Master Thesis

Czech Technical University in Prague

F3 Faculty of Electrical Engineering
Department of Cybernetics

Efficient Exploration of Body Surface with Tactile Sensors on Humanoid Robots

Maksym Shcherban

Supervisor: Mgr. Matěj Hoffmann, Ph.D.
Field of study: Cybernetics and robotics
Subfield: Robotics
August 2021

Acknowledgements

I would like to thank CTU in general for the quality of education; my thesis supervisor Matěj Hoffmann; and my wife for support.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 13. August 2021

Abstract

Sense of touch plays an important role in the life of a person, however it is underutilised and understudied in the field of robotics. Tactile sensory modality has large potential in many areas, from robots building and calibrating models of their bodies using tactile feedback, to enabling safe human-robot interaction.

Although rich literature exists on the topics of active learning and intrinsic motivation, authors rarely test their hypotheses on robots with the sense of touch. And when they do, they often prefer to use extremely simple simulations of planar manipulators.

In this thesis, I address this problem by developing an artificial skin simulator for the iCub humanoid robot used for research in cognitive developmental robotics, and using it for experiments in efficient exploration of the robot's body. I have successfully implemented the artificial skin simulator for the iCub humanoid robot and used it to perform a set of experiments in body surface exploration. I have applied the goal babbling exploration framework and exploration by disagreement algorithm to efficiently explore the simulated robot's body surface. I have also compared several inverse body models suitable for the task of tactile exploration.

Once the artificial skin simulator is accepted into the iCub codebase, it will make it easier for other researchers to perform experiments involving the sense of touch on a humanoid robot.

Abstrakt

Zatímco u člověka hraje hmat velmi důležitou roli, v robotice je mu věnována malá pozornost. Dotyková zpětná vazba má přitom obrovský potenciál, od automatické kalibrace robotů po bezpečnou interakci člověka s robotem.

V oblasti aktivního učení a vnitřní motivace existuje celá řada prací, ale málokdy jsou algoritmy testovány na robotech s dotykovou zpětnou vazbou. Pokud je taktilní modalita přece jen použita, jedná se často o velmi jednoduché simulace např. planárních manipulátorů.

V této práci jsem k existujícímu simulátoru robota iCub, který se používá v kognitivní vývojové robotice, v Gazebo přidal taktilní zpětnou vazbu. Simulátor jsem použil k sérii experimentů zaměřených na aktivní průzkum povrchu těla. K tomu jsem použil algoritmy založené na "goal babbling" a "exploration by disagreement". Také jsem porovnal různé způsoby implementace inverzních modelů na úloze taktilní explorace.

Po integraci simulátoru kůže do oficiálního iCub Gazebo simulátoru bude tento nástroj k dispozici široké komunitě uživatelů.

Klíčová slova: umělá kůže, humanoidní robot, iCub, aktivní učení, učení založené na zvědavosti, explorace, vnitřní motivace

Překlad názvu: Efektivní průzkum povrchu těla s taktilními senzory u humanoidních robotů

v

Contents

Chapter 1

Introduction

For humans, the sensation of touch is a crucial means of receiving information about the environment. Self-touch plays an important role in the development of human infants. Our entire bodies are covered with a dense network of touch sensors. Information flow from these sensors allows us to perform complex dexterous manipulation tasks, discern material properties, and navigate in conditions where visual feedback is scarce or completely unavailable.

Modern robots, on the other hand, underutilize the sense of touch, relying more on cameras and depth sensing devices. Commercially available autonomous mobile robots, like Roomba, only have crude bumper sensors that signal collisions with household objects and walls. Industrial robots use the sense of touch mainly in the form of limit switches that signal the boundaries of the operational space.

Equipping robots with full-body touch capabilities would expand their limits. For example, research on collaborative robots shows that touch-enabled robots can safely operate alongside human workers with minimal risk of physical harm. Touch sensors can be fused with visual sensors for more efficient autonomous navigation. Recently, artificial skin solutions have found their way to the collaborative robot industry through Airskin (pressure sensitive) and Bosch APAS (using proximity) [40].

With this work, I pursue the following goals:

1. Development of simulator for iCub humanoid robot with artificial skin.

2. Application of curiosity-driven active learning algorithms to the problem of exploration of body surface covered with artificial skin.

3. Comparison of experimental results to our previous work in [46, 15, 16].

The thesis is structured as follows. Chapter 2 briefly reviews related research in the areas of active learning, curiosity-driven learning, and touch-enabled robots. Chapter 3 presents the robot simulator, learning framework and algorithms. Chapter 4 presents my contribution in the form of simulated artificial skin and the environment for conducting experiments in exploration and learning. Chapter 5 contains experimental results: application of active learning algorithms to the problem of artificial skin exploration and comparison with previous results obtained in [46, 15, 16] with exploration framework based on goal babbling [38].

Chapter 6 summarizes the main conclusions from the experimental results. Results are further discussed in Chapter **??**. Possibilities for future work are outlined in Chapter 7.

Chapter 2

Related work

2.1 Sense of touch in robotics

Human skin is covered with tactile sensors and provides rich tactile feedback. The spatial coverage of human skin with touch receptors reaches 240 units per cm² at the fingertips ([51], [52]) and at the moment is unparalleled by artificial skins available for robots. Yamada et al. in [54] have developed an embodied brain model of the human foetus, complemented with an anatomically correct full-body skin model. In our previous work [46, 15, 16] we have experimented with body model learning on a simpler humanoid robot with artificial skin by employing the computational frameworks of intrinsic motivation and goal babbling.

Various designs and applications of the sense of touch in robotics have been studied by a number of researchers—see e.g., [7, 14] for surveys or the 2019 special issue of Proceedings of the IEEE [13]). Struckmeier et al. in [49] and Suresh et al. in [41] investigated the possibility of using touch for navigation, either in tactile only or in visuo-tactile SLAM algorithms. I did not use this research directly, but as a matter of fact, a robot exploring its own body is solving a SLAM problem.

Church et al. in [11] combined deep reinforcement learning with feedback from TacTip soft tactile sensor [53] in order to teach a robot arm to type on a Braille keyboard. Lloyd et al. in [27] used tactile feedback to learn adaptive control policies for pushing objects of various shapes and materials

towards a goal pose. Lepora and Lloyd in [26] devised a novel algorithm for controlling robots using soft tactile sensors called Pose-Based Servo Control. Roncone et al. [39] and Rustler et al. [40] employed self-contact and robot skin to calibrate the robot kinematics or the spatial coordinates of the tactile sensors, respectively. These are few examples of a multitude of ways in which robotic systems may be improved with the sense of touch.

■ 2.2 Active learning and intrinsic motivation

Active learning refers to a subfield of machine learning methods in which the agent is allowed to actively query for the next data point during the learning process. Settles and Burr conducted an extensive general survey of the active learning literature in [44]. Baranes et al. have developed the SAGG-RIAC framework [5] specifically for active learning of inverse models in high-dimensional redundant spaces. Another example is M. Rolf's goal babbling framework [37]. A sample application of this framework is given in [38], where an inverse model is learnt for a bionic elephant trunk robot.

Intrinsic rewards are those generated by the agent, in contrast to the rewards provided by the robot's environment. Curiosity is a type of intrinsic reward. Burda et al. performed a large-scale study of curiosity-driven learning [9]. Mori et al. in [31] showed that tactile-based curiosity induces the emergence of tactile-rich object-oriented behaviors. Committee disagreement is another type of intrinsic reward, first described by Seung et al. in [45]. Pathak et al. show in [35] and [34] how disagreement can be used for efficient exploration of the environment. I have based my artificial skin exploration algorithm on this research.

There are numerous other examples of intrinsic motivation applications. Sukhbaatar et al. in [50] show how intrinsic motivation can be combined with asymmetric self-play for efficient exploration. Schmidhuber in [42] use curiosity learning to build a control system which "actively tries to provoke situations for which it learned to expect to learn something about the environment". Intrinsic rewards can be used with reinforcement learning algorithms, an extensive survey of methods that combine intrinsic and extrinsic rewards has been performed by Aubert et al. in [2] and by Barto in [6].

