Bachelor Thesis

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

# Efficient self-exploration and learning of forward and inverse models on a humanoid robot with artificial skin

Maksym Shcherban

Supervisor: Mgr. Matěj Hoffmann, Ph.D.
Supervisor–specialist: doc. Ing. Karel Zimmermann, Ph.D.
Field of study: Robotics
Subfield: Cybernetics and robotics
May 2019

# Acknowledgements

I would like to express my deepest gratitude to Czech Technical University in Prague for the opportunity to get world-class education.

I would like to thank my thesis supervisor Mgr. Matěj Hoffmann, Ph.D. for providing an interesting research topic and for putting up with constant delays on my part.

I also want to thank my mother and my wife who supported me in my pursuit of knowledge and education. I love you.

# 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, 24. May 2019

# Abstract

A model of NAO humanoid robot was modified and artificial skin with touch-sensitive taxels was added to it. Simulation environment based on ROS and Gazebo physics simulator was developed. In this environment, a series of experiments in self-exploration inspired by research in developmental psychology was conducted. The goal was to learn forward and inverse models of robot's body using self touch with artificial skin as the only source of sensory feedback. Explauto library was used for the exploration and model learning. As little prior information about the structure of robot's kinematic chain was provided to exploration algorithms: only number of joints used in the experiment was known. Several forward models, inverse models and exploration strategies available in explauto library were tested and compared. Mean reaching error over several goals on robot's body was used as an empirical measure of the quality of learnt models. Results of the conducted experiments show that out of all options available in explauto, nearest neighbor model produces smallest mean reaching error. Results also show that goal-based exploration strategies are better than strategies based on motor babbling. They converge faster and allow learning of models with smaller reaching error. Among all exploration strategies available in explauto, fixed discretization of observation space produced best results. This strategy focuses exploration in regions of most interest, so that amount of new information added to the model is maximized. Some possibilities open for further investigation and research are discussed in the closing chapter of this work.



**Keywords:** artificial skin, humanoid robot, body representation, goal babbling, motor babbling, forward model, inverse model, exploration, intrinsic motivation, developmental robotics, ROS, Gazebo, NAO



**Supervisor:** Mgr. Matěj Hoffmann, Ph.D.
Praha, Resslova 307/9, místnost: E-211

# Abstrakt

Upravili jsme model humanoidního robota NAO v prostředí založeném na ROS a fyzickém simulátoru Gazebo tím, že jsme přidali umělou kůži citlivou na dotek. V tomto prostředí byla provedena série experimentů inspirovaných výzkumem vývojové psychologie v oblasti zkoumání vlastního těla. Cílem bylo naučit se dopředné a inverzní modely těla robota pomocí sebedotyku s umělou kůží jako jediným zdrojem zpětné vazby. Pro průzkum a učení modelů byla použita knihovna explauto. Cílem bylo využívat co nejméně informace o struktuře kinematického řetězce robota: byl znám pouze počet kloubů použitých v experimentu. Bylo testováno a porovnáno několik dopředných modelů, inverzních modelů a průzkumných strategií dostupných v knihovně explauto. Jako empirická míra kvality naučených modelů byla použita průměrná chyba dosažení několika cílů na těle robota. Výsledky provedených experimentů ukazují, že ze všech možností dostupných v knihovně explauto, poskytuje model nejbližšího souseda (nearest neighbor) nejmenší střední chybu dosažení. Výsledky také ukazují, že průzkumné strategie založené na dosažení cílů (goal babbling) jsou lepší než strategie založené na motor babbling. Konvergují rychleji a umožňují učení modelů s menší chybou. Ze všech výzkumných strategií, které jsou k dispozici v knihovně explauto, bylo dosaženo nejlepších výsledků pomocí pevné diskretizace pozorovacího prostoru. Tato strategie se zaměřuje na zkoumání v regionech, které jsou nejzajímavější, takže množství nových informací přidaných do modelu je maximalizováno. V závěrečné kapitole této práce jsou diskutovány některé možnosti dalšího výzkumu.



**Klíčová slova:** umělá kůže, humanoidní robot, reprezentace těla, goal babbling, motor babbling, dopředný model, explorace, inverzní model, vývojová robotika, ROS, Gazebo, NAO


**Překlad názvu:** Efektivní sebe-explorace a tvorba modelu těla u humanoidního robota s umělou kůží

# Contents

**1 Introduction** **1**

**2 Related work** **3**

**3 Methods** **5**

3.1 NAO humanoid robot . . . . . . . . . . 5

3.2 Artificial skin . . . . . . . . . . . . . . . . . . 6

3.3 Simulation environment . . . . . . . . 7

3.4 Exploration framework . . . . . . . . . 9

3.4.1 Explauto library . . . . . . . . . . . . 9

3.4.2 Action and observation spaces 10

3.4.3 Forward and inverse models . 16

3.4.4 Exploration strategies . . . . . . 18

3.4.5 Handling motor commands that produce no observation . . . . . . . . . 20

**4 Results** **21**

4.1 Experimental design . . . . . . . . . . 21

4.1.1 Progress Evaluation . . . . . . . . 22

4.2 Comparison of inverse models . . 22

4.3 Comparison of exploration strategies . . . . . . . . . . . . . . . . . . . . . . 23

4.4 Discretized progress exploration strategy . . . . . . . . . . . . . . . . . . . . . . . 24

4.5 SAGG-RIAC exploration strategy 26

**5 Conclusion, Discussion, Future Work** **27**

5.1 Conclusion . . . . . . . . . . . . . . . . . . . 27

5.2 Discussion . . . . . . . . . . . . . . . . . . . 28

5.3 Future work . . . . . . . . . . . . . . . . . . 29

**Bibliography** **31**

**A Project Specification** **35**

# Chapter 1

# Introduction

Most modern robots consist of rigid links connected by joints of various types, e.g. linear, rotational, spherical etc. Dimensions of robot's links and limits of robot's joints are provided by the manufacturer in technical documentation and manuals, and robot's inverse and forward kinematic functions are available. However, there are several reasons for robots to perform self-exploration and to build models of their bodies autonomously:

- Although it is presumed that all robots of same make and model will have perfectly identical dimensions and hence kinematic equations, this is not true. Imprecisions and errors can be introduced at any point during fabrication and assembly of the robot. Dimensions of robot's rigid links can also change during operation, for example because of thermal expansion and contraction of materials, or because of mechanical wear and tear. Therefore, a robot must be calibrated before it can be put to use and periodically during its operation. One method for such calibration involves self-exploration of the robots body.
- Robot's body can undergo undesirable changes due to damage taken from environment. In scenarios where robot must continue its operation regardless of taken damage, it would be necessary for the robot to assess sustained damage by means of self-exploration and change the internal model of it's body accordingly.
- Other paradigms for construction of robot's body exist, for example soft robotics. Soft robots are inspired by living organisms. Instead of system of rigid links, bodies of soft robots are made of compliant materials, often actuated by pneumatics or hydraulics. Such robots do not have

> fixed body morphology. Their motion is harder to model, and it would be beneficial for compliant robots to be able to learn the model of their ever-changing body.

Several ways to perform self-exploration exist. One of such methods, investigated in this work, involves covering surface of robot's body with a layer of capacitive artificial skin that responds to touch.

A parallel can be drawn between a robot that knows nothing about structure of it's body and an infant. Therefore, research in developmental psychology can be applied to the problem of robot learning model of it's body by means of self-exploration. Research shows that sensation of touch plays important role in early infant developing the model of their body.

# Chapter 2

# Related work

A body of research in developmental psychology studies development of reaching behaviors and emergence of forward and inverse body models in infants. Corbetta et al. reviewed the literature on development of reaching behaviors in infants [7] to study the evolution of views on this problem. Von Hofsten [29] studied quantitatively how infants 12 to 18 weeks old approach stationary and moving objects. He analyzed relative length of reaching paths, approach time, acceleration of movements and concluded that reaching skill improves extensively and in predictable way during the studied period. Focusing on reaching to own body, Hoffmann et al. observed how infants between 3 and 21 months react to vibrotactile stimulation [17]. In their experiments, buzzers were connected to various parts of infant bodies and their reactions to stimulation from attached buzzers were recorded and analyzed. They have reported developmental progression from general to specific movement patterns, more so in the first year of infant's lifetime.

