Doctoral Dissertation

Large-area Multimodal Soft Sensing Skin for Human-Robot Interaction

Quan Khanh Luu

Supervisor **Van Anh Ho**

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Materials Science

September 2024

Abstract

Soft-bodied robots with a sense of touch and multimodal sensing capabilities hold promise for the realization of fully autonomous, social, and human-friendly robotic systems. However, seamlessly integrating multimodal sensing functionalities into soft artificial skins remains a challenge due to compatibility issues between soft materials and conventional electronics. While vision-based tactile sensing has enabled efficient robotic touch, there has been limited exploration of this technique for intrinsic multimodal sensing in large-sized robot bodies. To address this gap, this study introduces a novel vision-based soft sensing technique, named *ProTac*, capable of operating either in tactile or proximity sensing modes, which relies on a soft functional skin that can actively switch its optical properties between opaque and transparent states. Compared to conventional sensing skins of various electronic elements, our system provides large-area multimodal sensing with a simple setup and minimal impact on the mechanical properties of the soft skin. Furthermore, this study proposes a novel learning mechanism to facilitate tactile inference on large-area robot bodies, alongside the development of a proximity sensing pipeline and multimodal sensing strategies. The effectiveness of the soft sensing technology is demonstrated through a soft *ProTac* link, which is integrated into newly constructed or existing commercial robot arms. Based on this framework, this study also explores the synergy between the robot's softness and its tactile-proximity sensing capabilities in facilitating task performance and enhancing safe interactions with the environment. Results suggest that robots integrated with the soft *ProTac* link, along with rigorous control formulation, are capable of mediating safe and purposeful control actions, which enhance safe interactions and facilitate motion control tasks that are challenging to achieve with conventional rigid robots.

Keywords: tactile sensing, multimodal perception, soft robotics, safety control, human-robot interaction.

Acknowledgment and Dedication

First and foremost, I would like to express my deepest gratitude to my supervisor, *Prof. Van Anh Ho*, who has inspired me a lot as a model researcher. His unwavering support, guidance, and expertise have been crucial in the completion of this dissertation. Starting my master's degree in 2019, I can barely remember how shy and lacking confidence I was in having open communication, sharing ideas, and expressing personal opinions. Throughout the five years under Prof. Van's guidance, his encouragement, trust in me with important tasks, and the opportunities he provided to connect me with peers across the world have not only improved my expertise but also significantly built my confidence and skills needed to nourish an independent researcher and an interdependent person as a whole. Therefore, with my deepest gratitude, my achievements today are dedicated to him.

I would like to extend my gratitude to my second supervisor, *Prof. Yonghoon Ji*, for his insightful suggestions on my research proposal and his support for this research topic. I also greatly appreciate *Prof. Minh Le Nguyen* and *Prof. Yorie Nakahira* for their kind advice, support, and guidance on my minor research project.

I would also like to extend my heartfelt thanks to my lab mates, especially *Dr. Nguyen Huu Nhan*, *Dr. Nguyen Quang Dinh*, *Mr. Le Dinh Minh Nhat*, *Mr. Nguyen Viet Linh*, and *Mr. Nguyen Thanh Khoi*, whose support has been crucial to the completion of this dissertation. That said, this journey would not have been complete without the assistance and encouragement of all other lab members.

Finally, to my cherished dad, *Mr. Luu An Da*, my beloved mom, *Mrs. Do Thi Ngoc Thanh*, and my dear sister, *Luu Khanh Vy*, I am profoundly grateful for your love, support, and sacrifices throughout this journey. I understand how disappointed you felt when I was not at home, and when setbacks made me feel down. But I want you to know that you are my motivation, driving me through difficult times. Please trust in me, and take care of yourselves, knowing that my accomplishments today are dedicated to you, with all my love and gratitude, Luu Khanh Quan.

Contents

List of Figures

List of Tables

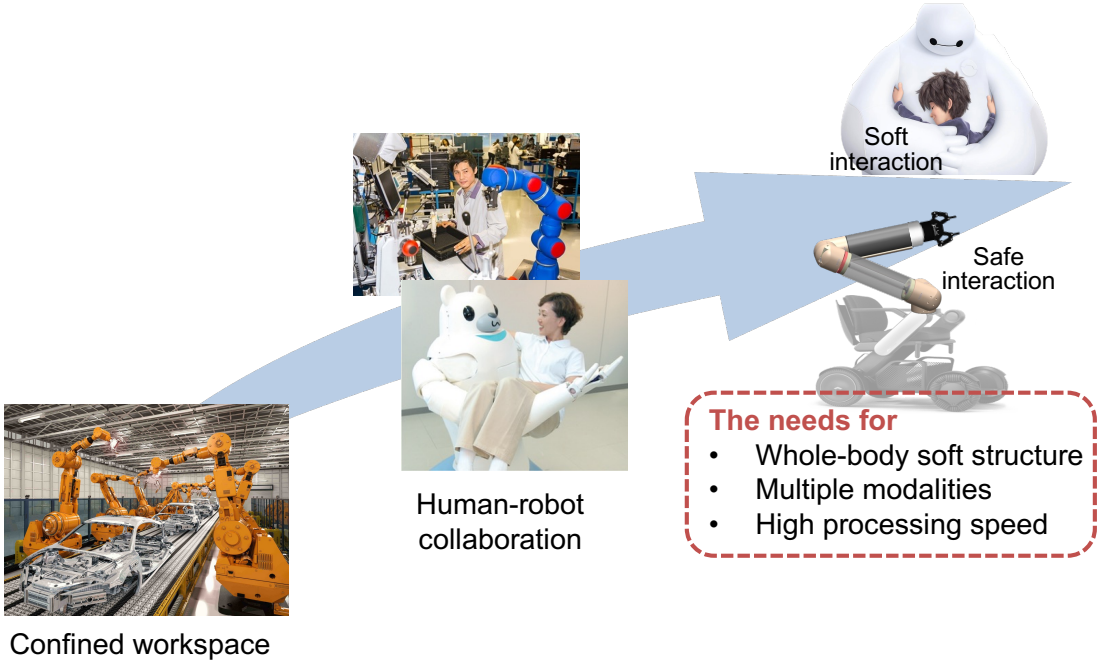
Chapter 1

Introduction

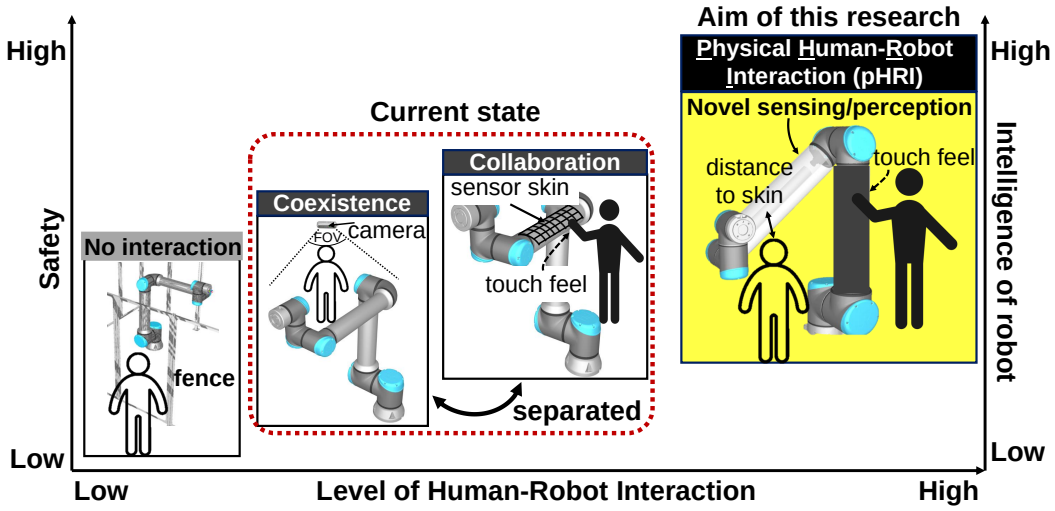
1.1 Background

Nowadays, there is a growing demand for robots to operate beyond safety enclosure zones, collaborating and working close to humans. These robots are anticipated to become versatile service assistants, seamlessly integrating into various aspects of our daily routines and industry sectors, including manufacturing, healthcare, agriculture, and others. Given these envisioned applications, it is crucial for such robots to exhibit adaptability to dynamic and unstructured environments, enabling safe and purposeful interaction with humans and surroundings. To achieve this objective, innovative robot structures employing soft materials, as in the so-called *soft robotics*, emerge as a pivotal solution for safe and adaptable interaction with the world, thanks to their inherently compliant nature. In addition, integrating *multi-modal* sensing into robotic systems to enhance their awareness of surroundings and interactions is a crucial factor for robots capable of engaging in purposeful human-robot and environment-robot interaction scenarios (see Fig 1.1a).

Of sensing modalities, **sense of touch** is crucial for tasks involving physical interactions, where it not only provides a diverse range of information such as interactive force, and texture but also is considered a non-verbal means of communication in human-human or human-machine interaction. Skin, the largest organ of the human body covering whole-limb or torso, possesses a tactile sensing system that has been inspiring the robotics community towards the creation of fully autonomous social and task-based machines with the sense of touch [20, 21]. However, large-area robotics skin developed for years faced complexity in system integration and data processing, since increasing the scale requires a great deal of embedded sensing



(a) Emergent needs of multimodal soft sensing skins for safe and pleasant interactions.



(b) Overview of research aim and current technologies

Figure 1.1: Positioning of this research within existing works

elements [4,22,23]. Recently, vision-based tactile sensors have emerged as an effective method for implementation of tactile sensing with reduced complexity in system design [24–26], which primarily applied to small-sized robotic devices (*e.g.*, robotic fingers or hands).

In addition to the tactile sensation, robots with **proximity perception** could further increase the safety of the robot. In fact, proximity sensing can avoid

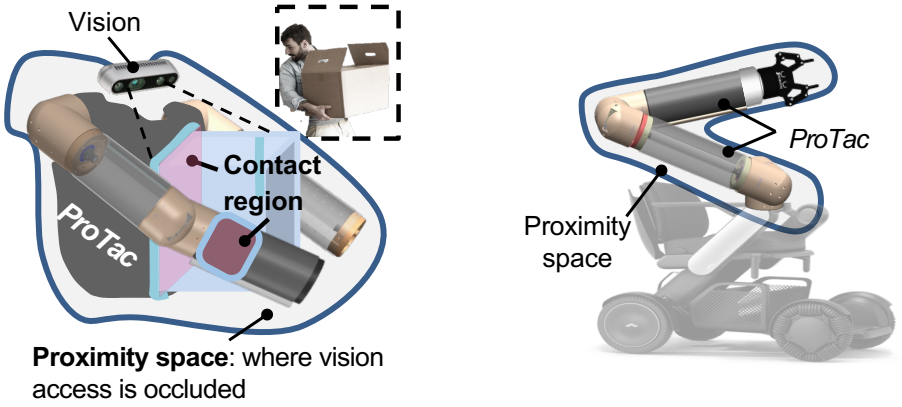


Figure 1.2: Concept of large-area/whole-body soft proximity-tactile sensing, its emerging needs, and prospective applications.

occlusions and blind spots in vision, and be a great complementary perception to the tactile modality [4] (see Fig. 1.2). To date, proximity sensation has been enabled through various transduction principles, such as resistance, capacitance, inductance, electromagnetic field strength, and light density [20]. However, the sensing performance in these technologies often behaves differently according to the material properties of target objects, which may cause difficulties in calibration and perception.

As a result, seamlessly integrating multiple sensing modalities into soft artificial skins remains a challenge due to compatibility issues between soft materials and conventional electronics. Therefore, this research aims to tackle this challenge by focusing on the development of a novel soft sensing mechanism and associated perception methodologies. Furthermore, it aims to demonstrate the potential applications of this advancement in robot task performances. The positioning of this research within the literature and its prospective applications are illustrated in Figure 1.1b and Figure 1.2, respectively.

1.2 Research questions

While soft bodies and multi-modal sensing are crucial for next-generation robots, seamlessly integrating tactile and proximity sensing, particularly into large-area soft bodies, has been underexplored. This leads to the first research question:

RQ1: What novel sensing mechanism can seamlessly facilitate tactile and proximity sensing for soft bodies at different scales, especially for large-area sensing?

Additionally, while there has been significant interest in enabling tactile sensing for robots, efficiently processing tactile information, particularly for large-sized soft bodies, remains challenging. This brings up the second research question:

RQ2: What learning mechanism can facilitate interpretation of the artificial sense of touch across a large robot body that is compatible with the proposed sensing mechanism identified in Question 1?

Finally, concerning high-level applications, conventional robots often encounter challenges in complex and safety-critical control tasks that involve close physical interactions with the environment, particularly in unstructured settings or when working closely with humans. These challenges, combined with the soft multi-modal sensing proposed in Questions 1 and 2, give rise to other crucial research questions:

RQ3: Can the efficiency and success of robotic tasks be enhanced by collectively leveraging the robot's **softness**, and **tactile-proximity sensing**? Can a soft-bodied robot equipped with active tactile sensing improve safety during interactions and exploration?

1.3 Originality and contributions

To address the aforementioned research questions, the key contributions of this dissertation are summarized as follows:

1. Design and fabrication of a soft proximity-tactile sensing device, named *ProTac*, which allows for the selective activation of either proximity or tactile sensing mode. This is made possible by utilizing vision-based sensing techniques and a unique mechanism of **skin transparency switching**, wherein a soft functional skin actively switches its optical properties between opaque and transparent states.
2. Development of a simulation and learning platform for vision-based tactile perception on large skin areas. In addition, the methodology for proximity sensing and perception is also proposed.

3. Showcase of *ProTac*-specific sensing strategies with two multimodal tasks, which aim to enhance motion control in cluttered environments and facilitate seamless human-robot interaction scenarios.
4. Integration of *ProTac* sensing for two safety control strategies, including creating reflex behavior and proximity-based adaptive speed regulation. The effectiveness is demonstrated using a *ProTac*-integrated robot arm.
5. Investigating the effectiveness of softness and tactile sensing in handling physical collisions, which validates the benefits of embodied soft tactile sensing in safety enhancement.

1.4 Significance

Softness and multimodal sensing are crucial for autonomous soft robots. However, their seamless integration and their interplay in task performance remain largely unexplored. This thesis attempts to explore whether softness and tactile-proximity sensing can enhance the efficiency and safety of robotic tasks. This is achieved through the novel design of a vision-based proximity-tactile sensing link with soft artificial skin, along with proposed platforms for learning perceptions and rigorous control frameworks. Additionally, this study aims to advance the field of sensing in soft robotics and establish the groundwork for exploring further robotic tasks that can be beneficial from softness and multimodal sensing. Lastly, the proposed sensing technology is expected to address a wide range of use cases and sectors in both industrial and service settings, which is challenging to achieve with sensorless or single-modal robotic systems alone.

1.5 Dissertation organization

The current chapter introduces the emergent needs and challenges of integrating tactile and multimodal perception for soft-bodied robots. Following this, research questions are presented, along with contributions to address them. The remaining chapters of this dissertation are organized as follows:

- Chapter 2 discusses existing techniques for implementing multimodal tactile

robotic skins. Also, relevant learning mechanisms of tactile perception will be discussed. Lastly, it outlines related applications of large-area robotic skins in robot control.

- Chapter 3 details the basic working principle, design criteria, and fabrication method for the proposed two-mode tactile and proximity sensing link, referred to as *ProTac* link.
- Chapter 4 describes in detail the proposed simulation and learning platform for large-area vision-based tactile sensing. The effectiveness of this method is demonstrated with the proposed *ProTac* link and another tactile link of a more complex shape.
- Chapter 5 presents a methodology for learning proximity perception specifically tailored for the *ProTac* link, followed by a performance evaluation.
- Chapter 6 explores the versatile use of the *ProTac* link in two different settings: first, as soft sensing links for a newly constructed robot arm, and second, as an extended link for a commercial robot arm. The integration aims to enhance control tasks in environment-robot, and human-robot interaction scenarios, leveraging the synergies of soft body and multimodal sensing.
- Chapter 7 addresses the question of whether a soft-bodied robot equipped with active tactile sensing can improve safety, and facilitate task performances.
- Chapter 8 concludes my thesis, summarizes findings, as well as discusses insights for future work.

1.6 Selected publications

A full list of my publications can be found in Google Scholar.

Journal publication:

[J1] Q. K. Luu, N. H. Nguyen and V. A. Ho, "Simulation, Learning, and Application of Vision-Based Tactile Sensing at Large Scale," in **IEEE Transactions on Robotics**, vol. 39, no. 3, pp. 2003-2019, June 2023, doi: 10.1109/TRO.2023.3245983.

[J2] S. T. Bui, Q. K. Luu, D. Q. Nguyen, N. D. M. Le, G. Loianno and V. A. Ho, "Tombo Propeller: Bioinspired Deformable Structure Toward Collision-

Accommodated Control for Drones," in **IEEE Transactions on Robotics**, vol. 39, no. 1, pp. 521-538, Feb. 2023, doi: 10.1109/TRO.2022.3198494.

IEEE Transactions on Robotics metrics: Impact Factor: 7.8, #2 journal in the field of Robotics by Google Scholar.

Conference paper:

[C1] Q. K. Luu, A. Albini, P. Maiolino and V. A. Ho, "TacLink-Integrated Robot Arm toward Safe Human-Robot Interaction," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), Abu Dhabi, UAE, 2024 (accepted).

[C2] T. T. Nguyen, Q. K. Luu et al., "ConTac: Continuum-Emulated Soft Skinned Arm with Vision-based Shape Sensing and Contact-aware Manipulation," *Robotics: Science and Systems* (RSS), Delft, Netherlands, 2024 (accepted).

[C3] Q. K. Luu, D. Q. Nguyen, N. H. Nguyen and V. A. Ho, "Soft Robotic Link with Controllable Transparency for Vision-based Tactile and Proximity Sensing," *IEEE International Conference on Soft Robotics*, Singapore, Singapore, 2023, doi: 10.1109/RoboSoft55895.2023.10122059.

[C4] Y. Osawa, Q. K. Luu, L. V. Nguyen and V. A. Ho, "Integration of Soft Tactile Sensing Skin with Controllable Thermal Display toward Pleasant Human-Robot Interaction," *IEEE/SICE International Symposium on System Integration* (SII), Ha Long, Vietnam, 2024, doi: 10.1109/SII58957.2024.10417383.

[C5] N. M. Dinh Le, Q. K. Luu et al., "Integration of Web of Tactile Things for Soft Vision-Based Tactile Sensor Toward Immersive Human-Robot Interaction," *IEEE/SICE International Symposium on System Integration* (SII), Ha Long, Vietnam, 2024, doi: 10.1109/SII58957.2024.10417344.

[C6] Q. K. Luu, H. M. La and V. A. Ho, "A 3-Dimensional Printing System Using an Industrial Robotic Arm," *IEEE/SICE International Symposium on System Integration* (SII), Iwaki, Fukushima, Japan, 2021, pp. 443-448, doi: 10.1109/IEEECONF49454.2021.9382645.

[C7] P. Van Nguyen, Q. K. Luu, Y. Takamura and V. A. Ho, "Wet Adhesion of Micro-patterned Interfaces for Stable Grasping of Deformable Objects," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), Las Vegas, NV, USA, 2020, doi: 10.1109/IROS45743.2020.9341095.

Under-review paper:

[UR1] Q. K. Luu, D. Q. Nguyen, N. H. Nguyen and V. A. Ho, "Vision-based Proximity and Tactile Sensing for Robot Arms: Design, Perception, and Control," under review for **IEEE Transactions on Robotics**, 2024 (revise and resubmit).

1.7 Patent

[P1] Van A. Ho, Quan K. Luu, N. H. Nguyen, "Contact-Proximity Detection Device, and Contact-Proximity Detection Method", Japanese Patent Application No. 2022-118796.

1.8 Honors and awards

2024 **JSPS Research Fellowship** for Young Scientists (DC2)

2024 **RSS Pioneer** - RSS Pioneers brings together top Ph.D. students, postdocs, and young industry members to foster creativity and collaborations surrounding challenges in all areas of robotics

2024 Finalist of Best Paper Award at 2024 IEEE/SICE International Symposium on System Integration (SII)

2021 Best Graduate Student for outstanding academic performance (Master course)

2021 MEXT Scholarship for Ph.D. course

2019 MEXT Scholarship for Master course

1.9 Supplementary materials

- Video demonstration on proximity-tactile sensing device and its applications (ProTac): https://youtu.be/5DhAhlTVxzg
- Video demonstration on simulation and learning platform for tactile sensing (SimTacLS): https://youtu.be/NN2u8YBLITY
- GitHub repository (ProTac): https://github.com/Ho-lab-jaist/protac.git
- GitHub repository (SimTacLS): https://github.com/Ho-lab-jaist/SimTacLS.git

Chapter 2

Related Work

2.1 Multimodal tactile sensor and sensing mechanism

2.1.1 Conventional technique

In the past decades, efforts to ensure safe human-robot interaction have primarily centered on conventional sensing technologies. For instance, collision monitoring and reactive strategies relying on proprioceptive sensors, such as force/torque sensors, have seen progressive development [27–29]. While effective in certain safety-critical and interactive scenarios [30, 31], these methods may prove inadequate in scenarios involving multiple contacts or complex interactions across large sensing areas. An alternative approach involves proactively avoiding collisions or planning collision-free robot trajectories, typically utilizing exteroceptive sensing devices such as onboard vision systems [32] or RGB-D/depth cameras [33, 34]. However, these methods are constrained by limited or obstructed sensing ranges and lack the tactile feedback necessary for close physical interactions and reliable collision detection.

2.1.2 Electronic skin

Tactile electronic skins, composed of arrays of distributed sensing elements employing diverse mechanotransduction principles (such as resistance, capacitance, inductance, electromagnetic field strength, and light density), have recently garnered significant attention [20]. This attention stems from their surface adaptability and scalability, enabling integration across various robotic components, ranging from small-scale robotic hands to larger body areas such as limbs or torsos [4, 35–38]. Notably, an integrated electronic skin constructed from networks of rigid hexagonal printed circuit boards has demonstrated the capability to provide sensory

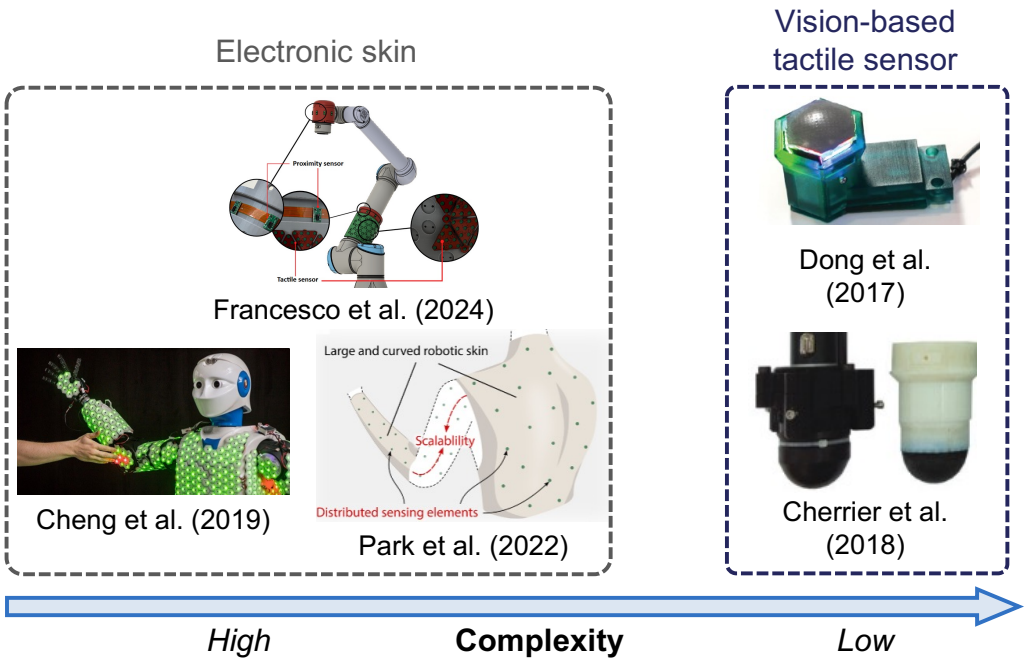


Figure 2.1: Overview of current technologies for robotic skins and tactile sensing. (Pictures adopted from [1–5]).

feedback across multiple modalities, including proximity, vibration, temperature, and light touch [39]. This has proven beneficial in various control frameworks and applications [3, 5, 40, 41]. However, the spatially distributed nature of such sensor networks, integrating numerous sensors and electronic components, poses challenges in fabrication. Moreover, not only are the acquisition and processing of data from these sensor networks highly intricate, but also some existing proximity sensing mechanisms depend heavily on the material properties of the target objects, leading to further challenges in calibration and perception. Additionally, the durability of these rigid electronic sensors is another concern when exposed to frequent physical contact. Lastly, seamlessly integrating such sensors into soft bodies also presents a significant challenge due to the incompatible interface of soft-rigid materials. Figure 2.1 highlights exemplary technologies for implementing robotic skins.

2.1.3 Vision-based sensing skin

Vision-based sensing technology, the so-called vision-based tactile (ViTac) sensor, has emerged as a viable approach for facilitating robotic touch perception with

minimal wiring and electronics, providing high spatial resolution at a reduced cost [24] (see Fig. 2.1). This method entails utilizing cameras to capture the deformation of artificial soft skin as a result of physical tactile stimuli, by exploiting visual features like reflective membranes or markers. These visual cues can be translated into tactile information, encompassing details such as contact location, force, vibration, object texture, and more (refer to [25] for a more thorough review). For instance, a family of GelSight sensors can accurately detect the detailed surface texture of a touched object, by processing tactile images documenting the deformation of a reflective membrane under RGB illumination, typically using photometric stereo algorithms [1, 6–8, 42–46]. On the other hand, GelForce sensors track reflective markers embedded inside elastomeric layers to infer traction fields or force distribution, which necessitates a learning or calibration process to establish the correlation between applied force and marker movements [47, 48]. Similarly, TacTip family [2], typically in a hemispherical shape, primarily relies on changes in the positions of printed markers and vision techniques to enable the sense of touch.

In addition to tactile sensing, Hogan et al. [9] introduced a novel concept of a visuotactile sensor capable of visualizing objects through the skin. Building upon this research, Jessica et al. [10, 49] proposed another iteration of a multimodal visuotactile sensor integrating RGB and Time-of-Flight (ToF) cameras. This sensor delivers both tactile feedback and proximity depth data by employing a selectively transmissive soft membrane. Additionally, recent advancements have led to the development of visuotactile sensors with various activation mechanisms, enabling multimodal tactile sensing and close-contact detection for robotic fingers [11, 50–52].

However, the aforementioned vision-based sensors were predominantly intended for small-scale devices with flat or curved sensing surfaces, typically deployed in robotic fingers for manipulation tasks. On the other hand, Lac et al. [12] introduced *TacLink*, a vision-based tactile link with highly soft skin and a large sensing area, where contact force can be inferred from skin deformation tracked through markers attached to the inner skin and two internal cameras.

Although TacLink holds promise of whole-arm tactile perception, there has been insufficient exploration of employing this vision-based method to enable multi-modal

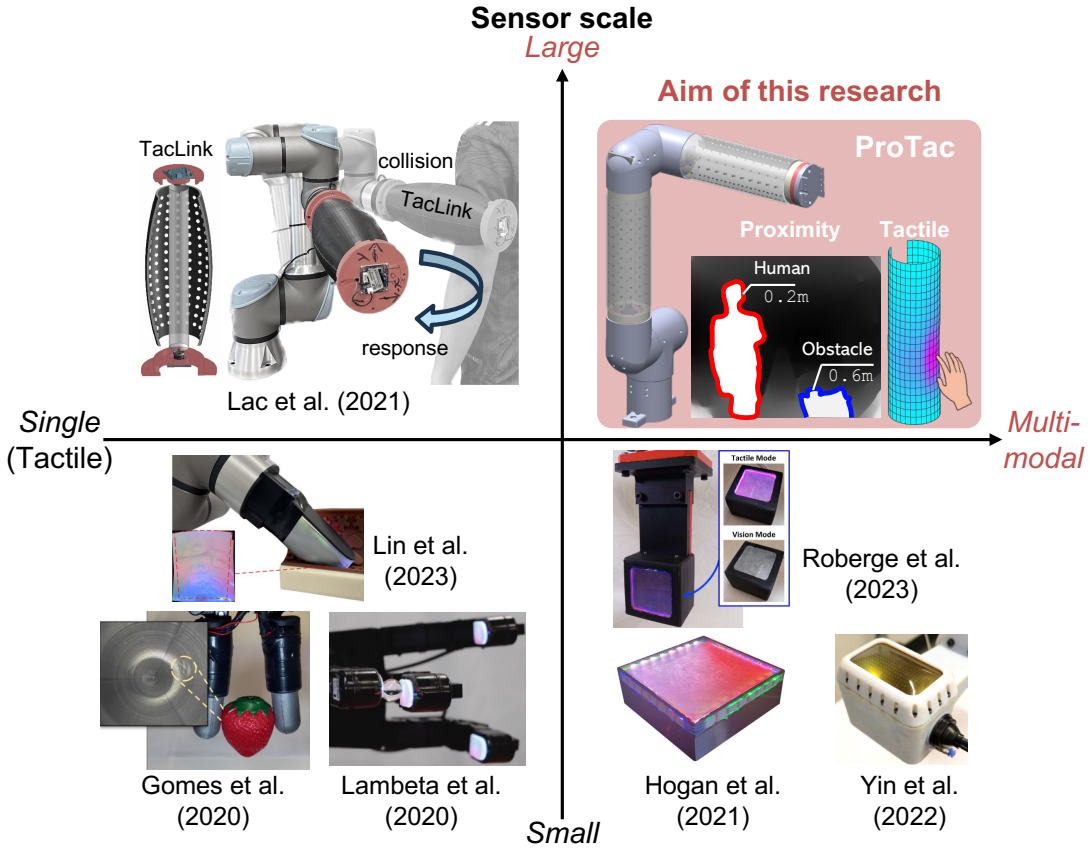


Figure 2.2: Exemplary vision-based sensing devices are categorized by their scale, applications, and the number of sensing modes, emphasizing the focus of this study on a large-area, multi-modal soft sensing skin. (Pictures adopted from [6–12])

sensing for large-scale soft robot bodies. Addressing this gap, we present the *ProTac* link, a vision-based two-mode robot link capable of seamlessly transitioning between proximity and tactile sensing modes. This functionality is facilitated by a soft functional skin with controllable transparency. Figure 2.2 highlights exemplary vision-based sensing devices categorized by their scale, applications, and the number of sensing modes, indicating this study's focus on the development of a large-area, multi-modal soft sensing skin.

2.2 Simulation and learning of vision-based tactile sensing

While recent advancements in vision-based tactile sensing and associated learning methods show promise for an efficient robotic sense of touch, training perception models for such tactile devices necessitates extensive tactile training datasets,

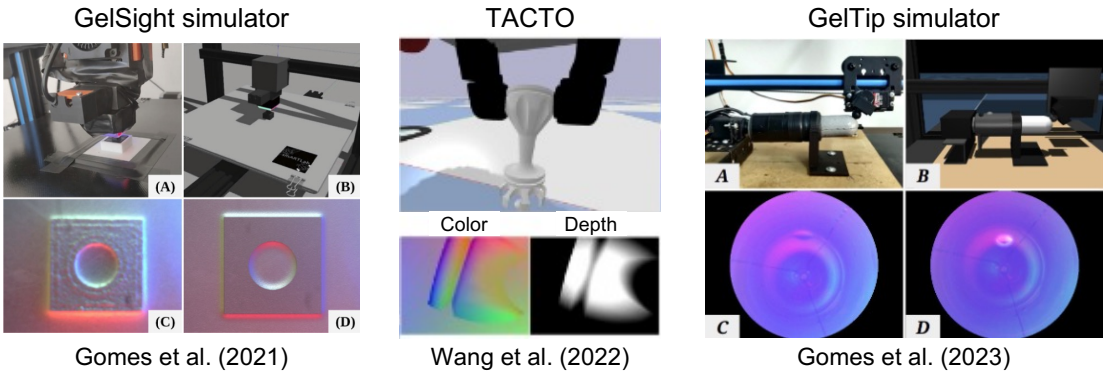
thereby adding complexity to the data collection process. Thus, there has been a surge in the development of simulation tools aimed at mitigating the need for laborious and time-consuming experimental setups to collect training data in tactile learning frameworks.

2.2.1 Tactile sensor simulation

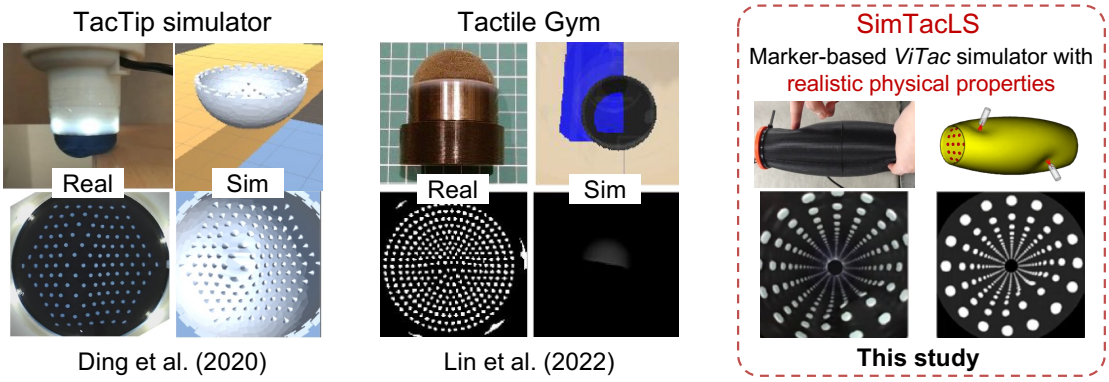
The ViTac sensors can be broadly categorized into two groups: those leveraging reflective light and those depending on the positions of visual markers to interpret tactile information, exemplified by GelSight [1] and TacTip [2] sensors, respectively (see Section 2.1.3 for more detail).

With respect to the former, Agarwal, *et al.* [53] and Gomes, *et al.* [54] employed physics-based models to simulate the optical responses of GelSight sensors upon contact with object surfaces. Similarly, Wang et al. introduced TACTO [55], an open-source simulation framework that integrates the PyBullet physics engine and Pyrender rendering engine, which was validated with two commonly used finger-sized ViTac sensors, namely OmniTact [43] and DIGIT [7]. Alternatively, Tacxim [56] simulates GelSight sensors using example-based photometric stereo in a data-driven manner, incorporating the inherent noise characteristic of real sensors. Furthermore, recent research efforts [57, 58] have aimed to enhance the performance of GelSight sensor simulations, as well as extend the simulation capabilities for sensors of round sensing surfaces. However, the previously mentioned simulators primarily concentrate on simulating the optical responses of GelSight sensors, neglecting the realistic reproduction of elastomeric skin deformation. Consequently, their contributions are not directly applicable to sensors that utilize marker-based tactile sensing with significant skin deformation like TacTip [2] or larger-scale ones like TacLink [12]. Figure 2.3a provides a brief review of optical simulators for light-based ViTac sensors.

Concerning marker-based ViTac sensors, which are more directly comparable to this study, a challenge arises in accurately replicating the deformation of soft skin with resultant markers' movements the movement in response to physical stimuli. This necessitates a thorough modeling of contact mechanics and material properties.



(a) exemplary optical simulators for ViTac sensors (Pictures adopted from [54, 55, 57])



(b) simulators for marker-based ViTac sensors and the proposed approach (Pictures adopted from [59, 60])

Figure 2.3: Exemplary simulators for ViTac sensors and this study's contribution for marker-based ViTac simulation with the reproduction of realistic physical properties.

Study in [59] endeavored to address this challenge by constructing and simulating the elastic behavior of the TacTip skin using the Unity physics engine. However, they employed a custom linear elastic model to approximate the skin's elastic properties. Another framework, named Tactile Gym, was proposed to generate virtual representations of physical contacts through depth imprints using a rigid contact model [60, 61], which was validated using finger-sized TacTip sensors of either hemispherical or rectangular shapes. Conversely, simulators that acquire ground-truth tactile data based on commercial Finite Element Method (FEM) simulators (*e.g.*, Abaqus) offer a systematic approach to tackle the challenge by discretizing the soft body into numerous sub-elements, which are then dynamically analyzed using hyper-elastic material models [62, 63]. However, the prohibitive

computational expenses and the limited interface of commercial FEM simulators constrain the practical application of these methods in real-time scenarios. Also, a notable limitation was the inadequate rendering of cases involving significant skin deformation upon contact with the environment, a critical issue in complex contact scenarios, compromising the accuracy of the sim2real transfer process. Lastly, Figure 2.3b highlights a couple of simulators for marker-based ViTac sensors and the contribution of this study to the simulation of large-area markered-based ViTac sensors with realistic physical properties.

2.2.2 Simulation-to-reality learning

Vision-based deep neural networks, trained on virtual or synthetic images, often exhibit suboptimal performance when evaluated with real-world visual inputs [64]. Discrepancies between simulated and real images, including unrealistic texture, color, and lighting conditions, are inevitable. To mitigate this sim2real (simulation-to-reality) gap, previous studies have implemented *domain randomization* to introduce variability in visual attributes within simulation environments, a strategy that has proven successful in various vision-based robotic applications [65] as well as small-scaled tactile sensors [54, 59]. On the other hand, some studies have adopted image-level *domain adaptation* techniques to enhance sim2real learning tailored for small-sized marker-based tactile sensors. For instance, [61] utilized a generative adversarial network to convert real marker-based tactile images into depth-based simulation images. However, the significant discrepancy between the two domains may lead to inaccurate reproduction of artificial skins under complex deformation states, especially those with intricate morphologies. Moreover, domain adaptation methods have been explored for GelSight sensors, focusing primarily on the optical properties of the sensors [66, 67], which is less relevant to marker-based sensors characterized by highly deformable skin like TacTip and TacLink/ProTac sensors.

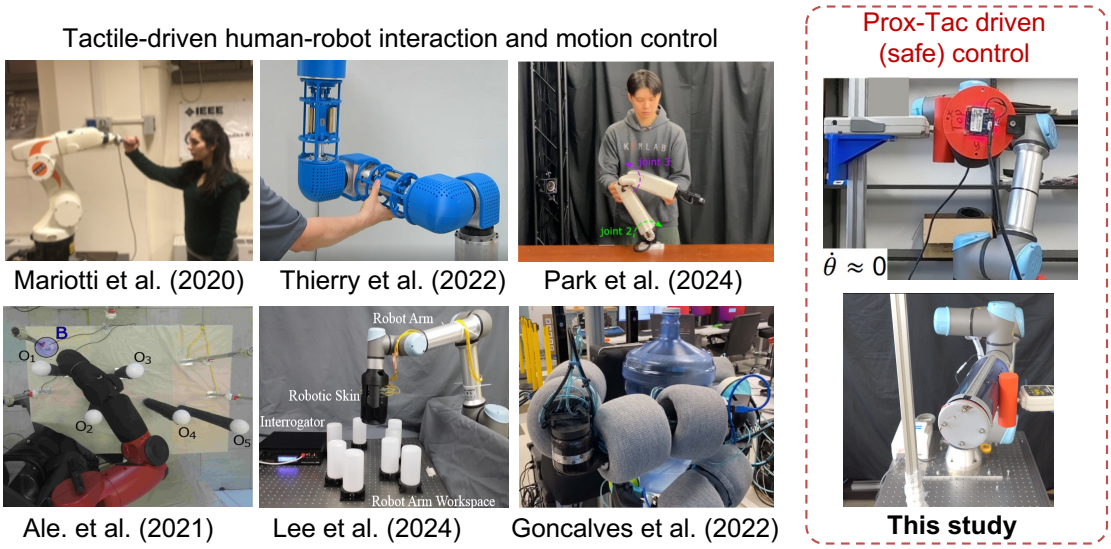


Figure 2.4: Exemplary robot control tasks driven by whole-arm tactile sensing, and the aim of this study for ProTac-driven robot control. (Pictures adopted from [13–18])

2.3 Multimodal sensing for robot control

On one hand, the ViTac sensors have demonstrated utility in small-scale manipulation tasks involving robotic hands or fingers [46, 68–70]. On the other hand, the utilization of conventional robot arms equipped with force/torque sensors [13, 31, 34], or electronic tactile skins [3, 15, 17, 18, 71] for physical human-robot interaction (pHRI) scenarios has also been extensively investigated in the past decade. Furthermore, whole-arm tactile sensing often finds applications in robotic links/arms where tactile information is beneficial for contact-rich manipulation tasks. For instance, systems have been developed to control a robot arm manipulating in cluttered environments where contact with surroundings is inevitable [14, 72]. Additionally, dual robot arms equipped with whole-body tactile sensing have also been showcased with manipulation of large-sized objects [16]. In the context of utilizing soft tactile skins for safe collision responses, the combined effect of a passive soft layer and active collision detection of an electronic tactile skin on eliminating impact forces has been thoroughly examined [73]. Nevertheless, the impact of softness and multimodal sensing of large-sized robot skins on robot task performances remains largely unexplored. Figure 2.4 reviews a handful of robot controls based on whole-arm tactile sensing and the aim of this study toward ProTac-driven control tasks.

Chapter 3

Soft Robotic Skin with Vision-based Tactile and Proximity Sensing: a Case Study on Robotic Links

This chapter explores a novel sensing mechanism that facilitates tactile and proximity sensing for large-sized soft robot bodies. Revisiting prior research on a compliant tactile sensing link (referred to as TacLink) with a dark skin [12], this device embeds cameras **inside** a soft skin to monitor its deformation by tracking the movement of markers upon contact with the environment. This study brought up the idea for the development of a novel two-mode sensing technology with a question:

"*What if the transparency of the sensor skin could be actively controlled, then the inside cameras may select to see the markers for tactile sensing mode, or observe the surrounding conditions for proximity sensing mode?*"

This chapter presents the design, underlying principle, and fabrication approach for a novel soft robotic skin, named *ProTac*, featuring vision-based proximity-tactile sensing. These capabilities are achieved by a soft functional skin that can actively switch its optical properties between **opaque** (not able to be seen through) for the *tactile* sensing mode and **transparent** (able to be seen through) for the *proximity* sensing mode. While *ProTac* sensing technology exhibits potential for robot bodies of diverse shapes and sizes, this thesis focuses on its application in a large-scale sensor design resembling a cylindrical skin shape, referred to as the *ProTac* link. This design choice mirrors the structure of lightweight industrial robot arms, making it practical for evaluating sensing algorithms, control strategies, and applications. The conceptual illustration of the *ProTac* technology is depicted in Figure 3.1.

