

Bachelor's Thesis in Robotics, Cognition, Intelligence

Active Tactile Exploration Based on Whisker-Inspired Sensory Array

Aktive taktile Erkundung auf der Grundlage eines schnurrhaar-inspirierten Sensorarrays

Supervisor Prof. Dr.-Ing. habil. Alois C. Knoll

Advisor M.Sc. Yixuan Dang

Author Valentin Safronov

Date April 01, 2025 in Munich

Disclaimer

I confirm that this Bachelor's Thesis is my own work and I have documented all sources and material used.

Munich, April 01, 2025



(Valentin Safronov)

Abstract

Object contour reconstruction is critical in robotics for tasks such as navigation and recognition. Whisker sensors offer a promising tactile modality due to their high spatial resolution, robustness under varying conditions, and low computational requirements. However, existing approaches typically fail to reconstruct objects with sharp corners, as the whisker detaches from the object, and do not allow for contour reconstruction in confined spaces. This work introduces a development platform integrating multiple magnetically transduced whisker sensors mounted on a robotic arm for active control. For single-whisker configurations, we propose an enhanced contour-following strategy paired with an object retrieval policy to manage unavoidable whisker detachments at sharp corners and accurately capture their contours. A tunneling policy is introduced for multi-whisker setups, and all control policies are combined into a single finite state machine that dynamically selects the active policy and determines which whiskers are relied on. Simulation results demonstrate that our method effectively reconstructs contours with sharp angles with sub-millimeter accuracy and maintains a centered trajectory in confined passages. Finally, we present a comprehensive framework for real-world experiments, including data collection, preprocessing, evaluation, storage, and visualization.

Kurzfassung

Die Rekonstruktion von Objektkonturen ist essenziell in der Robotik für Aufgaben wie Navigation und Objekterkennung. Whisker-Sensoren bieten aufgrund ihrer hohen räumlichen Auflösung, Robustheit unter wechselnden Bedingungen und geringen Rechenanforderungen eine vielversprechende taktile Modalität. Bestehende Ansätze scheitern jedoch häufig an der Rekonstruktion von Objekten mit scharfen Ecken, da sich die Whisker hierbei vom Objekt lösen, und ermöglichen keine Konturrekonstruktion in engen Räumen. Diese Arbeit stellt eine Entwicklungsplattform vor, die mehrere magnetisch transduzierte Whisker-Sensoren integriert, welche aktiv durch einen Roboterarm gesteuert werden. Für Ein-Whisker-Konfigurationen schlagen wir eine verbesserte Konturverfolgungsstrategie vor, kombiniert mit einer Strategie der Objektwiederaufnahme, um unvermeidliche Whisker-Ablösungen an scharfen Ecken zu bewältigen und deren Konturen präzise zu erfassen. Für Multi-Whisker-Konfigurationen wird eine Tunneling-Strategie eingeführt, und alle Steuerstrategien werden in einer einzigen Zustandsmaschine kombiniert, die dynamisch die aktive Strategie auswählt und bestimmt, auf welche Whisker vertraut wird. Simulationsergebnisse zeigen, dass die vorgestellte Methode Konturen mit scharfen Winkeln im Submillimeterbereich genau rekonstruiert und eine mittige Trajektorie in engen Passagen aufrechterhält. AbschlieSSend präsentieren wir ein umfassendes Framework für reale Experimente, inklusive Datenerfassung, Vorverarbeitung, Auswertung, Speicherung und Visualisierung.

Contents

1	Introduction	1
1.1	Biological Whiskers	1
1.2	Whisker-inspired Tactile Sensors	2
1.3	Active Tactile Perception	3
1.4	Overview of the Thesis	4
1.4.1	Simulation	4
1.4.2	Control System	5
1.4.3	Swiping Policy	5
1.4.4	Retrieval Policy	5
1.4.5	Tunneling Policy	5
1.4.6	Governing Policy	6
1.4.7	Hardware Setup	6
1.4.8	System Infrastructure	6
1.5	Key Contributions	6
2	Related Works	9
2.1	Whisker-Inspired Tactile Sensors	9
2.1.1	Magnetically Transduced Whisker Sensors	9
2.1.2	Piezoresistive Whisker Sensors	10
2.1.3	MEMS-based Whisker Sensors	10
2.1.4	Contour Reconstruction	11
2.2	Active Tactile Exploration	11
2.2.1	Contour Reconstruction	11
2.2.2	Environment Manipulation	12
3	Hardware Design	15
3.1	Magnetically Transduced Whisker Sensor	15
3.2	Whisker Platform	16
3.3	Data Acquisition	16
4	Control Algorithms	19
4.1	Overview	19
4.2	Variables	20
4.3	Data Preprocessing	20
4.4	Body Motion Control	21
4.5	Swiping Policy	21
4.6	Retrieval Policy	23
4.6.1	Premise	24
4.6.2	Whisking Back to the Edge	24
4.6.3	Transition to Swiping	25
4.7	Tunneling Policy	25

4.8	Finite State Machine	25
4.9	Additional features	26
4.9.1	Retrieval without Edge Reconstruction	26
4.9.2	Platform Tilt	26
4.9.3	Reverse Swiping	26
4.9.4	Positioning of Whiskers at Different Angles	26
4.9.5	Integration of Multiple Whiskers on Each Side	27
4.10	Comparison to the Baseline	27
4.10.1	Baseline Algorithm	27
4.10.2	Comparison	27
4.10.3	Limitations of the Suggested Control Algorithm	28
5	Experimental Setup	31
5.1	Swiping Policy	31
5.2	Retrieval Policy	33
5.3	Tunneling Policy	36
6	Infrastructure	41
6.1	Overview	41
6.2	Controller	42
6.3	Simulation	42
6.4	System Services	42
6.5	Dashboard	45
7	Conclusion	47
7.1	Summary	47
7.2	Future Work	48
7.3	Potential Improvements	48
A	Additional Information	53
A.1	Control Hierarchy	53
	Bibliography	55

List of Figures

1.1	Representation of the vibrissae system and its function [IMG22]	2
1.2	A taxonomy for whisker-based sensors [Say+22]	3
2.1	Side-by-side comparison of whisker sensor designs.	10
2.2	Operating principles of the PWWS by Guo et al. [Guo+24]. (a) Schematic of whisker deflection under force. (b) Cycle of electrical signals generated by the piezoelectric sensing unit. (c) Simulation of piezoelectric voltage within PVDF material under various constraints when subjected to external forces.	11
2.3	SEM image of the MEMS-based biomimetic whisker sensor by Wei et al. [Wei+19].	12
2.4	Schematic visualizations of a) straight line control, b) burrowing action primitive, and c–e) a sequence showing the progression of a clockwise excavate action primitive, by Brouwer et al. [Bro+24].	13
3.1	Single whisker sensor.	15
3.2	Whisker platform CAD model and sketch.	16
3.3	Assembled whisker platform with three whiskers and robotic arm mount. . . .	17
3.4	MLX90393 sensor hardware from [Mel17]	17
3.5	Whisker deflection profile model.	18
4.1	Angle Retrieval Strategy. Black [XE], [EY] — object surface, Green [BW] — the whisker, Blue [EY'] — the potential adjacent surface of the edge and its normal [WT], Blue circle — the targeted contact points.	24
4.2	Retrieval Policy in 8 steps.	29
4.3	FSM diagram of the whisker control system. The diagram illustrates the various states (Exploring, Swiping, Tunneling, Retrieval, Whisking, Failure) and the transitions triggered by contact establishment, detachment, and reattachment.	30
5.1	Contour Estimation of Disk With Swiping Policy	32
5.2	Contour Estimation of Rounded Rectangular Box With Swiping Policy	32
5.3	Contour Estimation of Complex Object With Swiping Policy	33
5.4	Contour Estimation of Octagon (Edges at 135°) With Swiping and Retrieval Policy	34
5.5	Contour Estimation of Box (Edges at 90°) With Swiping and Retrieval Policy .	35
5.6	Contour Estimation of Prism (Edges at 60°) With Swiping and Retrieval Policy	35
5.7	Contour Estimation of Wall (Edges at 90°) With Swiping and Retrieval Policy .	36
5.8	Navigation and Contour Estimation of Smooth Tunnel With Swiping and Tunneling Policy	37
5.9	Navigation and Contour Estimation of Zigzag Tunnel With Swiping and Tunneling Policy	38

5.10 Navigation and Contour Estimation of Round Tunnel With Swiping and Tun-	
neling Policy	39
6.1 Overview of the system infrastructure.	41
6.2 Data flow in the system infrastructure.	43
6.3 Simulation interface of the MuJoCo physics engine.	44
6.4 Data flow in the system infrastructure.	44
6.5 visualization of magnetic data of 6 whisker sensors in real-time using Grafana. .	45
A.1 Hierarchy of classes in the control system.	53

List of Tables

4.1 Overview of input and output variables used in the control system.	21
--	----

List of Algorithms

1 Steer the Platform to Target Position and Orientation	22
2 Steer Whisker to Target Position and Orientation	22
3 Swiping Policy	23
4 Tunneling Policy	25

Chapter 1

Introduction

Whisker-inspired tactile sensing represents a promising solution for active perception in robotics, especially in navigating unstructured or visually obscured environments. This thesis investigates the potential of biomimetic whisker sensors integrated into a robotic platform to achieve precise reconstruction and autonomous navigation through active tactile perception. By leveraging insights from biological whisker systems, we propose novel control strategies and demonstrate their efficacy in simulation and real-world scenarios.

1.1 Biological Whiskers

Over the past two decades, tactile perception has gained attention within robotics, a field traditionally dominated by visual sensing [Say+22]. Tactile sensing provides distinct advantages, especially when visibility is limited, in low-light conditions, or when environments are narrow, cluttered, or visually obstructed. Nature provides ample evidence of the effectiveness of tactile sensing and offers inspiration for developing tactile sensors.

Some animals possess tactile sensory hairs known as whiskers or vibrissae, complementing their vision. Whiskers enable them to interact effectively with their environment through non-intrusive mechanical contact. “Traditionally, whiskers are associated with diverse survival skills, including tactile discrimination, distance assessment, food acquisition, gap crossing, and social interaction” note Ibarra-Castaneda et al. [IMG22]. For example, seals and sea lions use their whiskers for hunting in dark and turbid environments. When prey swims nearby, the generated hydrodynamic vortices cause distinct, jerky deflections in the whiskers. These deflections reveal the hydrodynamic trail left by the prey, allowing seals and sea lions to locate their prey effectively [MB18]. Thus, whiskers function as passive sensors by responding directly to external excitation.

In contrast, some rodents, such as rats and squirrels, can move their whiskers in rhythmic motions known as *whisking*, enabling them to actively perceive their immediate surroundings and distinguish surface textures. Different textures produce unique slip-stick motion patterns and deflection magnitudes of whiskers, resulting in characteristic resonant vibrations that rodents interpret to identify surfaces and objects [Wol+08].

Rats depend on whisker-based navigation when traversing confined spaces like tunnels or burrows. Through coordinated whisking motions, rats efficiently detect the layout and contours of the environment, allowing them to move even in complete darkness [Wol+08].

The anatomical structure and functional mechanisms of biological whiskers are depicted in Figure 1.1. Whiskers come in various shapes and sizes, and one animal can have multiple whiskers, each with a different length and taper. Sensory detection is carried out solely by mechanoreceptors located at the base of each whisker follicle, without any additional

receptors along the whisker shaft [HRH17]. This simplicity, effectiveness, and variability of biological whiskers provide a foundation for developing biomimetic tactile sensors.

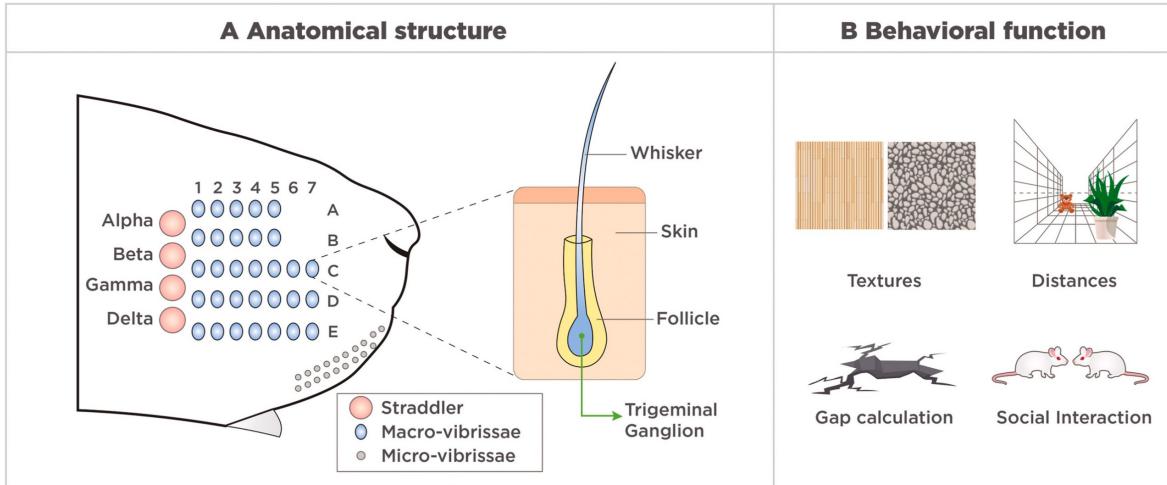


Figure 1.1: Representation of the vibrissae system and its function [IMG22]

1.2 Whisker-inspired Tactile Sensors

The versatility and effectiveness of biological whiskers have inspired numerous developments in the field. Whisker-inspired tactile sensors have extensive applications, including [Say+22]:

- Recognition of the surrounding objects and obstacle avoidance (extraction of the 3D whisker tip position)
- Exploration of unstructured environments
- Leak detection in pipelines
- Water flow detection in unmanned underwater vehicles
- Midair obstacle detection for drones
- Tactile sensing of heart valves
- Navigation in dark or visually obscured environments
- Study of the surface texture (like surface hardness and adhesiveness of food) [Par+24]

Whisker-inspired tactile sensors offer low power consumption, minimal computational requirements, and high spatial resolution. Some of the most commonly used whisker-inspired tactile sensors are [Say+22]:

- **Strain gauge sensors:** measure changes in electrical resistance caused by deformation of strain gauges attached to the whisker shaft.
- **Hall effect sensors:** detect changes in magnetic field strength caused by displacement or rotation of the magnet positioned at the whisker base.
- **Capacitive sensors:** detect changes in capacitance caused by deformation of the whisker.

- **Piezoelectric sensors:** generate an electrical charge in response to mechanical stress applied to the whisker.
- **Optical sensors:** change light path or intensity as the fiber-based whisker deflects.
- **Magnetoresistive sensors:** measure changes in electrical resistance caused by changes in magnetic fields when whisker deflection occurs.
- **MEMS-based sensors:** employs a capacitive or piezoelectric sensor in a microelectromechanical system format.

The taxonomy of whisker sensors is shown in Figure 1.2.

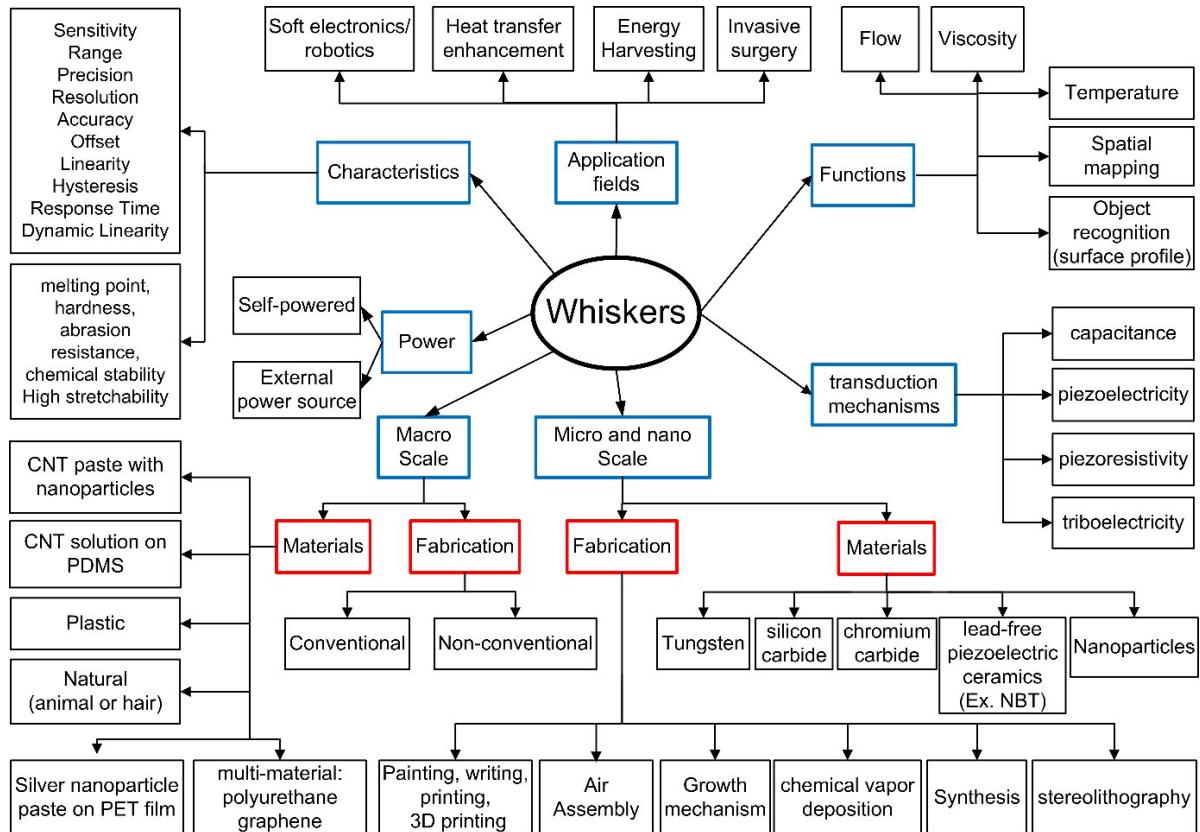


Figure 1.2: A taxonomy for whisker-based sensors [Say+22]

1.3 Active Tactile Perception

One of the most critical problems in robotics is **reconstruction of unstructured environments**. According to Katz et al. [KKB10], it is essential for efforts to develop **autonomous mobile manipulators**, that are said to have significant societal and economic impact. Operation in unstructured environments poses multiple challenges, such as:

- Limited knowledge of the surroundings
- Uncertainty in the robot's position
- Constraints on the robot's mobility

- Impermanence of the state of the environment

Thus, the reconstruction of unstructured environments plays a pivotal role in the robot's ability to navigate and manipulate objects in its environment.

Whisker sensors are particularly well-suited for this task due to their **non-intrusive nature** and **reconstruction precision**. They can be attached to the autonomous mobile manipulator, providing it with tactile information for accurate environment reconstruction, navigation, and obstacle avoidance. Furthermore, they are lightweight and low-cost, making them suitable for integration into small robots.