Artificial skin can be used in robotics for calibration of kinematic chains represented with the standard Denavit-Hartenberg parameters, as Roncone et al. did in [27]. In this work, I investigate different approach, inspired by developmental studies of infants. I make as few assumptions about the structure of the robot's kinematic chain as possible. Instead of learning parameters of the representation, I try to learn a non-parametric model of robot's body.

Rolf in his dissertation [26] conducted an exhaustive research of goal babbling exploration strategy. He demonstrated that goal-based exploration in observation space is more effective than exploration in action space based on motor babbling. He had also substantiated the necessity to add exploratory noise to exploration process in order to avoid singular configurations. He tested his ideas in several robotic morphologies: planar arm in 1D and 2D, humanoid robot and bionic elephant trunk in 3D. In all his experiments, the observation space directly corresponded to the position of robot's end effector in Euclidean space. In this work I have applied goal-based exploration strategies to efficient self-exploration of humanoid robot with artificial skin. Observation space in my experiments does not directly correspond to position in Euclidean space. Instead, it corresponds to artificial skin taxels activated by robot's movements.

Experiments similar to mine were conducted by Mannella et al. in [21]. However, they conducted their experiments on a planar robot in non-physical environment that only considered robot's kinematics. In contrast, I have conducted experiments in 3D environment that simulated robot's physics and geometry collisions.

Comprehensive study of intrinsically motivated exploration and active learning of inverse body models was done by Adrien Baranes, Pierre-Yves Oudeyer and Clément Moulin-Frier in [3], [2] and [22]. They introduce SAGG-RIAC (Self-Adaptive Goal Generation - Robust Intelligent Adaptive Curiosity) architecture for active learning of inverse models in redundant spaces. The main idea is to divide goal space into regions in a way that maximizes competence improvement for reaching those goals.

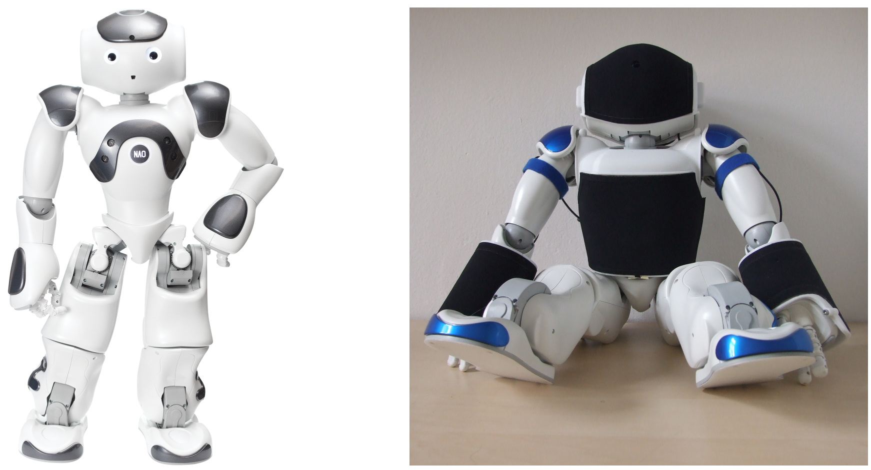
Explauto is an open source Python framework for active learning and exploration developed in the Inria FLOWERS research team [23]. Explauto implements numerous goal-based exploration strategies, several possibilities for representation of forward and inverse models, and provides tools for comparing quality of learnt models. All exploration processes and model learning in this work are based on explauto library.

Chapter 3

# Methods

## 3.1 NAO humanoid robot

NAO is an autonomous humanoid robot developed by Aldebaran Robotics primarily for education and research purposes [14]. NAO has height of 58 centimeters and weighs 5.5 kilograms (without modifications). NAO is powered by lithium battery which provides up to 90 minutes of autonomous operation. Robot's head is equipped with central processing unit and operating memory. In addition to that, NAO robot contains several custom designed integrated circuits based on Microchip 16 bit microcontrollers that are responsible for actuator servo-control, sensor and power management.



**(a) :** Front view of NAO robot

**(b) :** NAO with artificial skin

**Figure 3.1:** Aldebaran Robotics NAO humanoid robot.

| **Body Part** | **Number of sensors** (low resolution) | **Number of sensors** (high resolution) |
|---|---|---|
| Torso | 25 | 250 |
| Head | 24 | 240 |
| Left wrist | 27 | 270 |
| Right wrist | 27 | 270 |

**Table 3.1:** Number of artificial skin sensors per body part.

NAO has a total of 25 degrees of freedom (DOF). In this work, 12 DOF in the upper body of the NAO robot were utilized: 2 in the head and 5 in each of the arms. NAO is equipped with a variety of sensors (cameras, microphones, sonars, inertial measurement units, bumpers). However, these sensors are not utilized in this work.

## 3.2 Artificial skin

The artificial skin used in this project [20] is a network of capacitive touch and temperature sensors mounted on the robot's torso, head and wrists (Fig. 3.2). Artificial skin is constructed of interconnected triangular patches. Each patch contains 10 individual circular touch sensors and a microcontroller unit responsible for processing of sensory data and network communication.

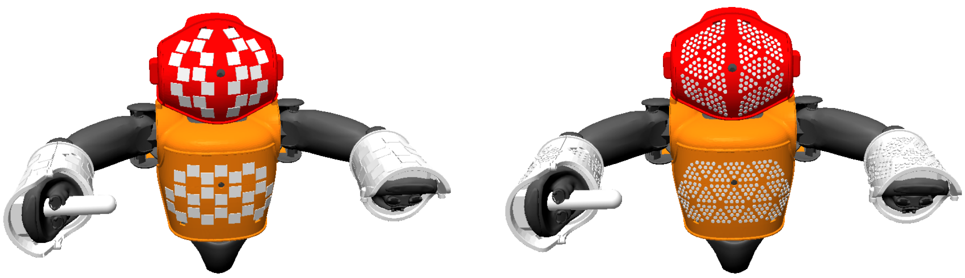
There are more than 1000 sensors on the robot's body (Tbl. 3.1). Individual touch sensors produce 8 bit output that indicates magnitude of pressure that is applied to the sensor. In this work, I model individual artificial skin sensors as binary devices with 1 bit output (on/off).

**(a) :** Physical artificial skin sensors

**(b) :** Artificial skin mounted on NAO robot

**Figure 3.2:** Artificial skin.

## 3.3 Simulation environment



**(a)** : Low resolution artificial skin  **(b)** : High resolution artificial skin

**Figure 3.3:** Model of NAO humanoid robot modified for the purpose of artificial skin simulation.

The simulation environment is based on ROS Melodic [25] and Gazebo 9 physics simulator [11]. Different aspects of the simulation are implemented in separate ROS nodes which communicate via ROS services and topics (Fig. 3.4). All program code developed for this project is available online in GitLab repository [12].