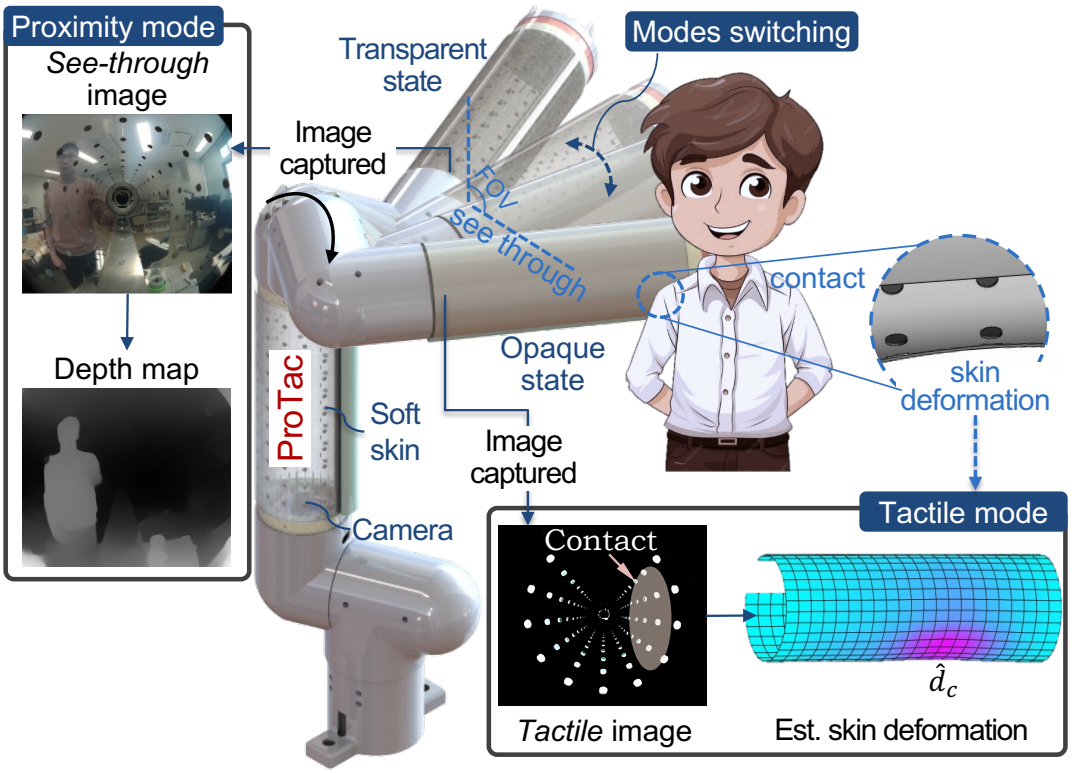
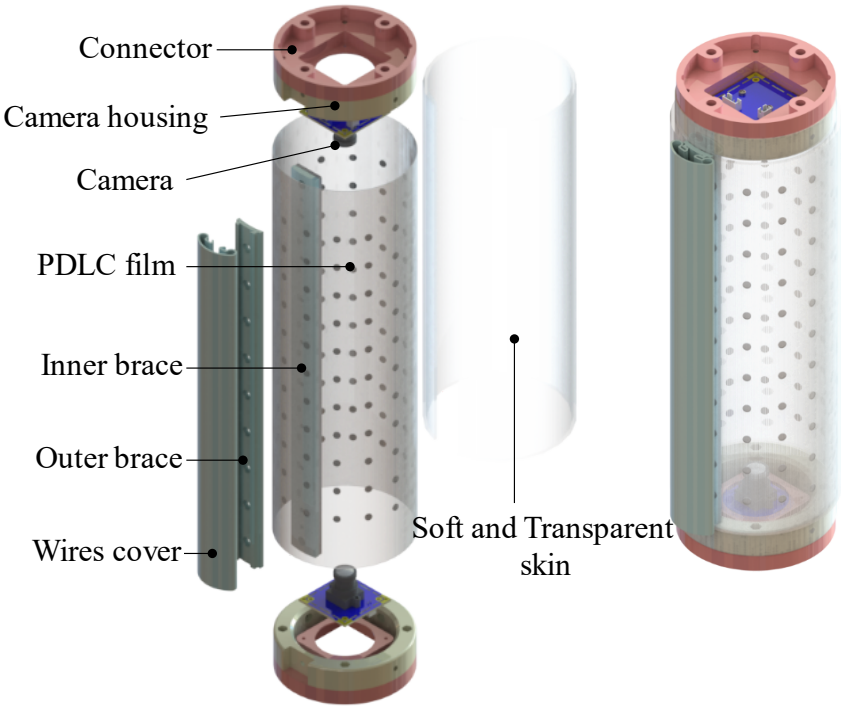


Figure 3.1: The conceptual overview of the vision-based *ProTac* sensing technology. *ProTac* can actively switch between proximity and tactile sensing modes, relying on input images captured by inner cameras and a *soft* functional skin with controllable transparency.

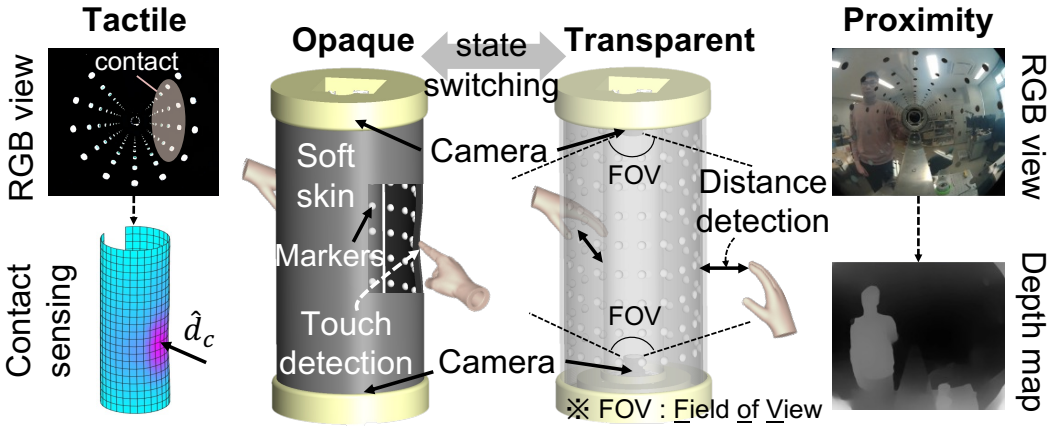
3.1 Design and working principle of ProTac

Figure 3.2a outlines the configuration of a *ProTac* link. The switchable proximity-tactile sensing functionality is achieved through internal cameras positioned at both ends and a soft functional skin capable of switching between opaque and transparent states. To realize this capability, the soft skin is structured with layers, comprising an outer transparent silicone layer and an inner flexible polymer-dispersed liquid crystal (PDLC) film, on which arrays of reflective markers are attached. The outer layer is designed to be soft and transparent to enhance the tactile experience and enable the see-through function of the *ProTac* skin. The inner PDLC film can actively transition between opaque and transparent states by applying an external voltage, with a rapid transition time of approximately 0.3 seconds. Consequently, the basic working principle of *ProTac* link is as follows (refer to Fig. 3.2b):

- *Tactile* mode: When the soft PDLC skin is in the *opaque* state, the camera can observe the movement of the markers so that the tactile information can be estimated without external light interference (refer to Chapter 4).
- *Proximity* mode: When the PDLC skin switches to the *transparent* state, the internal cameras can *see through* the skin so that the proximity information of nearby obstacles can be extracted from image views (see Chapter 5).



(a) Illustration of *ProTac* link's design



(b) Illustration of *ProTac*'s basic working principle

Figure 3.2: Design and working principle of the *ProTac* link.

In order to improve markers' visibility during tactile mode, a set of LEDs is positioned in a circular pattern inside the camera housing. These LEDs are switched off when the system operates in proximity mode to enhance the see-through effect. Additionally, mechanical housings and braces are employed to secure the cameras, as well as to shape the cylindrical skin.

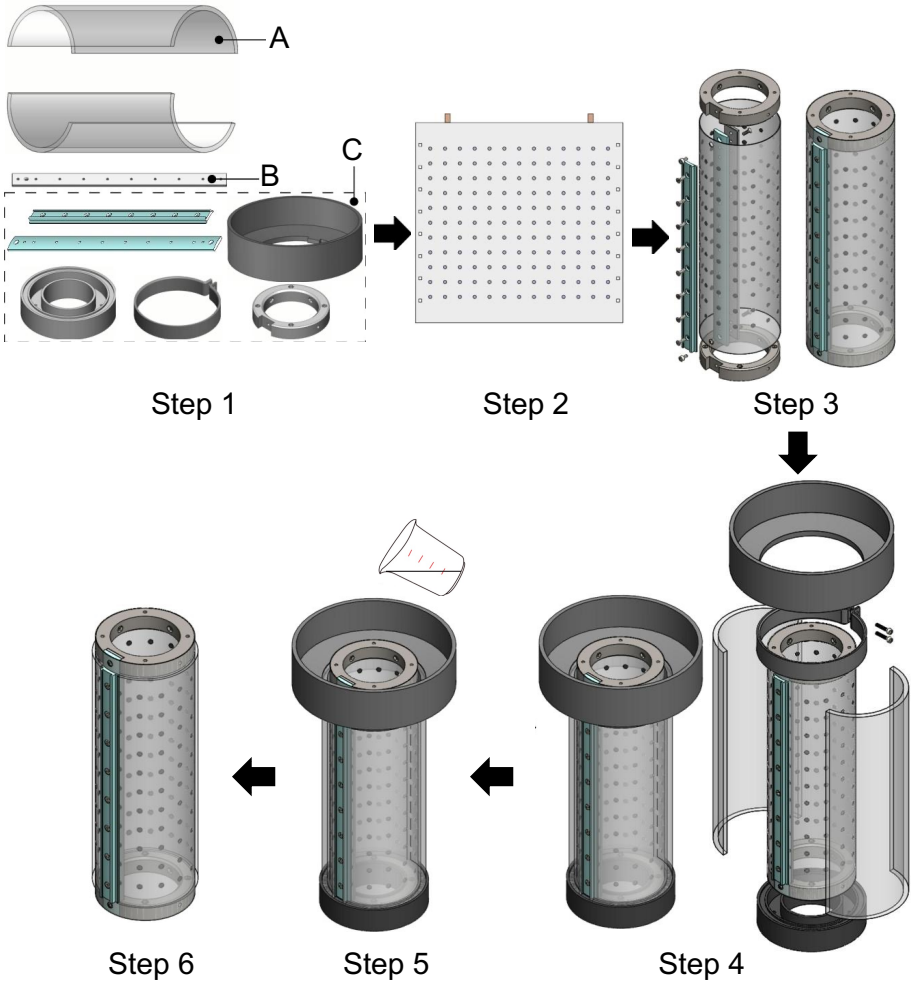


Figure 3.3: Fabrication process of the *ProTac* link. Step 1 - Preparing parts (A - Part was fabricated by laser cutting. B - Part was fabricated by machining cutting. C - Part was fabricated by 3D printing technique). Step 2 - Reflective markers arrangement onto a PDLC film. Step 3 - Shaping the PDLC film. Step 4 - Molding assembly. Step 5 - Pouring deformable and transparent silicone. Step 6 - Releasing mold for a finished *ProTac* sensor.

3.2 Fabrication of ProTac link

The entire fabrication process is illustrated in Figure 3.3. The proposed fabrication process aims for the desired durability and payload capacity of the soft *ProTac* link.

Here, structural analysis ensures its robustness under loads below 15 N. Additionally, transparency, soft skin uniformity, and marker reflectivity are crucial specifications affecting see-through ability and contact detection in proximity and tactile modes. To ensure high transparency, a commercial acrylic tube with a smooth surface finish is used for the outer mold, enhancing the efficiency of the see-through effect. Additionally, markers with a 3 mm diameter made from reflexive tape (R25 WHI, 3M Company) are employed to ensure performance in tactile mode. With these specifications in mind, we propose a fabrication process that involves six main steps (refer to Fig. 3.3). At first, laser cutting is used to form the outer mold (part A) into two halves, facilitating easier release upon separation. In step two, reinforcing braces (part B) are machined from steel, while other parts (C) are 3D-printed with PLA material. Subsequently, the marker matrix is adhered to the PDLC film (LC Magic, TOPAN Inc., Japan), forming a cylindrical skeleton with camera housing at both ends. The outer soft skin is constructed by filling the mold with transparent silicone liquid (Zoukei-mura, Japan) and curing it for a minimum of 24 hours. Finally, all mold components are removed and cameras are assembled to obtain a complete *ProTac* link.

ProTac is implemented with fish-eye cameras (ELP, 180° lens, 30 Hz), and a PC (Intel(R) i9-12900K 3.19 GHz, 64GB RAM, NVIDIA RTX 3090 GPU). The control for switching the skin transparency is regulated by the PC, which connects to the power control unit (LP1, TOPAN Inc., Japan) of the PDLC film through an RS232 serial port.

3.3 Structural analysis of ProTac link

In this section, we examine the structural robustness of the designed *ProTac* link. The structural analysis is conducted using FEM simulation in Abaqus[1], where the *ProTac* link is tested under compression, bending, and twisting loads (see Section 3.3.1). Lastly, the true failure points of the *ProTac* link are also verified through experimental loading tests in Section 3.3.2.

[1]Finite Element Analysis Software: https://www.3ds.com/products/simulia/abaqus

Table 3.1: The properties of materials used for *ProTac*'s structural simulation.

	Reinforcing brace	PDLC skin
Material	Stainless steel (SUS304)	PET plastic
Young modulus	210000 MPa	2164.8 MPa
Poisson ratio	0.30	0.35
Density	7930 kg/m^3	1345 kg/m^3

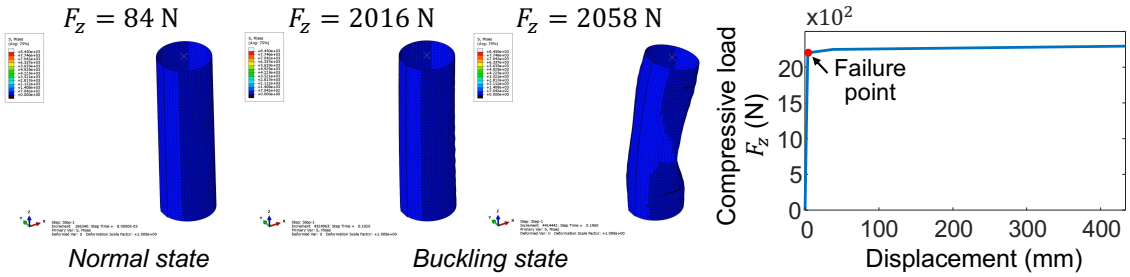
3.3.1 Simulation

Settings

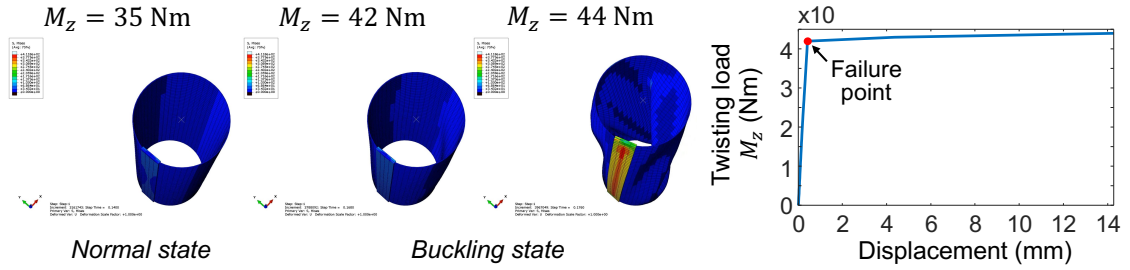
In the simulation, the model of the *ProTac* link is simplified to two core structural components: the reinforcing brace made from stainless steel (SUS304) and the PDLC skin modeled as a thin PET plastic sheet. The properties of the materials used for the structural simulation are summarized in Table 3.1. The simulations are conducted under three different static loading scenarios, including compressive, bending, and twisting loads. The loads are applied at one end of the *ProTac* link while the other end is fixed. The simulations are designed to gradually increase the respective loads, and the displacements of a selected point on the reinforcing brace are recorded to observe the *ProTac* link's deformation behavior.

Results

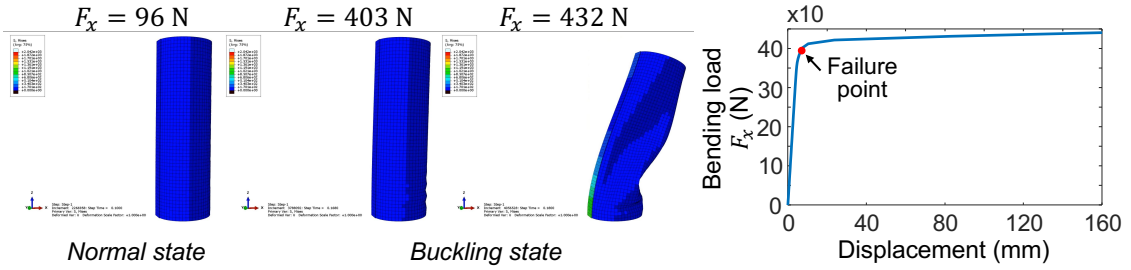
Figure 3.4 highlights the simulation results of the *ProTac* link's structural behaviors under compressive, twisting, and bending loads. Based on the load-displacement curves, the failure points of the *ProTac* structure can be determined for each type of applied load, indicating the load at which the soft *ProTac* skin begins to buckle. It can be seen that while the *ProTac* link exhibits strong robustness under compressive and twisting loads, it demonstrates structural weakness under bending load conditions, where the skin shows a high potential to buckle at the edge of the *ProTac* skin. This observation is confirmed by the experimental evaluation presented in the following section.



(a) Compressive load



(b) Twisting load



(c) Bending load

Figure 3.4: Simulation results of the *ProTac* link's structural behaviors under compressive, twisting, and bending loads. The failure point indicates the load at which the soft *ProTac* skin begins to buckle.

3.3.2 Experiment

The experiment aims to verify the deformation behaviors and failure points of the *ProTac* link under critical bending loads. It should be noted that due to the limitations in the loading capacities of the experimental system, it is difficult to confirm the failure points for compressive and twisting loads, which demonstrated high capacity through the simulations. Therefore, given that the *ProTac*'s structure is much weaker under bending loads, it is still reasonable to focus solely on verifying the *ProTac*'s flexural strength.

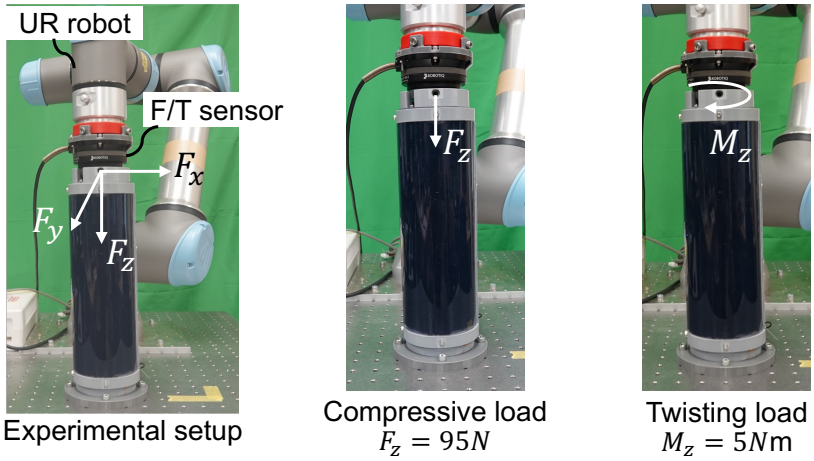


Figure 3.5: Experimental setup and measurements of the *ProTac* link's structural robustness under compressive and twisting loads.

The experimental setup is depicted in Figure 3.5. In this setup, the bottom end of the *ProTac* link was fixed to the table, while the other end was attached to a 6-degree-of-freedom force/torque sensor (Robotiq FT 300), which was motorized to apply forces using the UR5 robot arm. While it is infeasible to measure the *ProTac*'s failure points under compressive and twisting loads with the current experimental setup, Figure 3.5 shows that the *ProTac* is able to withstand compressive and twisting loads of 95 N and 5 Nm, respectively.

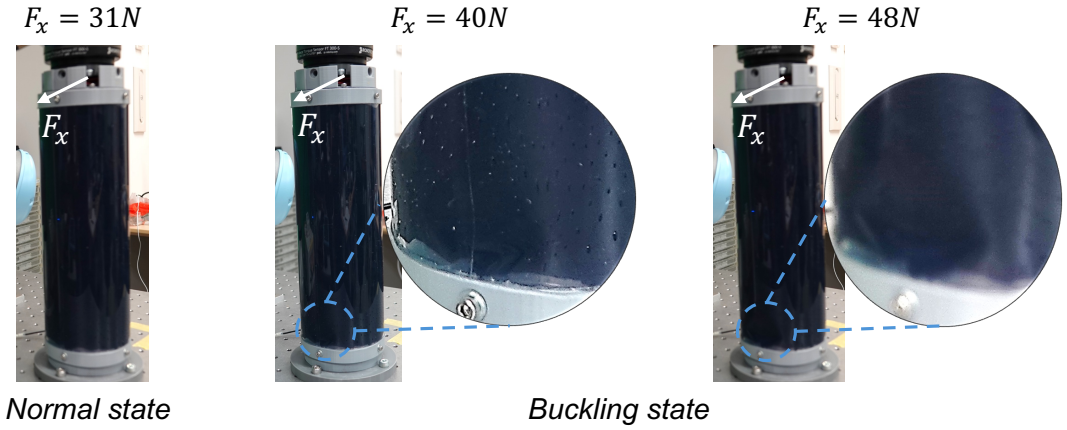


Figure 3.6: Measurement of the *ProTac* link's structural robustness under bending load, demonstrating its yield strength of around 40 N.

In terms of the bending load, Figure 3.6 shows that the *ProTac* link begins to buckle at the edge of the soft PDLC skin when the force applied along the x-axis exceeds 40 N. This indicates that the *ProTac* link's failure point under bending load

is approximately 40 N.

It should be noted that while the results confirm the weakest point in the *ProTac* structure under bending loads, and demonstrate satisfactory robustness under compressive and twisting loads, the observed failure points are inconsistent with the simulation results. This inconsistency can be attributed to imperfections in the experimental setup that cause measurement errors, particularly where all the degrees of freedom of the *ProTac* link cannot be completely fixed in bending experiments. However, since the *ProTac* link is still in its early stage of development, these preliminary results could sufficiently provide a glimpse into its structural weakness, which is valuable to guide further structural improvements. Upon improvements, a more comprehensive structural analysis will be necessary to advance the *ProTac* link towards commercialization and industrial-grade products.

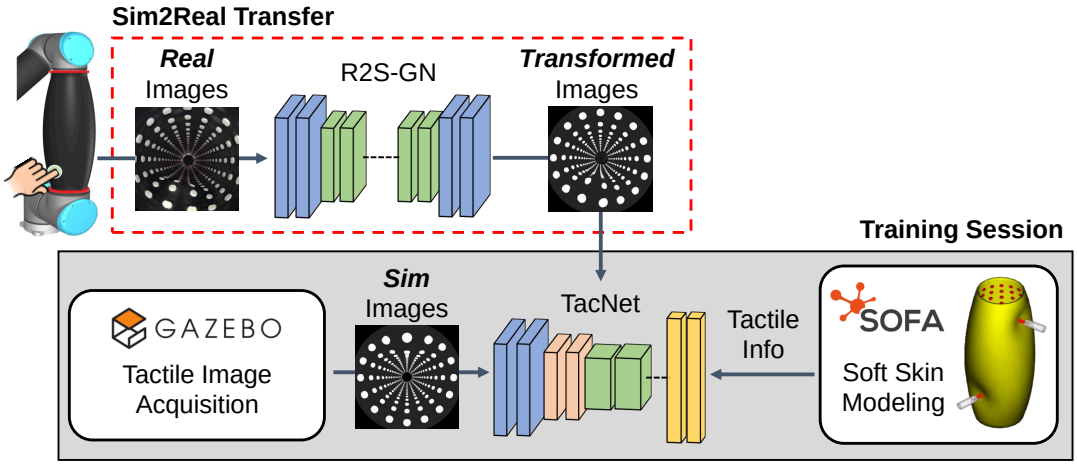
Chapter 4

Simulation, Learning of Vision-based Tactile Sensing

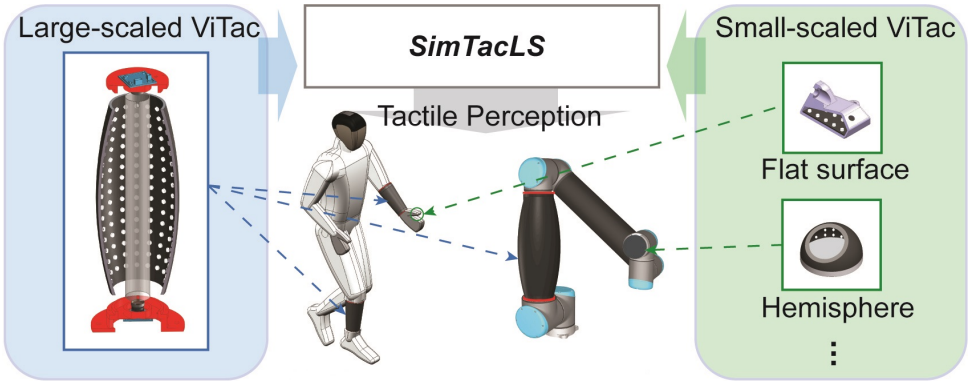
This chapter quests for a learning mechanism to facilitate the interpretation of the sense of touch on large-area artificial skins enabled by the vision-based tactile sensing principle. In recent years, vision-based tactile (ViTac) sensors have emerged as an efficient approach for implementing tactile sensing due to their simple design [24–26]. These sensors rely on the detection of soft artificial skin deformation upon contact with objects, achieved through optical tracking of visual features such as markers or reflective membranes. This information is then translated into tactile data, including contact location, force, vibration, and object texture.

Previously, we explored both analytical [12, 74] and supervised learning techniques [75, 76] for the vision-based tactile link (TacLink) to extract contact information from tactile images. While the former approach, involving model analysis and calibration, can achieve high sensing performance, its complexity in modeling and processing makes it less desirable. Conversely, data-driven methods, like the latter approach, require extensive data for labeling visual representations, necessitating a labor-intensive experimental data acquisition process [75]. This challenge becomes more profound in scenarios with larger skin areas and more intricate contact scenarios. Consequently, there is a growing need for a framework that enables simulation-based learning while accurately representing the physics of interactive contact in vision-based tactile sensing systems.

In this study, we introduce a novel platform, named SimTacLS, to simulate and learn vision-based tactile sensing for large-area marker-embedded compliant skin (see Fig. 4.1a). This platform leverages the SOFA physics engine to model the physical interactions of deformable tactile skins using the finite element method (FEM),



(a) *SimTacLS*: Simulation pipeline for large-scale ViTac sensors.



(b) Scalability and extendibility of *SimTacLS* platform.

Figure 4.1: *SimTacLS* overview. (a) A simulation pipeline, comprised of physics engines SOFA and Gazebo; was constructed to collect a labeled simulation dataset to train the TacNet model, including the information of tactile skin deformation (output) and virtual images (input); and a scheme of sim2real transfer learning was done through a generative network (R2S-GN) of real images into simulation ones. (b) Expected applications of *SimTacLS* to vision-based tactile sensors of diverse shapes and sizes.

while a plugin in the Gazebo environment facilitates the generation of virtual tactile images through the modeled skin geometries produced from the SOFA environment. The simulated images and corresponding skin deformation are employed as training input and output (labels) for a deep neural network named TacNet. Furthermore, to extend the TacNet model's efficacy to real-world tactile images, a couple of *sim2real* learning techniques are deployed. First, we introduce a real-to-simulation generative network (R2S-GN). This network employs a generative adversarial network (GAN) architecture to learn the transformation process from real to simulation domains of tactile images (Fig. 4.1a). Second, the domain randomization technique to

diversify perspectives of simulation tactile images is also examined to enable zero-shot learning of the TacNet model. Such a platform is envisioned as an easily implementable approach for diverse robotic systems of varying scales to attain tactile perception capabilities (see Fig. 4.1b).

The chapter presents the proposed platform in detail, starting with Finite Element (FE) modeling of the soft skin, particularly focusing on deriving a representation for multi-layered *ProTac* skin (refer to Section 4.1). Subsequently, the process of data collection and generation for labeled simulated tactile images along with corresponding global skin deformation is explained (see Section 4.2). Following this, the learning of skin deformation based on the simulation dataset and sim2real learning techniques are described in Section 4.3 and Section 4.4, respectively. Section 4.5 presents a methodology for extracting multi-point local contact information from the prediction of global skin deformation. The chapter concludes with the validation of this learning mechanism for the **tactile sensing mode** of the *ProTac* skin (see Section 4.6), as well as demonstrating its capability of learning tactile perception for a more complex-shaped soft tactile skin (Section 4.7).

4.1 Soft multi-layered skin modeling

This study employs a Finite Element (FE) algorithm through the *SOFA* simulation framework[1] for modeling the *ProTac* skin represented as a soft multi-layered structure. The key challenge lies in simulating the mechanical responses of the soft multi-layered skin to physical stimuli while ensuring a balance between accuracy and computational cost. Additionally, modeling the mechanical coupling between the PDLC film and the outer elastomeric layer is of significance. This section will elaborate on the proposed modeling strategy and the model's integration for physical interactions within the SOFA framework.

[1] https://www.sofa-framework.org/

4.1.1 Elastomeric skin modeling

The softness of an elastomeric skin presents a significant challenge in mechanical modeling due to the intrinsically nonlinear nature of soft materials. While hyper-elastic material models available in standard simulation platforms offer viable solutions [63, 77], accurately determining all required parameters through experimental means demands substantial effort. Furthermore, given our aim to implement the proposed platform in real-time robotic applications, computational efficiency is paramount. This study considers an FE modeling where connectivity among vertices of non-overlapping tetrahedron elements follows a linear constitutive relationship, characterized by Young's modulus E and Poisson's ratio ν. Experimental determination set E at $0.1\,\mathrm{N/mm^2}$ and ν at 0.49 [78]. To address potential unrealistic simulation outcomes stemming from this linear assumption, especially under significant deformations encompassing both large displacements and rigid rotations, a co-rotational FEM formulation is employed (for detailed elucidation, refer to [79]). This approach enables realistic simulations capturing the geometric nonlinearity of hyper-elastic materials, where small stresses result in large strains, in a computationally efficient manner.

At a given simulation time, the current geometrical state of a deformable elastomeric body can be obtained by solving the following dynamic equation:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{F}^{ext}(t) - \mathbf{F}^{\mathrm{int}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}^T\boldsymbol{\lambda}, \tag{4.1}$$

where $\mathbf{q} \in \mathbb{R}^n$ represents the 3D position of element nodes (N degrees of freedom), $\mathbf{M}(\mathbf{q})$ denotes the mass matrix, $\mathbf{F}^{ext}(t)$ signifies external forces (*e.g.*, gravity) at each time step t, and $\mathbf{F}^{int}(\mathbf{q}, \dot{\mathbf{q}})$ embodies internal forces acting on the system state. Equation 4.1 is integrated over a time interval $[t_1,\ t_2]$ using a backward Euler integration scheme [80]:

$$\mathbf{M}(\dot{\mathbf{q}}_2 - \dot{\mathbf{q}}_1) = dt\left(\mathbf{F}^{ext}(t_2) - \mathbf{F}^{int}(\mathbf{q}_2, \dot{\mathbf{q}}_2) + \mathbf{J}^T\boldsymbol{\lambda}\right). \tag{4.2}$$

By substituting the linearization of internal forces $\mathbf{F}^{int}(\mathbf{q}_2, \dot{\mathbf{q}}_2)$ using Taylor series

expansion with a first-order approximation and employing two relations $\dot{\mathbf{q}} = \mathbf{q}_2 - \mathbf{q}_1 = dt\dot{\mathbf{q}}_2$ and $\ddot{\mathbf{q}} = \dot{\mathbf{q}}_2 - \dot{\mathbf{q}}_1$ into Equation 4.2, we obtain:

$$\underbrace{\left(\mathbf{M} + dt^2\mathbf{K} + dt\mathbf{C}\right)}_{\mathbf{A}}\underbrace{\ddot{\mathbf{q}}}_{\mathbf{x}} = \underbrace{-dt^2\mathbf{K}\dot{\mathbf{q}}_1 + dt\left(\mathbf{F}_2^{ext} - \mathbf{F}_1^{int}\right)}_{\mathbf{b}} + dt\mathbf{J}^T\boldsymbol{\lambda}, \tag{4.3}$$

where $\mathbf{F}_2^{ext}$ denotes the external force at the subsequent time step, $\mathbf{K} = \frac{\partial \mathbf{F}^{\text{int}}}{\partial \mathbf{q}}$, and $\mathbf{C} = \frac{\partial \mathbf{F}^{int}}{\partial \dot{\mathbf{q}}}$ represent stiffness and damping matrices, respectively. The only unknown factor is $\mathbf{J}^T\boldsymbol{\lambda}$, which signifies the contribution of tactile interaction in the form of constraints. The Jacobian matrix $\mathbf{J}(\mathbf{q}) = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{q}}$ incorporates the normal and tangential constraint directions of $\boldsymbol{\lambda}$ (*i.e.*, contact forces) - equivalent to the magnitude of contact forces projected to the mapped degrees of freedom. Here, the contact responses adhere to a combination of Signorini's frictionless contact law [81] and Coulomb's frictional law [82], further details into this specific procedure can be found in Appendix A.

To solve the linear equations presented in Equation 4.3, the SOFA framework offers several methods. We opted for the sparse $\mathbf{LDL^T}$ factorization technique [81] to decompose matrix $\mathbf{A}$, where $\mathbf{D}$ represents a diagonal matrix and $\mathbf{L}$ denotes the sparse lower-triangular portion of matrix $\mathbf{A}$. Although this approach incurs considerable computational costs, it guarantees the reliability of the simulated mechanical behavior of the soft body, namely the tactile skin.

4.1.2 PDLC film modeling

The mechanical coupling between the PDLC film and the outer elastomer layer must be achieved while maintaining computational efficiency. Here, we simplify the PDLC film, which inherits characteristics from PET films, as a stiffening substrate that constrains the deformation of the outer elastomeric (soft) layer from its original representation. This contribution is modeled by incorporating *virtual elastic springs*, which connect all paired nodes of the soft layer's mechanical model individually, as depicted in Figure 4.2, where the soft layer is characterized by Young's modulus E and viscosity η, while the mechanical effects of the PDLC film are characterized by virtual springs with stiffness k_{vs}. This model operates under the assumption

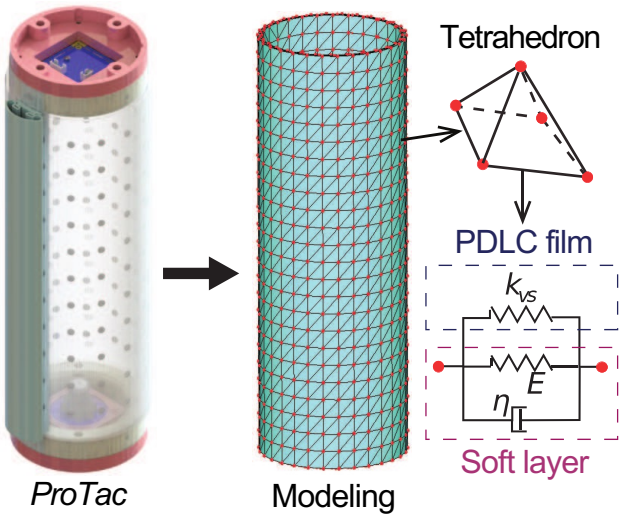


Figure 4.2: Modeling scheme of the two-layered *ProTac*'s skin.

that the PDLC film will not exceed its elastic limit point (*i.e.*, undergo plastic deformation). Any deviation from this assumption would lead to inaccuracies in the representation of soft multi-layered structures like the *ProTac* skin, resulting in the generation of unrealistic simulated tactile images. At an equilibrium deforming state t, the virtual springs generate internal forces $\mathbf{f}^{spring}$, which are proportional to the nodal displacement $\boldsymbol{\delta} \coloneqq \mathbf{q}(t) - \mathbf{q}(0)$, where $\mathbf{q}(t)$ and $\mathbf{q}(0)$ represent the current and rest positions, respectively. Consequently, the motion equation (4.3) for the entire lumped system of a soft multi-layered skin is updated accordingly.

$$\left(\mathbf{M}_{\boldsymbol{\Sigma}} + dt^2\bar{\mathbf{K}} + dt\mathbf{C}\right)\ddot{\mathbf{q}} = -dt^2\bar{\mathbf{K}}\dot{\mathbf{q}}_1 + dt\left(\mathbf{F}^{ext} - \bar{\mathbf{F}}^{int}\right) + dt\mathbf{J}^T\boldsymbol{\lambda} \tag{4.4}$$

where $\mathbf{M}_{\boldsymbol{\Sigma}}(\mathbf{q}) = \text{diag}[\cdots, \frac{M_s+M_f}{N}, \cdots] \in \mathbb{R}^{N\times N}$ with M_f is total mass of the PDLC film, $\bar{\mathbf{K}} = \frac{\partial \bar{\mathbf{F}}^{int}}{\partial \mathbf{q}}$ and $\bar{\mathbf{F}}^{int} = \mathbf{F}^{int} - \mathbf{f}^{spring}$, in which $\mathbf{f}^{spring} = k_{vs} \times \boldsymbol{\delta}$.

4.2 Simulated training data collection

In the SOFA framework, the mechanical representation of the soft skin consists of two distinct models: the bare skin and the markers, which are subsequently integrated within the simulation setup (see Figure 4.3). To manage multiple meshes efficiently, a discretization strategy is implemented: the mesh for the bare skin,

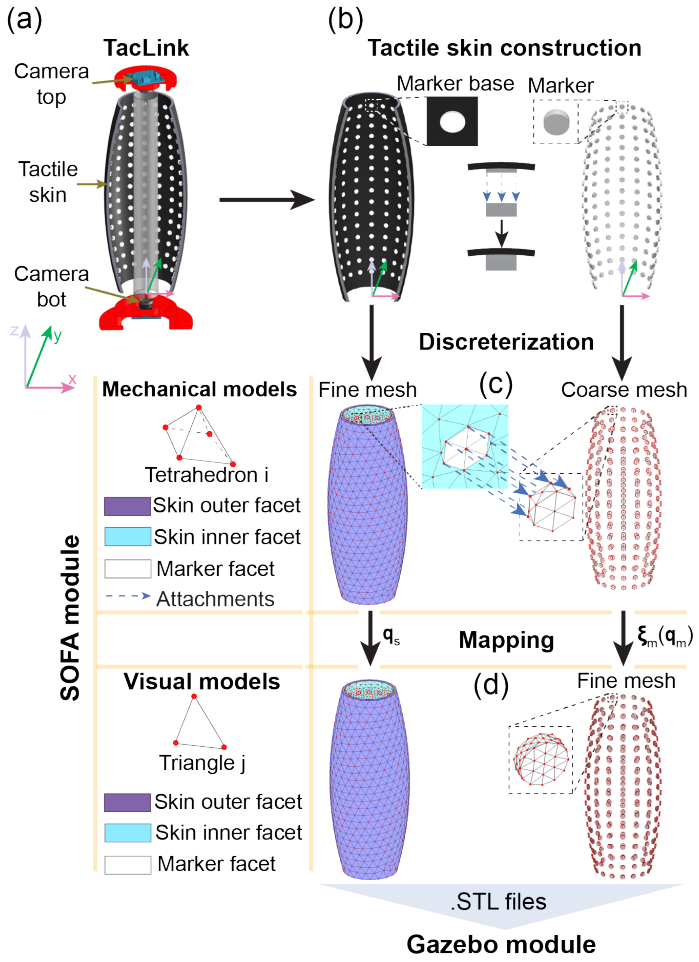


Figure 4.3: (a) Hardware architecture of a typical tactile skin. (b) cylinder markers attached to the tactile skin will be decomposed into two parts: marker bases and bodies. (c) Each tactile skin element will be imported to SOFA as a topological map of (c) tetrahedron elements for mechanical models and (d) triangular cells for visual models. Notice that, while the high-quality of the skin mesh remains in this mode, the meshes for markers in the visual model are refined significantly.

serving both mechanical analysis and visualization, employs a finer discretization (with a skin size element of 12 mm), while a coarser discretization (with a marker size element of 1.5 mm) is applied to reduce the computational complexity for the markers. Subsequently, the spatial coordinates of each degree of freedom in the visual models are synchronized with those in the mechanical models using a mapping function denoted as $\boldsymbol{\xi}_m$ before exporting the deformation states of the modeled skin to the Gazebo module for generating corresponding virtual tactile images.

