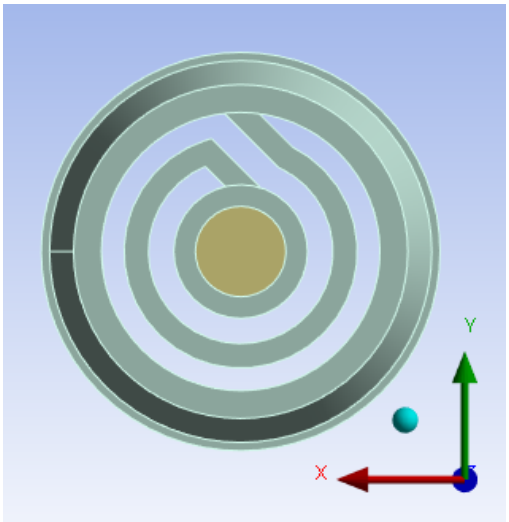

Figure 4.2: Orientation of Whisker module in coordinate system, viewed from sensor side inside the head.

The analysis focused on von-Mises stress, elastic strain, total deformation, and the resulting rotation angles of the whisker, representing the logical progression from applied forces to resultant structural responses.

Von-Mises Stress

Looking at the Von-Mises Stress, the significant values are all located at the whisker module along the circular structure and its connections. The maximal values all occur at the flexible holding arm at the edge of the connection to the inner linear arm leading in the circular arm. Here the values are highest at step 1 and 3 with 0.24 MPa, while having 0.22 MPa at step 2 and 4. It is worth mentioning that along the holding structure the stress is distributed quite evenly, varying between about 0.1 MPa and the already mentioned maximal value, which can be seen if only displaying values over this threshold seen in Figure 4.3. Looking at the whisker hair here only at the root is visible stress, always oriented toward the applied displacement.

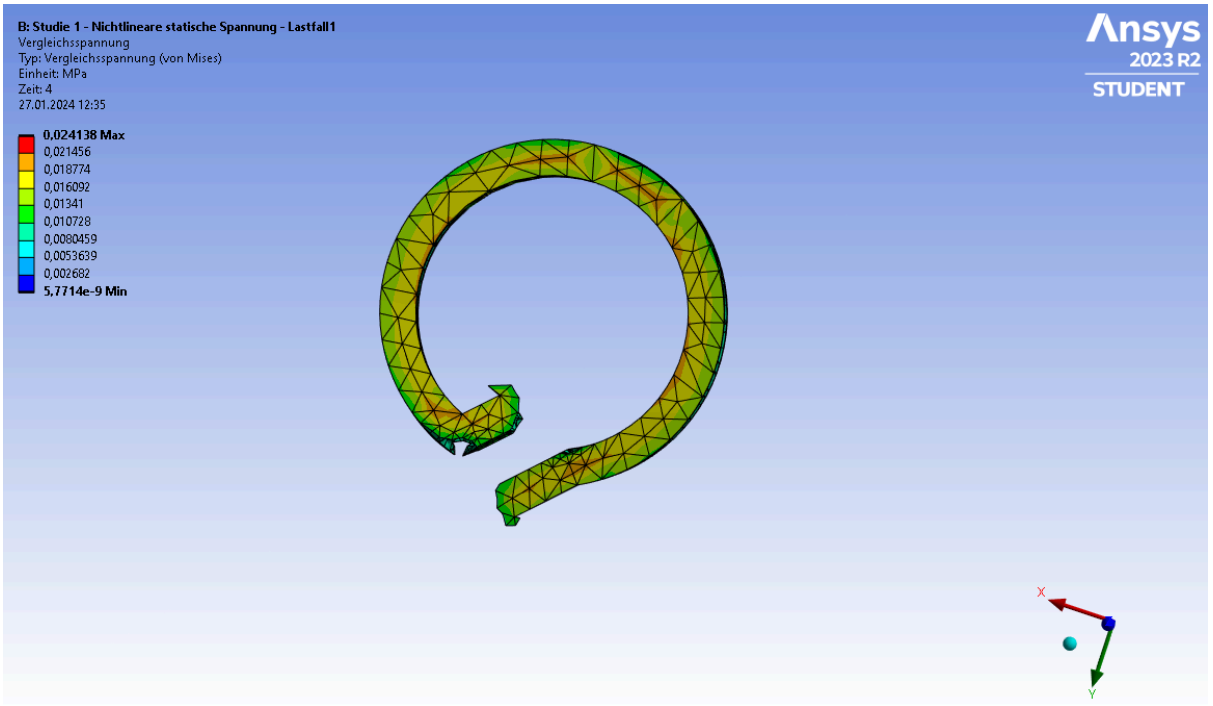

Figure 4.3: Nonlinear static tension of the whisker module outside. In this visualization only tensions over 0.01 MPa are shown.

Elastic strain

The elastic strain is very similar to the von-mises stress as a cause when only looking at one material. Also here the whisker and the outer ring of the whisker module has values at around 0%. The whisker module is especially strained at the flexible bearing arm, having its highest value again on the inner edge of the connection between inner linear arm and circular structure. Here the maximum is at 0.37% while the rest of this thin mechanism mainly experiences values between 0.2% and 0.37%.

Total Deformation

The overall system had in all loading steps a maximal deformation of 10 mm that occurred at the tip due to the loading condition. From there it decreased nearly linearly along the whisker. If only the whisker module is looked at, here the most deformation always occurred on the inside directed magnet surface, being at about 0.38 mm at the maximum at step 3 and 0.37 at the other steps. Here, it is observable that the deformation raises along the magnet holding cylinder the further away it is from the flexible bearing. This circular holding structure visibly experiences most deformation in the first half of the circular arm, with the exact location depending on the displacement direction. Here also it is visible that the further

away from the center where the whisker hair roots in the module, the higher is the force in the direction of displacement, which determines the rotational axis.

Magnet Deflection Angle

The angular deviation between the magnet's original and deformed orientations was remarkably consistent across all loading steps. Notably, the most significant rotational movement occurred perpendicular to the axis of applied force. For instance, when the force was applied in the x-direction, the predominant rotation was around the y-axis, and vice versa. The rotation around the axis of movement was minimal, registering negligible values below 0.0 degrees.

The absolute angular displacement consistently approached 9.5 degrees around the primary axis of rotation corresponding to the direction of applied force. For instance, when the force was applied in the x-direction, significant rotation was observed around the y-axis. Additionally, there was a slight but noticeable incremental rotation about the z-axis with each change in the direction of force application. Initially, this z-axis rotation was minimal at 0.0 degrees when the force was applied in the x-direction. It increased to 0.8 degrees with force applied in the y-direction, reached 1.6 degrees with force in the -x direction, and culminated at 2.3 degrees when the force was applied in the -y direction. All described values are presented in Table 4.1.

deflection direction	X-Angle (degrees)	Y-Angle (degrees)	Z-Angle (degrees)
x	0.0	**9.5**	0.0
y	**-9.5**	0.0	0.8
-x	0.0	**-9.5**	1.6
-y	**9.5**	0.0	2.3

Table 4.1: resulting rotational degrees of the magnet for displacement of 10mm at tip of whisker

After the simulation was conducted and analyzed, the deformed model of the -y deflection was exported as a stl file. This allowed us to further use it for the static magnetic simulation. Due to the property of nearly isotropy, it seemed reasonable to just do it for one loading case instead of doing the magnetic investigation four times.

4.1.3 Evaluating Magnetic Interactions within the Whisker System

The magnetostatic analysis yielded insights into the overall magnetic flux and its interaction with the deflected magnet, as captured in Figure 4.4. This figure illustrates the magnetic flux distribution within the system, comparing the original and deflected positions of the magnet in the ideal system. This simulation merges magnet, sensor and surrounding air to a simplified model.

Ideal Simulation Results

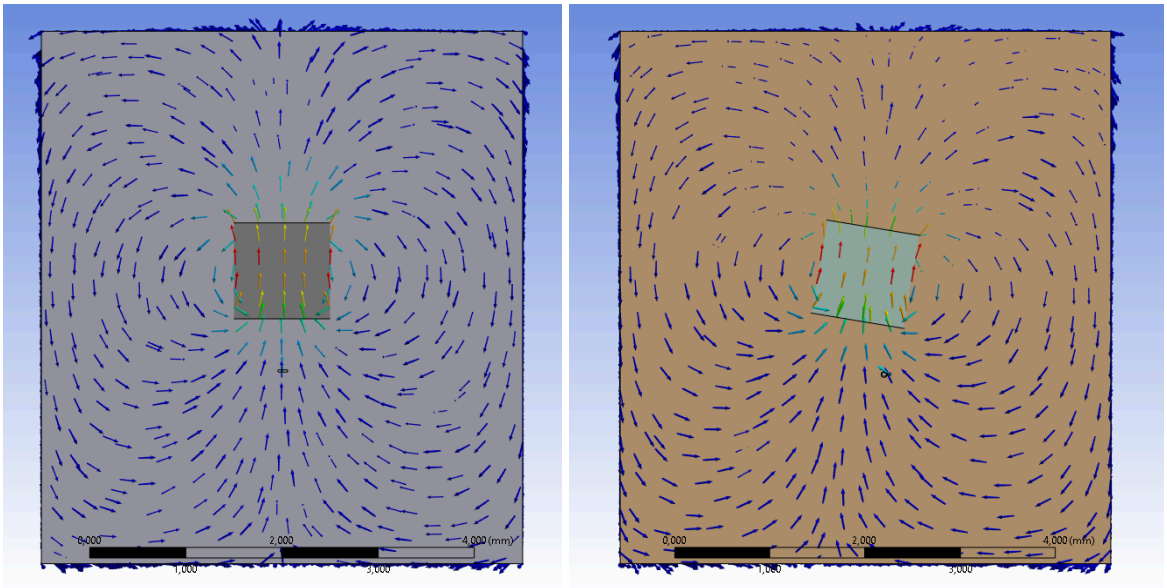

Figure 4.4: Overall magnetic flux at magnet-sensor system. Left: Origin position of magnet. Right: rotated magnet after whisker deflection in -y direction.