However, using whisker sensors for active tactile perception poses several challenges. For single whisker configurations, the following challenges arise:

- The whisker requires **active control** to maintain contact with the object.
- **The whisker might slip or detach** if the surface is uneven or rough or when sharp corners are encountered.
- The whisker **must maintain a certain deflection profile** to ensure accurate measurements.
- The whisker mustn't be compressed or bent too much, as this can lead to damage or wear.
- The whisker can bend in the wrong direction, preventing further exploration until it is repositioned.

For platforms integrating multiple whiskers, additional challenges arise:

- The platform must be aware of its size and shape so as not to collide with the environment.
- The platform must be able to **navigate in confined spaces**, such as tunnels or narrow passages.
- The platform must follow the above-mentioned requirements for each whisker.

To summarize, a robust whisker array and active control system are required.

1.4 Overview of the Thesis

This thesis focuses on the use of **whisker sensors for active tactile perception in unstructured environments**. We aim to develop a control algorithm that allows a robotic platform to **actively explore and reconstruct its environment using whisker sensors**. The ultimate goal is to integrate the whisker arrays into a robotic rat and let it **navigate autonomously in unstructured environments**. This thesis assumes that the navigation and whisker deflection happen in 2D and that all the objects are immovable.

1.4.1 Simulation

First, a **simulation backed by MuJoCo [TET12]**, a physics simulation framework maintained by Google DeepMind, is developed to test the proposed control algorithms. The whisker shaft is simulated as a composite of 40 elastic cables. The whisker's contact with the objects is

assumed to be frictionless. A massive platform with two integrated whiskers from the left and right sides is set up for active control. Whiskers are tilted by 15° away from the nose to avoid the whisker tips getting stuck in the walls during the tunneling policy testing due to whiskers approaching the wall head-on and not deflecting.

1.4.2 Control System

The control system is built upon specialized control policies for different tasks. It accepts platform position and rotation as inputs and outputs platform linear and angular velocity. It runs at a **frequency of 30 Hz**. Platform linear velocity always points at the target position, determined by the algorithm, with no smoothing or filtering. A **PID controller** driven by the platform yaw error is employed for angular velocity.

1.4.3 Swiping Policy

The basic building block is the **swiping policy**, which enables a single whisker to follow the smooth contour of an object. It balances between preserving optimal whisker deflection and tangentially following the object's surface. Without the desired deflection, the whisker tip will either detach or be measured with low accuracy. Tangential contour tracking is required to ensure a steady direction of movement, as deflection compensation does not imply a certain direction. To identify the geometry of the immediate object surface, a spline is constructed from the last 7 whisker tip points at a distance of at least 1mm from each other. The swiping policy is tested in simulation on various objects, including a disk, a rectangular box with rounded corners, and a complex object with moderate inflections. The swiping policy achieves **submillimeter accuracy** on all tests and is shown to pass angles of **up to 30° without detaching**.

1.4.4 Retrieval Policy

After that, we introduce the **retrieval policy**, which allows the whisker to reattach to the object if it detached at a sharp corner. It aims to reconstruct the edge contour and revert to the swiping policy. In order to do so, it tries for a contact a certain distance away from the edge, at a certain angle, until it comes into contact with the object. This contact at the opposite side of the edge is used to determine the edge angle. As edge reconstruction is required, the whisker swipes back to the edge until it detaches, and then the platform reorients and approaches the edge again, now properly aligned. Once a steady contact is established, the control is switched to the swiping policy. The retrieval policy is tested in simulation on various objects, including an octagon, a cube, a prism, and a wall. It consistently reattached the whisker to the object and reconstructed the edge contour, delivering an average reconstruction error of 1mm. The retrieval policy is shown to handle the **entire range of corner angles**, from 30° to 180° and to stick to the defined retrieval radius of 1cm. It generates a **stable platform trajectory**, only slightly diverging from the ideal trajectory, which would assume no whisker detachment and immediate reorientation at the edge.

1.4.5 Tunneling Policy

For navigation in confined passages, we develop the **tunneling policy**, aimed at keeping the platform at the tunnel's center at all times. It employs both whiskers to scan and recon-

struct the tunnel. The tunneling policy is based on the same ideas as the swiping policy; the whiskers are kept in contact with the walls while the platform is following the tunnel and avoiding collisions. Again, a spline is constructed to estimate the axis (centerline) of the tunnel and determine the surface tangent. The target position is then calculated to ensure optimal deflection of both whiskers and stable movement along the tunnel axis. The “pull” quantifies required deflection compensation in the normal direction. The tunneling policy is tested in simulation on various objects, including a smooth tunnel segment, a zigzag tunnel segment with angles up to 40°, and a round star-shaped tunnel loop. The tunneling policy is shown to reconstruct the contours of the tunnels with an average reconstruction error of 2mm. It can follow the tunnel axis with an average deviation of about 5mm for segments with no whisker detachment. It can navigate tunnels with sharp corners up to 40° without colliding with the walls – of course, only for otherwise smooth tunnels.

1.4.6 Governing Policy

A finite state machine is set up to combine the aforementioned specialized policies. The transitions occur at the change of the number of deflected whiskers. For instance, the swiping policy transitions to the tunneling policy if the second whisker becomes deflected. It is responsible for switching from free exploration to a swiping policy and handling whisker detachment inside tunnels, as both contacts are not always secured. No additional checks besides the basic functionality test are conducted.

1.4.7 Hardware Setup

In order to test the control system in real-world conditions, a whisker sensor array is designed and assembled. It supports 3 magnetically transduced whiskers on each side of the platform and features a mount for the Franka Emika Panda robotic arm equipped with a gripper. The whiskers made from nitinol wire are 7.5cm long and 0.2mm in diameter. Underneath them, a neodymium magnet is glued to the suspension. Adafruit MLX90393 Hall-effect sensor is placed 2–3mm away from the magnet and is used to measure the whisker deflection by detecting the change in the Y axis of the magnetic field. Whiskers are tilted by 15° away towards the nose to increase the contact search space in front of the platform.

1.4.8 System Infrastructure

The system infrastructure is designed to support real-time collection, processing, and visualization of sensor data. It consists of multiple services like message broker, database, controller, and dashboard web server. The dashboard backed by Grafana visualizes the sensor data from the whisker array in real time.

1.5 Key Contributions

The following key contributions are made in this thesis:

1. We propose **Retrieval policy** for object retrieval in the case of whisker detachment at sharp corners, managing the entire range of corner angles and ensuring edge contour

reconstruction. It involves edge angle resolution, whisking back, and reattaching the whisker to the object.

2. We develop a **Tunneling policy**, allowing navigation in tunnels for multi-whisker setups and show that it can maintain a centered trajectory in confined passages. It aims to keep the whiskers in contact with the walls while following the tunnel and avoiding collisions.
3. We present a new **whisker sensor array** design using magnetically transduced whiskers, with 3 whiskers on each side of the platform.
4. Additionally, we prepare a **simulation framework** for the whisker control system to automate the testing of the control policies. MuJoCo physics simulation framework is used to simulate the whisker sensor and the environment.

Chapter 2

Related Works

This chapter overviews recent advances in tactile sensing and active exploration strategies. It reviews whisker-inspired sensor designs—including magnetically transduced, piezoresistive, and Micro-Electro-Mechanical Systems (MEMS) based implementations—and examines their operating principles, advantages, and challenges. Additionally, the chapter discusses how these sensors are integrated into active tactile exploration tasks, such as contour reconstruction and environment manipulation.

2.1 Whisker-Inspired Tactile Sensors

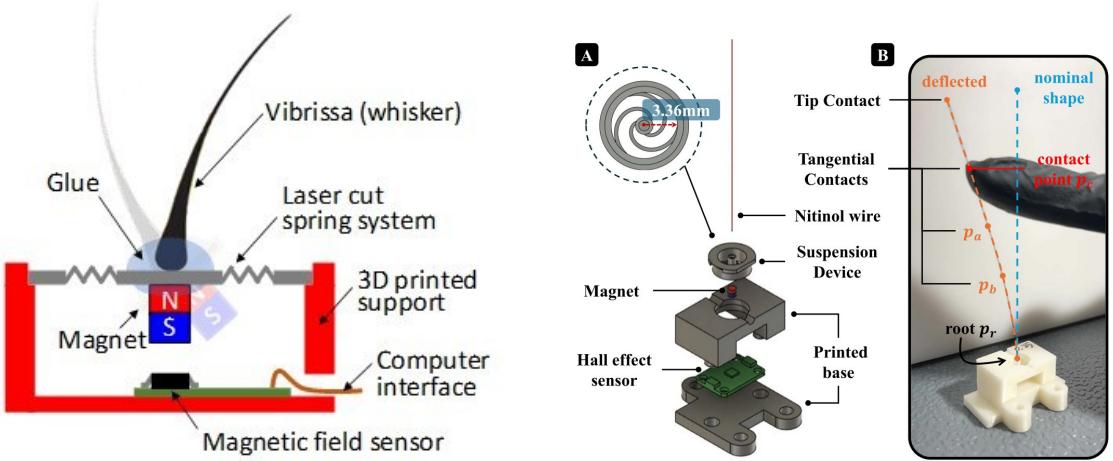
Several structural designs have been proposed for whisker sensors. They mainly differ in their transduction principle: how the normal contact force is transferred to the sensor base. The most common designs use passive, flexible cantilever whiskers integrated with strain gauges or piezoresistive elements to measure deflections. Capacitive and optical variants also exist but are less widely used. Additionally, although less common in the industry, magnetically transduced whiskers offer a low-cost and easily manufacturable alternative [Kim+19].

2.1.1 Magnetically Transduced Whisker Sensors

Although magnetically transduced whisker sensors have yet to be widely adopted, they provide high precision and sensitivity, making them suitable candidates for tactile sensing. Kim et al. [Kim+19] developed a magnetically transduced whisker sensor using a small permanent magnet attached to a flexible, cantilevered whisker. Figure 2.1a illustrates this sensor design. Its operating principle is straightforward: when the whisker contacts an object, deflection shifts the magnet embedded in a compliant, spring-like suspension relative to a magnetic sensor. Therefore, the magnetic field strength is proportional to the moment applied at the whisker base. Calibration is performed by measuring sensor output at various deflection angles and fitting the data using Gaussian process regression. One advantage is easy waterproofing, as the whisker does not directly contact the sensor. A limitation, however, is the rigidity of the whisker, complicating surface tracking.

Dang et al. [Dan+25] introduced a magnetically transduced whisker sensor employing a flexible nitinol wire as the whisker. They proposed a suspension device comprising three integrated, flexible spiral arms. Their sensor structure, shown in Figure 2.1b, resembles the design by Kim et al. They assume tip contact and model the tip position based on the whisker's deflection profile, parameterized using the one-dimensional measurement from a Hall sensor

along the respective axis. A limitation of their approach is the indistinguishability of tip and tangential whisker contacts, as both produce identical deflection profiles at the whisker base.



(a) Magnetically transduced whisker sensing mechanism schematic by Kim et al. [Kim+19].

(b) Structure of whisker-inspired tactile sensor by Dang et al. [Dan+25].

Figure 2.1: Side-by-side comparison of whisker sensor designs.

2.1.2 Piezoresistive Whisker Sensors

Piezoresistive whisker sensors are the most common type, notably applied in marine robotics to detect water flow and pressure variations.

Guo et al. [Guo+24] developed a Piezoelectric Wavy Whisker Sensor (PWWS) inspired by seal whiskers. The sensor comprises a flexible, waterproof Polydimethylsiloxane (PDMS) body and a thin sensing layer of Polyvinylidene Difluoride (PVDF). PDMS, a rubber-like material, protects the sensor from water, while PVDF is a polymer generating a small electrical voltage when bent or pressed. When water flows around an object, vortices form, pushing against the whisker, bending the PVDF layer, and producing a voltage as depicted in Figure 2.2. The generated voltage correlates with water flow parameters, enabling tracking of underwater disturbances.

2.1.3 MEMS-based Whisker Sensors

Wei et al. [Wei+19] developed a MEMS-based biomimetic whisker sensor replicating the tactile sensing mechanism of rats. The sensor is integrated onto a small silicon chip (6.8 mm^2) containing four identical sensing units. Each unit mimics the follicle arrangement of a rat's whisker, featuring a flexible whisker shaft attached to a central hub and four beams. Piezoresistors implanted on these beams form a Wheatstone bridge. Whisker bending from contact changes the resistance, producing an electrical signal. The sensor fabrication uses standard MEMS processes, and the resulting structure is shown in Figure 2.3. Production begins with a Silicon-on-Insulator (SOI) wafer, followed by ion implantation, insulating and metal layer deposition, and beam etching. Experiments indicate the sensor reliably measures contact distances (30 mm to 40 mm), recognizes object shapes (round, flat, beveled), and differentiates textures. Thus, this whisker sensor has significant potential for integration into robotic systems requiring tactile sensing.

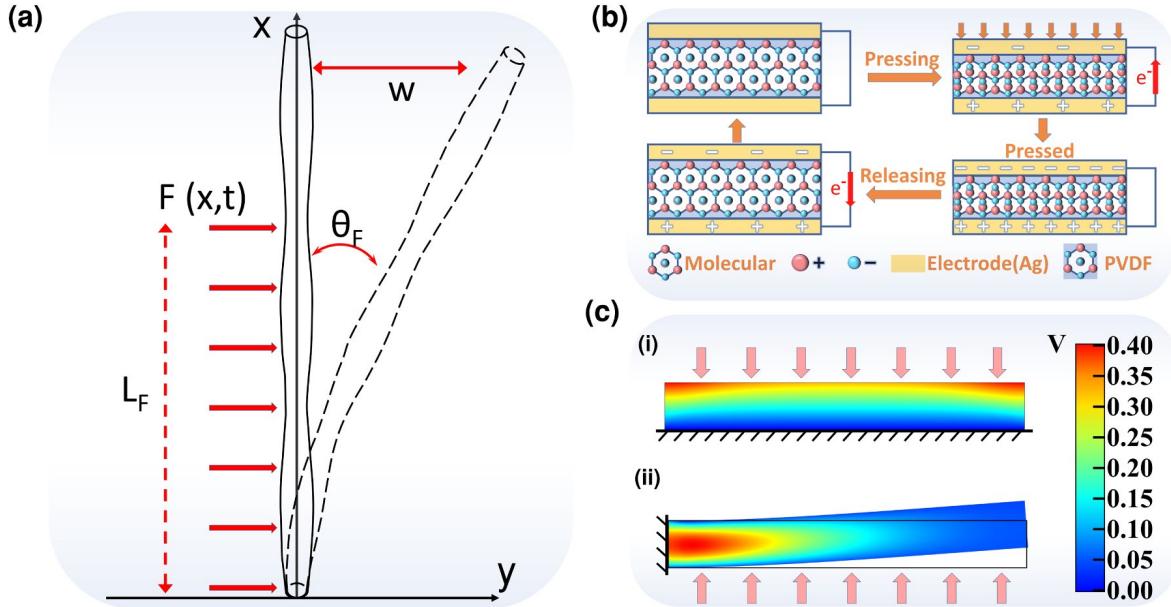


Figure 2.2: Operating principles of the PWWS by Guo et al. [Guo+24]. (a) Schematic of whisker deflection under force. (b) Cycle of electrical signals generated by the piezoelectric sensing unit. (c) Simulation of piezoelectric voltage within PVDF material under various constraints when subjected to external forces.

2.1.4 Contour Reconstruction

Whisker sensors are particularly suited to contour reconstruction due to their precise contact localization capabilities. However, two main issues affect their application:

1. Detachment or slippage of the whisker due to varying surface textures.
2. Inability to distinguish between tangential and normal contact.

Detachment or slippage issues can be mitigated using active control policies that actively reestablish whisker contact upon detachment. Distinguishing tangential from normal contacts can be addressed by additionally considering torque measurements alongside bending moments, as described by [HRH17], but this approach is considerably more complex for practical implementation.

2.2 Active Tactile Exploration

Active tactile exploration is a technique in robotics that gathers environmental information through touch. Typical applications include contact localization [HRH17], contour reconstruction [Lin+22], object recognition [Xia+22], and manipulation [Bro+24]. Whisker sensors, discussed in the previous section, offer a promising solution for these tasks.

2.2.1 Contour Reconstruction

Research on whisker-based active tactile exploration includes the work by Dang et al. [Dan+25], which introduces both a whisker sensor and an accompanying control system for active tac-

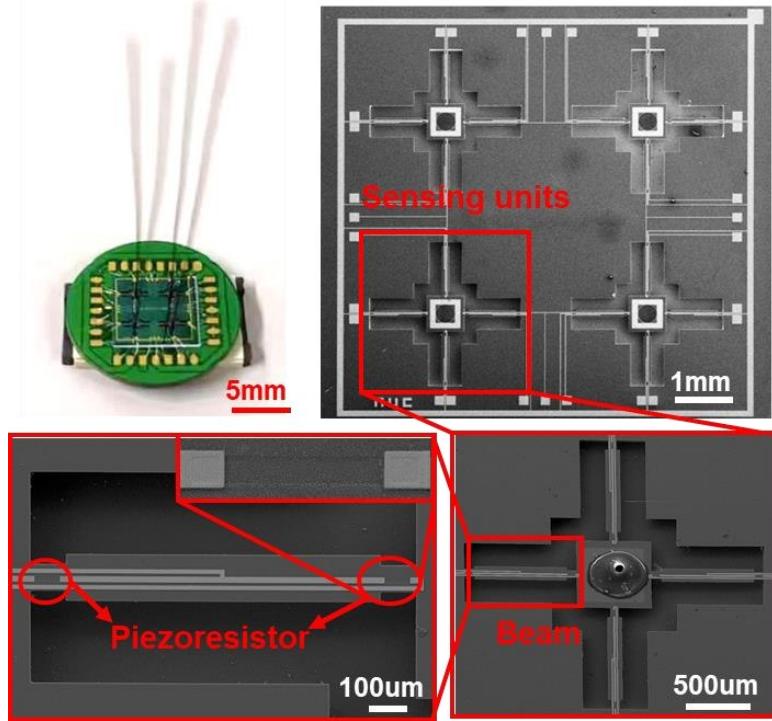


Figure 2.3: SEM image of the MEMS-based biomimetic whisker sensor by Wei et al. [Wei+19].

tile exploration. In their work, the whisker is actuated by a Franka Emika Panda robotic arm for object contour reconstruction. The control approach involves the following steps:

1. Hall sensor data from the whisker sensors is processed to determine the whisker tip position, reducing noise using Kalman filtering.
2. A spline is fitted through the last N whisker tip position measurements to calculate the curvature of the whisker.
3. The desired whisker rotation aligns with the surface normal at the next predicted contact point from the spline.
4. A PID controller maintains a constant whisker deflection profile. The controller adjusts the whisker's normal velocity based on the deflection error, compressing or relaxing the whisker as needed by moving along the surface normal.
5. The tangential velocity component is adjusted to keep the total velocity constant.

This approach achieves excellent results, reconstructing smooth surfaces with submillimeter accuracy. Small sharp angles can also be navigated, provided the whisker does not detach.