4.2.1 Skin deformation labeling

We utilize the above soft skin model to generate a dataset capturing the reality-like skin deformation across the entire skin (*i.e.*, *global* skin deformation), denoted as $\{\mathbf{D}^{\text{FEM}}\}$. This dataset, along with corresponding virtual tactile images (see Section 4.2.2), enables the learning of a deep neural network for estimating the global skin deformation, as discussed in Section 4.3. To streamline computational resources, we define the initial shape of the skin in its undeformed state as $\mathbf{X}_0 := [\mathbf{X}_{0,i} \in \mathbb{R}^3 \mid \mathbf{X}_{0,i} = \mathbf{q}_i(0), \forall i \in \mathcal{N}]$, where $\mathcal{N}$ represents the node indices on the skin surface ($|\mathcal{N}| = N_o$). Upon physical contact, the skin undergoes deformation, transitioning the original state $\mathbf{X}_0$ to a new deformed state $\mathbf{X} \in \mathbb{R}^{N_o \times 3}$. This deformation adheres to the dynamics outlined in the preceding section. Consequently, the skin deformation $\mathbf{D}^{\text{FEM}} = [\mathbf{D}_i^{\text{FEM}} \in \mathbb{R}^3, \forall i \in \mathcal{N}]$ is delineated by the nodal displacement vectors:

$$\mathbf{D}_i^{\text{FEM}} \coloneqq \mathbf{X}_i - \mathbf{X}_{0,i} \ , \ \forall i \in \mathcal{N}. \tag{4.5}$$

4.2.2 Virtual tactile image acquisition

Unlike a previous study that synthesized virtual images based on the mathematical derivation of a pinhole model for wide-angle lens cameras [63], in this paper, the entire process for generation and acquisition of virtual (simulated) tactile images is performed using the combination of *Gazebo* simulator and Robot Operating System (ROS) [83]. *Gazebo* is preferable in this process as it supports the extension of an RGB camera with a fish-eye lens resembling that used in the TacLink device, and *Gazebo* can be integrated with ROS, which facilitates the use of this simulation sensing framework for high-level robot perception, planning and control. Since only virtual RGB cameras with fisheye lens extensions are used in our approach, other simulators that offer similar functions, such as Unity and PyBullet [2] can also be utlized for our platform.

In the *Gazebo* environment, TacLink sensor is modeled as a robotic link using

[2] https://pybullet.org/wordpress/

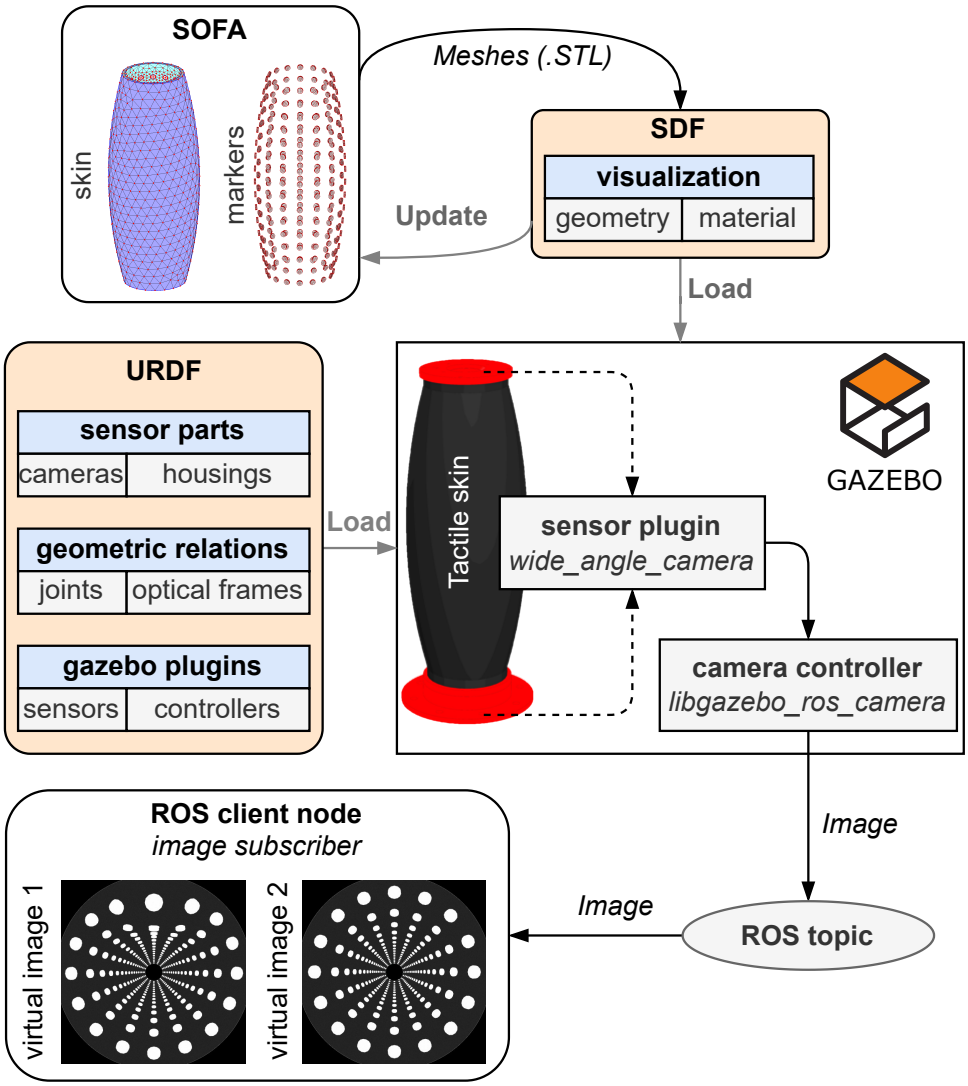


Figure 4.4: The workflow for the generation and acquisition of virtual tactile images is as follows: The *Gazebo* environment is set up according to the description of the TacLink sensor's URDF, including the relative camera positions and Gazebo plugins. Following this setup, the topological meshes of the skin and markers (.STL) are consecutively updated via SDFormat for image generation using the sensor plugin. The stream of image data published on the ROS topic can then be acquired and saved in the desired format for building the training dataset.

Unified Robot Description Format (URDF) [3], in which the geometric relations between sensor parts, such as housings and cameras, are defined precisely as the design of a real device. From the URDF description of TacLink sensor, *Gazebo* sensor plugin providing the camera type of `Wide Angle Camera Sensor` is installed to enable virtual cameras to render images of the artificial skin (tactile images). We also attempt to reproduce the image distortion effect caused by the projection through

[3]URDF is an XML format used by ROS to describe a robotics system.

a wide-angle fisheye lens to better describe the actual sensor behavior, which might improve the overall performance of sim-to-real transfer. To this end, the *Gazebo* sensor plugin is specified to accept a mapping function of a specific lens model. This specification enables automatic generation of the distortion effect in the virtual tactile images. Among the mapping functions proposed, the stereographic projection is considered to be suitable to our circular fisheye lens cameras, as described [84]. The stereographic mapping function of the lens is defined as $r = 2f \tan \frac{\theta}{2}$ [84]; where θ is the polar angle of a given point in the real world forming with the optical axis, r the radial position of such point on the image plane, and f the focal length of the lens; and this equation can be easily encoded and interfaced with the *Gazebo* plugin via URDF specification.

For every updated topological status of the meshes of the sensor's deformable skin and marker generated at each time step of the *SOFA* simulation, we utilize Simulation Description Format (SDF) [4] as a means to communicate them to *Gazebo* through the `gazebo/spawn_sdf_model` ROS service. An SDF file defines a detailed visualization of the artificial skin and embedded markers by specifying 1) *geometry* linked to the STL topological meshes for the realistic skin shape display; and 2) *material* which assigns colors to the skin and markers using a Blinn-Phong shading model [85]. The virtual cameras periodically capture images of the artificial skin, which are sequentially loaded into *Gazebo* environment via the sensor skin SDF specification, which facilitates generation of photorealistic tactile images. In addition, from the TacLink URDF, the `libgazebo_ros_camera` *Gazebo* plugin is enabled to establish communication between *Gazebo* and ROS via the `camera/image_raw` ROS topic, over which tactile images are published by *Gazebo* server under `Image` ROS messages. For image acquisition, a ROS client node is set up to subscribe the stream of tactile images which could then be processed and collected. The entire process of the generation and acquisition of virtual tactile images is encapsulated in Fig. 4.4. Detailed descriptions (XML file format) of the SDF and URDF including the virtual sensor specification (*e.g.*, optical frame, camera lens mapping function), sensor parts, and their geometric relations are included in the enclosed codebase.

[4] http://sdformat.org/

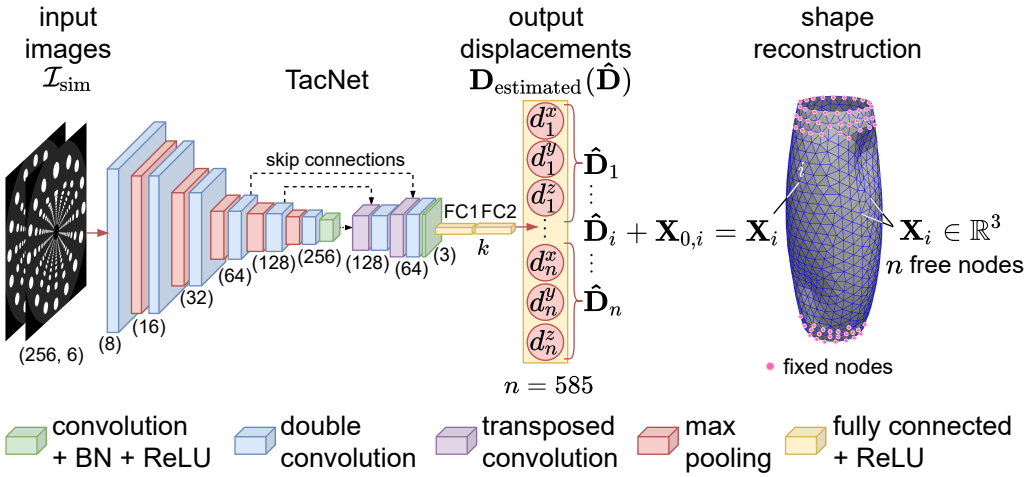


Figure 4.5: TacNet concept and architecture. It maps a pair of virtual tactile images $\mathcal{I}_{\text{sim}}$ to the displacements of free nodes $\mathbf{D}^{\text{est}}$ from which the deformation of artificial skin could be estimated. The soft skin is represented by a topological mesh consisting of fixed nodes (denoted by pink dots) and free ones (the other vertices of triangular cells).

4.3 TacNet-based skin deformation sensing

TacNet is designed to estimate the global deformation of soft skin under contacts by processing input tactile images. Existing methodologies, such as image processing techniques [12] and data-driven algorithms [75], typically translate tactile images into spatial changes among markers within a mesh representing deformable skin. In contrast to the conventional analytical approach, which operates at approximately 10 Hz [12], we opted for a deep learning strategy [75] to implement TacNet, targeting computational speeds of up to 100 Hz through GPU computing resources. Unlike prior methods that computed marker deviations for shape inference, our approach captures changes in artificial skin by monitoring the displacement of mesh nodes via visual cues of markers' movements imprinted on the tactile images. This approach eliminates the need for redesigning the network upon changes in marker design, aiming to facilitate transfer learning across tactile images with diverse marker distributions.

4.3.1 Problem description

Let us define a set of nodes $\mathcal{N} = \{1, \ldots, N\}$, $|\mathcal{N}| = N$ belonging to the mesh of the skin outer surface that comprises two subsets ($\mathcal{N} = \mathcal{B} \cup \mathcal{M}$): the fixed

nodes $\mathcal{B}$ are a collection of unmoved nodes under external stimuli within areas 20 mm away from two ends of the skin mesh which are deemed less interesting for tactile sensation; whereas the free/active nodes $\mathcal{M}$; the number of whose changes in position are observed to reconstruct the entire skin shape. Hence, this vision-based reconstruction problem can be formulated as a multi-output regression task (see Fig. 4.5): given input marker-featured tactile images $\mathcal{I}$, a network is designed to estimate the displacement vectors of each free node

$$\mathbf{D}_i^{\text{est}} \coloneqq \mathbf{X}_i - \mathbf{X}_{0,i} \ , \ \forall i \in \mathcal{M}, \tag{4.6}$$

where $\mathbf{X}_i \in \mathbb{R}^3$ is the 3-D position vector of one active/free node $i \in \mathcal{M}$, and $\mathbf{X}_{0,i} \in \mathbb{R}^3$ is the coordinates of the respective node under the original or non-deformed state of the artificial skin. Thus, from the estimated displacement vectors $\mathbf{D}^{\text{est}}$ and original nodal positions $\mathbf{X}_0$, the skin shape can be reconstructed as $\mathbf{X} = \mathbf{D}^{\text{est}} + \mathbf{X}_0$, with the positions of all fixed nodes always unchanged $\mathbf{X}_i = \mathbf{X}_{0,i} \ , \ \forall i \in \mathcal{B}$.

4.3.2 TacNet architecture

The architecture of TacNet derives its framework from the well-established Unet convolution networks [86]. Essentially, TacNet comprises a contracted convolutional pathway linked with an inverse up-convolutional pathway via skip connections, succeeded by two fully connected (FC) layers (refer to Fig. 4.5). The output signal, activated by the final two FC layers, is characterized by a dense single layer comprising $3n$ neurons, representing the estimated displacement vectors $\mathbf{D}^{\text{est}}$, where each set of 3 adjacent neurons corresponds to a displacement vector.

4.3.3 TacNet training and loss function

The training of TacNet is exclusively conducted on a simulated dataset, utilizing input data $\mathcal{I}_{\text{sim}}$ (images captured from simulated TacLink cameras) and corresponding output labels $\mathbf{D}^{\text{FEM}}$ (ground-truth displacement vectors), generated respectively within the *Gazebo*/ROS and SOFA environments (refer to Sections 4.2.2 and 4.2.1). The objective function employed is Mean Squared Error (MSE) loss, aiming to

minimize disparities between the ground-truth and estimated displacement vectors $(\mathbf{D}^{\text{FEM}}, \mathbf{D}^{\text{est}})$, thereby optimizing the weights of TacNet T_θ:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{MSE}}[\mathbf{D}^{\text{FEM}}, \text{T}_\theta(\mathcal{I}_{\text{sim}})], \tag{4.7}$$

where $\mathbf{D}^{\text{est}} = \mathcal{T}_\theta(\mathcal{I}_{\text{sim}})$ and $\mathcal{L}_{\text{MSE}}(\cdot)$ is MSE loss, given by:

$$\mathcal{L}_{\text{MSE}}(\mathbf{D}^{\text{FEM}}, \mathbf{D}^{\text{est}}) = \frac{1}{3n} \sum_{i \in \mathcal{N}} \sum_{j \in \{x,y,z\}} (d^j_{\text{FEM},i} - d^j_{\text{estimated},i})^2, \tag{4.8}$$

In (4.8), d^j_i , $\forall j \in \{x, y, z\}$ denote the components of displacement vector $\mathbf{D}_i$ at the respective skin node $i \in \mathcal{M}$ along the x, y, and z axes. Notably, the MSE loss encourages learning of both intensity and direction of displacement vectors by computing the difference in every vector component (or output neurons). For the optimization process (4.7), iterative Stochastic Gradient Descent (SGD) optimizer is employed, utilizing a learning rate of 0.015, which has been experimentally selected.

4.4 Sim-to-real transfer learning

4.4.1 Real-to-simulation generative network

The primary objective of the *real2sim* generative network (R2S-GN) lies in transfering real tactile images (I_{real}) into transformed images (I_{tf}) that closely resemble the visual domain of the simulation dataset (I_{sim}). These simulation-like images are then utilized as inputs for TacNet, ensuring the preservation of TacNet-based deformation sensing performance during real-world implementation (see Fig. 4.6). To achieve this goal, R2S-GN is trained in adversarial manner, functioning as a generator within a traditional GAN framework. In this setup, it competes against a discriminator to optimize its performance in the transformation task.

4.4.1.1 Problem description

Given simulation images $I_{\text{sim}} \in \mathcal{I}_{\text{sim}}$, and its real/actual counterparts $I_{\text{real}} \in \mathcal{I}_{\text{real}}$, the sim2real gap refers to the difference in simulation-trained TacNet model's

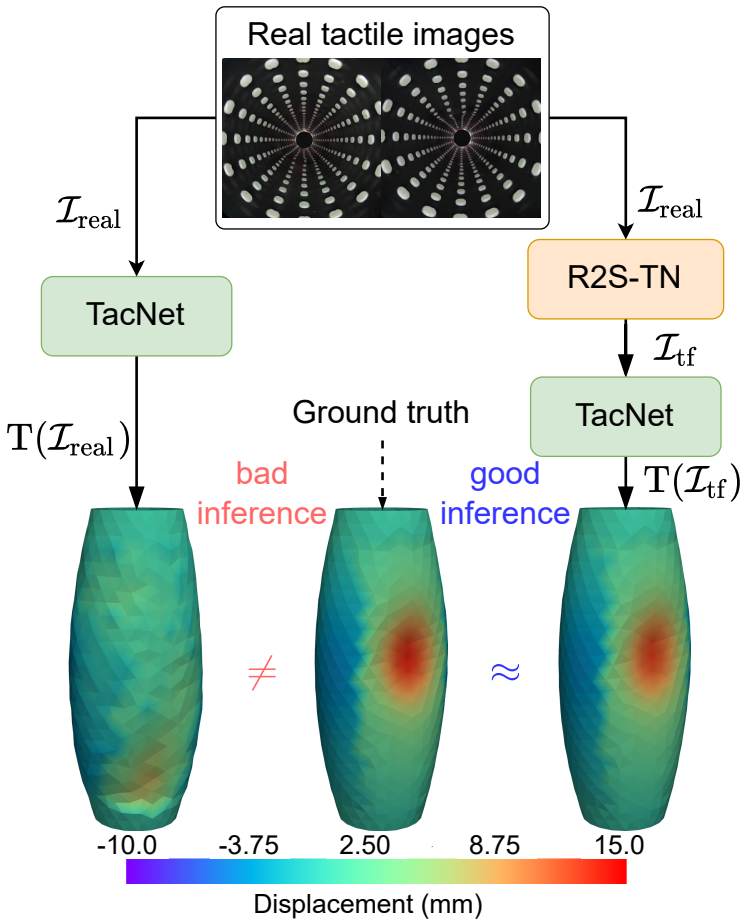


Figure 4.6: Tactile sensing sim2real problem and solution. Due to misalignment between real and simulation images, the performance of TacNet-based deformation sensing (T) is degraded (bad inference) as evaluated on real image samples ($\mathcal{I}_{\text{real}}$). R2S-GN tries to replicate as close simulation (virtual) images ($\mathcal{I}_{\text{tf}}$) as possible from the real ones in order to retain the TacNet performance (good inference) in the real data domain.

prediction $\mathrm{T}_\theta(\mathcal{I}_{\text{sim}}) \neq \mathrm{T}_\theta(\mathcal{I}_{\text{real}})$, that is caused by the discrepancies between simulation and real images $I_{\text{sim}} \neq I_{\text{real}}$ in terms of visual color and geometric perspective. R2S-GN, $\mathcal{G}_\phi$, learns a mapping from real images I_{real} to transformed ones $I_{\text{tf}} = \mathrm{G}_\phi(I_{\text{real}})$ such that TacNet performance could be preserved for the real-world dataset $\mathrm{T}(\mathcal{I}_{\text{sim}}) \approx \mathrm{T}(\mathcal{I}_{\text{tf}}) = \mathrm{T}(\mathrm{G}(\mathcal{I}_{\text{real}}))$. Toward this goal, the R2S-GN G_ϕ is trained in an adversarial manner to generate transformed images that cannot be distinguished from simulation ones, $I_{\text{tf}} \approx I_{\text{sim}}$, by competing against an adversarial trained discriminator, D_ψ, which on the other hand learns to do its best at discriminating the *real* simulation images I_{sim} with the *fake* transformed ones I_{tf}. Here, the *real* images refer to the image type that the R2S-GN generator tries to replicate, and should be distinguished from the actual images which indicate the

ones captured from the real-world sensing device.

4.4.1.2 Network architectures

We exploit the adapted version of U-Net convolutional network and PatchGAN model, as described in [87], for the architecture of R2S-GN generator (G_ϕ) and discriminator (D_ψ), respectively. The G_ϕ takes as input the downsampled real images (I_{real}, $256 \times 256 \times 3$) on a encoder path and outputs the transformed counterparts (I_{tf}) on a reverse decoder path. Meanwhile the discriminator (D_ψ) receives a $256 \times 256 \times 3$ pixel input image and the network classifies whether the images inputted is *real* or *fake*. Details of the network parameters for G_ϕ and D_ψ architectures can be found in [87].

4.4.1.3 R2S-GN Loss Function

We introduce a hybrid loss function $\mathcal{L}_{\text{R2S-GN}}$ for training the R2S-GN generative network (G_ϕ). This loss function consists of three components: the conditional generative adversarial network (cGAN) adversarial objective, ℓ_1 distance, and Structural Similarity Index (SSIM) loss.

Image appearance loss: Drawing inspiration from [88], we propose an appearance loss that combines ℓ_1 distance with the SSIM metric [89] for assessing image quality. This loss function, evaluated pixel by pixel, aims to align the appearance of the generated "fake" images I_{tf} with the "real" simulation images I_{sim} while preserving structural similarity. This alignment is crucial to ensure that the generated images maintain the same geometric characteristics as the simulation images, thus preserving the capabilities of the simulation-trained TacNet. Therefore, for a given batch of training samples, this loss is defined as:

$$\mathcal{L}_{\text{img}} = \alpha \left\| G_\phi(I_{\text{real}}) - I_{\text{sim}} \right\|_1 + \beta \frac{1 - \text{SSIM}(G_\phi(I_{\text{real}}), I_{\text{sim}})}{2}, \tag{4.9}$$

where we employ an 11×11 Gaussian kernel for SSIM computation.

Adversarial loss: Alongside the appearance loss, we incorporate the conditional Generative Adversarial Network (cGAN) objective [87] as an adversarial loss term.

For a given real tactile image I_{real}, the adversarial loss for the R2S-GN network Gϕ is represented as:

$$\mathcal{L}_{\text{adv}} = \log\Big(1 - \mathrm{D}_{\psi}\big(I_{\text{real}}, \mathrm{G}_{\phi}(I_{\text{real}})\big)\Big), \tag{4.10}$$

where D_{ψ} evaluates both the transformed tactile image $I_{\text{tf}} = \mathrm{G}_{\phi}(I_{\text{real}})$ and is conditioned on the input of G_{ϕ}, specifically I_{real}. This conditional discriminator has demonstrated enhanced performance in various image translation tasks [87], motivating its application in our real2sim network. Essentially, the R2S-GN G_{ϕ} aims to minimize this objective function by generating transformed images that deceive the adversarial discriminator D_{ψ} into classifying them as "real" simulation images. Consequently, the overall loss objective for R2S-GN G_{ϕ} combines the appearance loss with the conditional GAN criteria:

$$\mathcal{L}_{\text{R2S-GN}} = \underbrace{\mathcal{L}_{\text{img}}}_{\text{Appearance loss}} + \underbrace{\gamma \cdot \mathcal{L}_{\text{adv}}}_{\text{Adversarial loss}}, \tag{4.11}$$

where we set the hyperparameters $\alpha = 100$, $\beta = 200$, $\gamma = 1$, which are adjusted through empirical tuning.

Finally, to train the adversarial discriminator D_{ψ}, we employ the conditional Generative Adversarial Network (cGAN) objective outlined in [87]. For a single training sample, the discriminator loss is formulated as:

$$\mathcal{L}_{\text{G}} = \log\big(1 - \mathrm{D}_{\psi}(I_{\text{real}}, I_{\text{sim}})\big) + \log \mathrm{D}_{\psi}\big(I_{\text{real}}, \mathrm{G}_{\phi}(I_{\text{real}})\big). \tag{4.12}$$

The second term expresses the adversarial training behavior, where the discriminator aims to maximize the adversarial objective of the R2S-GN (Eq. 4.10), while the R2S-GN endeavors to minimize it. The comprehensive loss function (Eq. 4.12) indicates that the discriminator strives to effectively differentiate between the transformed images I_{tf} and the simulation ones I_{sim}, thereby penalizing the R2S-GN to generate I_{tf} that closely resemble the appearance of I_{sim}.

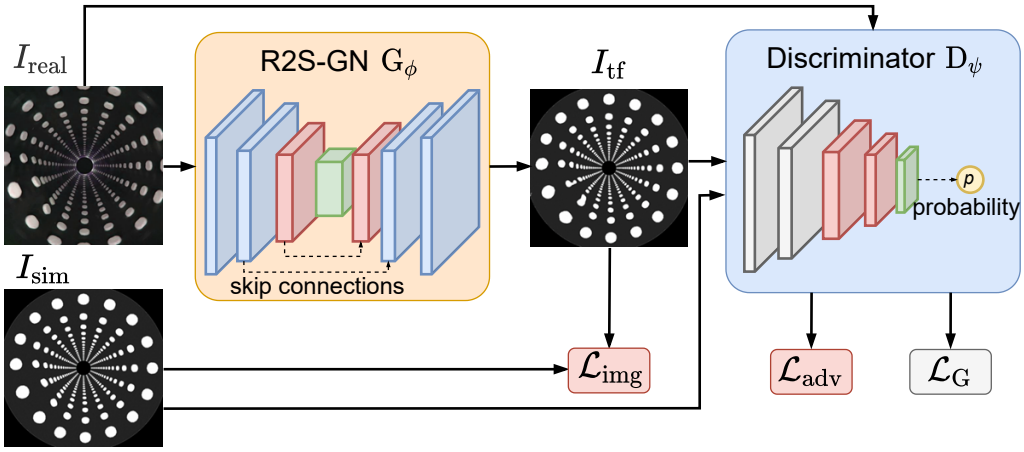


Figure 4.7: The training scheme for R2S-GN model, mostly following the procedure described in [19]; however with the modification for inclusion of R2S-GN loss.

4.4.1.4 R2S-GN Training

We adopt the typical adversarial Generative Adversarial Network (GAN) training procedure [19] to optimize the parameters of the R2S-GN G_ϕ network (see Fig. 4.7). Specifically, during discriminator training, we assign a positive class label (*real*) when the input is a simulation image, and a negative class label (*fake*) when the input is a transformed image. Meanwhile, for the R2S-GN network, alongside computing the $\mathcal{L}_{\text{img}}$ loss, we set the output label of D_ψ to the positive class (*real*) to facilitate the adversarial $\mathcal{L}_{\text{adv}}$ loss [87]. For optimization, we utilize the Adam optimizer with linear learning rate scheduling [90], initialized at 0.0002, and scheduled to decay at the 100^{th} iteration out of a total of 200 training steps.

4.4.2 Domain randomization

This section introduces a more straightforward approach to address the sim2real gaps of real and simulated tactile-image domains, in which the collection of real tactile images is not required. For this purpose, in the training phase of the TacNet model, we employ the domain randomization technique applied to the binary version of tactile images $\{\bar{\mathbf{I}}^{\text{tac}}\}_{\text{sim}}$, while the training procedure and loss function presented in Section 4.3 remains unchanged. The domain randomization involves performing affine transformations during the training process to diversify the perspective of tactile binary images, including translation, rotation, and scaling. This technique,

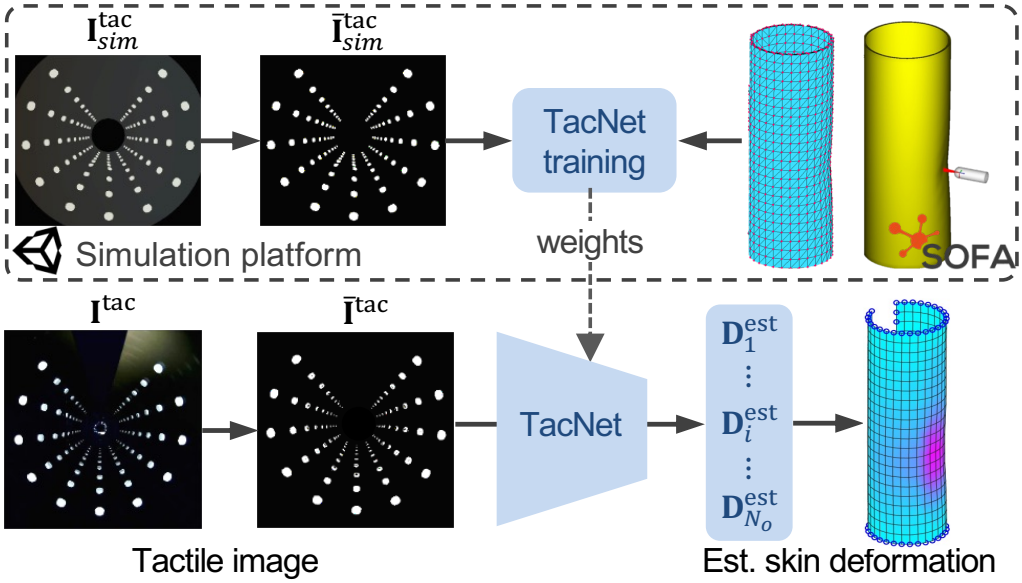


Figure 4.8: Illustration of tactile processing pipeline. TacNet model is trained using datasets comprising simulation tactile images and skin deformation states collected in the simulation platform (Unity-SOFA). To address the sim2real gap, the perspectives of simulation tactile images in binary format are randomized during the training process, facilitating the direct transfer of the TacNet model to real-world counterparts.

along with the high-fidelity physical modeling of the soft skin, facilitates zero-shot sim2real transfer, eliminating the need for real data or an additional network to mitigate the sim2real gap. The visual conceptualization of this approach is illustrated in Figure 4.8.

4.5 Large-area tactile perception

Large-scale vision-based tactile sensors offer opportunities for *multi-point* physical interactions, distinguishing them from their smaller counterparts like tactile fingertips. These sensors enable the extraction of various information from external stimuli, including the identification of contact intensities and contact locations on artificial skin surfaces. This capability holds significance in robotics applications, such as collision handling frameworks [27]. In addition to a contact event detection method (Section 4.5.1), we develop an algorithm for identifying multi-point contact intensities and locations across large-area skin surfaces (Section 4.5.2). This local contact information is inferred from the TacNet-based global deformation sensing.

4.5.1 Contact event detection

Detection of touch/contact events is fundamental in ensuring the safety of robotic systems [27]. Here, we introduce a method to extract signals indicating contact detection based on the prediction of global soft skin deformation.

The contact detection task can be framed as a binary classification problem, where given the displacement vectors $\mathbf{D}^{\text{est}}$ obtained from TacNet (4.5), we assign a *contact detection signal*. This signal is assigned a value of 0 for data indicating no contact and 1 for data indicating contact. Therefore, the contact detection signal can be expressed as:

$$\text{CD} = \begin{cases} 1, & \text{if } \exists i \, : \, \|\mathbf{D}_i^{\text{est}}\| \geq \epsilon_c \\ 0, & \text{otherwise.} \end{cases} . \tag{4.13}$$

In essence, a contact detection threshold ϵ_c is set on the estimated displacement magnitude of free skin nodes $\|\mathbf{D}_i^{\text{est}}\|$, $\forall i \in \mathcal{M}$, where ϵ_c is primarily determined by the accuracy of TacNet estimation, impacting detection sensitivity and accuracy. This threshold is calibrated to achieve a balance between precision and recall, which are standard metrics of a binary classifier's performance. We utilize the simulation dataset to establish the detection threshold, with the expectation of its effectiveness in real-world data.

4.5.2 Multi-point contact sensing and localization

Identifying the precise location of a physical contact on a robot body (*e.g.*, a link of a manipulator) is crucial for robot response [27]. Contact localization aims to determine the specific positions on the robot body where contacts occur. In our efforts to integrate the *ProTac* link for a robot arm, we devise an algorithm capable of identifying contact positions and their intensities at *multiple points* across the sensing link.

This identification approach operates under the assumption that any contacts between the sensor skin and external objects are point contacts, an assumption deemed reasonable in practical contexts [27]. The algorithm utilizes the principles of graph theory-based connected-component labeling [91] to isolate contact regions, a

process termed *contact region labeling* (CRL), from which the local contact positions are determined. Here, we conceptualize the mesh representation of the artificial skin as an undirected *graph*, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertices represent the mesh nodes ($|\mathcal{V}| = |\mathcal{N}| = N$) and encapsulate information regarding the displacement vectors estimated by TacNet ($\mathbf{D}^{\text{est}} \in \mathbb{R}^{3N}$). Furthermore, each graph node contains data regarding a fixed radial vector pointing toward the central axis of the skin to ascertain nodes deflected inward. Therefore, the radial vector at each node is defined as follows:

$$\mathbf{N}_i \coloneqq \begin{bmatrix} 0 & 0 & x_{0,i}^z \end{bmatrix}^\top - \mathbf{X}_{0,i}^\top \,, \; \forall i \in \mathcal{N}, \tag{4.14}$$

where x_0^z represents the z-component of nodal positions $\mathbf{X}_0$ in the undeformed state.

To execute CRL for extracting distinct contact regions, the first step to identify which nodes of the skin are likely to be experiencing contact. Thus, we introduce an N-tuple of binary *nodal contact signals* $\mathbf{s} = (s_1, \ldots, s_N) \in \mathbb{Z}_2^N$, where each s_i takes a binary value $s_i \in \{0, 1\}$ such that $s_i = 1$ indicates that node $i \in \mathcal{N}$ is in contact and is part of a contact region, while $s_i = 0$ signifies that the node remains intact. Specifically, the nodal contact signal for each node $i \in \mathcal{N}$ is computed as:

$$s_i = \begin{cases} 1, & \text{if } \|\mathbf{D}_i^{\text{est}}\| \geq \epsilon_d \wedge d_{\text{sim}}(\mathbf{D}_i^{\text{est}}, \mathbf{N}_i) > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{4.15}$$

where,

$$d_{\text{sim}}(\mathbf{D}_i^{\text{est}}, \mathbf{N}_i) = \frac{\mathbf{D}_i^{\text{est}} \cdot \mathbf{N}_i}{\|\mathbf{D}_i^{\text{est}}\| \, \|\mathbf{N}_i\|}. \tag{4.16}$$

In other words, a node is deemed to belong to a contact region if its nodal displacement exceeds a predetermined threshold ϵ_d and the direction of the displacement vector points towards the skin's central axis. The latter condition, that is $d_{\text{sim}}(\cdot) > 0$, helps confine contact regions to contain nodes deflecting inwards. Here, $d_{\text{sim}}(\cdot) \in [-1, 1]$ computes directional similarity (Eq. 4.16), quantifying $\cos \varphi_i$, where φ_i denotes the angle between the vectors $\mathbf{D}_i^{\text{est}}$ and $\mathbf{N}_i$.

Given the skin graph $\mathcal{G}$ and nodal contact signals $\mathbf{s}$, the CRL procedure is conducted to extract potential *multiple* distinct contact regions (refer to Algorithm

1: `CRL` function). This algorithm employs depth-first search (DFS) to traverse the vertices $\mathcal{V}$ of graph $\mathcal{G}$ containing the nodal information of $\mathbf{s}$. Along the search path, it assigns a *contact region label* $l \in \{1, \ldots, L\}$ (L denotes the number of contact regions) to each node with the signal $s_i = 1$, thereby labeling clusters of contacted nodes (or contact regions) separated by un-deformed nodes ($s_i = 0$) with the same region label l. Consequently, a set of labels $\mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{Z}_{L+1}^N$ is obtained, where each $y_i \in \{0, 1, \ldots, L\}$ represents the region label of node $i \in \mathcal{N}$, and $y_i = 0$ denotes nodes within the undeformed region. From $\mathbf{y}$, contact regions can be extracted using the node indexes. Hence, for a given contact region $\boldsymbol{R}_l$ with the region label l:

$$\boldsymbol{R}_l = \{i \in \mathcal{N} \mid y_i = l\}, \forall l \in \{1, \ldots, L\}. \tag{4.17}$$

Finally, within a contact region $\boldsymbol{R}_l$, the node $i_l^* \in \boldsymbol{R}_l$ at which the displacement magnitude is maximum is identified as the contact location:

$$i_l^* = \arg\max_{i \in \boldsymbol{R}_l} \left\| \mathbf{D}_i^{\text{est}} \right\|, \forall l \in \{1, \ldots, L\}. \tag{4.18}$$

From that, contact positions $\{\hat{\boldsymbol{x}}_1^c, \hat{\boldsymbol{x}}_2^c, \ldots, \hat{\boldsymbol{x}}_L^c\} \in \mathbb{R}^{3L}$ can be identified from the extracted contact regions, which are assumed to be the positions at which the skin is deformed most, or at the positions with the largest contact intensities:

$$\hat{\boldsymbol{x}}_l^c := \mathbf{X}_{0,i_l^*}, \forall l \in \{1, \ldots, L\}, \tag{4.19}$$

Also, the vectors of contact intensities $\{\hat{\boldsymbol{d}}_l^c\}$ at corresponding contact locations $\{\hat{\boldsymbol{x}}_l^c\}$ can be derived as:

$$\hat{\boldsymbol{d}}_l^c = \mathbf{D}_{i_l^*}^{\text{est}}, \forall l \in \{1, \ldots, L\}. \tag{4.20}$$

Thus, the magnitude of the contact intensity vector is referred to as the contact depth, that is $\|\hat{\boldsymbol{d}}_l^c\|$. For brevity, we utilize $\|\hat{\boldsymbol{d}}^c\|$ and $\hat{\boldsymbol{x}}^c$ to denote a single-point contact depth and location without the subscript index throughout the paper.

The sequential procedure for multi-point contact sensing is detailed in Algorithm 1, with its complexity primarily depending on the size of the skin graph, denoted by

Algorithm 1 Multiple-point Contact Localization

Input: $\mathcal{G}$: skin graph defined by $(\mathcal{V}, \mathcal{E})$; $\mathbf{X}_0$: initial nodal positions; $\mathbf{D}^{\text{est}}$: estimated nodal displacement vectors; $\mathbf{N}$: nodal radial inward vectors
Output: $\mathbf{X}_c$: multiple contact positions $(\mathbf{x}_1^c, \ldots, \mathbf{x}_L^c)$

Initialize: ϵ_c ▷ contact threshold
$\mathbf{s} \leftarrow$ new `List`
for each node v_i in $\mathcal{V}$ **do**
    if $\|\mathbf{D}_i^{\text{est}}\| \geq \epsilon_c$ and $d_{\text{sim}}(\mathbf{D}_i^{\text{est}}, \mathbf{N}_i) > 0$ **then** $s_i \leftarrow 1$ ▷ assign nodal contact signals (4.15)
    else $s_i \leftarrow 0$
    end if
end for
$\mathbf{y} \leftarrow \texttt{CRL}(\mathcal{V}, \mathcal{E}, \mathbf{s})$ ▷ obtain a list of contact region labels
$(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_L) \leftarrow \texttt{sortContactRegions}(\mathbf{y})$ ▷ see (4.17)
$(i_1^*, \ldots, i_L^*) \leftarrow \texttt{searchContactNodes}(\boldsymbol{R}, \mathbf{D}^{\text{est}})$ ▷ (see 4.18)
$\mathbf{x}_l^c \leftarrow \mathbf{X}_{0,i_l^*}$ for all $l \in \{1, \ldots, L\}$
return list of contact positions $(\mathbf{x}_1^c, \ldots, \mathbf{x}_L^c)$
function `CRL`($\mathcal{V}$, $\mathcal{E}$, $\mathbf{s}$) ▷ contact region labeling
    $l \leftarrow 1$ ▷ initialize contact region label
    $\mathbf{y} \leftarrow$ new `List`
    for each node v_i in $\mathcal{V}$ **do**
        if $s_i = 1$ and $y_i = \varnothing$ **then**
            $\mathbf{y} \leftarrow \texttt{DFS}(l, v_i, \mathcal{E}, \mathbf{s}, \mathbf{y})$
            $l \leftarrow l + 1$
        else if $s_i = 0$ **then** $y_i \leftarrow 0$
        end if
    end for
    return list of contact labels $\mathbf{y}$
end function
function `DFS`(l, v_i, $\mathcal{E}$, $\mathbf{s}$, $\mathbf{y}$) ▷ depth first search
    if $s_i = 0$ or y_i not $\varnothing$ **then return** $\mathbf{y}$
    end if
    $y_i \leftarrow l$
    for each neighbour node v_j of v_i in $\mathcal{E}(v_i)$ **do**
        $\mathbf{y} \leftarrow \texttt{DFS}(l, v_j, \mathcal{E}, \mathbf{s}, \mathbf{y})$
    end for
end function

$\mathcal{O}(|\mathcal{V}|+|\mathcal{E}|)$. Furthermore, the spatial resolution is determined by the fineness of the constructed skin mesh, introducing a trade-off between resolution and computational overhead; heightened resolution increases computational demands. Additionally, the assumption of point contact can be relaxed in scenarios where contacts result in concave deformations of the skin surface, leading the detector to approximate the

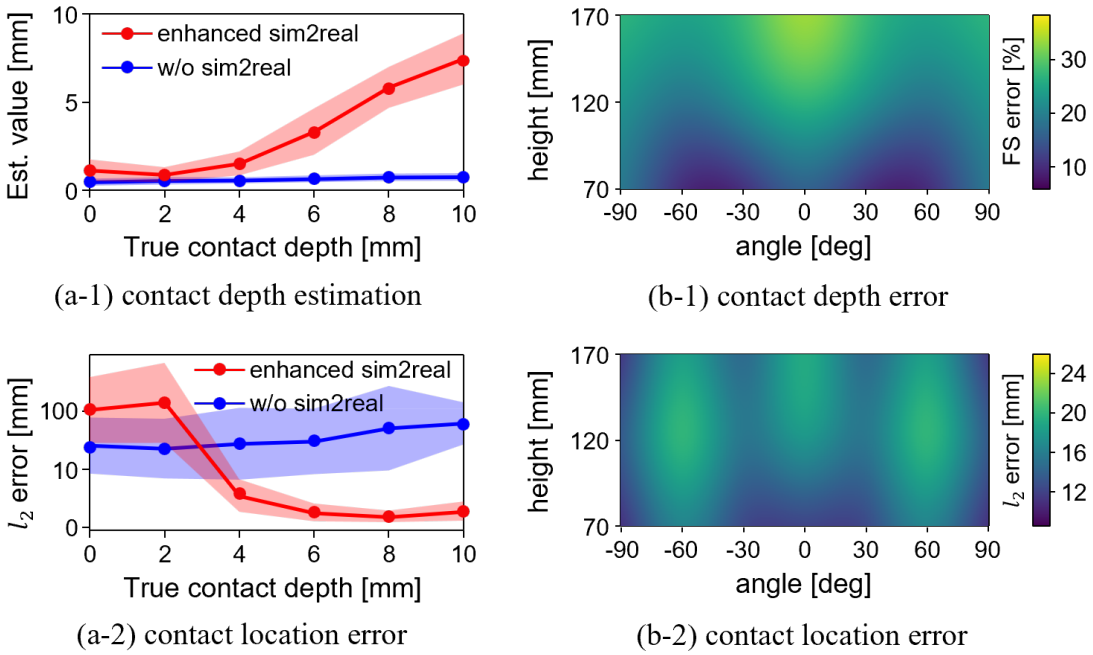
contact position at the most deeply displaced node; however, detection precision diminishes as the contact plane expands. Finally, instances of overlapping contact regions, such as when two discrete contact points are close to each other, may lead the detector to perceive distinct regions as a singular large contact area. This sensing behavior, influenced by the distance between contact points and the selection of threshold ϵ_d, as well as its impact on localization accuracy, is discussed in Section 4.7.