For analysis of the magnetic stimulation there was the overall magnetic flux selected as a metric to depict the magnetic field in the whole system. Additionally and more importantly for this study the directional magnetic flux in each of the three main axes of the sensor were calculated.

Undisturbed System

The overall magnetic flux ranged between 0 and 1039 mT with an average of 188 mT. It is important to mention that in this metric not only magnet and sensor is included but also the air, therefore the minimum values occur at the border of the system in the air only.

Regarding directional magnetic flux of the sensor's sensitivity area the values range because it is a volume and not a single point even if it is quite small. In x direction it is between -12

mT to 23 mT with an average of 6 mT. In y direction the values range between -1 and 12 mT with an averaged magnetic flux of 6 mT. The z directional flux has a small window of 149 to 150 mT with 150 mT as average.

After Whisker deflection in -y direction

In the deformed system the magnetic flux in x direction is between -132 mT and -64 mT with an average value of -98 mT. In the y direction the flux is bigger with a Minimum of 155 mT that goes to a maximum of 177 mT and its mean is 167 mT. For the z direction the smallest flux was at 129 mT and the highest at 134 mT with an average flux of 127 mT.

The most important values formerly mentioned are the average ones and a comparison between both cases can be seen in table 2. The range of the magnetic flux is caused because the sensor sensitivity is not a single point but a volume, so it is not constant in a magnetic field

magnetic flux component	undisturbed (mT)	After deflection (mT)
x	6	-98
y	6	167
z	150	127

Table 4.2: comparison of average magnetic flux at the sensor's sensitivity area of an idealized model in simulation. The deformed model used the results of the mechanical simulation with a displacement of -10 mm at the tip of the whisker in y direction.

4.1.4 Magnetic Values in Robot

To investigate if the theoretical findings of the previous section also apply in the actual mouse robot, the actual magnetic flux components at the whiskers were measured.

The default sensor values without touching the robot's whisker can be seen in Table 4.3. They are quite different from the simulated ones but as mentioned this is due to the violation of the assumption that the sensor is perfectly positioned behind the whisker module. Also the magnet's polarity makes a difference regarding positive or negative values.

Sensor number	Magnetic flux x (mT)	Magnetic flux y (mT)	Magnetic flux z (mT)
1	-2.99	-2.13	6.07
2	-2.61	6.01	-4.93
3	-3.6	-1.11	-9.83
4	14.65	13.46	-20.46

Table 4.3: Default Magnetic Flux Measurements in the Mouse Robot.

When defecting the whiskers about 1 cm from front changes in each value set can be detected. These events can be seen in a chronological sequence in Figure 4.5, where several observations can be made.

A very similar deflection causes different responses if only the individual components of the magnetic flux are compared. If they are inspected all together, it seems like the deflection amplitude of the values combined is about the same.

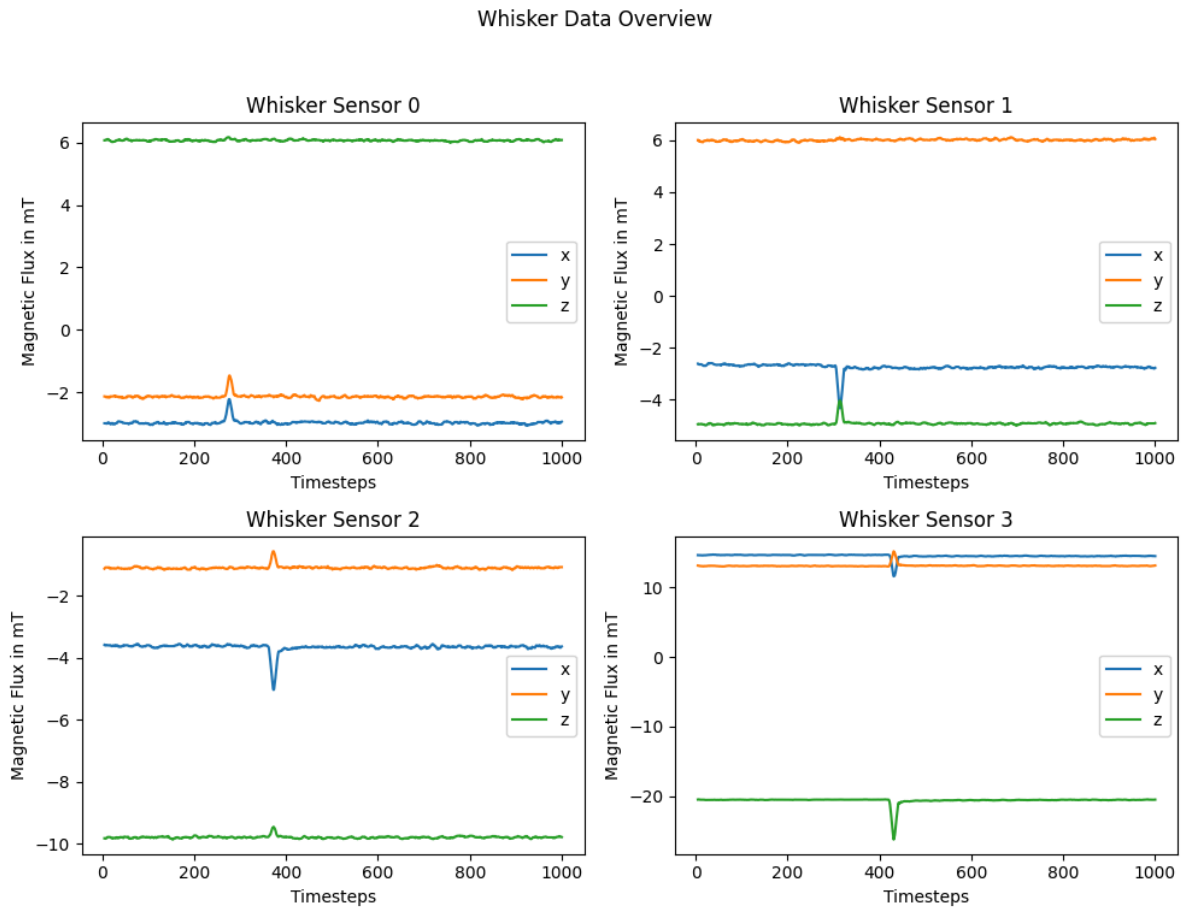

Figure 4.5: Magnetic Flux Data of the different whiskers when deflection of about 1 cm occurs from frontal direction. First column represents the left whiskers, the second one the right ones. The front and hind whiskers are in rows 1 and 2 respectively.

To further ensure the accuracy of our whisker sensor system, a validation test was conducted to assess sensor specificity and potential magnetic interference between whiskers. The test involved comparing magnetic field readings from a scenario where a single whisker was mounted with its magnet and sensor, against readings obtained with the full complement of whiskers installed. Results confirmed that individual sensors detected only their respective magnets' fields, without interference from adjacent sensors. This finding underscores the system's capability for precise and independent sensory input from each whisker, a critical aspect for the accurate replication of tactile sensing.

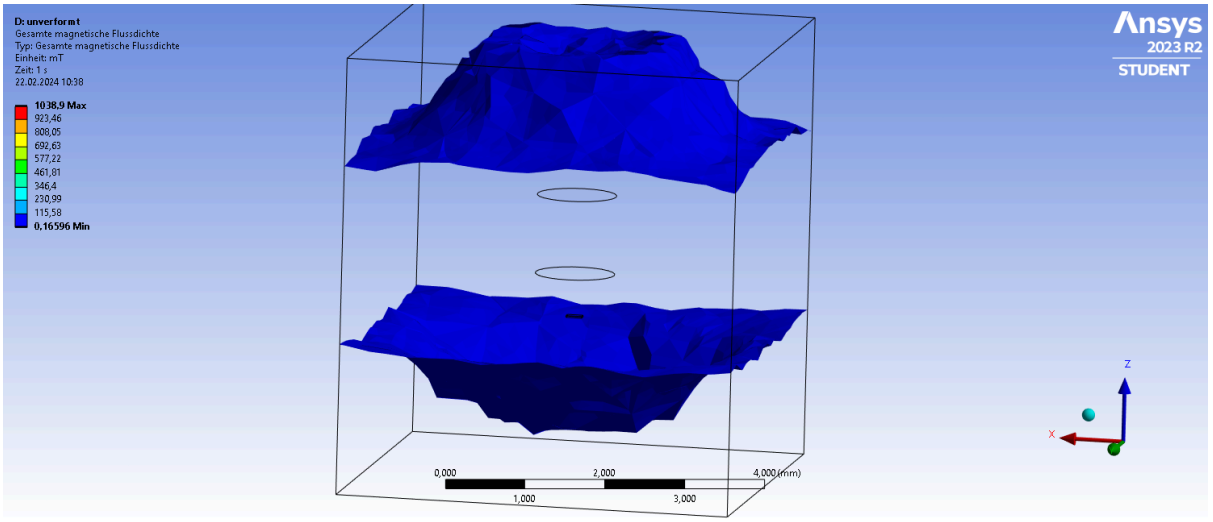

Figure 4.6: Visualization of non-deflected magnetic field where total flux is 7. This corresponds to the values measured in Sensor 1 - exact location can be determined by looking for the exact x,y,z component.