Simulator	Random External Forces	Force Sensors	Multiple Physics Engines	Realistic Rendering	Support for Soft Bodies	Open Source
Gazebo	+	+	+	-	**	+
NVIDIA Isaac	+	*	-	+	+	-
MuJoCo	+	+	-	-	+	-
Webots	+	+	-	-	-	+

Table 2.1: Comparison of features of robotic simulators [12]
* - NVIDIA added support of DoF force sensors in latest versions of Isaac SDK
** - not supported natively, but can be implemented with FEM plugin [10]

2.3 Comparison of robotic simulators

Ayala et al. conducted a quantitative comparison of three humanoid robot simulators in [3]. They evaluated the use of CPU, memory footprint, and disk access by the simulators. Collins et al. performed a more comprehensive review of physics simulators for robotic applications [12]. "Learning for Robotics" section of their work is particularly relevant for my thesis; it classifies the simulators according to several features important for robotic learning algorithms: the availability of a multitude of sensors, the ability to apply random external forces to a robot, realistic rendering etc. A brief summary of these features is given in table 2.1.

- Ability to apply *random external forces* is important for two reasons. Firstly, the addition of exploration noise is crucial for efficient exploration of the search space, as shown by M. Rolf in [37]. Secondly, the models trained on noisy data are more robust, and it is generally easier to transfer such models from simulation to the real robot. All simulators that I have taken into consideration support this feature. I am not using this feature directly, instead, I am adding noise when generating goals for exploration.

- Support for *force sensing* is crucial for the simulation of artificial skin with tactile feedback. Unfortunately, NVIDIA Isaac simulator did not have support for force sensing at the time when the decision about simulator choice was made.

- Gazebo allows the user to select one of *multiple physics engines*. Currently Gazebo supports 4 physics engines: ODE (default), Bullet, DART, and Simbody. Although this feature is not critical, it is beneficial to be able to compare the simulation results with several physics engines. In this work, I am only using ODE physics engine.

- *Photorealistic rendering* is important for machine learning applications where models are learnt from visual feedback. This feature is not critical for experiments in tactile exploration.

- Physics engines that simulate rigid bodies are known to generate *contact jitter* (for example, see an open issue for `ODE` engine [17]). Simulators that *support soft bodies*, *deformation*, and *soft touch* mitigate this problem. Unfortunately, the `Gazebo` simulator does not support this feature natively. However, Chen et al. show in [10] that this functionality can be implemented with a Gazebo plugin using the *finite element method* (FEM) to accurately model the deformations. I have solved this issue by adding a low-pass filter to the generated contact data.

- Out of the 4 simulators under consideration, only `Gazebo` and `Webots` are both open source and free to use for education and research purposes.

- The models for the iCub simulations are automatically generated from the iCub CAD designs in URDF format. All the abovementioned simulators support the `URDF` format, either natively or by automatically converting `URDF` files into an internal representation format (e.g. `SDF` in `Gazebo`)

Based on these considerations, I have decided to use Gazebo robotic simulator [25] in my thesis.

◼ 2.4 **Thesis contribution**

In our previous work [46, 15, 16], we used the Nao humanoid robot with artificial skin and focused on efficient self-exploration as well as learning of forward and inverse models of the robot's body.

Nao is a proprietary robotic platform developed by SoftBank Robotics Corporation. The iCub humanoid robot [30], on the other hand, is an open source humanoid robot that was designed specifically for studies in developmental cognitive robotics. Although the physical iCub robot comes with capacitive artificial skin, there has been limited support for iCub artificial skin in the simulation. Official iCub github repositories (`icub-main`, `icub-models`) come without proper support of the artificial skin simulation.

In this thesis, I add artificial skin support to the iCub humanoid robot Gazebo simulation. I then apply the curiosity-based active learning approach described by Pathak et al. in [34], [9], and [35], to the novel task of artificial skin exploration using the sensation of touch on a simulated iCub humanoid robot. Finally, I compare the efficiency of curiosity-based exploration with goal-based exploration techniques from our previous work [46, 15, 16].

Chapter 3

Methods

3.1 iCub humanoid robot with artificial skin

iCub is a humanoid robot for research in embodied cognition [30]. It was designed and developed specifically for the study of the development of cognitive abilities in physical agents. The iCub robot stands 104 cm tall and weighs 22 kg (Fig. 3.1b).

The latest version of the robot at the time of writing has 53 degrees of freedom (DoF) organized in the following way [20]:

- 6 DoF for the head, of which 3 DoF are in the neck and the other 3 DoF control the eyes

- 7 DoF for each arm: 3 DoF for the shoulder, 1 DoF for the elbow and 3 DoF for the wrist

- 9 DoF for each hand, allowing control of individual fingers

- 3 DoF for the torso

- 6 DoF for each leg: 3 DoF for the hip, 1 DoF for the knee and 2 DoF for the ankle

(a) : Frontal view of the iCub robot

(b) : iCub robot with exposed artificial skin [23]

Figure 3.1: iCub humanoid robot

iCub comes with artificial skin covering its torso, arms, legs, and hands (Fig. 3.1b). The skin is made up of capacitive tactile sensors placed on triangular flexible printed circular boards. Each triangle contains 10 sensors. Triangles are connected to one another and form a network that is controlled by a single MCU.

The number of taxels per relevant body part is: 440 taxels on the torso; 380 taxels on each upper arm; 230 taxels on each forearm; 104 taxels on each hand (44 on the palm and 12 on each of the 5 fingers).

(a) : Physical sensors of the iCub artificial skin [22]
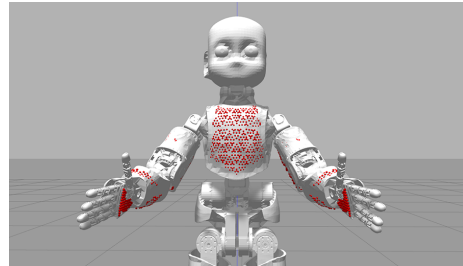
(b) : Simulation of the iCub robot with artificial skin in Gazebo 11

Figure 3.2: iCub artificial skin
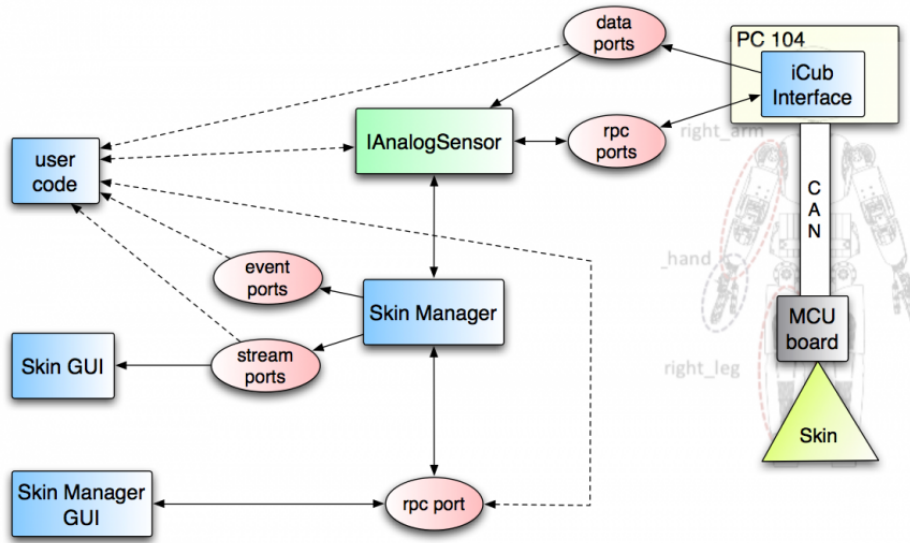
Figure 3.3: Tactile data flow [22].

Tactile data flow (schematically depicted in Fig. 3.3):

- Physical sensors produce *raw tactile data* and send it to `/icub/skin/part_name` YARP port. Raw data consists of vectors of integer values (value range `[0..255]`). 0 corresponds to the maximum pressure, and no pressure value is around 235, according to the technical documentation.