4.6 Performance evaluation: ProTac link

To assess the effectiveness of the SimTacLS framework, tactile perception experiments were carried out for the *ProTac* link in **tactile** mode and a barrel-shaped tactile link, TacLink (this section, and Section 4.7, respectively). Additionally, for sim2real learning, we validated the domain randomization and R2S-GN adversarial domain adaptation techniques with the *ProTac* link and TacLink, respectively. Model training and inference were performed on a desktop PC equipped with an AMD Ryzen™ Threadripper™ 3970X Processor, utilizing GPU acceleration (RTX 8000, NVIDIA). In this study, we utilized the Unet-based TacNet configuration ($k = 2048$) as a means for skin deformation sensing. The rationale behind selecting this TacNet configuration and its performance compared to various TacNet models are thoroughly discussed in Appendix B

4.6.1 Setups

The TacNet model was trained using 80% of the simulation dataset, comprising a total of 11025 data pairs. The remaining 20% was reserved for validation during the training process. Here, the quantity of output neurons is 1863, three times the number of mesh nodes representing the skin surface (*i.e.*, $N_o = 621$). For the evaluation, we collected a set of *unseen* real images by pressing an obstacle into the *ProTac* skin in its opaque state. The capability of *ProTac* to estimate contact depth and identify contact location was tested across various positions on the *ProTac* skin, defined by 2D cylindrical coordinates spanning a height range of $[70, 170]$ mm and an angular range of $[-90, 90]^\circ$. At each contact position, data was captured as



(a-1) contact depth estimation (b-1) contact depth error

(a-2) contact location error (b-2) contact location error

(a) Est. contact depth accuracy (b) Est. depth accuracy on the whole skin

Figure 4.9: *ProTac*'s **tactile** mode evaluation. The results highlight the contact sensing capability of *ProTac* link, characterized by contact depth estimation and contact localization over the entire skin area (a)-(b)

the obstacle's penetration increased incrementally from 0 to 10 mm, with intervals of 2 mm. Notably, by adopting the domain randomization technique (described in Section 4.4.2), later referred to as enhanced sim2real, the tactile sensing data were acquired by TacNet trained solely on the simulation dataset without the need for fine-tuning using real data, underscoring the zero-shot learning capability of our approach.

4.6.2 Results

Figure 4.9a depicts the evaluation of contact depth estimation and the associated detection error in contact location (assessed via the l_2-norm) against increasing true contact depth (indentation). The findings reveal that *ProTac* consistently provides responsive sensing signals when the indentation surpasses 4 mm within the 10 mm range. Specifically, contact depth estimation exhibits a linear correlation (Fig. 4.9a-1), while the detection error in contact location remains below 5 mm (Fig. 4.9a-2).

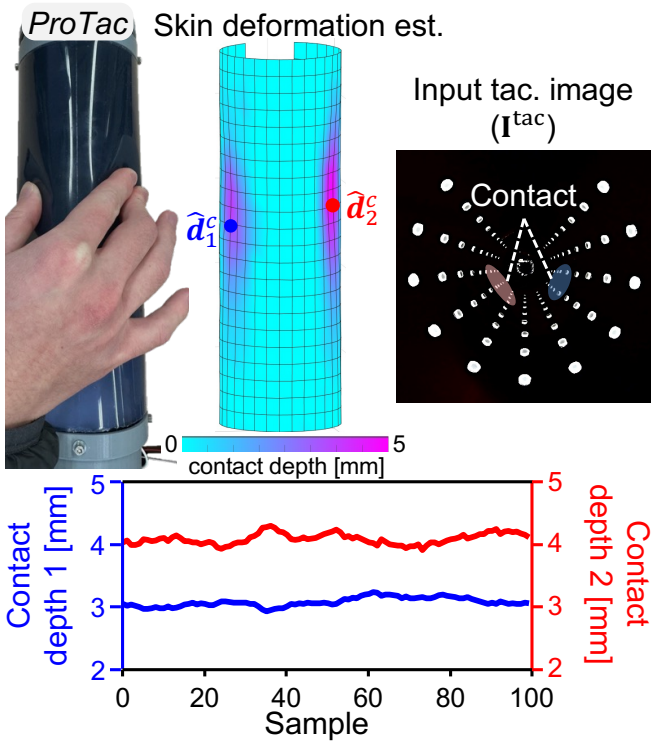


Figure 4.10: Demonstration *ProTac*'s ability to identify multi-point contact.

Furthermore, the results demonstrate that the incorporation of data augmentation via randomizing input image perspectives significantly enhances the sim2real transfer, resulting in significant improvements in tactile sensing performance.

Figure 4.9b illustrates the detection errors in contact depth and location across the entire large-area skin. Discrepancies in sensing accuracy across contact regions are attributed to the complex structures of the soft skin, which are not fully captured by the proposed skin model. Nevertheless, the observed accuracy levels are believed to remain satisfactory for large-area whole-arm tactile sensing and broader robotics applications [92].

Lastly, Figure 4.10 showcases a scenario involving two-point contact detection, where the estimation of skin deformation $\mathbf{D}^{\text{est}}$ facilitates the determination of contact depths $\{\hat{\boldsymbol{d}}_1^c, \hat{\boldsymbol{d}}_2^c\}$ at distinct contact locations, given the input tactile image $\mathbf{I}^{\text{tac}}$.

4.7 Performance evaluation: Barrel-shaped tactile link

This section validates the effectiveness of the proposed learning platform for the tactile perception of a large-area barrel-shaped tactile link (TacLink). In addition, we justify the efficiency of the R2S-GN adversarial domain adaptation technique (described in Section 4.4.1) in reducing the sim2real tactile sensing gap on the real sensing device.

4.7.1 Setups

We collected multiple datasets from both simulated and real-world domains to train and evaluate the feasibility of SimTacLS for the TacLink device. Details are provided in Table 4.1. Indentation locations were specifically designated among free nodes $\mathcal{M}$, resulting in a total of 585 sampled points for the "single-contact" dataset. To enable two-point contact sensing capability and evaluate the generalization of the adversarial sim2real learning technique with limited prior knowledge, we generated 500 contact pairs comprising two arbitrary points within $\mathcal{M}$ to form the "double-contact" dataset.

Table 4.1: Datasets used for model training and evaluation

Subject	**Tactile data**	**Simulation**	**Real**
TacNet	Images + Info	single + double	–
R2S-GN	Images	single	single
Evaluation	Images + Info	single + double	single + double*

*This dataset only contains some special scenarios used to evaluate SimTacLS and multi-contact localization accuracy.

An experimental setup was established for real tactile image collection (see Fig. 4.11). This setup included three motorized linear stages (Suruga Seiki Co., Japan), a rotating motor (Dynamixel XH430-W350-R, ROBOTIS, Inc., USA), and a stepping motor controller (DS102, Suruga Seiki Co., Ltd., Japan), all mounted on a testbed. The X-axis stage (PG750-L05AG-UA) operated a spherical-head indentor (12 mm diameter) designed to apply pressure to individual nodes, adjusting them to the desired contact depth on the skin. Movement of the indentor across the skin's outer surface and rotation of the TacLink sensor were achieved through horizontal

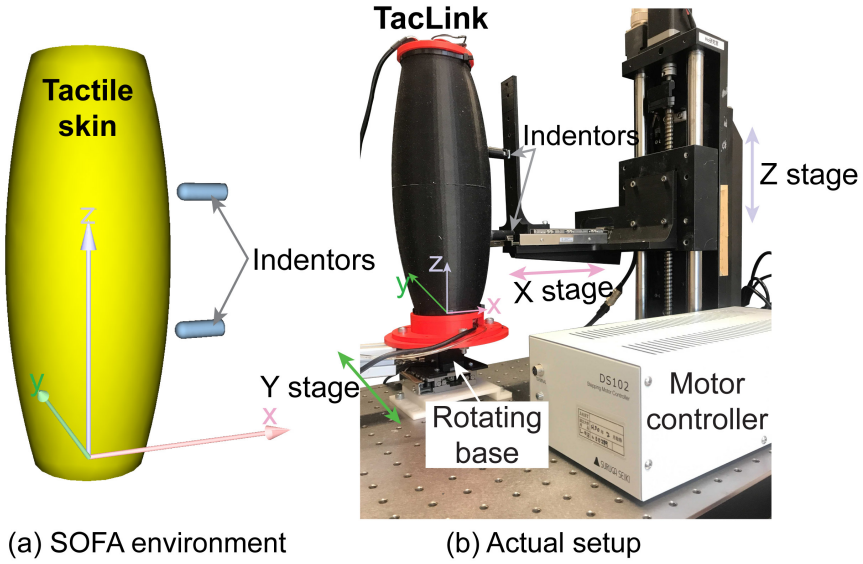


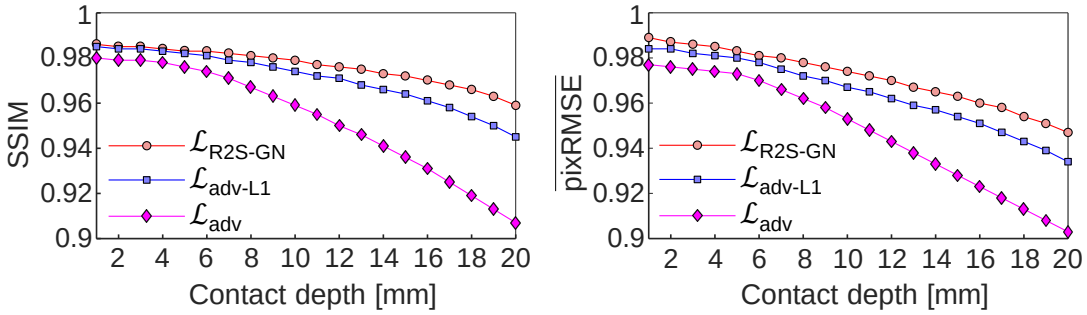
Figure 4.11: Setups for data collection in (a) simulation and (b) real-world.

motion facilitated by a Z-axis linear carrier (KZS18300) and rotation of the Z-axis motor, respectively. Meanwhile, the Y-axis stage was pre-positioned to ensure alignment of the indentor's nominal axis with the Z-axis of the reference coordinate system (*i.e.*, the centerline of the TacLink sensor).

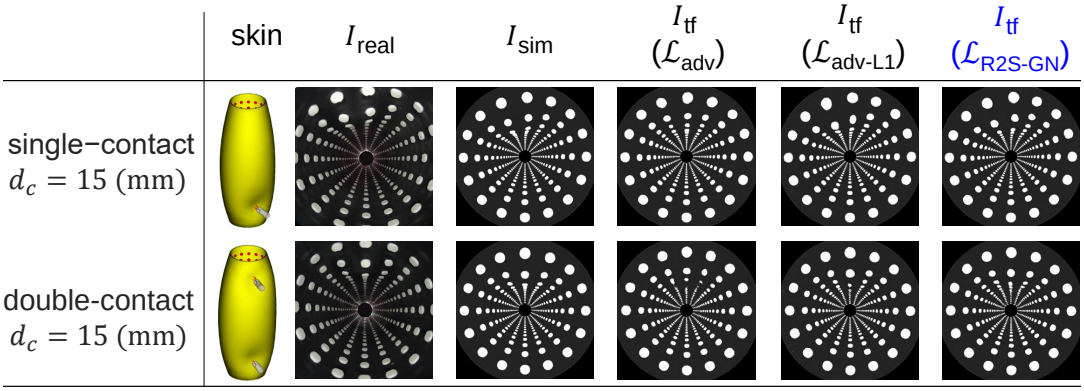
4.7.2 Image transformation evaluation with R2S-GN loss

The evaluation of R2S-GN performance centered on assessing the similarity between transformed and simulation images through spatial image structure. This involved measuring the structural similarity index (SSIM) and the complement of per-pixel root mean square error ($\overline{\text{pixRMSE}} = 1 - \text{pixRMSE}$) across pairs of transformed and baseline simulation images. Furthermore, we compared the performance of R2S-GN models trained using different loss functions, including the R2S-GN training loss $\mathcal{L}_{\text{R2S-GN}}$ (Eq. 4.11), solely adversarial loss $\mathcal{L}_{\text{adv}}$, and adversarial loss combined with ℓ_1-distance loss $\mathcal{L}_{\text{adv-L1}} := \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{L1}}$. The training involved 18640 pairs of single-contact actual-simulation images, while the evaluation was conducted with 4780 pairs of both single (4660 pairs) and double contacts (120 pairs).

Figure 4.12a depicts the structural similarity between tested simulation images and real images transformed by three R2S-GN network variants, each trained with a different loss function. It illustrates the evolution of sim-real similarity with



(a) Quantitative simulation-transformed image similarity measured by SSIM and complement of pixRMSE metrics

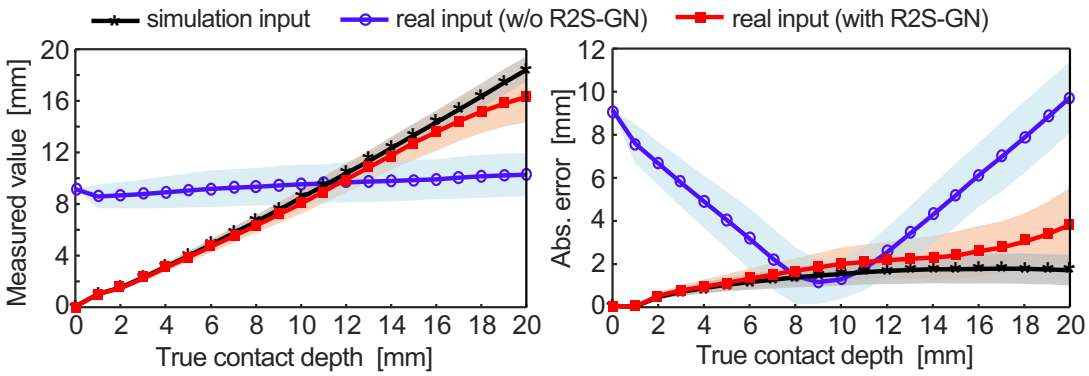


(b) Representative transformed images (I_{tf}) with variant training objectives comparing to the corresponding ground-truth simulation (I_{sim}) and real images (I_{real})

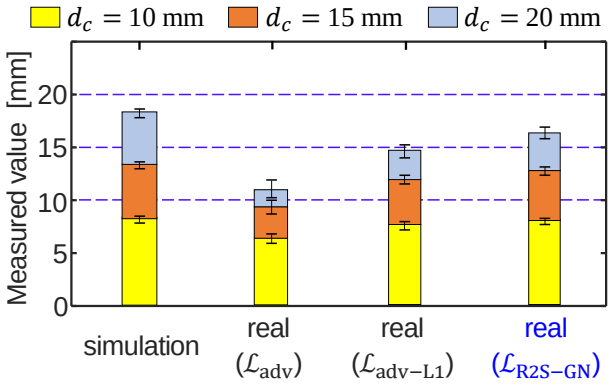
Figure 4.12: R2S-GN model evaluation with various training losses. (a) The spatial similarity between the transformed and *real* simulation images are measured by per-pixel SSIM and $\overline{\text{pixRMSE}}$ metrics (the higher the values the more similarity between the compared pairs of images). The graphs present the better performance of R2S-GN as trained with the proposed $\mathcal{L}_{\text{R2S-GN}}$ loss compared to the other two variants of losses. (b) Visualization of transformed images in the scenarios of single- and double-contact ($d_c = 15\,\text{mm}$).

increasing contact depth. Notably, the $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN network yields images more akin to the simulation baselines, boasting an average SSIM of 0.96 and an average $\overline{\text{pixRMSE}}$ of 0.95 at 20 mm contact depth, compared to 0.91 and 0.90 of the $\mathcal{L}_{\text{adv}}$-based model. While the former experiences a marginal drop of around 3.5 % in both SSIM and $\overline{\text{pixRMSE}}$ metrics across the observed range of contact depth $d_c \in [1, 20]$, the latter endures a more significant (7 %) decline in structural similarity.

For qualitative visualization of the similarity, Figure 4.12b showcases sample single- and double-contact tactile images with a 15 mm contact depth, demonstrating



(a) Estimated contact depth and absolute measurement error versus true value. ($\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN was used)



(b) Estimated contact depths with various input images compared against true values of 10, 15, 20 mm.

Figure 4.13: Evaluation of contact depth accuracy and its sim2real transferability, using the proposed sim2real method.

the R2S-GN's ability to generalize and generate unseen tactile images effectively, even in scenarios not encountered during training. This confirms the efficacy of the proposed R2S-GN loss. In the subsequent subsection, we evaluate the efficacy of variant RS2-TN models in addressing the sim2real gaps.

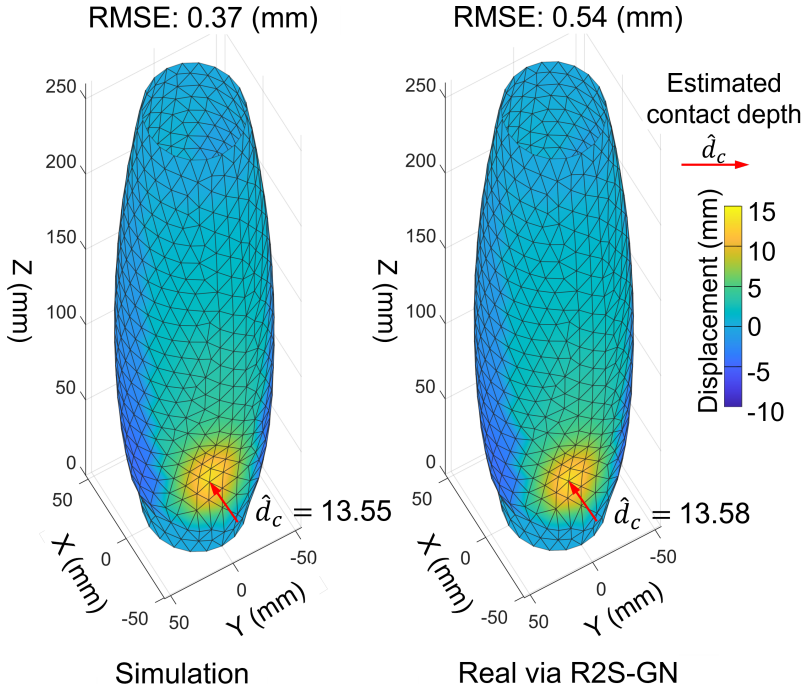
4.7.3 Evaluation of contact depth accuracy

While it is impractical to directly verify the accuracy of global skin deformation estimated by the TacNet model, the performance of the contact sensing can be accessed by the measurement error of local contact depths $\|\hat{\boldsymbol{d}}^c\|$. We evaluated the sensing accuracy on both simulated and real datasets to justify the effectiveness of

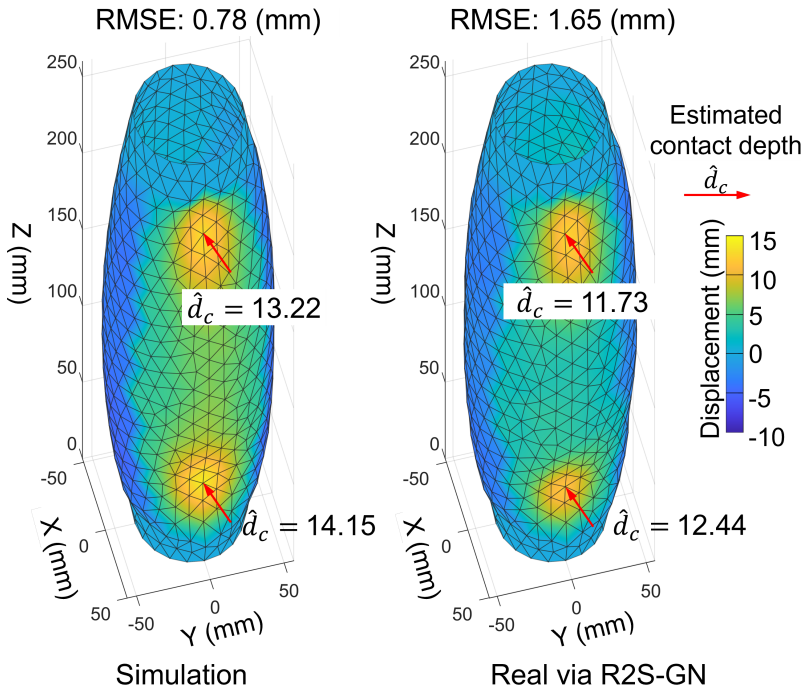
the adversarial sim2real learning technique. Here, TacNet was trained entirely on the simulation dataset, comprising both single- and double-contact images (a total of 28055 pairs of virtual tactile images), with 20% of each contact type data held out as a test fold for validation. To assess sim2real transferability, we conducted experiments on a subset of real double-contact images and a complete set of real single-contact images corresponding to the simulation test fold (refer to Table 4.1).

The experimental findings revealed an increase in measurement errors with true contact depth (d_c) in both simulation and $\mathcal{L}_{\text{R2S-GN}}$-based translated visual inputs, while pure real inputs, bypassing the R2S-GN model, exhibited significant errors, maintaining unchanged estimated values (refer to Figs. 4.13a). At $d_c = 20\,\text{mm}$, the absolute errors were below 2 mm and 4 mm, approximately corresponding to full-scale errors of 10% and 20% (with FS 20 mm) for simulation and translated inputs, respectively. Moreover, Figure 4.13b demonstrated that the $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN model outperformed the other two variants trained by $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{adv-L1}}$, reducing full-scale errors by approximately 25% and 10%, respectively, at $d_c = 20\,\text{mm}$. Additionally, we present the skin shape reconstruction visualization for two representative scenarios of single- and double-point contact with a depth of $d_c = 15\,\text{mm}$ (see Fig. 4.14). A similar sensing pattern between simulation and real (via $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN) samples was observed for single-contact, with an absolute error of approximately 1.5 mm. In the case of double-contact, the mean absolute errors at the two contact patches were $1.31 \pm 0.65\,mm$ and $2.92 \pm 0.50\,\text{mm}$ for the virtual and translated real input, respectively. The discrepancy between simulation and real data was more pronounced in double-contact scenarios because the R2S-GN had not been trained on double-touch data, likely resulting in greater dissimilarity in image structure, especially at large contact depths.

Importantly, the accuracy of contact depth estimation varies across different regions of the skin. To investigate this, we conducted an experiment where contact was initiated at ten locations along a longitudinal line of the skin, each with contact indentations of 5 mm and 10 mm. The obtained data, comparing contact depth values inferred from real images processed through the $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN, and the ground truths are illustrated in Fig. 4.15. The results indicate decreased sensitivity



(a) Skin reconstruction with single contact ($d_c = 15\,\text{mm}$)



(b) Skin reconstruction with double contact ($d_c = 15\,\text{mm}$)

Figure 4.14: The visualization of TacNet-based 3D skin shape reconstruction in the scenarios of single- and double contacts with true contact depth at 15 mm.

in the equatorial area of the skin. Nevertheless, the accuracy reported in this study can be enhanced through deliberate calibration, where parameters are identified differently for distinct contact regions. Notably, while contact depth accuracy varies

Height of test nodes along TacLink's vertical axis

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
h (mm)	239	219	189	168	138	118	96	65	48	26

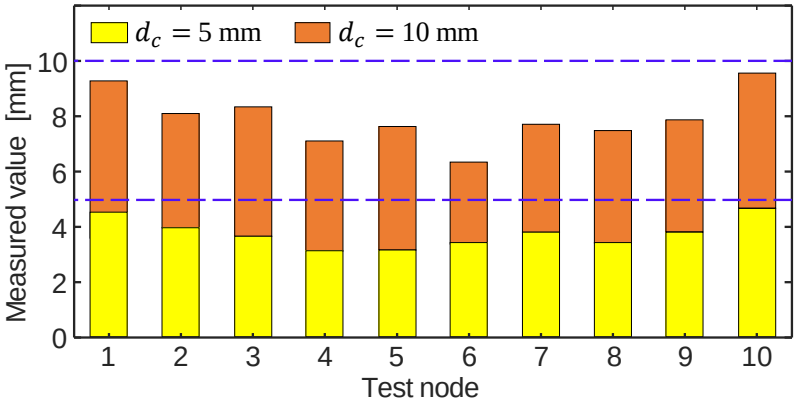


Figure 4.15: Evaluation of contact depth estimation at different contact regions on the tactile skin. (Estimated on real images via $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN)

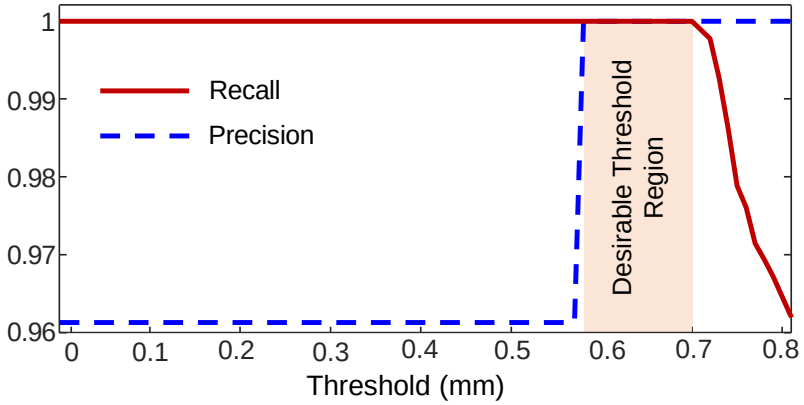
across skin regions, this issue is not witnessed with the contact localization task, as demonstrated in Section 4.7.5, where the variations of localization errors are insignificant across the entire sensing skin.

Overall, in the context of sim2real learning of soft vision-based tactile sensing on large body, the obtained results in this study, as far as we are aware, can establish a benchmark for future advancements. The sensing errors fall within an acceptable range when compared to prior studies [12, 75].

4.7.4 Evaluation of contact event detection

This section evaluates the performance of contact event detection using our proposed method with domain adaptation via the R2S-GN model. Initially, we identified an optimal contact detection threshold ϵ_c^* that maximizes detection capability using the simulation image dataset. The selection process was based on analyzing the precision-recall trade-off [90] of the touch classifier over a finite range of decision thresholds ϵ_c. As shown in the precision/recall plot (see Fig. 4.16a), we chose a contact threshold value of 0.6 mm for evaluation since it maximized contact sensing performance with 100% recall and precision.

The accuracy of contact detection, evaluated using the test simulation dataset



(a) Precision/Recall trade-off of the touch classifier evaluated on the virtual images.

Input type	Precision	Recall	Accuracy
sim (virtual)	1.00	1.00	1.00
real (w/o R2S-GN)	0.95	1.00	0.95
real (via R2S-GN)	1.00	1.00	1.00

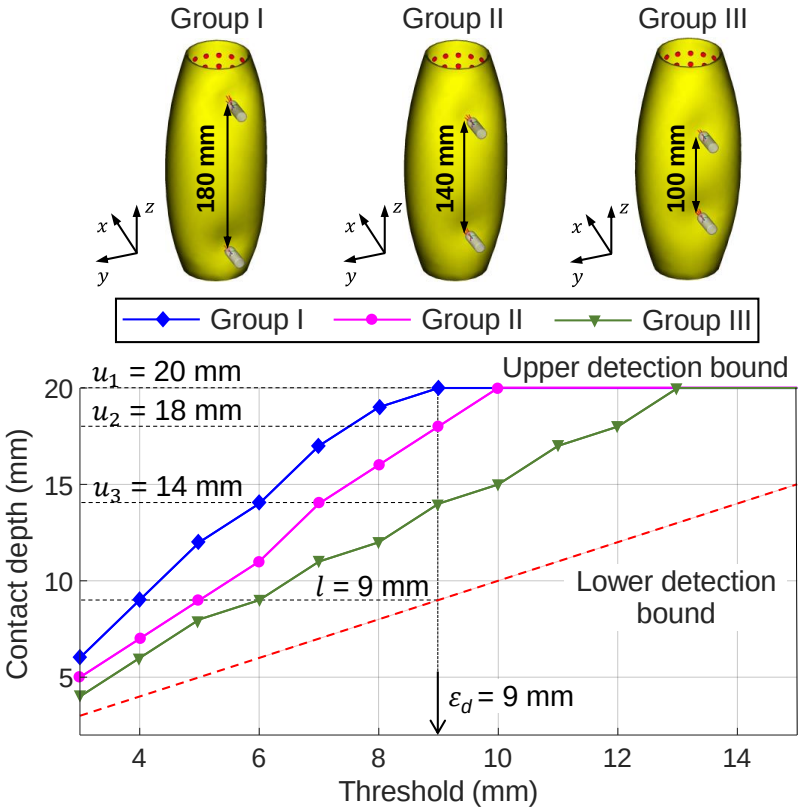
(b) The accuracy of contact classifier on different types of input images, with the decision threshold 0.6.

Figure 4.16: Evaluation of contact sensing task.

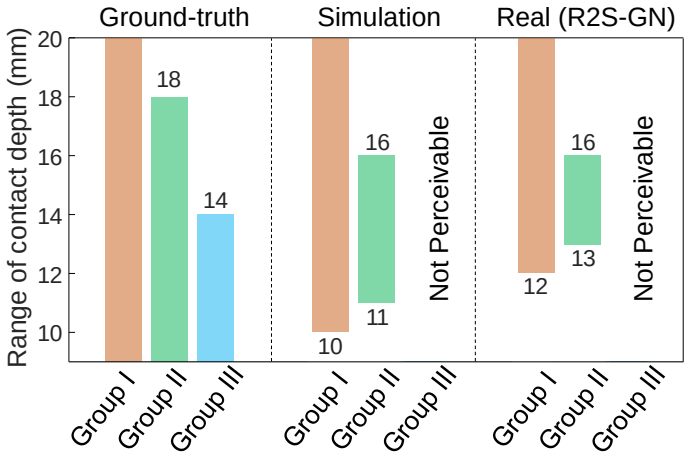
and corresponding real images, is presented in Table 4.16b. All pure real images capturing non-deformed skin were mistakenly classified as contact events, resulting in 95% precision. However, the results (Table 4.16b) indicate that this sim2real problem can be mitigated with the R2S-GN model. When real images were processed through the R2S-GN model, the threshold learned from the simulation could be successfully applied to the real images, maintaining the best precision and recall values (*i.e.*, 100%).

4.7.5 Evaluation of two-point contact localization

This section evaluates the accuracy and sim2real transferability of the contact localization task in double-contact scenarios. We conducted experiments with three distinct contact groups (I, II, III), differentiated by the vertical distance between the contact points ($180, 140, 100\,\text{mm}$, respectively, as depicted in Fig. 4.17a). For each contact group $i \in \{1, 2, 3\}$, we determined the range of contact depth $[l, u_i], i \in \{1, 2, 3\}$ for which the two separate contact regions could be identified



(a) The range of contact depth permitting successful two-point detection versus decision threshold (ϵ_d), evaluated on SOFA ground-truth data ($\mathbf{D}^{\text{FEM}}$)



(b) Sim2real comparison of contact depth range inside which the two-point contacts can be discriminated versus contact groups. ($\epsilon_d = 9\,\text{mm}$)

Figure 4.17: The study of sim2real transferability of two-point contact localization.

based on the threshold ϵ_d (Eq. 4.15) from SOFA simulation data ($\mathbf{D}^{\text{FEM}}$). Outside of the range $[l, u_i]$, the two contact points were mistakenly identified as one single

contact. Figure 4.17a demonstrates that Group I exhibited the widest detectable range, followed by Group II, and this range expanded with increasing threshold values ϵ_d. Moreover, Figure 4.17b compares these outcomes with the estimated displacement data ($\mathbf{D}^{\text{est}}$) derived from virtual and real tactile images at $\epsilon_d = 9\,\text{mm}$, which yielded the highest successful rate for two-point detection. Except for Group I, which remained acceptable in both scenarios, Group II experienced a notable decrease in detectable range, while all the two-point contacts in Group III (with a relatively close two-point distance) were inaccurately identified as a single large contact area. This limitation can be accounted for by the fact that our method estimates node displacements not only from the actual contact sites but also from the surrounding regions. Additionally, occlusion is more likely to occur when two points are closely situated. Situations, where two points share the same height or are vertically aligned, are anticipated to be less problematic than the tested cases due to the parallel alignment of every horizontal cross-section of the TacLink skin with image planes, thereby providing clearer visibility of the contact areas with less occlusion.

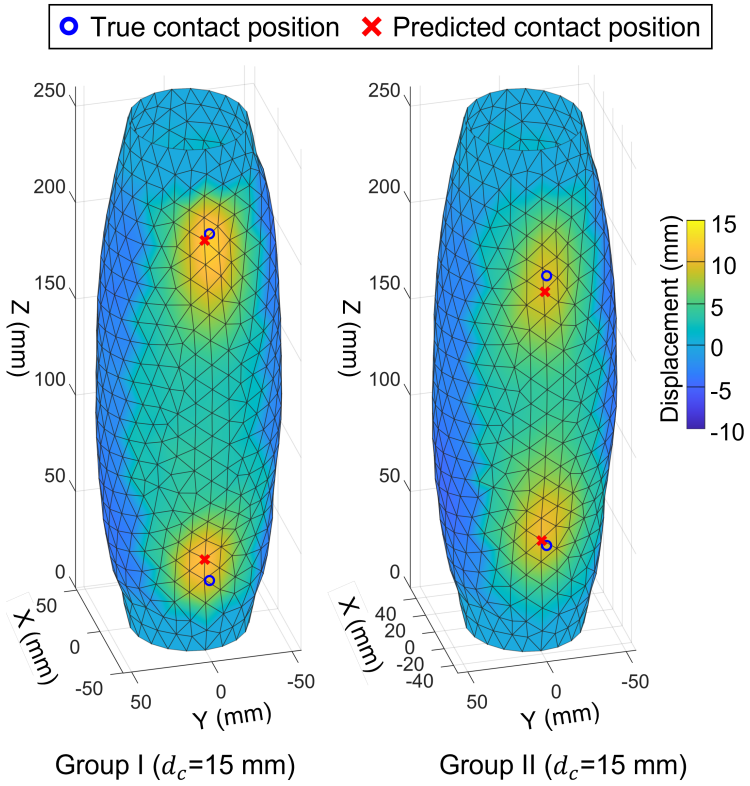
Table 4.18a illustrates the average localization errors between estimated and actual contact positions using simulation and transformed real tactile images for Groups I and II, while Figure 4.18b provides a visual representation of the localization task at $d_c = 15\,\text{mm}$. Overall, the results demonstrate the feasibility of sim2real transfer for multi-point contact localization. However, certain gaps persist in the sim2real transition due to TacNet being trained predominantly with a limited double-touch dataset and R2S-GN lacking relevant prior knowledge to manage such intricate interactions.

4.7.5.1 Discussion on two-point touch discrimination

As demonstrated in Section 4.7.5, the tactile sensor could discriminate two contact points minimally separated at a distance of 140 mm. This minimum distance often termed the two-point touch threshold, is indicative of human touch acuity. Typically, for larger body parts like arms or torso, the two-point touch threshold for humans is relatively low, usually around 45 mm. Therefore, the performance observed in

Contact Point	Localization error (mm)			
	Group I		Group II	
	sim	real	sim	real
$\mathbf{x}_1^c$	$6.81 \pm 0.$	7.19 ± 1.06	$1.49 \pm 0.$	4.86 ± 4.64
$\mathbf{x}_2^c$	$6.81 \pm 0.$	7.18 ± 1.05	$1.49 \pm 0.$	7.10 ± 4.64

(a) Double-point contact localization accuracy of simulation and transformed real dataset



(b) The demonstration for double contact localization with transformed real samples for Group I and II (d_c = 15 mm).

Figure 4.18: Evaluation of two-point contact localization accuracy.

this study, for a whole-arm ViTac system with highly compliant skin, is considered satisfactory. In fact, the tactile device's two-point touch threshold is adjustable by modifying the skin's morphology, such as altering the skin material or increasing the air pressure within the enclosed skin. It is anticipated that a skin with higher stiffness would result in a shorter detectable two-point distance due to fewer deflected nodes under the same applied force (as described by Eq. 4.3). Additionally, aside from a mechanical approach to adapt the sensing behavior, improving two-point spatial acuity can be achieved by utilizing contact forces, represented by the Lagrange

multipliers $\boldsymbol{\lambda}$ (as outlined in Eq. 4.1), as a basis for a contact region labeling algorithm rather than relying solely on nodal displacements. Thus, by leveraging the contact forces inferred by TacNet (trained on force labels obtained from the SOFA kernel), contact regions could be refined to include only the nodes physically in contact with the external stimuli.

Chapter 5

Proximity Perception

This chapter provides a detailed methodology and performance assessment of the *ProTac*'s proximity perception. While various methods for distance measurement from off-the-shelf RGB-D cameras or binocular/multi-view vision have been extensively studied in previous works [33, 93], the utilization of a critical configuration of opposing cameras, as seen in the *ProTac* link, has barely investigated. This study proposes a methodology for estimating the distance between the *ProTac* skin and the nearest obstacle or assessing collision risks with the surroundings by analyzing the internal camera view of the *ProTac* when its soft PDLC skin is *transparent* (see Fig. 5.1). Specifically, we employ a data-driven monocular depth estimation approach based on a Deep Neural Network (DNN) [94] to generate a depth map of the external space from the *ProTac*'s transparent view (Section 5.1), which serves as the foundation for distance measurement (Section 5.2) and risk assessment (Section 5.3). This approach allows for independent obstacle observation from any direction using each of the two opposing cameras, thereby extending sensing coverage and improving applicability to other sensor configurations. The fusion of sensing information from multiple camera views to enhance sensing performance is further discussed in Section 5.4. Lastly, the performance of the proximity sensing is discussed in Section 5.5.

5.1 Monocular depth estimation

In this study, we establish the mapping between *ProTac* images and estimated depth maps utilizing a DNN model trained via supervised learning. To accomplish this, we fine-tune the pre-trained MiDas model [94] using the *MannequinChallenge* dataset [95], which consists of video clips depicting motionless individuals resembling

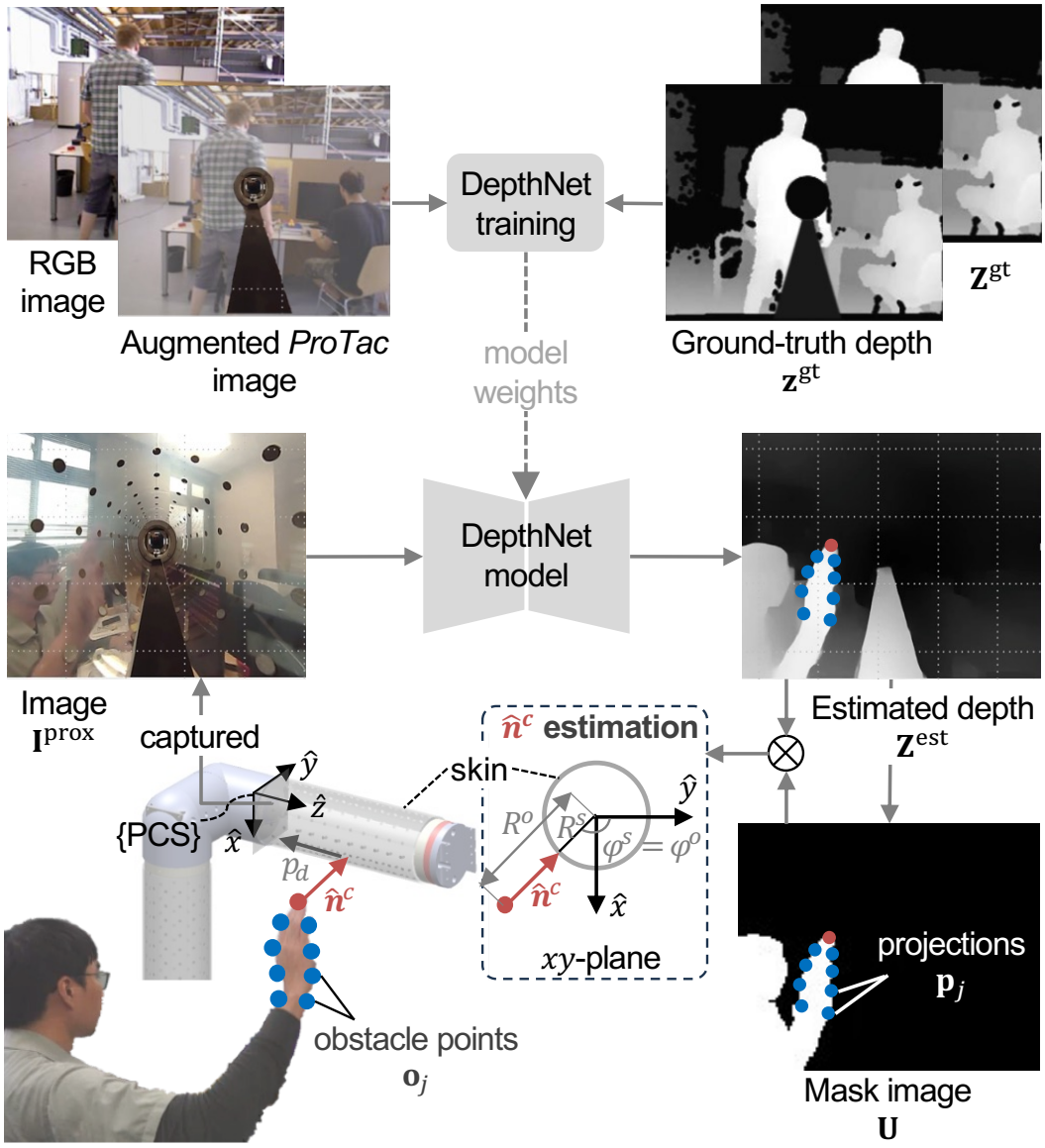


Figure 5.1: Illustration of proximity processing pipeline. The DepthNet model is fine-tuned on augmented *ProTac* images sourced from open-access datasets. Estimation of the distance to the *ProTac* skin $\hat{\boldsymbol{n}}^c$ relies on depth-map estimates $\mathbf{Z}^{\mathrm{est}}$ and a mask image $\mathbf{U}$ extracted using image processing techniques. It's important to note that while the illustration depicts obstacle points $\mathbf{o}_j$ and their projections $\mathbf{p}_j$, these may not accurately reflect real data points, and not all points are presented.

mannequins, publicly accessible on YouTube through Google AI. Initially, the image dataset is synthesized to replicate the *ProTac*'s transparent views $\mathbf{I}^{\text{prox}}$ (see Fig.5.1) by employing the alpha blending technique. Subsequently, the monocular depth estimation network (DepthNet) is trained to correlate the augmented images with corresponding depth images $\mathbf{Z}^{\text{gt}}$ generated by the MVS pipeline proposed in [96], which serve as ground-truths for model training.