When looking at the specific values that are measured at sensor 1 and comparing it with the simulation results it seems to be possible to determine its position by having the components of the magnetic flux. For better visualization in Figure 4.6 only the locations are shown that have a total magnetic flux value of about 7 mT, which corresponds to the vector length calculated from the measured components. Now when using the specific components only two locations fulfill these conditions, while one can determine the sensor location by using the node that is on the inner side of the head, closer to the ideal position.
Due to the coarseness of the mesh, there is no node fulfilling having exactly the measured values but the least distance is an approximation. When looking at the node that has the least total deviation from the several components, this points to the coordinates -3.992 -1.993 4.349 mT instead of -2.99, -2.13, 6.07 mT, which is a total difference of 1.99 mT.
Comparing the coordinates, this would mean the position is shifted by [-1.61631032 -1.83085323 -0.42742389] mm from the ideal sensor position.

When looking at the magnetic simulation in the deflected case the magnetic flux components change to [-3.74, 1.12, 4.96] mT. This represents a large difference to the behavior resulting from the ideal case, so the difference of origin minus deflected case is [-0.75,-1.01,1.58] mT instead of [-104,161,23]mT.
Also for sensor 2 this comparison was done and again shows large differences to ideal behavior with a difference of [1.3, -3.8, -1] mT.

Scenario	Δ Magnetic Flux x (mT)	Δ Magnetic Flux y (mT)	Δ Magnetic Flux z (mT)
Ideal position	-104	161	23
Sensor 1	-0.75	-1.01	1.58
Sensor 2	1.3	-3.8	-1

Table 4.4: Difference of simulated deflection in -y direction to origin position from ideal position and the inverse calculated position from sensor 1 and 2.

4.2 Deep Reinforcement Learning

In this section, DRL is used to make the robotic mouse autonomous so it will not walk into obstacles that are in front of it and sensed by the whiskers.

4.2.1 Model Training

The training of the DRL model was methodically structured into two distinct phases to implement a curriculum learning approach. This strategy was designed to incrementally increase the complexity of the learning environment, thereby facilitating the gradual development of the robot mouse's navigational capabilities.

The metrics mean episode length and mean episode reward are depicted in Figure 4.7, while the gray line represents Phase One and the blue one Phase Two.

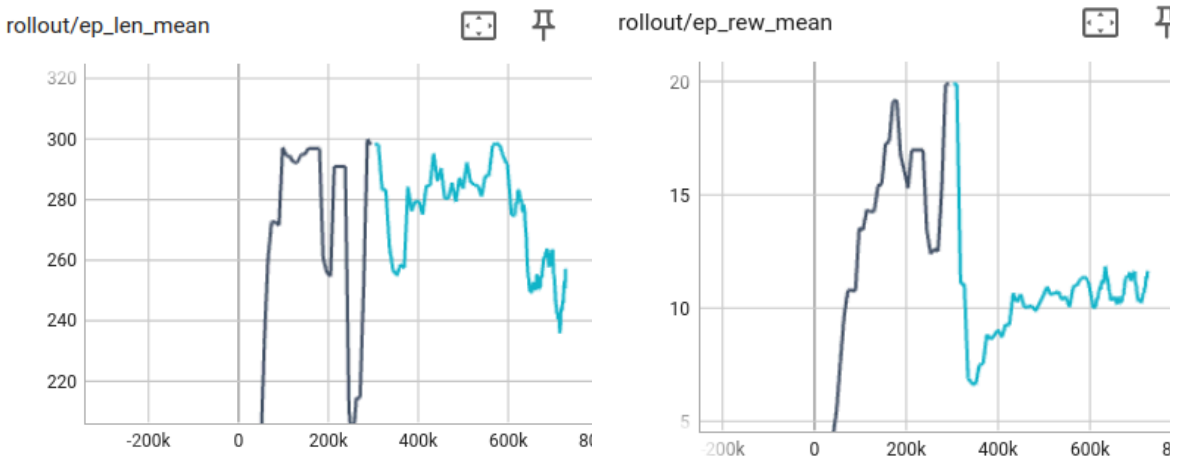

Figure 4.7: Tensorboard recording of learning process. Left: mean length of the episodes over time. Right: mean reward over the episodes.

Phase One: Learning Basic Locomotion

Initially, the model was trained in a simplified environment without obstacles, allowing the robot mouse to learn the fundamental mechanics of locomotion in pursuit of maximizing the reward. The absence of obstacles enabled the uninterrupted refinement of the walking behavior, with the goal of attaining the highest possible reward under these ideal conditions.

The training progressed until the model achieved an effective locomotion. This milestone was reached after approximately 300,000 timesteps and was achieved within a span of 20 minutes.

During the early training period, marked by the first 100,000 timesteps, a substantial increase in the mean episode length was observed, signaling rapid learning. The length approached the maximum attainable value, indicating that the robot mouse was learning to sustain walking behavior for longer durations. However, at around the 250,000 timestep mark, a notable decline in performance was recorded, where the mean episode length temporarily dropped to around 200. This setback was swiftly rectified, and the model soon recovered, consistently reaching the maximum episode length of 300.

In parallel with episode length, the mean episode reward exhibited a similar trajectory. A swift ascent in reward was noted, with the value climbing to approximately 19 by 170,000 timesteps. A subsequent performance dip reduced the mean reward to around 13, but the model demonstrated resilience and adaptability, eventually achieving the maximum reward of 20.

Phase Two: Obstacle Avoidance with Whiskers

In the second phase of training, the model, initialized with the learned parameters from the obstacle-free environment, was exposed to a more complex setting. The environment now included four stationary walls forming a rectangular constrained space and four boxes placed at random positions and orientations within this space.

Upon introducing the model to this new environment, a quantitative change in performance metrics was recorded. The mean episode length showed a decrease in the initial 50,000 timesteps, indicating an immediate impact of the environmental complexity on the model's performance. Subsequently, the episode length exhibited an increase, with values nearing the upper limit of 290 but after about 600,000 steps it declined to about 260.

Concurrently, the mean episode reward displayed a significant drop at the start of this phase, with values falling to an average of around 7. Over time, the mean reward displayed an upward trend, oscillating around an average value of approximately 10.5 by the end of the training period.

The training in this phase extended over a duration of 30 minutes, achieving about 750,000 timesteps.

4.2.2 Resulting Behavior

To analyze the behavior regarding chosen actions in a quantitative manner, 100 episodes were executed, and the resulting actions were recorded. The distribution of these actions is visualized in a normalized histogram in Figure 4.8, illustrating the preference of the model for certain movements over others.

Each action is represented by a number on the x-axis, corresponding to the set of movements described in the Methods section: 0 for moving straight forward, 1 for moving left forward, 2 for moving right forward and so forth up to 6, which represents moving backwards to the right. The y-axis represents the normalized frequency of each action, indicating how often the robotic mouse chose each movement during the trials.

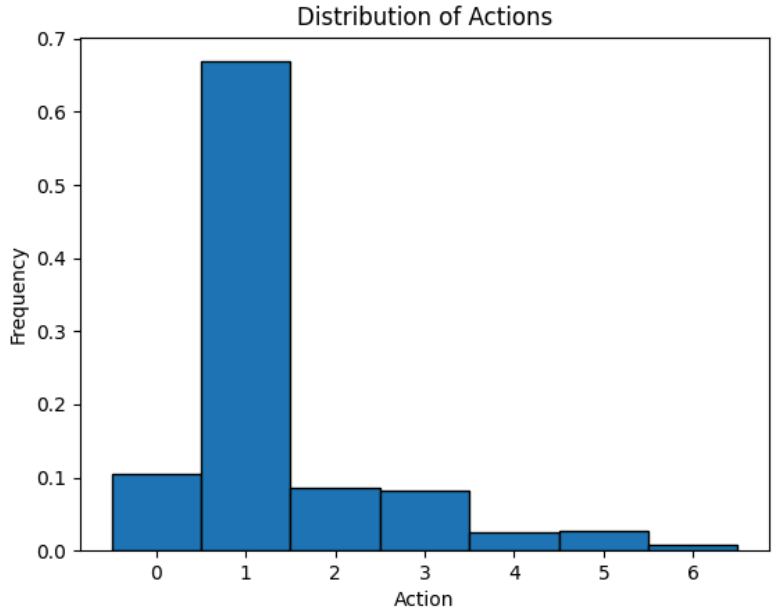

Figure 4.8: Histogram of Action Distribution. The discrete actions are obtained by 100 episodes and depicted in a normalized frequency distribution, facilitating a comprehensive understanding of the agent's action choices.