- `SkinManager` software module reads the raw tactile data, converts it into *compensated tactile data* and writes it to `/icub/skin/part_name_comp` YARP port. Compensated data consists of vectors of floating point values, where 0.0 corresponds to no pressure and 255.0 means maximum pressure.

- Compensated tactile data can be used by the user code, and it is also used for visualization by `iCubSkinGUI` module (Fig. 4.1).

3.2 Comparison of iCub with Nao humanoid robot

In my bachelor's thesis [46] and the follow-up research [15, 16], we have used the Nao humanoid robot. The iCub robot has a number of advantages over the Nao:

9
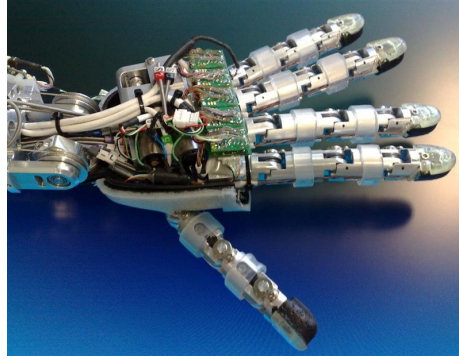
(a) : iCub's hand [43] **(b) :** Nao's hand

Figure 3.4: Comparison of robotic hands of iCub and Nao humanoid robots

1. iCub is distributed as Open Source following the GPL/FDL licenses [30], whereas Nao is a proprietary product developed and distributed by SoftBank Robotics. This makes research made with iCub more reproducible and accessible for the research community.

2. iCub has 7 DoF arm with anthropomorphic proportions, whereas Nao was not designed to closely resemble human anatomy.

3. Design of iCub's hand has close resemblance with the human hand. Each hand of the iCub robot has 9 degrees of freedom that enable meticulous control of individual fingers. Nao only has 1 controllable DoF which opens and closes all 3 fingers simultaneously (Fig. 3.4).

4. iCub robot comes with artificial skin covering its torso, legs, arms and hands. Nao, on the other hand, does not have skin sensing capabilities and has to be modified for the kind of research we are doing. These modifications turn out to be expensive and time-consuming.

3.3 Gazebo simulation environment

The simulation environment is based on YARP [29] and Gazebo 11 [25]. The Robotology github organization that brings together software for the iCub robot platform provides the `icub-models` repository [21] and `gazebo-yarp-plugins` [19] that enable the simulation of the iCub robot with Gazebo simulator. However, these repositories do not enable simulation of artificial skin. My contribution to the simulation environment is described in detail in Section 4.1.

■ 3.4 OpenAI gym

OpenAI gym is a toolkit for research in reinforcement learning (RL) [8]. The gym defines a standardized interface for the environments, which makes it easy to test learning algorithms on any environment that implements the gym API. The OpenAI gym environment I have developed is described in detail in Section 4.2.

■ 3.5 Curiosity-driven exploration framework

To efficiently explore the surface of the robot's body covered with artificial skin, I am using several exploration frameworks based on intrinsic motivation. In this section, I describe the general theoretical basis of intrinsically motivated learning, and then describe the particular framework I am using.

■ 3.5.1 Intrinsically motivated active learning

Intrinsic motivation (IM) describes learning methods where the agent does not receive any external reward from the environment. The agent performs actions and develops a certain behavior for the shear satisfaction of it, not for external reward [33], [4], [28], [2], [5].

Active learning describes a subset of machine learning algorithms where the agent is allowed to choose the data from which it learns [44]. These algorithms allow the agent to learn more accurate models with a smaller number of labeled instances, increasing the *sample efficiency.*

A number of measures of intrinsic motivation are described in the active learning literature:

- *Competence* is defined as the ability of the agent to successfully perform selected tasks. For example, competence is defined for robot's reaching attempts in [5] as *the similarity between the point in the task space y_f*

11

attained when the reaching attempt has terminated, and the actual goal y_g:

$$C = -\|y_f - y_g\| \tag{3.1}$$

We have successfully used this intrinsic motivation measure for exploration in our previous work [46, 15, 16].

- *Curiosity* can be used as an intrinsic reward. Pathak et al. use prediction error as a curiosity reward in [34].

- *Empowerment* is the measure of how much control the agent has over its environment [24]. To maximize empowerment, the agent is rewarded if it is heading towards areas where it can best control the environment.

- *Disagreement* intrinsic motivation works with ensembles of forward models and steers exploration to areas where the predictions of ensemble members disagree with each other the most [35].

Interesting effects may come from combining several measures of intrinsic motivation. For example, Rayyes et al. in [36] proposed a novel intrinsic motivation signal named *interest measurement* which combines competence-based and knowledge-based elements. This new signal, combined with *interest-driven goal babbling* exploration strategy and *online episodic mental replay* technique, allowed the authors to efficiently guide the exploration process and achieve very high sample efficiency.

■ 3.5.2 Exploration via disagreement

In self-supervised exploration via *disagreement* model, described in [35], the agent maintains an *ensemble* (or *committee*) of forward models

$$\vec{f} = \{f_1...f_n\} \tag{3.2}$$

Given the current state x_t and action a_t, the ensemble of forward models predict the next state estimates

$$\hat{x}_{t+1} = \{\hat{x}_{t+1}^1...\hat{x}_{t+1}^n\} \tag{3.3}$$

The disagreement intrinsic motivation measure is defined as the variance of these estimates:

$$r_t^i = \mathbb{E}_\vartheta \left[\| \vec{f}(x_t, a_t; \vartheta) - \mathbb{E}_\vartheta[\vec{f}(x_t, a_t; \vartheta)] \|_2^2 \right] \tag{3.4}$$

where ϑ is the vector of model parameters.

The above definition is model-agnostic. Pathak et al. use ensembles consisting of 5 deep neural networks in their work [35]. I am using *locally weighted linear regression* (LWLR) forward models [1, 32], as well as non-parametric *nearest neighbor* (NN) and *k nearest neighbors* (kNN) models.

Given a motor command $q \in \mathbf{Q}$, the LWLR forward model finds its k nearest neighbors in the database and computes a linear regression of their corresponding observations $x \in \mathbf{X}$. Here k and σ^2 are parameters of the model. The algorithm uses normalized Gaussian weights. Let d_i be the distance between q and its i-th nearest neighbor, w_i the i-th regression weight, then:

$$\begin{aligned} w_i' &= e^{-\frac{d_i^2}{2\sigma^2}} \\ w_i &= \frac{w_i'}{\sum_{j=1}^k w_j'} \end{aligned} \tag{3.5}$$

3.6 Goal-based exploration framework

We have used goal-based exploration framework in our previous research [46, 15, 16]. In this work, I am using previous results as a baseline for comparison with the results obtained by the disagreement exploration framework. Here I will briefly describe goal-based exploration architecture, for a more detailed account, please refer to [4], [32] or [16].

◼ 3.6.1 Action and observation spaces

Action space $\mathbf{Q}$ represents all possible actions of the agent. Each action $q \in \mathbf{Q}$ causes an outcome $x \in \mathbf{X}$ in some *observation space* $\mathbf{X}$. The causal relationship between action space and observation space is defined by some *forward function f* [37, p.5]:

$$
\begin{aligned}
f : \mathbf{Q} &\to \mathbf{X} \\
f(q) &= x
\end{aligned}
\tag{3.6}
$$

◼ 3.6.2 Random motor babbling

In *random motor babbling* exploration strategy, a motor configuration $q \in \mathbf{Q}$ is sampled uniformly from the action space. Selected action is executed, observation $x \in \mathbf{X}$ is recorded and the database is updated. This exploration strategy is the most naive and least effective method for learning forward and inverse models.

◼ 3.6.3 Random goal babbling

In *random goal babbling* exploration strategy, a *goal* $g \in \mathbf{X}$ is first sampled from the observation space. The agent then performs an action $q \in \mathbf{Q}$ to try and reach the selected goal. With each attempt, the agent is incrementally updating the model based on the feedback from environment.

Goals can be sampled either uniformly or based on some intrinsic motivation signal. For example, in our previous work [46, 15, 16] we have achieved good results with *discretized progress* goal sampling strategy: the observation space is discretized into x_{card} cells, a cell for goal generation is selected randomly, with a probability proportional to the current value of interest in each cell.