5.1.1 Loss function

In the training phase, the depth estimation may exhibit varying scales. To mitigate this issue, we implement a scale-invariant depth regression loss, following the approach outlined in [94]. Additionally, we enhance the learning process by disregarding uncertain depth pixels in the ground truth, particularly those within occluded regions caused by mechanical components of *ProTac*, to refine learning efficiency. Let $\mathbf{I}^m = (\mathbf{I}^m_j \in 0, 1, \forall j \in \{1, 2, \cdots, a \times b\}) \in \mathbb{Z}_2^{a \times b}$ denote the mask of the mechanical structures obstructing the transparent view $\mathbf{I}^{\text{prox}} \in \mathbb{R}^{a \times b \times 3}$ of *ProTac*, where $\mathbf{I}^m_j = 0$ indicates an occluded pixel. We define a set of indices for occluded pixels as $\mathcal{K} := \{j \mid \mathbf{I}^m_j = 0, \forall j \in \{1, 2, \cdots, a \times b\}\}$. Given the raw estimated and ground-truth depth maps $\mathbf{Z}^{\text{est}}, \mathbf{Z}^{\text{gt}} \in \mathbb{R}^{a \times b}$, the valid depth estimation and ground truth are represented as $\mathbf{z}^{\text{est}} := (\mathbf{Z}^{\text{est}}_k, \forall k \in \mathcal{K})$ and $\mathbf{z}^{\text{gt}} := (\mathbf{Z}^{\text{gt}}_k, \forall k \in \mathcal{K})$, respectively. Consequently, the scale-invariant regression loss is formulated as:

$$\mathcal{L}_{\text{DepthNet}} = \mathcal{L}_{\text{ssitrim}}(\mathbf{z}^{\text{est}}, \mathbf{z}^{\text{gt}}) + \alpha \mathcal{L}_{\text{grad}}(\mathbf{z}^{\text{est}}, \mathbf{z}^{\text{gt}}). \tag{5.1}$$

Frist, the initial term $\mathcal{L}_{\text{ssitrim}}$ penalizes the absolute disparity in depth values between $\mathbf{z}^{\text{est}}$ and $\mathbf{z}^{\text{gt}}$. Thus, the scale-invariant depth regression loss $\mathcal{L}_{\text{ssitrim}}$ can be defined as:

$$\mathcal{L}_{\text{ssitrim}} = \frac{1}{2|\mathcal{K}|} \sum_{j=1}^{U_m} |\bar{\mathbf{z}}^{\text{est}}_j - \bar{\mathbf{z}}^{\text{gt}}_j|, \tag{5.2}$$

where $\mathbf{z}^{\bar{\text{est}}}$, $\mathbf{z}^{\bar{\text{gt}}}$ represent the normalized depth prediction and ground-truth (i.e., zero mean and unit scale), and $|\mathcal{K}|$ denotes the number of valid pixels. To enhance training robustness, the top 20% of the largest residuals of depth deviation are

trimmed, ensuring that $|\bar{\mathbf{z}}_j^{\text{est}} - \bar{\mathbf{z}}_j^{\text{gt}}| \leq |\bar{\mathbf{z}}_{j+1}^{\text{est}} - \bar{\mathbf{z}}_{j+1}^{\text{gt}}|$ and $U_m = 0.8|\mathcal{K}|$ [94]. Second, the multi-scale gradient term $\mathcal{L}_{\text{grad}}$ promotes sharp depth discontinuities and smooth gradient transitions by calculating the sum of absolute differences between predicted depth derivatives and ground-truth depth derivatives (along the x and y directions) at multiple scales ($M = 4$), where the image resolution is halved at each scale level [97]:

$$\mathcal{L}_{\text{grad}} = \frac{1}{|\mathcal{K}|} \sum_{m=1}^{M} \sum_{j=1}^{|\mathcal{K}|} (|\nabla_x R_j^m| + |\nabla_y R_j^m|), \tag{5.3}$$

where R^m signifies the difference of depth maps at scale m, with $R_j = \bar{\mathbf{z}}_j^{\text{est}} - \bar{\mathbf{z}}_j^{\text{gt}}$. Detailed derivations of normalized depth values and loss functions are available in [94].

5.1.2 Network architecture and training

The DepthNet model, employed for monocular depth estimation, adopts a multi-scale ResNet architecture [98]. We initialize the DepthNet with the model weights specified in [94]. During the fine-tuning process, we utilize the Adam optimizer with a learning rate initialized at 10^{-4}, which linearly decays at the 50^{th} iteration over a total of 100 training steps. The hyperparameter α in the combined loss function (5.1) is empirically set to 0.1. Detailed information regarding the network architecture can be found in [98].

5.2 Distance estimation

Here, we describe a method to extrapolate the distance between an external obstacle and the *ProTac* link based on the depth map estimate $\mathbf{Z}^{\text{est}}$ derived from the fine-tuned DepthNet model.

To accomplish this objective, we initially extract the mask image $\mathbf{U} \in \mathbb{Z}_2^{a \times b}$ of nearby obstacles from $\mathbf{Z}^{\text{est}}$ through binary thresholding. This approach assumes that obstacles in close proximity would exhibit discernible, brighter pixel intensities. With a set of obstacle points denoting nearby obstacles on the mask image $\mathbf{U}$,

indexed by

$$\mathcal{O} = \{j \,|\, \mathbf{U}_j \wedge \mathbf{I}_j^m = 1, \forall j \in \{1, 2, \cdots, a \times b\}\}, \tag{5.4}$$

the 3D coordinates of the obstacle points $\mathbf{O} = [\mathbf{o}_j^\intercal, \forall j \in \mathcal{O}] \in \mathbb{R}^{|\mathcal{O}|\times 3}$ can be calculated from their projections $\mathbf{P} = [\mathbf{p}_j^\intercal, \forall j \in \mathcal{O}] \in \mathbb{R}^{|\mathcal{O}|\times 3}$ on the depth image $\mathbf{Z}^{\text{est}}$ using the pinhole model of the *ProTac*'s inner camera.

Given the fisheye-lens camera of the *ProTac* sensor is modeled as a traditional pinhole model, assuming that pixel sensors are square-shaped (*i.e.*, equal focal lengths f along the x- and y-axes: $f = f_x = f_y$), the spatial relationship between the 3D position of an obstacle point $\mathbf{o} = [o_x, o_y, o_z]^\intercal \in \mathbb{R}^3$ in the PCS (Protac Coordinate System) and its projection $\mathbf{p} = [p_x, p_y, p_d]^\intercal \in \mathbb{R}^3$ in the depth space, with p_d derived from the estimated map $\mathbf{Z}^{\text{est}}$ at a specific pixel location $(u, v)^\intercal$, can be deduced as follows (for brevity, the subscript j for $\mathbf{o}$ and $\mathbf{p}$ is omitted):

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \frac{f}{b + o_z} \begin{bmatrix} o_x \\ o_x \end{bmatrix}, \; p_d = o_z \tag{5.5}$$

with

$$p_x = u - c_x, \; p_y = v - c_y, \tag{5.6}$$

where b denotes the position of PCS origin in the camera frame; (c_x, c_y) is the pixel location of the image principal point on the pixel uv-coordinate. Thus, the obstacle location in Cartesian space could be algebraically calculated as:

$$o_x = \frac{(u - c_x)(b + o_z)}{f}, \; o_y = \frac{(v - c_y)(b + o_z)}{f}, \; o_z = p_d. \tag{5.7}$$

The calibration of model parameters $\{f, c_x, c_y, b\}$ and the fisheye-lens correction were conducted following the method proposed in [12].

The remaining problem involves determining the perpendicular distance from each obstacle point $\mathbf{o}$ to the skin surface. To simplify calculations, the obstacle point's Cartesian coordinates were converted to cylindrical coordinates $[R^o, \varphi^o, p_d]^\intercal$ in 3D space, where $R^o \in \mathbb{R}_{>0}$ and $\varphi^o \in (-\pi, \pi]$ represent the radial and angular

coordinates of the PCS. This conversion can be mathematically expressed as:

$$R^o = \sqrt{o_x^2 + o_y^2}, \quad \varphi^o = \arctan 2(o_y, o_x). \tag{5.8}$$

Next, consider a specific point $[R^s, \varphi^s, p_d]^\intercal$ on the skin surface with the same angular and axial coordinates as the obstacle point $\mathbf{o}$ ($\varphi^s = \varphi^o$). The normal distance vector $\hat{\boldsymbol{n}} \in \mathbb{R}^3$ between $\mathbf{o}$ and the skin surface can then be estimated as:

$$\hat{\boldsymbol{n}} = (R^o - R^s)\frac{\mathbf{r}}{\|\mathbf{r}\|}. \tag{5.9}$$

where $\mathbf{r}$ is the directional vector of the obstacle point $\mathbf{o}$, perpendicular to the cylindrical axis of Protac, defined as $\mathbf{r} := \mathbf{o}^\intercal - [0, 0, p_d]^\intercal$. Here, the radial coordinate R^s remains constant for all control points on the skin surface, as the current Protac design features a cylindrical skin shape with a radius R, ensuring $R^s = R$ for all (φ^s, p_d).

Finally, given $[\hat{\boldsymbol{n}}_j, \forall j \in \mathcal{O}]$ determined for each obstacle point $[\mathbf{o}_j, \forall j \in \mathcal{O}]$ (based on Eq. 5.9), the distance vector $\hat{\boldsymbol{n}}^c$ from an obstacle to the Protac skin can be defined as the closest obstacle points. Hence, we have:

$$\hat{\boldsymbol{n}}^c := \arg\min_{\hat{\boldsymbol{n}}_j} \|\hat{\boldsymbol{n}}_j\| \ , \ \forall j \in \mathcal{O}. \tag{5.10}$$

Subsequently, the distance estimation can be determined as the magnitude of the distance vector $\|\hat{\boldsymbol{n}}^c\|$.

5.3 Risk score

We introduce a *risk score* to assess the collision risks of nearby obstacles, offering an alternative approach to measure proximity compared to the conventional distance estimation $\|\hat{\boldsymbol{n}}^c\|$. Although the risk score doesn't provide a direct measurement of distance, it offers a more intuitive metric that increases as obstacles come closer to the *ProTac* link. Moreover, it demonstrates a higher sensitivity and maintains a consistent measurement range across different obstacles, unlike distance

measurements that necessitate thorough calibration for each unique obstacle scenario (as discussed in Sec. 5.5). Drawing from the observation that an obstacle's area expands as it approaches the *ProTac* link, we devise the risk score metric by integrating the obstacle's pixel area $A \in \mathbb{R}$ and the corresponding estimated distance $\|\hat{\boldsymbol{n}}^c\|$. Thus, while aligning with the direction of $\hat{\boldsymbol{n}}^c$, the magnitude of the risk score is computed as follows:

$$r = \frac{A - \|\hat{\boldsymbol{n}}^c\| A_0^2}{A_0 \|\hat{\boldsymbol{n}}^c\| (\eta - A_0)}, \tag{5.11}$$

where A_0 denotes the pixel area upon initial obstacle detection. Equation (5.11) yields the raw risk score value $A/A_0\|\hat{\boldsymbol{n}}^c\|$, subsequently normalized within the range $[A_0, \eta]$. We set $\eta = 5$ across all tested obstacles. The assessment of the risk score, accompanied by a comparison with the distance estimation $\|\hat{\boldsymbol{n}}^c\|$, is outlined in Section 5.5.

5.4 Multi-camera fusion

While employing a *single* camera for extracting proximity information from *ProTac* may widen the technology's applicability to diverse sensor designs, the integration of multiple camera perspectives could elevate the sensing efficacy, particularly in scenarios where the observable or measurement range is restricted. This section outlines a straightforward approach for fusing sensing data, encompassing either the risk score r or direct distance measurement $\|\hat{\boldsymbol{n}}^c\|$, obtained from two cameras, resembling the configuration of the current *ProTac* link. Denoting s_1 and s_2 as the proximity information acquired from Camera-1 and Camera-2, respectively, the fused sensing signal s at any given time instance can be calculated:

$$s = \max(s_1, s_2) \tag{5.12}$$

Here, the sensing signal s could represent either the risk score ($s := r$) or direct distance measurement ($s := \|\hat{\boldsymbol{n}}^c\|$). The effectiveness of this fusion methodology in improving *ProTac* proximity sensing capabilities is evaluated in Section 5.5.

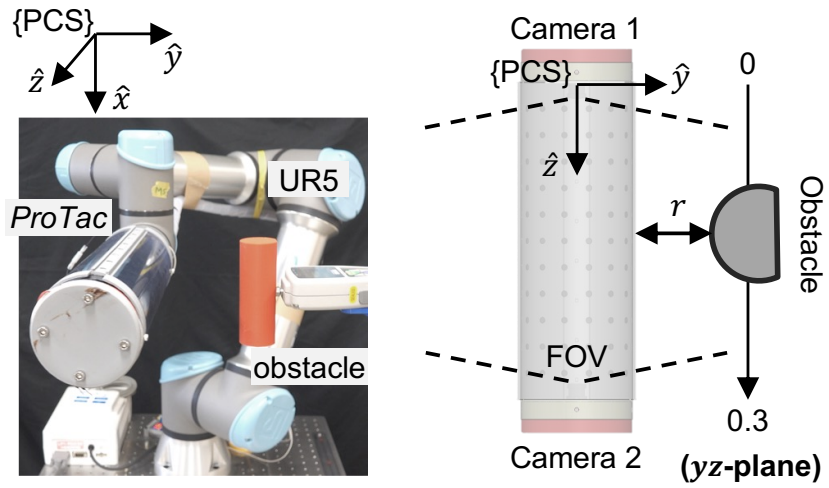


Figure 5.2: Experimental setup for the evaluation of proximity sensing performance.

5.5 Performance evaluation

This section demonstrates the *ProTac*'s capability in estimating the distance between the closest external obstacles and the *ProTac* skin $\|\hat{\boldsymbol{n}}^c\|$. Furthermore, we evaluate the risk-score metric r, demonstrating its advantage over direct distance measurement. The proximity sensing pipelines ran at approximately 22 Hz.

5.5.1 Setups

The experimental configuration is depicted in Fig. 5.2. The *ProTac* link is affixed to the end-effector of a UR5 robotic arm, which is directed linearly toward a stationary obstacle along the $\hat{y}$-axis of the ProTac Coordinate System (PCS). Throughout this motion, measurements of the distance $\|\hat{\boldsymbol{n}}^c\|$ and an equivalent risk score r were logged. The actual distance from the obstacle to the *ProTac* skin was inferred from the predetermined movements of the UR5, utilizing position feedback. To evaluate the repeatability of *ProTac*, this measurement procedure was iterated multiple times at various UR5 velocities ranging from 10 mm/s to 20 mm/s. The assessment encompassed two obstacles of distinct shapes: a cylinder-shaped obstacle and a phantom arm (refer to Fig. 5.3).

Furthermore, an additional experiment was conducted to explore how the *ProTac* system, equipped with a pair of opposing cameras, could enhance sensing

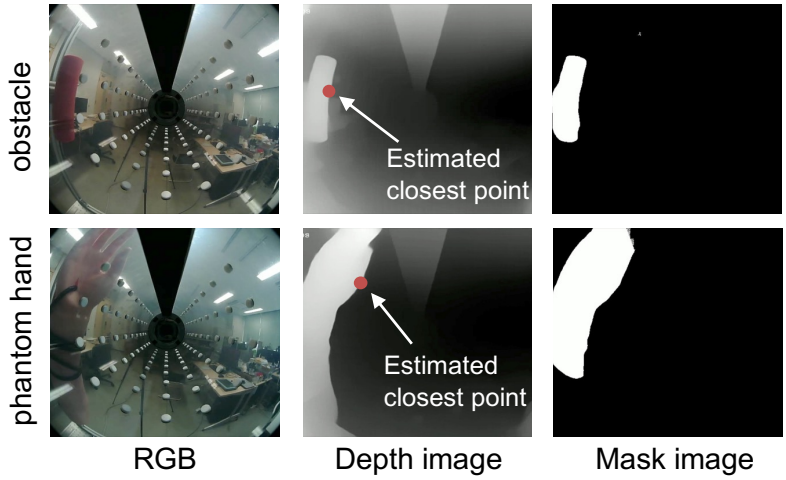


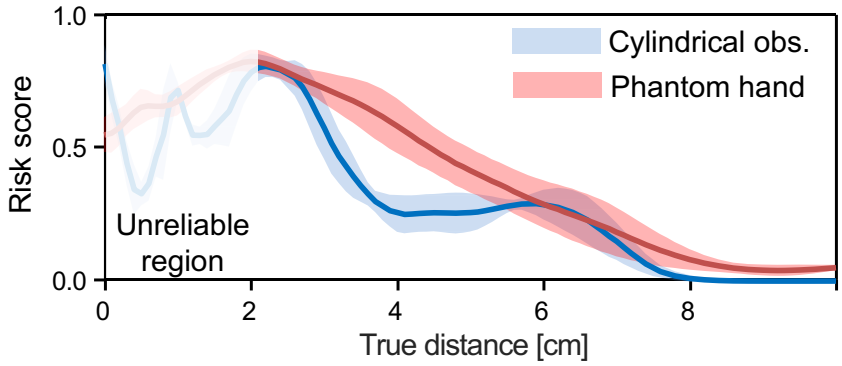
Figure 5.3: Samples of transparent *ProTac* views along with their processed images.

performance while adjusting the position of the obstacle along the z-axis (illustrated schematically in Fig. 5.2). In this experiment, the cylindrical obstacle was repeatedly displaced along the $\hat{z}$-axis of the PCS, ranging from 0 to 0.3 m, at varying speeds while maintaining a consistent distance from the *ProTac* skin. The corresponding risk score values were logged (see Sec. 5.4), and the outcomes are depicted in Figure 5.5.

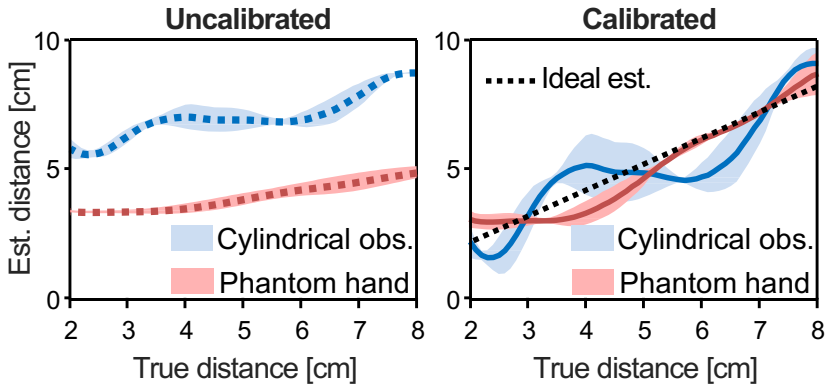
5.5.2 Results

Figure 5.3 illustrates the extraction of depth and mask images for nearby obstacles of interest using the *ProTac* view in the transparent state. The estimation of the risk score r and the distance $\|\hat{\boldsymbol{n}}^c\|$ in relation to the true distance are presented in Figures 5.4a and 5.4b, respectively. The results underscore the *ProTac*'s reliable measurement range from 2 cm to 8 cm. Notably, within this range, the risk-score estimation r displayed a linear trend, indicating a consistent measurement scale and greater sensitivity compared to the raw distance measurement $\|\hat{\boldsymbol{n}}^c\|$ (Fig. 5.4b, uncalibrated) for the two different obstacles. However, the *ProTac*'s direct distance measurement $\|\hat{\boldsymbol{n}}^c\|$ remains applicable through calibration for each specific obstacle, as evidenced in Figure 5.4b.

Regarding the performance of the two-camera fusion, as the obstacle moved linearly along the PCS's $\hat{z}$-axis, the ideal risk evaluation r should maintain a



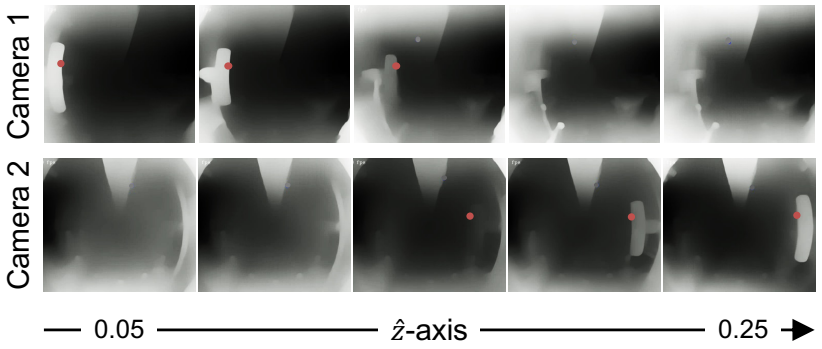
(a) risk score r



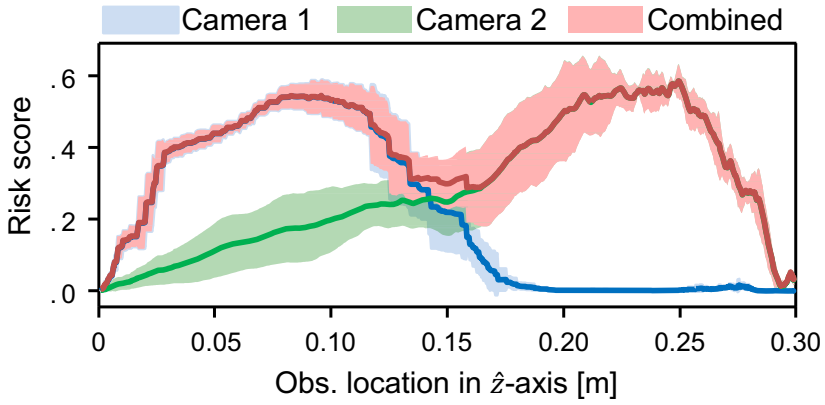
(b) estimated distance $\|\hat{\boldsymbol{n}}^c\|$

Figure 5.4: Performance of risk evaluation r and absolute distance estimation $\|\hat{\boldsymbol{n}}^c\|$ with respect to the true *ProTac*-obstacle distance for two different obstacles. Within a measurement range from 2 cm to 8 cm, the risk score r exhibited a consistent linear trend and maintained the same measurement scale for different obstacles, while calibration was required for the estimated distance values $\|\hat{\boldsymbol{n}}^c\|$.

consistent value within the field of view (FOV) of the inner cameras. However, when only one camera was utilized, the measurements of the risk score gradually deteriorated as the obstacle approached the far end from the measurement camera, whether it was Camera-1 or Camera-2 (see Fig. 5.5b). This degradation can be attributed to the substantial decline in depth-map estimations at the distal end from the respective camera, as illustrated in Figure 5.5a. Consequently, collectively leveraging measurements from both cameras restored the sensing signal over *ProTac*'s observable range (see Fig. 5.5b, red line). The combined estimation enhanced the degraded signal from one camera by leveraging the complemented measurement from the opposite one, underscoring the benefit of the current *ProTac*



(a) depth estimations



(b) risk score along $\hat{z}$-axis of PCS

Figure 5.5: Demonstrating the benefit of combining two cameras for proximity sensing performance. While the estimated risk score r by a single camera failed to observe or deteriorated with the obstacle moving to the far end along the principal $\hat{z}$-axis, either for Camera-1 or Camera-2, the combination of the two opposite cameras (red line) restored the sensing performance over the *ProTac*'s observable range within the field of view (FOV).

link's design in improving sensing performance.

In Chapter 6, we demonstrate how either direct distance measurement $\|\hat{\boldsymbol{n}}^c\|$ or the risk score metric r is utilized in various scenarios for safety control and tasks driven by *ProTac* two-mode sensing.

Chapter 6

ProTac-driven Control and Application

This chapter attempts to address the research question of whether ProTac *with softness and multimodal sensing can facilitate task performance.* In addition, this chapter demonstrates the applications of the *ProTac* link for integrating with newly constructed or existing commercial robot arms in performing control tasks. We explore *ProTac*-driven robotic tasks leveraging the combined capabilities of proximity and tactile perception, along with a unique *flickering* sensing mode and proposed sensing strategies (see Section 6.1). The first use case aims to facilitate robot motion and minimize potential damage to surroundings in cluttered environments by employing obstacle awareness and contact anticipation (see Section 6.2). The second use case aims to enhance a human-robot interaction scenario by leveraging the unique *flickering* sensing mode of *ProTac* (see Section 6.3). Notably, while we employ a distance-based mode-switching strategy to facilitate the first use case, the second scenario is enabled by a sensing strategy where *ProTac* modes are activated based on the intention of contacts.

Furthermore, we demonstrate the utilization of ProTac's proximity sensing in a couple of safety control strategies. We begin by providing a brief overview of an admittance control framework that can be combined with distance estimation in proximity for obstacle avoidance (see Section 6.4). Subsequently, we introduce a simple strategy to adjust the robot's speed according to the distance estimated by the *ProTac* link, which relies on the adaptive time-scaling of a trajectory (see Section 6.5).

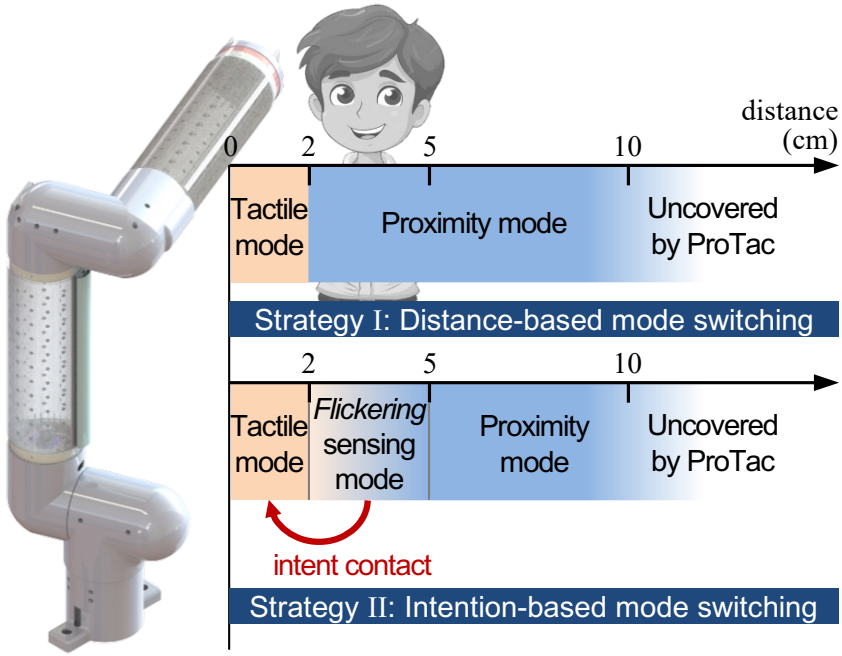


Figure 6.1: Illustration of strategies for *ProTac* mode switching. Strategy I: When the distance is below 2 cm, *ProTac* switches from *proximity* to *tactile* mode for contact anticipation. Strategy II: During *flickering* sensing mode, *ProTac* switches to *tactile* mode if intentional contact is detected; otherwise, it returns to *proximity* mode when minimal risk is observed. *Flickering* sensing mode is activated by constantly switching between the proximity and tactile modes at a high frequency.

6.1 ProTac flickering sensing mode and sensing strategies

To facilitate the utilization of *ProTac* for multimodal tasks, this subsection introduces an additional unique *ProTac* sensing mode named *flickering* sensing, along with two different sensing strategies for *ProTac* mode switching.

6.1.1 Flickering sensing mode

The "flickering" sensing mode refers to a *ProTac* operational mode where proximity and tactile sensing can be enabled nearly simultaneously. This mode is achieved by constantly switching between the proximity and tactile modes at a high frequency or at a certain switching period T_s. During the *flickering* mode, the last sample value estimated in one mode is retained for one switching period T_s when switching to the other mode, following the zero-order hold model.

6.1.2 Sensing strategies

Figure 6.1 illustrates the two strategies for switching among *ProTac* sensing modes, which are outlined as follows:

- Distance-based mode switching (Strategy I): *ProTac* begins in *proximity* mode. When obstacles are detected at a close distance or when high-risk level is observed, *ProTac* switches to *tactile* mode to anticipate contacts or collisions in a pre-contact phase. This mode switch occurs when the proximity sensing becomes unreliable, that is, below 2 cm.
- Intention-based mode switching (Strategy II): The *ProTac* initially operates in *proximity* mode. When a proximal distance is detected, typically below 5 cm, *ProTac* switches to *flickering* mode. If intentional contact is detected, *ProTac* switches to *tactile* mode; otherwise, it returns to *proximity* mode when minimal risk is observed.

6.2 Motion control with contact and obstacle awareness

The navigation of robot arms through cluttered environments often poses challenges to external perception and navigation systems, necessitating kinematic redundancy to reach target locations while avoiding collisions. However, collisions may be unavoidable in certain environments, such as densely wooded areas, making it crucial to minimize damage to the surroundings while achieving task objectives. Previous studies have predominantly focused on employing high-stiffness skin-based tactile sensing, without considering the surrounding environment prior to contact, to accomplish tasks [72]. In contrast, this study utilizes both soft skin-based tactile sensing and proximity perception to enhance awareness of obstacles during the pre-contact phase, thereby improving contact-constrained motion to mitigate impact forces (see Section 6.2.2 for results).

6.2.1 Problem formulation

We present a methodology for guiding robot motion towards a target location in Cartesian space $\boldsymbol{x}_{\mathrm{G}} \in \mathbb{R}^3$ that may be near or obstructed by obstacles while minimizing physical impacts on these obstacles. This involves constraining the estimated contact depth $\|\hat{\boldsymbol{d}}^c\|$ to remain below a specified threshold $d_{\max} \in \mathbb{R}_{>0}$. The task is framed as a constrained Quadratic Programming (QP) problem aimed at optimizing commanded joint velocities $\dot{\boldsymbol{\theta}}_d \in \mathbb{R}^n$ to minimize an objective function $\mathcal{J}(\dot{\boldsymbol{\theta}}_d)$. This objective function $\mathcal{J}(\dot{\boldsymbol{\theta}}_d)$ is quadratic in nature and captures the target location error, defined as:

$$\mathcal{J}(\dot{\boldsymbol{\theta}}_d) := \frac{1}{2}(\frac{\boldsymbol{K}_p}{k}\Delta\boldsymbol{x} - \boldsymbol{J}_e\dot{\boldsymbol{\theta}}_d)^\intercal(\frac{\boldsymbol{K}_p}{k}\Delta\boldsymbol{x} - \boldsymbol{J}_e\dot{\boldsymbol{\theta}}_d), \tag{6.1}$$

where $\Delta\boldsymbol{x} \in \mathbb{R}^3$ represents the directional vector towards the target location, defined as $\Delta\boldsymbol{x} := \boldsymbol{x}_{\mathrm{G}} - \boldsymbol{x}$, $\boldsymbol{J}_e \in \mathbb{R}^{3\times n}$ is the Jacobian matrix, and $\boldsymbol{K}_p \in \mathbb{R}^{3\times 3}$ is a positive-definite diagonal proportional matrix. Notably, we incorporate the proximity effect, accounting for potential obstacles before contact, directly into the optimization problem by adjusting the proportional matrix $\boldsymbol{K}_p$ with the time-scaling factor k (refer to Eq. 6.12). This adjustment means that the optimized velocity $\dot{\boldsymbol{\theta}}_d$ obtained from (6.1) is smaller when the robot is close to an obstacle (where $k > 1$), compared to when the obstacle is distant, that is $k = 1$.

Moreover, once the robot encounters the obstacle, we enforce a motion restriction $\mathcal{C}(\dot{\boldsymbol{\theta}}_d)$ on the contact, determined by $\hat{\boldsymbol{d}}^c$, according to the following expression:

$$\mathcal{C}(\dot{\boldsymbol{\theta}}_d) := (\hat{\boldsymbol{d}}^c + \beta\hat{\boldsymbol{e}}_c^\intercal\boldsymbol{J}_c\dot{\boldsymbol{\theta}}_d\hat{\boldsymbol{e}}_c)^\intercal(\hat{\boldsymbol{d}}^c + \beta\hat{\boldsymbol{e}}_c^\intercal\boldsymbol{J}_c\dot{\boldsymbol{\theta}}_d\hat{\boldsymbol{e}}_c) \leq d_{\max}^2, \tag{6.2}$$

where $\hat{\boldsymbol{e}}_c \in \mathbb{R}^3$ represents the unit vector of the estimated contact direction, defined as $\hat{\boldsymbol{e}}_c := \hat{\boldsymbol{d}}^c/\|\hat{\boldsymbol{d}}^c\|$, β is a parameter controlling the smoothness of the constrained motion, and $\boldsymbol{J}_c$ is the Jacobian matrix at the contact point $\boldsymbol{x}_c$. The derivation of $\boldsymbol{J}_c$ from the end-effector Jacobian $\boldsymbol{J}_e$ can be found in [99]. Hence, the commanded

Algorithm 2 Motion control with contact and obs. awareness

Input: $\mathbf{X}_\mathrm{G} := [\boldsymbol{x}_\mathrm{G}^1, \boldsymbol{x}_\mathrm{G}^2, \cdots]$: a sequence of target locations
Output: $\dot{\boldsymbol{\theta}}_d$: commanded joint velocities

mode $\leftarrow$ *proximity* ▷ activate proximity mode
for $\boldsymbol{x}_\mathrm{G}$ in $\mathbf{X}_\mathrm{G}$ **do**
  while $\|\Delta\boldsymbol{x}\| > 10^{-3}$ **do**
    r, $\hat{\boldsymbol{n}}^c$, $\hat{\boldsymbol{d}}^c \leftarrow$ obtain sensing signals from *ProTac*
    if $r \geq \epsilon_d$ and $c_\mathrm{sim}(\Delta\boldsymbol{x}, \hat{\boldsymbol{n}}^c) > 0$ **then**
      mode $\leftarrow$ *tactile* ▷ switch to tactile mode
    end if
    if $\|\hat{\boldsymbol{d}}^c\| \geq \epsilon_d$ **then**
      $\dot{\boldsymbol{\theta}}_d \leftarrow \arg\min \mathcal{J}(\dot{\boldsymbol{\theta}}_d)$, s.t. $\mathcal{C}(\dot{\boldsymbol{\theta}}_d) \leq d_\mathrm{max}^2$
    else
      $\dot{\boldsymbol{\theta}}_d \leftarrow \arg\min \mathcal{J}(\dot{\boldsymbol{\theta}}_d)$
    end if
  end while
  mode $\leftarrow$ *proximity* ▷ get back to proximity mode
end for

velocity $\dot{\boldsymbol{\theta}}_d$ is determined as follows:

$$\dot{\boldsymbol{\theta}}_d = \begin{cases} \arg\min \mathcal{J}(\dot{\boldsymbol{\theta}}_d), & \text{if } \|\hat{\boldsymbol{d}}^c\| \geq \epsilon_d \\ \arg\min \mathcal{J}(\dot{\boldsymbol{\theta}}_d), \text{ s.t. } \mathcal{C}(\dot{\boldsymbol{\theta}}_d) \leq d_\mathrm{max}^2, & \text{otherwise} \end{cases}, \tag{6.3}$$

where the constraint is applied only when the robot makes contact, identified by $\|\hat{\boldsymbol{d}}^c\|$ exceeding a threshold ϵ_d.

Finally, Algorithm 2 details the procedure for instructing the robot to sequentially reach multiple target locations, activating the respective sensing modes at different phases based on the evaluation of the risk level (Strategy I). In this process, *ProTac* transitions from *proximity* to *tactile* mode when the risk score $r \geq \epsilon_p$ and the obstacle obstructs the path to the target location, as determined by the cosine similarity $c_\mathrm{sim}(\Delta\boldsymbol{x}, \hat{\boldsymbol{n}}^c)$ between $\Delta\boldsymbol{x}$ and $\hat{\boldsymbol{n}}^c$:

$$c_\mathrm{sim}(\Delta\boldsymbol{x}, \hat{\boldsymbol{n}}^c) := \frac{\Delta\boldsymbol{x} \cdot \hat{\boldsymbol{n}}^c}{\|\Delta\boldsymbol{x}\|\|\hat{\boldsymbol{n}}^c\|} > 0. \tag{6.4}$$

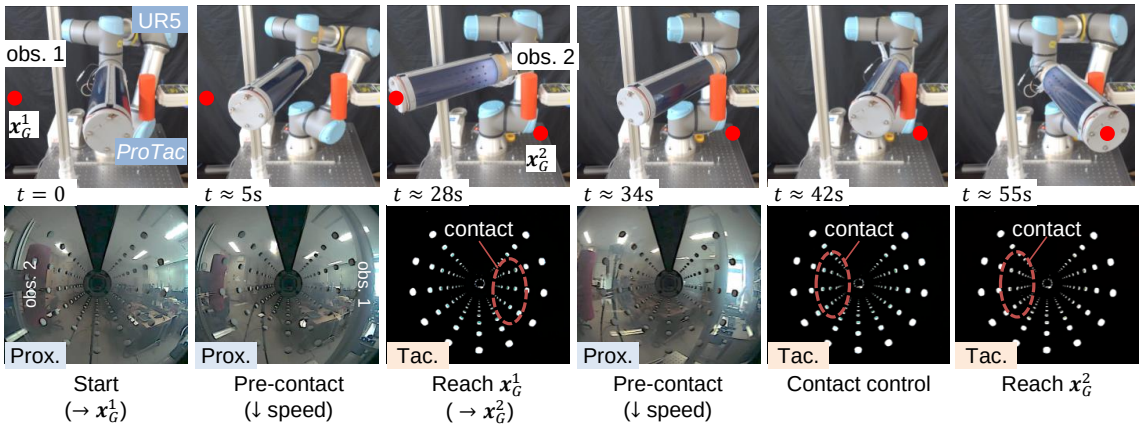


Figure 6.2: Video stills of motion roll out and corresponding images of *ProTac* views (obs. stands for obstacle). The red dots in the upper row's pictures indicate the target position for the end-effector. Refer to the supplementary video for a demonstration of these experiments: https://youtu.be/5DhAhlTVxzg

6.2.2 Experiment and evaluation: Motion control

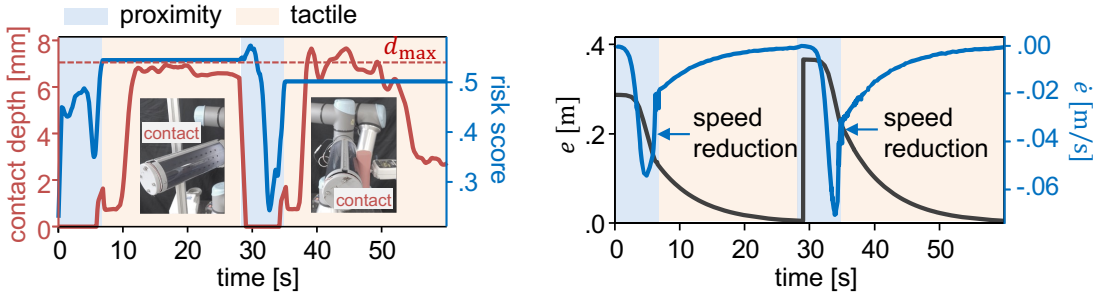
The efficiency of this task performance is validated with a 6-DoF UR5e robot arm, equipped with the *ProTac* as an extended link attaching to the robot's end-effector (see Fig. 6.2). This setup demonstrates the utilization of *ProTac* for existing commercial robot arms. Here, the commanded joint velocities $\dot{\boldsymbol{\theta}}_d$, derived from the proposed control strategies based on *ProTac* feedback, were regulated by the UR5's low-level controller and coordinated via ROS. The parameter values of the controllers are summarized in Table 6.1.

Table 6.1: Control parameters for *ProTac*-driven multimodal tasks

Parameter		Value	Unit
Diagonal proportional matrix	$\boldsymbol{K}_p$	$\mathrm{diag}(0.3, 0.3, 0.3)$	-
Regularization factor	β	0.5	-
Max. admissible contact depth	$d_{\max}$	7.0	mm
Contact threshold	ϵ_d	2.0	mm
Critical risk threshold	ϵ_p	0.45	-

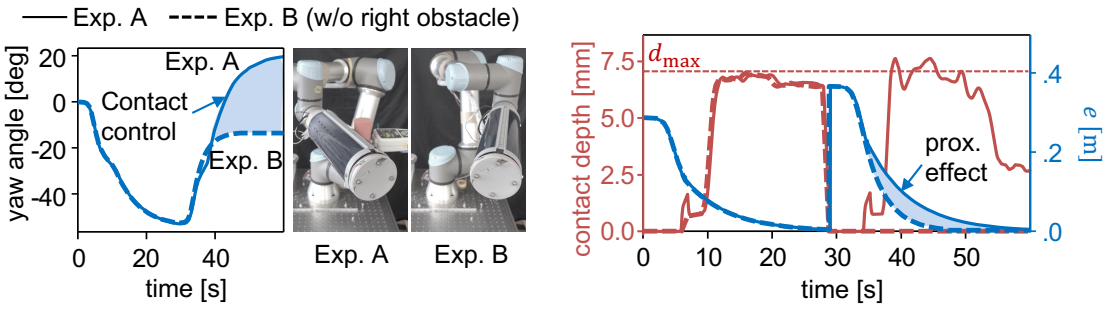
6.2.2.1 Setup

The experiment illustrates the effectiveness of integrating *ProTac* with proximity-tactile sensing modalities into the optimization controller, as detailed in Section 6.2. The numerical solution of the QP optimization problem (6.3) was conducted using



(a) logs of contact depth and risk score (b) error dynamics

Figure 6.3: *ProTac* measurements and position-controlled error dynamics.