However, whisker detachment remains a significant challenge, as sudden, unpredictable slips inevitably occur, necessitating a retrieval mechanism. Another limitation of the approach of Dang et al. is the inherent instability in contour tracking direction—small disturbances may cause unintended reversal in exploration direction. Nevertheless, this control policy forms a solid foundation for developing more robust solutions, including retrieval strategies.

2.2.2 Environment Manipulation

Complex manipulation tasks can be performed when the sensory array is reliable enough. Brouwer et al. [Bro+24] have developed tactile-informed action primitives designed to help

robots reach targets in densely cluttered environments without getting stuck. Their approach introduces two simple motion patterns: the “burrow” primitive adds a gentle, snaking side-to-side motion to clear a path, while the “excavate” primitive makes a spiraling, scooping motion that pushes obstacles aside. Both primitives are depicted in Figure 2.4. By integrating these primitives into a closed-loop control strategy informed by soft tactile sensors on the robot’s finger-like end-effector, the system can sense contact forces and adjust its motion in real-time. Both simulation and hardware experiments show that this strategy dramatically improves performance—reducing both the distance to the goal and the time to complete the reach—and significantly increases the success rate compared to traditional straight-line motion.

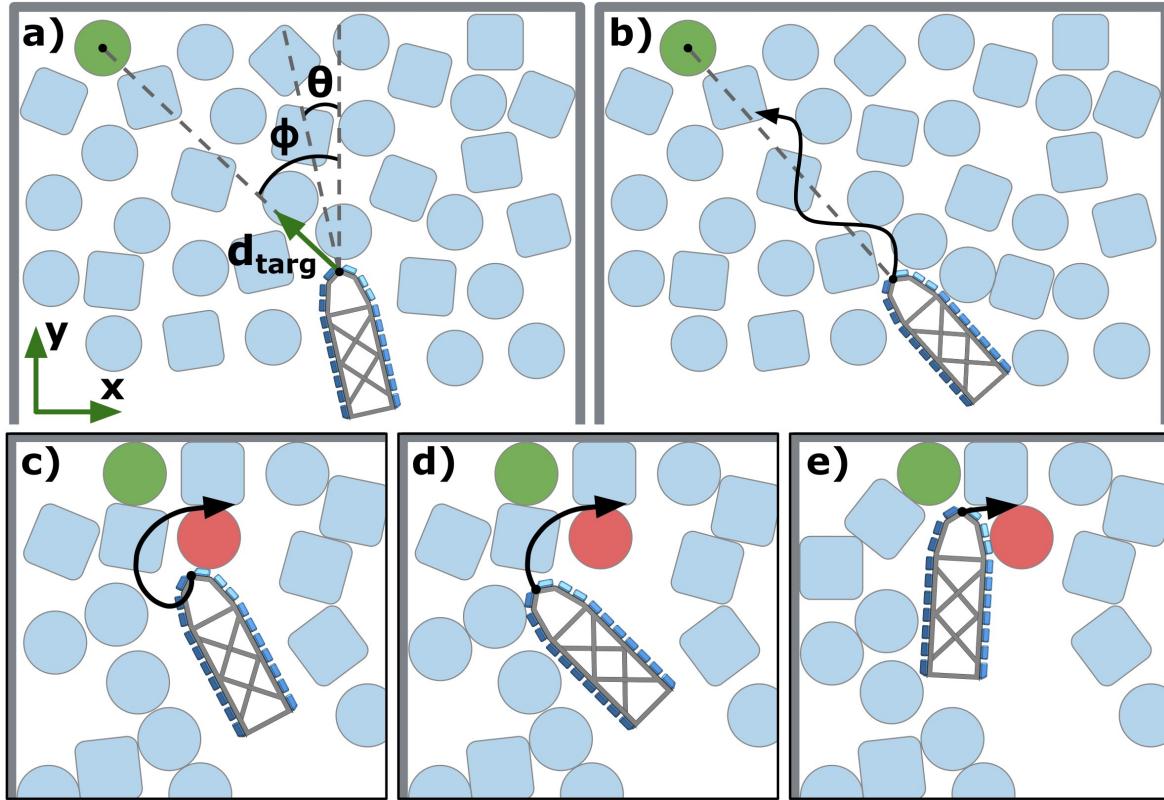


Figure 2.4: Schematic visualizations of a) straight line control, b) burrowing action primitive, and c–e) a sequence showing the progression of a clockwise excavate action primitive, by Brouwer et al. [Bro+24].

Chapter 3

Hardware Design

This chapter details the hardware design underlying the whisker sensor system. It covers the whisker mechanical structure, sensor integration, and whisker array assembly.

3.1 Magnetically Transduced Whisker Sensor

The structure of a single whisker sensor is shown in Figure 3.1.

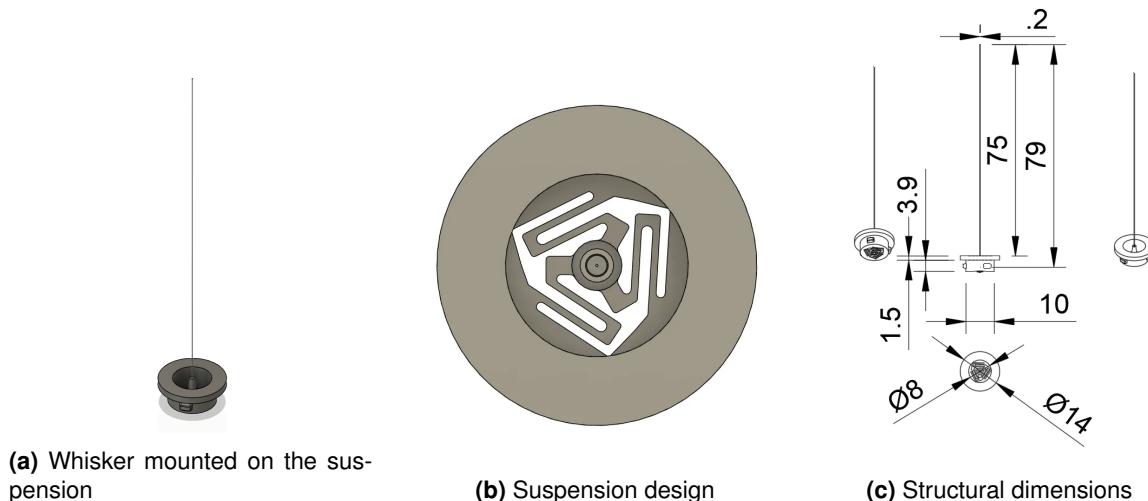


Figure 3.1: Single whisker sensor.

The whisker consists of:

- A flexible nitinol wire shaft (0.25 mm diameter, 75 mm length).
- A suspension system fabricated via 3D printing using Polylactic Acid (PLA) filament.
- An Adafruit MLX90393 magnetic sensor, configured to measure magnetic flux changes with a resolution of $0.15 \mu T/LSB$.

The whisker shaft is inserted into and glued to the suspension system. A neodymium permanent magnet, axially magnetized and aligned with the whisker shaft, is placed directly beneath the suspension. When the whisker deflects, the magnet rotates, altering the sensed magnetic field.

The suspension is designed to allow slight rotation of the whisker shaft while limiting axial movement. Three spring-like arms hold the whisker securely in place.

3.2 Whisker Platform

A whisker platform was developed to enable multi-whisker exploration and contour reconstruction. It consists of a triangular body with a 30° nose angle, two side clamps holding up to three whiskers, and a mount compatible with the Franka Panda robotic arm gripper. The platform, depicted in Figure 3.2, measures 90 mm × 60 mm × 35 mm and is 3D printed using PLA filament. Figure 3.3 shows the printed and assembled platform, designed to roughly resemble a rat's shape.

The MLX90393 sensors are secured by side clamps designed with cuts to accommodate 4-pin JST connectors on both sides. All components are assembled using M2 and M3 screws.

The clamps and suspension bases are designed to allow whiskers to be quickly plugged into place and locked by slight rotation, enabling easy replacement. No external power supply beyond the JST connector is necessary.

Whiskers are mounted at a 15° angle relative to the platform's mirror plane, slightly pointing forward to enlarge the contact search area. However, this orientation poses the risk of head-on collisions during tunneling, potentially causing excessive deflection, permanent deformation, or breaking. Implementing safeguards within the control system to mitigate this is beyond the scope of this thesis. Thus, in simulations, whiskers are oriented slightly backward to avoid these issues.

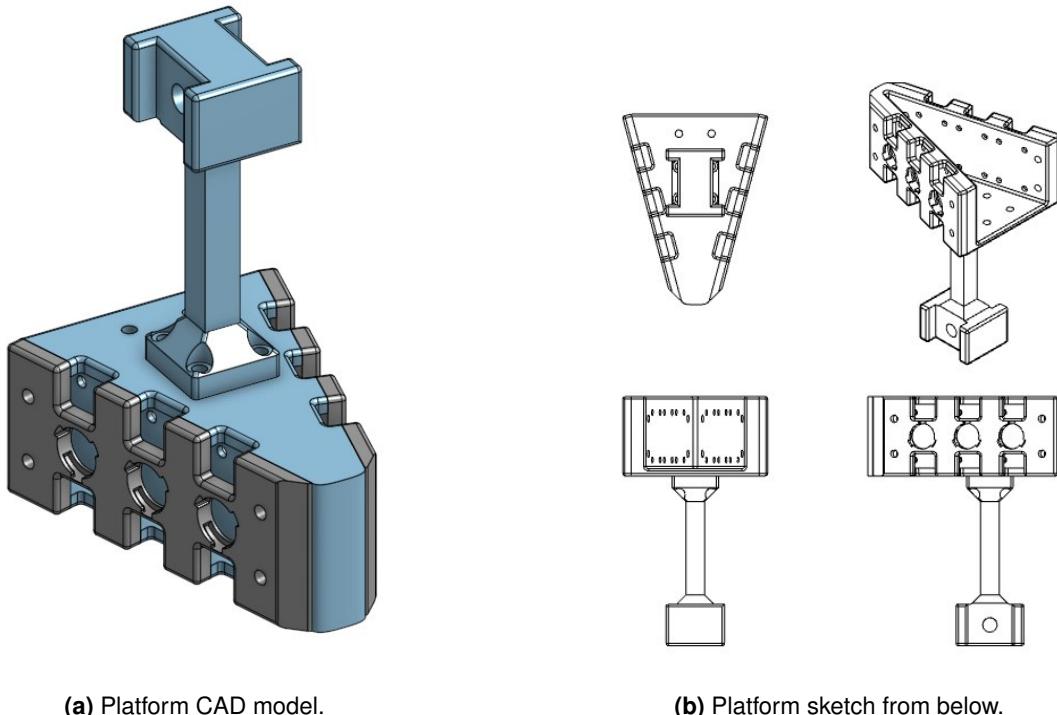


Figure 3.2: Whisker platform CAD model and sketch.

3.3 Data Acquisition

The Adafruit MLX90393 development board depicted in Figure 3.4, which hosts a Hall sensor, is placed approximately 2 mm to 3 mm beneath the suspension-mounted magnet. The sensor

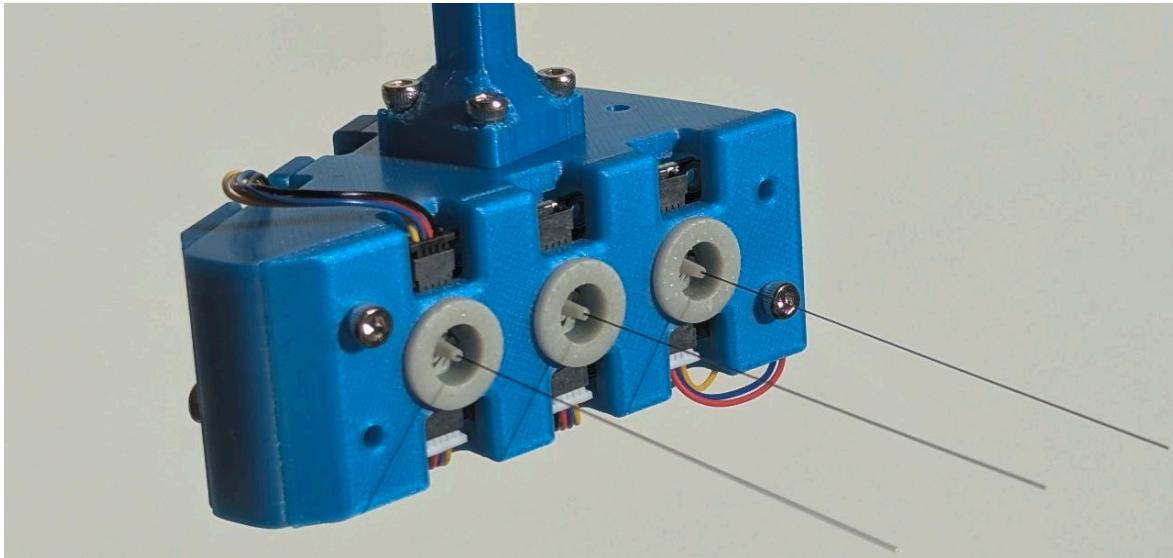
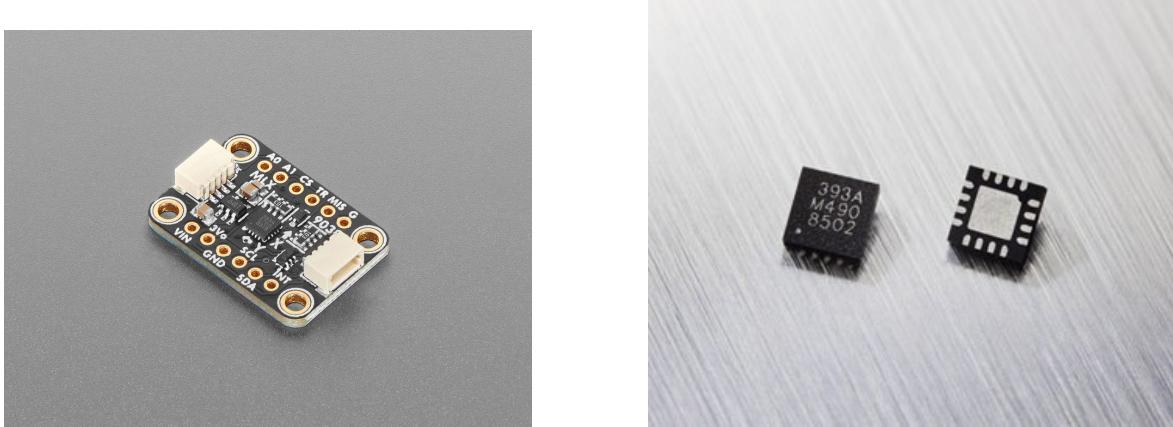


Figure 3.3: Assembled whisker platform with three whiskers and robotic arm mount.

utilizes a 16-bit Analog-to-Digital Converter (ADC) providing magnetic flux density measurements along X , Y , and Z axes. It is configured to measure with a resolution of $0.15 \mu\text{T}$. Only the Y -axis component is used in practice, as it uniquely captures rotation caused by planar whisker deflections. The sensors communicate via the I²C protocol, operating in slave mode. When multiple sensors are used, they are connected in a daisy-chain arrangement.

Two drivers have been developed for the MLX90393 sensors: one in C++ for the ESP32 microcontroller and another in Python for the Raspberry Pi 5. Both implementations follow the official MLX90393 Triaxis Magnetic Node specification [Mel17]. Continuous burst mode allows reading sensor data at regular intervals, reaching a data acquisition rate of 300 Hz. This high rate facilitates effective signal filtering, especially during rapid whisker movements. The control loop operates at approximately 30 Hz, constrained by the actuator's responsiveness.



(a) Adafruit MLX90393 dev board (25.4 mm × 17.8 mm × 1.6 mm)

(b) Melexis MLX90393 chip (3.1 mm × 2.9 mm × 1 mm)

Figure 3.4: MLX90393 sensor hardware from [Mel17]

The deflection model from the work of Dang et al. [Dan+25] converts the sensor readings into whisker deflection angles. The model characterizes the deflection profile using two key components: the root position, denoted as p_r , and a measurement model that maps the

tangential contact location $c_t = (x_{bt}, y_{tb})$ (expressed in the sensor base frame) to a deflection measurement z . In this setup, the bending of the whisker shaft is proportional to the rotation of its suspension system, and the sensor extracts the flux change along the Y-axis—where the variation is most significant—to represent z .

To calibrate this model, the sensor is mounted on a two-dimensional motorized stage, which moves in 3 mm steps along both axes. 180 data sets are collected by recording the magnetic sensor data and the tangential contact positions as the stage moves in a grid-like pattern. These data are then used to fit a fifth-order bivariate polynomial, yielding a function $z_c = f(x_b, y_b)$ that describes the deflection arc shape at a given moment. Note that the default root position is set as $p_r = (0, 0)$, even though slight linear displacements at the center of rotation may occur due to the current spring design—a factor identified for future improvement.

In practice, the provided model covers the operational deflection range (-5×10^{-5} rad to 5×10^{-4} rad) and starts to diverge from the actual deflection profile at larger angles. The polynomial is stabilized by clipping values that exceed realistic deflection limits to rectify this. Figure 3.5 illustrates this tweaked deflection profile.

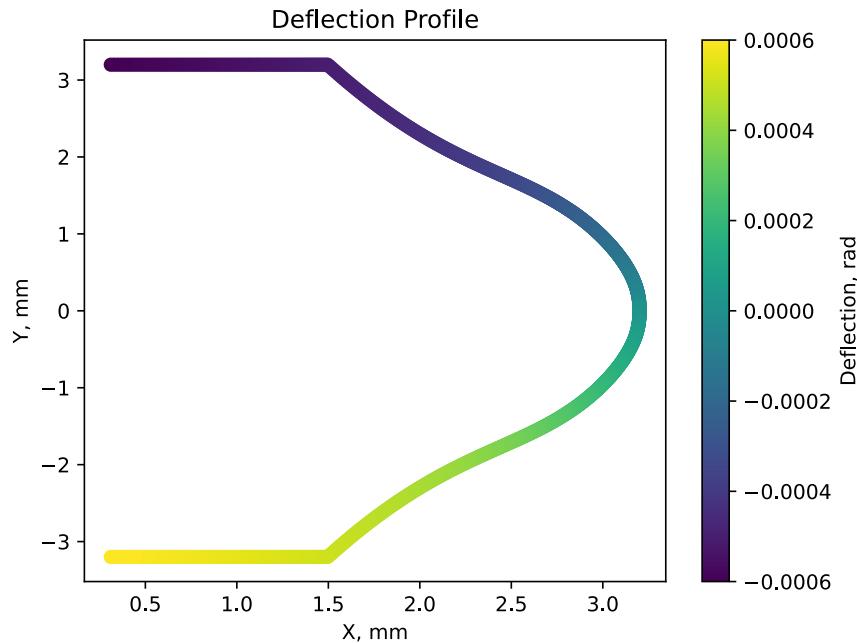


Figure 3.5: Whisker deflection profile model.