Chapter 4

iCub Gazebo simulation with artificial skin and OpenAI gym environment

In this chapter, I give a detailed account of my contribution. All code, data files and experiment results are available in the public GitLab repository [18]. Video demonstration of the developed simulation environment is available online at [47, 48].

4.1 Gazebo simulation of artificial skin

As the basis for the simulation, I have used `iCub_2_5_visuomanip` model from `icub-models` repository [21]. This model has fully articulated hands and is capable of performing complex manipulation tasks. In order to add a simulation of artificial skin to Gazebo, I had to complete several steps.

Firstly, a physical model of the sensors had to be added to the model. I have modeled each taxel as a small sphere. 3D coordinates relative to the parent link of taxels for the torso, arms, and palms of the hands were available in `iCubSkin` module of `icub-main` repository. For the taxels on the robot's fingers, I had to deduce coordinates from iCub CAD designs.

Secondly, I have created a contact sensor plugin for Gazebo that registers detected taxel activations, converts them to iCub *compensated tactile data*, and sends the data to `/icubSim/skin/part_name_comp` YARP port.

A similar plugin was developed for my bachelor's thesis to connect Gazebo simulation of Nao robot with ROS ports. The plugin from previous work was slowing down the simulation, causing an experiment of 1000 time steps to last for several hours, slowing down the entire research. In this thesis, I have solved this issue in the following way: instead of creating an instance of a sensor plugin for each body part covered with artificial skin, I have created a single model plugin instance that subscribes directly to `~/physics/contacts` Gazebo topic and filters out relevant tactile information. This solution helped to speed up the simulation to a factor of approximately 0.75 of real-time speed. Now a simulation of 1000 time steps lasts for about 20 minutes.

Another problem with the simulation was a *"flickering contact"*. This was caused by a long known, but still unsolved issue with the physics engine used in Gazebo [17]. Due to the numerical nature of the dynamic physics simulation of the rigid bodies, contact forces between surfaces caused tiny bumps and jumps, such that in consecutive time steps the surfaces would oscillate between contact *on* and *off* states.

I have solved this issue by reducing the update frequency of tactile data from 50 Hz (used on the physical robot) to 20 Hz. Between updates, the contact data is accumulated in the plugin's internal buffer, and any flickering occurring between two consecutive updates gets filtered out. The effect can be clearly seen in the videos: 50 Hz update rate causes visible contact flickering [48], while at 20 Hz update rate this problem is mitigated [47].

Thirdly, I have developed an OpenAI gym environment to implement, test, and compare algorithms for exploration of the robot's artificial skin. This environment is described in detail in the next section.

◼ 4.2   OpenAI gym environment

I have developed `icub_skin` OpenAI gym environment that defines an interface to the simulated iCub robot.

◼ 4.2.1   Observation

The **observation** is a composite object that contains 3 nested objects:

(a) : Torso

(b) : Forearm

(c) : Hand
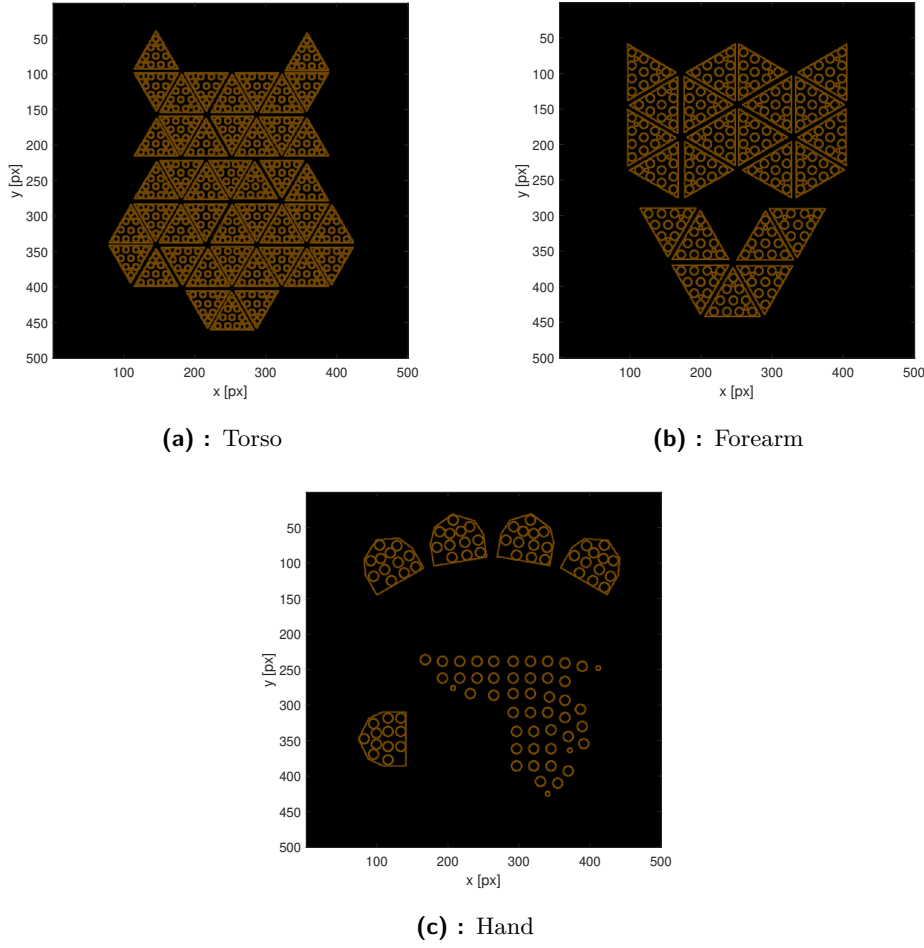
Figure 4.1: Visualization of tactile data in iCubSkinGUI module

1. *joints* object contains the actual observed state of all joints in the robot's head, torso, arms, and hands:

```
"joints": Dict({
  "head": Box(6),
  "left_arm": Box(16),
  "right_arm": Box(16),
  "torso": Box(3)
})
```

2. *skin* object contains vectors of binary values corresponding to each artificial skin taxel, where 0 indicates *no touch* and 1 indicates *touch*. There are 7 vectors in this object: for the torso, and for the left/right upper arm, forearm, and hand.

```
"skin": Dict({
```

```
  "left_arm": Box(768),
  "left_forearm": Box(384),
  "left_hand": Box(192),
  "right_arm": Box(768),
  "right_forearm": Box(384),
  "right_hand": Box(192),
  "torso": Box(768)
})
```

3. *touch* object contains coordinates of the touch for each of the body parts. These coordinates are computed using *center of mass* formula:

$$\vec{x} = \frac{\sum \vec{x}_i m_i}{\sum m_i} \tag{4.1}$$

Where the mass of each point is taken to be the pressure exerted on the corresponding taxel. If there is no touch on some body part, it returns $\vec{0}$ vector. 2D coordinates of individual taxels are deduced from the images used for visualization by the `iCubSkinGUI` interface (Fig. 4.1).

```
"touch": Dict({
  "left_arm": Box(2),
  "left_forearm": Box(2),
  "left_hand": Box(2),
  "right_arm": Box(2),
  "right_forearm": Box(2),
  "right_hand": Box(2),
  "torso": Box(2)
})
```

■ 4.2.2 Actions

The **action** object contains target position values for each joint in the robot's upper body. The shape of the `action` object is the same as the `joints` observation object described above. The limits for each joint are taken from the `URDF` model of the iCub robot. The **reset** action sets default movement speed for all joints and sends an action to move joints to the *home posture*.

■ 4.2.3 Reward function

Since the goal of this work is to find effective exploration algorithms for the skin surface, the **reward** function of the environment returns 1 for every

newly discovered taxel. During the reset, a Boolean lookup table is initialized with `false` value for each taxel. After each action, the taxels that were activated during the current time step are checked against the lookup table, and for each taxel that was not activated before the reward is increased by 1.

4.2.4 Episode termination

In accordance with the experiments conducted in our previous research, the episode is **terminated** after 1000 time steps.