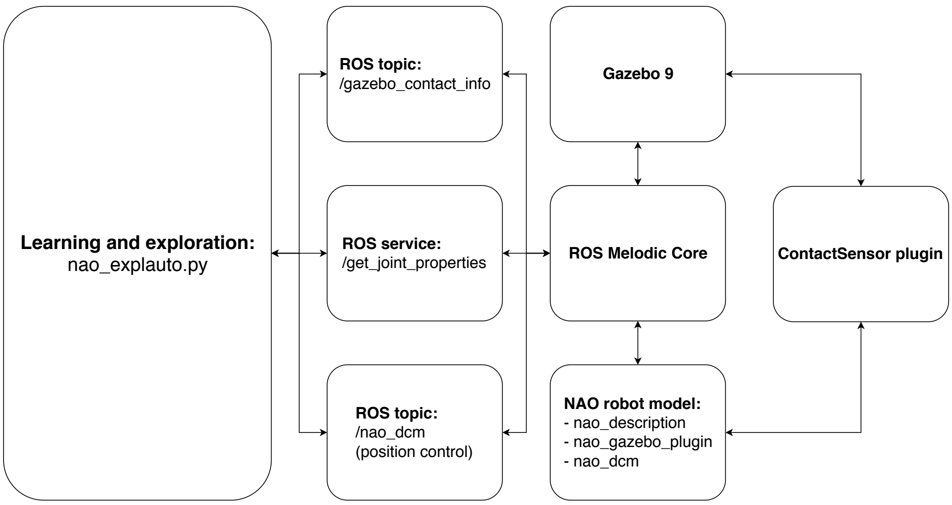



**Figure 3.4:** Essential ROS nodes, topics and services forming the backbone of simulation environment architecture.

A model of default NAO humanoid robot is available online in URDF format (Fig. 3.3). I have made a number of augmentations to that model:

- Robot's base torso link was fixed in space for the purpose of this simulation.
- Several parts of the robot's body were removed because they were unused in the simulation: legs, gripper, fingers, cameras, sonar sensors.
- New child links representing plastic casings that house the artificial skin were added to robot's torso, head and both wrists.
- Numerous contact sensors (taxels) were placed as child links of the plastic casings. Touch sensory feedback was enabled with Gazebo ContactSensor plugin (courtesy of Ing. Martin Jílek, jilekma1@fel.cvut.cz) adapted to support Gazebo 9 API.
- A cylindrical pen tool with spherical endpoint was attached to robot's wrist.

Process of exploration, model learning and evaluation is coordinated from `nao_explauto` ROS node written in Python. The node implements a custom environment within explauto framework [23]. It communicates with Gazebo simulation through 3 channels:

- By calling `/gazebo/get_joint_properties` ROS service, the node acquires information about current joint states of the simulated NAO humanoid robot.
- By subscribing to `/gazebo_contact_info` ROS topic, the node acquires information about current sensory feedback from the artificial skin.
- By publishing to `/nao_dcm/**joint**_position_controller/command` ROS topic, the node applies motor commands to the simulated robot.

| Joint name | Lower limit [rad] | Upper limit [rad] |
|---|---|---|
| `HeadPitch` | −0.67 | 0.51 |
| `HeadYaw` | −2.09 | 2.09 |
| `LShoulderRoll` | −0.31 | 1.33 |
| `LShoulderPitch` | −2.09 | 2.09 |
| `LElbowRoll` | −1.54 | −0.03 |
| `LElbowYaw` | −2.09 | 2.09 |
| `LWristYaw` | −1.82 | 1.82 |
| `RShoulderRoll` | −1.33 | 0.31 |
| `RShoulderPitch` | −2.09 | 2.09 |
| `RElbowRoll` | 0.03 | 1.54 |
| `RElbowYaw` | −2.09 | 2.09 |
| `RWristYaw` | −1.82 | 1.82 |

**Table 3.2:** Limits of joint values [19, p.20].

## 3.4 Exploration framework

### 3.4.1 Explauto library

Explauto is an open-source Python library developed in the **Inria FLOWERS** research team [23]. Explauto is a framework designed to study, model and simulate exploration and learning in robotic agents. High-level architecture of explauto framework (Fig. 3.5):

- The role of interest models is to provide goals for the sensorimotor model. An interest model implements the active exploration process.
- Sensorimotor model implements the iterative learning process from sensorimotor experience. It uses the internal model of robotic agent to perform forward and inverse predictions provided by the interest model.
- Sensorimotor system encapsulates physical properties of the interaction between the robot's body and the environment in which it evolves.

Explauto library provides several forward and inverse models, multiple motor- and goal-based exploration strategies and tools for evaluation and comparison of learnt models.

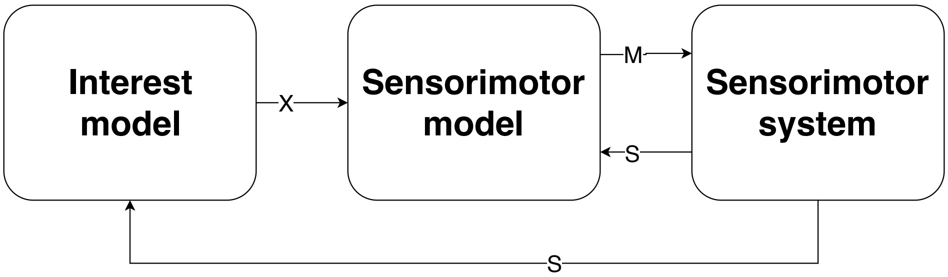



**Figure 3.5:** The Explauto framework architecture.

### 3.4.2 Action and observation spaces

*Action space* $\mathbf{Q}$ represents all possible actions of the robot. 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* $f$ [26, p.5]:

$$\begin{aligned} f : \mathbf{Q} &\to \mathbf{X} \\ f(q) &= x \end{aligned} \tag{3.1}$$

In explauto framework, exploration is performed in the *interest space*. Explauto framework defines interest space as an alias for either action space or observation space. If the interest space corresponds to action space, the exploration is performed using motor babbling strategies. If the interest space corresponds to observation space, the exploration is performed using goal babbling strategies.

Explauto framework uses the notion of *lazy learning*. Lazy learning methods defer processing of training data until a query needs to be answered [1]. During exploration process a database of attempted actions and corresponding observations is maintained. Forward and inverse models are constructed implicitly from this database. The models are therefore improved with every new entry added to the database. Entries in the database are tuples:

$$\begin{aligned} &(q, x) \\ q &\in \mathbf{Q} \\ x &\in \mathbf{X} \end{aligned} \tag{3.2}$$

An action $q \in \mathbf{Q}$ is defined in this work as a particular configuration of robot's joints. The upper body of NAO humanoid robot has a total of 12 degrees of freedom (DOF): 2 joints in the head and 5 joints in each arm. Joints have upper and lower limits on their angles (Tbl. 3.2), therefore in the scope of this work we can define action space $\mathbf{Q}$ of the robot as:

$$\mathbf{Q} \subseteq \mathbb{R}^{12} \tag{3.3}$$

Depending on the performed experiment, a subset of available joints is selected to constitute robot's action space. E.g. in an experiment that considers head and one of the arms, the action space would be $\mathbf{Q}_e \subseteq \mathbb{R}^7$.

It is important to note that the action space of the robot is non-convex, and the outcome in observation space depends not only on the executed action, but also on previous configuration of the robot. For example, consider the following scenario: robot's hand is located on the left side of robot's torso, and a command is executed that should move robot's hand to the right side of the torso. In that case, the hand will collide with torso and artificial skin on the left side of torso will be activated, producing wrong observation. This issue is overcome by periodically resetting the simulation to *home posture*, as suggested in [26, p.51].

Dimensionality and underlying structure of observation space $\mathbf{X}$ depends on the amount of prior information about artificial skin configuration. Three possibilities were investigated in this work:

**No prior information** about artificial skin configuration is available. The sensory feedback information is encoded as a set of binary values, each value representing a state of a single taxel (on/off). The observation space is discrete, its dimensionality is equal to the number of taxels $\mathbf{T}$ in the artificial skin, which grows quickly both with surface area of the skin and with its spatial resolution (number of taxels per unit area):

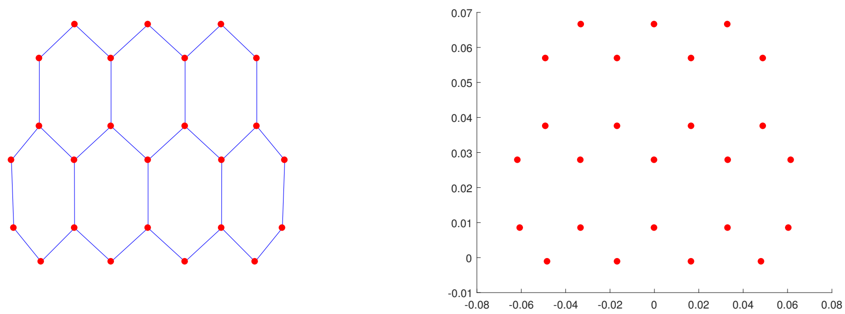
$$\mathbf{X}_0 = \{0, 1\}^{\mathbf{T}} \tag{3.4}$$

Although Euclidean distance between two points in observation space can be calculated in this representation, it doesn't provide any meaningful information, because two observations that are close to each other do not necessarily represent two taxels that are close to each other in the artificial skin. Therefore this representation cannot be used to construct interest models by exploration in observation space. This option was successfully used to construct NN-based sensorimotor model using motor babbling.