Chapter 4

Control Algorithms

This chapter presents the control strategies that enable the whisker sensor system to accurately reconstruct object contours through tactile exploration. It details a Finite State Machine (FSM) based framework that integrates multiple control policies: swiping for continuous contour tracking, retrieval for re-establishing lost contacts and tunneling for balanced navigation between surfaces. The algorithms leverage real-time sensor feedback, spline-based trajectory planning, and PID-based body motion control to ensure smooth, adaptive interactions with diverse object geometries.

4.1 Overview

The whisker control algorithm is developed to reconstruct an object's contour by swiping along its surface. Before we delve into the details of the control algorithm, we need to mention that its implementation is available on GitHub¹.

The following key requirements were determined for the control model design:

1. **Precise Contour Reconstruction:** The algorithm must accurately capture the local shape at the point of contact, ensuring that the detailed profile of the object is recorded.
2. **Accurate Curve Following:** It must adeptly follow both smooth curves and sharp angles, all while maintaining an optimal whisker deflection. The constant whisker deflection is crucial for a predictable and small whisker tip position resolution by the deflection model.
3. **Robust Detachment Recovery:** Sudden changes in the object's geometry can cause the whisker to detach. The system is, therefore, designed to quickly detect such events and recover contact with minimal disruption.
4. **Collision Avoidance:** Considering the platform's dimensions, the algorithm incorporates measures to prevent collisions with the object during contour tracking.

Assumptions: The development of this control strategy is based on the following assumptions:

1. The contour is captured in a 2D (xy) plane, meaning that both the platform and whiskers operate with planar motion.
2. All objects are rigid and remain stationary during the contact process.

¹<https://github.com/SVDouble/RatteChan>

3. Contact occurs exclusively at the tip of the whisker.
4. Forces exerted by the whiskers do not significantly affect the platform's motion.
5. The platform's absolute position in the world frame is continuously known.
6. The initial trajectory of the platform ensures that at least one whisker will eventually come into contact with the object.
7. Although multiple objects may be present, the system focuses on a single object at any given time.

To meet these requirements, the control algorithm is implemented as a FSM that sequences the following policies:

1. **Swiping Policy:** The whisker performs a continuous swiping motion along the object's curve while maintaining an optimal deflection profile.
2. **Retrieval Policy:** In the event of whisker detachment, this policy analyzes the detachment reason, re-establishes contact, and resumes the swiping motion, ensuring contour continuity.
3. **Tunnelling Policy:** When operating between two surfaces, the algorithm maintains a centered trajectory, ensuring balanced contact on both sides.

This structured approach enables the control system to dynamically adapt to varying contact scenarios.

4.2 Variables

It is necessary to consider the variables of the control algorithm. Inputs come from three main sources:

1. **Platform Sensors:** Provide the platform's position, orientation, and whisker deflection.
2. **Geometric Configuration:** Information such as whisker placement.
3. **Algorithm Configuration:** Parameters like the whisker deflection threshold.

The output variables define the target linear and angular velocities of the platform. A summary of these variables is provided in Table 4.1.

4.3 Data Preprocessing

The measured whisker deflection δ_i^t is noisy due to vibrations and transient disturbances. To reduce this noise, a low-pass Butterworth filter is applied. The filter is initially seeded with the neutral deflection and updated with each new measurement.

Name	Values	Source	Description
Control Inputs: Platform			
${}^w \mathbf{r}^t$	$\mathbb{R}^2, [x, y]^T$	Measured	Radius vector in world coordinates.
${}^w \alpha^t$	\mathbb{R}	Measured	Yaw angle (orientation) in world coordinates.
v_{total}	\mathbb{R}^+	Configuration	Total platform velocity.
Control Inputs: Whisker $i = 1, r$			
δ_i^t	$[-\delta_{i,\max}, \delta_{i,\max}]$	Measured	Deflection due to contact forces.
α_i^t	\mathbb{R}	${}^w \alpha^t + \alpha_{i,\text{body}}$	Yaw angle (orientation) of the whisker base in world coordinates.
$orient_i^t$	$\{-1, 0, 1\}$	$sgn(\delta_i^t) \cdot side_i$	Valid swipe orientation with current deflection: -1 for clockwise, 0 for undefined, and 1 for counterclockwise.
$side_i$	$\{-1, 1\}$	$\begin{cases} -1, & \text{if } \alpha_{i,\text{body}} < 0, \\ 1, & \text{otherwise.} \end{cases}$	Platform side where the whisker is fixed, -1 is left and 1 is right
$\delta_{i,\text{thr}}$	\mathbb{R}^+	Deflection Model	Deflection threshold value for contact detection ($\delta_{i,\text{thr}} \ll \delta_{i,\max}$).
$\delta_{i,\text{target}}$	$[-\delta_{i,\max}, \delta_{i,\max}]$	Deflection Model	Target deflection value for small reconstruction error.
${}^w \mathbf{r}_{i,\text{body}}$	\mathbb{R}^2	Robot Geometry	Offset from the platform center to the whisker base.
$\alpha_{i,\text{body}}$	$[-\pi, \pi)$	Robot Geometry	Angle for whisker placement relative to the platform.
Control Outputs			
${}^w \mathbf{v}^t$	$\mathbb{R}^2, [v_x, v_y]^T$	-	Linear velocity vector in world coordinates.
${}^w \omega^t$	\mathbb{R}	-	Angular velocity (yaw) in world coordinates.

Table 4.1: Overview of input and output variables used in the control system.

4.4 Body Motion Control

Each control policy produces a target position for either the platform body or the whisker, which is then executed by the actuators. To achieve accurate motion, the body motion control algorithm computes the necessary linear and angular velocities to reach the desired position. A PID controller is used to generate the target angular velocity based on the difference between the current and desired orientations. Meanwhile, the linear velocity is maintained at a constant magnitude so that the resulting velocity vector is directly determined by the desired direction (see Algorithm 1). In cases where the target position is specified for the whisker rather than the body, the algorithm applies a coordinate transformation to adjust the linear velocity accordingly. This transformation compensates for the offset between the whisker and the body frame, ensuring the control command is executed correctly (see Algorithm 2).

4.5 Swiping Policy

The swiping policy aims to produce a smooth motion of the whisker along the object's contour while collecting high-precision shape data. To achieve this, the whisker must maintain

Algorithm 1 Steer the Platform to Target Position and Orientation

- 1: Require ${}^w\mathbf{r}^{t+1}$, ${}^w\alpha^{t+1}$
 - 2: ${}^w\omega^{t+1} \leftarrow \text{PID}({}^w\alpha^{t+1} - {}^w\alpha^t)$
 - 3: ${}^w\mathbf{v}^{t+1} \leftarrow v_{\text{total}} \cdot \frac{{}^w\mathbf{r}^{t+1}}{\|{}^w\mathbf{r}^{t+1}\|}$
 - 4: Return ${}^w\mathbf{v}^{t+1}$, ${}^w\omega^{t+1}$
-

Algorithm 2 Steer Whisker to Target Position and Orientation

- 1: Require ${}^w\mathbf{r}_{\text{wsk}}^{t+1}$, ${}^w\alpha^{t+1}$
 - 2: $({}^w\mathbf{v}^{t+1}, {}^w\omega^{t+1}) \leftarrow \text{steer_body}({}^w\mathbf{r}_{\text{wsk}}^{t+1}, {}^w\alpha^{t+1})$
 - 3: ${}^w\mathbf{r}_{\text{corr}} \leftarrow [0, 0, {}^w\omega^{t+1}] \times \mathbf{r}_{\text{wsk, body}}$ ▷ Correct for whisker offset (pivot shift)
 - 4: ${}^w\mathbf{v}^{t+1} \leftarrow v_{\text{total}} \cdot \frac{{}^w\mathbf{v}^{t+1} + {}^w\mathbf{r}_{\text{corr}}}{\|{}^w\mathbf{v}^{t+1} + {}^w\mathbf{r}_{\text{corr}}\|}$
 - 5: Return ${}^w\mathbf{v}^{t+1}$, ${}^w\omega^{t+1}$
-

an optimal deflection profile because the accuracy of the deflection model depends on the current deflection. This requires a balance between accurately following the object's curvature and preserving the target deflection. The idea is demonstrated in Algorithm 3. For contour following, a B-Spline is used to represent the object's contour by incorporating the latest whisker tip position at each control iteration:

$$\mathbf{r}_j = (x_j, y_j), \quad j = 0, \dots, n - 1.$$

The local contour is given by

$$S(u) = \sum_{j=0}^{n-1} r_j B_{j,k,t}(u),$$

where $B_{j,k,t}(u)$ denotes the B-Spline basis functions of degree k defined on the knot vector t . We parameterize the keypoints uniformly by setting

$$u_k = 1 + \frac{k}{n-1}, \quad k = 0, 1, \dots, n-1.$$

This B-Spline formulation provides a smooth approximation of the object's contour while allowing local control over the curve shape. The keypoint collection is given in Algorithm 3 (lines 5–7 highlighted in yellow).

The tangent at the end of the spline is then used to determine the desired whisker orientation (see Algorithm 3, lines 12–14 highlighted in blue). To account for the whisker's deflection, the target whisker base-tip offset is calculated from the deflection model and compared with the desired target deflection. The resulting whisker position is derived as a weighted sum of the vector that minimizes the deflection error and ensures the spline following (see Algorithm 3, lines 18–19 highlighted in green). For transient conditions—when the whisker deflection is below a predefined threshold or the spline has not yet been constructed—the swiping policy retains the existing control values.

The contour is reconstructed by sequentially connecting the spline keypoints. The swiping motion stops when the latest keypoint is closer to the starting position than the distance between consecutive keypoints and when a predetermined trajectory length has been achieved.

Algorithm 3 Swiping Policy

```

1: If  $|\delta_{\text{wsk}}^t| < \delta_{\text{wsk,thr}}$  Then           ▷ Check that the whisker touches the object
2:   Return  ${}^w\mathbf{v}^t, {}^w\omega^t$ 
3: End If
4:
5:    ${}^s\mathbf{r}_{\text{tip}}^t \leftarrow \text{wsk.defl\_model}(\delta_{\text{wsk}}^t)$ 
6:    ${}^w\mathbf{r}_{\text{tip}}^t \leftarrow {}^w\mathbf{r}^t + {}^w\mathbf{r}_{\text{wsk,body}} + \mathbf{R}_{xy}^2({}^w\alpha_{\text{wsk}}^t) \cdot {}^s\mathbf{r}_{\text{tip}}^t$ 
7:   wsk.spline.add_keypoint( ${}^w\mathbf{r}_{\text{tip}}^t$ )
8: If not wsk.spline.has_enough_points() Then      ▷ Check that the spline is complete
9:   Return  ${}^w\mathbf{v}^t, {}^w\omega^t$ 
10: End If
11:
12:    ${}^w\boldsymbol{\tau}_{\text{spline}}^t \leftarrow \frac{\text{wsk.spline}(u=u_{k1}) - \text{wsk.spline}(u=u_{k0})}{\|\text{wsk.spline}(u=u_{k1}) - \text{wsk.spline}(u=u_{k0})\|}$           ▷ Object surface tangent
13:    ${}^w\theta_{\text{spline}}^t \leftarrow \arctan2({}^w\boldsymbol{\tau}_{\text{spline}}^t)$                                 ▷ Object surface angle
14:    ${}^w\alpha^{t+1} \leftarrow {}^w\theta_{\text{spline}}^t$ 
15:    ${}^s\mathbf{r}_{\text{tip,target}}^t \leftarrow \text{wsk.defl\_model}(\delta_{\text{wsk,target}} \cdot \text{sgn}(\delta_{\text{wsk}}^t))$     ▷ Desired whisker base-tip offset
16:    $\Delta\mathbf{r}_{\text{tip}}^t \leftarrow \mathbf{R}_{xy}^2({}^w\alpha_{\text{wsk}}^t) \cdot ({}^s\mathbf{r}_{\text{tip,target}} - {}^s\mathbf{r}_{\text{tip}}^t)$           ▷ ( $\forall t {}^w\mathbf{v}^{t+1} \downarrow \Delta\mathbf{r}_{\text{tip}}^t \implies \delta_{\text{wsk}}^t \xrightarrow[t \rightarrow \infty]{} \delta_{\text{wsk,target}}$ )
17:    ${}^s\mathbf{r}_{\text{tip,neutral}}^t \leftarrow \text{wsk.defl\_model}(0)$                                 ▷ Undeformed whisker base-tip offset
18:    $w_{\text{defl}}^t \leftarrow \frac{\|\Delta\mathbf{r}_{\text{tip}}^t\|}{\|{}^s\mathbf{r}_{\text{tip,target}} - {}^s\mathbf{r}_{\text{tip,neutral}}\|}$ 
19:    ${}^w\mathbf{r}_{\text{wsk}}^{t+1} \leftarrow {}^w\mathbf{r}_{\text{wsk}}^t + w_{\text{defl}}^t \cdot \frac{-\Delta\mathbf{r}_{\text{tip}}^t}{\|\Delta\mathbf{r}_{\text{tip}}^t\|} + (1 - w_{\text{defl}}^t) \cdot \frac{{}^w\boldsymbol{\tau}_{\text{spline}}^t}{\|{}^w\boldsymbol{\tau}_{\text{spline}}^t\|}$           ▷ Weighted sum
20:    $({}^w\mathbf{v}^{t+1}, {}^w\omega^{t+1}) \leftarrow \text{steer\_wsk}({}^w\mathbf{r}_{\text{wsk}}^{t+1}, {}^w\alpha^{t+1})$ 
21:   Return  ${}^w\mathbf{v}^{t+1}, {}^w\omega^{t+1}$ 

```

4.6 Retrieval Policy

The retrieval policy re-establishes contact with the object when the whisker detaches, minimizing platform movement and resuming swiping as quickly as possible. Detachment can occur due to:

- Object Surface:** The curve's angle is too sharp for the whisker to follow.

- Slippage:**

- (a) **Forward Slippage:** Occurs when the angle increases, but the whisker remains momentarily engaged due to traction.
- (b) **Backward Slippage:** Occurs when an inadequate deflection profile leads to tension buildup.

The main challenge is determining the actual object profile for continued navigation. Since the swiping policy handles minor slippage, the focus is on detachment due to the object's surface. The expected contact angle ranges from approximately 10–20° (as the whisker moves from the deflected position toward neutral) up to 180°. A test point on the opposite side of the edge is used to determine the true edge angle, as the spline cannot be updated.

4.6.1 Premise

If the whisker remains detached beyond a predefined threshold, it is considered disengaged, and the retrieval policy takes over from the swiping policy. Figures 4.2a and 4.2b illustrate the retrieval premise.

Angle Resolution

The angle resolution strategy operates as follows:

1. The whisker returns to a starting angle-detection position.
2. It targets the nearest out of all potential contact points at a fixed distance from the last known position; the locus of such points is a circle.
3. It moves sequentially from one candidate point to another until the contact is established, as shown in Figure 4.1. For every candidate point on the circle, the whisker's yaw is adjusted to contact the object at a predefined angle. The object's surface is assumed to be a straight line from the edge to the candidate point.
4. Once the contact is re-established, the edge angle is determined as an angle between the spline tangent before the detachment and the line connecting the edge and the contact point. The required deflection threshold to detect the contact is set high enough for the deflection model to have a sufficiently small error during the next step with such a deflection.

See Figures 4.2c and 4.2d for an illustration of the angle resolution strategy.

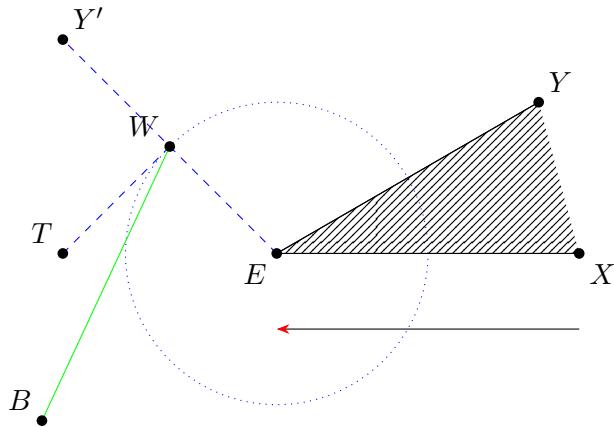


Figure 4.1: Angle Retrieval Strategy. Black $[XE]$, $[EY]$ — object surface, Green $[BW]$ — the whisker, Blue $[EY']$ — the potential adjacent surface of the edge and its normal $[WT]$, Blue circle — the targeted contact points.

4.6.2 Whisking Back to the Edge

The whisker moves back along the opposite side of the edge by linearly steering towards the edge. As the radius is small with respect to the length of the whisker, such trajectory doesn't affect the deflection much, and the deflection profile is therefore preserved. The selected radius guarantees that the distance from the contact point to the edge is sufficient to construct a new spline and predict the edge angle reliably. Whisking back is shown in Figure 4.2e.

4.6.3 Transition to Swiping

After analyzing the opposite side, the whisker intentionally detaches and repositions. It is repositioned with a slight overshoot of the target attachment angle to ensure proper alignment when contact is re-established. The whisker then reverses the direction of movement to re-engage with the object. Once the new spline is constructed on the opposite side, the control is transferred back to the swiping policy, thereby completing the retrieval maneuver. See Figure 4.2f, 4.2g and 4.2h for an illustration of the repositioning and approach.

4.7 Tunneling Policy

The tunneling policy ensures that the platform maintains a centered trajectory between two surfaces, which optimizes whisker contact on both sides. The idea is demonstrated in Algorithm 4. A midpoint spline is generated by computing keypoints defined as the midpoints between the left and right whisker contacts (Algorithm 4, lines 1–3 highlighted in yellow). This B-Spline is a continuous reference for steering the platform through tunnel-like environments. The target angle is determined by the tangent of the midpoint spline (Algorithm 4, lines 6–8 highlighted in blue). The target position is calculated as the current position plus the weighted sum of the spline tangent and deflection “pull” (Algorithm 4, lines 9–10 highlighted in green). This pull evaluates the difference between the current and target deflections as a ratio for both whiskers and shows whether normal movement is needed to correct the deflection.