Chapter 5

# Experiments and results

5.1 Body surface exploration

A number of experiments were conducted with the main goal to assess and compare several exploration strategies, forward and inverse models, and to test if the artificial skin simulation works properly.

5.1.1 Experimental design

Experiments were conducted in the simulated environment using Gazebo 11 simulator. Each experiment started with the selection of the exploration strategy, forward and inverse models.

All experiments were performed using the left arm of the simulated iCub robot, with the goal to explore the artificial skin on the robot's torso. The last 9 degrees of freedom of the arm were locked in a position with the thumb protruding outwards perpendicular to the palm of the robot's hand.

Therefore, the exploration was performed using the first 7 DoF of the robot's arm (shoulder, elbow, and wrist). Exploration always starts from the designated *home position* (Fig. 5.1) with the following joint coordinates for the left arm: `[-60,40,80,105,-60,24,0,60,90,0,0,0,0,0,0,0]`
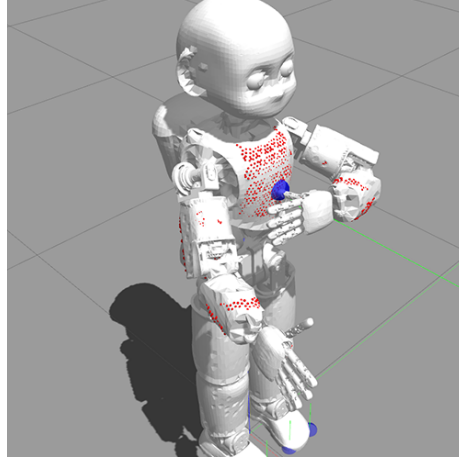
(a) : iCub robot in the home position
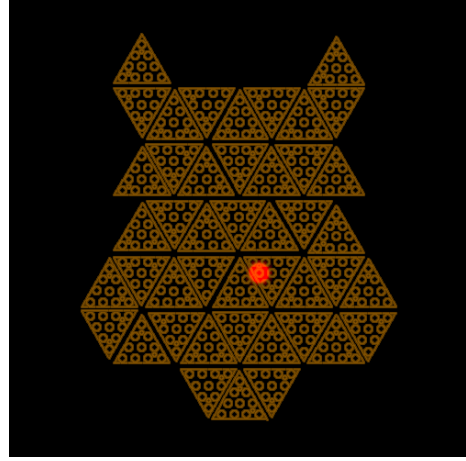
(b) : Torso skin taxel activation when in home position, the observation is $(276, 306)$[px] (see Section 4.2.1)

Figure 5.1: The home position

Each experiment was conducted for 1000 time steps, in accordance with our previous research [46, 15, 16]. During the exploration phase, zero-mean normally distributed exploration noise with $\sigma = 0.03$ is added to the inverse predictions of the model.

Every 100 time steps, the current state of the model is saved and the competence of the model is assessed on the testing taxel set. During the assessment stage, no exploratory noise is added to the model predictions.

▪ 5.1.2 Training and testing taxel sets

The exploration was performed within the artificial skin on the iCub's torso. Skin on the torso consists of 44 triangles. Each triangle houses 10 taxels, for a total of 440 taxels on the torso. The entire surface of the skin can be considered the training set.

For the testing set, I have selected the central taxel of each triangle. Therefore, the testing set on the torso consists of 44 taxels (Fig. 5.2a).

Every 100 iterations of the exploration algorithm the robot would try to reach for the taxels in the testing set, and the results of the evaluation were recorded for further processing:

(a) : Testing grid on iCub's torso   **(b) :** Testing grid on iCub's forearm
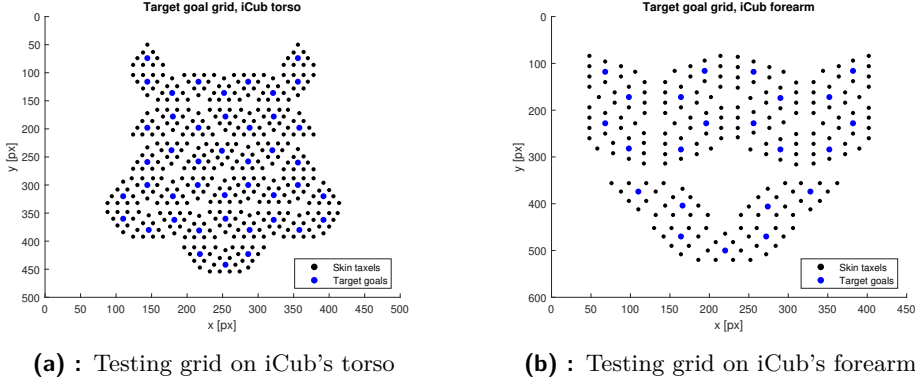
Figure 5.2: The testing taxel set

1. Goal taxels x_g

2. Actual observations x_o

3. Reaching error, computed as Euclidean norm between the goal taxel coordinates in observation space and the actual observation:

$$e_r = \|\vec{x}_g - \vec{x}_o\|_2 = \sqrt{(x_g^x - x_o^x)^2 + (x_g^y - x_o^y)^2} \tag{5.1}$$

Additionally, the total number of taxels that were touched at least once during the episode is recorded.

5.1.3 Random motor babbling

Random motor babbling is the weakest of the tested exploration strategies. On average, random motor babbling produced less than 10 touches during an experiment run of 1000 time steps (Fig. 5.5b).

1. It does not use feedback from the environment to adapt the exploration process in any way.

2. It suffers greatly from *the curse of dimensionality*, i.e. the exponential growth of the complexity of the exploration task with the growing number of degrees of freedom.

3. The portion of the action space where touches with the artificial skin can be observed is small compared to the whole volume. Random motor babbling produces almost no touch events (Fig. 5.3), which results in very poor data efficiency and quality of learned models.

23
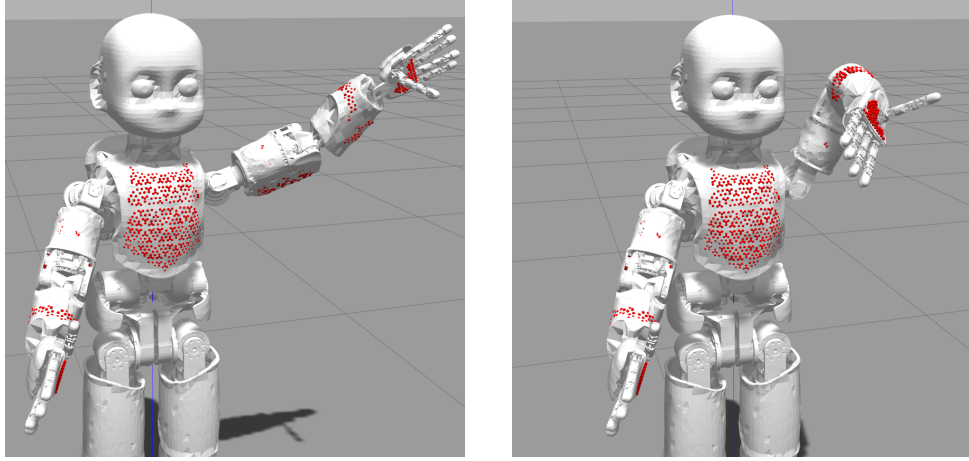
Figure 5.3: Examples of poses produced with random motor babbling exploration strategy

5.1.4　Goal-based exploration

In goal-based exploration paradigm, the agent actively generates goals in the observation space and tries to reach them. After each goal reaching attempt, the agent incrementally updates the internal model. This approach achieves great sample efficiency and allows for efficient exploration of the body surface.

A small drawback of the goal-based exploration techniques is the *bootstrapping problem*. In order to try and reach a goal, the agent must already have some crude inverse model available to him. I achieve bootstrapping by explicitly initializing the agent with the information about the home position and the corresponding observation (Fig. 5.1).

5.1.5　Random goal babbling

In the random goal babbling strategy (Section 3.6.3), the agent generates the goals randomly in the observation space and then tries to reach them using the inverse model. This strategy turns out to be much more effective than random motor babbling (Fig. 5.5).