The histogram clearly illustrates that action '1', indicative of a forward-left movement, was the dominant choice, comprising approximately 68% of the total actions taken. In stark contrast, action '6', which corresponds to remaining stationary, was the least executed action.
Within the subset of forward movements, proceeding straight and steering right were selected with comparable frequency. For movements in the reverse direction, a straight backward trajectory was preferred over the left and right backward movements, which were chosen with equal frequency.

This distribution highlights the robot's preference for specific movements within the testing environment and provides insight into its decision-making process during navigation.

For a qualitative analysis of an explicit example several metrics are recorded to do a further investigation for understanding the learned dynamics of the model.
In Figure 4.9 these recordings are depicted having the walked path with collision points, highlighting whisker contact, illustrated as well as the chronological sequence of the whisker angles. Additionally the gained rewards can be seen in the same time line, showing the interaction of the environment and the agent.

The red dots represent a collision of a whisker with a wall, while the blue line shows the path the robot walked. The robot starts in the center of the figure.

The sequence of movement begins with the robot initiating a forward locomotion towards the left until an encounter with an obstacle triggers both right whiskers. Following a brief retreat and maneuvering phase, the robot resumes its leftward advance until it brushes against another barrier with its right whiskers. The robot attempts to navigate around the obstruction, this time engaging the left whiskers as well. After several adjustments, it successfully reorients and continues to traverse in a leftward circular pattern. Eventually, as it nears the maximum step count, the robot once more makes contact with a wall, engaging both right whiskers and one on the left, signaling the end of the trial before any further walking can take place.

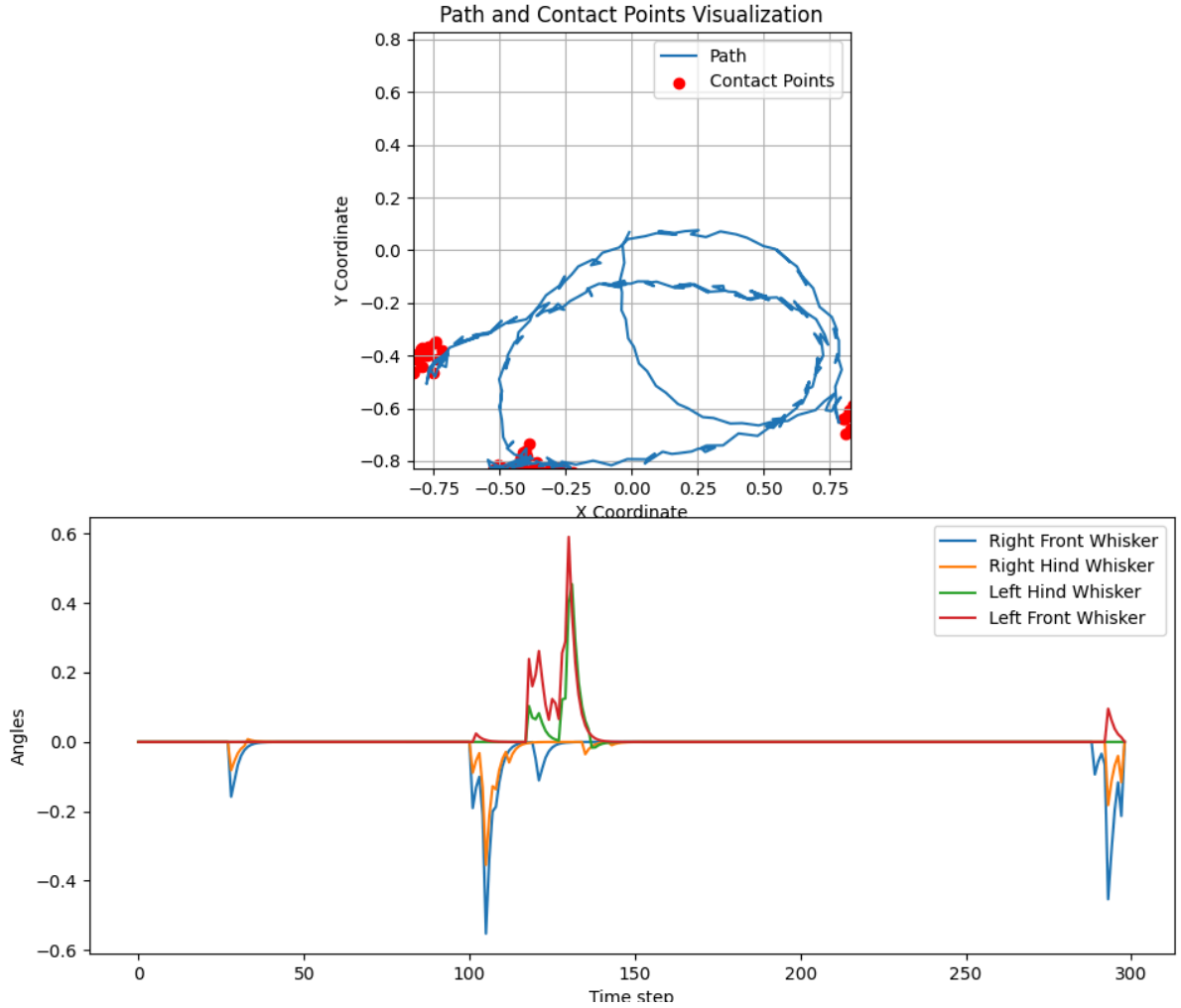

Figure 4.9: Walked path with highlighted whisker contact and recorded whiskers angles.

During the whole trail, only collisions on the boundaries of the restricted area occur, at three locations in total. The whiskers demonstrate deflections within a limited span, remaining within -0.6 to 0.6 radians, which translates to an angular range of approximately -34 to +34 degrees. Activation typically induces a change of less than 0.25 radians in absolute value, equivalent to roughly 14 degrees.

Notably, the right whiskers show more frequent activations than those on the left, and the front whiskers are activated more often than the hind ones.

4.2.3 Robustness Evaluation to Parameter Changes

The robustness of the trained neural net was critically assessed through a series of tests designed to understand its response to variations in operational parameters, specifically walking speed and sensor noise, for which it had not been explicitly trained. For consistency and to ensure that any differences in performance could be attributed to these variables,

obstacle characteristics such as size, position, and orientation were kept unchanged throughout these tests.
The findings from these evaluations are shown in Table 4.5 and offer revelations about potential problems for the robot implementation, because in reality there are always disturbing factors unlike in simulations.
These reveal the robustness and resilience of the neural network of the robot mouse in the face of unexpected environmental and operational changes.

Baseline Performance: Under the same conditions as those used during training, NerMo demonstrated a solid baseline performance, with a mean return of 10.1 and a standard deviation of 5.3, while showing a mean episode length at 263. This established a benchmark for assessing the impact of parameter changes on its operational efficiency.

Decreased Walking Speed: Reducing the walking speed frequency by 20% significantly impacted NerMo's performance. The evaluation yielded a mean reward value decreased to 3.8, and the standard deviation adjusted to 2.7 at a mean episode length of 221.

Sensor Noise:
The introduction of Gaussian probability noise was one in a slight and an extensive manner.
Applying small noise of 0.005, which corresponds to about 0.3°, to the whisker angles affected NeRmos performance minimally to 9.8 as mean reward with an standard deviation of 4.0. The mean episode length of 260 remains about the same to the baseline.
With increasing noise also the performance dropped drastically. When the 0.05, equivalent to about 3 degrees, the mean return reduces to 0.4 with a standard deviation of 0.9. Also the mean episode length is reduced to 117.

Decrease Walking Speed and Small Noise:
A composite test involving both reduced speed and the introduction of small noise further challenged NerMo, resulting in a mean return of 4.1 and a standard deviation of approximately 2.0. The mean episode length of this test was 234.

Case	Reward Mean +/- Std	Mean episode length
Training conditions	10.1 +/- 5.3	263
80% Speed	3.8 +/- 2.7	221
Small noise 0.3°	9.8 +/- 4.0	260
High noise 3°	0.4 +/- 0.9	117
80 % Speed decrease + small noise 0.3°	4.1 +/- 2.0	234

Table 4.5: Different testing scenarios of parameter changes and their influence on the behavior.

4.3 Robot Implementation

The integration of the neural network into the robotic mouse, NeRmo, served as a practical demonstration of the novel whisker sensor's applicability in real-world settings. This implementation was particularly challenging due to the transition from a simulated training environment to physical reality, necessitating advanced signal processing to align sensor outputs with the simulated whisker angles.

To ensure fidelity to the conditions under which the neural network was trained, NeRmo's operational parameters, such as walking speed and the interval between observations, were meticulously adjusted to closely mirror the training environment. These adjustments are crucial for the reliable performance of the neural network in controlling the robot's movements and are detailed further in the Methods chapter.