Algorithm 4 Tunneling Policy

```

1:  ${}^w\mathbf{r}_{\text{tip},\text{r}}^t \leftarrow {}^w\mathbf{r}^t + {}^w\mathbf{r}_{\text{r,body}} + \mathbf{R}_{xy}^2({}^w\alpha_{\text{r}}^t) \cdot \text{r.defl\_model}(\delta_{\text{r}}^t)$ 
2:  ${}^w\mathbf{r}_{\text{tip},\text{l}}^t \leftarrow {}^w\mathbf{r}^t + {}^w\mathbf{r}_{\text{l,body}} + \mathbf{R}_{xy}^2({}^w\alpha_{\text{l}}^t) \cdot \text{l.defl\_model}(\delta_{\text{l}}^t)$ 
3: midpoint_spline.add_keypoint( $({}^w\mathbf{r}_{\text{tip},\text{r}}^t + {}^w\mathbf{r}_{\text{tip},\text{l}}^t)/2$ )
4: If not midpoint_spline.has_enough_points Then Return  ${}^w\mathbf{v}^t$ ,  ${}^w\omega^t$  End If
5:
6:  ${}^w\boldsymbol{\tau}_{\text{spline}}^t \leftarrow \text{midpoint\_spline}(u=1) - \text{midpoint\_spline}(u=0)$ 
7:  ${}^w\mathbf{n}_{\text{spline}}^t \leftarrow \mathbf{R}_{xy}^2(\pi/2 \cdot \text{orient}_{\text{r}}^t) \cdot {}^w\boldsymbol{\tau}_{\text{spline}}^t$ 
8:  ${}^w\theta_{\text{spline}}^t \leftarrow \text{arctan2}({}^w\boldsymbol{\tau}_{\text{spline}}^t)$ 
9:  $\text{pull}_{\text{total}} \leftarrow \text{clip}\left(\frac{\delta_{\text{r}}^t - \delta_{\text{r,target}}}{\delta_{\text{r,target}}} - \frac{\delta_{\text{l}}^t - \delta_{\text{l,target}}}{\delta_{\text{l,target}}}, \text{min}=-1, \text{max}=1\right)$ 
10:  ${}^w\mathbf{r}^{t+1} \leftarrow {}^w\mathbf{r}^t + {}^w\boldsymbol{\tau}_{\text{spline}}^t + \text{pull}_{\text{total}}/2 \cdot {}^w\mathbf{n}_{\text{spline}}^t$ 
11:  $({}^w\mathbf{v}^{t+1}, {}^w\omega^{t+1}) \leftarrow \text{steer\_body}({}^w\mathbf{r}^{t+1}, {}^w\theta_{\text{spline}}^t)$ 
12: Return  ${}^w\mathbf{v}^{t+1}$ ,  ${}^w\omega^{t+1}$ 
```

4.8 Finite State Machine

The policies are specialized, and only in combination can the whisker harness their full potential. A FSM combines these policies, with each policy represented as a state. Transitions

occur when specific events arise, such as “whisker has come into contact with the surface” or “whisker has become disengaged.” The state chart of the FSM for the discussed policies is depicted in Figure 4.3.

4.9 Additional features

Additional modifications can be applied to the control algorithm depending on the specific requirements of the task.

4.9.1 Retrieval without Edge Reconstruction

If the goal is to quickly analyze the object’s surface without precise edge reconstruction, the whisking step in the retrieval policy can be skipped so that the angle resolution stage directly leads to swiping on the opposite side.

4.9.2 Platform Tilt

The platform can be tilted slightly toward the object for improved stability during swiping (i.e., to handle sharper angles without separation). This requires a simple modification of the body motion control algorithm by adjusting the target orientation by a small angle. The tilt increases the duration of whisker contact with the surface as the platform maintains its direction. It is beneficial given that the spline is constructed only at the contact point and cannot predict the future curve far away.

4.9.3 Reverse Swiping

If it is necessary for the platform to backtrack its path, the swiping policy can be modified to follow the spline in the reverse direction. This is achieved by redefining the principal movement direction (normally aligned with the “nose” of the platform) in reverse mode. Consequently, the body motion control algorithm computes the linear velocity using an oppositely offset yaw. The retrieval policy is adjusted accordingly, as it depends solely on the position of the whisker base and is independent of the platform’s overall orientation.

4.9.4 Positioning of Whiskers at Different Angles

Improving the adhesion of the whisker to the object can be achieved by positioning the whisker at an angle relative to the platform while maintaining the platform’s tangential alignment with the object. This approach follows the same principle as tilting but preserves the platform’s angle relative to the object. This modification is particularly important in tunneling scenarios, where both whiskers must negotiate turns without becoming stuck. When entering a tunnel, the first whisker contacts the surface, followed by the second whisker. It is critical that the second whisker is positioned at an appropriate angle relative to the platform (typically slightly behind the main body). If the whisker points directly in the direction of movement, it may insert into surface cavities and bend as if the platform were moving in the opposite direction, risking damage due to excessive deflection. Positioning the whisker slightly backward prevents direct head-on contact with the tunnel wall.

4.9.5 Integration of Multiple Whiskers on Each Side

Multiple whiskers can be integrated on each side of the platform to:

- Increase the precision of surface resolution.
- Enable single-shot swiping without the need for retrieval, as the retrieval radius is smaller than the distance between two whisker bases.
- Detect contacts at a wider range by extending the search space.

4.10 Comparison to the Baseline

The control algorithm by Dang et al. [Dan+25] is used as a baseline for comparison, as this work directly builds upon it.

4.10.1 Baseline Algorithm

The baseline algorithm from previous work implements the swiping policy as follows:

1. A Kalman filter is applied for preprocessing, using measurement noise covariance to estimate the whisker deflection offset.
2. The swiping policy employs a single PID controller that takes the whisker deflection error as input and returns the desired velocity in the direction of the object (in the sensor frame).
3. The total velocity is maintained, with the principal direction velocity in the local frame calculated as the complement to the velocity toward whisker compression or relaxation.
4. The output velocity in the world frame is obtained by rotating the local frame velocity by the whisker tip spline tangent.

The underlying idea is that to track the surface, the whisker deflection profile must remain constant; hence, the platform moves closer to the object when the whisker is too relaxed and moves away when it is too strained.

4.10.2 Comparison

The baseline control can be characterized as follows:

Pros:

- The whisker deflection offset is less susceptible to noise due to filtering.
- The swiping policy is straightforward and is designed to maintain a constant whisker profile.

Cons:

- Filtering does not resolve the inherent noise or limitations of an ill-posed deflection model; subtle changes in whisker deflection that may be important for surface reconstruction can be lost.

- Maintaining a constant deflection profile ensures surface tracking but does not guarantee complete contour capture.
- The platform's velocity is relatively slow, and sudden changes in the principal direction—due to whisker slippage—can result in a 180° change in the spline direction, potentially causing reverse motion.

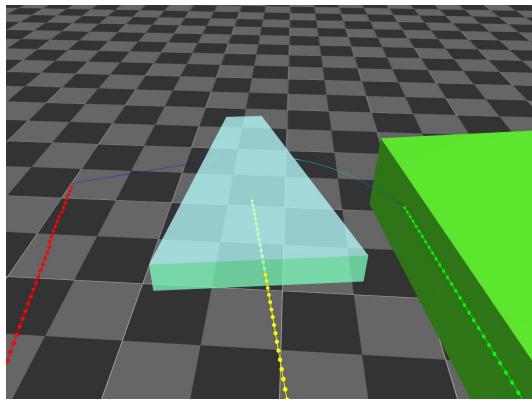
The following steps are possible to rectify the shortcomings of the baseline algorithm:

1. To enhance reactivity, remove the Kalman filter and apply smoothing directly to the deflection signal (using the low-pass Butterworth filter described in Section 4.3).
2. To modify the swiping policy to balance the importance of following the spline with maintaining the deflection profile.
3. To introduce a whisker orientation parameter (clockwise or counterclockwise) to prevent the whisker profile from deflecting in the opposite direction during control. This orientation is set upon initial contact with the surface and is preserved until the whisker disengages. It is employed in the calculation of the desired whisker base-tip offset.

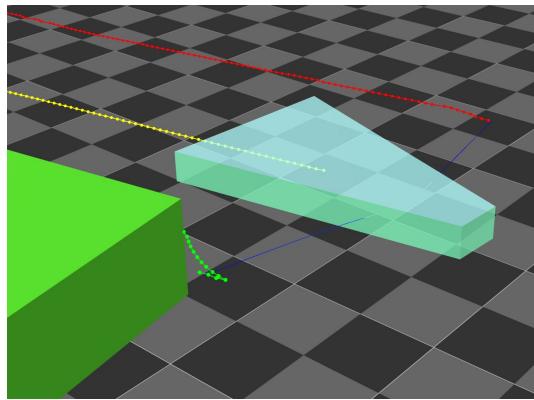
4.10.3 Limitations of the Suggested Control Algorithm

The proposed control algorithm has several limitations:

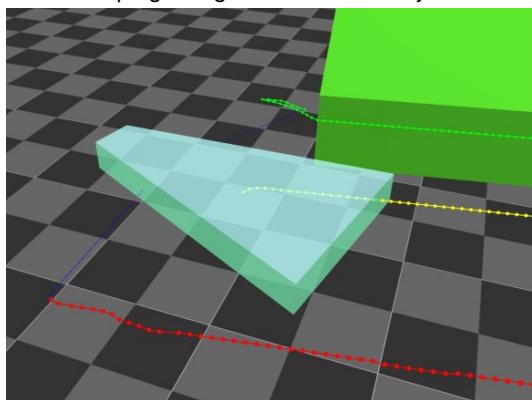
- It does not consider the platform dimensions to prevent non-whisker contacts.
- It requires parameter tuning (e.g., velocity, desired deflection, contact deflection threshold, PID parameters, etc.).
- It is not failsafe regarding whisker integrity, as extreme deflections are not directly forbidden.
- It does not allow for non-constant velocity and does not optimize the velocity for the environment.
- It can lead to situations where the platform might get stuck (for instance, at the tunnel end, since the minimal space required for platform rotation without damaging the whisker is not calculated).
- There are other limitations imposed by the assumptions discussed earlier.



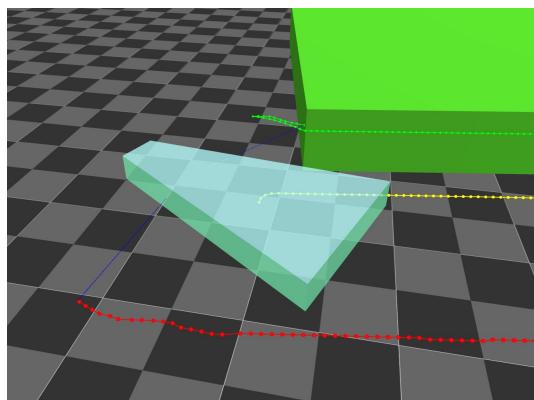
(a) Initial situation
Swiping along the side of the object.



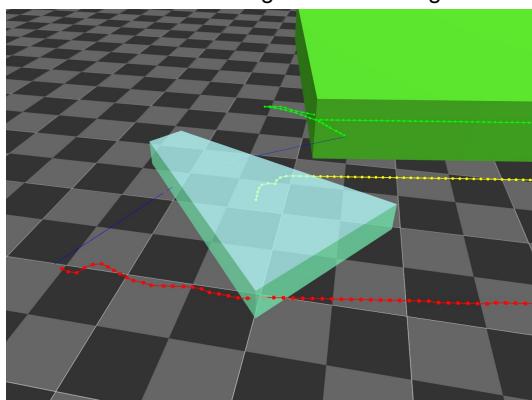
(b) Disengagement
Whisker has detached.



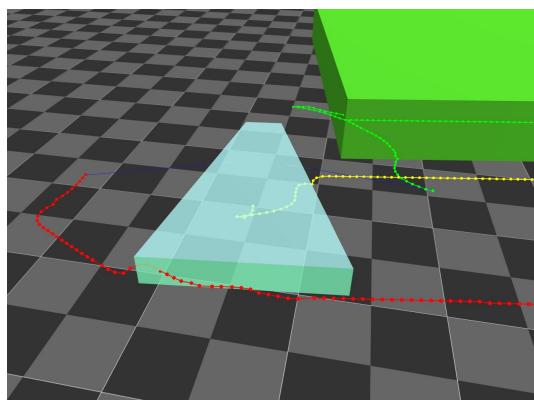
(c) Rotation
Whisker is rotating around the edge.



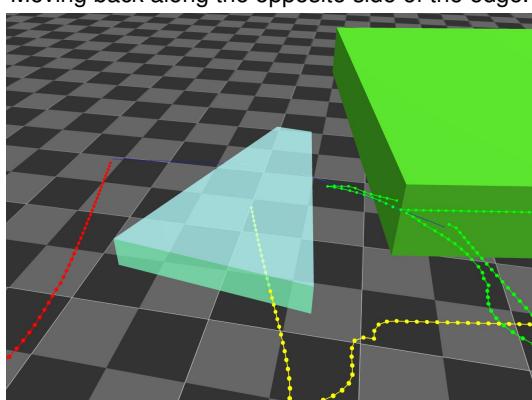
(d) Retrieval
Whisker has retrieved the contact.



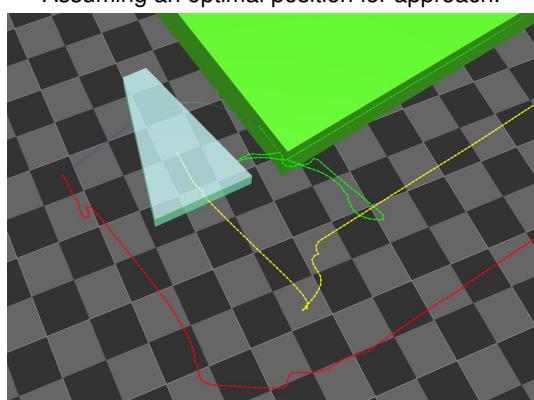
(e) Whisking
Moving back along the opposite side of the edge.



(f) Repositioning
Assuming an optimal position for approach.



(g) Approach
Moving towards the object.



(h) Transition to Swiping
Transferring control to the Swiping Policy.

Figure 4.2: Retrieval Policy in 8 steps.

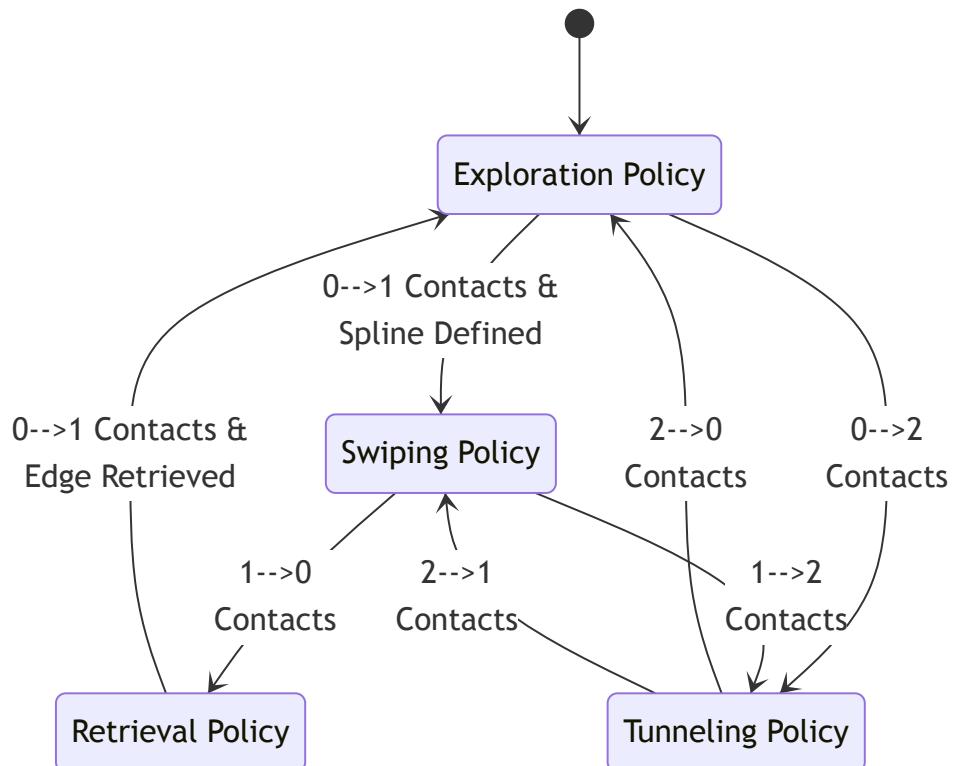


Figure 4.3: FSM diagram of the whisker control system. The diagram illustrates the various states (Exploring, Swiping, Tunneling, Retrieval, Whisking, Failure) and the transitions triggered by contact establishment, detachment, and reattachment.

Chapter 5

Experimental Setup

We developed a simulation framework, tools for collecting metrics, and an automatic test runner to validate the proposed policies. It is backed by MuJoCo which provides a realistic simulation of contact dynamics. Our developed framework is available on GitHub¹.

5.1 Swiping Policy

For the swiping policy, a single whisker executes a sweeping motion over the object. Performance is quantified by evaluating the mean absolute error and the standard error in contour reconstruction, with each estimated point \mathbf{y}_i matched to its closest reference point \mathbf{x}_i .

The estimated contour, $C_{\text{est}} = \{\mathbf{y}_i\}_{i=1}^N$, is determined by the control algorithm, which selects points based on whisker deflection and excludes those recorded during the whisking-back phase in retrieval. The reference contour, $C_{\text{ref}} = \{\mathbf{x}_i\}_{i=1}^N$, consists of points on the object's surface, each being the closest match to the corresponding estimated point, and is derived from high-resolution sampling to serve as ground truth.

The absolute error is $d_i = \|\mathbf{x}_i - \mathbf{y}_i\|$, which is the Euclidean distance between the reference and estimated points. To assess performance, we compute the mean absolute error

$$\bar{d} = \frac{1}{N} \sum_{i=1}^N d_i,$$

and the standard error

$$\sigma_d = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (d_i - \bar{d})^2}.$$

The mean absolute error represents the average distance between the reference and estimated points.

Figures 5.1–5.3 display results for various objects. Estimated contour points are colored green if their error deviates from the mean by less than the standard error and red otherwise. Red points indicate the most inaccurate measurements, corresponding to the worst 32 percentile of the error distribution.

Figure 5.1 shows the contour estimation of a disk using the swiping policy. The disk is the simplest object; the mean reconstruction error is $0.8 \text{ mm} \pm 0.5 \text{ mm}$, indicating a high level of accuracy. No prominent red regions are observed, except for occasional deviations due to the disk's polygonal approximation at the entry point.

¹<https://github.com/SVDouble/RatteChan>

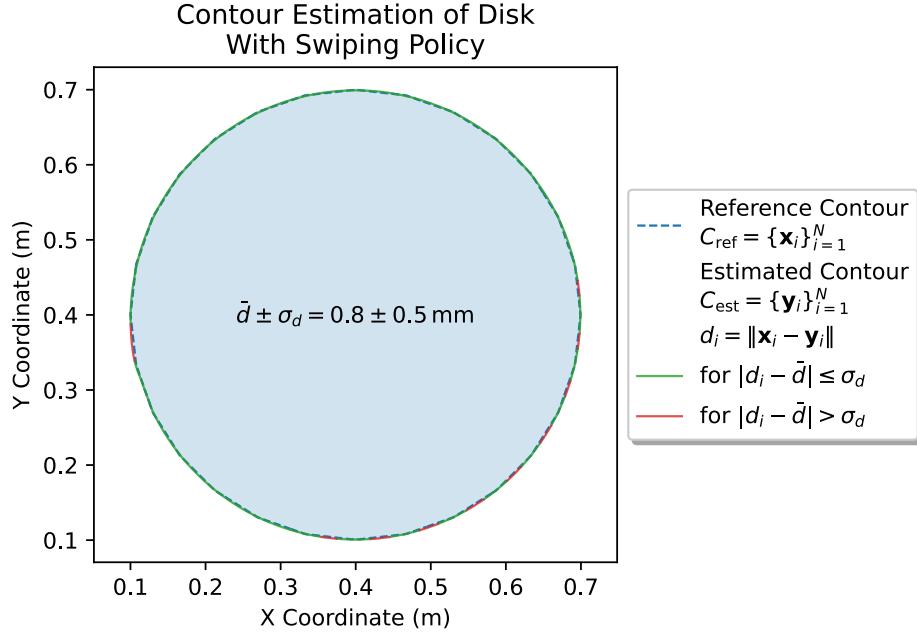


Figure 5.1: Contour Estimation of Disk With Swiping Policy

Figure 5.2 illustrates the contour estimation of a rounded rectangular box. The rounded rectangular box is a more complex object with sharper corners, yet it shows similar accuracy and mean error. The mean reconstruction error is $0.8 \text{ mm} \pm 0.6 \text{ mm}$, with a single red region after the first corner.