(a) log of robot rotation (b) logs of contact depth and positional error

Figure 6.4: The motion resulted from contact constraint (Exp. A) and contact-free motion (Exp. B).

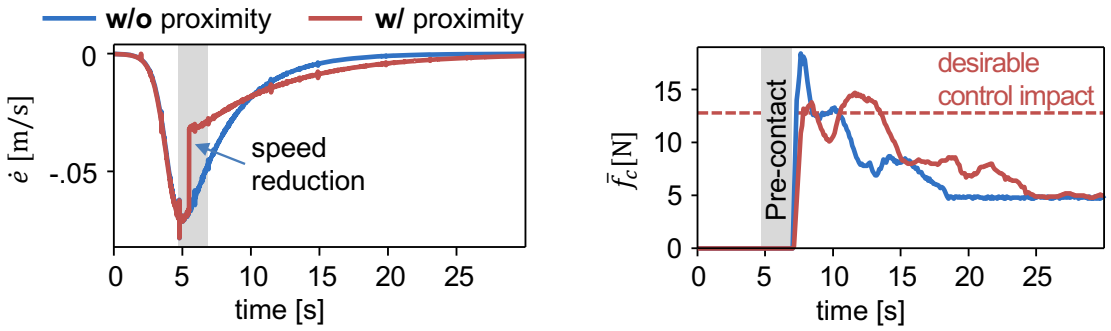
the CVXPY optimization library[1]. In this experiment, the *ProTac*-integrated robot was instructed to sequentially reach two target locations $\boldsymbol{x}_{\mathrm{G}}^{1}$ and $\boldsymbol{x}_{\mathrm{G}}^{2}$ (indicated by red circles in Fig. 6.2), while aiming to limit the contact depth magnitude $\|\hat{\boldsymbol{d}}^{c}\|$ below $d_{\max}$ to mitigate potential contact impacts. Both obstacles were strategically positioned in close proximity to each other and the target locations, simulating a cluttered environment (refer to Fig. 6.2). Additionally, obstacle-2 (on the right) was equipped with a force gauge (ZTS50N, IMADA Inc., Japan) to measure the actual impact force exerted on the *ProTac* link upon contact.

6.2.2.2 Result

Figure 6.2 depicts the performance of the robot's task and the corresponding views captured by *ProTac* in various sensing modes. The snapshots captured at

[1]CVXPY is a Python tool designed for solving convex optimization problems.

approximately $t \approx 42\,\text{s}$ and $t \approx 55\,\text{s}$ illustrate how the controller utilized the **softness** of *ProTac* skin to compensate for errors in reaching the goal. This ability to adapt and gently conform to obstacles is challenging, if not impossible, to achieve with a rigid link. From a quantitative perspective, Figure 6.3 presents the *ProTac* measurements (Fig. 6.3a) and the evolution of positional error $e := \|\Delta \boldsymbol{x}\|$ relative to the target locations (Fig. 6.3b) throughout the experimental scenario. It is evident from Figure 6.3 that as the robot approached the target location, it slowed down its speed based on the increasing risk score r and transitioned to the *tactile* mode once $r \geq \epsilon_p$. Upon contact, the robot endeavored to maintain the contact depth around the predetermined permissible threshold $d_{\max}$. Remarkably, when the location error e approached zero, the contact depth (or deformation) remained observable on the soft *ProTac* link (refer to Fig. 6.3), indicating that achieving the target location might not be feasible with a rigid link due to the inability to accommodate such deformations.



(a) positional error rate (b) measured contact force

Figure 6.5: The effect of proximity sensing on the impact reduction for contact-constrained motion control.

Figure 6.4 presents the effect of contact-based control and obstacle awareness in two scenarios, denoted as Exp-A and Exp-B, with and without an obstacle on the right side (*i.e.*, obs. 2). In Exp-A, the robot experienced a notable yaw rotation to adapt to the obstacle contact (see Fig. 6.4a), while exhibiting slower convergence to the target compared to Exp-B due to reduced speed mediated by obstacle awareness via proximity sensing (see Fig. 6.4b). Furthermore, the integration of proximity sensing into motion control effectively mitigated the impact of the *ProTac* link with

an obstacle, as demonstrated in Figure 6.5. With proximity sensing, Figure 6.5b shows that the actual impact force measured by the force gauge $\bar{f}_c$ was suppressed to the desired threshold, while a significant peak impact force was observed in the scenario without proximity integration.

These outcomes confirm the efficacy of the soft ProTac *skin, incorporating multi-modal sensing and optimization control, in enhancing motion control that may challenging with a conventional rigid link.*

6.3 Human-robot interaction with flickering sensing

Human-robot interaction typically unfolds in two phases: a coexistence phase, where robots operate alongside humans with safety measures like collision avoidance or speed adjustments, and a physical interaction phase, where robots engage in direct physical interaction with humans [34]. However, smoothly transitioning between these phases or discerning human intent for physical interaction poses significant challenges, necessitating advanced perception and learning techniques [31]. To tackle this issue, we adopt the proposed intention-based mode switching (Strategy II), incorporating the flickering sensing capability of *ProTac*. This enables the detection of human-intended contacts, facilitating seamless transitions from coexistence to physical interaction states for the robot.

6.3.1 Problem formulation

Imagine a situation where a robot equipped with *ProTac* initially operates in a coexistence phase, moving at a base velocity $\dot{\boldsymbol{\theta}}_d := \dot{\boldsymbol{\theta}}_0$, while *ProTac* is in proximity mode. During this phase, if a human presence is detected, the system switches to flickering mode. In flickering mode, the controller adjusts the speed by updating the reduced velocity $\dot{\boldsymbol{\theta}}_d$ based on the factor k (see Sec. 6.5), or halts robot movement ($\dot{\boldsymbol{\theta}}_d = \mathbf{0}$) when the risk score r surpasses a certain threshold ϵ_p. In summary, the commanded joint velocity $\dot{\boldsymbol{\theta}}_d$ is determined based on the following conditions

Algorithm 3 Human-robot interaction with *flickering* sensing

Input: $\dot{\boldsymbol{\theta}}_0$: base joint velocities, T_e: execution time
Output: $\dot{\boldsymbol{\theta}}_d$: commanded joint velocities

mode $\leftarrow$ *proximity* ▷ get in *coexistence* state
$\dot{\boldsymbol{\theta}}_d \leftarrow \dot{\boldsymbol{\theta}}_0$ ▷ initialize normal operation
while $t < T_e$ **do**
  r, $\hat{\boldsymbol{d}}^c \leftarrow$ obtain sensing signals from *ProTac*
  if *human* detected **then**
    mode $\leftarrow$ *flickering*
    $\dot{\boldsymbol{\theta}}_d \leftarrow \mathbf{0}$ **if** $r \geq \epsilon_p$ **else** $\dot{\boldsymbol{\theta}}_d \leftarrow \dot{\boldsymbol{\theta}}_0/k$ ▷ k, see (6.12)
    if $\|\hat{\boldsymbol{d}}^c\| \geq \epsilon_d$ **then**
      mode $\leftarrow$ *tactile* ▷ get into *interaction* phase
      while $L < 2$ **do** ▷ L denotes #contacts
        $\dot{\boldsymbol{\theta}}_d \leftarrow$ obtain from $\hat{\boldsymbol{d}}_c$ (refer to [92])
      end while
    end if
  else
    mode $\leftarrow$ *proximity* ▷ return to *coexistence* phase
    $\dot{\boldsymbol{\theta}}_d \leftarrow \dot{\boldsymbol{\theta}}_0$
  end if
end while

(assuming *ProTac* is in flickering mode):

$$\dot{\boldsymbol{\theta}}_d = \begin{cases} \dot{\boldsymbol{\theta}}_0/k, & \text{if human detected and } r < \epsilon_p \\ \mathbf{0}, & \text{else human detected and } r \geq \epsilon_p \\ \dot{\boldsymbol{\theta}}_0, & \text{otherwise } \textit{not} \text{ human detected} \end{cases}, \tag{6.5}$$

Moreover, if the human goes away, the robot speed returns to the base profile $\dot{\boldsymbol{\theta}}_0$. However, the transition to the physical interaction phase occurs when human-intended contact is detected, signaled by the estimated contact depth $\|\hat{\boldsymbol{d}}^c\|$ surpassing a contact threshold ϵ_d.

Upon transitioning, *ProTac* switches to tactile mode to initiate the interaction phase. During this phase, the commanded velocity $\dot{\boldsymbol{\theta}}_d$ is determined by physical interactions with humans. To guide robot motion in response to the contact depth vector $\hat{\boldsymbol{d}}^c$, we adopt the strategy proposed in [92] (further details of this strategy are discussed in Chapter 7). Finally, recognition of human-intended two-point contact can serve as a condition to terminate the interaction phase, enabling the robot to resume normal operation. An overview of this scenario is provided in Algorithm 3.

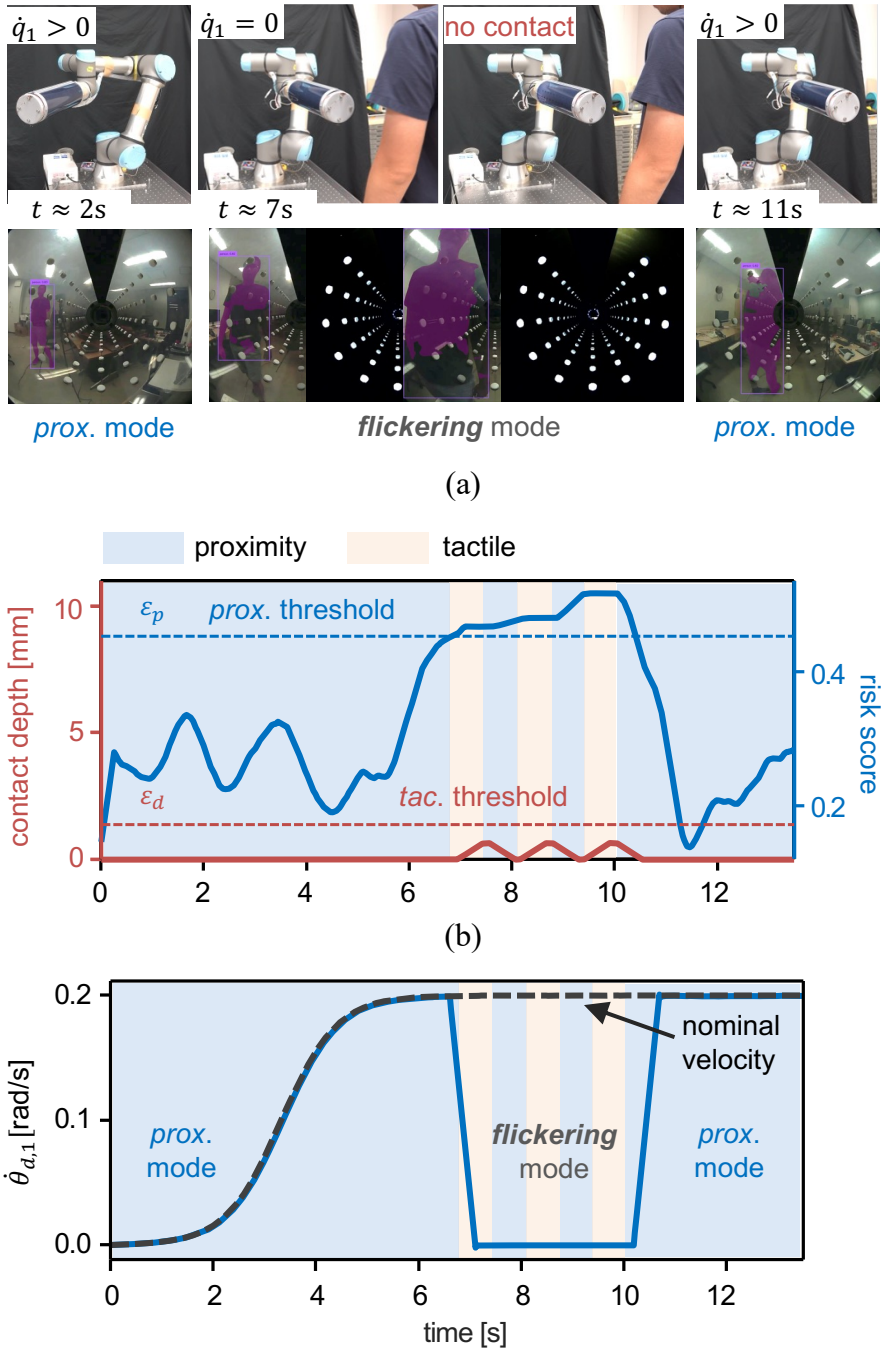
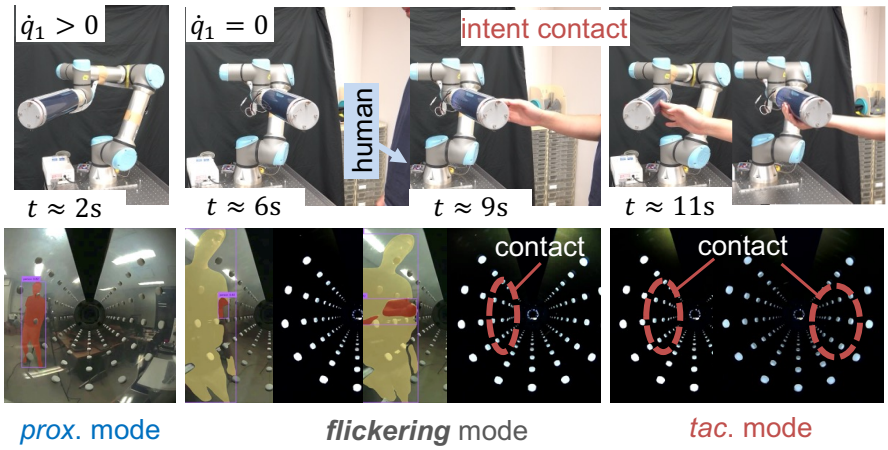


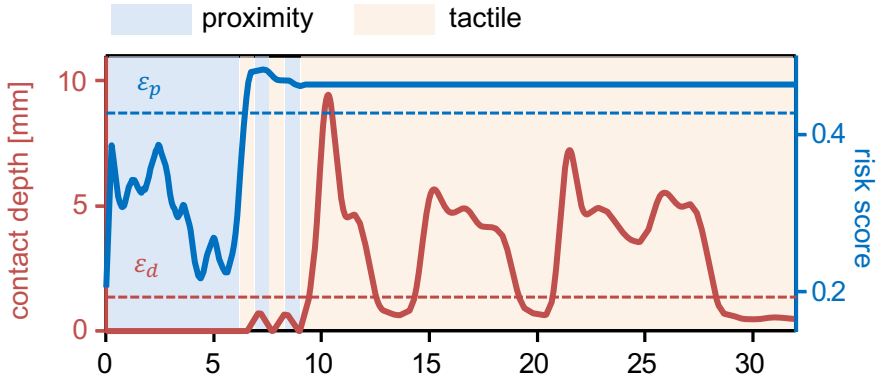
Figure 6.6: Demonstration of a human-robot interaction scenario (Scenario A): a human passerby without any interaction intention.

6.3.2 Experiment and evaluation: HRI scenario

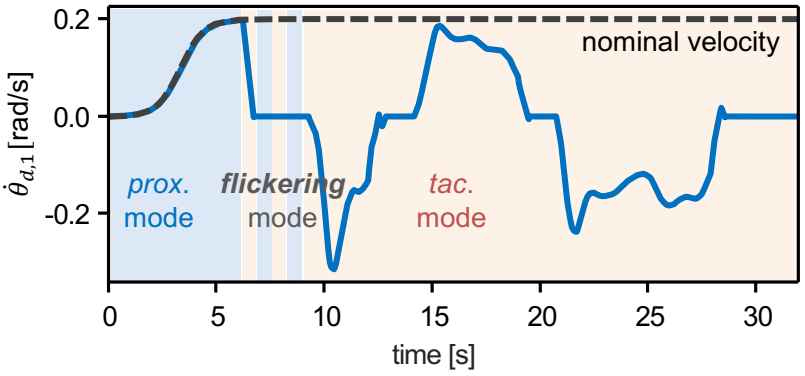
The efficiency of this task performance is validated with the same configuration and control pipeline as in the previous task (demonstrated in Section 6.2.2), where



(a)



(b)



(c)

Figure 6.7: Demonstration of a human-robot interaction scenario (Scenario B): *ProTac* identifies human contact in *flickering* sensing mode, enabling tactile-based interaction where the human guides the robot's motion.

the *ProTac* is utilized as an extended link for the 6-DoF UR5e robot arm. The parameter values of the controllers are summarized in Table 6.1.

6.3.2.1 Setup

The experiment aims to demonstrate the efficacy of the unique *ProTac* sensing modes in facilitating seamless human-robot interaction scenarios (see Section 6.3). We employed the DEVA pipeline, an open-source tool [100], to detect and track potential humans appearing in the *ProTac*'s see-through image view. Initially, the robot arm moved at a nominal velocity profile in the joint space, denoted as $\dot{\boldsymbol{\theta}}_0 = [\dot{\theta}_1, \mathbf{0}_5]_0^\intercal$. Two scenarios were examined as follows:

- Scenario A: involved a human passerby without any intention of interaction, who approached and then moved away from the robot (see Fig. 6.6).
- Scenario B: featured a human intending physical touch interaction, approaching and touching the robot to initiate interaction (see Fig. 6.7).

6.3.2.2 Result

Figures 6.6a and 6.7a depict the experimental setups and *ProTac* views for various sensing modes, showcasing Scenario A and Scenario B, respectively. Furthermore, the robot's behavior is determined through the commanded velocity profile of the base joint $\dot{\theta}_{d,1}$ illustrated in Figures 6.6c and 6.7c for Scenario A and Scenario B, respectively. Specifically, as shown in Figures 6.6b and 6.7b, the approaching human induced an increasing risk score r detected in the *proximity* mode. Once r exceeded the critical risk threshold ϵ_p, the robot halted ($\dot{\theta}_{d,1} = 0$), activating the *flickering* mode, evident from the alternating orange and blue-shaded strips in Figures 6.6b and 6.7b. In the *flickering* mode, during Scenario A, no contact was detected ($\|\hat{\boldsymbol{d}}^c\| < \epsilon_p$), and the human moved away ($r < \epsilon_p$), as observed in Fig. 6.6b. Consequently, the robot reverted to the nominal velocity profile $\dot{\theta}_{d,1} := \dot{\theta}_{0,1}$ (Fig. 6.6c). Conversely, in Scenario B, contact occurred, identified by $\|\hat{\boldsymbol{d}}^c\| \geq \epsilon_p$, immediately triggering the *tactile* mode and interaction phase (as shown in Fig. 6.7b). Consequently, the robot responded to the human's touch-based interaction (see Fig. 6.7c).

The obtained results demonstrate the effectiveness of the multimodal ProTac *sensing and the* flickering *mode in facilitating seamless transitions between the* proximity *and* tactile *modalities for different human-robot interaction phases, potentially*

enhancing human-robot interaction scenarios.

6.4 Admittance-based reactive control

This section presents a robot arm system driven by *ProTac* with reflex behavior, allowing the robot to react to nearby obstacles. This behavior can be utilized for safety applications, such as spontaneous collision avoidance or collision reaction.

6.4.1 Problem formulation

Toward the goal, we employ an admittance controller [30], which treats the robot as a mass-spring-damper system, formulated as:

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}}_d + \boldsymbol{D}_v \dot{\boldsymbol{x}} + \boldsymbol{K}_v \boldsymbol{x} = \boldsymbol{f}_{\text{ext}}, \tag{6.6}$$

where $\boldsymbol{M}_v \in \mathbb{R}^{3\times3}$ is the virtual positive-definite inertia matrix, $\boldsymbol{D}_v \in \mathbb{R}^{3\times3}$ is the virtual positive-definite diagonal damping matrix, and $\boldsymbol{K}_v \in \mathbb{R}^{3\times3}$ is the virtual positive-definite diagonal stiffness matrix. Here, $\boldsymbol{x} = [x_x, x_y, x_z]^\intercal \in \mathbb{R}^3$ and $\dot{\boldsymbol{x}} = [\dot{x}_x, \dot{x}_y, \dot{x}_z]^\intercal \in \mathbb{R}^3$ represent the position and velocity states of the robot end-effector, and $\ddot{\boldsymbol{x}}_d$ is the desired end-effector acceleration. Thus, the Cartesian-space admittance control law is derived as:

$$\ddot{\boldsymbol{x}}_d = \boldsymbol{M}_v^{-1}(\boldsymbol{f}_{\text{ext}} - \boldsymbol{D}_v \dot{\boldsymbol{x}} - \boldsymbol{K}_v \boldsymbol{x}). \tag{6.7}$$

Here, $\boldsymbol{f}_{\text{ext}} := \boldsymbol{f}_v$ represents the virtual repulsive force $\boldsymbol{f}_v$, which is linked to the distance vector from the obstacle estimated by *ProTac*, denoted as $\hat{\boldsymbol{n}}$. The mapping function is defined as:

$$\boldsymbol{f}_v = f_v \frac{\hat{\boldsymbol{n}}^c}{\|\hat{\boldsymbol{n}}^c\|}, \tag{6.8}$$

having the same direction as $\hat{\boldsymbol{n}}$ but with a magnitude given by:

$$f_v = \frac{f_v^{\max}}{1 + e^{(\|\hat{\boldsymbol{n}}^c\|(2/\rho)-1)\gamma}}. \tag{6.9}$$

This choice of mapping function is aimed at ensuring the smoothness of the robot's reactive response [33], where $f_v^{\max}$ denotes the maximum magnitude of the resultant virtual force, and γ represents a shape factor. As an obstacle approaches the *ProTac* skin, the repulsive vector's magnitude f_v increases, reaching its maximum value $f_v^{\max}$ when the estimated distance $\|\hat{\boldsymbol{n}}^c\|$ approaches zero. The virtual force gradually decreases as the distance to the obstacle approaches or extends beyond the value ρ, that is $\|\hat{\boldsymbol{n}}^c\| \geq \rho$.

Given the resultant virtual force $\boldsymbol{f_v}$ and the control law (6.7), the desired joint accelerations $\ddot{\boldsymbol{\theta}}_d \in \mathbb{R}^n$ (where n is the number of robot joints) can be obtained as per [99]:

$$\ddot{\boldsymbol{\theta}}_d = \boldsymbol{J}_e^{\dagger}(\ddot{\boldsymbol{x}}_d - \dot{\boldsymbol{J}}_e\dot{\boldsymbol{\theta}}) \tag{6.10}$$

where $\boldsymbol{J}_e^{\dagger} \in \mathbb{R}^{n\times 3}$ denotes the Moore-Penrose pseudoinverse of the end-effector Jacobian, defined by $\dot{\boldsymbol{x}}_d = \boldsymbol{J}_e\dot{\boldsymbol{\theta}}_d$. Subsequently, the commanded joint velocities can be computed as $\dot{\boldsymbol{\theta}}_d = \int \ddot{\boldsymbol{\theta}}_d$.

In this study, our primary investigation focuses on reflex behavior within 3D linear space, omitting considerations for rotational components to simplify implementation processes. Additionally, it's noteworthy that while this study specifically assesses the framework for proximity-based sensing signals, the control law (6.7) can also integrate the contact depth vector $\hat{\boldsymbol{d}}^c$ by defining $\boldsymbol{f}_{\text{ext}} := \hat{\boldsymbol{d}}^c$ to foster robot-safe behavior in response to contacts (details about this behaviour are elaborated in Chapter 7).

6.4.2 Experiment and evaluation: Obstacle avoidance

To assess the efficacy of integrating the *ProTac* link for inducing safe responses in robot arms with environmental awareness, we employed a bespoke 4-DOF (degree-of-freedom) robot arm, featuring two *ProTac* links serving as the upper arm and forearm components (referred to as the *ProTac*-integrated robot, as depicted in Fig. 6.8a). This setup aims to illustrate an additional application of *ProTac* for newly developed robot arms, complementing its deployment in existing commercial robots as demonstrated in previous tasks. It's noteworthy that, in this study, only the

Table 6.2: Control parameters for *ProTac*-driven safety controllers

Parameter		Value	Unit
Virtual inertia matrix	$\boldsymbol{M}_v$	$\mathrm{diag}(1.0, 1.0, 1.0)$	-
Virtual damping matrix	$\boldsymbol{D}_v$	$\mathrm{diag}(1.5, 1.5, 1.5)$	-
Virtual stiffness matrix	$\boldsymbol{K}_v$	$\mathrm{diag}(2.0, 2.0, 2.0)$	-
Max. virtual force	$f_v^{\max}$	0.45	N
Shape factor	γ	6.0	-
Proximal threshold	ρ	0.065	m
Max. time-scaling value	$k^{\max}$	2.0	-
Time-scaling shape factor	γ'	20.0	-

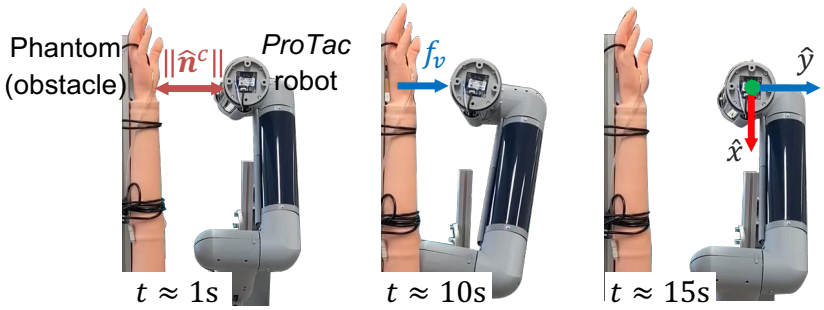
forearm was active for showcasing the demonstrations of the *ProTac*-driven control strategies. The *ProTac* links were interconnected via revolute joints actuated by electric motors (Dynamixel-P series, Robotis). In this system, coordination among control strategies, the *ProTac* sensing interface, and motor control were facilitated through ROS (robot operating system). The commanded joint velocities $\dot{\boldsymbol{\theta}}_d$, derived from the control laws, were regulated by the embedded motors' built-in motion controller. The parameters of the utilized controllers are detailed in Table 6.2.

6.4.2.1 Setup

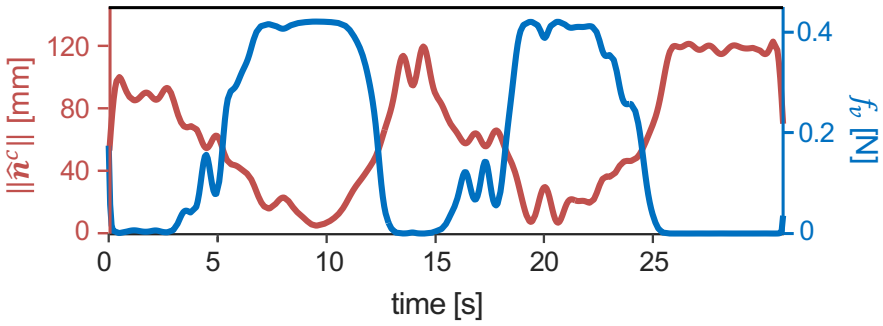
This experiment showcases how the *ProTac* link effectively facilitates the robot's reflexive response for obstacle avoidance, as outlined in Section 6.4. A phantom arm was moved back and forth in relation to the forearm *ProTac* link, prompting the movement of the *ProTac*-integrated robot in reaction to the estimated distance between the link and the phantom arm $\|\hat{\boldsymbol{n}}^c\|$ (refer to Fig. 6.8a).

6.4.2.2 Result

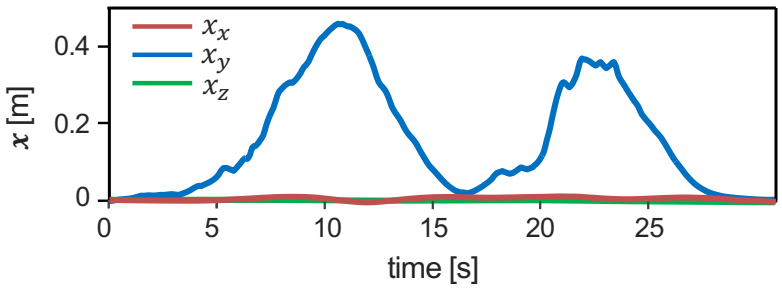
Figure 6.8b shows that the distance estimation $\|\hat{\boldsymbol{n}}^c\|$ led to variations in the virtual force f_v. Consequently, these variations influenced the robot's displacement along the $\hat{y}$-axis, as depicted in Fig. 6.8c, based on the admittance control law in Eq. (6.7). For example, as the phantom arm gradually approached the robot between $t \approx 4\,\mathrm{s}$ and $t \approx 10\,\mathrm{s}$, the magnitude of the virtual force f_v steadily increased, reaching approximately $f_v^{\max} = 0.45\,\mathrm{N}$. Consequently, the robot transitioned from its initial position $x_y = 0$ to a position approximately $0.4\,\mathrm{m}$ away from the obstacle,



(a) Experimental setup and rollout of *ProTac*-integrated robot's motion in response to the approaching phantom arm.



(b) Estimated distance (red line) and the resultant magnitude of virtual repulsive force (blue line).



(c) Robot displacement in response to the virtual force.

Figure 6.8: Demonstration of *ProTac*-driven reactive control. This scheme allows the avoidance of an approaching obstacle (c), which relies on the virtual repulsive force resultant from the *ProTac*-based distance estimation (b). The demonstration was performed using a custom-build robot integrated with *ProTac* links (a).

as observed in Figs. 6.8a and 6.8b. In contrast, around $t \approx 15\,\text{s}$, when the phantom arm retreated from the robot, the virtual force diminished ($f_v \approx 0$), allowing the robot to revert to its initial resting position.

6.5 Distance-based speed regulation

In this section, we present a time-scaling approach aimed at proactively modifying the robot's velocity in real-time, which relies on the magnitude of the estimated distance provided by *ProTac*, thereby facilitating effective speed adaptation without requiring a complete re-planning of a predetermined trajectory.

6.5.1 Problem formulation

Given a desired trajectory in Cartesian space $\boldsymbol{x}_d(s)$, where $s(\tau) : [0,1] \mapsto [0,1]$ is a time-scaling function parameterized by $\tau = t/T$, with $t \in [0,T]$ representing time and T denoting the base motion time of the trajectory, we compute the robot's velocity profile $\dot{\boldsymbol{x}}_d = [\dot{x}_x^d, \dot{x}_y^d, \dot{x}_z^d]^\intercal \in \mathbb{R}^3$. This velocity scales linearly with the base motion time T as:

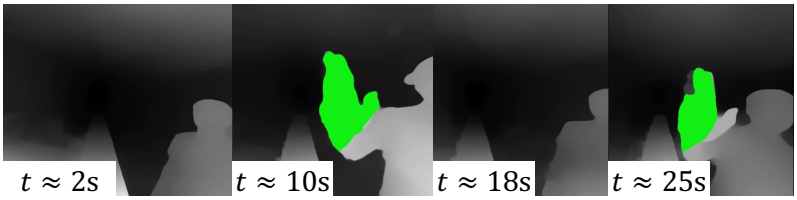
$$\dot{\boldsymbol{x}}_d = \frac{d\boldsymbol{x}_d}{ds} \cdot \frac{ds}{d\tau} \cdot \frac{1}{kT}. \tag{6.11}$$

Here, the time-scaling factor $k \geq 1 \in \mathbb{R}$ is introduced to adjust the robot velocity $\dot{\boldsymbol{x}}_d$ by scaling the motion time kT. To dynamically adjust the robot speed based on the estimated distance $\|\hat{\boldsymbol{n}}^c\|$, we compute the factor k from $\|\hat{\boldsymbol{n}}^c\|$ as:

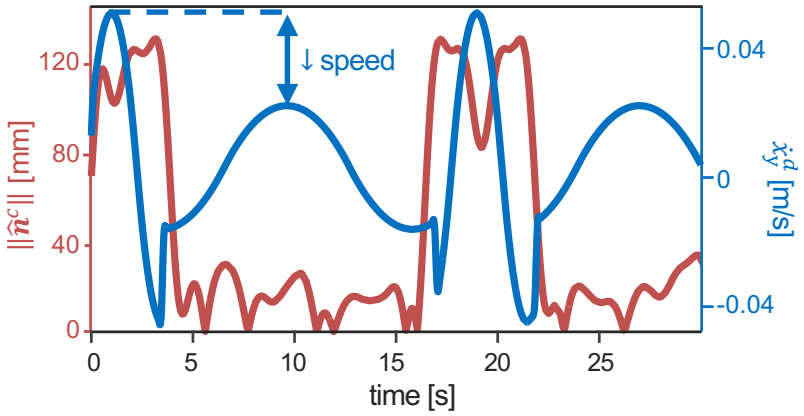
$$k = \frac{k^{\max} - 1}{1 + e^{(\|\hat{\boldsymbol{n}}^c\|(2/\rho)-1)\gamma'}} + 1. \tag{6.12}$$

This mapping function, akin to (6.8), aims to ensure motion smoothness in the presence of sensing noise, with a different shape factor γ' and an interval $[1, k^{\max}]$. The robot moves at the base speed over the duration T if the estimated distance to an obstacle $\|\hat{\boldsymbol{n}}^c\|$ exceeds ρ (*i.e.*, $\|\hat{\boldsymbol{n}}^c\| \geq \rho$), corresponding to $k = 1$. As the obstacle approaches, the speed gradually decreases with the increasing scale factor k, reaching $k^{\max}$ as $\|\hat{\boldsymbol{n}}^c\|$ approaches 0. Finally, with the desired Cartesian-space velocity $\dot{\boldsymbol{x}}_d$ computed using the resulting scale factor k in Eq. (6.11), the commanded joint velocities are computed via the pseudoinverse of the robot Jacobian as $\dot{\boldsymbol{\theta}}_d = \boldsymbol{J}_e^\dagger \dot{\boldsymbol{x}}_d$.

It's worth noting that while the estimated distance $\|\hat{\boldsymbol{n}}^c\|$ is utilized for problem formulations in this section, the risk score metric r can be employed in these formulas as well. This can be achieved by substituting $\|\hat{\boldsymbol{n}}^c\|$ with $1/r$ and adjusting predefined



(a) Rollout of ProTac-based depth images. The green-shaded area indicates a group of obstacle points with a distance below ρ. (Here, $\rho = 65\,\text{mm}$)



(b) Logs of *ProTac*-estimated distance (red line) and resultant Cartesian robot velocity along the $\hat{y}$-axis (blue line).

Figure 6.9: Demonstration of *ProTac*-driven speed regulation. The reduced robot speed is enabled based on the distance estimated between the *ProTac* link and approaching human.

parameters (such as γ, γ', ρ), ensuring the overall functionality of the proposed controllers remains intact.

6.5.2 Experiment and evaluation: Adaptive speed control

The efficiency of this task performance is validated with the same configuration and control pipeline as demonstrated in Section 6.4.2, where the *ProTac*-integrated robot arm is employed. The parameter values of the controllers are presented in Table 6.2.

6.5.2.1 Setup

This experiment demonstrates the efficacy of regulating robot speed based on the estimated distance $\|\hat{\boldsymbol{n}}^c\|$ provided by *ProTac* (see Section 6.5). In this scenario, a human approached the robot integrated with *ProTac*, which was moved back and forth periodically along a predetermined trajectory linearly aligned with the $\hat{y}$-axis.

Fig. 6.9a illustrates the depth images captured by *ProTac*, revealing the human approaching with their raising hand. Within these images, the green-shaded region highlights obstacle points with distance below the predefined threshold ρ.

6.5.2.2 Result

As depicted in Fig. 6.9b, around $t \approx 10\,\mathrm{s}$ and $t \approx 25\,\mathrm{s}$, the planned velocity $\dot{x}_y^d$ of the robot along the $\hat{y}$-axis was proportionally scaled by approximately $k^{\max}$ times (here, $k^{\max} = 2$), reaching a peak of approximately $0.025\,\mathrm{m/s}$, coinciding with the approach of the human hand or when the estimated distance $\|\hat{\boldsymbol{n}}^c\|$ neared the forearm *ProTac* link. Conversely, when the human remained beyond the predefined distance ρ, the robot reverted to its initial speed, peaking at approximately $0.05\,\mathrm{m/s}$, as illustrated in Fig. 6.9b.

Thus, the system not only underscores the effectiveness of integrating the *ProTac* link for safety controls but also highlights the viability of a next-generation robot arm equipped with soft proximity-tactile sensing. Such integration holds promise for enhancing safety in human-robot interaction scenarios.

Chapter 7

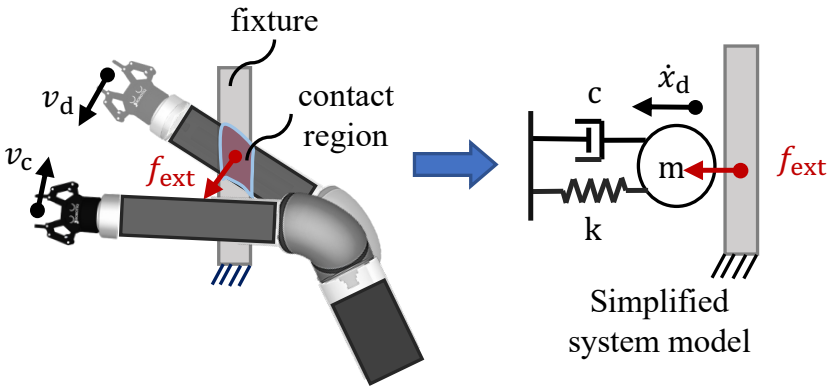
Tactile-driven Control Task

This chapter clarifies the safety mechanism employing embodied soft tactile sensing and discusses potential applications of the large-area sensor for tactile-driven tasks. In Section 7.1, we validate the effectiveness of the soft tactile link (TacLink) in mitigating physical impacts and facilitating reactive controls to handle unexpected collisions. Additionally, we demonstrate the utilization of tactile sensing for two contact-based tasks, namely whole-arm nonprehensile manipulation (see Section 7.2), and intuitive motion guidance (Section 7.3), where the TacLink is integrated with a custom-built robot arm. The evaluation of the proposed strategies is presented in the respective sections.

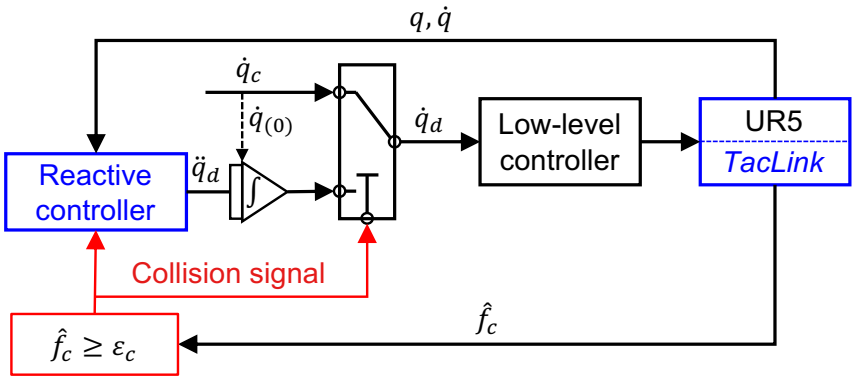
7.1 Safety mechanism with soft tactile sensing

Assessing the efficacy of embodied soft tactile sensing in mitigating impacts of unforeseen collisions, particularly through reactive responses, has the potential to enhance safety in human-robot interaction scenarios. Therefore, building upon the developed soft TacLink platform, this section aims to clarify the following points:

- Examining the performance of the TacLink in facilitating reactive responses to physical collisions, characterized by response time and peak impact force. The obtained results can establish a benchmark for assessing the effectiveness of soft tactile-sensitive skins in collision-handling tasks.
- Conducting a comparative analysis to demonstrate how the softness of the TacLink impacts reactive control and other collision responses, in comparison to those observed with a traditional *rigid* robot link. The outcomes are anticipated to inform the development of new safety standards for soft skin-based collaborative robots in human-robot interaction environments.



(a) Conceptual implementation of the collision reactive control



(b) Block diagram of the collision reactive controller

Figure 7.1: A kinematics control scheme allowing a robot with tactile sensing link to respond to a physical impact safely.

7.1.1 Collision response strategy

Problem: *The goal is to address situations where a robot equipped with TacLink collides with an unexpected obstacle at a certain velocity, as illustrated in Fig. 7.1. Thus, the objective is to enable the robot to be aware of the collision and respond to that collision by reversing its velocity direction to escape from the collision area.*

To meet our control objectives, we implement a kinematics admittance controller [101], which models the robot system as a mass-spring-damper system (refer to Fig. 7.1a) with virtual inertia, damping, and stiffness components. Specifically, this reactive controller aims to accelerate the robot in a manner that enables it to respond effectively to the contact force detected by the TacLink sensor. Thus, considering a 6-degree-of-freedom (DOF) robot arm with joint positions represented

by $\mathbf{q} \in \mathbb{R}^6$, the admittance control law can be formulated as:

$$\ddot{\mathbf{q}}_d = \mathbf{M}^{-1}(\hat{\boldsymbol{\tau}}_c - \mathbf{C}\dot{\mathbf{q}} - \mathbf{K}\mathbf{q}), \tag{7.1}$$

where $\mathbf{M} \in \mathbb{R}^{6\times6}$ denotes the positive-definite diagonal virtual inertia matrix, $\mathbf{C} \in \mathbb{R}^{6\times6}$ is the positive-definite diagonal virtual damping matrix, and $\mathbf{K} \in \mathbb{R}^{6\times6}$ is the positive-definite diagonal virtual rotational stiffness matrix. Also, $\hat{\boldsymbol{\tau}}_c$ represents the resulting external torque, computed as:

$$\hat{\boldsymbol{\tau}}_c = \mathbf{J}_c^{\mathrm{T}}\hat{\mathbf{F}}_c. \tag{7.2}$$

Here, $\mathbf{J}_c \in \mathbb{R}^{6\times6}$ denotes the Jacobian matrix at the contact point $\mathbf{x}_c \in \mathbb{R}^3$, while $\hat{\mathbf{F}}_c \in \mathbb{R}^6$ signifies the generalized external contact force relative to a coordinate system {C} (refer to Fig. 7.2).