**Incidence graph.** The structure of artificial skin is encoded as an incidence graph, where nodes of the graph represent artificial skin taxels, and nodes representing neighboring taxels are connected by an edge (Fig. 3.6a). The dimensionality of observation space in this case also equals the number of taxels $\mathbf{T}$:

$$\mathbf{X}_I = \{0, 1\}^{\mathbf{T}} \tag{3.5}$$

However, as distance between two points in observation space can be extracted from the graph using Dijkstra's algorithm [8], this option can be used to construct interest models using goal babbling and other efficient exploration strategies in observation space.




**(a) :** Incidence graph

**(b) :** Planar projection

**Figure 3.6:** Observation spaces for artificial skin on robot's torso.

**Planar projection.** Taxels are projected onto a flat surface, and the coordinates of their planar projections are used as points in continuous observation space. A separate two-dimensional observation subspace is used for every body part. If several taxels on a single body part are activated as a result of robot executing an action, the centroid of their planar projections is calculated. Let $\mathbf{B}$ be the number of body parts considered in the experiment. The observation subspace is two-dimensional for every body part, hence the observation space is:

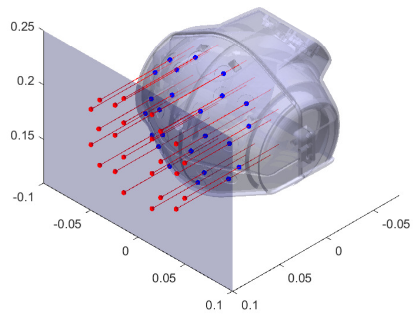
$$\mathbf{X}_P \subseteq \mathbb{R}^{2 \times \mathbf{B}} \tag{3.6}$$

Observation subspace for every body part is centered at the origin $(0, 0)$. This option is the most beneficial:

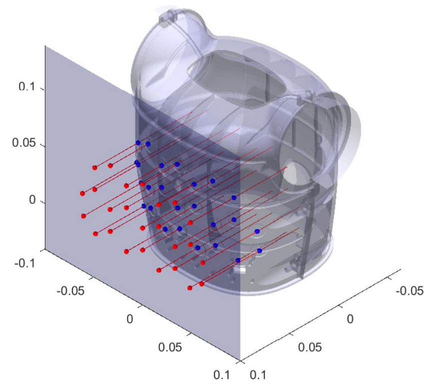
- Dimensionality of the observation space remains low as it does not depend directly on amount of taxels or area of the artificial skin.
- Euclidean distances between points in observation space provide meaningful information.
- Interest models can be constructed naturally in explauto framework using goal babbling and other efficient exploration strategies in observation space.

Two methods of projecting artificial skin taxels onto a flat surface were used in this work:

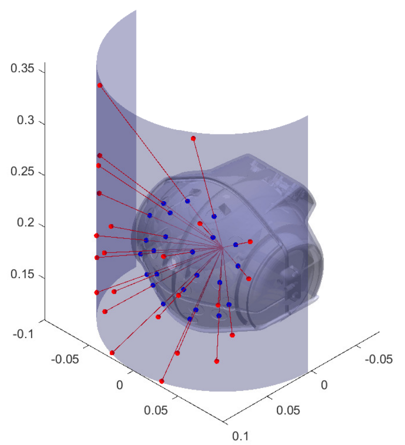
- Parallel projection onto a plane (Fig. 3.7a, 3.7b) was used for low resolution artificial skin.
- Central projection onto a cylindrical surface (Fig. 3.7c, 3.7d) was used for high resolution artificial skin.



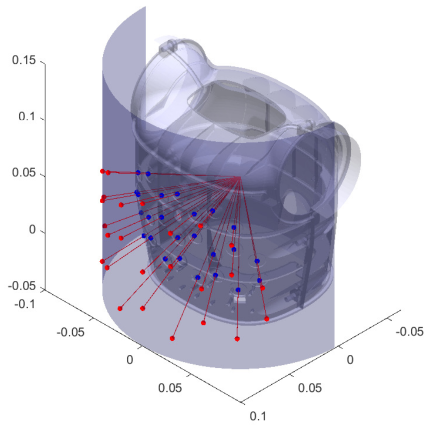

**(a) :** Parallel projection of artificial skin taxels on robot's head onto a plane




**(b) :** Parallel projection of artificial skin taxels on robot's torso onto a plane

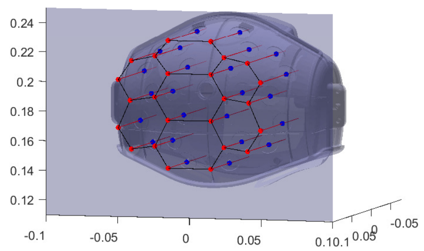



**(c) :** Central projection of artificial skin taxels on robot's head onto a cylindrical surface

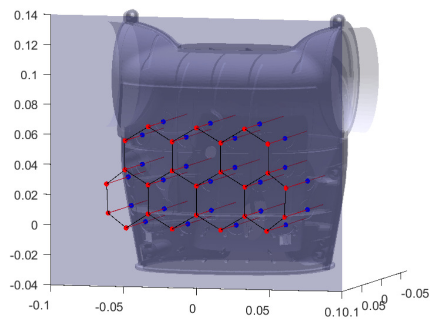



**(d) :** Central projection of artificial skin taxels on robot's torso onto a cylindrical surface

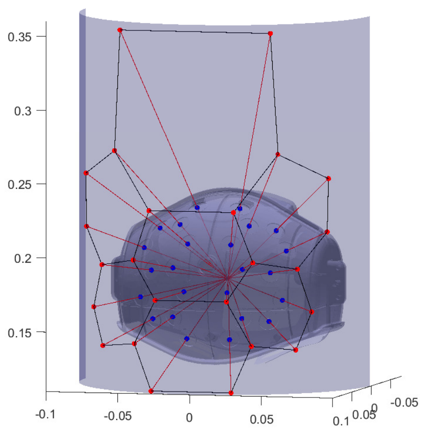
**Figure 3.7:** Projection representations of artificial skin taxels.



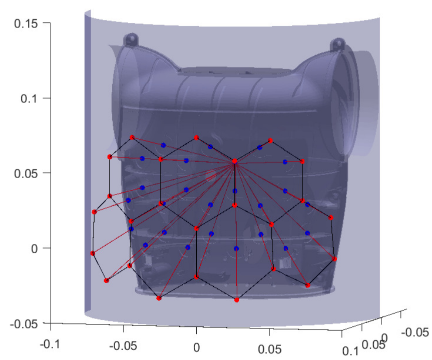

**(a) :** Parallel projection of artificial skin taxels on robot's head onto a plane




**(b) :** Parallel projection of artificial skin taxels on robot's torso onto a plane




**(c) :** Central projection of artificial skin taxels on robot's head onto a cylindrical surface




**(d) :** Central projection of artificial skin taxels on robot's torso onto a cylindrical surface

**Figure 3.8:** Projection representations of artificial skin taxels (continued).

### 3.4.3 Forward and inverse models

#### Forward models available in explauto framework

**Forward model** $\hat{f}$ approximates the forward function $f$ and predicts the outcome of an action [26, p.9]:

$$\hat{f}(q) = \hat{x} \tag{3.7}$$

**Nearest neighbor (NN).** Given a motor command $q \in \mathbf{Q}$, the NN forward model scans the database and returns observation $x \in \mathbf{X}$ that corresponds to motor command that is closest to $q$ [9].

**Weighted nearest neighbor (WNN).** Given a motor command $q \in \mathbf{Q}$, the WNN forward model finds its $k$ nearest neighbors in the database and returns a weighted average of their corresponding observations $x \in \mathbf{X}$, where weight coefficients are proportional to Euclidean distance between $q$ and database entries [9]. Here $k$ is a parameter of the model.