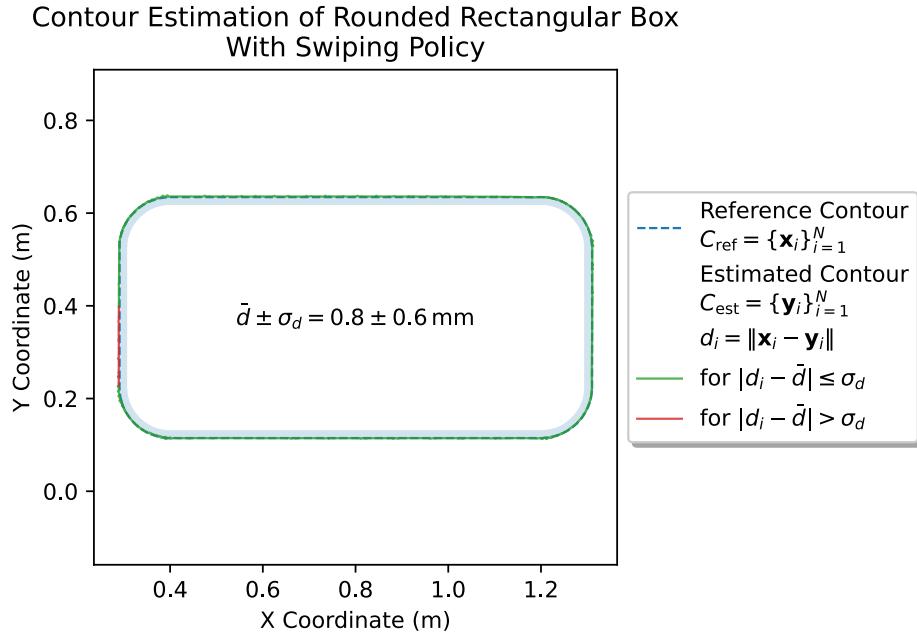


Figure 5.2: Contour Estimation of Rounded Rectangular Box With Swiping Policy

Figure 5.3 shows the contour estimation of a complex object. Noticeable red regions appear at the inner inflections. Since the inner angles are not smooth, the platform must quickly adjust to surface changes, which leads to a higher error rate. At the initial moments, the whisker becomes overly deflected, causing the error to increase as the deflection model

performs worse outside its normal operating range. The mean error remains comparable to that observed for the disk and the rounded rectangular box.

Additionally, the swiping policy can successfully handle a 30° angle without causing the whisker to detach. This angle is near the threshold where whisker detachment occurs, and the retrieval policy is triggered.

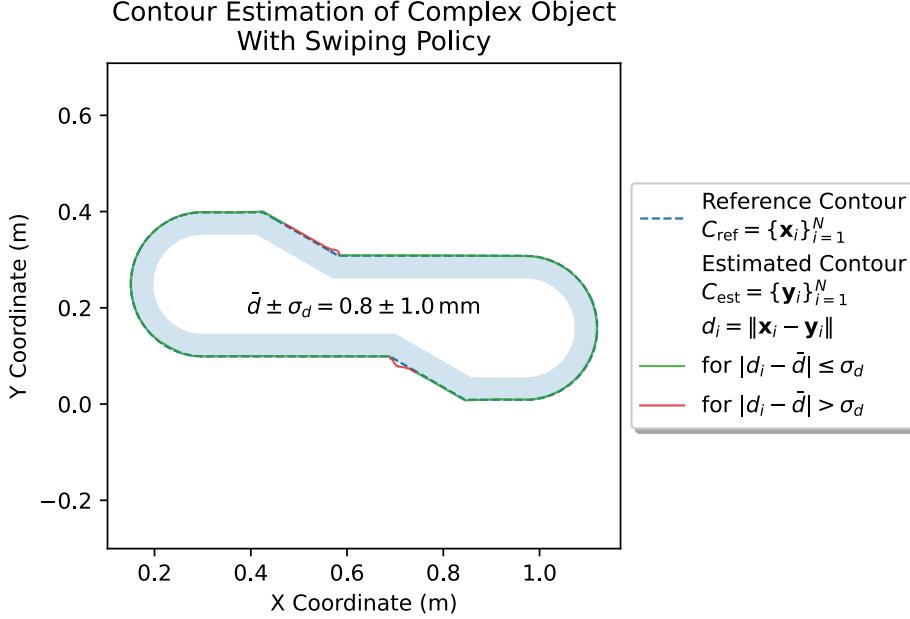


Figure 5.3: Contour Estimation of Complex Object With Swiping Policy

5.2 Retrieval Policy

Figures 5.4–5.7 show results for different polygonal objects. Let k_{edge} be the index of an edge point and k_{retr} the index of its corresponding retrieved point, with the same indexing for both events. The following metrics evaluate the retrieval policy:

- **Mean absolute error:** $\bar{d} = \frac{1}{N} \sum_{i=1}^N d_i$, discussed in Section 5.1.
- **Mean retrieval radius:** $\bar{r} = \frac{1}{N} \sum_{j=1}^N \|\mathbf{r}_{\text{edge},j} - \mathbf{r}_{\text{retr},j}\|$, where $\mathbf{r}_{\text{edge},j}$ is the j th edge point and $\mathbf{r}_{\text{retr},j}$ its corresponding retrieved point.
- **Mean retrieval distance:** $\bar{d}_{\text{retr}} = \frac{1}{N} \sum_{j=1}^N \sum_{i=i_{\text{edge},j}}^{i_{\text{retr},j}-1} \|\mathbf{r}^{t_{i+1}} - \mathbf{r}^{t_i}\|$, which calculates the trajectory length between the time of detachment and the time of retrieval, where \mathbf{r}^t denotes the body position at time t .

In the plots, the edge points \mathbf{r}_{edge} are shown as red crosses, while the retrieved points \mathbf{r}_{retr} appear as black crosses.

The mean retrieval radius is defined as

$$\bar{r} = \frac{1}{N} \sum_{j=1}^N \|\mathbf{r}_{\text{edge},j} - \mathbf{r}_{\text{retr},j}\|,$$

quantifies the proximity of a retrieved point to the edge, which is critical for accurately determining the edge direction. A smaller \bar{r} corresponds to a faster and more precise retrieval.

Ideally, this radius should be less than the distance between the whisker tip in its optimally deflected state (with $\delta = \delta_{\text{target}}$) and its neutral state (with $\delta = 0$), which is approximately 3 cm. The control algorithm sets the target mean retrieval radius to 1 cm; values below 3 cm are acceptable, provided the contact angle remains within the desired range.

The mean retrieval distance, given by

$$\bar{d}_{\text{retr}} = \frac{1}{N} \sum_{j=1}^N \sum_{i=k_{\text{edge},j}}^{k_{\text{retr},j}-1} \|\mathbf{r}^{t_{i+1}} - \mathbf{r}^{t_i}\|,$$

measures the trajectory length from the time of detachment until the system resumes normal operation after retrieval. A longer trajectory indicates more unnecessary movement, reducing the efficiency of the retrieval policy. Note that the whisker's overshoot may increase \bar{r} for large, flat contact angles, even if the backward motion is not required to establish optimal contact.

The retrieval performance for the octagon with 135° edges is presented in Figure 5.4. With its high number of edges, this object demands a stable retrieval policy. The octagon exhibits a mean reconstruction error of $1.1 \text{ mm} \pm 0.8 \text{ mm}$, comparable to the performance observed with the swiping policy. It also has a mean retrieval radius of $20.9 \text{ mm} \pm 1.7 \text{ mm}$, about one-quarter of the whisker length. This radius is higher than the target of 1 cm because fine adjustment of the whisker's position is unnecessary. The low relative standard deviation indicates high reproducibility. The retrieval policy was triggered 8 times in total.

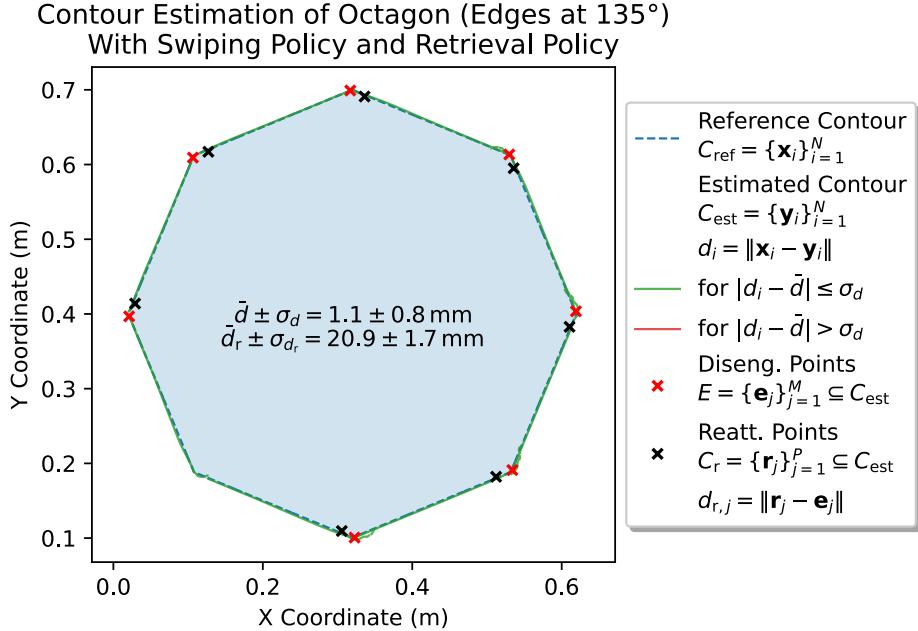


Figure 5.4: Contour Estimation of Octagon (Edges at 135°) With Swiping and Retrieval Policy

The retrieval on a box with 90° edges is shown in Figure 5.5. The box's sharper edges present a greater challenge for the whisker. It achieves a mean reconstruction error of $0.7 \text{ mm} \pm 0.4 \text{ mm}$, demonstrating good performance. It also has a mean retrieval radius of $11.2 \text{ mm} \pm 1.9 \text{ mm}$, which is close to the 1 cm target while preserving retrieval consistency.

The performance on a prism with 60° edges is illustrated in Figure 5.6. The prism yields a mean reconstruction error of $0.8 \text{ mm} \pm 0.6 \text{ mm}$, in line with the swiping policy. It also has a mean retrieval radius of $8.8 \text{ mm} \pm 0.6 \text{ mm}$, approaching the target value. However, an inflation in the trajectory is observed at the retrieved edges. This is likely due to the transition between the undeflected and deflected states. Such overshooting—especially when the

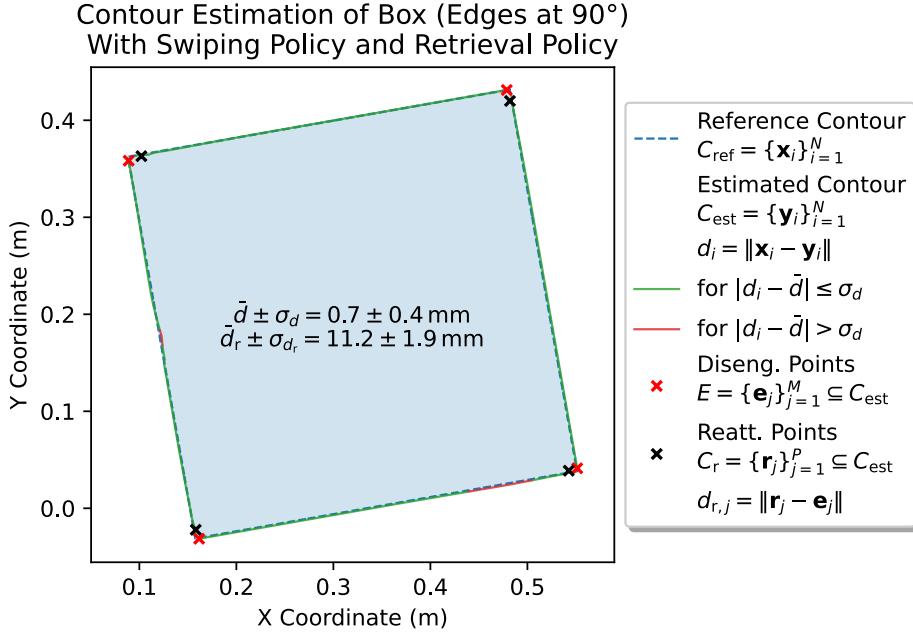


Figure 5.5: Contour Estimation of Box (Edges at 90°) With Swiping and Retrieval Policy

whisker contacts the edge away from its tip—is necessary to accelerate deflection compensation during subsequent swiping. It also helps minimize the platform’s rocking in the normal direction. A more sophisticated deflection model that accounts for contact along the whisker shaft could potentially mitigate this effect.

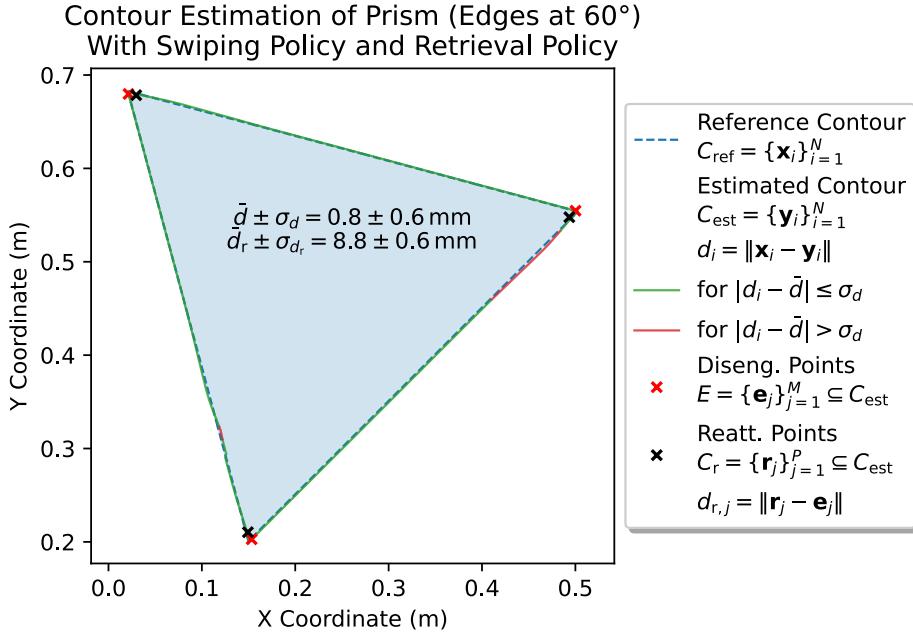


Figure 5.6: Contour Estimation of Prism (Edges at 60°) With Swiping and Retrieval Policy

The wall experiment with 90° edges is presented in Figure 5.7. This scenario is challenging because the wall’s width is only 1 cm, matching the target retrieval radius. Nonetheless, the system achieves a mean retrieval radius of $10.2 \text{ mm} \pm 0.7 \text{ mm}$, nearly meeting the target. This performance is partly due to the wall’s precise 1 cm width, which limits the possibility of

contacts farther away. Video recordings confirm that the whisker tip is well-aligned and contacts the opposite edge accurately. If the wall were about 25% shorter, the policy could still handle it through deflection at the whisker's shaft, provided the side length remains above a critical threshold. Conversely, when the target radius is doubled, the platform executes a 180° turn to reach the edge, rendering the wall undetectable. In this case, the mean reconstruction error is $1.1 \text{ mm} \pm 0.6 \text{ mm}$, consistent with the performance of the swiping policy.

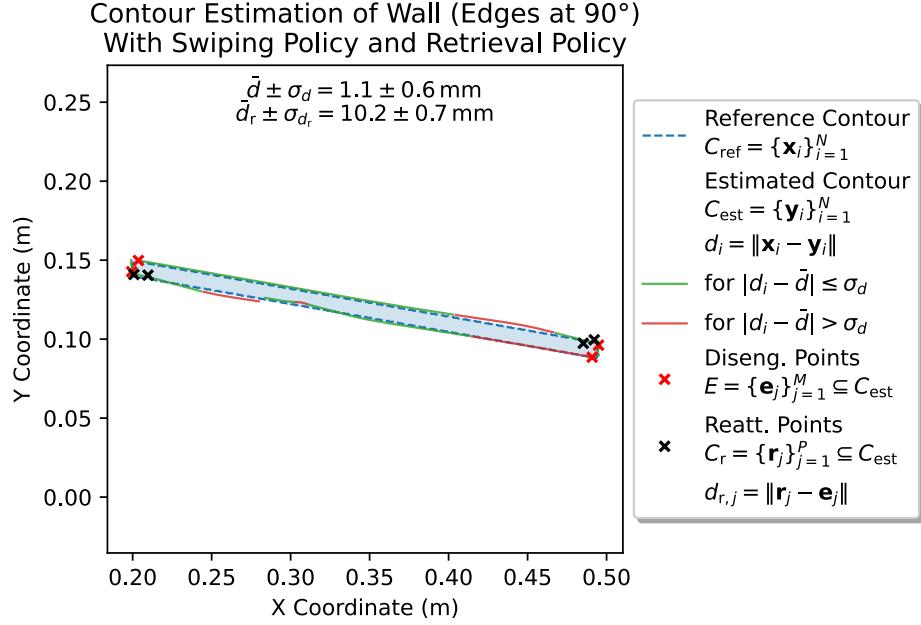


Figure 5.7: Contour Estimation of Wall (Edges at 90°) With Swiping and Retrieval Policy

5.3 Tunneling Policy

A tunneling policy extends the swiping policy to navigate confined environments. Its goal is to maintain a centered trajectory in the tunnel to avoid collisions and ensure accurate contour estimation.

For the tunneling policy, we employ the following metrics:

- **Mean absolute reconstruction error:**

$$\bar{d} = \frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K d_{i,k},$$

where

$$d_{i,k} = \|\mathbf{x}_{i,k} - \mathbf{y}_{i,k}\|$$

is the distance between the i th reference point $\mathbf{x}_{i,k}$ and the corresponding estimated point $\mathbf{y}_{i,k}$ of the k th whisker.