7.1.2 System integration and implementation

While the described control scheme for handling collisions can be adapted for various tactile sensing skins, our specific implementation focuses on a robot system that utilizes the soft TacLink as an extended sensorized link for a commercial robot arm (see Figure 7.2 for the setup); thus, our examination is restricted to scenarios where contacts occur on the extended link.

As the TacLink sensor predominantly measures the contact depth $\hat{d}_c := \|\hat{\boldsymbol{d}}^c\|$ in the direction normal to the skin surface (refer to Eq. 4.20), the resultant generalized contact force simplifies to $\hat{\mathbf{F}}_c = [0,\ 0,\ \hat{f}_c,\ 0,\ 0,\ 0]^{\mathrm{T}}$, where $\hat{f}_c$ signifies the estimated contact force along the z axis. Furthermore, the mapping of contact depth to the equivalent contact force $\hat{f}_c$ is necessary to enable the reactive control law (see Eq. 7.1). This conversion is achieved by modeling the soft skin at the contact point as an elastic spring element α_f

$$\hat{f}_c = \alpha_f \cdot \hat{d}_c. \tag{7.3}$$

The calibration process for the stiffness constant α_f and the accuracy of calibrated contact forces across different TacLink regions are detailed in Appendix C.

Furthermore, the system operates in two distinct phases: the *before* and *after* collision states, as illustrated in Figure 7.1b. Before a collision event, the robot's motion is regulated based on a predefined joint velocity reference $\dot{\mathbf{q}}_c$. Upon collision detection, the system transitions to a collision-reactive control scheme, initiating the robot's response through the computation of the control law $\ddot{\mathbf{q}}_d$ (Equation 7.1), yielding reactive joint velocities $\dot{\mathbf{q}}_d$ via temporal integration. Consequently, the overall control system can be expressed as:

$$\dot{\mathbf{q}}_d = \begin{cases} \int_{t_0}^{t} \ddot{\mathbf{q}}_d \, dt + \dot{\mathbf{q}}_c(t_0), & \text{if } \hat{f}_c \geq \epsilon_c \\ \dot{\mathbf{q}}_c, & \text{otherwise,} \end{cases} \tag{7.4}$$

where t_0 denotes the instance of collision occurrence, signified by the estimated contact force $\hat{f}_c$ surpassing a predefined threshold ϵ_c. This threshold is determined based on the hysteresis property of the TacLink sensor (refer to Appendix C). Ultimately, the resultant joint velocity $\dot{\mathbf{q}}_d$ is governed by the robot's low-level controller. Here, the controllers have access to the robot's positional and velocity joint states ($\mathbf{q}$, $\dot{\mathbf{q}}$) via the built-in controller. A comprehensive evaluation of the system's efficacy and the controller's performance in collision handling is reported in Section 7.1.3.

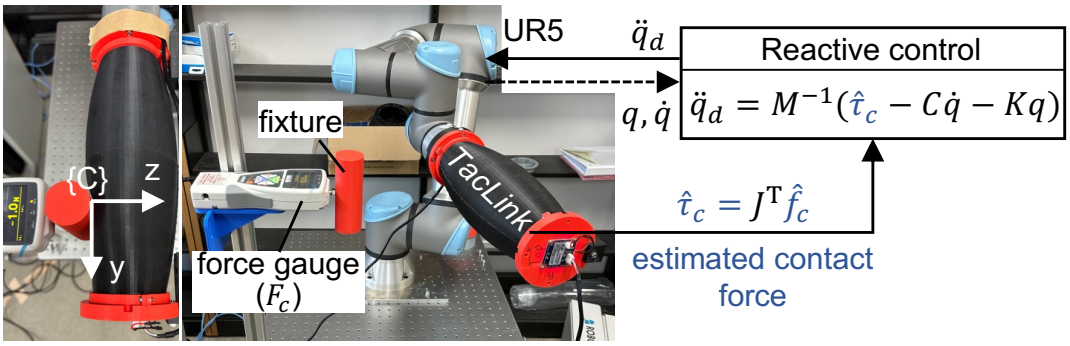


Figure 7.2: System integration of the vision-based tactile link and UR5e robot arm and setup schemes for the sensing and collision experiments.

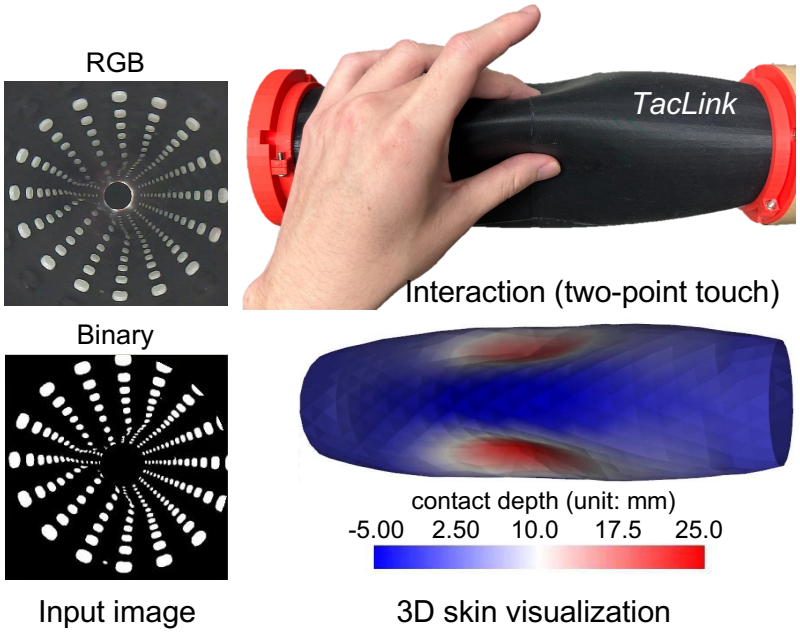


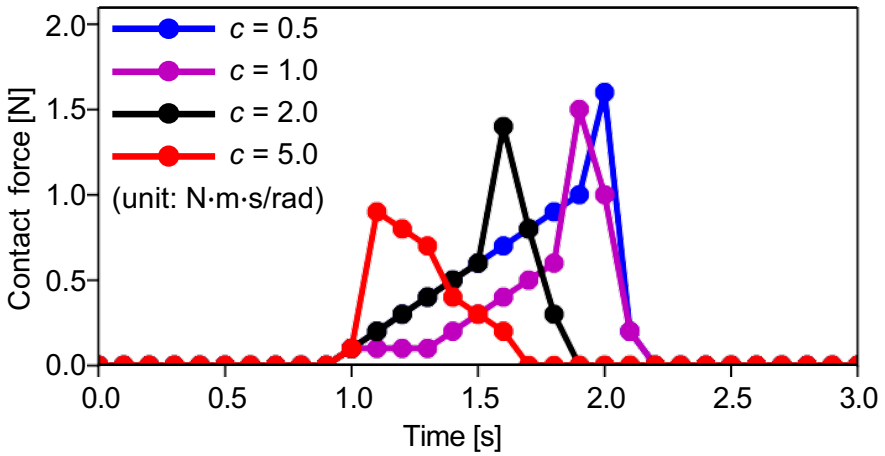
Figure 7.3: Visualization of tactile sensing, by which the skin shape under deformation can be constructed from the input image.

7.1.3 Experiment: Characterization of collision responses

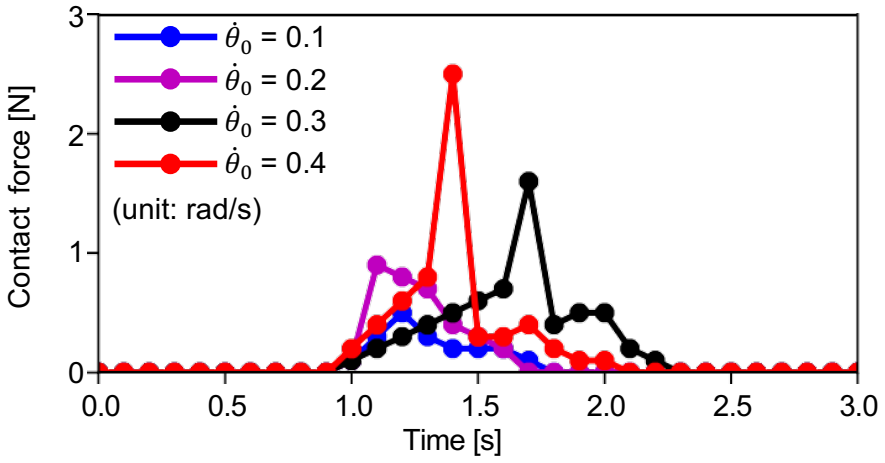
The experimental setup employed to assess the effectiveness of the proposed control system is depicted in Figure 7.2. Operating on a Ubuntu PC with GPU acceleration, the system facilitated low-latency vision-based tactile sensing, achieving a frequency of approximately 100 Hz. The operation of the TacLink sensor is illustrated in Fig. 7.3. Furthermore, the control infrastructure was established using ROS (Robot Operating System), leveraging the *ros-control* framework for the implementation of low-level velocity regulation. In order to establish communication with the UR5 robot, an official driver package tailored for the UR5 controller interface was utilized.

7.1.3.1 Settings

This section demonstrates the effectiveness of the robot integrated with the TacLink sensor in responding to unforeseen collisions, employing the reactive control strategy described in Section 7.1.1. The assessment of the controller's performance was conducted across various control parameters and initial robot velocities $\dot{\theta}_0$. To simplify the analysis, experiments were conducted on the movement of the robot's base joint



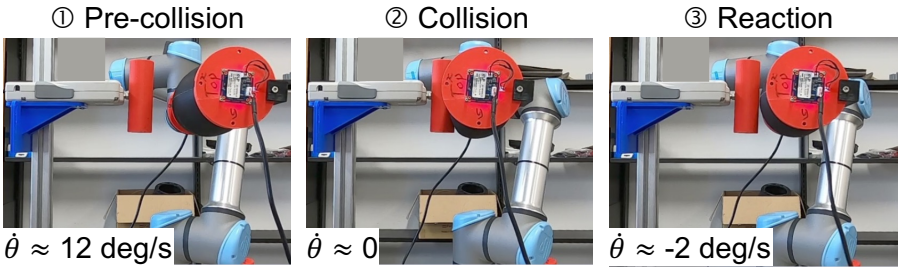
(a) Contact force with the variance of rotational damping coefficient



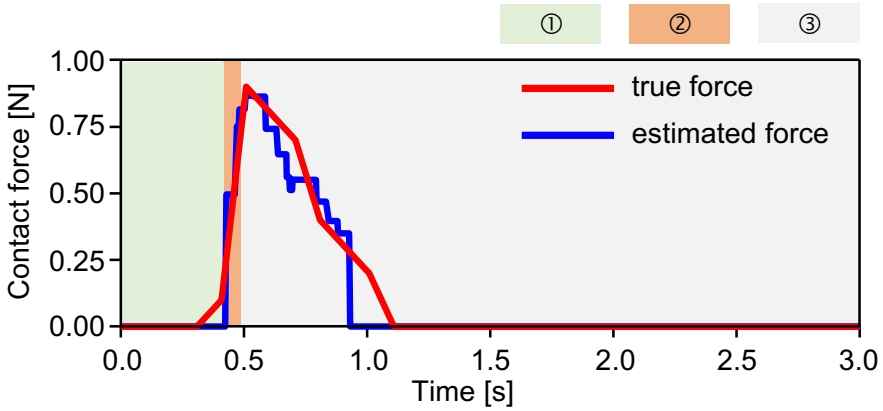
(b) Contact force with the variance of initial (pre-contact) joint speed

Figure 7.4: The robot's behavior with different control parameters of the proposed reactive controller.

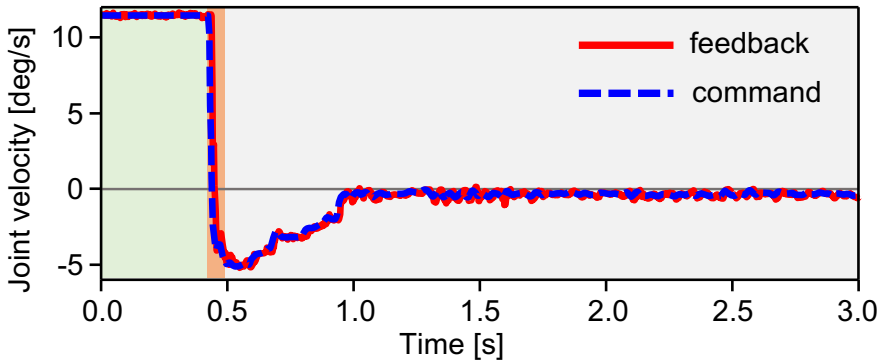
(joint 1), with the robot's configuration defined as shown in Figure 7.2. Notably, the moment arm of the contact force at the fixture was set to $l_f = 0.5\,\mathrm{m}$, and the matrices of controller parameters were set to $\mathbf{M} = \mathrm{diag}(0.15, [\mathbf{0}]_5^{\mathrm{T}})$, $\mathbf{C} = \mathrm{diag}(c, [\mathbf{0}]_5^{\mathrm{T}})$, with intentional assignment of zeros to $\mathbf{K}$ to eliminate the necessity of specifying a reference resting position. Evaluation of the robot response's performance centered on the determination of the *peak impact force* during the collision, *recovery duration* from collision onset until complete dissipation of contact force, and *reactive duration* measured from collision onset to the initiation of controller response. Smaller values in these metrics denote better controller performance. Here, we limit the



(a) Video stills capturing robot response in collision experiment



(b) True and estimated contact force during the collision



(c) Command joint velocity and its feedback

Figure 7.5: The behavior of robot with the application of collision reaction strategy over time.

maximum linear velocity of the robot at the contact point to 0.2 m/s, which complies with ISO/TS 15066 standard [102] to guarantee safe human-robot collaborative operations in the Power and Force Limiting setting.

7.1.3.2 Results

The investigation into the robot's response to collisions under various controller parameters is highlighted in Figure 7.4. As depicted in Figure 7.4a, both the peak impact force and recovery duration exhibited a reduction with an increase in the rotational damping coefficient c, given a constant initial speed of $\dot{\theta}_0 = 0.2\,\text{rad/s}$. This observation can be explained by the fact that a higher damping coefficient generates a more substantial resistant force opposing the robot's collision direction, thereby mitigating both the peak contact force and recovery duration. Figure 7.4b illustrates the robot's responses to collisions at various pre-collision speeds $\dot{\theta}_0$, with a fixed damping coefficient of $c = 5$. Additionally, Figure 7.5 presents the robot's motion and behavior across three phases: 1) pre-collision, 2) collision, and 3) recovery/reactive phase. Indeed, the correspondence between the estimated contact force and true force values (refer to Figure 7.5b) validates the efficacy of the integrated *soft* sensing system for safety control tasks, despite some observed hysteresis in the sensing signal, particularly for minor skin deformations, as discussed in Appendix C. Furthermore, Figure 7.5c illustrates the commanded base joint velocity derived from the desired joint acceleration $\ddot{q}_d$ computed via the proposed control law, demonstrating the controller's attempts to move the robot away from the collision region by reversing its direction of motion.

Init. speed (rad/s)	*Reactive time* (ms)	*Recovery time* (ms)	*Peak impact force* (N)
$\dot{\theta}_0 = 0.1$	62.0 ± 16.2	820.0 ± 40.0	1.00 ± 0.09
$\dot{\theta}_0 = 0.2$	54.0 ± 29.9	860.0 ± 49.0	1.14 ± 0.16
$\dot{\theta}_0 = 0.3$	72.0 ± 21.4	780.0 ± 40.0	1.42 ± 0.26
$\dot{\theta}_0 = 0.4$	64.8 ± 17.3	640.0 ± 49.0	1.66 ± 0.54

Table 7.1: Characterization of soft reactive response: response times and peak impact force

Table 7.1 presents the characterization of the *soft* reactive response for different initial velocities (with $c = 5$). The findings underscore a reactive duration, recovery duration, and peak impact force of less than 80 ms, 900 ms, and 2.5 N, respectively, for initial velocities up to 0.4 rad/s; which is equivalent to an impact linear velocity of 0.2 m/s. The delay in reactive time can be ascribed to the hysteresis of the TacLink

sensor.

In short, the findings suggest that the TacLink-integrated robot can responsively move away from the collision area within a timeframe of roughly 900 *ms, while containing the maximum contact force below* 2.5 *N, even under an impact velocity of* 0.2 *m/s. Notably, this level of contact force falls well below the hazardous threshold for humans [101]. Lastly, this characterization serves as a benchmark for evaluating the efficacy of other soft tactile-sensitive skins in handling collisions with reactive control.*

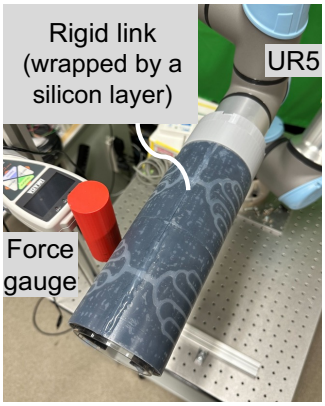
7.1.4 Comparative study

7.1.4.1 Settings

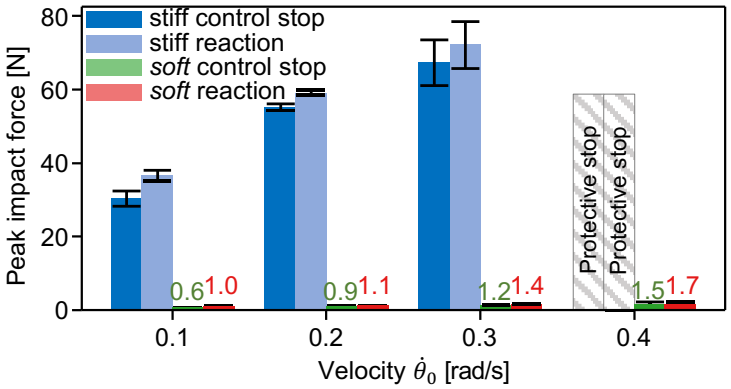
This section aims to validate the efficacy of the soft tactile link in mitigating significant impact forces in comparison to a rigid link. The rigid link, made from an acrylic pipe, was affixed to the end-effector of the UR5, matching the soft *TacLink* in size and weight. Additionally, the rigid link was encased in a silicone rubber pad (refer to Figure 7.6a) to protect it from severe collision impacts. To estimate the external contact torque in the scenarios of stiff-link collisions which lack direct feedback from the tactile sensor, we employed the $\hat{\tau}$-observer method [101]. This method relies on simplified joint-space dynamics of the robot base, along with inherent joint torque feedback provided by the UR5's low-level controller. Subsequently, collision experiments were conducted to compare the peak impact force and the robot's response between the rigid link and the soft tactile link, by employing two collision handling strategies:

1. (Stiff/Soft) Control stop: immediately stops the robot's motion upon contact detection.
2. (Stiff/Soft) Reactive control: the method described in Section 7.1.1.

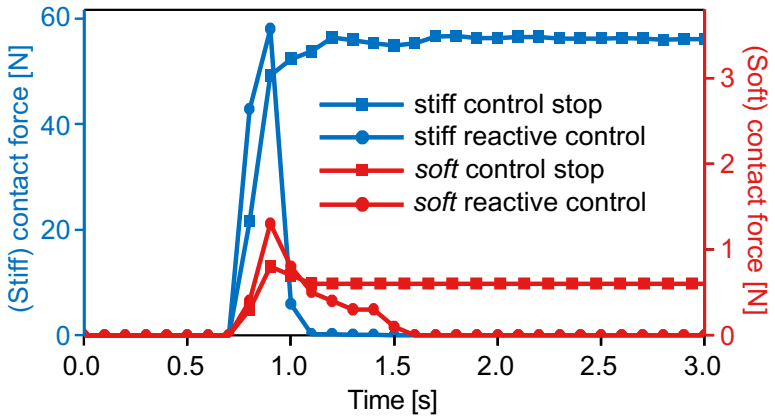
The (stiff/soft) prefix denotes the type of link with which the controller is tested. These collision experiments were conducted at various pre-contact speeds $\dot{\theta}_0$, with five trials executed for each velocity.



(a) *Stiff* collision setup



(b) Comparison of peak impact force



(c) Comparison of transient responses ($\dot{\theta}_0 = 0.2\,\text{rad/s}$)

Figure 7.6: Comparison of collision handling performance between a *stiff* and tactile-enabled *soft* link with two different control strategies. The results displayed in (b)-(c) demonstrate the effectiveness of leveraging the soft mechanism with tactile sensing to facilitate reactive control and contact responses. It is observed that the utilization of the soft *TacLink* significantly mitigates peak impact forces. At $\dot{\theta}_0 = 0.4\,\text{rad/s}$ (b), the UR5's built-in controller triggers the *protective stop* signal in the trials of stiff collisions to halt the robot's motion (the observed peak impact forces are not reported for this case).

7.1.4.2 Results

Figure 7.6 provides an overview of the collision response outcomes. As illustrated in Figure 7.6b, collisions involving the stiff link exhibited significantly higher peak impact forces compared to those involving the soft TacLink, irrespective of the applied control strategies. Notably, at $\dot{\theta}_0 = 0.4\,\text{rad/s}$, the UR5's built-in controller activated the *protective stop* signal in response to extreme impact forces detected in stiff link collisions, while the soft link consistently maintained lower impact forces,

more or less 2 N, across varying velocities (refer to Figure 7.6b). Furthermore, although the stiff reactive control showcased comparable transient responses to its soft counterpart, dissipating contact forces over time, it yielded notably higher peak forces compared to the soft reactive control (see Fig. 7.6c). For instance, at $\dot{\theta}_0 = 0.2\,\text{rad/s}$, the average peak impact force induced by the soft reactive control was approximately 1.1 N, nearly 54 times smaller than that caused by the stiff reactive controller.

These findings confirm the efficacy of employing TacLink *in mitigating peak impact forces, particularly with reactive control. This suggests the potential for the development of safer robotic arms constructed entirely from soft tactile links.*

7.2 Nonprehensile manipulation by whole-arm pushing

7.2.1 Method

This section demonstrates the utilization of a robot arm system equipped with the TacLink to manipulate an object towards a goal by *pushing*, where the contact information provided by the TacLink is exploited to guide the robot's motion. To simplify the task, we confined the object manipulation to a $\hat{y}_s - \hat{z}_s$ plan of the frame {s}. Here, the system received feedback on the 3D position of the pushed object $\mathbf{x}_{\text{object}} \in \mathbb{R}^3$ detected through *contact* with TacLink. This information is utilized to compute the desired spatial velocity ${}^c\mathcal{V}_d \in \mathbb{R}^6$ relative to the contact frame {c}, guiding the object towards the predefined goal, based on the typical impedance strategy. The contact frame {c}, represented by the rotation matrix $\mathbf{R}_c$, was defined with its origin at the contact location $\hat{\boldsymbol{x}}^c$ (derived from Eq. 4.19), with its $\hat{y}_c$- and $\hat{z}_c$-axes aligned along the outward normal of the contact plane and the z-axis of the TacLink frame, respectively, while the $\hat{x}_c$-axis completed the right-hand rule. By considering the object position ($\mathbf{x}_{\text{object}} \equiv \hat{\boldsymbol{x}}^c$) and the goal location $\mathbf{x}_{\text{goal}} \in \mathbb{R}^3$, the pushing task was executed to ensure that the pushing direction $\mathbf{n}_{\text{push}}$ remained perpendicular to the contact plane ($\hat{y}_c \equiv \mathbf{n}_{\text{push}}$)

$$\mathbf{n}_{\text{push}} := \frac{\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{object}}}{\|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{object}}\|}. \tag{7.5}$$

Hence, the required angular velocity $\boldsymbol{\omega}_d \in \mathbb{R}^3$ to align the robot arm with the desired pushing direction can be determined as

$$\boldsymbol{\omega}_d = k_\omega \hat{\omega}_d \bar{\theta}, \tag{7.6}$$

where $k_\omega > 0$ represents the proportional gain of angular velocity, and $\hat{\omega}_d\bar{\theta} \in \mathbb{R}^3$ signifies the exponential coordinates of a rotation matrix $\bar{\mathbf{R}} := \text{Rot}(\hat{\omega}_d, \bar{\theta}) = \mathbf{R}_{\text{push}}\mathbf{R}_c^{\text{T}} \in SO(3)$; where $\mathbf{R}_{\text{push}} := [\hat{x}_s,\ \mathbf{n}_{\text{push}},\ \hat{x}_s \times \mathbf{n}_{\text{push}}]$, rotating the contact frame {c} towards the pushing direction. Furthermore, for pushing toward the target goal, the commanded linear velocity is determined as

$$\mathbf{v}_d = k_v(\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{object}}), \tag{7.7}$$

In addition, adhering to a typical safe human-robot interaction scenario, we imposed a condition on the proposed pushing control to halt robot motion in case an unintended external contact occurred, that is, a contact different from the one established with the pushed object. Thus, assuming a contact always exists with the target object,

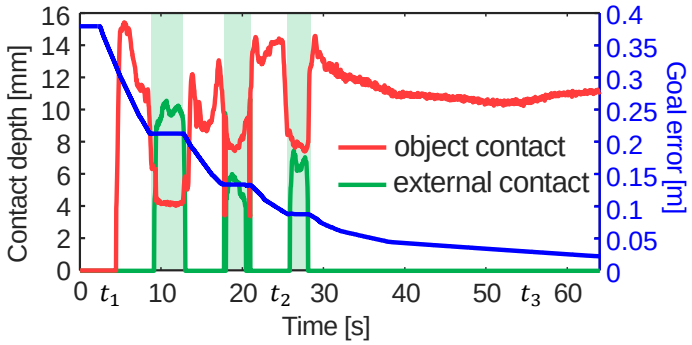
$${}^c\mathcal{V}_d = \begin{cases} [\mathbf{0}]_{6\times 1}, & \text{if } L \geq 2 \\ \mathbf{R}_c^{\text{T}}[\boldsymbol{\omega}_d,\ \mathbf{v}_d]^{\text{T}}, & \text{otherwise.} \end{cases} \tag{7.8}$$

Ultimately, the desired twist ${}^c\mathcal{V}_d$ was converted to commanded joint velocity $\dot{\boldsymbol{\theta}} \in \mathbb{R}^3$ through the Jacobian ${}^c\mathbf{J} \in \mathbb{R}^{6\times 3}$ at the contact point.

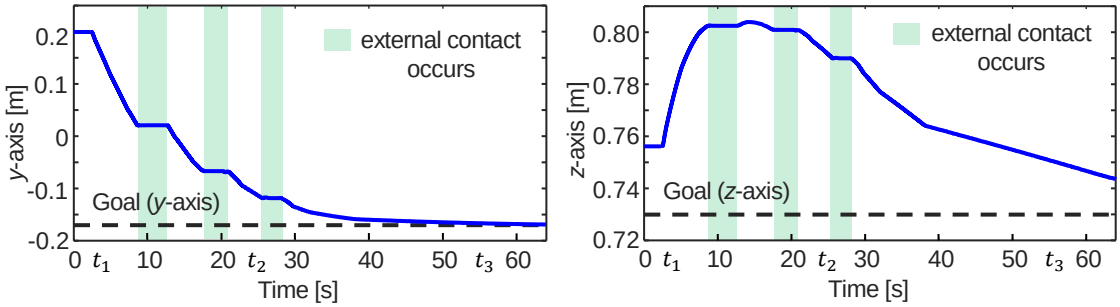
7.2.2 Experiment

The resultant behaviors of the proposed contact-based pushing strategy are depicted in Figure 7.7, with the designated goal position set at $\mathbf{x}_{\text{goal}} = [-0.01,\ -0.17,\ 0.73]^{\text{T}}$, and the proportional gains calibrated experimentally as $k_v = 0.12$ and $k_w = 0.35$. Throughout the pushing trial, the primary contact with the object (a water-filled bottle) maintained a relatively consistent contact intensity of approximately 14 mm, except when an unplanned contact occurred (see Fig. 7.7b). Upon a sudden human touch triggering a secondary (external) contact phase, all robot motion

(a) Video stills of contact-based object pushing experiment with the possible occurrence of two-point contact.



(b) Time-log of contact depth and positional error of the contacted object w.r.t the goal location.



(c) Time-log of the contacted object position measured along y- and z- axes of the space frame {s}. Goal lines indicate the positional references.

Figure 7.7: The experiment of contact-based object pushing. An object, whose position is identified through contact with the TacLink, is guided to a goal location $\mathbf{x}_{\text{goal}} = [-0.01, -0.17, 0.73]^{\text{T}}$ on a y-z plane of a table via pushing. When unexpected contacts (external contacts) occurred, the robot motion was temporarily halted, then resumed after the external contact broke. The observations of the external contacts are green-shaded. The demonstration can be found in the video https://youtu.be/NN2u8YBLITY.

stopped, leading to the object position remained unchanged (see Fig 7.7c). As time progressed, the pushed object gradually approached the preset goal, a process spanning approximately 60 s in total. However, a minor degree of settling error along the z-direction persisted, resulting in a goal error of roughly 0.05 m. Addressing this may require further improvements with a more advanced control policy.

Table 7.2: Control parameters for pushing and intuitive motion controllers

Parameter		Value	Unit
P-gain (angular velocity)	k_ω	0.35	s^{-1}
P-gain (linear velocity)	k_v	0.12	s^{-1}
Goal location	$\mathbf{x}_{\text{goal}}$	$[-0.01,\ -0.17,\ 0.73]^{\mathrm{T}}$	m
Virtual pivot point	${}^b\mathbf{r}_c$	$[0,\ 0,\ -0.13]^{\mathrm{T}}$	m
Linear velocity scale	k_d^v	1.20	s^{-1}
Angular velocity scale	k_d^ω	0.15	s^{-1}
Stroke distance threshold	ϵ_s	8	mm

7.3 Tactile-driven intuitive motion guidance

7.3.1 Method

This section showcases the deployment of TacLink as a haptic interface device for intuitively guiding the motion of the robot arm (see Fig. 7.8). Here, we strategically translate different tactile actions, encompassing single/multi-point push and stroke, into a desired robot twist ${}^b\mathcal{V}_d \in \mathbb{R}^6$. For single-point push actions occurring at the contact location $\mathbf{x}^c$ (Eq. 4.19), we encode the estimated contact depth vector $\hat{\boldsymbol{d}}_1^c$ (Eq. 4.20) and the normal direction $\mathbf{n}(\hat{\boldsymbol{x}}_1^c) := \mathbf{N}_{i_1^*}$ (Eq. 4.14) to the spatial velocity on the $\hat{x}_b - \hat{y}_b$ plane of the end-effector frame $\{\mathrm{b}\}$ as

$$[v_x,\ v_y,\ v_z]^{\mathrm{T}} = k_d^v \|\hat{\boldsymbol{d}}_1^c\| \mathbf{n}(\hat{\boldsymbol{x}}_1^c), \tag{7.9}$$

where k_d^v is a constant used to appropriately scale the resulting linear velocity. Furthermore, we employ distinguishable two-point contact as an interface for directing rotational motion, where a virtual pivot point ${}^b\mathbf{r}_c$ is positioned at the center of TacLink. Consequently, the rotational motion around the axes of the $\{\mathrm{b}\}$ frame can be mediated by

$$[w_x,\ w_y,\ w_z]^{\mathrm{T}} = k_d^v [{}^b\mathbf{r}_c \times \|\hat{\boldsymbol{d}}_1^c\| \mathbf{n}(\mathbf{x}_1^c) + {}^b\,\mathbf{r}_c \times \|\hat{\boldsymbol{d}}_2^c\| \mathbf{n}(\mathbf{x}_2^c)]. \tag{7.10}$$

Since we restricted the push direction to the normal of a contact plane for simplicity, we disregarded rotation/twist around the z-axis ($w_z = 0$), as well as linear motion ($v_z = 0$). However, linear velocity along the z-direction could be induced by

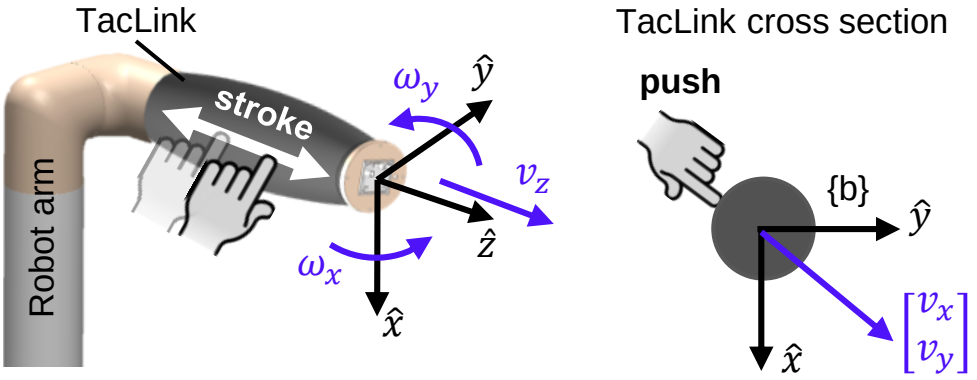


Figure 7.8: Conceptual illustration for tactile-based motion guidance.

detecting the stroke action. To ensure robust stroke detection, we introduced a fixed time window $T_w = W.\Delta t$ (where W is the window size), during which possible sliding motion on the skin surface is evaluated at a determined interval Δt. Therefore, at each time step Δt in the time window T_w, we measured the distance between the current contact position $\mathbf{x}^c_{k\Delta t}$ and the previous one $\mathbf{x}^c_{(k-1)\Delta t}$ as

$$\Delta x_k = \| \mathbf{x}^c_{k\Delta t} - \mathbf{x}^c_{(k-1)\Delta t} \|, \quad \forall k \in \{1, 2, \cdots, W\}. \tag{7.11}$$

Let us denote $\mathcal{X} = \{\Delta x_k\}$, where $|\mathcal{X}|$ is the number of its elements, and $\mathcal{K} = \{\Delta x_k \mid \Delta x_k \geq \epsilon_s\}, \forall k \in \{1, 2, \cdots, W\}$; with ϵ_s serving as a distance threshold to ensure stroke classification accuracy (see Table 4.18a). Thus, the stroke action (SA) along the z-axis is identified by

$$\text{SA} = \begin{cases} 1, & \text{if } |\mathcal{K}| \geq \eta |\mathcal{X}| \\ 0, & \text{otherwise (classified as push action)} \end{cases}, \tag{7.12}$$

where η is a classification ratio experimentally set to 0.3. When a stroke occurs at $t \geq T_w$, the linear velocity along the z-axis is determined by

$$v_z = \text{sgn}(x^{c,z}_{t+\Delta t} - x^{c,z}_t) k^\omega_d \frac{\left\| \mathbf{x}^c_{t+\Delta t} - \mathbf{x}^c_t \right\|}{\Delta t}, \tag{7.13}$$

where $x^{c,z}_{t+\Delta t}$ and $x^{c,z}_t$ denote the z-coordinates of $\mathbf{x}^c_{t+\Delta t}$ and $\mathbf{x}^c_t$, respectively, and k^ω_d is a constant scaling factor for the angular velocity. Finally, the resulting desired twist ${}^b\mathcal{V}_d = [v_x,\ v_y,\ v_z,\ w_x,\ w_y,\ 0]^{\mathrm{T}}$ is mapped to commanded joint velocity $\dot{\boldsymbol{\theta}}$ through

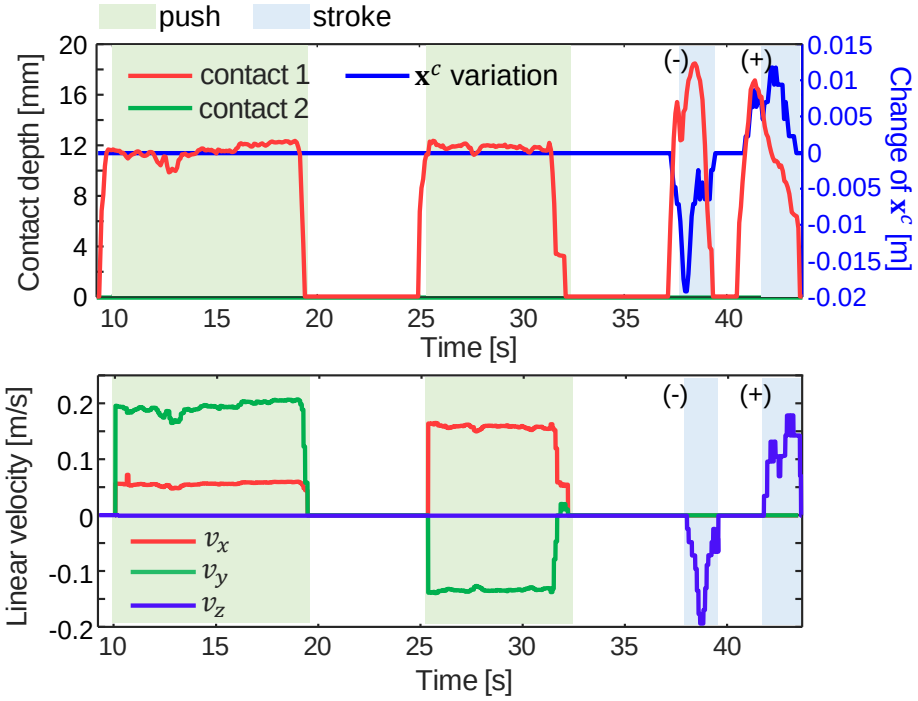


Figure 7.9: Time-log of contact depth and resulted robot linear velocity with respect to push and stroke contact action (shaded green and blue, respectively).

Table 7.3: Logs of estimated contact locations in the interaction experiments

Action	Time [s]	Contact location $\mathbf{x^c}$ [mm]					
		x_1	y_1	z_1	x_2	y_2	z_2
Push	28.00	-41	34	-127	-	-	-
Stroke	37.00	-22	-48	-154	-	-	-
Stroke	37.06	-20	-48	-174	-	-	-
Two-point	25.00	-2	-45	-224	3	45	-49
Two-point	85.00	-42	-3	-196	34	2	-64

*The time of two-point actions is referred to Fig. 7.10, while that of the other ones respecting to Fig. 7.9. (The contact position is expressed in the {b} frame)

the end-effector Jacobian ${}^b\mathbf{J}$.

7.3.2 Experiment

We conducted several experiments to validate the proposed motion guidance scheme, covering various contact actions and scenarios, with the controller's parameters detailed in Table 7.2. In the initial demonstration involving *push* and *stroke* actions, these actions were distinguished by unique patterns of contact depth, as depicted in Figure 7.9. Notably, strokes, typically executed by a human digit,

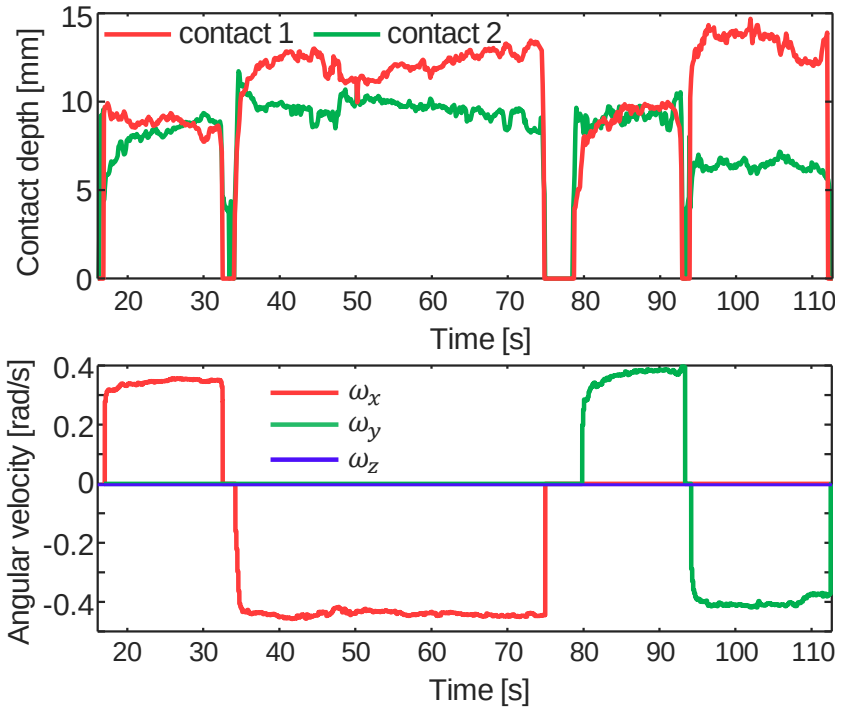


Figure 7.10: Time-log of contact depth and robot angular velocity resulted from two-point contact action at different contact locations.

induced sharp changes in the contact depth profile, whereas a push action exhibited stable intensity. The linear velocity along the z-axis resulting from stroke actions was directly proportional to the rate of contact positions over the time interval $\Delta t = 0.03\,\text{s}$. Conversely, robot motion along the other two axes was initiated by push actions, with the velocity resulting from the estimated contact depth and location (refer to Table 7.3 for the logs of estimated contact location). It is worth noting that the push/stroke classification, requiring a window size of $W = 8$, introduces a delay in the robot's response of at least 0.24 s, equivalent to the time window T_w. Furthermore, the delay may increase due to misclassifications between single and two-point contact scenarios. While this delay could be mitigated with more sophisticated classification algorithms (*e.g.*, machine learning techniques), it can allow users to feel safer during the interaction phases. Moreover, under the two-point contact scenario, as illustrated in Figure 7.10 and the corresponding pairs of contact positions (see Table 7.3), the robot exhibited rotation around either the $\hat{x}_b$ or $\hat{y}_b$ axes.