The signal processing involved in this transition was critical. It entailed the transformation of raw magnetic flux changes into a 'pseudo whisker angle,' allowing the neural network to interpret the sensor readings as if they were the angles of virtual whiskers it had 'experienced' during the simulation phase. For an in-depth explanation of this signal processing technique and its role in NeRmo's operation, the reader is referred to the Methods chapter.

The robot was placed in a static environment with about 1m x 1m space and after an initial check of the proper functionality of the whiskers, controlled by the neural net.

The walking and obstacle avoiding behavior was successful, despite some backwards movements occurring without having an obstacle in front of it.

5 Discussion

This chapter undertakes a comprehensive analysis and interpretation of the findings previously presented, aiming to contextualize these results within the broader scope of the research. The objective is to draw meaningful conclusions, illuminating the complexities of the whisker sensor technology and the operational dynamics of the robot mouse system.

Subsequent to this analysis, attention shifts towards the identification of limitations inherent to the study, especially with focus on possible modifications that would enhance either whisker sensor or the mouse robots sensing and actuation capabilities. This facilitates a detailed discussion on the prospective directions for future modifications and possible research.

Lastly, the conclusion is drawn from this investigation, summarizing the procedure as well as the findings.

5.1 Analysis and Interpretation

This section deals with a detailed analysis and interpretation of the research findings. Initially, the focus is on examining the simulated behavior of the novel sensor system, aiming to clarify the implications of the simulation results and setting it into context regarding measured values. This happens for both the mechanical simulation as well as for the magnetic simulation to explain the interplay between each other.
Following this, an exploration of the DRL outcomes is conducted, where the emphasis is on understanding the behaviors learned by the system, its responses to scenarios not covered during training, and the implications of these findings.
Finally, this section closes with an evaluation of the real robotic system's performance, considering how the simulated and DRL insights translate into physical world application. This comprehensive approach ensures a thorough understanding of the novel sensor system's capabilities implemented in the mouse robot.

5.1.1 Mechanical simulation

The mechanical simulation of this novel whisker sensor system underscores its efficacy as a highly sensitive tactile sensor, capable of detecting minute deflections with significant accuracy. The small total deformation observed in the whisker module, even under varied stress conditions, speaks to the system's robust design, minimizing the risk of collision between the inner and outer cylinder that would compromise sensor performance. This feature is particularly crucial as it ensures the whisker's operational integrity at a wide range of deflections, still providing meaning at broader displacements of the whisker.

The uniform distribution of von-Mises stress across the holding arm not only demonstrates the structural efficiency of the design but also its isotropic nature, ensuring consistent sensor responses under different loading scenarios. This consistency is of great importance for reliable tactile sensing, as it guarantees that the sensor's output remains predictable and relatively uniform, irrespective of the direction of applied force.
The correlation between elastic strain distribution and stress were expected as the whisker module is only made out of one material, therefore the same tensions will always lead to the same strain as response to external forces.

The detailed analysis of magnet deflection angles further enriches the understanding of the sensor's mechanical behavior, revealing a high sensitivity to directional forces due to different whisker impact directions. The slight deviations observed in deflection across different axes illuminate the system's subtle anisotropic properties, which, while minimal, necessitate careful consideration in sensor orientation to ensure uniform behavior across the sensor array. It is obvious that particularly the rotation about the z-axis is highly dependent on the impact orientation.

The revelation of these anisotropic tendencies not only invites a deeper examination of the sensor's mechanical design but also prompts a reevaluation of mounting strategies to optimize performance. By recommending the upward orientation of the outer structure's connection to the circular arm, it is proposed as a fast practical solution to equalize these properties across all whiskers - thereby enhancing the system's overall tactile sensing capabilities.

5.1.2 Magnetic Simulation and Sensor Data

In the simulated ideal environment, where the sensor is positioned directly behind the magnet, a slight rotation of the magnet leads to a significant variation in the magnetic flux components measured at the hall sensor's sensitivity area, while also exhibiting large values in the undeflected case.

In the specific case of the deflection of the whisker in the -y direction, the remarkable changes in magnetic flux due to deformations demonstrate high sensitivity. This sensitivity facilitates the discrimination between impact amplitudes and directions, underscoring the sensor's ability to effectively distinguish between different axes of deformation. This represents an essential feature for applications requiring precise spatial resolution.

Theoretically, the magnet's position relative to the sensor can be determined from the measured values. However, this is contingent upon the magnet's axis orientation being perpendicular to the sensor's sensitivity plane - a condition that may not always be met. The closer the actual setup aligns with this ideal condition, the more accurate the inverse position localization will be.

The observation that directional magnetic flux components exhibit significant variations across different sensor positions highlights the intricate non-linear relationship between mechanical stress, as manifested in magnet rotation, and the magnetic response.

Furthermore, the distinct differences in magnetic flux components between the ideal and actual robot system illuminate the challenges of applying theoretical models to practical applications. Although the ideal model provides valuable insights into anticipated behaviors, actual sensor system performance might differ due to unaccounted factors in the simulation, notably imperfect positioning.

5.1.3 Deep Reinforcement Learning

The training and behavioral analysis of the DRL model showcases that the approach is suitable to achieve a form of autonomous behavior just using sparse whisker signals. The in-depth insights are necessary to help explain what is going on and why the neural net behaves like it did.

In examining the training metrics, first the evaluation of the mean episode reward is done, given its greater significance relative to the mean episode length, even though the latter is introduced earlier in the results. This prioritization underscores the importance of understanding the agent's performance through the lens of rewards accumulated over time, which offers a more direct insight into the efficacy of the training process.

The first training phase metrics are just mentioned briefly due to its simplicity. Both the mean episode reward and the mean episode length exhibit a rapid increase, a trend anticipated to continue had the training not been halted. This pause was prompted by a visual inspection, which deemed the model's current state as adequately prepared for the subsequent stage of curriculum learning.

The introduction of obstacles in the second phase added a layer of complexity, a development that is evident in the training metrics.
When examining the average reward over time, fluctuations are noticeable even upon reaching convergence. It's crucial to acknowledge that the reward function does not serve as an absolute measure of the agent's performance - rather, it is significantly influenced by boundary conditions such as the maximum episode length, maze dimensions, and obstacle size. The observed variability in rewards is primarily attributed to the randomness introduced by the obstacles. Not only does the size of the obstacles change across episodes, but their location and orientation do as well. Consequently, the available space and the complexity required for navigation within the confined area vary from one episode to another. Episodes that demand less maneuvering around obstacles facilitate more straightforward, forward-directed actions, culminating in higher rewards.
The mean episode length is an indicator for the occurrence of the early stopping condition, when the mouse robot turns upside down. Low episode lengths means the simulation is frequently stopped before reaching the maximum of simulation steps per episode. A convergence of the mean episode length near to the maximal episode length means strategies are found to avoid behavior that causes early stopping. The variation in mean episode length likely stems from similar factors affecting the mean reward. It is noteworthy that the strong correlation between these two metrics is not as strong as expected, particularly towards the latter stages of training. Common intuition might suggest that longer episodes afford more opportunities to gather rewards. However, observations from the final phases of training reveal that the mean episode length does not reach its earlier peaks, while the mean episode reward is at its highest. This discrepancy highlights a nuanced

aspect of the training process, where the optimization of strategies, rather than the mere extension of episode duration, contributes to improved performance outcomes.

The model's ability to prioritize certain movements over others, particularly favoring forward navigation, illustrates an emergent strategy for obstacle avoidance of the custom reward function. This preference is not just a product of the learning algorithm's effectiveness but also reflects the strategic integration of previous sensory inputs to optimize navigation even though whisker inputs are just a sparse observation.
It seems like the preference of walking forward to the left over walking right is just a random phenomena, when training the model several times the option walking right will probably also become the preferred action sometimes, when an obstacle is sensed and maneuvering is necessary to avoid it.

The action distribution's bias toward specific movements sheds light on the underlying strategies that the DRL model develops to maximize efficiency and safety in navigation. This behavior indicates a sophisticated interpretation of sensory information, where the robotic mouse utilizes the whisker sensors not just for detecting obstacles but also for guiding its exploratory actions. Analyzing the distribution of actions can result in several hypotheses. First the least selected action of staying still seems to be avoided because the agent does not benefit from this action. If maneuvering is needed due to obstacle detection it is more useful to directly walk backwards so the robot can return to forward movement quicker. Secondly, in the group of backward directed movement stepping back without spine deflection is preferred because this reduces the risk of walking into a walk when doing a turn backwards. Colliding with a wall would not be sensed by the robot due to the lack of tactile perception in the body including limbs and tail, but still would be punished with negative rewards.