- **Mean absolute axis error:**

$$\bar{d}_{\text{axis}} = \frac{1}{N} \sum_{i=1}^N d_{\text{axis},i},$$

where

$$d_{\text{axis},i} = \|\mathbf{a}_i - \mathbf{a}_m\|$$

is the distance between the point \mathbf{a}_i on the tunnel axis closest to the i th estimated midpoint and the axis \mathbf{a}_m , with the estimated midpoint defined as

$$\mathbf{m}_i = \frac{\mathbf{x}_{i,1} + \mathbf{y}_{i,r}}{2}.$$

The tunnel axis, shown as pink dashed lines in the figures, is defined as the line equidistant from the tunnel walls. Thus, the mean absolute axis error quantifies the estimated midpoints (marked in black) deviation from the tunnel axis.

Figures 5.8–5.10 illustrate smooth, zigzag, and round tunnels.

The smooth tunnel in Figure 5.8 represents the simplest case. The whisker successfully navigates the tunnel while maintaining a centered trajectory. The mean absolute reconstruction error is $1.3 \text{ mm} \pm 1.6 \text{ mm}$, and the mean absolute axis error is $4 \text{ mm} \pm 4 \text{ mm}$. For reference, the tunnel is approximately 10 cm wide, and each whisker is 7.5 cm long. The difference between the reconstruction error and the axis error is due to a lag in the midpoint trajectory relative to the tunnel axis. This lag occurs because the control system cannot immediately adjust the platform's position owing to its inertia and the continuously changing tunnel direction. Additionally, heightened deflection near the operational limit of the deflection model and a transition effect when the whiskers first come into contact with the tunnel walls contribute to this error.

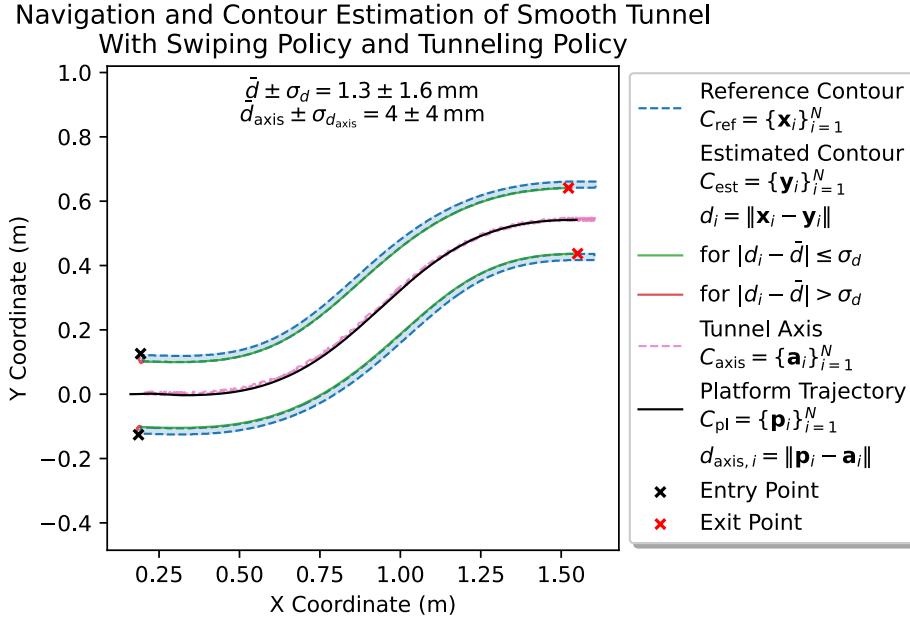


Figure 5.8: Navigation and Contour Estimation of Smooth Tunnel With Swiping and Tunneling Policy

The zigzag tunnel in Figure 5.9, with angles of 20° and 40° , is more complex due to its sharper corners. In this case, the mean absolute reconstruction error is $2 \text{ mm} \pm 2 \text{ mm}$, while the mean absolute axis error increases to $9 \text{ mm} \pm 10 \text{ mm}$. This increase is attributed to the difficulty of maintaining a centered trajectory, as the midpoint trajectory deviates from the tunnel axis at the corners. Furthermore, the left whisker detaches at the first zigzag, while the right whisker remains in contact with the wall. Because the left whisker cannot maintain contact at the sharp angle, the swiping policy is applied to the right whisker until the left whisker re-establishes contact. The platform's nose contacts the wall for a short period, but it

quickly recovers. Overall, the performance is satisfactory, as the whiskers navigate the tunnel and accurately estimate the contour. For even sharper zigzag angles, the platform may fail to pass through the tunnel if its nose prods a wall, as no recovery policy is implemented for such cases. Such failures would be indicated by increased actuator force or a reduced platform speed, which are not measured in the current setup.

Navigation and Contour Estimation of Zigzag Tunnel With Swiping Policy and Tunneling Policy

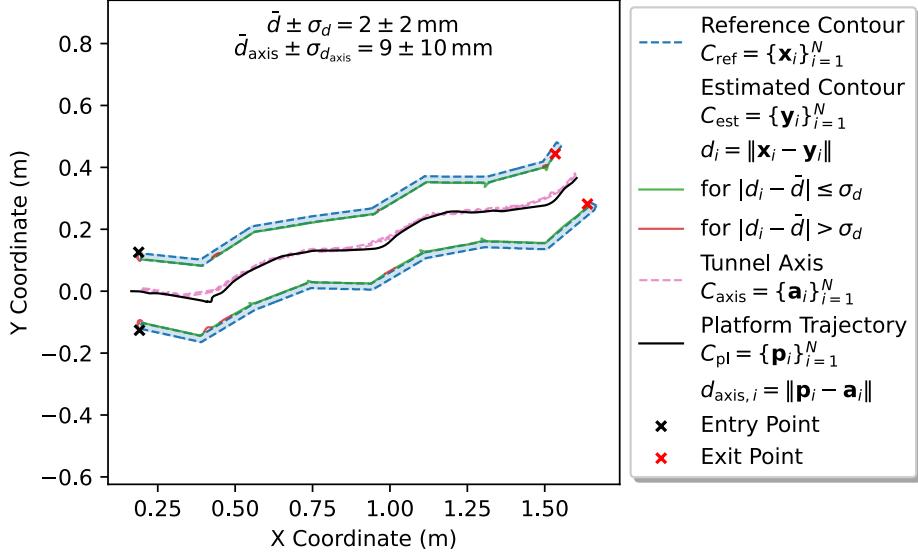


Figure 5.9: Navigation and Contour Estimation of Zigzag Tunnel With Swiping and Tunneling Policy

The round tunnel in Figure 5.10 tests the endurance of the tunneling policy. It involves a longer trajectory and a star-shaped loop. A pronounced transition effect is observed at the start, as the whisker adjusts to the tunnel's curvature and the initial position (marked by black crosses) is not aligned with the tunnel axis. Skidding is also noticeable at the sharper curves. The performance remains satisfactory, with a mean absolute reconstruction error of $3 \text{ mm} \pm 2 \text{ mm}$ and a mean absolute axis error of $5 \text{ mm} \pm 6 \text{ mm}$. The increased reconstruction error is likely due to operating outside the optimal range of the deflection model at the curves. This is evident in segments of the tunnel marked in red, where the outer whisker (right, in this case) experiences excessive compression.

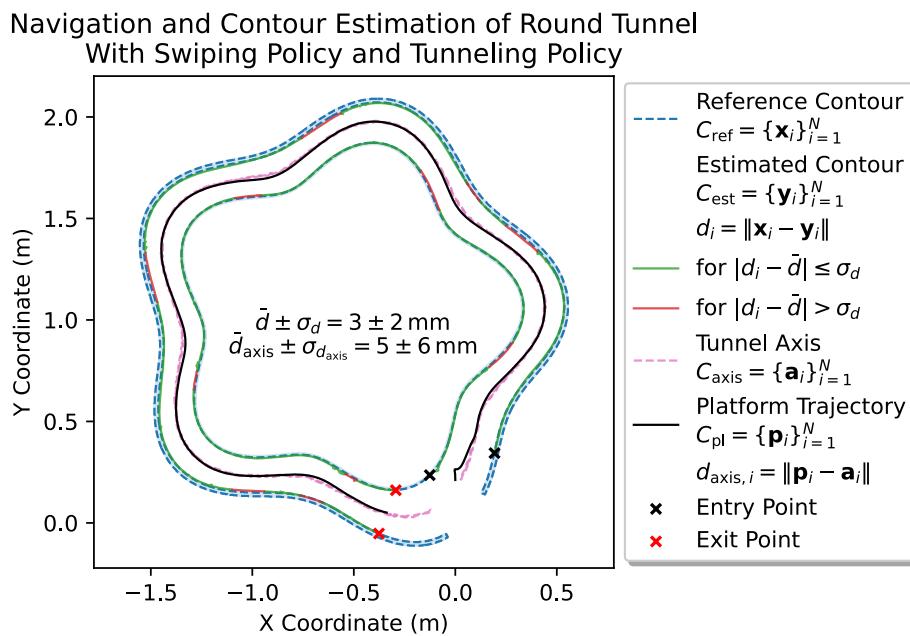


Figure 5.10: Navigation and Contour Estimation of Round Tunnel With Swiping and Tunneling Policy

Chapter 6

Infrastructure

The following chapter details the integrated system infrastructure that underpins our whisker sensor platform. It covers the real-time data acquisition, processing, and visualization components and the modular software services orchestrated in Docker containers. This infrastructure seamlessly connects hardware elements—such as the whisker platform and the actuator—with the control unit and simulation environment.

6.1 Overview

The system infrastructure takes care of real-time sensor data collection, processing, and visualization. Its implementation is available on GitHub¹ under the `services` directory. The architecture comprises three main entities: the whisker platform, the controller, and the actuator. The platform provides two Inter-Integrated Circuit (I2C) buses for reading sensor data, each equipped with three Adafruit MLX90393 sensors. The sensors are connected in a daisy chain via a 4-pin JST connector. This allows us to avoid the need for soldering. Theoretically, up to 16 sensors can be connected per bus, but we are using only three, as the development boards are limited to four sensors per bus. The controller device, a Raspberry Pi 5, was chosen because it has sufficient processing power and is small enough to be mounted on the robotic arm. Currently, the actuator is simulated; however, it will eventually be replaced by a robotic arm. Figure 6.1 summarizes the system architecture.

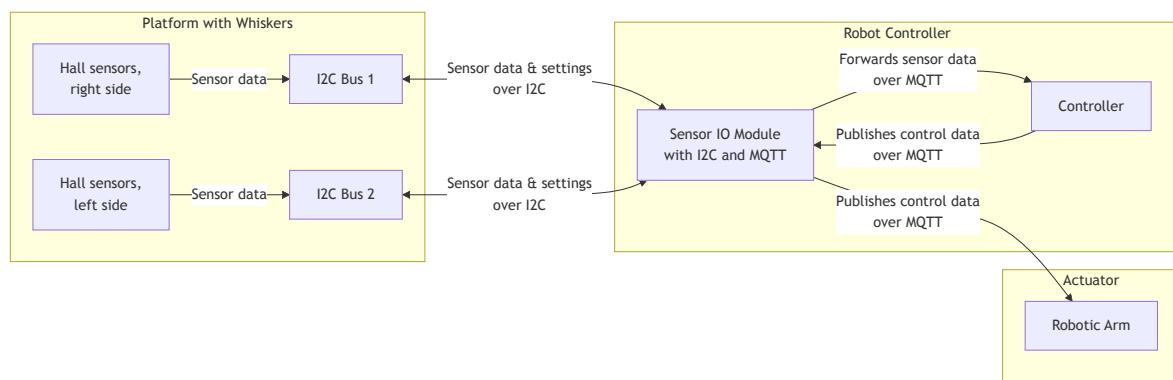


Figure 6.1: Overview of the system infrastructure.

¹<https://github.com/SVDouble/RatteChan>

6.2 Controller

The control loop of the system is depicted in Figure 6.2. This loop continuously processes sensor data, evaluates control policies, and generates control messages. Figure A.1 illustrates the hierarchy of components within the controller.

6.3 Simulation

The simulation framework backed by MuJoCo is used for testing the controller. It provides a realistic environment for validating control algorithms. Figure 6.3 depicts the simulation interface of MuJoCo.

The simulation framework is designed to be modular, so the controller has no dependency on the simulation framework. That makes the integration of new frameworks or sensor data sources and sinks possible. Ten test cases are provided to test the performance of 3 different policies. The results are automatically evaluated and plotted for each test case, similar to Figures 5.1–5.10. An accompanying video is provided for each test case, showing the simulation in action. MuJoCo allows us to follow, pause, and analyze the simulation. The simulation is set up so that the trajectories of the whisker tips and the platform center of mass are displayed continuously. This is a sanity check for the simulation, as the deflection model and tip calculations are error-prone.

6.4 System Services

The system comprises several interconnected services, as illustrated in Figure 6.4. All of them run in Docker containers to guarantee a consistent and separate environment across different machines. Docker containers are lightweight, portable, and self-sufficient units that encapsulate all the necessary components to run a software application. They are orchestrated using Docker Compose², a tool for defining and running multi-container applications. The control system runs on any machine with Docker installed, like Raspberry Pi 5, making it easy to deploy.

- **Sensor IO Module:** This module reads sensor data from the robot's I2C buses and publishes the data via Message Queuing Telemetry Transport (MQTT). It also forwards control commands from the controller to the actuator. Currently, the built-in I2C buses of the Raspberry Pi are used for simplicity. But generally speaking, the Sensor IO module can be decoupled from the controller and run on a separate device.
- **Controller:** The controller processes sensor data and generates control signals. It also publishes statistics for further monitoring.
- **MQTT Broker:** As the central messaging hub, the MQTT Broker receives and distributes sensor and control data. It is based on the subscriber-publisher model. This component is critical for system responsiveness.
- **Telegraf:** Telegraf collects sensor data and controller performance metrics. The data is then forwarded to InfluxDB for storage.

²<https://docs.docker.com/compose/>

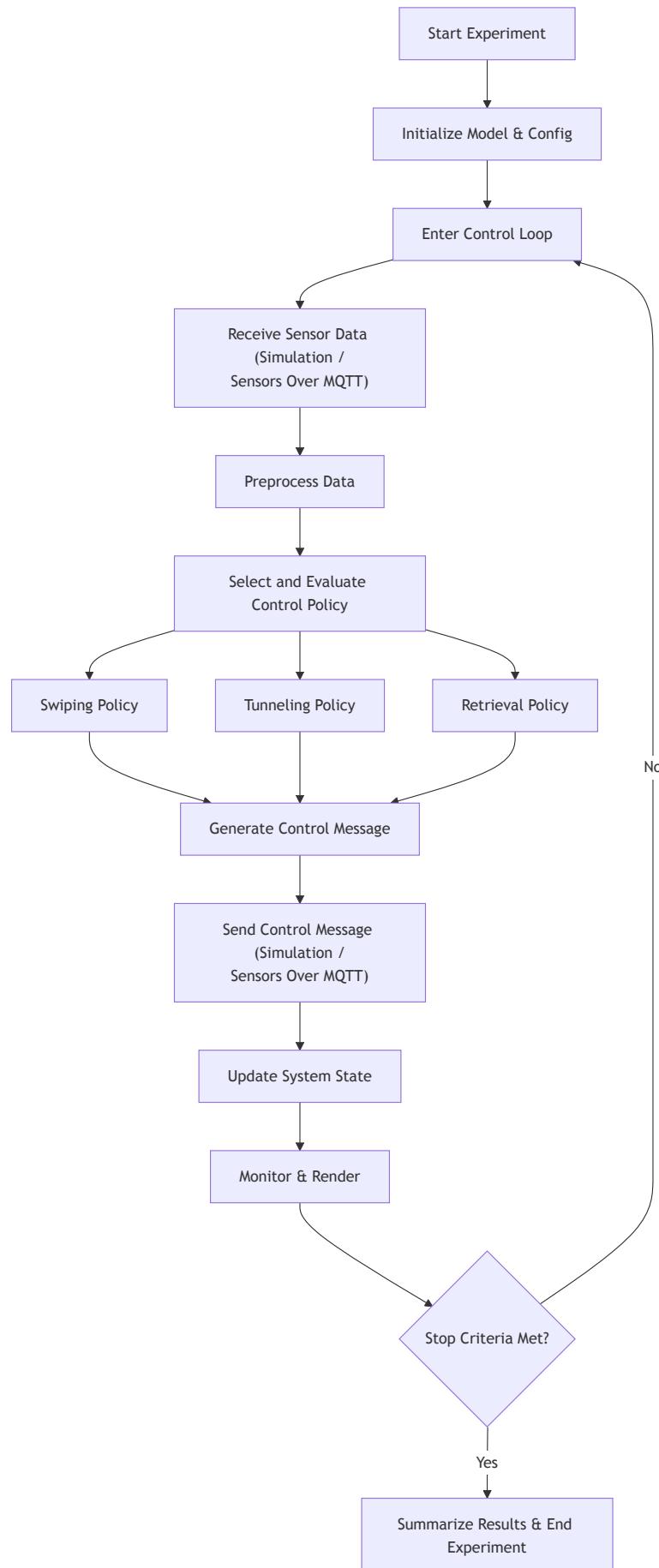


Figure 6.2: Data flow in the system infrastructure.

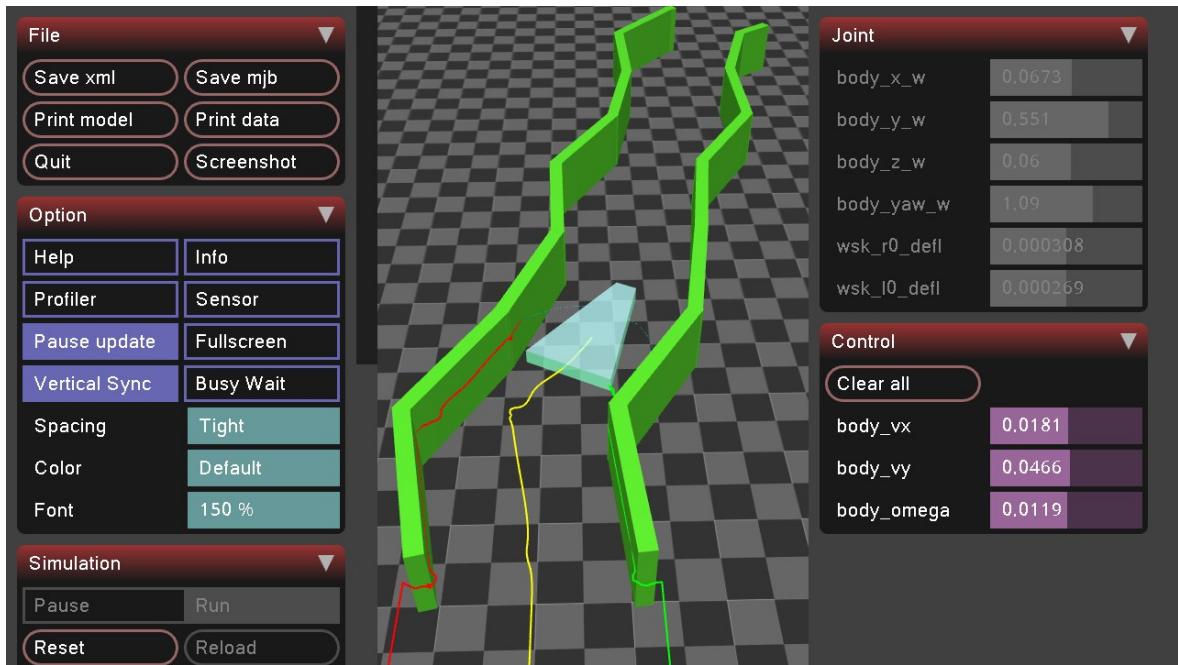


Figure 6.3: Simulation interface of the MuJoCo physics engine.