**Locally weigthed linear regression (LWLR)** [1, 9]. 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 of their corresponding observations $x \in \mathbf{X}$. Here $k$ and $\sigma^2$ 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:

$$\begin{aligned} w_i' &= e^{-\frac{d_i^2}{2\sigma^2}} \\ w_i &= \frac{w_i'}{\sum_{j=1}^{k} w_j'} \end{aligned} \tag{3.8}$$

### Inverse models available in explauto framework

**Inverse model** suggests an action necessary to produce a desired outcome in observation space [26, p.9]:

$$\hat{f}^{-1}(x^*) = \hat{q} \tag{3.9}$$

**Nearest neighbor (NN).** Given an observation $x \in \mathbf{X}$, the NN inverse model scans the database and returns motor command $q \in \mathbf{Q}$ that corresponds to observation that is closest to $x$ [9].

**Weighted nearest neighbor (WNN).** Given an observation $x \in \mathbf{X}$, the WNN inverse model finds its $k$ nearest neighbors in the database and returns a weighted average of their corresponding motor commands $q \in \mathbf{Q}$, where weight coefficients are proportional to Euclidean distance between $x$ and database entries [9]. Here $k$ is a parameter of the model.

**Optimization inverse models.** Given an observation $x \in \mathbf{X}$, these inverse models use optimization algorithm to return motor command $q$ that minimizes error [9]:

$$e(q) = \left\| \hat{f}(q) - x \right\|^2 \tag{3.10}$$

Several optimization algorithms are available in explauto framework:

- **COBYLA** (Constrained Optimization by Linear Approximation) is a numerical optimization method designed for constrained problems where the derivative of objective function is not known. This optimization algorithm works by constructing linear polynomial approximations to the objective and constraint functions [24].
- **BFGS** (Broyden–Fletcher–Goldfarb–Shanno algorithm) is a quasi-Newtonian optimization method that uses iterative approximations of the Hessian matrix to find stationary points of the objective function [5, 10, 13, 28].

- **L-BFGS-B** (Limited-memory BFGS with bound constraints) [6] is a variation of BFGS algorithm that is adapted for limited computer memory and is capable of handling constraints on variables of the following form:

$$l_i \leq x_i \leq u_i \tag{3.11}$$

- **CMA-ES** (Covariance Matrix Adaptation - Evolutionary Strategy) is an evolutionary optimization algorithm that adapts arbitrary, normal mutation distributions within a completely derandomized adaptation scheme [15].

### 3.4.4 Exploration strategies

#### Random motor babbling

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 database is updated. This exploration strategy is the most naive and least effective method for learning forward and inverse models.

#### Random goal babbling

**Goal babbling** is the bootstrapping of a coordination skill by repetitively trying to accomplish multiple goals related to that skill [26, p.23]. In random goal babbling, a goal $x \in \mathbf{X}$ is sampled uniformly from the observation space. Robot then uses inverse model learnt so far to execute action $q \in \mathbf{Q}$.

#### Goal babbling with direct optimization

In random goal babbling, robot executes a single motor command $q \in \mathbf{Q}$ after selecting a goal. However, it is possible to improve this single movement by performing an optimization process. A temporary surrogate forward model is created and optimized in vicinity of the chosen goal for a maximum of $n_{max}$ iterations. After the optimization process is completed, resulting motor command $q_{opt}$ is inserted into the main model database.

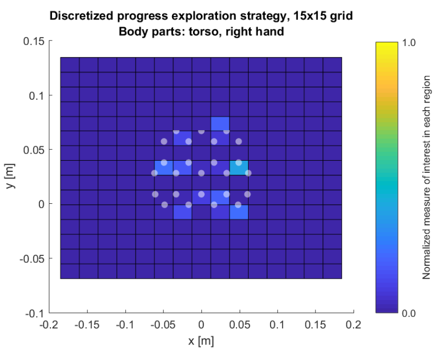
Here $n_{max}$ is a parameter of the strategy. This strategy uses CMA-ES optimization method and LWLR surrogate forward model. The main drawback of this strategy is additional time required to perform optimization process.

### Discretized progress

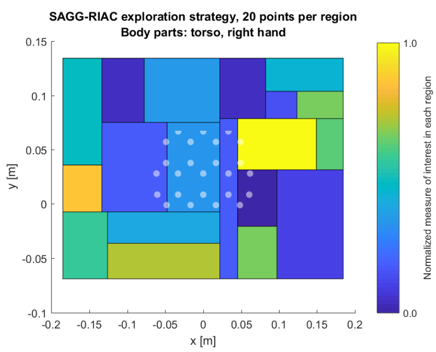
Interest space is statically discretized into $x_{card}$ cells. A cell for goal generation is selected randomly, with probability proportional to current value of interest in each cell. After the cell is selected, a goal $x_g \in \mathbf{X}$ is generated randomly inside the selected cell. Here $x_{card}$ is a parameter of the strategy.

Interest value $I$ is computed as the absolute value of derivative of competence $C$ for each cell [9]. This means that interest value of a cell is high when competence either rapidly increases or rapidly declines. Distance $d$ between selected goal $x_g$ and actual observation $x$ is used as the measure of competence:

$$
\begin{aligned}
C &\equiv d = \|x_g - x\| \\
I &= \left|\frac{dC}{dt}\right|
\end{aligned}
\tag{3.12}
$$




**(a) :** Discretized progress, 15x15 grid. Iteration number: 250




**(b) :** SAGG-RIAC, max 20 points per region. Iteration number: 250

**Figure 3.9:** Exploration strategies with discretization of interest space. Warmer colors indicate regions with higher value of interest.

### SAGG-RIAC

**SAGG-RIAC** (Self-Adaptive Goal Generation - Robust Intelligent Adaptive Curiosity) is an intrinsically motivated goal exploration mechanism which allows active learning of inverse models in high-dimensional redundant robots [3]. This exploration strategy relies on dynamic discretization of interest space. When a goal is generated inside a region of observation space, a total number of goals in that region is checked. If the total number of goals exceeds value of parameter $n_{max}$, the region is split into two sub-regions along alternating axes, analogous to k-d tree [4]. The splitting is performed in a way that allows to maximally discriminate sub-regions according to their levels of interest.

### 3.4.5 Handling motor commands that produce no observation

Motor commands that produce no observation constitute a significant portion of robot's action space. Such motor commands produce no contact between robot's end effector and the surface of artificial skin. In this work, I have tested two strategies of handling such motor commands:

1. If a motor command $q \in \mathbf{Q}$ produces no observation, ignore it and do not update the model database.
2. If a motor command $q \in \mathbf{Q}$ produces no observation, create a virtual observation $x_v \in \mathbf{X}$, set all its coordinates to infinity and update model database with a tuple $(q, x_v)$.

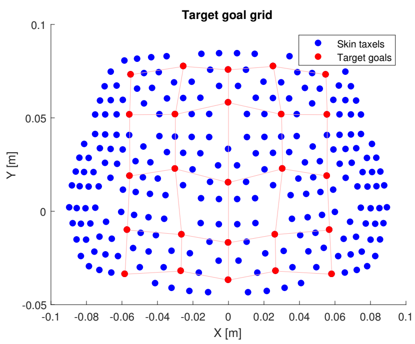
It may seem that the second strategy is more robust, because it provides larger amount of information to the model. Experiments show that choice of strategy for handling motor commands that produce no observation have negligible effect on the quality of learnt model. However, first strategy requires less memory to store the model database.

# Chapter 4

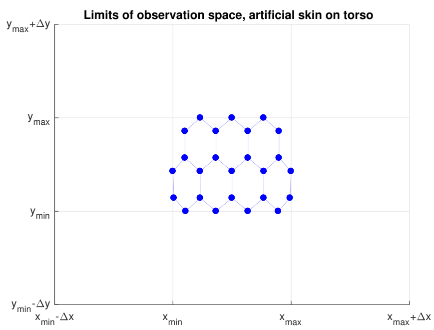
# Results

## 4.1 Experimental design

A series of experiments was performed to estimate the quality of models learnt with explauto library. In each performed experiment, the simulation was executed for 500 iterations. In each iteration, a motor command $q \in \mathbf{Q}$ was obtained either directly (motor babbling) or using one of goal-based exploration strategies. The motor command was executed in the simulation environment, a corresponding observation $x \in \mathbf{X}$ was obtained and model database was updated with tuple $(q, x)$.