The qualitative assessment of a specific trial sheds light on the trained behavior of the agent. It showcases the ability to navigate within a constrained space, adhering predominantly to a leftward trajectory, which aligns with the expectations set by the quantitative action distribution analysis. The robotic mouse limits its exploration to only a part of the available area, creating circular movement patterns that require minimal maneuvering - a strategy likely adopted to maximize reward accumulation.
The whisker system proves effective in preventing wall collisions by enabling forward progression, with the tactile sensors timely detecting obstacles. The predominance of right whisker activations can be attributed to the robot's leftward circular path, which naturally

results in more frequent contacts on the right side, illustrating the robot's learned preference for this direction of movement.
The intense whisker activity observed mid-trial indicates that maneuvering to avoid obstacles is not always a straightforward task. The robot appears to make several attempts before successfully navigating past this point, suggesting a trial-and-error approach to overcoming this navigational challenge before it can resume its forward-directed movement.

The introduction of Robustness tests were done by exposing the agent to conditions it was not explicitly trained on. This was the preparation of potential problems that could occur later in the robot implementation, due to differences between actual setup and the modeled simulation.

The initial solid performance of NerMo sets a benchmark, illustrating the system's efficiency under optimal training conditions. This benchmark is critical for contextualizing the impact of subsequent variations in operational parameters, keeping the environment unaltered.
The observed decrease in both mean reward and episode length, following a reduction in walking speed, can be attributed to two primary factors. Firstly, the reduced speed likely limits the distance the robot can cover within the same timeframe, directly impacting its ability to gain higher rewards. Secondly, the shorter episode lengths mean an increased occurrence of early terminations, triggered by the robot flipping upside down. A plausible explanation for this phenomenon is the more frequent updating of the motor controller at reduced speeds, resulting in smaller changes in the gait phase prior to action transitions. Such frequent updates could compromise the stability of the robot's locomotion, potentially increasing the risk of early termination events.
Both hypotheses are suggesting that NerMo's performance may only be sensitive to changes in walking speed in the simulation, as the real robot does not have these problems.
The introduction of slight sensor noise had a minimal impact on performance, indicating a degree of resilience to small perturbations in sensory input. However, a more substantial noise level led to a dramatic decrease in both mean reward and episode length. This significant drop in performance with higher noise levels highlights the importance of sensor precision in maintaining operational efficiency. This problem could be eliminated by implementing noise into the training, while the gaussian noise of 3 degrees is relatively high and would in any scenario lead to accuracy loss of the whisker sensors.
The combined effects of reduced speed and small noise present a more complex challenge, reducing NerMo's mean return and episode length to a value below the slightly noisy case, but not as drastically as only walking speed reduction. The latter observation is especially interesting, because introduction of noise should always lead to a reduced performance as it

reduces the accuracy of the sensor. Even though no explicit explanation for this could be found, the combined test implies that NerMo can handle minor simultaneous disturbances.

The robustness test revealed interesting insights of the neural net behavior to unexpected environmental conditions. It is important to mention that some problems do not occur in the real robot that happened in the simulation, like the upside down flipping. When operating NeRmo it was never observed that the robot turned on its back. In the simulation there is the high probability this happens due to modeling imperfections that lead to balance problems. Therefore, a low mean episode reward in combination with a low mean episode length does not necessarily mean this scenario wont work in the real robot.
It's also crucial to recall that the reward mechanism in this study is specifically structured to prioritize distance traveled. Should the objective shift to a different aim, the outcomes of the robustness tests would likely vary significantly. For example, if the goal were to maximize exploration while minimizing early terminations, the focus would shift towards maintaining stability rather than solely focusing on obstacle avoidance and forward locomotion. This adjustment in objectives would naturally alter the performance criteria and, consequently, the evaluation of the system's robustness under varying conditions.

5.1.4 Robotic Implementation

The current implementation still shows large differences in magnetic flux components across each whisker sensor. This means probably that the hall sensors are not optimally aligned to the magnet in the whisker module as they are in the magnetic simulation as an ideal scenario. Due to small differences occurring at even deviations of one tenth of a millimeter, this was predictable but the gap between the magnetic flux values of the real scenario and the ideal model are still larger than expected. These differences also seem to indicate there are differences in their overall and directional sensitivity, because the relationship between rotation of the magnet and measured magnetic flux is strongly non-linear.

For the implementation of obstacle avoidance this wasn't a problem because if the three components of magnetic flux are processed in a way the provided input for the neural net matches the values in the DRL simulation. This procedure on one hand eliminates informational content reducing the dimensionality of the data. On the other hand in our scenario the exact impact directions don't play a significant role due to the static nature of the environment in combination with a preferred forward directed action. The impact is

probably coming from front so a change in magnetic flux would mean a rise in whisker angle which should trigger the obstacle avoidance behavior.
This applied transformation from magnetic flux to whisker angle is only possible because the dynamics of the whisker system are more or less clear. If the components of the magnetic field change due to a whisker collision induced rotation of the magnet, their absolute sum changes as well. As we know the angle rises as well in a determined way, because the impact is from forward, this rise should be enough information for the neural net to decide what to do. This allows a transfer of a simulation trained neural net to the real robot without running into problems at all at the decision process.

5.2 Challenges and Limitations

This research aims to integrate artificial tactile whiskers into a robotic platform using DRL for improved navigation and interaction in neurorobotics. While showing promising feasibility, it acknowledges challenges and limitations crucial for guiding future advancements.

The development and implementation process, from whisker module design to simulation and robotic integration, was influenced by technological, methodological, and practical factors. These included limitations in FE simulation coarseness due to software constraints and challenges in transferring virtual models to real sensors. Additionally, computational constraints and complexities in model transfer for real robot navigation from simulation training influenced the DRL component.

This project was conducted alongside another student's related thesis, leading to complementary rather than comprehensive aspects in isolation. This collaborative approach enriches the research scope, allowing each thesis to focus on distinct facets of the broader project. This may result in instances where the research presented here intersects with or relies upon findings detailed separately.

Acknowledging these limitations is not an admission of shortfall but underscores the study's complexity regarding the challenges tackled. In the following sections, these limitations are detailed to define the boundaries of the current findings and to pave the way for future explorations that surpass these initial constraints.

5.2.1 Whisker System

The developed whisker system consisting of a magnet and 3d Hall-Sensor does have much more limited capabilities than biological counterparts.
Firstly, the artificial whiskers fall short in their ability to detect textures, a capability intrinsic to biological systems. Additionally, the whisker system lacks the active sensing mechanisms that allow biological whiskers to actively probe and explore their surroundings. The design of the nylon string as artificial tactile hair also significantly differs from the rod-shaped whiskers of biological mice, potentially affecting the fidelity of environmental interaction. Moreover, the number of whiskers employed in the mouse robot is considerably fewer than that found in natural examples, limiting the spatial resolution and scope of sensory input.

5.2.2 Finite Element Simulations

The accuracy of FE simulations is significantly influenced by the coarseness of the mesh used to represent geometries. Due to restrictions imposed by the student license, the study was limited to employing a coarser mesh, leading to a lower resolution of data points. Furthermore, to maintain the model's feasibility, several assumptions were made.
For instance, the mechanical simulation presupposed uniform and isotropic material properties, while in reality there are inherent material variations and the anisotropy introduced by the 3D printing process.

In the magnetic simulation this concerns the use of an idealized model, not only neglecting surrounding material but also assuming an ideally positioned sensor. Moreover, it did not account for potential disruptive influences, such as electromagnetic fields generated by electronic components like the PCB. The resolution and accuracy limitations of the sensor itself were also not taken into account.

5.2.3 Deep Reinforcement Learning

The DRL process applied showed potential for enhanced outcomes.
One limitation aspect was the extensive requirement for experimentation with hyperparameters and the exploration of various agent models. Given the high computational demands of DRL, efforts were made to minimize the extent of this experimentation and to limit the number of models trained for time reasons.

Additionally, the extent of experiments conducted to assess the model was not as extensive as it could have been. A more thorough investigation into the behavior and robustness of the neural net could have provided deeper insights into its decision-making processes. Ideally, training could have incorporated a wider range of conditions to test the model's resilience, such as variations in speed, environmental noise, and different levels of friction between the robot and the ground.

Another notable limitation was the agent's lack of a genuine memory system, which would significantly help in efficient navigation. The strategy of incorporating only the most recent observations was a practical approach to circumvent the complexities of implementing advanced memory algorithms like Long Short-Term Memory networks. Although these more complex systems might offer superior performance over simple frame stacking, their exclusion was a calculated decision to manage the project's scope and computational feasibility.

5.3 Potential Improvements

Subsequent enhancements for the whisker sensors and the robotic mouse are crucial for the continuous progress of this project. This section introduces potential modifications to the existing system, describing the progressive steps necessary for the development of the robot. Further improvements would have the potential to lead to a robotic mouse that serves as a platform for computational neuroscience investigations and experimental research. The high informational content of the whisker sensor, coupled with its applicability across a broad spectrum of research areas and use cases, extends beyond the scope of this thesis.

5.3.1 Active Whiskers

Integrating active whisking capabilities into NeRmo presents a significant advancement in replicating the tactile exploration efficiency found in biological counterparts. Implementing small DC motors to generate back-and-forth whisker motions could substantially enhance the robot's ability to interact with its environment. This approach necessitates a sophisticated yet feasible mechanism, where a simple yet effective method involves mounting the whisker sensors on a flexible base. This setup would allow a motor to apply tension to the whiskers, inducing deformation, and upon releasing the tension, the material would revert to its original state.
An active sensing mechanism, inspired by the natural whisking behavior of rodents, promises to improve the robot's spatial awareness and object detection accuracy.