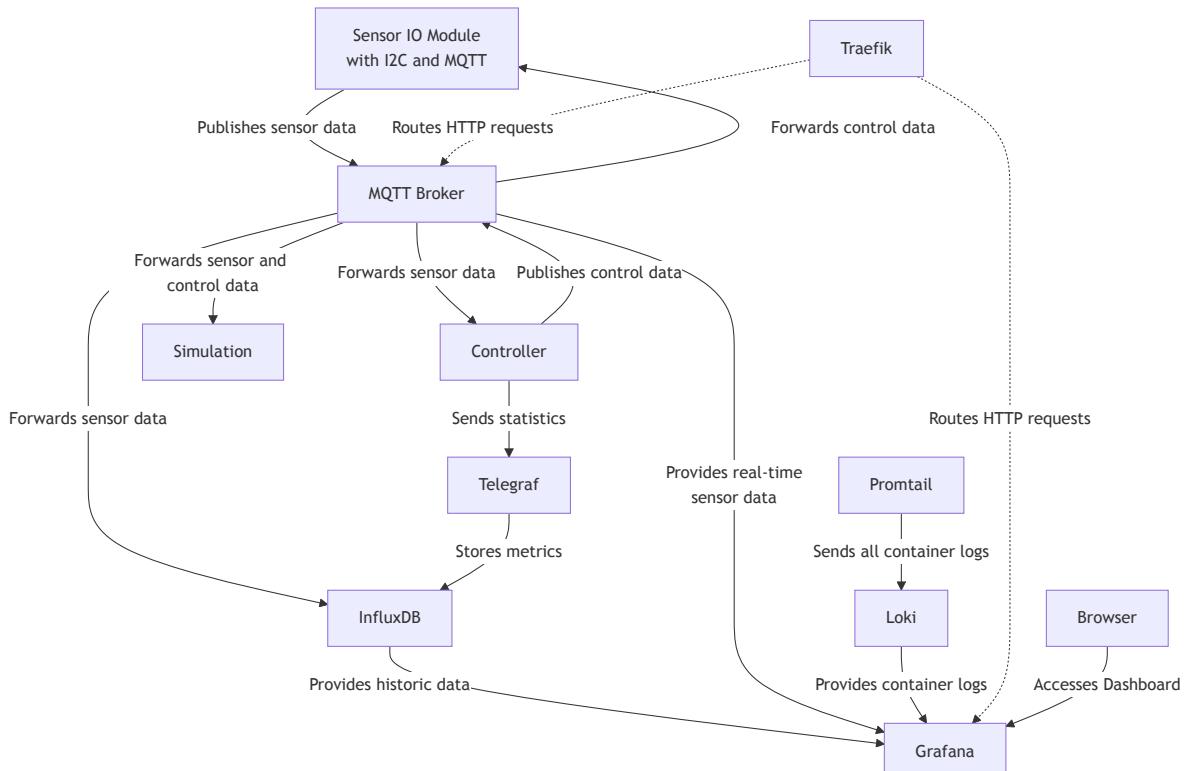


Figure 6.4: Data flow in the system infrastructure.

- **InfluxDB:** InfluxDB is a time-series database that stores historical sensor data and controller performance metrics. It serves as a persistent storage solution for the system during the execution of the experiments.
- **Grafana:** Grafana provides a web-based dashboard that visualizes the sensor data in real time. It allows users to monitor the system's status interactively.
- **Promtail:** Promtail collects logs from the various services and forwards them to Loki.
- **Loki:** Loki aggregates the logs provided by Promtail and makes them available for visualization via Grafana. This centralized logging system helps troubleshoot and monitor system health.
- **Traefik:** Traefik is a reverse proxy for communication between the services and the outside world. For instance, it allows access to the Grafana dashboard via a web browser.

6.5 Dashboard

The system dashboard uses Grafana, a powerful analytics and monitoring platform. It provides a web-based interface for visualizing sensor data and control signals. Grafana Live with MQTT is employed to transmit real-time system status. Figure 6.5 shows a minimalistic dashboard with magnetic data from six whisker sensors sampled in real time.



Figure 6.5: visualization of magnetic data of 6 whisker sensors in real-time using Grafana.

The plans for the dashboard include:

- Visualization of whisker deflection in 2D
- State machine visualization

- Performance metrics of the contour reconstruction algorithm
- System cycle time and overload monitoring
- Historical data analysis and trend monitoring.

Chapter 7

Conclusion

This chapter summarizes our work's contributions and outlines the roadmap for refining autonomous active tactile exploration. The following sections present future directions for the project, emphasizing the transition from simulation to real-world testing and addressing key challenges identified during experimentation.

Furthermore, we discuss opportunities for enhancing the whisker array platform and its control policies, focusing on advanced deflection models, increased sensor integration, and improved collision detection mechanisms.

7.1 Summary

This work lays the groundwork for achieving **autonomous active tactile exploration of unstructured environments using a whisker array**. Four specialized control policies are implemented to steer the whisker array platform for contour reconstruction and tunnel navigation. A **simulation framework backed by MuJoCo** is established to evaluate the whisker control system.

All policies can capture contours with **millimeter-range precision** while navigating complex environments. The swiping policy reconstructs object contours while preserving exploration direction and maintaining whisker contact. It is shown that the swiping policy can reconstruct smooth contours and **overcome angles of up to 30° without detachment**. The retrieval policy is proposed to handle whisker detachment at sharp corners. It is shown that the retrieval policy successfully **reconstructs object edges** and enables smooth whisker detachment handling, ensuring complete contour capture. The tunneling policy is presented for navigating confined spaces. Multiple test scenarios show the tunneling policy successfully **navigates tunnels with a diameter of 22 cm, effectively managing whisker reattachments at sharp corners**. Furthermore, a governing policy is proposed to **handle transitions between exploration, swiping, retrieval, and tunneling policies**. Its efficacy is demonstrated clearly in scenarios such as the zigzag tunnel, where brief whisker detachment occurs, and in all tests involving the retrieval policy.

For testing in real-world scenarios, the **whisker array platform** is designed and assembled, supporting three whiskers on each side. It can be mounted on the gripper of the Franka Emika Panda robotic arm, which acts as an actuator for the whisker array. Finally, a **system infrastructure** for real-time sensor data visualization and evaluation is developed. A **dashboard** is set up to visualize whisker deflection data and the platform's position.

7.2 Future Work

Future directions for this project are the following:

1. It is essential to test the whisker platform not only in simulation but also in **real-world scenarios**. Since the platform is already assembled, the next step is testing the control system with the platform attached to the Franka Emika Panda robotic arm. The arm acts as an actuator, moving the whisker array platform based on evaluated control policies. Real-world experiments will differ from simulations due to **friction at contact points** between whiskers and objects, as well as **noise from surface textures** affecting deflection measurements. Another critical factor is the behavior of nitinol whiskers under axial loads. This is especially relevant for the tunneling policy, where whiskers often collide head-on with surfaces, causing the tip to become stuck and resulting in unpredictable deflections.
2. The whisker **deflection model must be recalibrated** separately for (a) simulations and (b) real-world environments. Using ground truth whisker tip positions in simulations can simplify tuning the control policies. End-to-end control tests provide a realistic system evaluation but complicate debugging as component errors propagate. The current deflection model accurately covers only a limited deflection range, which is insufficient for the tunneling policy. The platform may be placed in tight spaces, leading to large whisker **deflections exceeding the model's valid range**. Significant deviations were found during testing of the tunneling policy, even in simple scenarios, with predicted whisker tip positions typically closer to the platform than their actual positions. Recalibrating the deflection model will improve contour reconstruction performance by increasing whisker tip position accuracy.

7.3 Potential Improvements

We would like to mention modifications for enhancing the whisker array platform and refining its control strategies:

- An **advanced whisker deflection model** is required to detect contact along the entire whisker length. This involves processing additional axes of the hall sensor. For this, more advanced suspensions might be required, which **allow both rotation and axial displacement of the whisker**.
- **More whiskers** can be included in the contour reconstruction. The control policies evaluate only a maximum of one whisker per side. Using even two whiskers per side would significantly speed up contour reconstruction, eliminating the need for swiping back during retrieval. The platform dimensions allow the rear-most whisker to seamlessly capture the edge contour. The retrieval policy would then be limited to resolving the encountered edge angles.
- **Variable speed** of the whisker array platform is required to enable faster exploration. Currently, the platform moves at a constant speed of 5 cm/s, except during the retrieval policy's repositioning phase, when it is allowed to rotate in place.
- An **advanced exploration policy and environment mapping** capability is required for the whisker array to achieve Simultaneous Localization and Mapping (SLAM) in unstructured environments. This involves discovering new objects and determining optimal exploration paths.

- For a robust tunneling policy, **collision detection for the platform** is necessary. Currently, collisions involving the platform's nose, frequently occurring in confined spaces, are not detected. As a result, exploration halts since the whisker array cannot move to the desired position.
- For 3D contour reconstruction, a **Printed Circuit Board (PCB) with multiple Hall-effect sensors** is required. The simplest arrangement would involve three whisker sensors positioned in a triangle with side lengths significantly shorter than the whisker length. Due to their large size, such an arrangement is currently not achievable with Adafruit MLX90393 development boards.
- Integrating the whisker array into the **robotic rat** would represent the pinnacle of the project.

Acknowledgements

I would like to express my gratitude to my supervisor, **Yixuan Dang**, for his guidance and support throughout this project. I would also like to thank my fellow student **Yichen Zhang**, who also worked on the whisker project as part of his bachelor thesis, for his help and collaboration. Finally, I am grateful to my family and friends for their support during my studies.

Appendix A

Additional Information

A.1 Control Hierarchy

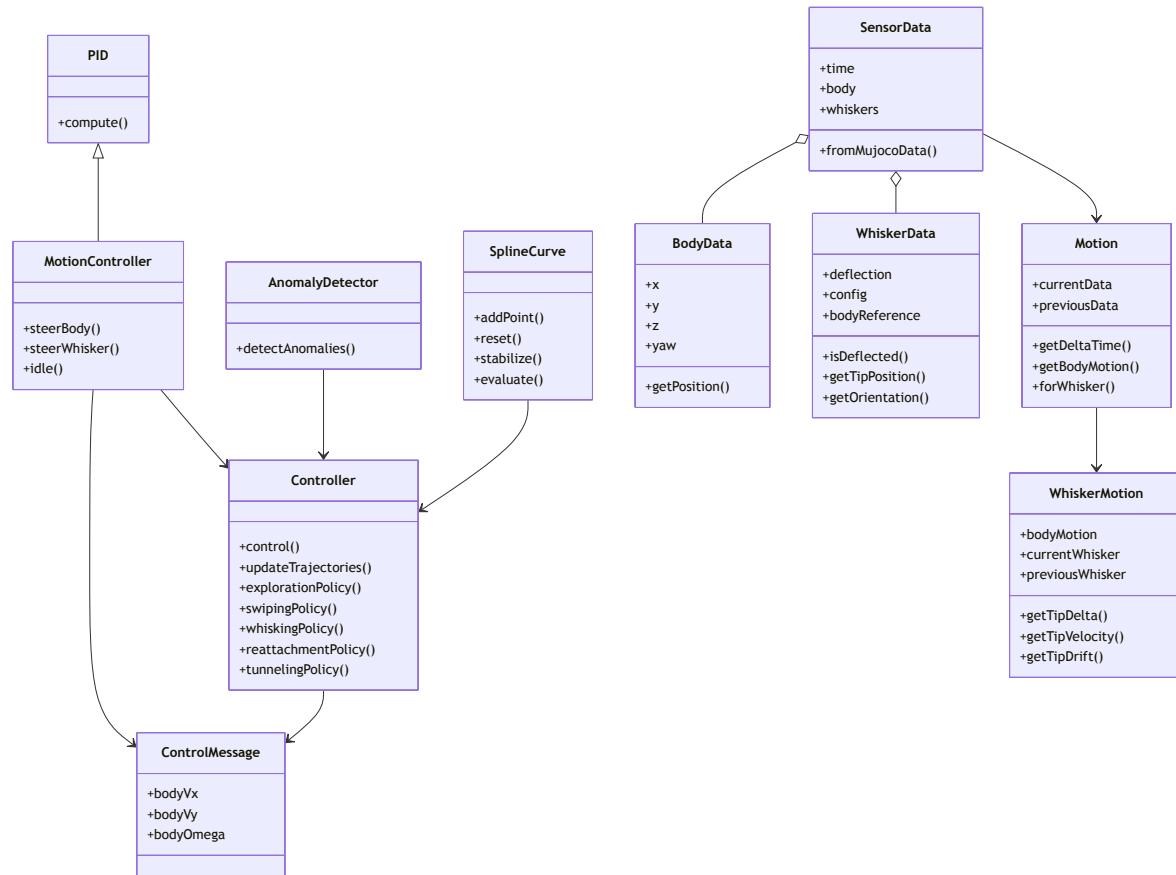


Figure A.1: Hierarchy of classes in the control system.

Bibliography

- [Bro+24] Brouwer, D., Citron, J., Choi, H., Lepert, M., Lin, M., Bohg, J., and Cutkosky, M. R. “Tactile-Informed Action Primitives Mitigate Jamming in Dense Clutter”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)* (2024), pp. 7991–7997. URL: <https://api.semanticscholar.org/CorpusID:267682320>.
- [Dan+25] Dang, Y., Xu, Q., Zhang, Y., Yao, X., Zhang, L., Bing, Z., Roehrbein, F., and Knoll, A. “Whisker-based Active Tactile Perception for Contour Reconstruction”. Unpublished manuscript. 2025.
- [Guo+24] Guo, L., Liu, J., Wu, G., Xu, P., Wang, S., Liu, B., Li, Y., Guan, T., Wang, H., Si, J., Du, T., and Xu, M. “Piezoelectric wavy whisker sensor for perceiving underwater vortex from a bluff body”. In: *Sensors and Actuators A: Physical* 365 (2024), p. 114875. ISSN: 0924-4247. DOI: <https://doi.org/10.1016/j.sna.2023.114875>. URL: <https://www.sciencedirect.com/science/article/pii/S0924424723007240>.
- [HRH17] Huet, L. A., Rudnicki, J. W., and Hartmann, M. J. “Tactile Sensing with Whiskers of Various Shapes: Determining the Three-Dimensional Location of Object Contact Based on Mechanical Signals at the Whisker Base”. In: *Soft Robotics* 4.2 (2017). PMID: 28616371, pp. 88–102. DOI: 10.1089/soro.2016.0028. eprint: <https://doi.org/10.1089/soro.2016.0028>. URL: <https://doi.org/10.1089/soro.2016.0028>.
- [IMG22] Ibarra-Castaneda, N., Moy-Lopez, N. A., and Gonzalez-Perez, O. “Tactile information from the vibrissal system modulates hippocampal functioning”. In: *Current Research in Neurobiology* 3 (2022), p. 100034. ISSN: 2665-945X. DOI: <https://doi.org/10.1016/j.crneur.2022.100034>. URL: <https://www.sciencedirect.com/science/article/pii/S2665945X22000079>.
- [KKB10] Katz, D., Kenney, J., and Brock, O. “How Can Robots Succeed in Unstructured Environments?” In: (Dec. 2010).
- [Kim+19] Kim, S., Velez, C., Patel, D. K., and Bergbreiter, S. “A Magnetically Transduced Whisker for Angular Displacement and Moment Sensing”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 665–671. DOI: 10.1109/IROS40897.2019.8968518.
- [Lin+22] Lin, M. A., Reyes, E., Bohg, J., and Cutkosky, M. R. *Whisker-Inspired Tactile Sensing for Contact Localization on Robot Manipulators*. 2022. arXiv: 2210.12387 [cs.RO]. URL: <https://arxiv.org/abs/2210.12387>.
- [Mel17] Melexis. *MLX90393 Datasheet*. Datasheet, accessed 2025-03-31. 2017. URL: <https://cdn-learn.adafruit.com/assets/assets/000/069/600/original/MLX90393-Datasheet-Melexis.pdf?1547824268>.
- [MB18] Muthuramalingam, M. and Bruecker, C. *Seal and Sea lion Whiskers Detect Slips of Vortices Similar as Rats Sense Textures*. 2018. arXiv: 1812.09105 [physics.flu-dyn]. URL: <https://arxiv.org/abs/1812.09105>.

- [Par+24] Park, J., Kim, M., Park, J., Hong, M., Im, S., Choi, D., Kim, E., Gong, D., Huh, S., Jo, S.-U., Kim, C., Koh, J.-S., Han, S., and Kang, D. “Active Whisker-Inspired Food Material Surface Property Measurement Using Deep-Learned Mechanosensor”. In: *Advanced Intelligent Systems* n/a.n/a (2024), p. 2300660. DOI: [https://doi.org/10.1002/isbury.202300660](https://doi.org/10.1002/aisy.202300660). eprint: [https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/isbury.202300660](https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/isbury.202300660). URL: <https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/isbury.202300660>.
- [Saygh+22] Sayegh, M.-A., Daraghma, H., Mekid, S., and Bashmal, S. “Review of Recent Bio-Inspired Design and Manufacturing of Whisker Tactile Sensors”. In: *Sensors* 22.7 (2022). ISSN: 1424-8220. DOI: 10.3390/s22072705. URL: <https://www.mdpi.com/1424-8220/22/7/2705>.
- [TET12] Todorov, E., Erez, T., and Tassa, Y. “MuJoCo: A physics engine for model-based control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109.
- [Wei+19] Wei, Z., Shi, Q., Li, C., Yan, S., Jia, G., Zeng, Z., Huang, Q., and Fukuda, T. “Development of an MEMS based biomimetic whisker sensor for tactile sensing”. In: *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. 2019, pp. 222–227. DOI: 10.1109/CBS46900.2019.9114501.
- [Wolfe+08] Wolfe, J., Hill, D. N., Pahlavan, S., Drew, P. J., Kleinfeld, D., and Feldman, D. E. “Texture coding in the rat whisker system: slip-stick versus differential resonance”. In: *PLoS biology* 6.8 (2008), e215.
- [Xia+22] Xiao, C., Xu, S., Wu, W., and Wachs, J. “Active Multiobject Exploration and Recognition via Tactile Whiskers”. In: *IEEE Transactions on Robotics* 38.6 (Dec. 2022), pp. 3479–3497. ISSN: 1941-0468. DOI: 10.1109/tro.2022.3182487. URL: <http://dx.doi.org/10.1109/TRO.2022.3182487>.