**(a) :** Grid of target goals used to estimate the quality of a model




**(b) :** Limits of observation space

**Figure 4.1:** Details of experimental design, artificial skin on robot's torso.

In experiments involving exploration strategies based on goal babbling, observation space $\mathbf{X}$ was limited to a region with area 9 times larger than the area of artificial skin (Fig. 4.1b):

$$\begin{aligned}\Delta x &= x_{max} - x_{min} \\ \Delta y &= y_{max} - y_{min} \\ \mathbf{X} &= \langle x_{min} - \Delta x, x_{max} + \Delta x \rangle \times \langle y_{min} - \Delta y, y_{max} + \Delta y \rangle\end{aligned} \tag{4.1}$$

As suggested in [26, p.34], exploratory noise was added to motor commands executed during learning process. Without added noise, goal-driven exploration may produce degenerated data sets and get stuck in intermediate local minima. I have used Gaussian noise with $\sigma = 0.05$. No exploratory noise was added when testing the quality of learnt models.

### 4.1.1 Progress Evaluation

Every 10 iterations, exploration process was paused and the quality of model learnt so far was estimated. Results were recorded for further analysis. The following procedure was used to estimate the quality of a model:

1. A $5 \times 5$ grid of goals was generated and fed to the model (Fig. 4.1a). These target goals were positioned at the coordinates of some artificial skin taxels, in a pattern that resembles a slightly distorted rectangular grid.
2. The model attempted to reach target goals, and reaching error for each goal was recorded. Mean reaching error computed over all target goals was used as an empirical measure of model quality. For high quality models this empirical measure would approach 0.

## 4.2 Comparison of inverse models

Experiments showed that out of the three inverse models provided by explauto framework, **nearest neighbor (NN)** inverse model was the most precise. NN inverse model guarantees that the motor command $q \in \mathbf{Q}$ inferred by the model will produce some observation $x \in \mathbf{X}$ on artificial skin.

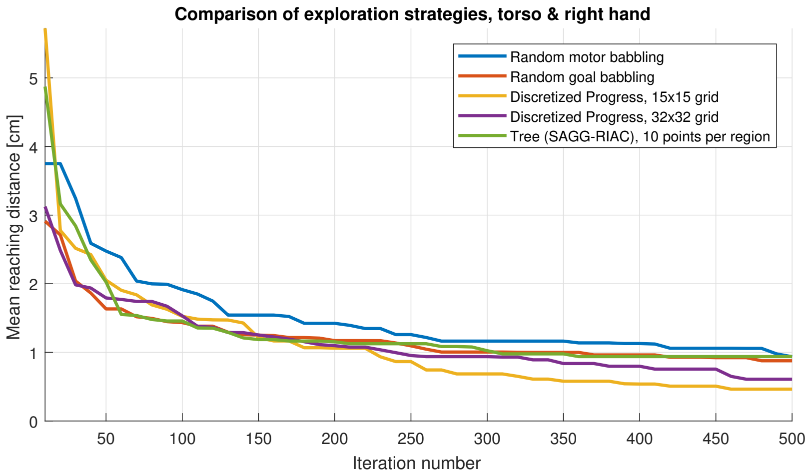


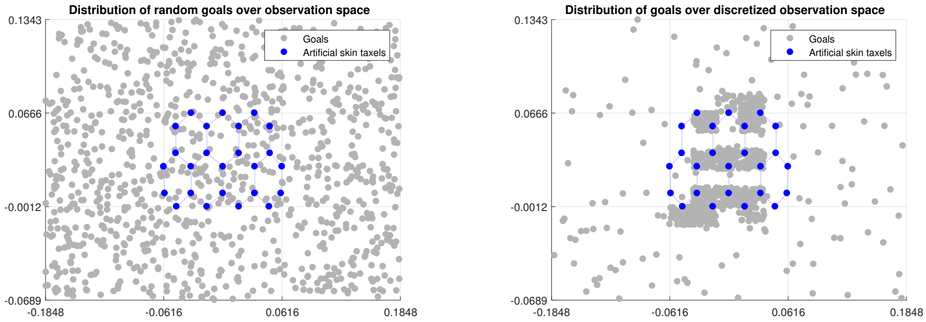

**Figure 4.2:** Comparison of exploration strategies.

Other available inverse models use weighted sums or linear regression to infer motor commands. Execution of motor command inferred in such manner may produce no contact with artificial skin and hence no observation whatsoever. These models may be well suited for environments where the position of robot's end effector in 2- or 3-dimensional Euclidean space is observed directly, e.g. as in [26, p.57] or [21]. However, in environments where observations are limited exclusively to sensation of touch, usage of NN inverse model may be preferred.

## 4.3 Comparison of exploration strategies

Fig. 4.2 illustrates how mean reaching error to target goals gradually drops during the exploration process. The rate of decrease is higher at the beginning of exploration process. In this phase, the discovered artificial skin taxels are those that are easy to reach, and hence have higher probability of discovery. After about 250 iterations, the exploration process with random motor babbling and random goal babbling strategies virtually stops, new taxels are discovered rarely by pure chance. On the contrary, exploration strategies with discretization of observation space continue to discover new taxels by focusing exploration in regions of higher interest.

Fig. 4.3 illustrates the advantage of advanced goal-based strategies over random goal babbling. Goals generated with random goal babbling (Fig. 4.3a) are distributed uniformly over observation space $\mathbf{X}$. Exploration does not take advantage of the fact that only part of the observation space is reachable.



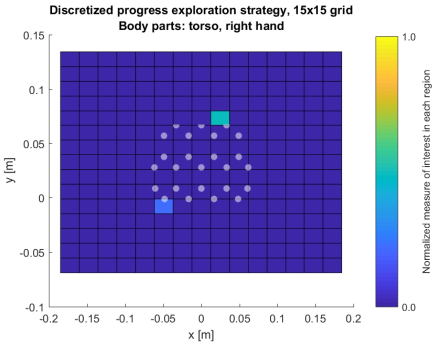

**(a) :** Random goal babbling **(b) :** Discretized progress, 15x15 grid

**Figure 4.3:** Distribution of goals generated in experiments involving exploration strategies based on goal babbling.

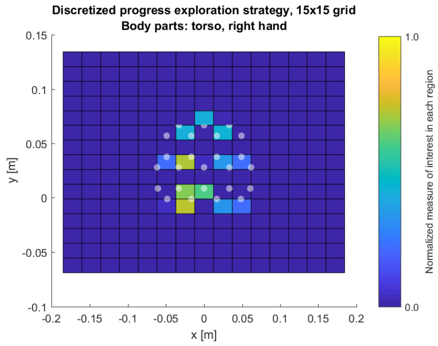
On the contrary, the majority of goals generated with advanced exploration strategies like discretized progress (Fig. 4.3b) are located in regions of high interest. Some taxels on the robot's artificial skin are not discovered after 500 iterations of exploration process. The reason for this is purely kinematic – these taxels are harder for the robot to reach.

## 4.4 Discretized progress exploration strategy

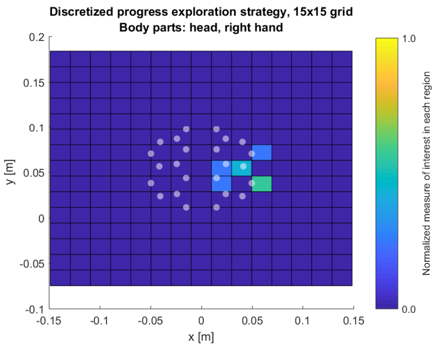
Experiments showed that discretized progress exploration strategy is very effective for exploration of artificial skin and learning the model of robot's body. This strategy divides observation space into a fixed grid of rectangular regions. I have tested two discretization sizes, with 15x15 (Fig. 4.4) and 32x32 (Fig. 4.5) grid. Regardless of the size of discretization, this strategy conducted effective exploration and produced good quality model with mean reaching error to target goals of about 5 mm.