5.3.2 Better Hall Sensor Positioning and Calibration

Uniform sensor sensitivity across the artificial whisker array is crucial for accurate spatial mapping. Adjusting the relational positioning of each whisker to its corresponding Hall effect sensor ensures consistent signal processing. This improvement demands further design changes in the head of the mouse robot, potentially in an iterative manner. The position of the whisker modules as well as the mounting of the PCBs has to be examined for potential sources of error to minimize the existing deviations between ideal positioning and the actual measured data from the sensors.

Given the inherent variability in whisker positioning due to factors that are difficult to change like manufacturing processes, a calibration script becomes essential. This script would be developed in an experimental manner to determine the precise location of each Hall effect sensor relative to its whisker, adjusting the data processing algorithms accordingly. Such calibration not only enhances sensor accuracy but also compensates for any initial misalignments, ensuring reliable angle calculations and magnetic field interpretation.

5.3.3 More Whiskers into the Head

Expanding the number of whiskers implemented in the head of the mouse robot addresses the limitation of spatial sampling by the current system. Mimicking the dense whisker layout of rodents could significantly improve the robot's environmental perception and obstacle detection capabilities. This adjustment requires careful consideration of magnetic field interference among closely positioned whiskers, aiming to replicate the complex sensory integration observed in biological systems.

5.3.4 Modular PCBs

Adopting a modular approach to PCB design for each whisker sensor offers flexibility in modifying the whisker array. This design allows for quick adjustments to the number of whiskers, overcoming current limitations related to fixed distances between sensors. Modular PCBs facilitate experimentation with different whisker layouts, potentially enhancing the robot's tactile sensing capabilities.

6 Conclusion

This chapter deals with the conclusion drawn from the presented work. Also an outlook is given on possible future research including artificial whiskers and the mouse robot.

6.1 Summary

This research was about developing a novel tactile whisker sensor, implementing it into a neurorobotic mouse and applying it in obstacle avoidance tasks using deep reinforcement learning. For this purpose, literature was investigated about the whiskers in biological rodents, their modeling approaches and existing artificial whisker sensors as well as the state of the art in DRL.

This information was used to select a suitable method and develop a low cost prototype of a novel artificial whisker sensor. After the concept was proven to work in the testing phase, additional efforts were made to optimize the whisker module regarding space efficiency to fit into the mouse robot's head. To implement the whisker system as well as the Hall-Effect sensor with the PCB into the head, the CAD design of the head was modified and 3D printed. With the aid of a Finite Elements simulation further optimizations were conducted to improve mechanical properties like isotropy of the whisker module. Additionally, a magnetic simulation was implemented to get more insights into the magnetic field and its response to whisker deflection. The results of the idealized model were compared to the actual data obtained by the sensors implemented into the robot. The simulated magnetic field was analyzed to find at which coordinates the values from the real sensors occur to show potential for improvements.

For conducting DRL for obstacle avoidance with the novel sensor system, the whiskers were implemented in the mouse robot model of the existing mujoco simulation. Moreover, the custom environment was setted up to prepare the interaction between the physics simulation and the DRL algorithm. Afterwards, several trainings were conducted to generate obstacle avoidance behavior using whiskers as only sensors for perceiving the external world. The resulting training metrics were examined and the behavior of the trained neural net was further analyzed regarding different aspects. This included the general distribution of the actions as well as a qualitative analysis of a resulting walking path. To check for potential problems due to differences between reality and simulation, the neural net was exposed to potential perturbation factors it was not trained on like occurance of noise or difference in walking speed.

Then everything was set up to allow the neural net using real time sensor values obtained by the actual mouse robot. This included several steps, for adjusting the robot conditions to the one in the DRL simulation. First signal processing for conversion of the magnetic flux values to whisker angles in the horizontal plane was done. Afterwards, the temporal dynamics of the communication between laptop and robot were handled in an appropriate manner. To validate the function, the robot was exposed to a static environment and in a visual inspection the obstacle avoidance was done successfully, even though it is not perfect due to the limited sensing capabilities of the robot.

6.2 Future Research Directions

By providing a robot mouse platform with tactile perception this study paths the way for further research. This could be from particular importance for investigations regarding rodents and whiskers in computational neuroscience.

Building a model of the sensorimotor system and implementing it in the robot would be a interesting investigation. An approach like this already was realized in a simulation in the research from Antonietti and colleagues in 2022. They used a spiking neural net to mimic a slick non-slick experiment that is commonly used in rodents to test their learning capabilities.

Head movement plays an important role in rodents. One application and research topic of the mouse robot would be to include head movements to investigate more about how rodents use them to make their exploration behavior more efficient. Head movements seem to increase the informational content perceived by the whisker system to perceive the environment. This additional motion would reduce the need of time and energy consuming navigation using the legs just to get more information about the spatial surroundings. This way in the same amount of time more space can be explored without the need for hardware changes and also decreases the difference to the biological system.

With the developed tactile sensor system enough spatial information should be obtainable from the environment to create some kind of 3D map with it. One challenge here is definitely to determine the location of the mouse, so there would be the condition that the localization of the robot is done one way or the other. This could allow researchers to realize a robot that uses tactile whisker information like rodents do to construct a 3D map of the environment. This could benefit for contour reconstruction of an object for identification or also give the robot the capability to step over a small obstacle if the physical conditions are met.

Bibliography

Adibi M (2019) Whisker-Mediated Touch System in Rodents: From Neuron to Behavior. Front. Syst. Neurosci. 13:40. doi: 10.3389/fnsys.2019.00040

Antonietti, A., Geminiani, A., Negri, E., D'Angelo, E., Casellato, C., & Pedrocchi, A. (2022). Brain-inspired spiking neural network controller for a neurorobotic whisker system. Frontiers in Neurorobotics, 16. https://doi.org/10.3389/fnbot.2022.817948

Bauer, G. B, Reep, R. L, & Marshall, C. D. (2018). The Tactile Senses of Marine Mammals. International Journal of Comparative Psychology, 31. http://dx.doi.org/10.46867/ijcp.2018.31.02.01 Retrieved from https://escholarship.org/uc/item/1vk1c9z1

Beem, H. (2016). Seal whiskers inspire marine technology. Ocean. Mag, 51, 82-85.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J.W., Petrov, M., Pinto, H.P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., & Zhang, S. (2019). Dota 2 with Large Scale Deep Reinforcement Learning. ArXiv, abs/1912.06680.

Carvell GE, Simons DJ. Biometric analyses of vibrissal tactile discrimination in the rat. Journal of Neuroscience. 1990; 10(8):2638–48. https://doi.org/10.1523/JNEUROSCI.10-08-02638.1990 PMID: 2388081

Clements, T. N., & Rahn, C. D. (2006). Three-dimensional contact imaging with an actuated whisker. IEEE Transactions on Robotics, 22(4), 844-848. doi:10.1109/TRO.2006.878950.

Dai, W., Chen, S., Huang, Z., Xu, Y., & Kong, D. (2022). LiDAR Intensity Completion: Fully Exploiting the Message from LiDAR Sensors. Sensors (Basel, Switzerland), 22. https://doi.org/10.3390/s22197533.

De Oliveira, Rodrigo & Fernandes Araújo, Ramon Cristian & Barros, Fabrício & Segundo, Adriano & Zampolo, Ronaldo & Fonseca, Wellington & Dmitriev, Victor & Brasil, Fernando. (2017). A System Based on Artificial Neural Networks for Automatic Classification of Hydro-generator Stator Windings Partial Discharges. Journal of Microwaves, Optoelectronics and Electromagnetic Applications. 16. 628-645. 10.1590/2179-10742017v16i3854.

Delamare, J., Sanders, R., & Krijnen, G. (2016, October). 3D printed biomimetic whisker-based sensor with co-planar capacitive sensing. In 2016 IEEE SENSORS (pp. 1-3). IEEE.

Diamond, M. E.; Arabzadeh, E., 2013. Whisker sensory system - from receptor to decision. Progress in Neurobiology. Vol. 103, pp. 28–40. Available from DOI:10.1016/j.pneurobio.2012.05.013.

Diamond, M. E., von Heimendahl, M., & Arabzadeh, E. (2008). Whisker-mediated texture discrimination. PLoS biology, 6(8), e220.

Fox, C.W., Evans, M.H., Pearson, M.J., Prescott, T.J., Towards hierarchical blackboard mapping on a whiskered robot, Robotics and Autonomous Systems (2012), doi:10.1016/j.robot.2012.03.005

Grant RA, Breakell V, Prescott TJ. 2018 Whisker touch sensing guides locomotion in small, quadrupedal mammals. Proc. R. Soc. B 285: 20180592. http://dx.doi.org/10.1098/rspb.2018.0592

Grant RA, Mitchinson B, Fox CW, Prescott TJ. Active touch sensing in the rat: Anticipatory and regulatory control of whisker movements during surface exploration. Journal of Neurophysiology. 2009; 101 (2):862–74. https://doi.org/10.1152/jn.90783.2008 WOS:000263120300033. PMID: 19036871

Guic-Robles E, Valdivieso C, Guajardo G. Rats can learn a roughness discrimination using only their vibrissal system. Behavioural Brain Research. 1989; 31(3):285–9. https://doi.org/10.1016/0166-4328(89)90011-9 WOS:A1989R824800010. PMID: 2914080

Harris, Justin A., Rasmus S. Petersen, and Mathew E. Diamond. "Distribution of tactile learning and its neural basis." Proceedings of the National Academy of Sciences 96.13 (1999): 7587-7591.