(a) : Random goal babbling   **(b) :** Exploration by disagreement
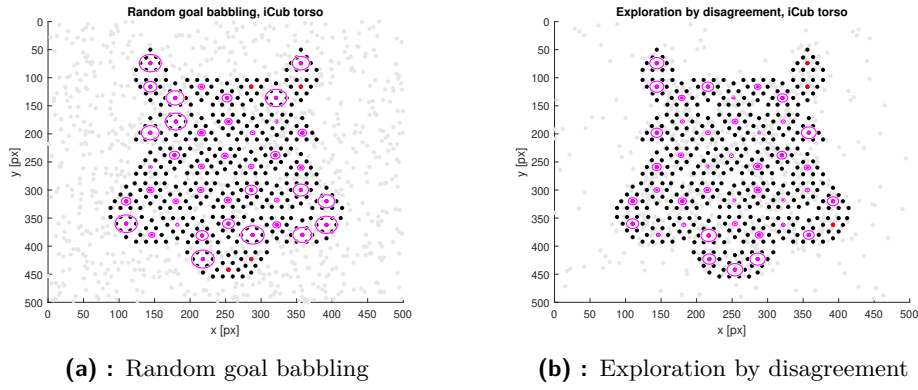
Figure 5.4: Qualitative results of the exploration experiments after 1000 time steps. Light gray: goals. Black: regular taxels. Magenta: testing taxels with error/3. Red: unreached taxels.
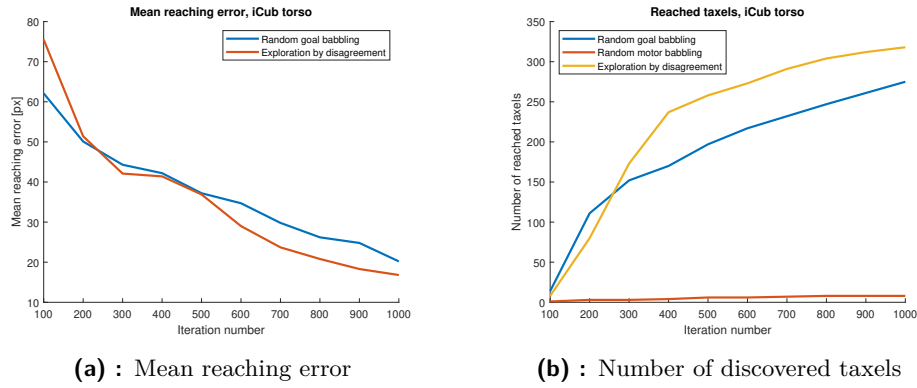
5.1.6 Exploration via disagreement

I have adapted exploration by disagreement (Section 3.5.2) to the problem of body surface exploration in the following way. I maintain a committee of 5 forward models initialized with different random weights. At each time step, the following algorithm is used:

1. Sample several random goals from the observation space.

2. For each sampled goal, query the inverse model for the action to reach that goal.

3. Use the committee of forward models to obtain estimates for each action.

4. Select the goal for which the variance in forward estimates is the largest.

In this way, exploration is steered toward the areas where the forward models disagree the most.

5.1.7 Summary of the exploration experiments

Qualitative results of the experiments are shown in Fig. 5.4. Results for random motor babbling are not shown here due to the extremely low number of taxel activations and poor quality of the learnt models.

(a) : Mean reaching error **(b) :** Number of discovered taxels

**Figure 5.5:** Quantitative results of the exploration experiments

As you can see, the random goal babbling exploration strategy (Fig. 5.4a) is less effective than exploration by disagreement (Fig. 5.4b). The latter focuses exploration on areas where there is "more to learn", which results in an overall better exploration performance.

Quantitative results are shown in Fig. 5.5. These results confirm that random motor babbling is practically unusable in high DoF exploration tasks, random goal babbling provides a feasible solution. However, intrinsically motivated exploration strategies, like exploration by disagreement, provide the best results, both in terms of minimizing the reaching error and discovering the larger skin surface area.

▌ 5.2 Tests of reading of sensor activation

During the development of the artificial skin simulation and when running the exploration experiments, I have monitored the quality and stability of the sensor activation. Unfortunately, I did not come up with any automated way of testing, instead I had to rely on visual inspection of `iCubSkinGUI` visualization of skin touches.

As stated above, I was able to achieve good stability with almost no flickering of the contact. Video demonstration of the current state of contact activation is available online at [47].
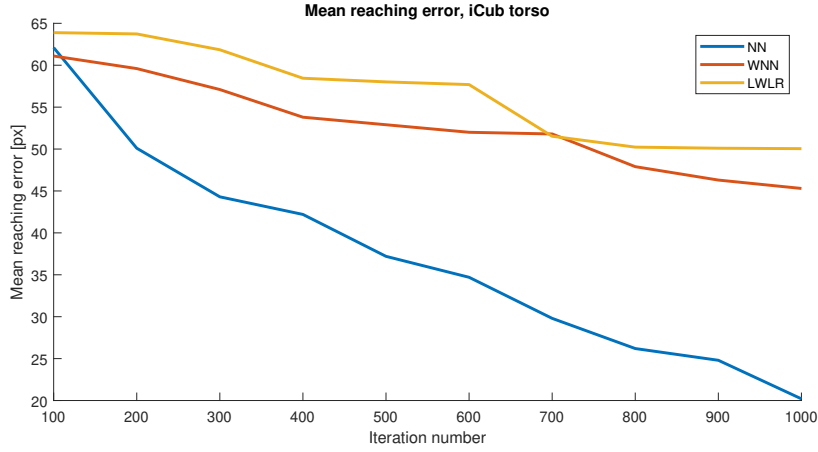
Figure 5.6: Comparison of mean reaching error for different models

5.3 Comparison of inverse models

To compare different inverse models, I have used the random goal babbling setting. Below is a brief description of each model and the numerical results of their comparison. All of these models are non-parametric *lazy learning* models. They maintain a database of tuples (q, x), where q is a motor command and x is a corresponding observation. When a forward or inverse query is performed, these models find 1 or more closest points, perform online processing, and return the estimated result.

1. **Nearest neighbor** (NN) model simply finds and returns the point closest to the query point.

2. **Weighted nearest neighbor** (WNN) model finds n nearest neighbors of the query and returns their average weighted by the distance to the query point.

3. **Locally weighted linear regression** (LWLR) model computes a linear regression of the n nearest neighbors of the query point.

Numerical results of comparing these 3 models (Fig. 5.6) confirm the findings from my previous research [46]. Although WNN and LWLR models are more effective than NN in the Cartesian manipulation tasks, NN model is more effective for skin exploration, because regression and weighted average often produce motor commands that result in no touch between the finger and the skin surface.

Chapter 6

Conclusion

In this thesis, I have developed an artificial skin simulator for the iCub humanoid robot in Gazebo 11. The simulator is finished, tested, and I continue working on a pull request to integrate the simulator into the official iCub codebase on GitHub [21]. This work will be beneficial to the research community working in the area of cognitive developmental robotics, enabling them to incorporate tactile sensory modalities into their simulated experiments.

After developing the simulator, I have implemented an OpenAI gym environment to work with the artificial skin simulation and performed a series of experiments in exploration of the body surface using the sensation of touch. I have adapted the exploration by disagreement framework to this task and compared the obtained results with our previous work [46, 15, 16]. With this, I have mainly confirmed our previous findings:

- Exploration by goal babbling is more effective than motor babbling, and intrinsically motivated exploration is more effective than goal babbling.

- When working with the sensation of touch, the primitive nearest neighbor inverse model works better than more complicated models based on regression, because it more often returns estimates that generate touch between the end effector and the surface of the skin.

All code, data files and experiment results are available online in the associated GitLab repository [18]. Video demonstrations of the artificial skin simulator are available online at [47, 48].

Chapter 7

Discussion and future work

7.1 Pending pull request from robotology github organization

In this thesis, I have added the support of tactile artificial skin to the Gazebo simulation of the iCub robot. This functionality was previously not available in the open source iCub codebase. I am currently in the process of preparing and testing a pull request to be accepted into `icub-main` and `icub-models` repositories that will add the abovementioned artificial skin functionality to the iCub codebase.