The demonstrated task performance, facilitated by our large-scale tactile device,

is anticipated to offer preliminary insights into the development of more comprehensive and efficient motion guidance strategies for robot learning and teaching. For video demonstrations of robot motion guidance and other applications of TacLink, please refer to the video available at the following link[1].

[1] https://youtu.be/NN2u8YBLITY

Chapter 8

Discussion and Conclusion

8.1 Discussion

8.1.1 ProTac design and fabrication

The *ProTac* link distinguishes itself from other large-scale vision-based tactile sensors by seamlessly integrating both proximity/visual perception and tactile sensing capabilities with a soft body. This functionality is enabled by the soft PDLC skin. In our design approach, the PDLC skin is shaped into cylindrical structures, using a multi-layered brace made from steel, and polylactic acid (PLA). While this configuration can enhance the link's payload capacity, it introduces a trade-off in the form of a "blind" spot corresponding to the area of the brace. Additionally, the relatively high stiffness imparted by the flexible PDLC film compromises the sensor's sensitivity compared to counterparts like TacLink [12]. Mitigating this drawback may necessitate the customization of PDLC films made from silicon material directly, rather than utilizing polyethylene terephthalate (PET) as in existing commercial products.

Moreover, the fabrication methodology proposed for the *ProTac* link, which relies on molding techniques, is anticipated to enable the production of links in various sizes and shapes. This can be achieved through customization of the outer molds and PDLC film. Consequently, the proposed process retains its versatility and effectiveness in fabricating a wide range of soft robot bodies equipped with *ProTac* sensing technology.

8.1.2 Tactile perception

In this dissertation, the contact intensity is assessed through the quantity of skin deformation because such information can reveal features of tactile perception (contact location, size of contact area, vibration, etc.) on large-sized tactile skin. Human mechanoreceptors cannot convey in detail how much force is acting on the skin. In addition, large-scale sensing is usually aimed at human-robot interactions rather than task-based ones, where information of force is deemed redundant. On top of that, toward a simulation framework for interactive robotics systems, TacNet was designed to be easily adaptable for different physical attributes, especially contact forces, other than the prediction of nodal displacements (skin deformation). In the future, for the physical formulation of interactive control problems, we aim to replace the current output signals of TacNet with nodal forces (which can be extracted from SOFA-based simulation), from which multi-contact forces and locations at a large-scale skin can be effectively inferred. In fact, contact force information ($\boldsymbol{\lambda}$ in Eq. 4.1) modeled from the SOFA kernel could be targeted to train TacNet models in which the same proposed sensing methods can be applied to extract high-level perception.

8.1.3 Proximity perception

The proximity perception of *ProTac* is facilitated by the core of a monocular depth-map estimation, given single image inputs in the transparent skin state. This methodology enhances the versatility of the *ProTac* sensing technology, making it applicable to various sensor configurations and designs, including those with single or multiple camera setups, with minimal adjustments required for fine-tuning and calibration. However, the transparency of the skin and the see-through effect still have impacts on the sensing performance and measurement range of *ProTac*. Therefore, refining the fine-tuning process, advancing DNN-based depth estimation, or integrating data from two cameras could potentially expand the measurement range of *ProTac*. Moreover, as illustrated in Section 6.3, human detection and segmentation can be accomplished using a commercially available DEVA model with *ProTac* perspectives, indicating the potential of *ProTac* for other visual perceptions

beyond distance measurement and risk assessment.

Last but not least, the *flickering* sensing mode suggests the possibility of simultaneous proximity-tactile sensing through advancements in signal processing or data-driven techniques, which leaves room for further developments in future works.

8.2 Conclusion

This dissertation presents a novel vision-based proximity-tactile sensing technology for soft robotic systems, named **ProTac**, accomplished by actively controlling the skin's transparency. The technology is demonstrated with a soft robotic link featuring proximity-tactile sensing capabilities. Compared to conventional tactile sensors of various electronic elements, our system provides large-area multimodal sensing with a simple setup and minimal impact on the mechanical properties of the soft skin (no embedded sensing elements), as well as offering no interference between the two modalities. The realization of the *ProTac* link with the desirable operational modes has successfully addressed Research Question 1 (**RQ1**).

The soft *ProTac* link offers the flexibility to operate independently in tactile or proximity modes, or concurrently in flickering mode. Perception capabilities are realized through sim-to-real learning techniques for tactile mode and monocular depth-map estimation for proximity mode. Notably, this study introduces SimTacLS, a pipeline tailored for simulating and training a vision-based tactile sensor, integrating soft contact mechanics of the skin. This pipeline serves as a valuable tool for generating tactile training data, and learning tactile perception in simulated environments, alleviating the need for labor-intensive experimental setups. Thus, in addressing **RQ2**, the findings demonstrate that tactile perceptions of physical devices, including contact depth and localization, can achieve satisfactory performance levels through the application of domain adaptation or domain randomization techniques at a reduced cost of experimental setups.

Additionally, in addressing **RQ3**, the dissertation presents the utilization of the multi-modal *ProTac* link with either a custom-built *ProTac* robot or a commercial

industrial robot arm to facilitate various robotic tasks involving safe human-robot interaction and motion control. Specifically, the incorporation of soft skin-based multimodal sensing with optimization control facilitates robot motion in cluttered environments, while limiting impact forces, the performance challenging to achieve with high-stiffness robot skins or rigid links. Moreover, *ProTac* enables seamless operation throughout different phases of a safe and efficient human-robot interaction scenario, alongside control strategies integrated with *ProTac* for distance-based collision avoidance and speed regulation. These findings substantiate the benefits of integrating softness and multimodal sensing to enhance robotic task performances.

Finally, the last chapter illustrates the efficacy of employing soft skin and vision-based tactile sensing for mitigating unexpected collisions, by minimizing peak impacts and recovering from contacts, particularly through a reactive control mechanism. The findings indicate that the robot integrated with TacLink can responsively move away from the impact zone within a duration of approximately 900 ms, while constraining the maximum contact force to below 2.5 N, even when subjected to an impact velocity of 0.2 m/s. This level of contact force falls well below the safety threshold for human interaction [101]. Moreover, the comparative analysis underscores the advantages of the highly soft sensing skin in mitigating significant impacts resulting from collisions and enabling controls that may pose challenges with rigid robot structures (**RQ3**).

8.3 Future work

My future work first focuses on the integration of the developed *soft* multi-modal sensing with thermal display technology [103]. This integration aims to equip whole-body and whole-arm robots with the capability to engage in affectionate and safe interactions with humans. Importantly, I aim to develop a novel perception and control framework centered on Large Language Models (LLMs). This framework leverages the contextual understanding of LLMs to discern the semantics of interactive actions through tactile perception and to identify potential risks based on proximity awareness, from which safe actions or haptic feedback can be coordinated

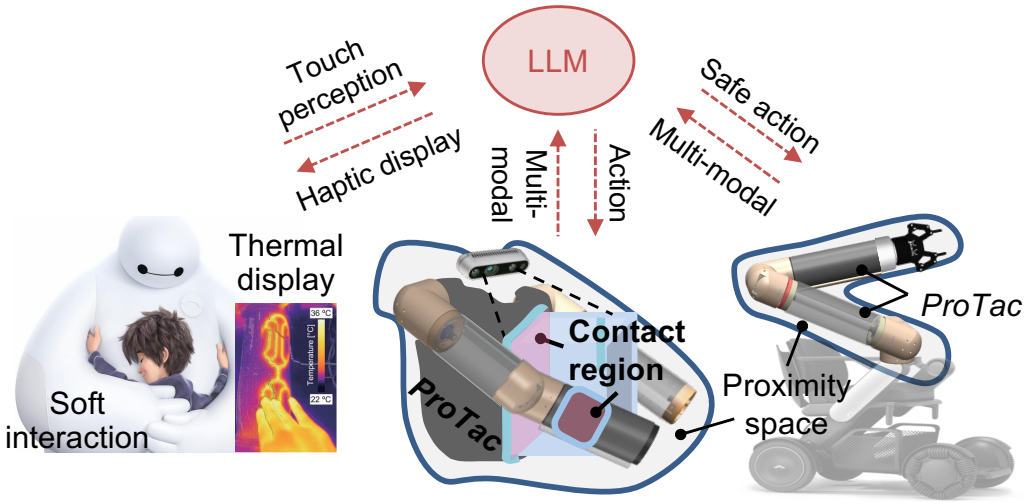


Figure 8.1: Illustration of a perception and control framework leveraging LLMs (large language models), built on our multi-modal *soft* sensing technology.

accordingly (see Fig. 8.1). For instance, a comfortable thermal sensation or affectionate actions can be presented based on the tactile actions perceived from humans. Additionally, I am exploring the use of LLMs to enable novel perception techniques, such as emotion detection driven by tactile feedback, leveraging our multi-modal soft sensing technology.

Another research avenue involves further exploration of novel embodied control strategies that harness the unique capabilities of our *soft* multi-modal sensing solutions. These strategies aim to facilitate control tasks that are difficult to achieve with rigid bodies. Furthermore, I aim to establish control strategies, along with new safety standards for our soft sensing devices in safety-critical scenarios.

Furthermore, my further research endeavors include studying neuromorphic processing for tactile perception, aiming to enhance sensing efficiency across various parts of robot bodies. The immediate objective is to establish a platform for large-scale robotic skins capable of conveying event-based signals, with event-based cameras serving as the foundation. This platform will facilitate further investigation into spatial-temporal encoding and processing techniques for diverse tactile stimuli and modalities.

Lastly, I foresee the use of our developed soft sensing technologies to enhance distant human-machine interaction. For example, doctors and medical robots can greatly benefit from such technologies during teleoperation, where the connection

between soft tactile sensing and haptic feedback is crucial. Such a connection may also find applications in AR systems, allowing users to explore and interact with surroundings via their tactile-driven avatars. In this regard, our proposed WoTT framework [104, 105] could serve as an enabler to facilitate haptic data communication over the Internet.

Appendix A

Contact model

A collision detection protocol [106] is implemented in SOFA to supervise physical contacts among simulated objects. Once a collision is well-detected, it is necessary to apply the corresponding contact responses to update the current state, such as the positions of the nodes. In SOFA, the collision consequences generally adhere to a combination of Signorini's frictionless contact law [81] and Coulomb's frictional law [82]. This is mathematically expressed by the complementary condition below:

$$\begin{cases} \text{In contact: } \boldsymbol{\Delta}_n = 0 \Rightarrow \boldsymbol{\lambda}_n > 0 \\ \text{No contact: } \boldsymbol{\Delta}_n > 0 \Rightarrow \boldsymbol{\lambda}_n = 0 \end{cases} \tag{A.1}$$

where $\boldsymbol{\Delta}_n$ and $\boldsymbol{\lambda}_n$ are the gap between two contact opponents and the contact force measured along normal direction n, respectively. This relation defines the contact occasion whenever $\boldsymbol{\Delta}_n = 0$ and $\boldsymbol{\lambda}_n > 0$ and vice versa. With regard to this law, friction force $\boldsymbol{\lambda}_t$ observed along tangential direction t is imposed to the friction cone which is dependent on friction coefficient μ and normal force Δ_n:

$$\begin{cases} \text{Stick: } \dot{\boldsymbol{\delta}}_t = \mathbf{0} \Rightarrow \|\boldsymbol{\lambda}_t\| < \mu \|\boldsymbol{\lambda}_n\| \\ \text{Slip: } \dot{\boldsymbol{\delta}}_t \neq \mathbf{0} \Rightarrow \boldsymbol{\lambda}_t = -\mu \|\boldsymbol{\lambda}_n\| \frac{\dot{\boldsymbol{\delta}}_t}{\|\dot{\boldsymbol{\delta}}_t\|} \end{cases}, \tag{A.2}$$

where $\dot{\boldsymbol{\delta}}$ is the tangential velocity. In this regard, the term $\mathbf{J}^{\mathbf{T}}\boldsymbol{\lambda}$ can be divided into normal contact element ${\mathbf{J}_n}^{\mathbf{T}}\boldsymbol{\lambda}_n$ and tangential (frictional) element ${\mathbf{J}_t}^{\mathbf{T}}\boldsymbol{\lambda}_t$. In order to describe the system state within the contact phase between points i and j for instance, the following process will be performed:

- *Step 1:* The dynamic equation of each object (Eq. 4.3) should be re-written as below:

$$\mathbf{A}_{i,j}\left(\mathbf{x}_{i,j}^{free} + \mathbf{x}_{i,j}^{cor}\right) = \mathbf{b}_{i,j} + dt\mathbf{J}_{i,j}^{T}\boldsymbol{\lambda}, \tag{A.3}$$

in which, $\mathbf{x}_{i,j}^{free}$ (or $\ddot{\mathbf{q}}_{i,j}^{free}$) is $free-constrained$ motions of object i and j with the assumption of no contact between them (*i.e.*, $\boldsymbol{\lambda} = \mathbf{0}$). Solving independently their motion equations yields $\ddot{\mathbf{q}}_{i,j}^{free}$ as well as corresponding positions $\mathbf{q}_i$ and $\mathbf{q}_j$ which will then be utilized to calculate the gap $\boldsymbol{\Delta}^{free}$ and its derivative in time $\dot{\boldsymbol{\Delta}}^{free}$:

$$\begin{aligned} \boldsymbol{\Delta}^{free} &= \boldsymbol{\xi}(\mathbf{q}_i{}^{free}) - \boldsymbol{\xi}(\mathbf{q}_i{}^{free}) \\ \Rightarrow \dot{\boldsymbol{\Delta}}^{free} &= \mathbf{J}_i \dot{\mathbf{q}}_i^{free} - \mathbf{J}_j \dot{\mathbf{q}}_j^{free} \end{aligned} \tag{A.4}$$

- *Step 2:* Whereas $\mathbf{x}_{i,j}^{cor}$ (or $d\dot{\mathbf{q}}_{i,j}^{cor}$) represents correction motions which are responsible for minimizing the interpenetration resulting from $free$ motions indicated in the previous step. By doing that, colliding objects (or contact nodes) are enforced to respect constraint laws (static contact and friction laws). In this regard, during the time interval of the contact phase, dynamic change of the actual deviation $\boldsymbol{\Delta}$ is expressed using the following linearization:

$$\dot{\boldsymbol{\Delta}} = \dot{\boldsymbol{\Delta}}^{free} + \mathbf{J}_i \ddot{\mathbf{q}}_i^{cor} + \mathbf{J}_j \ddot{\mathbf{q}}_j^{cor} \tag{A.5}$$

in which, $\ddot{\mathbf{q}}_{i,j}^{cor}$ are obtained by solving Eq. 4.3 with $\mathbf{b}_{i,j} = \mathbf{0}$, then Eq. A.5 becomes:

$$\dot{\boldsymbol{\Delta}} = \dot{\boldsymbol{\Delta}}^{free} + \underbrace{dt \left[\mathbf{J}_i \mathbf{A}_i^{-1} \mathbf{J}_i^T + \mathbf{J}_j \mathbf{A}_j^{-1} \mathbf{J}_j^T \right]}_{\mathbf{W}} \lambda, \tag{A.6}$$

where $\mathbf{W}$ is compliance matrix homogeneous to system matrix $\mathbf{A}$. Thanks to this operator, mechanical behaviors of two sets of contact points in response to their physical interaction (or constraint laws) are mutually dependent on geometrical as well as material properties of each object stored in system matrix $\mathbf{A}$. Hence, in the case of *multi-contact*, we can use this mechanical coupling to describe the effect of one contact on each other and vice versa.

- *Step 3:* Equation A.6 combined with Signorini's law (shown in A.1) and Coulomb's law (shown in A.2) exposes a nonlinear complementarity problem (NCP) [107]. Solving this problem for each detected collision one by one using optimization-based Gauss-Seidel algorithm [108] provides the value of

Lagrange multipliers $\boldsymbol{\lambda}$.

- *Step 4:* Once $\boldsymbol{\lambda}$ (whether single- or multi-contact case) are converged, we can apply the corrective motion into the dynamic equation A.3 to rectify the positions of object i and j so they fulfill the contact and friction laws:

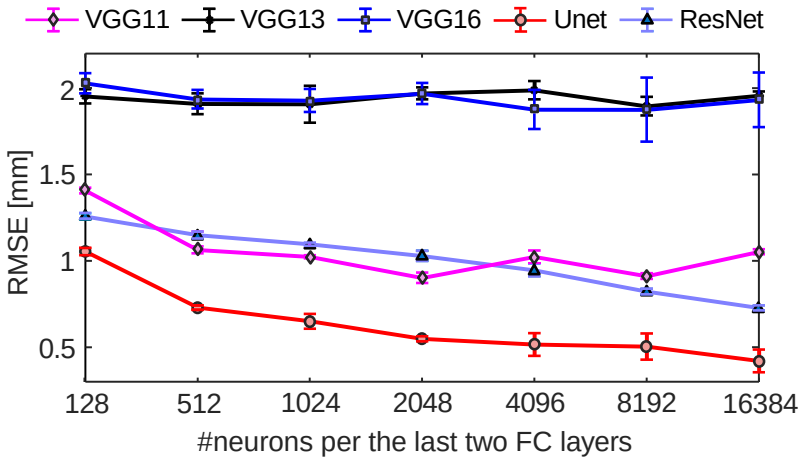
$$\begin{aligned} \mathbf{q}_{i,j}(t+dt) &= {\mathbf{q}_{i,j}}^{free} + \ddot{\mathbf{q}}_{i,j}^{cor} \\ \text{where } \ddot{\mathbf{q}}_{i,j}^{cor} &= {\mathbf{A}_{i,j}}^{-1} {\mathbf{J}_{i,j}}^{T} \boldsymbol{\lambda} \end{aligned} \tag{A.7}$$

Appendix B

TacNet configuration evaluation

To determine the TacNet architecture most suited to our problem, we conducted a preliminary evaluation of TacNet performance with Unet, ResNet, and three variant VGG configurations [109] (*i.e*, VGG-11, -13 and -16) and each model with varying number of neurons per the last two FC layers $k = \{2^n | n \in \mathbb{Z} : 8 \leq n \leq 14\}$. The model performance was measured by 5-fold cross validation, based on the average RMSE of output neurons, with 20% of the simulation data withheld as test fold.

As shown in Fig. B.1a, Unet-based TacNet significantly outperformed the ResNet-based architecture and three variants of VGG architectures in terms of RMSE metric. We attribute this to the high model complexity of VGG models [109] that are prone to the overfitting problem, leading to worse generalization compared to the Unet configuration. Moreover, the number of neurons per the last two hidden FC layers k influence validation accuracy. Specifically, regarding the Unet-based model, the model performance is substantially improved as the hidden layers are extended, which is confirmed by a drop in RMSE of around 60%. Considering memory efficiency and computational speed, however, it is reasonable to choose the Unet model followed by two 2048-neuron FC layers as a standard TacNet configuration (see Fig. B.1b). At this TacNet configuration ($k = 2048$), its average RMSE stands at 0.55 mm compared to 0.42 mm of the model with $k = 16384$, but it saves 1215 MB in memory usage and can boost the processing speed to around 200 Hz (GPU is required), which enables the use of TacNet-based sensing in mobile robotics systems (*e.g.*, drones, mobile manipulators).



(a) TacNet performance by configurations

k	Memory usage (MB)	Inference time (ms)
512	17	4.76 ± 2.49
1024	28	4.72 ± 2.50
2048	54	4.82 ± 2.49
4096	132	4.99 ± 2.49
8192	382	5.45 ± 2.14
16384	1269	6.96 ± 2.66

(b) Specifications of Unet-based TacNet with various number of neurons k

Figure B.1: TacNet performance by various network configurations. (a) 5-fold cross validation accuracy (RMSE metric) of TacNet by varying number of neurons k per the last two FC layers and backbone network architectures (Unet and VGG). Under the same training conditions, Unet-based TacNet achieves better performance compared to that of VGG counterparts (smaller RMSE value is better). Based on the specifications (b), it is reasonable to adopt 2048 neurons for the last two FC layers of Unet-based TacNet, which strikes a balance between accuracy, memory usage and inference time.

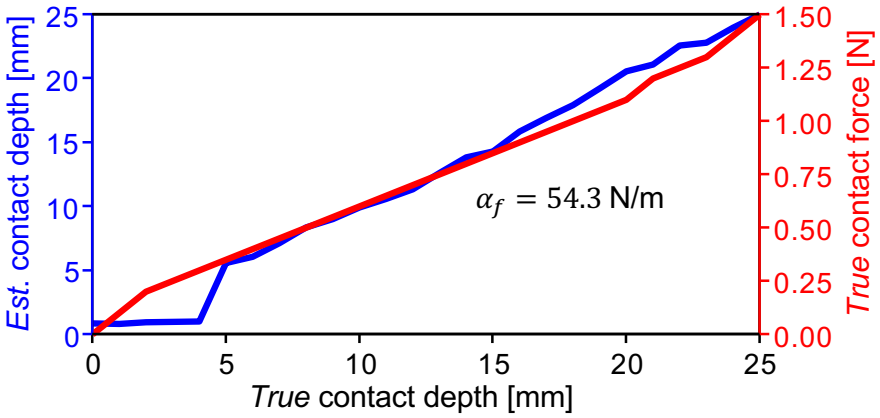
Appendix C

Force calibration

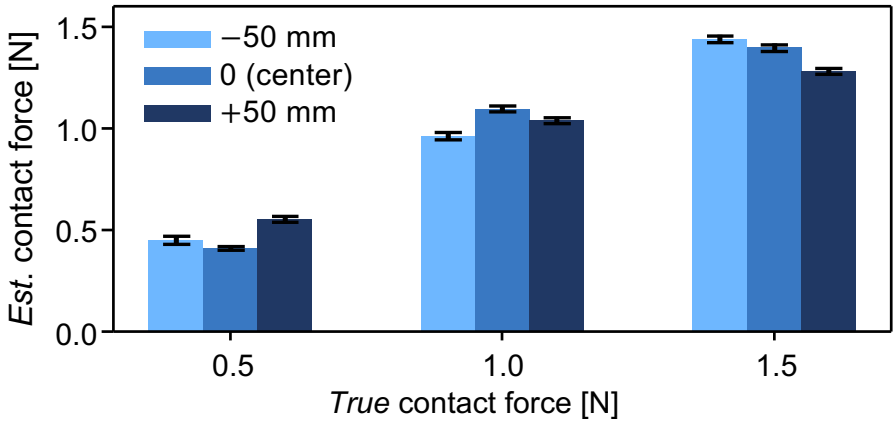
In order to determine the elastic spring constant α_f required for the force mapping (Eq. 7.3), the TacLink was pressed against a fixture affixed to a force gauge (Imada, ZTA), aligned perpendicularly to the skin surface (refer to Figure 7.2). By gradually increasing the depth of contact in 1 mm intervals, we recorded both the estimated contact depths provided by the TacLink and the corresponding true force values. The relationship between the true contact depths and the estimated values, along with the correlation between contact depth and true contact force, is depicted in Figure C.1a.

The result shows that the contact depth signal displayed a hysteresis trait, only discernible in response to local contact depths exceeding 5 mm, thus establishing a minimum detectable contact force threshold of approximately 0.4 N for the TacLink. Additionally, the presence of the internal acrylic bone limited the upper range of force sensing to approximately 1.5 N. Utilizing the observed correlation between contact depth and force illustrated in Figure C.1a, the quasi-static linear elastic model yielded a stiffness constant of $\alpha_f = 54.3\,\mathrm{N/m}$ to characterize the force-displacement relationship.

Figure C.1b reports the accuracy of calibrated contact forces estimated by the TacLink at three different locations on the sensing skin, along its vertical axis: i) at the center region, ii) 50 mm to the left, and iii) 50 mm to the right of the center location. The results showed a consistent sensing pattern among the different contact regions, with the estimated values positively correlated with the true ones. However, there were moderate differences in accuracy between the contact locations. These discrepancies can be attributed to the varying skin stiffness in different regions due to its specific morphology. While we applied a single stiffness constant of a simple



(a) The correlation between contact depth and force



(b) Estimated contact force at different contact regions

Figure C.1: The quantitative evaluation of tactile sensing performance with regard to the contact depth and calibrated contact force.

linear model to calibrate force values for the entire skin, it may not fully capture the regional variations. Moreover, the maximum deviation of approximately 17 % in contact force between the three regions was observed when the skin experienced a large contact depth of 25 mm. This can be accounted for by the limitations of a linear elastic model in accurately representing large deformations. Nevertheless, considering the observed sensing accuracy and the alignment of measured values with actual ones, the TacLink sensor remains effective for various robotics applications, while leaving room for further improvements.

Figure C.1b illustrates the accuracy of calibrated contact forces estimated by the TacLink at three distinct locations along the vertical axis of the sensing skin: i) at the central region, ii) 50 mm to the left, and iii) 50 mm to the right of the central position.

The findings revealed a consistent sensing trend across different contact regions, with estimated values exhibiting a positive correlation with true ones. However, moderate disparities in accuracy were observed among the contact locations, attributable to variations in skin stiffness owing to its specific morphological characteristics. Hence, a single stiffness constant α_f applied to calibrate force values across the entire skin might not comprehensively address regional differences. Furthermore, a maximum deviation of approximately 17 % in contact force among the three regions was noted under conditions of substantial skin deformation (25 mm), highlighting the limitations of a linear elastic model in accurately capturing large deformations. Nonetheless, considering the reasonable alignment of measured values with true ones and high elasticity, the TacLink sensor proves potential for effective collision handling, albeit with potential avenues for enhancement.

References

[1] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017.

[2] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft Robotics*, vol. 5, no. 2, pp. 216–227, 2018, pMID: 29297773.

[3] G. Cheng, E. Dean-Leon, F. Bergner, J. Rogelio Guadarrama Olvera, Q. Leboutet, and P. Mittendorfer, "A comprehensive realization of robot skin: Sensors, sensing, control, and applications," *Proceedings of the IEEE*, vol. 107, no. 10, pp. 2034–2051, 2019.

[4] K. Park, H. Yuk, M. Yang, J. Cho, H. Lee, and J. Kim, "A biomimetic elastomeric robot skin using electrical impedance and acoustic tomography for tactile sensing," *Science Robotics*, vol. 7, no. 67, p. eabm7187, 2022.

[5] F. Giovinazzo, F. Grella, M. Sartore, M. Adami, R. Galletti, and G. Cannata, "From cyskin to proxyskin: Design, implementation and testing of a multi-modal robotic skin for human–robot interaction," *Sensors*, vol. 24, no. 4, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/4/1334

[6] D. F. Gomes, Z. Lin, and S. Luo, "Geltip: A finger-shaped optical tactile sensor for robotic manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9903–9909.

[7] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, and R. Calandra, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.

[8] Z. Lin, J. Zhuang, Y. Li, X. Wu, S. Luo, D. F. Gomes, F. Huang, and Z. Yang, "Gelfinger: A novel visual-tactile sensor with multi-angle tactile image stitching," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5982–5989, 2023.

[9] F. R. Hogan, M. Jenkin, S. Rezaei-Shoshtari, Y. Girdhar, D. Meger, and G. Dudek, "Seeing through your skin: Recognizing objects with a novel visuotactile sensor," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1217–1226.

[10] J. Yin, G. M. Campbell, J. Pikul, and M. Yim, "Multimodal proximity and visuotactile sensing with a selectively transmissive soft membrane," in *2022 IEEE International Conference on Soft Robotics (RoboSoft)*, 2022, pp. 802–808.

[11] E. Roberge, G. Fornes, and J.-P. Roberge, "Stereotac: A novel visuotactile sensor that combines tactile sensing with 3d vision," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6291–6298, 2023.

[12] L. Van Duong and V. A. Ho, "Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 390–403, 2021.

[13] E. Mariotti, E. Magrini, and A. D. Luca, "Admittance control for human-robot interaction using an industrial robot equipped with a f/t sensor," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6130–6136.

[14] A. Albini, F. Grella, P. Maiolino, and G. Cannata, "Exploiting distributed tactile sensors to drive a robot arm through obstacles," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4361–4368, 2021.

[15] T. Laliberté and C. Gosselin, "Low-impedance displacement sensors for intuitive physical human–robot interaction: Motion guidance, design, and

prototyping," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1518–1530, 2022.

[16] A. Goncalves, N. Kuppuswamy, A. Beaulieu, A. Uttamchandani, K. M. Tsui, and A. Alspach, "Punyo-1: Soft tactile-sensing upper-body robot for large object manipulation and physical human interaction," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, 2022, pp. 844–851.

[17] K. Park, K. Shin, S. Yamsani, K. Gim, and J. Kim, "Low-cost and easy-to-build soft robotic skin for safe and contact-rich human–robot collaboration," *IEEE Transactions on Robotics*, vol. 40, pp. 2327–2338, 2024.

[18] S. Lee, J. I. Kim, Y. Baek, D. Chang, J. Lee, Y. S. Park, D. Lee, and Y.-L. Park, "Fiber-optic force sensing of modular robotic skin for remote and autonomous robot control," *IEEE Transactions on Robotics*, vol. 40, pp. 2373–2389, 2024.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[20] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—from humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.

[21] S.-H. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[22] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 389–400, 2011.

[23] T.-H.-L. Le, P. Maiolino, F. Mastrogiovanni, and G. Cannata, "Skinning a robot: Design methodologies for large-scale robot skin," *IEEE Robotics Automation Magazine*, vol. 23, no. 4, pp. 150–159, 2016.

[24] K. Kamiyama, H. Kajimoto, N. Kawakami, and S. Tachi, "Evaluation of a vision-based tactile sensor," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2, 2004, pp. 1542–1547 Vol.2.

[25] U. H. Shah, R. Muthusamy, D. Gan, Y. Zweiri, and L. Seneviratne, "On the Design and Development of Vision-based Tactile Sensors," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 4, p. 82, 2021.

[26] S. Zhang, Z. Chen, Y. Gao, W. Wan, J. Shan, H. Xue, F. Sun, Y. Yang, and B. Fang, "Hardware technology of vision-based tactile sensor: A review," *IEEE Sensors Journal*, pp. 1–1, 2022.

[27] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[28] U. Kim, G. Jo, H. Jeong, C. H. Park, J.-S. Koh, D. I. Park, H. Do, T. Choi, H.-S. Kim, and C. Park, "A novel intrinsic force sensing method for robot manipulators during human–robot interaction," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2218–2225, 2021.

[29] T. Laliberté and C. Gosselin, "Low-impedance displacement sensors for intuitive physical human–robot interaction: Motion guidance, design, and prototyping," *IEEE Transactions on Robotics*, pp. 1–13, 2021.

[30] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3356–3363.

[31] S. Golz, C. Osendorfer, and S. Haddadin, "Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3788–3794.

[32] S. Kuhn and D. Henrich, "Fast vision-based minimum distance determination between known and unkown objects," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 2186–2191.

[33] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 338–345.

[34] A. De Luca and F. Flacco, "Integrated control for phri: Collision avoidance, detection, reaction and collaboration," in *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2012, pp. 288–295.

[35] V. A. Ho, S. Hirai, and K. Naraki, "Fabric interface with proximity and tactile sensation for human-robot interaction," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 238–245.

[36] J. C. Yang, J. Mun, S. Y. Kwon, S. Park, Z. Bao, and S. Park, "Electronic skin: Recent progress and future prospects for skin-attachable devices for health monitoring, robotics, and prosthetics," *Advanced Materials*, vol. 31, no. 48, p. 1904765, 2019.

[37] G. Pang, G. Yang, and Z. Pang, "Review of robot skin: A potential enabler for safe collaboration, immersive teleoperation, and affective interaction of future collaborative robots," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 3, pp. 681–700, 2021.

[38] H. Park, K. Park, S. Mo, and J. Kim, "Deep neural network based electrical impedance tomographic sensing methodology for large-area robotic tactile sensing," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1570–1583, 2021.

[39] P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401–410, 2011.

[40] Q. Leboutet, E. Dean-Leon, F. Bergner, and G. Cheng, "Tactile-based whole-body compliance with force propagation for mobile manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 330–342, 2019.

[41] S. Armleder, E. Dean-Leon, F. Bergner, and G. Cheng, "Interactive force control based on multimodal robot skin for physical human-robot collaboration," *Advanced Intelligent Systems*, vol. 4, no. 2, p. 2100047, 2022.

[42] A. C. Abad and A. Ranasinghe, "Visuotactile sensors with emphasis on gelsight sensor: A review," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 7628–7638, 2020.

[43] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine, "Omnitact: A multi-directional high-resolution touch sensor," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 618–624.

[44] W. Li, A. Alomainy, I. Vitanov, Y. Noh, P. Qi, and K. Althoefer, "F-touch sensor: Concurrent geometry perception and multi-axis force measurement," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4300–4309, 2021.

[45] W. K. Do and M. Kennedy, "Densetact: Optical tactile sensor for dense shape reconstruction," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6188–6194.

[46] J. Zhao and E. H. Adelson, "Gelsight svelte: A human finger-shaped single-camera tactile robot finger with large sensing coverage and proprioceptive sensing," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 8979–8984.

[47] K. Sato, K. Kamiyama, N. Kawakami, and S. Tachi, "Finger-shaped gelforce: Sensor for measuring surface traction fields for robotic hand," *IEEE Transactions on Haptics*, vol. 3, no. 1, pp. 37–47, 2010.

[48] C. Sferrazza and R. D'Andrea, "Sim-to-real for high-resolution optical tactile sensing: From images to three-dimensional contact force distributions," *Soft Robotics*, 2021.

[49] J. Yin, P. Shah, N. Kuppuswamy, A. Beaulieu, A. Uttamchandani, A. Castro, J. Pikul, and R. Tedrake, "Proximity and visuotactile point cloud fusion for contact patches in extreme deformation," 2023.

[50] S. Zhang, Y. Sun, J. Shan, Z. Chen, F. Sun, Y. Yang, and B. Fang, "Tirgel: A visuo-tactile sensor with total internal reflection mechanism for external observation and contact detection," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6307–6314, 2023.

[51] S. Athar, G. Patel, Z. Xu, Q. Qiu, and Y. She, "Vistac toward a unified multimodal sensing finger for robotic manipulation," *IEEE Sensors Journal*, vol. 23, no. 20, pp. 25 440–25 450, 2023.

[52] W. Fan, H. Li, W. Si, S. Luo, N. Lepora, and D. Zhang, "Vitactip: Design and verification of a novel biomimetic physical vision-tactile fusion sensor," 2024.

[53] A. Agarwal, T. Man, and W. Yuan, "Simulation of vision-based tactile sensors using physics-based rendering," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1–7.

[54] D. F. Gomes, P. Paoletti, and S. Luo, "Generation of gelsight tactile images for sim2real learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4177–4184, 2021.

[55] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra, "TACTO: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3930–3937, 2022.

[56] Z. Si and W. Yuan, "Taxim: An example-based simulation model for gelsight tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2361–2368, 2022.

[57] D. F. Gomes, S. Luo, and P. Paoletti, "Beyond flat gelsight sensors: Simulation of optical tactile sensors of complex morphologies for sim2real learning," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, Eds., 2023.

[58] Y. Zhao, K. Qian, B. Duan, and S. Luo, "Fots: A fast optical tactile simulator for sim2real learning of tactile-motor robot manipulation skills," *IEEE Robotics and Automation Letters*, pp. 1–8, 2024.

[59] Z. Ding, N. F. Lepora, and E. Johns, "Sim-to-real transfer for optical tactile sensing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1639–1645.

[60] Y. Lin, J. Lloyd, A. Church, and N. F. Lepora, "Tactile gym 2.0: Sim-to-real deep reinforcement learning for comparing low-cost high-resolution robot touch," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 754–10 761, 2022.

[61] A. Church, J. Lloyd, raia hadsell, and N. F. Lepora, "Tactile sim-to-real policy transfer via real-to-sim image translation," in *5th Annual Conference on Robot Learning*, 2021.

[62] C. Sferrazza, T. Bi, and R. D'Andrea, "Learning the sense of touch in simulation: a sim-to-real strategy for vision-based tactile sensing," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4389–4396.

[63] C. Sferrazza and R. D'Andrea, "Sim-to-real for high-resolution optical tactile sensing: From images to three-dimensional contact force distributions," *Soft Robotics*, 2021.

[64] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," *Proc. Australas. Conf. Robot. Automat.*, p. 1–8, 2015.

[65] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

[66] W. Chen, Y. Xu, Z. Chen, P. Zeng, R. Dang, R. Chen, and J. Xu, "Bidirectional sim-to-real transfer for gelsight tactile sensors with cyclegan," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6187–6194, 2022.

[67] X. Jing, K. Qian, T. Jianu, and S. Luo, "Unsupervised adversarial domain adaptation for sim-to-real transfer of tactile images," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.

[68] L. Pecyna, S. Dong, and S. Luo, "Visual-tactile multimodality for following deformable linear objects using reinforcement learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3987–3994.

[69] N. F. Lepora, Y. Lin, B. Money-Coomes, and J. Lloyd, "Digitac: a digit-tactip hybrid tactile sensor for comparing low-cost high-resolution robot touch," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9382–9388, 2022.

[70] A. Butterworth, G. Pizzuto, L. Pecyna, A. I. Cooper, and S. Luo, "Leveraging multi-modal sensing for robotic insertion tasks in r&d laboratories," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–8.

[71] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical human–robot interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 41–48, 2016.

[72] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *The International Journal of Robotics Research*, vol. 32, no. 4, pp. 458–482, 2013.

[73] P. Svarny, J. Rozlivek, L. Rustler, M. Sramek, and et al., "Effect of active and passive protective soft skins on collision forces in human–robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102363, 2022.

[74] N. M. D. Le, N. H. Nguyen, D. A. Nguyen, T. D. Ngo, and V. A. Ho, "Viart: Vision-based soft tactile sensing for autonomous robotic vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 2, pp. 1420–1430, 2024.

[75] V. A. Ho and S. Nakayama, "Iotouch: whole-body tactile sensing technology toward the tele-touch," *Advanced Robotics*, vol. 35, no. 11, pp. 685–696, 2021.

[76] S. Yoshigi, J. Wang, S. Nakayama, and V. A. Ho, "Deep learning-based whole-arm soft tactile sensation," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 132–137.

[77] N. H. Nguyen and V. A. Ho, "Tactile compensation for artificial whiskered sensor system under critical change in morphology," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3381–3388, 2021.

[78] N. H. Nguyen and V. A. Ho, "Mechanics and morphological compensation strategy for trimmed soft whisker sensor," *Soft Robotics*, vol. 9, no. 1, pp. 135–153, 2022.

[79] J. Allard, H. Courtecuisse, and F. Faure, "Chapter 21 - implicit fem solver on gpu for interactive deformation simulation," in *GPU Computing Gems Jade Edition*, ser. Applications of GPU Computing Series, W. mei W. Hwu, Ed. Boston: Morgan Kaufmann, 2012, pp. 281–294.

[80] T. Liu, C. Zhao, Q. Li, and L. Zhang, "An efficient backward euler time-integration method for nonlinear dynamic analysis of structures," *Computers & Structures*, vol. 106-107, pp. 20–28, 2012.

[81] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical Image Analysis*, vol. 18, no. 2, pp. 394–410, 2014.

[82] J. A. González, K. Park, C. A. Felippa, and R. Abascal, "A formulation based on localized lagrange multipliers for bem–fem coupling in contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 6, pp. 623–640, 2008.

[83] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[84] F. Bettonvil, "Fisheye lenses," *WGN, Journal of the International Meteor Organization*, vol. 33, no. 1, pp. 9–14, Feb. 2005.

[85] J. F. Blinn, "Models of light reflection for computer synthesized pictures," *SIGGRAPH Comput. Graph.*, vol. 11, no. 2, p. 192–198, 1977.

[86] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[87] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2017.

[88] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[89] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[90] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. O'Reilly Media, Inc., 2017.

[91] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognition*, vol. 70, pp. 25–43, 2017.

[92] Q. K. Luu, N. H. Nguyen, and V. A. Ho, "Simulation, learning, and application of vision-based tactile sensing at large scale," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2003–2019, 2023.

[93] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, ser. Cambridge books online. Cambridge University Press, 2003. [Online]. Available: https://books.google.co.jp/books?id=si3R3Pfa98QC

[94] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 03, pp. 1623–1637, mar 2022.

[95] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. T. Freeman, "Mannequinchallenge: Learning the depths of moving people by watching frozen people," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4229–4241, 2021.

[96] ——, "Learning the depths of moving people by watching frozen people," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4516–4525, 2019.

[97] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[99] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. USA: Cambridge University Press, 2017.

[100] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee, "Tracking anything with decoupled video segmentation," in *ICCV*, 2023.

[101] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3356–3363.

[102] I. . Robotics, "Robots and robotic devices — collaborative robots," in *ISO/TS 15066:2016*, 2016.

[103] Y. Osawa, Q. K. Luu, L. V. Nguyen, and V. A. Ho, "Integration of soft tactile sensing skin with controllable thermal display toward pleasant human-robot interaction," in *2024 IEEE/SICE International Symposium on System Integration (SII)*, 2024, pp. 369–375.

[104] V. C. Pham, Q. K. Luu, T. T. Nguyen, N. H. Nguyen, Y. Tan, and V. A. Ho, "Web of tactile things: Towards an open and standardized platform for tactile things via the w3c web of things," in *Intelligent Information Systems*, J. De Weerdt and A. Polyvyanyy, Eds. Cham: Springer International Publishing, 2022, pp. 92–99.

[105] N. M. Dinh Le, T. T. Nguyen, Q. K. Luu, N. H. Nguyen, V. C. Pham, Y. Tan, and V. A. Ho, "Integration of web of tactile things for soft vision-based tactile sensor toward immersive human-robot interaction," in *2024 IEEE/SICE International Symposium on System Integration (SII)*, 2024, pp. 1343–1348.

[106] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, *SOFA: A Multi-Model Framework for Interactive Physical Simulation.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 283–321.

[107] L. Yong, "Nonlinear complementarity problem and solution methods," in *Artificial Intelligence and Computational Intelligence.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 461–469.

[108] F. Dubois, C. Duriez, A. Kheddar, and C. Andriot, "Realistic haptic rendering of interacting deformable objects in virtual environments," *IEEE Transactions on Visualization & Computer Graphics*, vol. 12, no. 01, pp. 36–47, jan 2006.

[109] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9*, 2015.