Hartmann, M. J., Johnson, N. J., Towal, R. B., & Assad, C. (2003). Mechanical characteristics of rat vibrissae: Resonant frequencies and damping in isolated whiskers and in the awake behaving animal. Journal of Neuroscience, 23(16), 6510-6519.

Harvey, A., Roberto Bermejo, H., Philip Zeigler, M. (2001). Discriminative whisking in the head-fixed rat: optoelectronic monitoring during tactile detection and discrimination tasks. Somatosensory & motor research, 18(3), 211-222.

Hires, S. A., Pammer, L., Svoboda, K., & Golomb, D. (2013). Tapered whiskers are required for active tactile sensation. eLife. Advance online publication. http://dx.doi.org/10.7554/elife.01350.001.

Huet LA, Emnett HM, Hartmann MJZ (2022) Demonstration of three-dimensional contact point determination and contour reconstruction during active whisking behavior of an awake rat. PLoS Comput Biol 18(9): e1007763. https://doi.org/10.1371/journal.pcbi.1007763

Huet, L. A., Schroeder, C. L., & Hartmann, M. J. Z. (2015). Tactile signals transmitted by the vibrissa during active whisking behavior. Journal of Neurophysiology. Advance online publication. http://dx.doi.org/10.1152/jn.00011.2015.

Jenks, Robert A., et al. "Self-motion and the shaping of sensory signals." Journal of neurophysiology 103.4 (2010): 2195-2207.

Kim, S., Kubicek, R., Paris, A., Tagliabue, A., How, J. P., & Bergbreiter, S. (2020, October). A whisker-inspired fin sensor for multi-directional airflow sensing. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1330-1337). IEEE.

Knutsen, P. M., Pietr, M., & Ahissar, E. (2006). Haptic object localization in the vibrissal system: behavior and performance. Journal of Neuroscience, 26(33), 8451-8464.

Kottapalli, A. G. P., Asadnia, M., Miao, J. M., & Triantafyllou, M. S. (2015, January). Harbor seal whisker inspired flow sensors to reduce vortex-induced vibrations. In 2015 28th IEEE International Conference on Micro Electro Mechanical Systems (MEMS) (pp. 889-892). IEEE.

Krupa DJ, Matell MS, Brisben AJ, Oliveira LM, Nicolelis MAL. Behavioral properties of the trigeminal somatosensory system in rats performing whisker-dependent tactile discriminations. Journal of Neuroscience. 2001; 21(15):5752–63. https://doi.org/10.1523/JNEUROSCI.21-15-05752.2001 PMID: 11466447

Lederman, S. J., & Klatzky, R. L. (1987). Hand movements: A window into haptic object recognition. Cognitive psychology, 19(3), 342-368.

Lepora, N. F., Pearson, M., & Cramphorn, L. "TacWhiskers: Biomimetic Optical Tactile Whiskered Robots," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 7628-7634, doi: 10.1109/IROS.2018.8593653.

Luo, S., Bimbo, J., Dahiya, R., & Liu, H. (2017). Robotic tactile perception of object properties: A review. Mechatronics, 48, 54-67. https://doi.org/10.1016/j.mechatronics.2017.11.002

Mathew E. Diamond*, Moritz von Heimendahl*, Per Magne Knutsen‡, David Kleinfeld and Ehud Ahissar‡ Where' and 'what' in the whisker sensorimotor system 2008

Mohammed Salman, Martin J. Pearson, "Advancing whisker based navigation through the implementation of Bio-Inspired whisking strategies", 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp.767-773, 2016.

Narvekar, Sanmit; Peng, Bei; Leonetti, Matteo; Sinapov, Jivko; Taylor, Matthew E.; Stone, Peter. (2020). Curriculum learning for reinforcement learning domains: a framework and survey. J. Mach. Learn. Res. 21, 1, Article 181 (January 2020)

Pammer L, O'Connor DH, Hires SA, Clack NG, Huber D, Myers EW, et al. The mechanical variables underlying object localization along the axis of the whisker. Journal of Neuroscience. 2013; 33 (16):6726–41. https://doi.org/10.1523/JNEUROSCI.4316-12.2013 WOS:000317723000004. PMID: 23595731

Polley DB, Rickert JL, Frostig RD. Whisker-based discrimination of object orientation determined with a rapid training paradigm. Neurobiology of Learning and Memory. 2005; 83(2):134–42. https://doi.org/10.1016/j.nlm.2004.10.005 WOS:000227513200006. PMID: 15721797

Quist, B. W., Seghete, V., Huet, L. A., Murphey, T. D., & Hartmann, M. J. Z. (2014). Modeling forces and moments at the base of a rat vibrissa during non-contact whisking and whisking against an object. Journal of Neuroscience, 34, 9828–9844. https://doi.org/10.1523/JNEUROSCI.1707-12.2014.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research, 22(268), 1–8.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature 323, 533–536. doi:10.1038/323533a0.

Schmidt, M., Witte, H., Zimmermann, K., Niederschuh, S., Helbig, T., Voges, D., Husung, I., Volkova, T., Will, C., & Behn, C. (2014). Technical, non-visual characterization of substrate contact using carpal vibrissae as a biological model: An overview. In Proceedings of the 58th IWK, Ilmenau Scientific Colloquium: Shaping the Future by Engineering (pp. 8). Ilmedia.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347

Stocking, J. B., Eberhardt, W. C., Shakhsheer, Y. A., Calhoun, B. H., Paulus, J. R., & Appleby, M. (2010, November). A capacitance-based whisker-like artificial sensor for fluid motion sensing. In SENSORS, 2010 IEEE (pp. 2224-2229). IEEE.

Tiwana, M. I., Tiwana, M. I., Redmond, S. J., Lovell, N. H., & Iqbal, J. (2016). Bio-inspired PVDF-based, mouse whisker mimicking, tactile sensor. Applied Sciences, 6(10), 297.

Todorov, E., Erez, T., & Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 5026-5033). IEEE. https://doi.org/10.1109/IROS.2012.6386109

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., Cola, G. de, Deleu, T., Goulão, M., Kallinteris, A., KG, A., Krimmel, M., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Shen, A. T. J., & Younis, O. G. (2023, March). Gymnasium. Zenodo. https://doi.org/10.5281/zenodo.8127026

Tsujimura, T., & Yabuta, T. (1989). Object detection by tactile sensing method employing force/torque information. IEEE Transactions on Robotics and Automation, 5(4), 444-450. doi:10.1109/70.88059.

Warren, R. A., Zhang, Q., Hoffman, J. R., Li, E. Y., Hong, Y. K., Bruno, R. M., & Sawtell, N. B. (2021). A rapid whisker-based decision underlying skilled locomotion in mice. Elife, 10, e63596.

Wang, Dasong & Snooks, Roland. (2021). Artificial Intuitions of Generative Design: An Approach Based on Reinforcement Learning. 10.1007/978-981-33-4400-6_18.

Wang, S., Bhatia, A., Mason, M. T., & Johnson, A. M. Contact localization using velocity constraints. In International Conference on Intelligent Robots and Systems, 2020.

Wei, Z., Shi, Q., Li, C., Yan, S., Jia, G., Zeng, Z., ... & Fukuda, T. (2019, September). Development of an MEMS based biomimetic whisker sensor for tactile sensing. In 2019 IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 222-227). IEEE.

Yan, W., Kan, Q., Kergrene, K., Kang, G., Feng, X. Q., & Rajan, R. (2013). A truncated conical beam model for analysis of the vibration of rat whiskers. Journal of Biomechanics, 46(12), 1987-1995. https://doi.org/10.1016/j.jbiomech.2013.06.015

Yang, S., & Wang, C. (2011). On Solving Mirror Reflection in LIDAR Sensing. IEEE/ASME Transactions on Mechatronics, 16, 255-265. https://doi.org/10.1109/TMECH.2010.2040113.

Zhao, C., Zhang, S., Xie, T., & Zeng, L. (2022). A novel whisker sensor with variable detection range for object positioning. Review of Scientific Instruments, 93(3).

Zhenshan Bing et al., Lateral flexion of a compliant spine improves motor performance in a bioinspired mouse robot. Sci. Robot.8(2023).DOI:10.1126/scirobotics.adg7165

List of Figures

List of Tables

List of Abbreviations

MDP	Markov Decision Process
POMDP	Partially Observable Markov Decision Process
DRL	Deep Reinforcement Learning
RL	Reinforcement Learning
FE	Finite Elements
FEA	Finite Elements Analysis
PPO	Proximal Policy Optimization
TPU	Thermoplastic Polyurethane
PLA	Polylactic Acid
FDM	Fused Deposition Modeling

Selbstständigkeitserklärung