Once completed and accepted, this pull request will make it easier for other researchers to perform experiments in developmental and cognitive robotics that involve the sensation of touch.

7.2 Mimicking tactile receptor distribution in humans

For the artificial skin simulation implemented in this thesis, I have used 3D coordinates of taxels exported from the iCub CAD designs. Thus, the simulated artificial skin, which mirrors the artificial skin of the physical iCub

robot, has the uniform density of taxels. This density is dictated by the placement of capacitive sensors on PCB triangles.

This is nothing like the human anatomy. Human skin is covered with touch receptors called mechanoreceptors that provide us the sense of touch [51]. The density of these receptors varies in a wide range, with the most dense sensor coverage on the face and on the hands, reaching 240 receptors per cm² [51]. Grating orientation discrimination tests on humans give threshold values of 0.51 mm at the lip, 0.58 mm at the tongue, and 0.94 mm at the finger [52].

Yamada et al. have implemented an anatomically correct distribution of tactile sensors in their embodied model of a human foetus [54] (although their density data is based on less accurate human two-point discrimination data; see [52] for details).

I suggest it would be beneficial to create a model of artificial skin for the iCub robot with an option to choose between two distributions of the tactile sensor density: uniform or one that more closely resembles human anatomy.

7.3 Utilizing more degrees of freedom of the iCub robot

Similar to our previous research [46, 15, 16], I have used one of the iCub's fingers as a rigid end-effector to generate touch events. This approach was necessary for the Nao humanoid robot, however, iCub has an anthropomorphic arm and hand design with 5 fingers controlled by 9 degrees of freedom. For that reason, the iCub robot is capable of accomplishing much more agile manipulation tasks.

7.4 Combining different types of forward models when exploring via disagreement

Pathak et al. in[35] have used an ensemble of neural network forward models initialized with random weights. In this thesis, I have also used an ensemble

of homogeneous linear regression models initialized with random values. It seems to me that it would be worthwhile to investigate an exploration by disagreement process where forward models would be of different kinds, for example, a combination of several linear regressors with several neural networks.

7.5 Implementing the discretized goal babbling exploration strategy

In our previous work [46, 15, 16], we have achieved the best results in exploration when using the discretized goal babbling (DGB) exploration framework. Unfortunately, I was not able to implement this strategy in the new iCub environment. In the future, I would like to finish this task and compare DGB results with exploration by disagreement.

7.6 Technical issues with the simulation

When working on the OpenAI gym environment, I came across a minor technical difficulty. Sometimes, possibly if the simulated iCub robot exerts an effort larger than some threshold value, some of its joints will signal "hardware failure" and switch to uncontrollable idle compliant mode. During the entire development process this happened only a few times, and I have mitigated this by limiting the maximum speed of the joints, however this remains an issue I neither fully understand nor know how to fix.

Bibliography

[1] ATKESON, C. G., MOORE, A. W., AND SCHAAL, S. Locally weighted learning for control. In *Lazy learning*. Springer, 1997, pp. 75–113.

[2] AUBRET, A., MATIGNON, L., AND HASSAS, S. A survey on intrinsic motivation in reinforcement learning. *arXiv preprint arXiv:1908.06976* (2019).

[3] AYALA, A., CRUZ, F., CAMPOS, D., RUBIO, R., FERNANDES, B., AND DAZELEY, R. A comparison of humanoid robot simulators: A quantitative approach. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (2020), IEEE, pp. 1–6.

[4] BARANES, A., AND OUDEYER, P.-Y. Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), IEEE, pp. 1766–1773.

[5] BARANES, A., AND OUDEYER, P.-Y. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems 61*, 1 (2013), 49–73.

[6] BARTO, A. G. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*. Springer, 2013, pp. 17–47.

[7] BARTOLOZZI, C., NATALE, L., NORI, F., AND METTA, G. Robots with a sense of touch. *Nature Materials 15*, 9 (2016), 921–925.

[8] BROCKMAN, G., CHEUNG, V., PETTERSSON, L., SCHNEIDER, J., SCHULMAN, J., TANG, J., AND ZAREMBA, W. OpenAI gym. *arXiv preprint arXiv:1606.01540* (2016).

[9] BURDA, Y., EDWARDS, H., PATHAK, D., STORKEY, A. J., DARRELL, T., AND EFROS, A. A. Large-scale study of curiosity-driven learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (2019), OpenReview.net.

[10] CHEN, J., DENG, H., CHAI, W., XIONG, J., AND XIA, Z. Manipulation task simulation of a soft pneumatic gripper using ROS and Gazebo. In *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)* (2018), IEEE, pp. 378–383.

[11] CHURCH, A., LLOYD, J., HADSELL, R., AND LEPORA, N. F. Deep reinforcement learning for tactile robotics: Learning to type on a braille keyboard. *IEEE Robotics and Automation Letters 5*, 4 (2020), 6145–6152.

[12] COLLINS, J., CHAND, S., VANDERKOP, A., AND HOWARD, D. A review of physics simulators for robotic applications. *IEEE Access* (2021).

[13] DAHIYA, R., AKINWANDE, D., AND CHANG, J. S. Flexible Electronic Skin: From Humanoids to Humans [Scanning the Issue]. *Proceedings of the IEEE 107*, 10 (2019), 2011–2015.

[14] DAHIYA, R., METTA, G., VALLE, M., AND SANDINI, G. Tactile sensing – from humans to humanoids. *IEEE Transactions on Robotics 26*, 1 (2010), 1–20.

[15] GAMA, F., SHCHERBAN, M., ROLF, M., AND HOFFMANN, M. Active exploration for body model learning through self-touch on a humanoid robot with artificial skin. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (2020), IEEE, pp. 1–8.

[16] GAMA, F., SHCHERBAN, M., ROLF, M., AND HOFFMANN, M. Goal-directed tactile exploration for body model learning through self-touch on a humanoid robot. *IEEE Transactions on Cognitive and Developmental Systems (TCDS)* (Accepted), 1–15.

[17] Github issue regarding ODE bug causing flickering contacts in Gazebo. `https://github.com/osrf/collision-benchmark/issues/6`. [Online; accessed 06-08-2021].

[18] GitLab repository with code developed for this thesis. `https://gitlab.fel.cvut.cz/body-schema/code-icub-gazebo-skin`. [Online; accessed 11-08-2021].

[19] HOFFMAN, E. M., TRAVERSARO, S., ROCCHI, A., FERRATI, M., SETTIMI, A., ROMANO, F., NATALE, L., BICCHI, A., NORI, F., AND

TSAGARAKIS, N. G. YARP based plugins for Gazebo simulator. In *International Workshop on Modelling and Simulation for Autonomous Systems* (2014), Springer, pp. 333–346.

[20] iCub Joints - iCub Tech Docs. `https://icub-tech-iit.github.io/documentation/icub_kinematics/icub-joints/icub-joints/`. [Online; accessed 06-08-2021].

[21] iCub Models. `https://github.com/robotology/icub-models`. [Online; accessed 06-08-2021].

[22] iCub Tactile Sensors (iCub Skin). `http://wiki.icub.org/wiki/Tactile_sensors_(aka_Skin)`. [Online; accessed 06-08-2021].

[23] iCub with exposed skin. `https://rsdahiya.com/research/projects/`. [Online; accessed 06-08-2021].

[24] KLYUBIN, A. S., POLANI, D., AND NEHANIV, C. L. Empowerment: A universal agent-centric measure of control. In *2005 IEEE Congress on Evolutionary Computation* (2005), vol. 1, IEEE, pp. 128–135.

[25] KOENIG, N., AND HOWARD, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2004), vol. 3, IEEE, pp. 2149–2154.

[26] LEPORA, N., AND LLOYD, J. Pose-based tactile servoing: Controlled soft touch using deep learning. *IEEE Robotics & Automation Magazine* (2021).

[27] LLOYD, J., AND LEPORA, N. F. Goal-driven robotic pushing using tactile and proprioceptive feedback. *IEEE Transactions on Robotics* (Accepted).