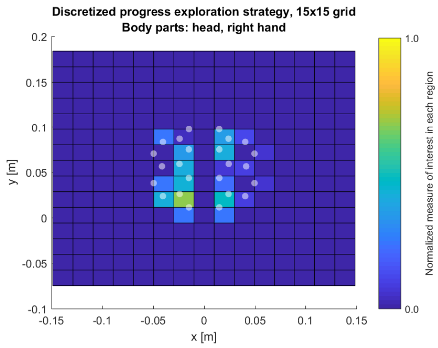

**(a) :** Body parts: torso, right hand. Iteration number: 100




**(b) :** Body parts: torso, right hand. Iteration number: 250

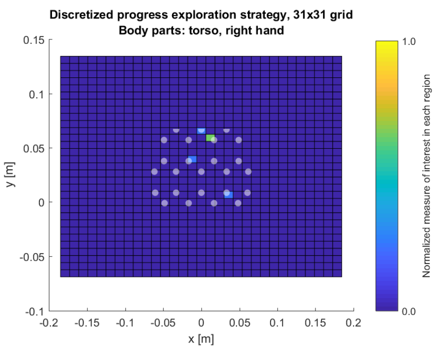



**(c) :** Body parts: head, right hand. Iteration number: 100

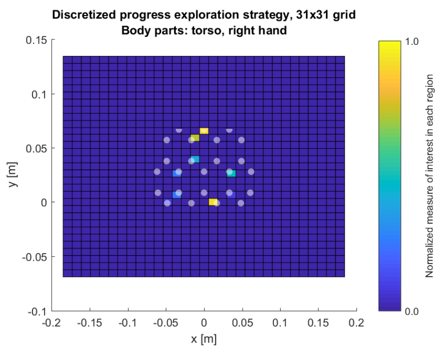



**(d) :** Body parts: head, right hand. Iteration number: 250

**Figure 4.4:** Fixed discretization of observation space with discretized progress exploration strategy, 15x15 grid. Warmer colors indicate regions with higher value of interest.




**(a) :** Body parts: torso, right hand. Iteration number: 100

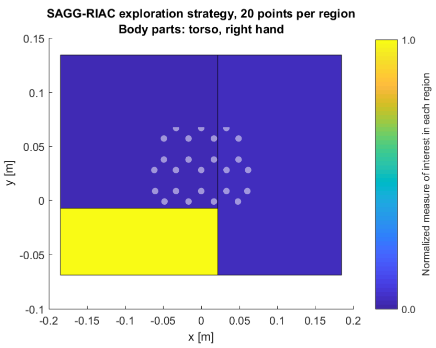



**(b) :** Body parts: torso, right hand. Iteration number: 200

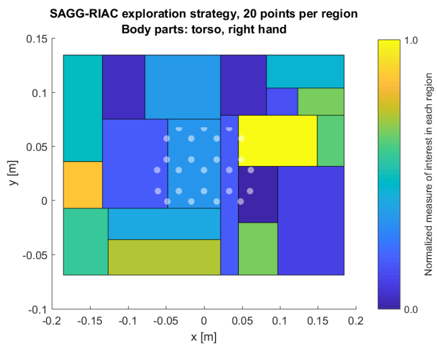
**Figure 4.5:** Fixed discretization of observation space with discretized progress exploration strategy, 32x32 grid.

## 4.5 SAGG-RIAC exploration strategy

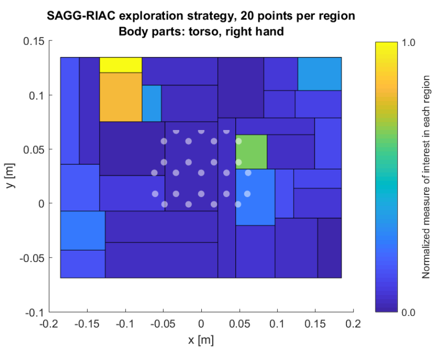
Contrary to discretized progress, this exploration strategy divides observation space dynamically. This method did not work well with my experimental design, resulting in body models of pure quality. For some reason, this strategy assigned high values of interest during active phase of exploration to regions that contained no taxels, resulting in performance close to that of random goal babbling. This phenomenon requires further investigation.



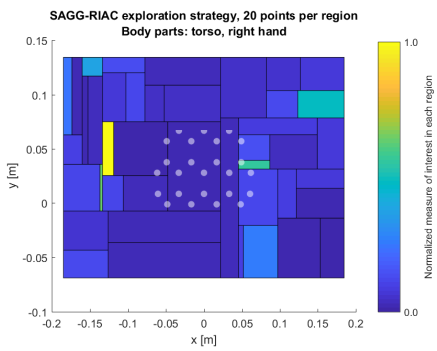

**(a)** : Observation space after first two divisions. Iteration number: 80




**(b)** : Active phase of exploration process. There are several regions of high interest. Iteration number: 250




**(c)** : By the end of exploration process, almost no new information can be obtained. Iteration number: 400




**(d)** : Final division of observation space. Iteration number: 500

**Figure 4.6:** Division of observation space into sub-regions with SAGG-RIAC [3] exploration strategy (implemented in explauto library [9] as tree interest model). Warmer colors indicate regions with higher value of interest.

# Chapter 5

# Conclusion, Discussion, Future Work

## 5.1 Conclusion

This work consisted of two primary parts:

1. Modelling of artificial skin on NAO humanoid robot and building simulation environment for the experiments.
2. Performing experiments with self-exploration and learning of robot's body models.

First part of this work was completed in full. Two versions of artificial skin models were created:

1. Low-resolution model of artificial skin with fewer taxels
2. High-resolution model of artificial skin that closely resembles the arrangement of artificial skin on the real NAO robot

The simulation environment based on ROS Melodic and Gazebo 9 was built. It included Gazebo plugins for processing skin contact events, ROS topics and services for communication between simulation environment and exploration libraries. Explauto library developed in the **Inria FLOWERS** research team [23] was used for the purposes of exploration and model learning.

A series of experiments was performed and their results analyzed. The experimental results confirmed that goal strategies based on discretization of observation space outperform motor babbling strategies.

Out of all possibilities for representing forward and inverse model of robot's body available in explauto library, nearest neighbor (NN) model provided best results. Reasons for and implications of this are discussed in section 5.2.

All program codes and data files created during this project are available online at the CTU Faculty of Electrical Engineering GitLab repository [12]. A Youtube channel was created [30] for demonstration purposes. Youtube channel contains recorded videos of exploration process along with informative data visualization overlays.

## 5.2 Discussion

Explauto framework is based on the notion of lazy learning. Essentially the library keeps a database of motor actions and corresponding observations. No explicit model is created that can be represented with mathematical function or formula. Instead, to answer a request for forward or inverse prediction, the model queries the maintained database.

The biggest drawback of this approach is computational time and space complexity. The size of the database grows linearly with the number of samples. With every new request for forward or inverse prediction, the model must perform new search and possible regression computation. Although explauto stores data effectively using k-d tree data structure [4], such method is still time consuming.

The experiments have shown that nearest neighbor (NN) inverse model in explauto framework provides best results in the quality of learnt models. There are several reasons for that:

- The only source of sensory feedback in the experiments was the artificial skin. It was therefore very important to maintain contact between end effector tool and the surface of artificial skin. NN model guarantees that there will be contact with artificial skin (for a bootstrapped model that contains at least one entry).
- Other models available in explauto framework (WNN, LWLR) rely on interpolation and regression of motor commands. Kinematic chain in robot's arm used for self-touch is redundant, therefore a combination of different poses for reaching same taxel may produce wrong result.

Nearest neighbor model would perform poorly in noisy environments, where false activation of taxels could be possible, e.g. by insects landing on the surface of artificial skin. Once a faulty action-observation tuple is entered into the database, it remains there indefinitely and affects the result of future model predictions.