[28] MANNELLA, F., SOMOGYI, E., JACQUEY, L., O'REGAN, K., BALDASSARRE, G., ET AL. Know your body through intrinsic goals. *Frontiers in Neurorobotics 12* (2018), 30.

[29] METTA, G., FITZPATRICK, P., AND NATALE, L. YARP: yet another robot platform. *International Journal of Advanced Robotic Systems 3*, 1 (2006), 8.

[30] METTA, G., SANDINI, G., VERNON, D., NATALE, L., AND NORI, F. The iCub humanoid robot: an open platform for research in embodied cognition. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems* (2008), pp. 50–56.

[31] MORI, H., MASUDA, M., AND OGATA, T. Tactile-based curiosity maximizes tactile-rich object-oriented actions even without any extrinsic rewards. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (2020), IEEE, pp. 1–7.

[32] MOULIN-FRIER, C., ROUANET, P., AND OUDEYER, P.-Y. Explauto: An open-source python library to study autonomous exploration in developmental robotics. In *4th International Conference on Development and Learning and on Epigenetic Robotics* (2014), IEEE, pp. 171–172.

[33] OUDEYER, P.-Y., KAPLAN, F., AND HAFNER, V. V. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation 11*, 2 (2007), 265–286.

[34] PATHAK, D., AGRAWAL, P., EFROS, A. A., AND DARRELL, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning* (2017), PMLR, pp. 2778–2787.

[35] PATHAK, D., GANDHI, D., AND GUPTA, A. Self-supervised exploration via disagreement. In *International Conference on Machine Learning* (2019), PMLR, pp. 5062–5071.

[36] RAYYES, R., DONAT, H., AND STEIL, J. Efficient online interest-driven exploration for developmental robots. *IEEE Transactions on Cognitive and Developmental Systems* (2020).

[37] ROLF, M. *Goal babbling for an efficient bootstrapping of inverse models in high dimensions*. PhD thesis, CoR-Lab, Bielefeld University, 2012.

[38] ROLF, M., AND STEIL, J. J. Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. *IEEE Transactions on Neural Networks and Learning Systems 25*, 6 (2013), 1147–1160.

[39] RONCONE, A., HOFFMANN, M., PATTACINI, U., AND METTA, G. Automatic kinematic chain calibration using artificial skin: self-touch in the iCub humanoid robot. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), pp. 2305–2312.

[40] RUSTLER, L., POTOCNA, B., POLIC, M., STEPANOVA, K., AND HOFFMANN, M. Spatial calibration of whole-body artificial skin on a humanoid robot: comparing self-contact, 3d reconstruction, and cad-based calibration. In *Humanoid Robots (Humanoids), IEEE-RAS International Conference on* (2021), pp. 445–452.

[41] S. SURESH, M. BAUZA, K.-T. YU, J. MANGELSON, A. RODRIGUEZ, AND M. KAESS. Tactile SLAM: Real-time inference of shape and pose from planar pushing. In *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA* (May 2021).

[42] SCHMIDHUBER, J. Curious model-building control systems. In *Proc. International Joint Conference on Neural Networks* (1991), pp. 1458–1463.

[43] SCHMITZ, A., MAGGIALI, M., NATALE, L., AND METTA, G. Touch sensors for humanoid hands. In *19th International Symposium in Robot and Human Interactive Communication* (2010), IEEE, pp. 691–697.

[44] SETTLES, B. *Synthesis Lectures on Artificial Intelligence and Machine Learning.* Morgan & Claypool Publishers, 2012.

[45] SEUNG, H. S., OPPER, M., AND SOMPOLINSKY, H. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory* (1992), pp. 287–294.

[46] SHCHERBAN, M. Efficient self-exploration and learning of forward and inverse models on a Nao humanoid robot with artificial skin. Bachelor's thesis, Faculty of Electrical Engineering, Czech Technical University in Prague, 2019.

[47] Simulated skin contact, port update rate at 20 Hz. `https://www.youtube.com/watch?v=RAdbk1aOJFY`. [Online; accessed 12-08-2021].

[48] Simulated skin contact, port update rate at 50 Hz. `https://www.youtube.com/watch?v=xccVWibz_KQ`. [Online; accessed 12-08-2021].

[49] STRUCKMEIER, O., TIWARI, K., SALMAN, M., PEARSON, M. J., AND KYRKI, V. ViTa-SLAM: A bio-inspired visuo-tactile SLAM for navigation while interacting with aliased environments. In *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)* (2019), IEEE, pp. 97–103.

[50] SUKHBAATAR, S., LIN, Z., KOSTRIKOV, I., SYNNAEVE, G., SZLAM, A., AND FERGUS, R. Intrinsic motivation and automatic curricula via asymmetric self-play. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* (2018), OpenReview.net.

[51] SUNDARAM, S. How to improve robotic touch. *Science 370*, 6518 (2020), 768–769.

[52] VAN BOVEN, R. W., AND JOHNSON, K. O. The limit of tactile spatial resolution in humans: grating orientation discrimination at the lip, tongue, and finger. *Neurology 44*, 12 (1994), 2361–2361.

[53] WARD-CHERRIER, B., PESTELL, N., CRAMPHORN, L., WINSTONE, B., GIANNACCINI, M. E., ROSSITER, J., AND LEPORA, N. F. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft robotics 5*, 2 (2018), 216–227.

[54] YAMADA, Y., KANAZAWA, H., IWASAKI, S., TSUKAHARA, Y., IWATA, O., YAMADA, S., AND KUNIYOSHI, Y. An embodied brain model of the human foetus. *Scientific reports 6*, 1 (2016), 1–10.

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Shcherban Maksym**   Personal ID number: **469834**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

Branch of study: **Robotics**

II. Master's thesis details

Master's thesis title in English:

Efficient Exploration of Body Surface with Tactile Sensors on Humanoid Robots

Master's thesis title in Czech:

Efektivní průzkum povrchu těla s taktilními senzory u humanoidních robotů

Guidelines:

1. Assess the suitability of different simulators for simulating and perceiving continuous touch. Examples: Gazebo, OpenSim, Mujoco, NVIDIA Isaac Gym.
2. Choose one of the simulators and develop a model of the iCub humanoid robot with two options for skin sensors:
◦ Mimicking the sensors on the real robot.
◦ Mimicking the tactile receptor distribution in humans.
3. Test the functionality of reading of sensor activations with special focus on continuous self-contact.
4. With the goal of the robot to efficiently explore its skin surface using self-contact, compare the effectiveness of variants of active / intrinsically motivated exploration in motor and goal spaces (Baranes and Oudeyer, 2013; Rolf and Steil, 2013; Gama et al. 2020) with curiosity-driven learning (Burda et al., 2019; Mori et al., 2020).
5. Explore different methods for constructing forward and inverse models (Nearest Neighbor, weighted NN, Locally Weighted Linear Regression, etc.) - the input space consists of the robot's joints, the goal space will be the robot's artificial skin.

Bibliography / sources:

[1] Baranes, Adrien and Oudeyer, Pierre-Yves. "Active learning of inverse models with intrinsically motivated goal exploration in robots." Robotics and Autonomous Systems 61.1 (2013): 49-73.
[2] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," in ICLR, 2019.
[3] Gama, F., Shcherban, M., Rolf, M. and Hoffmann, M. "Active exploration for body model learning through self-touch on a humanoid robot with artificial skin", Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob). IEEE, 2020.
[4] Mori, H.; Masuda, M. & Ogata, T. (2020), Tactile-based curiosity maximizes tactile-rich object-oriented actions even without any extrinsic rewards, in Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob).
[5] Rolf, M., & Steil, J. J. (2013). Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. IEEE transactions on neural networks and learning systems, 25(6).

Name and workplace of master's thesis supervisor:

Mgr. Matěj Hoffmann, Ph.D.,   Vision for Robotics and Autonomous Systems,   FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **06.01.2021**   Deadline for master's thesis submission: **13.08.2021**

Assignment valid until: **30.09.2022**

_____
Mgr. Matěj Hoffmann, Ph.D.
Supervisor's signature

_____
prof. Ing. Tomáš Svoboda, Ph.D.
Head of department's signature

_____
prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

_____._____
Date of assignment receipt

_____
Student's signature