## 5.3 Future work

In this work I have considered outputs from simulated artificial skin taxels as binary variables (on/off). However, the real physical artificial skin is capable of producing outputs as 8-bit unsigned integers. The range of output values could be utilized during learning process to estimate strength of contact with artificial skin and to improve the quality of learnt models.

All robot configurations that produce no contact with artificial skin are mapped to a single point in observation space. This is a big drawback that may complicate the learning process. One possible solution to this problem would be to incorporate visual feedback either from robot's own cameras or from external cameras. Another possible solution would be to utilize temperature sensors within the artificial skin and to use end effector tool with heating. This could potentially enable the robot to feel the end effector positioned in close proximity to the surface of artificial skin but not directly touching it.

Another possibility for efficient self-exploration could be to use continuous sliding micro motions along the surface of artificial skin. In such experimental setup, broken contact with artificial skin could be resolved either by reverting the motion several steps back and continuing it in another direction, or by resetting and restarting the simulation from the home posture. Outputs from artificial skin taxels could be used to calculate contact strength, which in turn could be fed into a feedback controller that would help produce constant contact strength and smooth exploration of entire robot's body.

It would be worthwhile to implement and test other learning methods, e.g. reinforcement learning and convolutional neural networks, and compare obtained models with models produced in this work using explauto framework, lazy learning and exploration strategies based on goal babbling.

In this work, I have made as few assumptions about the structure of robot's kinematic chain as possible. Only the number of joints was required so that sensorimotor system could execute motor commands. A possibility for future work is to impose the structure of robot's kinematic chain and perform learning by tuning parameters of the model. For example, one could assume open kinematic chain, describe it in terms of Denavit-Hartenberg parameters [16] and try to learn these parameters.

In my experiments, I have primarily used planar projection of artificial skin taxels onto a flat surface as my observation space. Another possibility for representation of artificial skin is somatosensory robotic homunculus [18]. Much like a planar projection, it also is a distorted 2D representation of artificial skin. However, this representation can be learned with self-organizing map (SOM) algorithm. A combination of this artificial skin representation with exploration strategies from explauto library could produce interesting results.

# 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] 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.

[3] 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.

[4] Bentley, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM 18*, 9 (1975), 509–517.

[5] Broyden, C. G. The convergence of a class of double-rank minimization algorithms: 2. the new algorithm. *IMA journal of applied mathematics 6*, 3 (1970), 222–231.

[6] Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing 16*, 5 (1995), 1190–1208.

[7] Corbetta, D., Wiener, R. F., Thurman, S. L., and McMahon, E. The embodied origins of infant reaching: implications for the emergence of eye-hand coordination. *Kinesiology Review 7*, 1 (2018), 10–17.

[8] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische mathematik 1*, 1 (1959), 269–271.

[9] Explauto. https://github.com/flowersteam/explauto. [Online; accessed 13-05-2019].

[10] Fletcher, R. A new approach to variable metric algorithms. *The computer journal 13*, 3 (1970), 317–322.

[11] Gazebo. http://gazebosim.org/. [Online; accessed 11-03-2019].

[12] GitLab Code Repository. https://gitlab.fel.cvut.cz/body-schema/code-nao-simulation/tree/master/gazebo9. [Online; accessed 13-05-2019].

[13] Goldfarb, D. A family of variable-metric methods derived by variational means. *Mathematics of computation 24*, 109 (1970), 23–26.

[14] Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. Mechatronic design of NAO humanoid. In *2009 IEEE International Conference on Robotics and Automation* (2009), IEEE, pp. 769–774.

[15] Hansen, N., and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation 9*, 2 (2001), 159–195.

[16] Hartenberg, R. S., and Denavit, J. A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics 77*, 2 (1955), 215–221.

[17] Hoffmann, M., Chinn, L. K., Somogyi, E., Heed, T., Fagard, J., Lockman, J. J., and O'Regan, J. K. Development of reaching to the body in early infancy: From experiments to robotic models. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (2017), IEEE, pp. 112–119.

[18] Hoffmann, M., Straka, Z., Farkaš, I., Vavrečka, M., and Metta, G. Robotic homunculus: Learning of artificial skin representation in a humanoid robot motivated by primary somatosensory cortex. *IEEE Transactions on Cognitive and Developmental Systems 10*, 2 (2018), 163–176.

[19] Kofinas, N. Forward and inverse kinematics for the NAO humanoid robot. *Technical University of Crete* (2012).

[20] Maiolino, P., Maggiali, M., Cannata, G., Metta, G., and Natale, L. A flexible and robust large scale capacitive tactile system for robots. *IEEE Sensors Journal 13*, 10 (2013), 3910–3917.

[21] 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.

[22] Moulin-Frier, C., and Oudeyer, P.-Y. Exploration strategies in developmental robotics: a unified probabilistic framework. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)* (2013), IEEE, pp. 1–6.

[23] 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.

[24] Powell, M. J. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*. Springer, 1994, pp. 51–67.

[25] ROS Melodic. `http://www.ros.org/`. [Online; accessed 13-05-2019].

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

[27] Roncone, A., Hoffmann, M., Pattacini, U., and Metta, G. Automatic kinematic chain calibration using artificial skin: self-touch in the icub humanoid robot. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), IEEE, pp. 2305–2312.

[28] Shanno, D. F. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation 24*, 111 (1970), 647–656.

[29] von Hofsten, C. *Development of visually directed reaching: The approach phase*. Department of psychology, University of Uppsala, 1979.

[30] Youtube channel: Efficient self-exploration. `https://www.youtube.com/channel/UCv7Rc3MT8jR3cUhCBgcBe7A`. [Online; accessed 13-05-2019].

# BACHELOR‘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**

## II. Bachelor’s thesis details

Bachelor’s thesis title in English:

**Efficient Self-exploration and Learning of Forward and Inverse Models on a Humanoid Robot with Artificial Skin**

Bachelor’s thesis title in Czech:

**Efektivní sebe-explorace a tvorba modelu těla u humanoidního robota s umělou kůží**

Guidelines:

1. Development of simulation framework with Nao humanoid and artificial skin (collision detection) in Gazebo simulator. Experimentation with libraries for efficient exploration (e.g., https://github.com/flowersteam/explauto).
2. Study the literature on efficient exploration of search spaces, in particular motor vs. goal babbling, active learning and possibly reinforcement learning.
3. Application of a selection of these methods on the simulator, seeking an efficient exploration of the robot’s own body, in particular using self-touching behaviors. The input space consists of the robot’s motors, the goal space will be the robot’s artificial skin.
4. Explore different methods for constructing forward and inverse models (Nearest Neighbor, weighted NN, Locally Weighted Linear Regression, etc.)
5. Investigate or discuss the effect of different problem formulations, in particular the amount of prior knowledge about the input and goal spaces (e.g. metric on the 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] Hoffmann, Matej, et al. "Development of reaching to the body in early infancy: From experiments to robotic models." IEEE International Conference on Development and Learning and Epigenetic Robotics. IEEE, 2017.
[3] Mannella, Francesco, et al. "Know your body through intrinsic goals." Frontiers in neurorobotics 12 (2018): 30.
[4] Rolf Matthias, Steil Jochen J., and Gienger Michael. "Goal babbling permits direct learning of inverse kinematics." IEEE Transactions on Autonomous Mental Development 2(3) (2010): 216-229.

Name and workplace of bachelor’s thesis supervisor:

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

Name and workplace of second bachelor’s thesis supervisor or consultant:

**doc. Ing. Karel Zimmermann, Ph.D., Vision for Robotics and Autonomous Systems, FEE**

Date of bachelor’s thesis assignment: **24.01.2019** Deadline for bachelor thesis submission: **24.05.2019**

Assignment valid until: **20.09.2020**

| | | |
|---|---|---|
| Mgr. Matěj Hoffmann, Ph.D.<br>Supervisor’s signature | doc. Ing. Tomáš Svoboda, Ph.D.<br>Head of department’s signature | prof. Ing. Pavel Ripka, CSc.<br>Dean’s signature |

## III. Assignment receipt

The student acknowledges that the bachelor’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 bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature